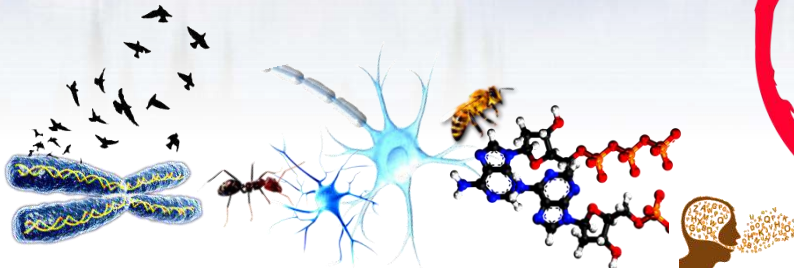


# ■ OPTİMİZASYON ■ TEMEL KAVRAMLAR & YÖNTEMLER

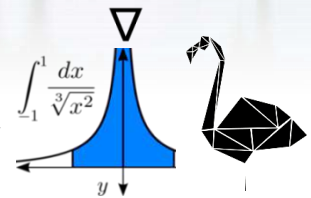
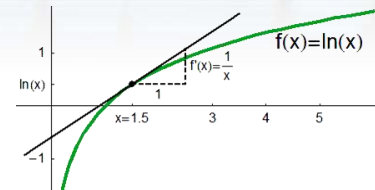
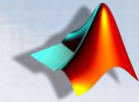
MÜHENDİSLİK  
ÇÖZÜMLERİ



Doç. Dr.  
Y. Volkan PEHLİVANOĞLU



MATLAB  
UYGULAMALARI





Optimizasyon: Temel Kavramlar & Yöntemler  
1'inci Baskı  
Ankara - 2017  
© Y. Volkan Pehlivanoglu





# TEMEL KAVRAMLAR OPTİMİZASYON YÖNTEMLER



Y. Volkan Pehlivanoglu  
Ankara 2017

## Önsöz

**T**

ürkçe'ye sonradan ilave edilen optimizasyon veya diğer bir ifade ile eniyileme kelimesinin kökeni, Latince'deki "optimas" sözcüğünden gelir. Bu kelime, kaynağı itibari ile "Ops" ifadesine bağlanmakta olup, tarımda bereket ve bolluğu temsil etmektedir. "Ops", Yunan mitolojisindeki "Rhea" mitinin Roma medeniyetindeki karşılığıdır. "Rhea",

Anadolu topraklarında somutlaştırılan ve genel manada dağların ve doğanın miti olan "Cybele" isimli Frigya medeniyeti tanrıçasına dayanır. Bu bağlamda eniyileme kavramının, bereket ve bolluk kavramları ile örtüşürülerek Anadolu topraklarında filizlendiği söylenebilir. Hayatımızın hemen hemen her evresinde özgür irademize dayanarak bir takım seçimler yapmaktayız. Söz konusu seçimleri yaparken kimi zaman içsel isteklerimizi, kimi zamansa çevresel etkileri göz önünde bulunduruyoruz. Bununla beraber eylemlerimizde, mevcut şartlar altında yapabileceğimizin en iyisini hedefleriz. Optimizasyon kavramı, en iyiyi belirlemede dikkate alınan süreçler üzerinde durmakta olup, yazarı olduğumuz kitapta bahse konu kavram değerlendirilmekte ve okuyucuya belirli bir derinlikte ama etraflıca bilgiler verilmektedir. Temel kavramlara ilave olarak kitabın yoğunlaştığı konular mühendislik uygulamaları bağlamındaki yöntemlerdir. Bu amaçla deterministik ve sezgisel yöntemler örneklerle anlatılmakta, yer yer yazılım uygulamaları ile konular pekiştirilmektedir. Hemen her mühendislik disiplininin kendisine bir uygulama alanı bulabileceği "Optimizasyon: Temel Kavramlar & Yöntemler" isimli eserin okuyan herkese faydalı olmasını dilerim...

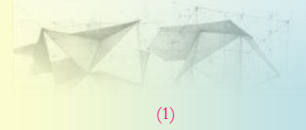
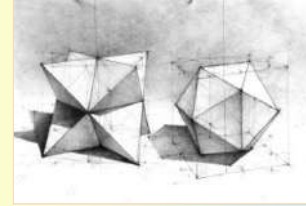




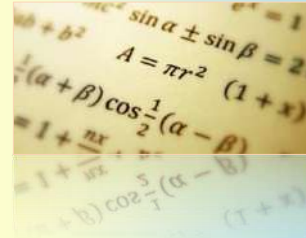
## TARİHSEL GELİŞİM & TEMEL KAVRAMLAR

- 1.1 Eniyileme kavramı ve tarihsel gelişim 2
- 1.2 Matematiksel temeller 10
  - 1.2.1 Geometrik kavramlar 10
    - a. Nokta 10
    - b. Doğru ve doğru parçası 11
    - c. Hiperdüzlem ve yarımuzay 11
    - ç. Çokkenar 12
    - d. Eliptik eğriler 13
    - e. Komşuluk ilişkisi 13
    - f. İlgin ve dışbükey kümeler 14
    - g. Kümede sıralama 16
    - ğ. Kümelerde sıralama 18
  - 1.2.2 Cebir 18
    - a. Etiketleme sembolleri ve dizinler 19
    - b. 0-Dizinli sayılar 19
    - c. 1-Dizinli sayılar 20
      - (1) Yöney 21
      - (2) Dördey 23
      - (3) 4-Yöney 24
    - ç. 2-Dizinli sayılar 24
      - (1) Doğrusal denklem takımlarının çözümü 27
        - (a) Doğrusal bağımsızlık 30
        - (b) Tutarlılık 32
        - (c) Sayısal hassasiyet 32
      - (2) Kare matris 33

1



(1)



(2)



- (3) 0-Matrisi 33
  - (4) Köşegen elemanlar 34
  - (5) I-Matrisi ve birim yöney 34
  - (6) Matrisin devriği 35
  - (7) Matriste simetri 36
  - (8) Matrisin belirteci 36
  - (9) Matrisin normu 39
  - (10) Matrislerde toplama ve çıkarma 40
  - (11) Matrislerde çarpma 41
  - (12) Matrislerde bölme 44
  - (13) Doğrusal denklem takımlarında çözüm hassasiyeti 47
  - (14) Matrislerde dikeygenlik 49
  - (15) Matrislerde özdeğer ve özyöney 51
- d. İkilenik form 54
- e. Doğrusal denklem takımlarında çözüm yöntemleri 62
- 1.2.3 Analiz 63
- a. Fonksiyon 63
  - b. Limit 64
  - c. Süreklilik 66
  - ç. Değişim ve türevlenebilirlik 67
  - d. Temel bazı türev alma yöntemleri 74
  - e. Taylor serisi 75
  - f. Gradyan kavramı 78
  - g. Birincil ve ikincil türev matrisleri 80
  - ğ. Analizin en temel önermesi 84
    - (1) İntegral 84
    - (2) Önermenin ilk kısmı 86
    - (3) Önermenin ikinci kısmı 87
    - (4) İntegral ve türev arasındaki ilişki 87
    - (5) Temel bazı integral alma yöntemleri 90



(3)



(4)

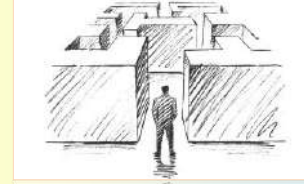


- h. Rastsal sayılar 94
- 1.2.4 Analizde sayısal yöntemler 96
- a. Sayısal türev yöntemleri 97
- b. Sayısal integral yöntemleri 99
- c. Örneklemeye 101
- (1) Sistematik örneklemeye yöntemleri 101
- (a) Doğrusal örneklemeye 102
- (b) Etmensel örneklemeye 102
- (c) Merkezi karışım örneklemeye 103
- (ç) Latin çoklu küp tasarımı 104
- (d) Doğrusal olmayan örneklemeye 105
- (e) Taguchi örneklemeye dizinleri 106
- (2) Rastsal örneklemeye yöntemleri 107
- (3) Karma örneklemeye 108
- 1.2.5 Modelleme yöntemleri 109
- 1.3 Eniyileme sürecinde temel kavramlar 110
- 1.3.1 Tasarım yöneyi 111
- 1.3.2 Amaç fonksiyonu 120
- 1.3.3 Tasarım kısıtları 124
- 1.3.4 Hal denklemleri 131
- 1.3.5 Optimal değerin tanımı 133
- 1.3.6 Eniyileme problemlerinin sınıflandırılması 138
- 1.3.7 Eniyileme problemlerinde zorluk derecesi 159
- 1.3.8 Eniyileme sürecinde yakınsama 162
- 1.3.9 Eniyileme yöntemlerinin sınıflandırılması 163

Bilim Damlası 165

Yararlı Kitaplar 171

Kaynakça 172



(5)



(6)

## DETERMİNİSTİK YÖNTEMLER

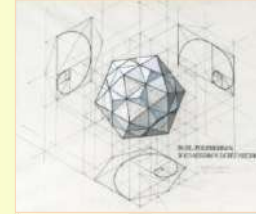
- 2.1 Giriş 176
- 2.2 Grafik yöntemi 176
- 2.3 Doğrusal programlama yöntemleri 188
  - 2.3.1 “Simplex” yöntemi 193
    - a. Standart forma dönüşüm 194
    - b. “Simplex” tablosu ve pivot işlemleri 197
  - 2.3.2 “Big-M Simplex” yöntemi 205
  - 2.3.3 “Karmarkar” yöntemi 210
  - 2.3.4 “Karmarkar” yönteminin ilgin seçeneği 217
  - 2.3.5 Macar algoritması 222
  - 2.3.6 Eşleniklik 231
- 2.4 Doğrusal olmayan programlama yöntemleri 236
  - 2.4.1 Analitik yöntemler 236
    - a. Analitik yöntemlerde eşitlik kısıtlı eniyileme süreci 242
      - (1) Yerine koyma yöntemi 243
      - (2) “Lagrange” çarpanları yöntemi 244
    - b. Analitik yöntemlerde eşitsizlik kısıtlı eniyileme süreci 252
    - c. Genelleştirilmiş KKT şartları 254
  - 2.4.2 Sayısal yöntemler 257
    - a. Tek boyutlu arama yöntemleri 257
      - (1) Altın kesit yöntemi 261
      - (2) “Fibonacci” yöntemi 270
      - (3) “Quasi-Newton” yöntemi 279
      - (4) “Secant” yöntemi 285
      - (5) “Newton” yöntemi 288

Tarihten Bir Sayfa 291

2



(7)



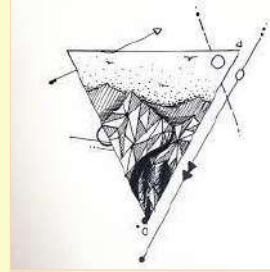
(8)







- b. Çok boyutlu arama yöntemleri 294
- (1) "Powel" yöntemi 295
  - (2) Endik azalma yöntemi 310
  - (3) Eşlenik gradyan yöntemi 319
  - (4) "Newton" yöntemi 326
  - (5) Yaklaşık "Newton" yöntemi 330
- c. Sayısal yöntemlerde kısıtların dikkate alınması 336
- (1) Ardışık doğrusal programlama yöntemi 339
  - (2) İkilenik programlama 346
  - (3) Uygun doğrultu yöntemleri 352
  - (4) Genelleştirilmiş azaltılmış gradyan yöntemi 362
  - (5) Ardışık ikilenik programlama yöntemi 371
  - (6) Ceza fonksiyonu yöntemleri 381
    - (a) İçsel nokta yöntemleri 382
    - (b) Dışsal nokta yöntemleri 385
- İrfan Damlası 390  
Kaynakça 391



(9)



## SEZGİSEL YÖNTEMLER

- 3.1 Giriş 396
- 3.2 “Nelder-Mead” yöntemi 399
- 3.3 Tabu algoritması 408
- 3.4 “Markow” zinciri 423
- 3.5 Rastlantısal tarama yöntemi 426
- 3.6 Benzetimli tavlama yöntemi 437
- 3.7 Karınca kolonisi algoritması 451
- 3.8 Yapay arı sürüsü algoritması 475
- 3.9 Parçacık sürü algoritması 489
- 3.10 Genetik algoritma 508
  - 3.10.1 Elitizm 513
  - 3.10.2 Seçim 514
  - 3.10.3 Çaprazlama 519
    - a. Yeknesak çaprazlama 519
    - b. Tek-nokta çaprazlama 519
    - c. İki-nokta çaprazlama 520
    - ç. Ara-nokta çaprazlama 520
    - d. Sezgisel çaprazlama 520
    - e. Aritmetik çaprazlama 521
    - f. BLX- $\alpha$  çaprazlama 521
  - 3.10.4 Mutasyon 521
    - a. Gauss mutasyonu 522
    - b. Yeknesak mutasyon 523
    - c. Adaptif olası mutasyon 523
    - ç. Titreşimli mutasyon 523

İrfan Damlası 544

Kaynakça 545

3



(10)



(11)

## VEKİL MODEL YÖNTEMLERİ

- 4.1 Giriş 552
- 4.2 Vekil model yaklaşımı 554
- 4.3 Regresyon yöntemleri 561
  - 4.3.1 Tekil doğrusal regresyon 564
  - 4.3.2 Çoğul doğrusal regresyon 567
  - 4.3.3 Çok değişkenli doğrusal regresyon 570
  - 4.3.4 Polinomial doğrusal regresyon 573
  - 4.3.5 Doğrusal olmayan regresyon 576
  - 4.3.6 Regresyonla ilgili ilave hususlar 578
- 4.4 Mesafe ağırlıklı yaklaşım 579
- 4.5 Kriging yöntemi 586
  - 4.5.1 Sıradan Kriging yöntemi 594
  - 4.5.2 Basit Kriging yöntemi 614
  - 4.5.3 Tümel Kriging 615
  - 4.5.4 Ko-Kriging 616
- 4.6 Destek yöney yöntemi 619
  - 4.6.1 Kesin sınıflandırma 623
  - 4.6.2 Esnek sınıflandırma 630
  - 4.6.3 Dönüşüm temelli sınıflandırma 633
  - 4.6.4 Destek yöney regresyon 643
  - 4.6.5 Doğrusal olmayan destek yöney regresyon 646
- 4.7 Yapay sinir ağları 652
- 4.8 Bulanık mantık 673

Tarihten bir Sayfa 706  
Kaynakça 707

4



(12)



(13)



## ÇOK AMAÇLI ENİYİLEME SÜREÇLERİ

- 5.1 Çok amaçlı eniyilemede tarihsel gelişim ve temel kavramlar 714
- 5.2 Genel maksat fonksiyonu 721
- 5.3 Pareto hattı 723

Bilim Damlası 736

Kaynakça 738

5



(14)

- [1] [http://www.concordcarlisle.org/jolsen/wp-content/uploads/sites/110/2015/09/geometry\\_by\\_gaciu000-d5q952h.jpg](http://www.concordcarlisle.org/jolsen/wp-content/uploads/sites/110/2015/09/geometry_by_gaciu000-d5q952h.jpg)
- [2] <https://www.nottingham.ac.uk/Mathematics/Images-Multimedia/466x234-new-research-group-images/sincosequationpaper466x234.jpg>
- [3] <http://cavalcantsobrelotigre.blogspot.com.tr/2012/11/antony-gormley-sirry.html>
- [4] <http://the189.com/wordpress/wp-content/uploads/2012/09/Series-of-Sculptures-by-Antony-Gormley-image1.jpg>;  
[http://ex-chamber-memo5.up.n.seesaa.net/ex-chamber-memo5/image/7Antony\\_Gormley\\_MEET-thumbnail2.jpg?d=a1](http://ex-chamber-memo5.up.n.seesaa.net/ex-chamber-memo5/image/7Antony_Gormley_MEET-thumbnail2.jpg?d=a1)
- [5] <https://i.dawn.com/large/2017/05/590fd5164e913.jpg>
- [6] <http://mccom.com/blog/wp-content/uploads/2014/01/163162706-Drawing-of-Dart-on-Target.jpg>
- [7] <http://rsos.royalsocietypublishing.org/content/royopensci/4/5/170202/F19.large.jpg>
- [8] [https://cdn.shopify.com/s/files/1/1202/1220/articles/9b7452\\_eeef35a53f48a45e19ae1923d3432f60d\\_1024x1024.jpg?v=1457613089](https://cdn.shopify.com/s/files/1/1202/1220/articles/9b7452_eeef35a53f48a45e19ae1923d3432f60d_1024x1024.jpg?v=1457613089)
- [9] <https://www.pinterest.com/explore/geometric-mountain-tattoo/>
- [10] <https://s-media-cache-ak0.pinimg.com/736x/f5/bc/b8/f5bcb8b9ec824149def9a2c8aa7b5044.jpg>
- [11] <https://lisarisingberry.files.wordpress.com/2015/02/dna-photo.jpg>
- [12] <https://salastia.files.wordpress.com/2013/03/cortex-in-metallic-pastels-gold-leaf.jpg>
- [13] <http://saimg-a.akamaihd.net/saatchi/60997/art/1559099/774292-7.jpg>
- [14] [http://pravlife.org/sites/default/files/field/image/13228133\\_990553551051906\\_1729412736\\_n.jpg](http://pravlife.org/sites/default/files/field/image/13228133_990553551051906_1729412736_n.jpg)

**kapak:** <http://www.onlyhdpic.com/images/Collections/hd-pics-photos-nature-ice-mountain-scenery-lake-desktop-background-wallpaper.jpg>





## TARİHSEL GELİŞİM & TEMEL KAVRAMLAR



1

“ Var olandan daha mükemmel evrenin varlığı mümkün değil... ”

Al-Ghazali<sup>(1)</sup>

“ Sonsuz sayıda muhtemel evrenler söz konusuydu ve Tanrı, sahip olduğu sonsuz akıl ve iyilikle en mükemmel evreni seçti”

Gottfried Leibniz<sup>(2)</sup>

	Sayfa
1.1 Eniyileme kavramı ve tarihsel gelişim	2
1.2 Matematiksel temeller	10
1.3 Eniyileme sürecinde temel kavramlar	110
Bilim Damlası	165
Yararlı Kitaplar	171
Kaynakça	172





## 1.1 Eniyileme kavramı ve tarihsel gelişim



niyileme<sup>i</sup> kelimesi, soyut bağlamda bir kararın verilmesi ya da somut manada sistemin veya sistem parçalarının tasarımlarının yapılması süreçlerinde elde edilecek somut ürünün mümkün merteye mükemmel olabilmesi için yapılan eylem, takip edilen süreç veya yöntem anlamına gelmektedir. Aynı kelime ailesinden eniyi<sup>ii</sup> sözcüğü, eniyileme sonrasında elde edilen mükemmel ürünün değeri, derecesi veya miktarı şeklinde tanımlanabilir<sup>(3)</sup>. Ancak bu tanım doğası itibariyle bazı belirsizlikler içermektedir. Zira en mükemmel durumdan kasıt, bazı hallerde en azı<sup>iii</sup>, bazı hallerde en fazlayı<sup>iv</sup>, genel bağlamda da en uç<sup>v</sup> değeri içerebilir. İlave diğer bir husus, her zaman olmasa da bazı hallerde söz konusu mükemmel durumun, belirli şartlar altındaki hali ifade edebileceğidir. Türkçe'ye sonradan dâhil edilen eniyileme kelimesinin Batı dillerindeki kökeni Latince'deki "optimas" kelimesine dayanmaktadır<sup>(4)</sup>. Bu kelime köken itibari ile "Ops" sözcüğüne bağlanmakta olup M.Ö.<sup>vi</sup> 8'inci yüzyılda Roma mitolojisine giren ve İtalya'da yaşamış Sabine kabilesine ait yeryüzü tanrıçasının adıdır. Tanrıçanın adı, seçenek anlamı ile beraber tarımda bereket ve bolluğu da temsil etmektedir. "Ops", Yunan mitolojisindeki "Rhea" tanrıçasının Roma medeniyetindeki karşılığıdır. "Rhea" ise Anadolu topraklarında somutlaştırılan ve genel manada dağların ve doğanın tanrıçası olan "Cybele" isimli Frigya medeniyeti tanrıçasına dayanır. Bu bağlamda eniyileme kavramının, bereket ve bolluk kavramları yanı sıra seçme ve irade mefhumları ile de örtüşürülerek Anadolu topraklarında filizlendiği söylenebilir.

Öte yandan eniyilemenin en az veya en çok kavramları ile örtüşmesi nedeniyle, tarihte pek çok bilim insanının değişik sorunları çözme adına eniyileme ile dolaylı veya doğrudan ilgilendikleri ya da ilişkili kavramlar üzerinde çalıştığı söylenebilir. Örneğin M.Ö. 300'lü yıllarda Euclid (M.Ö. 325-M.Ö. 265) iki nokta arasındaki en kısa mesafenin bir çizgi olduğunu, aynı çevre uzunluğuna sahip bir dikdörtgen geometrisinde azami alanın ancak bir kare şekli ile elde edilebileceğini göstermişti. M.Ö. 100'lü yıllarda İskenderiye'li Heron (M.Ö. 10 - M.Ö. 75), ışığın iki nokta arasındaki mesafeyi asgari uzunlukla ya da diğer bir ifade ile bir doğru hattı ile kat ettiğini beyan etmişti.

<sup>i</sup> İngilizce'de "Optimization", Almanca'da "Optimierung", Fransızca'da "Optimisation".

<sup>ii</sup> İngilizce'de "Optimum", Almanca ve Fransızca'da "Optimale".

<sup>iii</sup> The minimum

<sup>iv</sup> The maximum

<sup>v</sup> The extremum

<sup>vi</sup> M.Ö.: Milattan Önce



**Şekil 1.1** En az ya da en fazla tercihleri, eniyileme süreçlerini betimleyen olgulardır, (resim, miktarı temsil etmektedir).



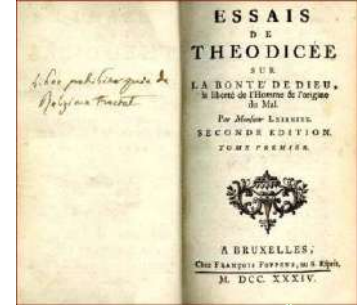
**Şekil 1.2** Ops: eski İtalya'da bereket ve bolluğu temsil eden mitolojik imge.



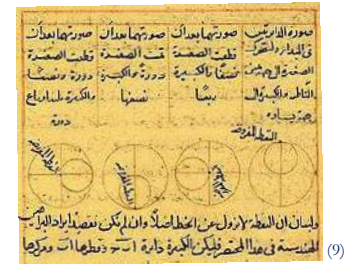
Ortaçağ döneminde gerek felsefi ve gerekse de matematiksel ve fiziksel bilimlerde bazı örnekler görmek mümkündür. Bu örneklerden ilki Al-Ghazali'ye (1058–1111) ait felsefi tanımlamalardır. 11'inci yüzyılda yaşayan Ghazali'ye göre var olandan daha mükemmel evrenin varlığı mümkün değildi. Böyle bir yaklaşımın, antik Yunan felsefecileri tarafından da dillendirildiği söylenebilir. Ancak Ghazali'nin haleflerinden ayrıldığı nokta, Tanrı'nın yaratmada kendine özgü bir irade sahibi olmasıydı, diğer bir ifade ile yaratmada belirli bir zorunluluğu yoktu, seçenekler arasından kendi iradesi ile birini seçmekteydi. Üstelik ona göre Tanrı, geçen zamanın her anında evrenin her zerresini ve her oluşunu sürekli olarak yaratmakta ve özgür iradesini sergilemekteydi<sup>(5)</sup>. 12'nci yüzyılda yaşayan Sharaf al-Tusi (1135–1213), felsefeden ziyade özellikle matematikle ilgilenen bir bilim insanı olup eniyileme açısından onun önemi, türev kavramını kullanmasıdır. Al-Tusi, özellikle 3'üncü dereceden kübik denklemin azami değerinin bulunmasında, denkleme ait türev alma işlemlerini gerçekleştiren birisiydi. Bu yönüyle beklili de tarihte türev kavramını kullanan ilklereydi<sup>(6)</sup>.

Daha çok astronomi ile ilgili çalışmalarıyla tanınmasına rağmen 17'nci yüzyılda yaşamış olan Kepler'in (1571–1630) pratik hayata dair yaptığı bazı tanımlamalar ilginçtir. Örneğin şarap fıçısının en uygun şeklinin nasıl olması gerektiği üzerinde durmuş, boşanma sonrası kendisine yeni bir eş arayışına girdiğinde ise dinamik programlama alanında örnek bir problem olan sekreter problemini<sup>i</sup> tanımlamıştı. Aynı dönemde yaşayan de Fermat (1601–1665), daha sonra detaylarına değineceğimiz gradyan esaslı eniyileme süreçlerinin temelini oluşturan eğim hesaplama ile azami-askari noktaların bulunması üzerine çalışmıştı<sup>(7)</sup>. Fermat'ın diğer bir katkısı da daha önce İskenderiye'li Heron'un (10–70), sonralarında İbn-i Heysem'in (965–1039) ifade ettiği, "ışığın iki nokta arasındaki mesafeyi askari uzunlukla ya da diğer bir ifade ile bir doğru hattı ile kat ettiği" düşüncesini zamanla örtüştürmesi ve "en kısa süre" ilkesini dillendirmesiydi. Ona göre ışık, iki nokta arasındaki güzergâhı mümkün olan en kısa sürede kat edecek şekilde seçmekteydi. 18'inci yüzyılın felsefeci ama aynı zamanda felsefi düşüncelerini matematikle örtüştüren düşünürlerinden von Leibniz'in (1646–1716) mümkün olan en iyi evren ile ilgili

<sup>i</sup> "Sekreter problemi" veya "Sultanın çeyizi" problemi olarak da bilinen bu probleme göre bir yönetici  $n$  tane aday sekreter arasından kendisine göre en uygun adayı seçmek istemektedir. Bu gaye ile her bir sekreteri rastgele bir şekilde mülakata alır. Her bir adayın mülakatının hemen sonrasında ise onun hakkında karar verilir. Dolayısıyla o anda reddedilen adayın bir daha geri gelmesi söz konusu değildir. Mülakatlar esasında yönetici her bir sekreter adayını değerlendirmektedir<sup>(8)</sup>. Ancak bir sonraki adayın özellikleri bilinmediği için verilen ret kararının yanlış olma ihtimali de söz konusudur. Dolayısıyla bu problemde en iyi adayın seçiminde dikkate alınacak en uygun strateji önem kazanmaktadır. Ayrıca karar verme sürecinin dinamik bir süreç olduğu da unutulmamalıdır.



Şekil 1.3 Leibniz'e ait 1734 tarihli kitabın kapak sayfası.



Şekil 1.4 Nasir al-Din al-Tusi isimli bilim insanına ait el yazmasında geçen geometri, ileri-geri hareketini dönü hareketine dönüştürmesi açısından önemlidir.

düşüncelerini burada ifade etmeden geçmek doğru olmaz. Leibniz'e göre sonsuz sayıda olası evrenler vardı ve Tanrı sahip olduğu en üstün erdemlilik ve iyilikle onların arasından en iyiyi seçmekteydi. Eğer içinde bulunduğumuz evren ve dünya mümkün olan en iyi evren ve dünya olmasaydı, Tanrı bu evreni ve içindekileri seçmezdi<sup>(11)</sup>. Leibniz'in tanımladığı bu yaklaşım, eniyileme süreci açısından önem arz eden bazı temel kavramları da beraberinde getirmektedir. Bunlardan ilki “bilgi ve erdemlilik”, ikincisi “iyilik” ve “değer atfetme” ve sonuncusu da olasılıklar arasından bilgi ve değer atfetme ile “irade gösterme” ve “seçme” şeklinde sıralanabilir. Bununla beraber sonsuz bilgi ve erdemlin söz konusu olmadığı bir değerlendirmede, Leibniz'in yaklaşımına ek olarak “arayış” kavramının da dikkate alınması gerektiği söylenebilir. Felsefi yaklaşımlarına ilave olarak Leibniz, Sharaf al-Tusi'nin öncülük ettiği türev kavramını ve yaygın olarak kullanılan  $dy/dx$  diferansiyel gösterimini de literatüre kazandırmıştır. Çağdaşı Newton (1642–1727), daha çok fiziksel olgular üzerine çalışmış ve eniyileme süreçlerinde kullanılan pek çok yönteme öncülük etmiştir. Newton'un literatüre kazandırdığı eniyileme ile ilgili ünlü problemlerinden ilki, aşgari dirence sahip geometri sorunudur. Esasen sorundan kasıt, su üzerinde ilerleyen geminin, minimum dirençle karşılaşacak şekilde tasarlanmasıydı<sup>(12)</sup>. Newton'un üzerinde çalıştığı ve aynı dönemde Johann ile Jacob Bernoulli kardeşlerin ve daha pek çoklarının üzerinde uğraştığı diğer bir eniyileme problemi de “en kısa sürede varma” problemiydi. Bu problem aynı zamanda “varyasyonlar hesabı” olarak tanımlanan matematik dalının doğuşunu da göstermekteydi. 18'inci yüzyılın ilk yarısında dikkate değer başka bir örnek eniyileme problemi, en az çevre uzunluğu ile aynı uzayı kaplayan düz kenarlara sahip geometrinin bulunmasıydı. Bu ifade aynı zamanda en az malzeme harcayarak aynı uzayın kaplanması manasına da gelmekteydi. Samuel Koenig (1712–1757), yaptığı çalışmalar neticesinde köşegene sahip en ekonomik geometrinin doğadaki bal peteği yapısının temelini oluşturduğu gösterdi<sup>(13)</sup>. Aynı dönemde yaşayan Leonhard Euler (1707–1783), devasa matematiksel katkıları yanında eniyileme açısından önem arz edecek tarzda “varyasyonlar hesabı”<sup>iii</sup> dalını genelleştiren çalışmalar yapmıştı. Çağdaşı Pierre-Louis Maupertuis (1698–1759), fiziğin ünlü ilkelerinden biri olan “en az eylem” ilkesini kurgulayan kişidir. “En az eylem” ilkesi, herhangi bir cismin uzaydaki bir noktadan diğerine gitmek için mümkün olan olası yörüngeler arasında eylem artışının en az olacağı



**Şekil 1.5** “En kısa sürede varma” probleminin analitik çözümü dönele geometrisini verir<sup>(10)</sup>.

<sup>i</sup> “En kısa sürede varma” problemi, özünde Yunanca kökenli kelimelerden oluşan “Brachistochrone’s problem” olarak bilinir. Sorun, belli bir yükseklikte bulunan bir fasulye tanesinin sürtünmesiz ortamda serbest bırakıldığında yere en kısa sürede varması için dikkate alınacak yolun şeklinin bulunmasıdır. Yapılan analitik çözümler, en kısa sürede varmayı garantileyen şeklin parametrik bir denkleme sahip olan dönele (cycloid) şekli olduğunu göstermektedir. Buna göre  $a$  daire yarıçapı ve  $\theta$  açı değeri olmak üzere dönele geometrisinin koordinatları  $x = a(\theta - \sin\theta)$  ve  $y = a(1 - \cos\theta)$  denklemlerine göre elde edilebilir.

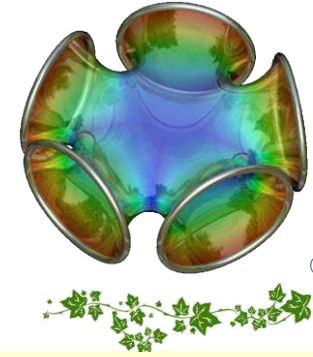
<sup>iii</sup> Calculus of variations



yörüngeyi seçeceği savı üzerine dayanır<sup>(14)</sup>. Joseph-Louis Lagrange (1736–1813), gerek “varyasyonlar hesabı” ve gerekse de “en az eylem” ilkelerine temel teşkil edecek nitelikteki matematiksel yaklaşımları geliştirmiştir. Bunlardan belki de en ünlüsü, Euler-Lagrange denklemi olarak tanımlanan denklemdir. Lagrange’ın kuramsallaştırdığı diğer bir problem de “minimal alan” problemi veya nam-ı diğer “Plateau’s” problemidir. 18’inci yüzyılın sonlarına doğru genel çerçevede sıralama problemleri olarak tanımlanabilecek ve ayrık eniyileme süreçlerine iyi bir örnek olan kaynakların dağıtımı ve nakliyesi problemlerini Gaspard Monge’nin (1746–1818) yaptığı çalışmalarda görüyoruz.

19’uncu yüzyıl, eniyileme problemlerinin çözülmesi bağlamında pek çok çözüm algoritmasının doğuşuna sahne olmuştur. Örneğin Adrien-Marie Legendre (1752–1833) ve Carl Friedrich Gauss (1777–1855) “en küçük kareler” yöntemini gündeme getirmiş, Joseph Fourier (1768–1830) mekanik sistemlerde karşılaştığı problemlerin çözümünde doğrusal programlama yöntemlerini kurgulamış, Augustin-Louis Cauchy (1789–1857) gradyan esaslı çözüm algoritmalarını geliştirmiştir<sup>(16)</sup>. Bu arada ilk defa Legendre tarafından kullanılan “ $\theta$ ” şeklindeki kısmi türev simgesini de ifade etmek gerekir. Özellikle gradyan esaslı çözüm algoritmalarında önem arz eden “Jacobi” matrisinin Carl Jacobi’ye (1804-1851) ait olduğunu belirtelim. Diğer taraftan ilgi alanı çoğunlukla istatistiksel termodinamik olsa da Willard Gibbs’in (1839–1903) termodinamik ilkelerinin tanımlanmasında ve kimyasal denge kavramının kurgulanmasında kullandığı minimum enerji hali, eniyileme açısından önem arz eder. Özellikle termodinamiğin ikinci yasası ile gündeme gelen entropinin doruklaştırılması kurgusu, eniyileme bağlamında olmasa bile bir en uç nokta yönelimini göstermesi açısından kayda değerdir. Yine Gibbs’in termodinamik süreçlerin incelenmesinde kullandığı grafik yöntemlerinin, dışbükey analiz kurgulamasındaki katkıları unutulmamalıdır<sup>(17)</sup>.

19’uncu yüzyılın kayıtlarını geçerken William Hamilton’ın (1805–1865) geliştirilmiş “en az eylem” ilkesinden de mutlaka bahsedilmelidir. Daha önce de Fermat’ın “en kısa süre” ve Maupertuis’in “en az eylem” ilkelerinden bahsedilmişti. Hamilton, söz konusu iki ilkenin benzerliklerine dikkat çekip esasen bu ikilinin tek bir çerçevede birleştirilebileceğini gösterdi. Bazılarına göre geliştirilmiş “en az eylem” ilkesi ya da diğer adıyla “Hamilton” ilkesi, fiziğin en temel yasalarından birisidir. Eniyileme açısından ilkenin önemi, fiziksel kanunları Euler-Lagrange denkleminin bir uç nokta çözümü olarak ifade etmesidir. Daha açık bir ifade ile evrendeki fiziksel kanunlar esasen yerel ya da genel, ancak kesinlikle optimal bir noktaya ait



(15)

**Şekil 1.6** Sınır şartı olarak altı çemberli tel kafes söz konusu olduğunda minimal yüzey alan çözümü, (sabun köpüğünün doğal olarak aldığı geometri).



**Şekil 1.7** Evrendeki en temel ilkelerden birisinin “en az eylem” ilkesi olduğu düşünülmektedir, (resim sanatsal bir bakış açısıyla evreni temsil etmektedir).





çözümlerdir<sup>i</sup>. Bu açıdan bakıldığında, yüzyıllar önce Gazali'nin ya da yaklaşık bir asır önce Leibniz'in felsefi bağlamda ifade ettiği en mükemmel evren yaklaşımı, daha somut bir hal almaktadır.

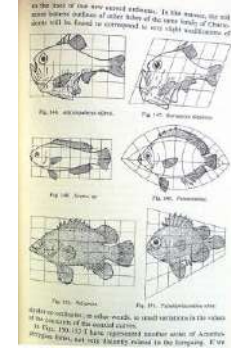
20'nci yüzyıl, gerek varyasyonlar matematiği ve gerekse de tamamen farklı yöntemlerin geliştirildiği, yöntemlerin çeşitliliği açısından oldukça yaygın ama aynı zamanda derinlik içeren bir çağdır. Bu yüzyıl Farkas'ın (1847–1930) ünlü optimalite şartı önermesi ile başlatılabilir. Söz konusu önerme, daha sonra lineer programlama ve dışbükey problemlerde optimalite şartlarının ispatlanmasında kullanılmıştı. Genel optimal çözümün varlığı bağlamında önem arz eden dışbükeylik kavramı da yine aynı dönemlerde yaşayan Johan Jensen'e (1859–1925) dayandırılabilir. 1917 yılında H. Hancock (1867-1944) tarafından yazılan “Theory of maxima and minima” isimli eser, belki de eniyileme literatüründe kullanılan ilk ders kitabıdır. Kitap, içerik olarak bir değişkenli fonksiyonlardan başlamakta ve yakınsama konularını da içeren analitik fonksiyonlarla sonlanmaktadır<sup>(18)</sup>. Aynı yıl biyomatematikçi D. W. Thompson (1860–1948) tarafından yayımlanan “On growth and form” isimli eser, yaşayan canlıların formlarının nasıl şekillendiğine dair yaklaşımları içerir. Thompson'ın ifadesi ile doğa ve dolayısıyla canlılar, Maupertuis'in beyan ettiği gibi mümkün olan en iyi dünyaya doğru meyletmektedir<sup>(19)</sup>. 19'uncu yüzyılın sonlarında ve biraz da 20'nci yüzyılın başlarında dikkate değer diğer bir gelişim, finans sektörü kaynaklı olarak ortaya çıkan çok amaçlı eniyileme<sup>ii</sup> süreçleridir. Bu meyanda F. Y. Edgeworth (1845-1926) ve V. Pareto'nun (1848-1923) çalışmaları, birden fazla amacın gözetildiği eniyileme süreçlerinin sistematik olarak analizini içerir.

Formlar arasındaki ilişkilerle ilgili diğer bir çalışma, M. Morse'a (1892-1977) ait olan ve “Morse kuramı” olarak anılan çalışmadır. Söz konusu kuram, varyasyonlar matematiğinin genelleştirilmesini sağlayan, matematiksel manada bir manifold<sup>iii</sup> üzerinde tanımlı fonksiyonun uç değerleri (“extremum” noktalar) ile manifoldun topolojik özellikleri arasındaki ilişkileri tanımlayan bir yaklaşımdır. Bu yaklaşımın eniyileme açısından açılımı, optimal olarak tanımlanan uç noktalar ve o noktalara varmayı sağlayan türev değerleri ile yüzeyin topolojik özellikleri

<sup>i</sup> “Hamilton” ilkesine göre bir dinamik sistemin  $t_1$  ve  $t_2$  anları arasında bir konumdan diğer bir konuma hareketi için Euler-Lagrange entegralinin en küçük olduğu yörünge kullanılır.

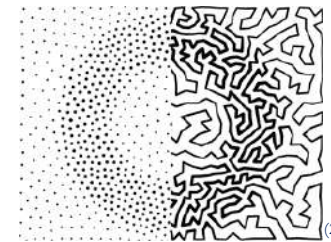
<sup>ii</sup> Multi-objective optimization

<sup>iii</sup> Manifold küçük ölçekte birbirine benzeyen geometrilerdir. Örneğin bir doğru ile bir çember küçük ölçekte birbirleri ile aynı topolojik özelliğe sahiptir. Her ikisi de bir boyutlu manifold olarak adlandırılır. Nihayetinde bir doğru eğilip bükülerek, yani belirli bir dönüşümle bir çembere dönüştürülebilir. Benzer şekilde yüzeyler de iki boyutlu manifold olarak adlandırılır. Örneğin bir kürenin yüzeyi küçük ölçekte düz bir plaka olarak ele alınabilir. Plakayı eğip bükerek bir küreye dönüştürmek mümkündür<sup>(21)</sup>.



(20)

**Şekil 1.8** D. W. Thompson'ın kitabından bir sayfa: canlı formlarının değişimine dair geometrik yaklaşım.



(22)

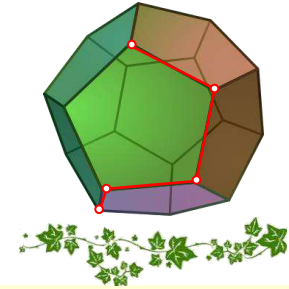
**Şekil 1.9** Gezici satıcı problemi yaygın bir sıralama problemidir.

arasındaki bağların kurulmasıdır. J. Viner (1892-1970) tarafından gündeme getirilen zarf kuramı, optimal değerlerin optimal değerini hesaplama adına yapılan bir yaklaşıma ifade etmektedir. 1931 yılında Viner'in daha çok ekonomi ve maliyet üzerine yazdığı makale ile gündeme getirdiği zarf kuramı, farklı değişkenler ve parametreler içeren bir fonksiyonun parametre değeri değiştikçe optimal değerinin nasıl değiştiğini açıklar. Takip eden sene içerisinde K. Merger (1902–1985) tarafından formal olarak tanımlanan ve kendi Almanca tabiri ile “Botenproblem” olarak ifade edilen “gezici satıcı problemi” eniyileme literatürüne katılmıştır. Merger esasen mesafe geometrisi olarak tabir edilen matematiksel alan üzerinde çalışmış ve aralarındaki mesafelerin belli ve sabit olduğu noktalar kümesi üzerinde durmuştur. Doğrusal programlamanın matematiksel modelleri L. V. Kantorovich'in (1912–1986) çalışmalarına dayandırılabilir<sup>(23)</sup>. Ancak bu konuda etkin bir çözüm yöntemi sunan bilim insanı G. Dantzig'dir (1914-2005). Amerika Birleşik Devletleri (ABD) ordusu için çalışan Dantzig, doğrusal problemlerin çözümünde “simplex” yöntemini geliştirerek eniyileme literatürüne geçmiştir. Problemin farklı bir bakış açısı ile çözümü J. von Neumann (1903–1957) tarafından “ikizlik kuramı” adı altında dile getirilmiştir. 1949 yılında ABD’de düzenlenen uluslararası bilgi şöleni ise eniyileme tarihindeki ilk kongre niteliğindedir.

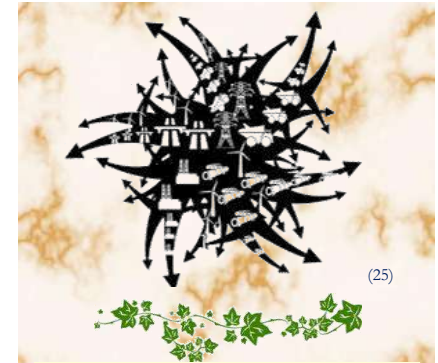
20'nci yüzyılın ilk yarısında H. W. Kuhn (1925- ) ve A. W. Tucker (1905–1995), doğrusal olmayan problemlerde gerek şart bağlamında optimalite şartlarını içeren meşhur KT (Kuhn-Tucker) şartlarını yayımladılar. Bununla beraber söz konusu yayından önce, W. Karush (1917–1997) tarafından aynı hususların yüksek lisans tezi olarak 1939 yılında yayımlandığını da burada belirtmek gerekir<sup>(24)</sup>. Aynı dönemde yayımlanan diğer bir yayın içeriği, H. Robbins (1915-2001) ve S. Monro'ya (1920-1995) ait stokastik ya da diğer bir ifade ile rastlantısal yaklaşım yöntemidir. Söz konusu yöntem, eniyileme yöntemlerinin gelişimi açısından önemli bir yayın olup bir manada yöntem anlayışının çehresini değiştirmiştir denebilir. Tarihsel gelişim açısından eniyileme yöntemlerine bakıldığında, bu yöntemlerin büyük oranda varyasyonlar matematiğine dayandırıldığını ve dolayısıyla türev kavramları ile ilişkilendirildiğini görebiliriz. Bu yönüyle eniyileme yöntemleri, oldukça sağlam matematiksel kuramlara ve dolayısıyla da deterministik ya da diğer bir ifade ile belirlemci bir çatıya dayanır. Sebep-sonuç ilişkisi sarmalına dayalı ve kesin sonuçlar içeren yöntemlere alışkın eniyileme ve daha genelinde

<sup>i</sup>Zarf kuramını bir örnek ile açıklamak mümkündür. Örneğin  $y = x^2 - ax$  şeklinde bir fonksiyon olsun. Burada  $x$  değişken ve  $a$  ise parametre olarak tanımlansın. Fonksiyonun her bir  $a$  parametre değeri için sahip olacağı en uygun değer farklı olacaktır. Zarf kuramı fonksiyonun optimal değerinin  $a$  parametresine göre nasıl değiştiğini ifade eder. Buna göre söz konusu değişim  $\partial y^*/\partial a = \partial y/\partial a \{x = x^*(a)\}$  ile elde edilebilir. Burada \* işareti en uygun değeri göstermektedir.

<sup>ii</sup> Almanca “botenproblem” ifadesini İngilizce’de “messenger problem” olarak tanımlamak mümkündür.



Şekil 1.10 “simplex” yönteminde düğüm noktaları önem arz eder.



Şekil 1.11 Rastal süreçler, eniyilemede yeni yöntemlerin kapısını aralamıştır, (resim, kaotik düzeni temsil etmektedir).

matematik topluluğu, rassal süreçlere dayalı problem ve yöntemlere ilk dönemlerde kuşku ile bakmıştır. Ancak ortaya çıkışına paralel süreç içerisinde özellikle bilgisayar ve programlama yöntemlerinin hızla gelişim göstermesi, rastsallığa dayalı yöntemlerin daha yaygın ve çeşitli hale gelmesine vesile olmuştur. 1950’li yıllarda değişik seçenekleri olan “yaklaşık Newton” ve “eşlenik gradyan” yöntemlerinin geliştirildiğini söyleyebiliriz. Diğer taraftan optimal kontrol kuramının da varyasyonlar matematiğinden ayrı bir disiplin olarak literatüre dahil edildiği söylenebilir. Söz konusu çalışmaların arka planında, mekanik ve elektromekanik sistemlerin gelişimi ve uzay yarışının gündemde olması yatmaktadır. Bu bağlamda R. Bellman’a (1920–1984) atfedilen optimallik ilkesi, dönemin güncel çalışmalarından birisidir. Aynı dönemde rastsal süreçler ve bilgisayarların kullanımı ile ilgili ilk çalışmalardan birisi de N. A. Barricelli’ye (1912–1993) ait olup(26), söz konusu çalışma evrimsel algoritmaların ilk benzetim çalışması niteliğindedir.

1960’lı yıllarda doğrusal olmayan problemlerde uygulanabilecek nitelikteki genelleştirilmiş “simplex” yönteminin geliştirildiğini, “ardışık ikilenik programlama” yaklaşımı ve sürümlerinin yayımlandığını belirtebiliriz. Gradyan esaslı yöntem niteliğindeki söz konusu yaklaşımlar yanında rastsal yöntemler için de verimli bir dönem olan bu yıllarda L.A. Rastrigin’in (1929-1998) gündeme getirdiği gradyan bilgisine ihtiyaç duyulmayan “rassal tarama” yönteminden ve L. W. Fogel’in (1928–2007) sunduğu evrimsel programlama yaklaşımlarından bahsedilebilir. Bu dönemde gündeme gelen diğer bir yaklaşım, J.A. Nelder (1924–2010) ve R. Mead (1938–2015) tarafından teklif edilen “Nelder-Mead simplex” yöntemi. Söz konusu yöntem esasen ne rastsal ve ne de gradyan esaslı bir yöntem olup daha çok deneme ve yanılma<sup>iii</sup> esaslı bir yöntem grubuna yakındır.

1970’li yıllar ve sonrası deterministik yaklaşımlardan ziyade rastsal yöntemlerin zenginleştiği dönemlerdir. Bununla beraber dönemin deterministik yaklaşım bağlamındaki en önemli katkısı, N. K. Karmarkar’ın (1957- ) doğrusal problemlerinin çözümüne yönelik geliştirdiği Karmarkar yöntemidir(27). Öte yandan Pareto’ya ait “Manuale di economia politica” adlı eserin İngilizce’ye tercüme edilmesiyle beraber çok amaçlı eniyileme yöntemlerinde ciddi bir ivmelenme olduğu söylenebilir(28). Rassal süreçler açısından dönemin en önemli yeniliği, J. H. Holland’ın (1929–2015) önerdiği genetik algoritmalar. Bu meyanda onun yazdığı 1975 tarihli

<sup>i</sup> Random search

<sup>ii</sup>“Simplex” kavramı esasen üçgen şeklinin genelleştirilmiş kullanımıdır. Üçgen geometrisi bir çokkenar geometrisidir. Çokkenar ise kenarları düz olan geometrilerin genel adıdır. “Simplex” kavramının eniyilemede kullanılmasının nedeni üçgeni oluşturan üç noktadan hareketle elde edilen iki noktadan daha iyi bir üçüncü noktanın elde edilmesidir.

<sup>iii</sup> Deneme ve yanılma temelli yöntemler genellikle sezgisellik kavramı ile örtüştürülür.



**Şekil 1.12** Genetik algoritmaya ait kodlama yaklaşımı DNA yapısı ile benzeştirilir.

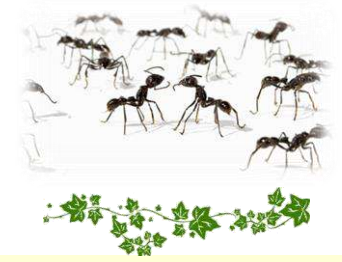


**Şekil 1.13** Parçacık sürü algoritması sürü davranışlarından esinlenir.



“Adaptation in natural and artificial systems” isimli kitap<sup>(30)</sup>, sadece algoritma önerilerini değil, aynı zamanda algoritmaların arka planında yatan matematiksel kuramı da açıklamaya çalışır. 1980’li yılların başlarında aynı tasnifteki diğer bir yöntem olan “benzetimli tavlama” yaklaşımını görmekteyiz. S. Kirkpatrick ve arkadaşları tarafından dillendirilen söz konusu yöntem, deneme ve yanılma esaslı olup bazı yönlerden N. C. Metropolis’in (1915–1999) 1950’lerde dile getirdiği yöntemle, daha net bir ifade ile rastsal seçim yerine belirli bir ölçüte göre seçme esasına dayanır. 1980’li yılların diğer bir getirisi F. W. Glover’in (1937- ) öne sürdüğü “tabu tarama” yaklaşımıdır. Glover aynı zamanda sezgisel yöntemlerin matematiksel karşılığı olan “metaheuristik” kavramını da literatüre katan kişi olarak geçer. 1990’lı yılların başında doğadaki canlıların kullandığı besin arama yöntemleri, eniyileme süreçleri açısından esin kaynağı olmuştur denebilir. Örneğin M. Dorigo’nun (1961- ) önerdiği karınca koloni eniyileme<sup>ii</sup> yöntemi<sup>(31)</sup>, temelinde karıncaların besin arama ve en kısa sürede ona ulaşma sürecini algoritmik yöntemle benzeştiren bir yaklaşımdır. J. Kennedy (1950- ) ve R. C. Eberhart ise sürü psikolojisi temelli yaklaşımlarını parçacık sürü eniyileme<sup>iii</sup> yöntemi olarak literatüre tanıtmış ve algoritmik olarak nispeten basit bir yaklaşımla karmaşık eniyileme süreçlerinin çözümüne olanak sağlamışlardır. Doğadaki canlı yaşam formlarının yapı ve davranışlarını taklit eden son kayda değer yaklaşım D. Karaboğa’ya (1965-) ait olan yapay arı koloni algoritmalarıdır. Yapay arı koloni algoritması<sup>(32)</sup>, bal arısı kolonilerinin zengin nektarlı yiyecek kaynaklarını aramada ortaya koydukları zeki davranışları örnek alan bir algoritmayı içerir<sup>iv</sup>. Günümüzde sezgisel yöntemler oldukça geniş bir yelpazeye yayılmış olup çıkış kaynağı itibarıyla fiziksel, kimyasal, biyolojik ve hatta sosyal süreçler ile örtüşen yöntemler söz konusudur<sup>(33)</sup>. Geçmişten günümüze süregelen en iyiyi bulma çabaları gerek kaynağı ve gerekse de doğası itibarı ile oldukça farklı yöntemlerin doğmasına vesile olmuştur. Bu süreç aslında mebdai ya da diğer bir ifade ile ilkesi kendinde olan bir süreç olup bu bağlamda en iyiyi bulma yöntemlerinin en iyisini bulma arayışı akıp giden zamana bağlı olarak devam edecektir. Bir eniyileme yönteminin diğerinden daha iyi olması temelde iki ölçüte dayandırılabilir:

► “Daha kısa sürede<sup>v</sup> daha hassas<sup>vi</sup> en iyilere ulaşmak...”



**Şekil 1.14** Karınca sürüsü eniyileme algoritmaları özellikle sıralama problemlerinde tercih edilebilir.

<sup>i</sup> Simulated annealing

<sup>ii</sup> Ant colony optimization

<sup>iii</sup> Particle swarm optimization

<sup>iv</sup> Artificial bee colony optimization

<sup>v</sup> Efficiency

<sup>vi</sup> Accuracy







## 1.2 Matematiksel temeller

Eniyileme yöntemleri gerek kavram bazında ve gerekse de çözümleme bağlamında temel matematiksel yöntemleri kullanır. Söz konusu yöntemler analitik yöntemler olabileceği gibi sayısal ve hatta bazı uygulamalarda deneysel yöntemleri içerebilir. Bu nedenle özellikle analitik ve sayısal matematik ile ilgili bazı bilgilerin gözden geçirilmesi, eniyileme yöntemlerinin anlaşılması açısından gereklidir. Bahse konu çerçevede dâhilinde öncelikle geometrik kavramlar, devamında lineer cebir, akabinde diferansiyel ve integral, nihayetinde ise bazı sayısal yöntemlere değinilecektir.

### 1.2.1 Geometrik kavramlar

Geometri kavramı, matematiğin alt bilim dallarından biridir ve şekil, boyut, şekiller arası ilişkiler ile şekilsel büyüklükler üzerine yoğunlaşır. İçinde yaşadığımız evren, zaman ve mekân kavramlarına dayalı olup özellikle mekân ya da diğer bir ifade ile konum, boyutsal özellikler taşır. Her ne kadar algılanan evren, fiziksel konum itibariyle üç boyutlu olsa da matematiksel tanımlamada sonsuz sayıda boyuttan bahsedilebilir. Boyut sayısı kuramsal olarak artırılabilir ve temel tanımlamalar farklı boyutsal büyüklüklerdeki uzay için geçerli olabilir. Bu kavramlardan ilki ve belki de en temeli noktadır.

#### a. Nokta

Geometride nokta, parçası olmayan; uzunluğu, alanı veya hacmi bulunmayan geometrik olgu olarak tanımlanır. Nokta, farklı boyutlarda<sup>i</sup> olabilir. Örneğin iki boyutlu bir uzayda herhangi bir noktanın tanımlanabilmesi için  $(x_1, x_2)$  gibi iki sayısal değer verilmesi gerekir:

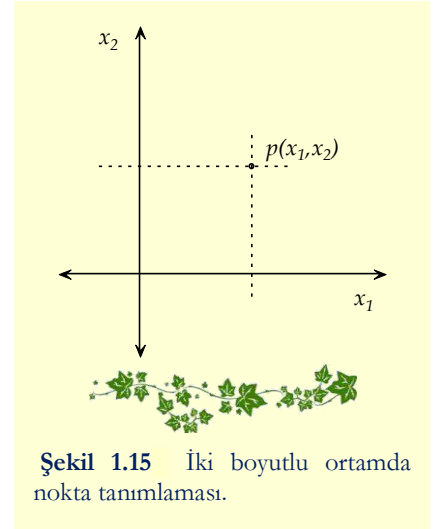
$$p(x_1, x_2) \quad (1.1)$$

Boyut sayısı arttığı takdirde noktanın tanımlanması için boyut sayısı kadar sayısal değer belirtilmelidir. Dolayısıyla  $n$  boyutlu bir ortamda her hangi bir nokta;

$$p(x_1, x_2, \dots, x_i, \dots, x_n) \quad (1.2)$$

şeklinde ifade edilebilir. Nokta aynı zamanda yöneysel bir kavram olarak da gösterilebilir. Her hangi bir noktanın konumunun tanımlanması müteakip ifadedeki gibi yazılabilir:

<sup>i</sup> Dimension



Şekil 1.15 İki boyutlu ortamda nokta tanımlaması.



$$\mathbf{p}^T = [x_1 \quad x_2 \quad \cdots \quad x_i \quad \cdots \quad x_n] \quad (1.3)$$

### b. Doğru ve doğru parçası

Nokta sayısının birden fazla olması halinde noktalar arasındaki ilişkiler önem kazanır. Bu bağlamda iki farklı nokta arasındaki en basit ilişkiye doğru parçası denir. Doğru parçası, herhangi bir alana veya hacme sahip olmasa da belirli bir mesafeye ya da diğer bir ifade ile uzunluğa sahiptir.

Doğru parçasının tanımlanması adına çok boyutlu bir ortamda  $p_1$  ve  $p_2$  gibi iki nokta söz konusu olsun. Söz konusu iki nokta arasındaki doğruyu;

$$\mathbf{p} = \theta \mathbf{p}_2 + (1 - \theta) \mathbf{p}_1 \quad (1.4)$$

denklemleri ile ilişkilendirmek mümkündür. Burada  $\theta$  değeri,  $[0,1]$  şeklinde tanımlandığı takdirde doğru parçası geometrisi, sınır tanımı yapılmadığı takdirde doğru elde edilir. Dikkat edileceği üzere iki nokta arasında veya dışında pek çok ara nokta tanımlamak mümkündür.

Diğer taraftan böyle bir tanımlama boyut sayısından bağımsız olup 2 boyutlu bir ortamda nasıl doğruyu gösteriyorsa üç veya daha büyük boyutlu bir ortamda da yine doğruyu göstermektedir. Ancak bu yalın tanımlama, boyutlar arasındaki herhangi bir ilişkiyi göstermez. Böyle bir ilişki için hiperdüzlem<sup>ii</sup> kavramı tanımlanır.

### c. Hiperdüzlem ve yarımuzay

Hiperdüzlem, bulunulan ortam uzayını ikiye bölen düzlemler olup bölüm sonrası oluşan uzaya yarımuzay<sup>iii</sup> adı verilir. Örneğin bir boyutlu ortamda hiperdüzlem, nokta olup iki boyutlu ortamda doğru ve üç boyutlu ortamda düzlemdir. Dikkat edileceği üzere  $n$  boyutlu bir ortam uzayının hiperdüzlem boyutu  $n - 1$  mertebesinde dir. Hiperdüzlemin gösterimi;

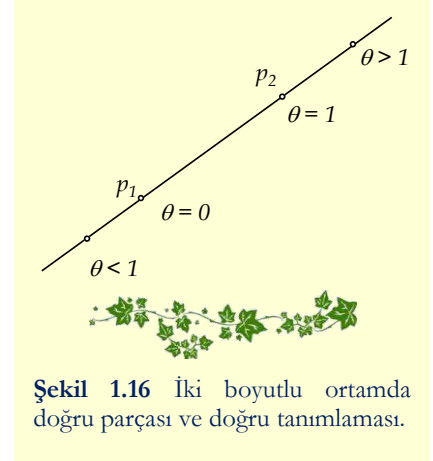
$$\mathbf{a}^T \mathbf{x} = b \quad (1.5)$$

şeklinde verilebilir.

<sup>i</sup>  $[0,1]$  ifadesi  $0 \leq \theta \leq 1$  anlamına gelmektedir.

<sup>ii</sup> Hyperplane

<sup>iii</sup> A half-space



Şekil 1.16 İki boyutlu ortamda doğru parçası ve doğru tanımlaması.





Burada  $a$  sabit değerleri içeren yöneyi,  $x$  çözüm noktalarını içeren kümeyi ve  $b$  sabit bir değeri ifade eder. Bu bağlamda  $n$  boyutlu bir uzayda hiperdüzlem;

$$a_1x_1 + a_2x_2 + \dots + a_ix_i + \dots + a_nx_n = b \quad (1.6)$$

şeklindeki bir doğrusal denklemi temsil eder. Buna mukabil her bir yarımuzay;

$$\begin{aligned} a^T x &< b \\ a^T x &> b \end{aligned} \quad (1.7)$$

şeklindeki eşitsizlikler ile temsil edilebilir.

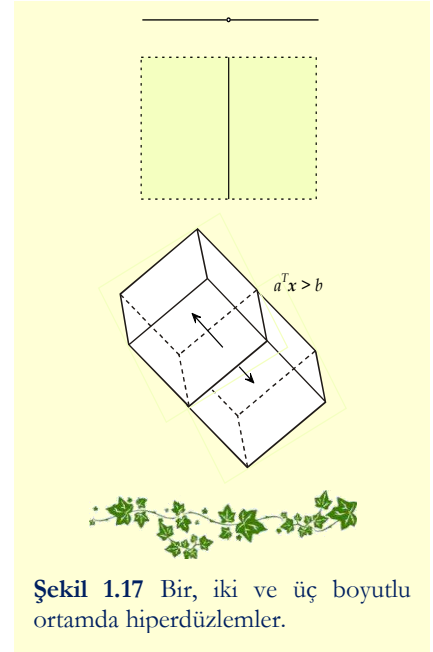
### ç. Çokkenar

Çokkenar<sup>i</sup>, doğrusal kenarlara sahip olan geometrilere verilen genel bir tanımlamadır. Dolayısıyla sonlu sayıdaki farklı hiperdüzlem ve yarımuzayların oluşturacağı kesişim geometrisi çokkenarı temsil eder. Çokkenarın kapalı olması şart değildir, dolayısıyla bir hiperdüzlem de çokkenar olarak düşünülebilir. Çokkenarın gösterimi;

$$\begin{aligned} a_j^T x &\leq b_j, & j = 1, \dots, m \\ c_j^T x &= d_j, & j = 1, \dots, k \end{aligned} \quad (1.8)$$

denklem takımları ile tanımlanır. Görüldüğü üzere burada hiperdüzlem ve yarımuzayların oluşturdukları geometri söz konusudur. İki boyutlu bir ortam uzayında oluşturulan çokkenar geometrisine çokgen<sup>ii</sup> adı verilir. Buna göre en az sayıdaki kenara sahip kapalı çokgen şekli üçgendir. Bu geometriye aynı zamanda “simplex” adı da verilir. Üç boyutlu ortamdaki çokkenar ise çokyüzlü<sup>iii</sup> olarak tanımlanır. En az sayıdaki yüze sahip kapalı çokyüzlü geometrisi, üçgen prizması olup “2-simplex” tanımı farklı derecede de olsa söz konusu geometri için geçerlidir. Herhangi bir çokkenarın kenarlarının kesişim noktaları doğrusal denklem takımlarının çözüm kümesini verir. Dolayısıyla  $n$  boyutlu bir lineer denklem takımının çözümünü veren nokta esasen bir çokkenarın kesişim noktasıdır. Kapalı bir çokkenarın kapladığı bölge ise yarımuzayların kesişimi olup bu bölge yarımuzay şartlarını sağlayan çözüm bölgesi niteliğindedir.

<sup>i</sup> Polytope  
<sup>ii</sup> Polygon  
<sup>iii</sup> Polyhedron



Şekil 1.17 Bir, iki ve üç boyutlu ortamda hiperdüzlemler.



Şekil 1.18 İki boyutlu ortamda açık ve kapalı, üç boyutlu ortamda kapalı çokkenar örnekleri.





#### d. Eliptik eğriler

Eliptik<sup>i</sup> eğriler, doğrusal olmayan bir karaktere sahip olup bünyesinde eliptik ve toparlak<sup>ii</sup> geometrileri barındırır. Eliptik geometri, aynı zamanda içinde sonsuz sayıdaki noktayı barındıran bir küme olup  $n$  boyutlu bir ortamda kümeye ait noktalar;

$$C = \{x \mid (x - x_c)^T A^{-1} (x - x_c) \leq 1\} \quad (1.9)$$

$$x \in R^n$$

$$A = A^T$$

şeklinde tanımlanır. Burada  $x_c$  eliptik eğrinin merkezini ve  $A$  kümeye ait noktaların merkezden ne kadar uzakta konumlanabileceğini belirleyen matrisi ifade eder. Matrisin tanımı;

$$A = r^2 I \quad (1.10)$$

olarak yapıldığında ortaya çıkan geometri, toparlak geometri olup burada  $r$  toparlağın yarıçapını,  $I$  birim matrisi göstermektedir. Matrisin farklı bir karaktere sahip olması halinde eliptik eğrinin her eksendeki en uzak mesafesi farklılık arz eder. Bu mesafeler esasen  $A$  matrisinin özdeğerlerine denk düşer. Bu bağlamda söz konusu mesafeleri  $\{\sqrt{\lambda_i}, i = 1, 2, \dots, n\}$  şeklinde ifade etmek mümkündür.

#### e. Komşuluk ilişkisi

Herhangi bir  $x$  noktasının belli bir yakın bölgesindeki noktaların oluşturduğu kümeye, o noktanın komşuluğu adı verilir. Bu bağlamda  $n$  boyutlu bir ortamda komşuluğu ve komşuluk<sup>iii</sup> noktasını;

$$C = \{y \mid \|y - x\| < \varepsilon\} \quad (1.11)$$

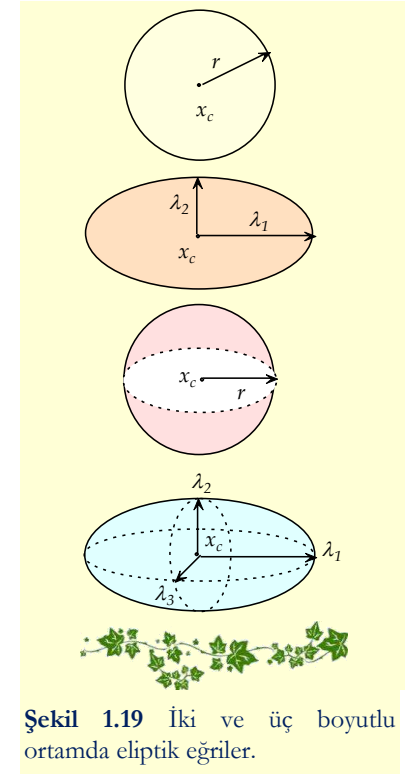
$$y, x \in R^n$$

ifadeleri ile tanımlayabiliriz. Böyle bir tanımlama eliptik bir eğriye, daha açık bir ifade ile toparlak geometrisine benzer. Dolayısıyla komşuluk, yarıçapı  $\varepsilon$  olan eliptik eğri setini temsil etmektedir.

<sup>i</sup> Eliptik eğri veya elips İngilizce'de "ellipsoid, ellipse" olarak geçer.

<sup>ii</sup> Toparlaklıktan kastedilen mükemmel eğrisellik olup iki boyutlu ortamda söz konusu geometri daireyi, üç boyutlu ortamda ise küreyi temsil eder. Bu tür geometriler İngilizce'de "ball" olarak geçer.

<sup>iii</sup> Neighborhood



Şekil 1.19 İki ve üç boyutlu ortamda eliptik eğriler.

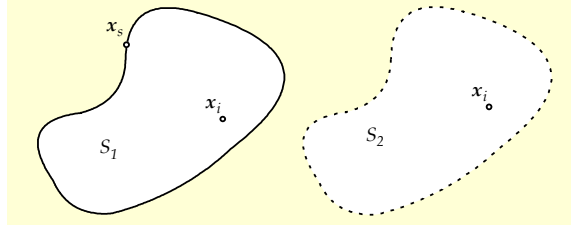


Şekil 1.20 İki boyutlu ortamda  $S$  kümesi,  $x$  noktası ve komşuluğu.

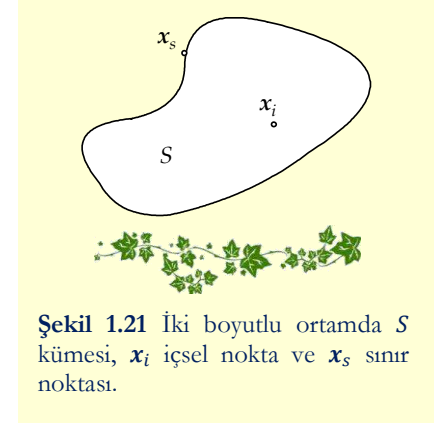




Komşuluk tanımı, sınır noktası<sup>i</sup> ile içsel noktası<sup>ii</sup> tanımlarının temelini de oluşturur. Buna göre  $S$  setine ait bir nokta ve o noktanın komşuluğundaki tüm noktalar  $S$  setine ait ise söz konusu nokta içsel nokta,  $\{x_i\}$ , olarak adlandırılır. Buna karşılık yine  $S$  setine ait bir nokta ve o noktanın komşuluğundaki noktaların bazıları  $S$  setine ait ve bazıları da ait değil ise söz konusu nokta sınır noktası,  $\{x_s\}$ , olarak isimlendirilir.



Şekil 1.22 İki boyutlu ortamda açık ve kapalı küme örneği.



Şekil 1.21 İki boyutlu ortamda  $S$  kümesi,  $x_i$  içsel nokta ve  $x_s$  sınır noktası.

Komşuluk kavramının getirdiği diğer bir tanımlama, açık ve kapalı küme tanımlarıdır. Açık küme; içsel noktaların olduğu, ancak sınır noktalarının olmadığı kümeye denir. Kapalı kümede içsel ve sınır noktaları vardır. Sınır kavramı matematiksel manada  $\{\leq\}$  ile tariflenebilir. Dolayısıyla açık kümede ilişki  $\{<\}$  şeklinde olacaktır.

#### f. İlgin ve dışbükey kümeler

İlgin<sup>iii</sup> kümesi,  $C$  küme elemanlarından herhangi iki noktanın üzerinden geçen doğruya ait noktaların yine  $C$  kümesi elemanı olması halinde tanımlanan kümedir. Bu bağlamda bir doğru üzerinde bulunan noktaları içeren küme ilgin kümedir. Bahse konu ilgin kümeye verilebilecek örneklerden birisi;

$$C = \{x \mid Ax = b\} \quad (1.12)$$

$$A \in R^{m \times n}, \quad b \in R^m$$

olup burada  $C$  kümesi bir lineer denklem takımının çözüm kümesidir. Örneğin iki boyutlu bir ortamda söz konusu denklem takımı;

<sup>i</sup> Boundary point

<sup>ii</sup> Interior point

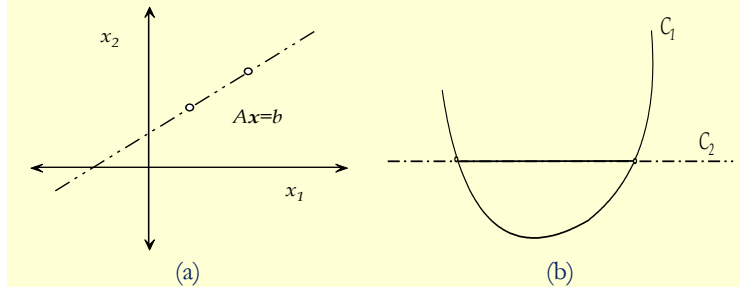
<sup>iii</sup> Affine





$$\begin{bmatrix} a_{11} & a_{12} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = b_1 \quad (1.13)$$

şeklinde yazılabilir. Böyle bir denklem takımını sağlayan tüm noktalar bir doğru üzerinde olacaktır. Dolayısıyla denklem takımının çözümünü sağlayan çözüm kümesi, Şekil 1.23(a)'da görülebileceği gibi bir ilgin kümedir.



Şekil 1.23 İki boyutlu ortamda ilgin ve dışbükey küme örnekleri.

Dışbükey<sup>i</sup> küme, ilgin kümeden farklılık arz eder. Dışbükey kümesi,  $C$  küme elemanlarından herhangi iki noktanın birbirini engelsiz bir şekilde görmesi halinde tanımlanan kümedir. Söz konusu tanımlama, doğrusal veya eğrisel ama çizgisel bir karaktere sahip kümede;

$$\begin{aligned} p_1, p_2 &\in C \\ \theta p_2 + (1 - \theta)p_1 &\notin C \\ 0 < \theta < 1 \end{aligned} \quad (1.14)$$

için geçerlidir. Bir dışbükey küme ile ilgin küme arasındaki fark Şekil1.23(b)'de görülmektedir. Burada  $C_1$  kümesi ilgin değil ama dışbükey,  $C_2$  kümesi hem dışbükey ve hem de ilgin kümedir. Dışbükeylik tanımlaması, kümenin çizgisel olmaması halinde ve mesela bir alan, hacim ya da daha büyük boyutlu bir ortam olması halinde değişir. Bu durumda;

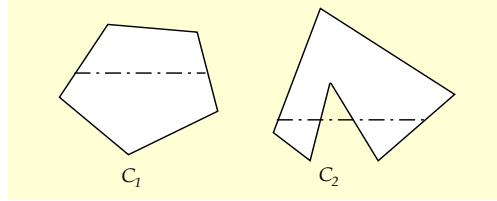
$$\begin{aligned} p_1, p_2 &\in C \\ \theta p_2 + (1 - \theta)p_1 &\in C \end{aligned} \quad (1.15)$$

tanımlaması geçerlidir.

<sup>i</sup> Convex



Müteakip şekilde iki boyutlu ortamda alana sahip dışbükey olan ve olmayan küme örnekleri görülmektedir.



Şekil 1.24 İki boyutlu ortamda dışbükey ( $C_1$ ) ve dışbükey olmayan ( $C_2$ ) kümeler.

### g. Kümede sıralama

İçinde farklı noktaları barındıran kümelerde sıralama önemli bir kavram olup farklı boyut ve ölçütlere göre sıralama yapmak mümkündür. Tamamen sıralanmış küme kavramı öncelikle bir kümeye ve küme elemanları arasındaki ilişkiye dayanır. Küme elemanları arasındaki ilişkinin simgesel olarak " $\leq$ " veya sözel bağlamda "küçük eşit" olması halinde küme kısmi sıralanmış küme<sup>ii</sup> olarak tanımlanır. Sıralamanın getirdiği iki önemli kavram söz konusu olup bu kavramlardan ilki azamilik ve diğeri de asgariliktir. Literatürde asgari eleman olarak geçen kavram<sup>iii</sup> şu şekilde tanımlanabilir:

- ▶ Kısmi olarak sıralanmış bir  $(P, \leq)$  kümesi ve bu küme tarafından kapsanan  $S \subset P$  altkümesi dâhilinde eğer  $m \in S$  noktası ve tüm  $p \in S$  noktaları için  $m \leq p$  geçerli ise  $m$  noktası  $S$  altkümesinin minimum elemanıdır.

Aynı yaklaşım azami eleman tanımı için de yapılabilir. Dolayısıyla maksimum eleman için;

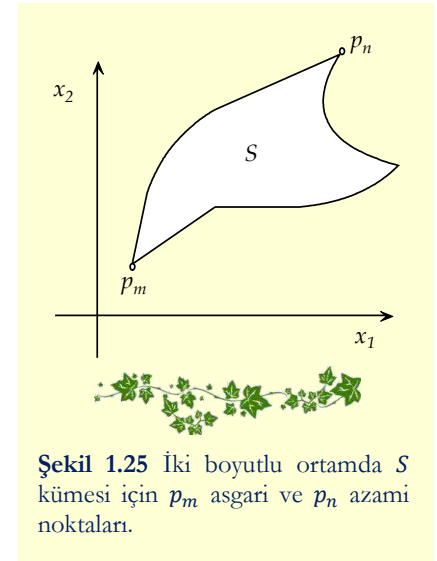
- ▶ Kısmi olarak sıralanmış bir  $(P, \leq)$  kümesi ve bu küme tarafından kapsanan  $S \subset P$  altkümesi dâhilinde eğer  $n \in S$  noktası ve tüm  $p \in S$  noktaları için  $p \leq n$  geçerli ise  $n$  noktası  $S$  altkümesinin maksimum elemanıdır.

tanımı yapılabilir.

<sup>i</sup> Totally ordered set

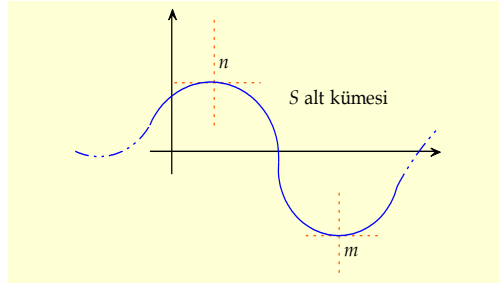
<sup>ii</sup> Partially ordered set

<sup>iii</sup> Minimal element



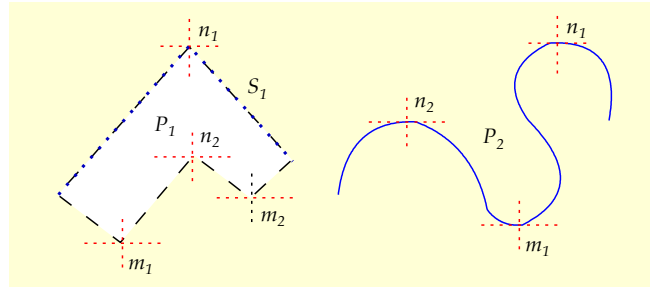
Şekil 1.25 İki boyutlu ortamda  $S$  kümesi için  $p_m$  asgari ve  $p_n$  azami noktaları.





Şekil 1.26 İki boyutlu ortamda  $S$  altkümesi için maksimum ve minimum noktalar.

Öte yandan bu iki tanımlama şu hususu da beraberinde getirir: farklı alt kümeler farklı asgari veya azami elemanlara sahip olabilirler. Dolayısıyla  $S$  altkümelerine ait maksimum veya minimum değerler arasında da bir azami ve asgari nokta tanımlaması yapmak mümkündür. Müteakip şekilde  $P_1$  kümesine ait koyu noktali hat şeklindeki  $S_1$  alt kümesinin  $n_1$  azami noktası, aynı zamanda  $P_1$  kümesinin de en azami noktasıdır. Buna mukabil başka bir alt kümeye ait  $n_2$  noktası, o alt kümenin azami noktası iken  $P_1$  kümesi için en azami nokta değildir. Benzer şekilde  $P_1$  kümesinin asgari noktası  $m_1$  noktasıdır. Diğer bir örnek küme olan  $P_2$  kümesinin en azami noktası  $n_1$ , en asgari noktası ise  $m_1$  noktasıdır.

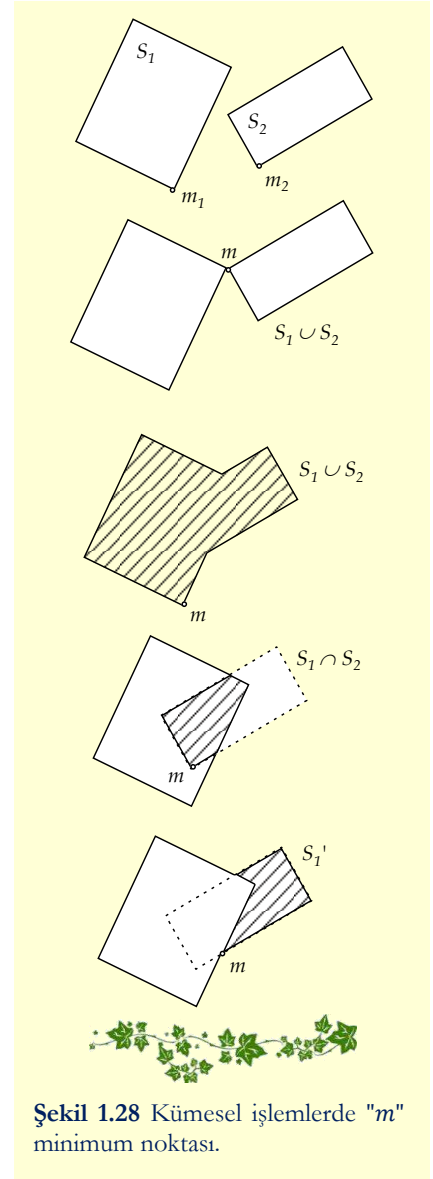


Şekil 1.27 İki boyutlu ortamda  $P_1$  ve  $P_2$  gibi kümelere ait maksimum ve minimum noktalar.

Bu bağlamda azami veya asgari ama nihayetinde uç noktalar için yerel<sup>i</sup> ve genel<sup>ii</sup> uç nokta tanımlamak mümkündür. Buna göre  $S_i$  gibi altkümelere ait uç noktalara yerel uç nokta denir.

<sup>i</sup> Local

<sup>ii</sup> Global



Şekil 1.28 Kümesel işlemlerde " $m$ " minimum noktası.

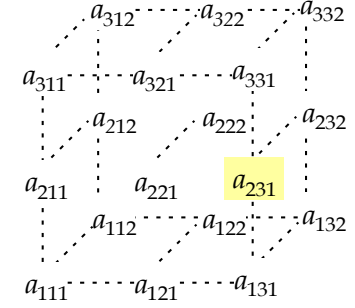




Uç noktanın azami nitelikte olması halinde yerel maksimum, asgari nitelikte olması halinde yerel minimum adı verilir. Birden çok sayıda uç noktanın söz konusu olması halinde müteakip tanımlar yapılabilir:

- ▶ Kısmi olarak sıralanmış bir  $(P, \leq)$  kümesi ve bu küme tarafından kapsanan  $S_i \subset P$  altkümeleri dâhilinde eğer  $n \in S_i$  noktası ve tüm  $p \in P$  noktaları için  $p \leq n$  geçerli ise  $n$  noktası  $P$  kümesinin genel maksimum elemanıdır. Eğer  $m \in S_i$  noktası ve tüm  $p \in P$  noktaları için  $m \leq p$  geçerli ise  $m$  noktası  $P$  kümesinin genel minimum noktasıdır.

Böyle bir tanımlamaya göre Şekil 1.27'de verilen  $P_1$  kümesinin genel maksimumu  $n_1$ , genel minimumu  $m_1$  noktasıdır. Diğer uç noktalar olan  $n_2$  ve  $m_2$  noktaları, sırasıyla yerel maksimum ve yerel minimum noktalarıdır.



Şekil 1.29  $3 \times 3 \times 2$  ebadında bir sayı dizini örneği.

## ğ. Kümelerde sıralama

Aynı veya farklı nitelikteki birden fazla sayıda kümenin söz konusu olduğu hallerde kümeler arası ilişkiler önem kazanır. Bu bağlamda kümeler arasında; birleşim, kesişim, fark ve tümlenme söz konusu olabilir. Dikkate alınan işleme göre sıralama ve azami/asgari nokta tespiti farklılıklar gösterir. Buna göre Şekil 1.28 dikkate alındığında,  $S_1$  ve  $S_2$  gibi iki farklı kümenin minimum noktaları  $m_1$  ve  $m_2$  noktalarıdır. Söz konusu kümelerin tek bir nokta itibari ile birleşimi söz konusu olduğunda oluşan  $S_1 \cup S_2$  kümesinin  $m$  asgari noktası aynı zamanda temas noktası olur. Zira söz konusu noktadan başka hiçbir nokta her iki kümenin üyesi değildir. Kümelerin kesişir şekilde birleşimi söz konusu olduğunda ortaya çıkan  $S_1 \cup S_2$  kümesinin  $m$  asgari noktası genel minimum noktadır. Kümelerin kesişimi sonrası ortaya çıkan  $S_1 \cap S_2$  kesişim kümesinin genel minimum noktası  $m$  noktasıdır. Öte yandan belli bir kesişim bölgesine sahip  $S_1 \cup S_2$  birleşim kümesi dikkate alındığında  $S_1$  kümesinin tümleyeni olan  $S_1^c$  kümesinin genel minimum noktası yine  $m$  noktasıdır. Özellikle kümesel işlemlerden kesişim işlemi sonucu ortaya çıkan yeni kümenin genel minimum noktasının, kesişim noktaları üzerinde olduğu gözlenmektedir. Bu bağlamda kesişim kümelerinin genel minimum noktasının, kesişim hattı dâhilinde aranması gerektiği söylenebilir.

## 1.2.2 Cebir

Kökeni Arapça'da "kırılmış parçaları yeniden birleştirmek" anlamına gelen Cebir ismini, aynı zamanda söz konusu bilim dalının kurucusu olarak da görülen Al-Khwarizmi'ye (780-850)



borçludur<sup>(34)</sup>. Aritmetikten farklı olarak cebir dalında, sayısal ifadelerle beraber harfsel simgelerin kullanımı söz konusudur. Lineer cebir, yöney uzayları ve doğrusal eşleşmelerle<sup>i</sup> ilgilenir. Daha detay bir bakış açısıyla lineer cebir; sayı dizinleri ve uzaylarını, doğrusal dönüşümleri, özdeğer ve özyöneyle<sup>ii</sup> inceler.

### a. Etiketleme simgeleri ve dizinler

Sayıların etiketlenmesi ya da daha açık bir ifade ile belli bir düzen içerisinde sıralanması ve düzenlenmesi genellikle alt indislerle gerçekleştirilir. Bu bağlamda  $a$  gibi herhangi bir sayının sayı dizini<sup>iii</sup> içerisindeki konumu;  $a_i$ ,  $a_{ij}$ ,  $a_{ijk}$  veya daha fazlası gibi alt indislerle gösterilebilir. Söz konusu alt indisler pozitif tam sayılar olup;

$$i = 1,2,3, \dots, n; j = 1,2,3, \dots, m; k = 1,2,3, \dots, p \quad (1.16)$$

şeklinde sıralanabilirler. Genellikle ilk alt indise satır numarası, ikinci alt indise sütun numarası ve son indise de derinlik numarası adı verilir. Bu bağlamda Şekil 1.29'da görülen  $a_{231}$  sayısı etiketlemesi; 2'nci satır, 3'üncü sütun ve 1'inci derinlik konumunda bulunan sayı anlamına gelir. Farklı düzenleme ve sıralamalar farklı sayı dizinlerini oluşturur. Bununla beraber sayı dizinlerini alt indis sayısına göre gruplandırmak mümkündür. Buna göre 0-dizinli, 1-dizinli, 2-dizinli ve nihayetinde 3-dizinli sayı grupları örnek dizinler olarak sıralanabilir. Daha büyük dizinler kuramsal olarak düşünülebilir, ancak fiziksel dünyada karşılıklarını ifade etmek pek mümkün değildir.

### b. 0-dizinli sayılar

0-dizinli bir dizi<sup>iv</sup> esasen dizin olmayıp, herhangi bir alt indise ihtiyaç duyulmadan ifade edilen sayıyı içerir. Bu tür sayıları aynı zamanda sayıl olarak ifade etmek mümkündür. Bununla beraber daha üst seviyeli dizinlerin doğal olarak kapsayacağı bir konumlama olup;

$$[a] \quad (1.17)$$

veya

<sup>i</sup> Linear mapping

<sup>ii</sup> Özdeğer İngilizce'de "eigenvalue", özyöney ise "eigenvector" olarak geçer.

<sup>iii</sup> Sayı dizini İngilizce'de "tensor" veya "array of numbers" olarak geçer.

<sup>iv</sup> İngilizce'de "0-rank tensor" olarak geçer.



$$a_{ijk}, \quad i = j = k = 1 \quad (1.18)$$

şeklinde dizinsel bir gösterim de mümkündür.

Fiziksel manada sayılı, sadece şiddeti olan bir olguyu ifade eder. Doğada sayısal bağlamında ifade edilebilecek olgular arasında basınç, sıcaklık veya yoğunluk gibi fiziksel özellikler sayılabilir:

$$\begin{aligned} p &= 1 \text{ atm} \\ T &= 15 \text{ }^\circ\text{C} \\ \rho &= 1,2256 \text{ kg/m}^3 \end{aligned} \quad (1.19)$$

Zira bu tür olgular, belirli bir mekân ve zamanın karşılık geldiği noktaya ait olgulardır. Söz konusu özellikler herhangi bir yöne sahip olmayıp, sadece belli bir şiddete sahiptirler.

### c. 1-dizimli sayılar

1-dizimli sayılar sadece bir alt indise sahip olup;

$$a_i, \quad i = 1, 2, \dots, n \quad (1.20)$$

veya

$$\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_n \end{bmatrix} \quad (1.21)$$

biçiminde ya da;

$$\mathbf{a}^T = [a_1 \quad a_2 \quad a_3 \quad \dots \quad a_n] \quad (1.22)$$

şeklindeki devriği<sup>iii</sup> ile gösterilebilir. Fiziksel dünyada örnek olarak dikkate alınabilecek 1-dizimli sayılar arasında yöney, dördey ve 4-yöney<sup>iii</sup> kavramları sayılabilir.

<sup>i</sup> Scalar

<sup>ii</sup> Transpose

<sup>iii</sup> İngilizce'de yöney "vector", dördey "quaternion" ve 4-yöney "4-vector" olarak geçer.



## (1) Yöney

“Euclidean” uzayının bir kavramı olan yöney kavramı, fiziksel bilimlerde yönü ve şiddeti olan olgu olarak tanımlanır. Hatırlanacağı üzere daha önce tanımlanan sayıl, sadece şiddeti olan olguydu. Yöney kavramına verilebilecek örnekler arasında konum, hız veya ivme sayılabilir. Konum esasen bir noktanın tanımlanmasıdır. Ancak noktanın belirli bir merkeze göre tanımlanması yapıldığı için mekânda ikinci bir nokta ortaya çıkmakta olup bu da iki nokta arası konumlanma anlamına gelir. Dolayısıyla konum belli bir yöne sahiptir. O yöndeki mesafe, konum yöneyinin şiddetini ifade eder. Söz konusu konumdan başka bir konuma geçildiğinde yine ikincil bir yönlenme gerçekleştirilmiş olup bu durumda tanımlama belli bir merkeze göre değil, ilk ve ikinci konumlara göre yapılmaktadır. Bu değişimin birim zamanda gerçekleştiği miktara hız yöneyi adı verilir. Daha açık bir ifade ile hız yöneyi, ilk ve ikinci konumlar arası yöne ve aradaki mesafeye ki yöneyin şiddeti manasında gelir, sahip bir yöneydir. Konum yöneyi, “Euclidean” uzayı olarak tanımlanan üç boyutlu ortamda şiddet ve yönü içerecek şekilde;

$$\mathbf{r} = \|\mathbf{r}\| \mathbf{e}_r \quad (1.23)$$

olarak tanımlanabilir. Burada  $\|\mathbf{r}\|$ , yöneyin şiddetini göstermekte olup bazen sadece sayıl niteliğindeki  $r$  ile de simgelenabilmektedir;  $\mathbf{e}_r$  ise o yöndeki birim yöneyi<sup>i</sup> ifade eder. Bu ifadeye göre;

$$\mathbf{e}_r = \frac{\mathbf{r}}{r} \quad (1.24)$$

eşitliğini yazmak mümkündür. Tabiatıyla birim yöneyin şiddeti birdir. Diğer taraftan yöneyi mevcut yönü ile değil, standart bir koordinat sistemini dikkate alarak tanımlamak da mümkündür. Bu durumda  $n$  boyutlu bir ortam uzayında aynı konum yöneyini;

$$\mathbf{r} = \sum_{i=1}^n r_i \mathbf{e}_i \quad (1.25)$$

şeklinde ifade edebiliriz. Burada  $r_i$ , bahse konu yöneyin koordinat sisteminin  $i$ 'nci eksenine ait bileşenini,  $\mathbf{e}_i$  aynı eksene ait birim yöneyi<sup>iii</sup> ifade eder. Bu durumda yöneyin şiddetini;

<sup>i</sup> Magnitude

<sup>ii</sup> Normalized vector

<sup>iii</sup> Eksen üzerindeki birim yöney İngilizce'de “unit vector” olarak geçer.

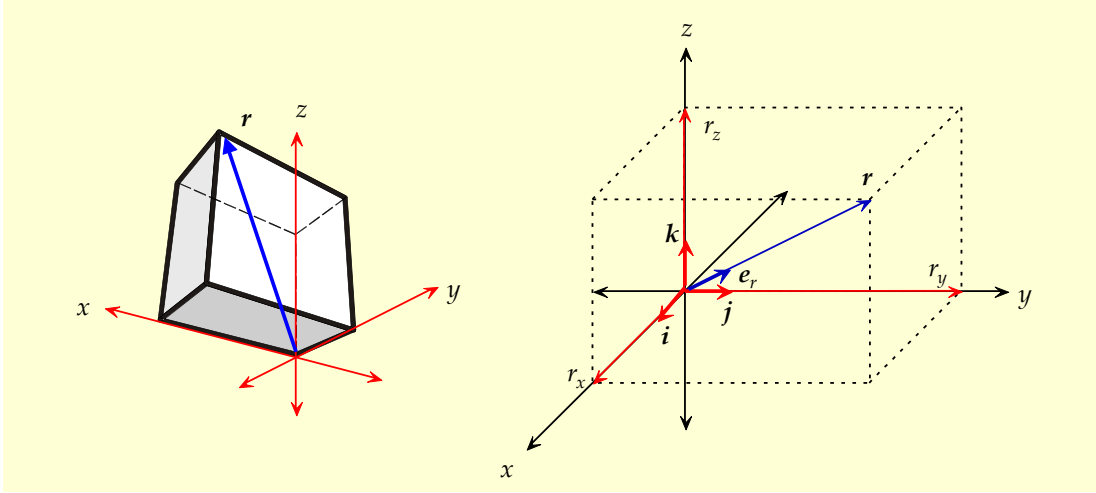




$$r = \left[ \sum_{i=1}^n r_i^2 \right]^{\frac{1}{2}}$$

(1.26)

denklemini ile bulabiliriz.



Şekil 1.30 Üç boyutlu ortamda yöney ve bileşenleri.

“Euclidean” uzayında Kartezyen koordinat sistemini kullanarak konum yöneyini:

$$\mathbf{r} = r_x \mathbf{i} + r_y \mathbf{j} + r_z \mathbf{k}$$

(1.27)

biçiminde ifade etmek mümkündür. Burada yöneyin her bir eksenindeki şiddetine karşılık  $\mathbf{i}$ ,  $\mathbf{j}$  ve  $\mathbf{k}$  gibi birim yöneyler tanımlanmaktadır. Söz konusu birim yöneyleri birbirlerine dik<sup>i</sup> olup şiddetleri de birdir. Diğer taraftan birim yöneylerin çıkış noktası karmaşık sayılardır ve İrlandalı matematikçi Hamilton (1805-1865) tarafından üç boyuta genelleştirilmiştir.

<sup>i</sup> Kartezyen koordinatlarda “Standard basis” olarak geçer.

<sup>ii</sup> Dik konumlanma İngilizce’de “orthogonality” olarak geçer.







## (2) Dördey

Dördeyi kavramı, 0-dizimli sayı ile 1-dizimli sayı grubunun birleşimi olarak tanımlanabilir. Bu bağlamda dördeyi;

$$\mathbf{p} = (t, \mathbf{r}) \quad (1.28)$$

ifadesi ile temsil edebiliriz. Burada  $t$  sayılı,  $\mathbf{r}$  “Euclidean” uzayında üç bileşeni olan yöneyi ifade eder. Fiziksel olarak uzay-zaman ortamında bir konumu ifade ettiğimizde; sayıl niteliğindeki  $t$  zamanı, yöney niteliğindeki  $\mathbf{r}$  ise konumu gösterir. Dördey kavramının diğer bir fiziksel manası, konum ve o konum etrafındaki dönüyü ifade etmesidir. Buna göre  $t$  dönü açısını,  $\mathbf{r}$  dönülen eksenini gösterir. Kartezyen koordinat sisteminde bir dördeyi;

$$\mathbf{p} = t + r_x \mathbf{i} + r_y \mathbf{j} + r_z \mathbf{k} \quad (1.29)$$

şeklinde açabiliriz. Dördeyin eşleniğini<sup>i</sup>;

$$\mathbf{p}^* = (t, -\mathbf{r}) \quad (1.30)$$

olarak tanımlamak mümkündür. Dolayısıyla  $\mathbf{p}$  ve eşleniği  $\mathbf{p}^*$  dördeylerinin birbirlerine göre bir aynasal yansıma olduğunu söyleyebiliriz. Bir dördey ile başka bir dördeyin çarpımını bize yeni çarpım işlemlerini<sup>iii</sup> tanımlamaya yönelmiştir.

$\times$	1	$\mathbf{i}$	$\mathbf{j}$	$\mathbf{k}$
1	1	$\mathbf{i}$	$\mathbf{j}$	$\mathbf{k}$
$\mathbf{i}$	$\mathbf{i}$	-1	$\mathbf{k}$	$-\mathbf{j}$
$\mathbf{j}$	$\mathbf{j}$	$-\mathbf{k}$	-1	$\mathbf{i}$
$\mathbf{k}$	$\mathbf{k}$	$\mathbf{j}$	$-\mathbf{i}$	-1

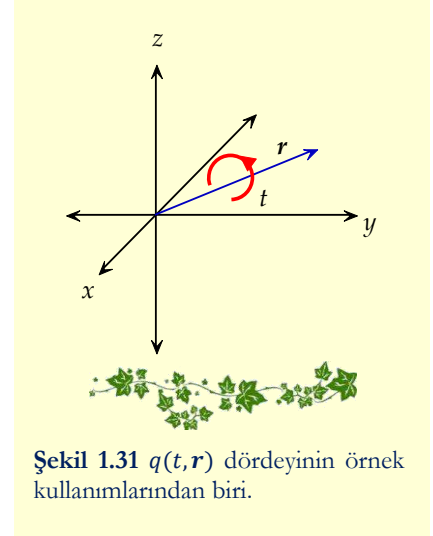
Çizelge 1.1 Dördey çarpım çizelgesi.

Bu bağlamda dikkate alınan işlem tanımlamaları, dördey çarpım çizelgesi olup işlemlerin temelinde karmaşık sayı kavramı ve  $i = \sqrt{-1}$  tanımı yatmaktadır. Dolayısıyla sayıl niteliğindeki  $t$  kısmı reel ve yöney niteliğindeki  $\mathbf{r}$  kısmı sanal sayıları temsil eder. Buna göre  $\mathbf{p}$  ve  $\mathbf{p}'$  gibi iki dördeyin çarpımını aşağıdaki gibi ifade edebiliriz;

<sup>i</sup> Quaternion

<sup>ii</sup> Complex conjugate

<sup>iii</sup> Buradaki işlem kavramı İngilizce’de “metric” olarak geçer.



Şekil 1.31  $q(t, \mathbf{r})$  dördeyinin örnek kullanımlarından biri.



$$\begin{aligned} \mathbf{p} &= t + r_x \mathbf{i} + r_y \mathbf{j} + r_z \mathbf{k} \\ \mathbf{p}' &= t' + r'_x \mathbf{i} + r'_y \mathbf{j} + r'_z \mathbf{k} \\ \mathbf{p} \mathbf{p}' &= (t, \mathbf{r})(t', \mathbf{r}') \\ \mathbf{p} \mathbf{p}' &= (tt' - \mathbf{r} \cdot \mathbf{r}', tr' + rt' + \mathbf{r} \times \mathbf{r}') \end{aligned} \quad (1.31)$$

Yapılan çarpım işlemlerinde iki özel işlem tanımlanmakta olup bunlardan ilki "." işareti ile gösterilen sayısal çarpım ve ikincisi de "×" işareti ile gösterilen yöneysel çarpımdır. Dikkat edileceği üzere iki dördeyin çarpımından yine başka bir dördey elde edilmektedir.

### (3) 4-Yöney

Yöney ve akabinde tanımlanan dördey kavramları, her ne kadar 1-dizimli bir sayı grubunu temsil etse de her ikisi de "Euclidean" uzayına ait kavramlardır. Oysa günümüzde zaman kavramı, sayıl olmaktan ziyade yöneysel bir kavram olarak dikkate alınmakta ve bu tür uzay-zaman ortamına "Minkowski" uzayı adı verilmektedir. Böyle bir ortamda 4-yöney kavramı geçerli olup tanımı;

$$\mathbf{A} = A_0 \mathbf{e}_0 + A_1 \mathbf{e}_1 + A_2 \mathbf{e}_2 + A_3 \mathbf{e}_3 \quad (1.32)$$

şeklindedir. Aynı tanımı, "Euclidean" uzayını dikkate alarak

$$\mathbf{A} = A_0 \mathbf{e}_0 + A_x \mathbf{i} + A_y \mathbf{j} + A_z \mathbf{k} \quad (1.33)$$

biçiminde yazmak da mümkündür. 4-yöney gösterimi, dördey gösterimi ile de betimlenebilir:

$$\mathbf{A} = (ct, \mathbf{r}) \quad (1.34)$$

Böyle bir gösterimde  $c$  ışık hızını,  $t$  zamanı ve  $\mathbf{r}$  konumu ifade eder. İki farklı 4-yöneynin çarpımı sadece sayısal çarpım olarak dikkate alınır. Dolayısıyla ortaya sayısal bir ifade çıkar.

### ç. 2-dizimli sayılar

2-dizimli sayılar iki alt indise sahip olup;

$$a_{i,j}, \quad \begin{matrix} i=1,2,\dots,n \\ j=1,2,\dots,m \end{matrix} \quad (1.35)$$

veya



$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \cdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix} \quad (1.36)$$

biçiminde gösterilir. 2-dizimli sayı eldesi; genellikle iki 1-dizimli sayı grubunun farklı bir biçimde çarpımıyla, diğer bir ifade ile diyadik çarpım yöntemi ile elde edilebilir. Dolayısıyla dizin sayısı, bir üst gruba yükselmiş olur. Diyadik çarpımını farklı formlarda yazmak ve ifade etmek mümkündür. Örneğin;

$$\mathbf{a} = \sum_{i=1}^n a_i \mathbf{e}_i, \quad \mathbf{b} = \sum_{j=1}^m b_j \mathbf{e}_j \quad (1.37)$$

iki farklı yöney olmak üzere diyadik çarpımını;

$$\mathbf{ab} = \sum_{i=1}^n \sum_{j=1}^m a_i b_j \mathbf{e}_i \mathbf{e}_j \quad (1.38)$$

şeklinde yazmak mümkündür. Aynı diyadik çarpımı, 2-dizimli sayı grubu şeklinde ifade etmek istersek;

$$\mathbf{ab} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_n \end{bmatrix} [b_1 \quad b_2 \quad b_3 \quad \cdots \quad b_m] = \begin{bmatrix} a_1 b_1 & a_1 b_2 & \cdots & a_1 b_m \\ a_2 b_1 & a_2 b_2 & \cdots & a_2 b_m \\ \vdots & \vdots & \cdots & \vdots \\ a_n b_1 & a_n b_2 & \cdots & a_n b_m \end{bmatrix} \quad (1.39)$$

sayı dizinini yazabiliriz. Bu iki farklı gösterimin temelinde birim yöneylerin;

$$\mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}_{n \times 1}, \quad \mathbf{e}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}_{n \times 1}, \quad \mathbf{e}_i = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix}_{n \times 1} \quad (1.40)$$

biçimindeki tanımları yatmaktadır.

<sup>i</sup> Dyadic





$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_i \\ \vdots \\ x_n \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_i \\ \vdots \\ b_n \end{bmatrix} \quad (1.42)$$

şeklindeki  $n$  boyutlu vektörlerle gösterilebilir. Bilinmeyen değişkenlere ait  $a_{ij}$  katsayıları ise;

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1j} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2j} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{i1} & a_{i2} & \cdots & a_{ij} & \cdots & a_{in} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nj} & \cdots & a_{nn} \end{bmatrix} \quad (1.43)$$

tarzındaki 2-dizimli bir sayı grubu ile temsil edilebilir. Bahse konu üç sayı grubunu;

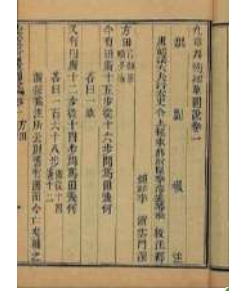
$$Ax = b \quad (1.44)$$

eşitliği ile ilişkilendirmek mümkündür.

### (1) Doğrusal denklem takımlarının çözümü

Matris ve matrislerle ilgili kavramlara geçmeden önce doğrusal denklem takımlarının çözümü üzerinde durmanın faydalı olacağı değerlendirilmektedir. İfade edildiği üzere doğrusal denklem takımlarının çözümü üzerine yapılan çalışmalar, oldukça eskilere dayanmakta olup bu meyanda M.S. 3'üncü yüzyılda Liu Hui (220-?) tarafından elden geçirilen ve tekrar basılan “Matematik sanatı üzerine dokuz bölüm” isimli imzasız kitaptan bir alıntı, örnek olarak verilebilir;

- ▶ “Verilen 3 deste iyi kalite çeltik, 2 deste orta kalite çeltik ve 1 deste düşük kalite çeltik 39 Dou<sup>i</sup> gelmektedir. 2 deste iyi kalite çeltik, 3 deste orta kalite çeltik ve 1 deste düşük kalite çeltik 34 Dou gelmektedir. 1 deste iyi kalite çeltik, 2 deste orta kalite çeltik ve 3 deste düşük kalite çeltik ise 26 Dou yapmaktadır. Söyleyin bakalım her bir kalitedeki çeltiğin destesi kaç Dou yapmaktadır? <sup>(36)</sup>”



Şekil 1.33 “Matematik sanatı üzerine dokuz bölüm” adlı eserden örnek bir sayfa <sup>(35)</sup>.

<sup>i</sup> Dou bir hacim birimi olup miktar olarak yaklaşık 2 litre demektir.







Bahsi geçen kitapta daha sonra sorunun çözüm yöntemi anlatılmakta olup ifade edilen yöntem, yaklaşık 2000 yıl sonra Gauss tarafından geliştirilen ve günümüzde Gauss eleme yöntemi olarak adlandırılan yöntemin temelini oluşturmaktadır. Sözel olarak belirtilen bu ifadeleri, matematiksel olarak ifade etmek istersek:

$$\begin{aligned}3x_1 + 2x_2 + x_3 &= 39 \\2x_1 + 3x_2 + x_3 &= 34 \\x_1 + 2x_2 + 3x_3 &= 26\end{aligned}$$

şeklindeki üç doğrusal denklemi içeren bir denklem takımını yazabiliriz. Burada  $x_1$  iyi kalite çeltiğin kapladığı hacmi,  $x_2$  orta kalite çeltiğin kapladığı hacmi ve  $x_3$  düşük kalite çeltiğin kapladığı hacmi temsil etmektedir.

Lineer denklem takımlarının çözümü bağlamında üç değişik durumdan bahsedilebilir. Söz konusu haller, çözümün varlığı veya sayısı dikkate alınarak tanımlanır. Buna göre; bir denklem takımının çözüm kümesi olabilir, sadece bir çözüm noktası olabilir veya herhangi bir çözümü olmayabilir.

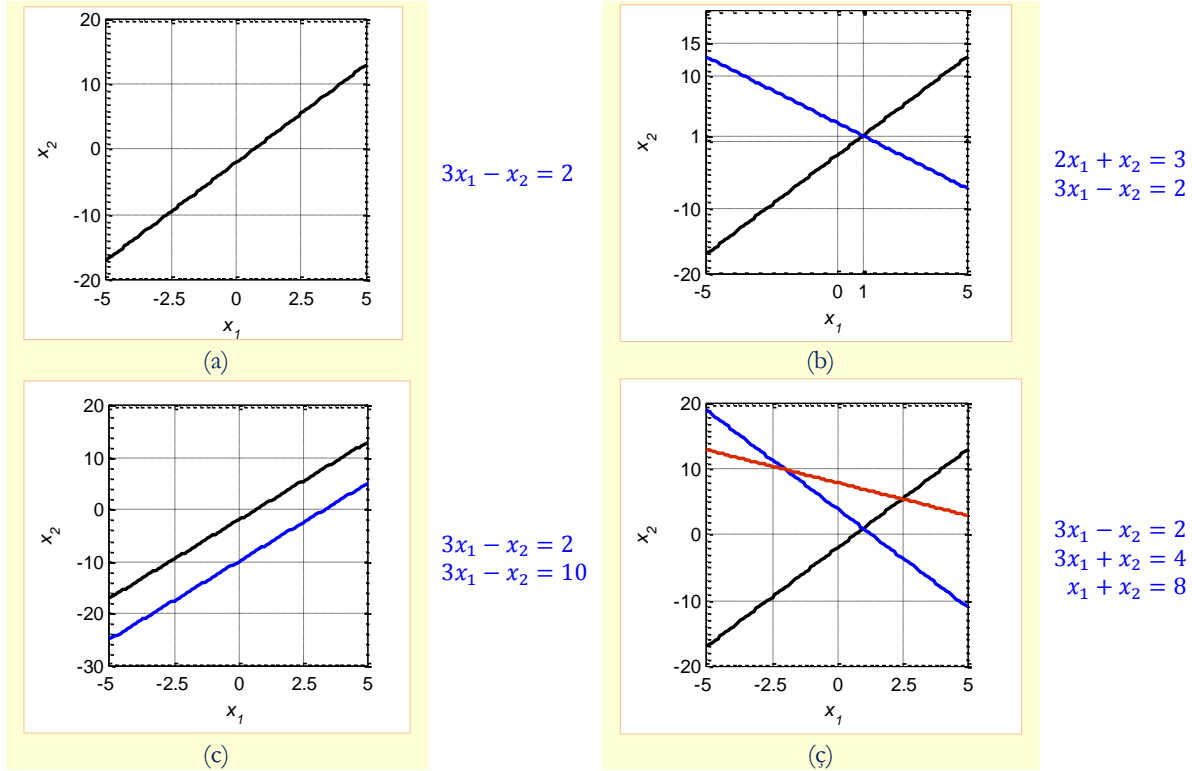
Örneğin Şekil 1.34(a) dâhilinde tanımlı denklemi sağlayan sonsuz sayıda çözüm noktası ya da diğer bir ifade ile noktalardan oluşan çözüm kümesi mevcuttur. Zira doğru üzerindeki her nokta, denklemi sağlamaktadır. Burada dikkat çeken bir diğer husus, iki bilinmeyen değişken sayısına karşılık sadece bir denklemin varlığıdır. Aynı şeklin (b) seçeneğinde, tanımlı denklemleri sağlayan çözüm kümesi sadece bir noktadan ibarettir: (1,1). Verili (c) seçeneğindeki denklemleri sağlayan çözüm kümesi ise boş kümedir. Şekilden de görüldüğü üzere burada doğrusal denklemleri sağlayan çözüm kümeleri birbirine paralellik arz etmektedir. Dolayısıyla denklem takımının çözümü yoktur. Benzer bir durum (ç) seçeneğinde verili denklemler için geçerlidir. Böyle bir durumda her üç denklemi sağlayan bir çözüm noktasından bahsedilemez.

Her zaman doğru olmasa da verilen örneklerden bazı sonuçlar çıkarmak mümkündür. Buna göre bilinmeyen niteliğindeki bağımsız değişkenlerin sayısından daha az sayıda denklemin olması halinde<sup>i</sup> çözüm kümesi sonsuz sayıda olabilir. Bilinmeyen sayısı ile denklem sayısının aynı olması halinde genellikle sadece bir çözüm vardır. Bilinmeyen sayısından çok daha fazla sayıda denklem içeren denklem takımının genellikle çözümü yoktur.

---

<sup>i</sup> Denklem takımı, denklem sayısının bilinmeyen değişken sayısından az olması halinde “undetermined system”, fazla olması halinde sistem “overdetermined system” olarak adlandırılır.





Şekil 1.34 Doğrusal denklem takımlarında olası çözüm seçenekleri.

Verilen örneklerden çıkarılan sonuçları iki kavrama dayandırmak mümkündür. Bu kavramlar, doğrusal bağımsızlık<sup>i</sup> ve tutarlılık<sup>ii</sup> kavramlarıdır. Bilinmeyen sayısı ile denklem sayısının aynı sayıda olması, her zaman sadece bir çözüm anlamına gelmeyebilir. Denklem ve değişken sayılarından başka, denklemlerin bağımsız ve tutarlı olmaları da gerekir. İlave olarak çözümde önem arz eden bir diğer konu da sayısal hassasiyet olarak tanımlanabilecek ve denklem takımlarının, sayısal basamak değerlerine aşırı hassas olup olmamasını irdeleyen husustur.

<sup>i</sup> Linear independence

<sup>ii</sup> Consistency



### (a) Doğrusal bağımsızlık

Doğrusal bağımsızlık, yöneyler arasında geçerli olan bir kavramdır. Buna göre  $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n$  gibi  $n$  tane yöneyin söz konusu olduğu bir yöney kümesi ve her bir yöneye ait  $\alpha_1, \alpha_2, \dots, \alpha_n$  gibi sayıların olduğu bir küme arasında:

$$\sum_{i=1}^n \alpha_i \mathbf{v}_i = 0 \quad (1.45)$$

eşitliğinin geçerli olabilmesi için eğer tüm  $\alpha_i$  değerlerinin sıfır olması gerekiyorsa bu takdirde söz konusu yöney kümesi doğrusal olarak bağımsızdır. Eşitliğin sağlanması adına herhangi bir  $\alpha_i$  değerinin sıfırdan farklı olması halinde yöney kümesi dâhilindeki yöneylerin bazıları arasında lineer bir ilişki söz konusudur. Bu söylem, yöney kümesindeki bazı yöneylerin, aynı küme üyesi olan diğer yöneylerin uygun katsayılar kullanılarak toplanması ile elde edilebileceği anlamına da gelir. Daha önce 2 boyutlu bir ortamda verilen müteakip denklem takımını yeniden ele alabiliriz:

$$\begin{aligned} 3x_1 - x_2 &= 2 \\ 2x_1 + x_2 &= 3 \end{aligned}$$

Söz konusu bu denklemlerin katsayılarını içeren  $A$  matrisini:

$$A = \begin{bmatrix} 3 & -1 \\ 2 & 1 \end{bmatrix}$$

şeklinde ifade edebiliriz. Bu ifadede ilk satır  $\mathbf{v}_1^T = [3 \quad -1]$  yöneyini, ikinci satır ise  $\mathbf{v}_2^T = [2 \quad 1]$  yöneyini temsil etmektedir. Söz konusu yöneyler arasında:

$$\sum_{i=1}^n \alpha_i \mathbf{v}_i = 0, \quad \alpha_1 \times \mathbf{v}_1 + \alpha_2 \times \mathbf{v}_2 = 0$$

ifadesi ve buradan da;

$$\begin{aligned} 3\alpha_1 + 2\alpha_2 &= 0 \\ -\alpha_1 + \alpha_2 &= 0 \end{aligned}$$

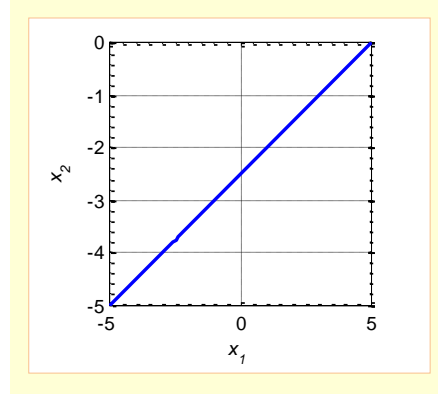
ilişkisi yazılabilir. Elde edilen denklemler çözüldüğünde:



$$\alpha_1 = 0, \quad \alpha_2 = 0$$

sonucuna ulaşılır ki bu da yöneylerin doğrusal olarak bağımsız olduklarını gösterir. Nitekim denklem takımının çözüm kümesi sadece bir noktayı içermektedir. Lineer bağımlılığa örnek olarak müteakip iki değişkenli iki denklemi dikkate alalım:

$$\begin{aligned} x_1 - 2x_2 &= 5 \\ 3x_1 - 6x_2 &= 15 \end{aligned}$$



Bir önceki örneğe benzer şekilde, söz konusu bu denklemlerin katsayılarını içeren  $A$  matrisini:

$$A = \begin{bmatrix} 1 & -2 \\ 3 & -6 \end{bmatrix}$$

biçiminde ifade edebiliriz. Bu ifadede ilk satır  $\mathbf{v}_1^T = [1 \quad -2]$  yöneyini, ikinci satır  $\mathbf{v}_2^T = [3 \quad -6]$  yöneyini temsil etmektedir. Söz konusu yöneyler arasında:

$$\begin{aligned} \sum_{i=1}^n \alpha_i \mathbf{v}_i &= \alpha_1 \times \mathbf{v}_1 + \alpha_2 \times \mathbf{v}_2 = 0 \\ \alpha_1 + 3\alpha_2 &= 0 \\ -2\alpha_1 - 6\alpha_2 &= 0 \\ \alpha_1 &= -3\alpha_2 \end{aligned}$$

ilişkisi yazılabilir. Dolayısıyla  $\alpha_i$  katsayıları sıfırdan farklı değerler de alabilmekte olup yöneyler doğrusal olarak birbirlerine bağımlıdır. Aralarında doğrusal bağımsızlık söz konusu değildir. Nitekim her iki denklemi sağlayan noktalar aynı noktalar olup çözüm sadece bir noktayı değil, sonsuz sayıda noktayı içermektedir.



### (b) Tutarlılık

Tutarlılık, yöneyler arasındaki ilişkiden ziyade denklemler arasındaki ilişki ile ilgilidir. Denklem takımındaki denklemler birbirleri ile çelişki içerisinde olmamalı, mantıksal bir uyumluluk sergilemelidir. Aksi halde denklemler arasında yapılan işlemler neticesinde eşitliğin doğasına aykırı sonuçlarla karşılaşılabilir. Örneğin daha önce de ifade edilen müteakip denklem takımını dikkate alalım:

$$\begin{aligned} 3x_1 - x_2 &= 2 \\ 3x_1 - x_2 &= 10 \end{aligned}$$

Bu iki denklem birbirinden çıkarıldığında;

$$\begin{aligned} 3x_1 - x_2 &= 2 \\ 3x_1 - x_2 &= 10 \\ - \dots\dots\dots & \\ 0 &= -8 \end{aligned}$$

sonucu elde edilebilir. Görüldüğü üzere burada eşitliğin doğasına aykırı bir şekilde mantıksal bir tutarsızlık söz konusudur. Aynı denklem aynı anda iki farklı sonuca eşit olamaz. Nitekim daha önce yapılan çözümlerde de söz konusu denklemlerin birbirlerine paralel doğrusal karaktere sahip oldukları, dolayısıyla ortak bir çözüm noktasına asla sahip olamayacakları ifade edilmişti.

### (c) Sayısal hassasiyet

Denklem takımlarında ifade edilen katsayılar ve eşitliklerin sağ tarafındaki sabit sayılar, çözümde rol oynayan sayılardır. Bazı hallerde sayısal değerlerdeki küçük bir farklılık, çözüm kümesini tamamen değiştirebilir. Örneğin müteakip denklem takımını dikkate alalım:

$$\begin{aligned} 0.835x_1 + 0.667x_2 &= 0.168 \\ 0.333x_1 + 0.266x_2 &= 0.068 \end{aligned}$$

Bu denklem takımının çözümü sonucunda;

$$x_1 = + 668.0, \quad x_2 = - 836.0$$

değerleri elde edilebilir. Aynı denklem takımının ikinci denklemine ait  $a_{22}$  değeri 0.266 yerine 0.265 alınırsa bu takdirde çözüm:

$$x_1 = + 1.0, \quad x_2 = - 1.0$$





olarak hesaplanabilir. Görüldüğü üzere sol taraftaki bir sayıya ait oldukça küçük bir değişiklik, çözüm kümesini aşırı derecede değiştirmektedir. Böyle bir hassasiyete sahip denklem takımları genellikle kötü-şartlı denklem takımı olarak adlandırılır. Sayısal hassasiyetin olmadığı ya da sayısal değerlerdeki küçük değişikliklerin sonucu fazla etkilemediği denklem takımları ise iyi-şartlı denklem takımı olarak adlandırılır. Tutarlı ve doğrusal bağımsızlığa sahip bir doğrusal denklem takımının çözümü sadedinde pek çok yöntemden bahsedilebilir. Ancak bu yöntemlere girmeden önce matris kavramı ve işlemleri üzerinde durmak faydalı olacaktır.

## (2) kare-matris

Kare-matris<sup>iii</sup>, satır ve sütun sayısının eşit olduğu 2-dizimli sayı grubudur. Bu meydana kare-matrisini;

$$a_{i,j} \quad \begin{matrix} i=1,2,\dots,n \\ j=1,2,\dots,n \end{matrix} \quad (1.46)$$

şeklinde ifade edebiliriz.

## (3) 0-matrisi

0-matrisi<sup>iv</sup>, dizine ait tüm elemanların sıfır olduğu matristir. Dolayısıyla 0-matrisi;

$$a_{i,j} = 0 \quad \begin{matrix} i=1,2,\dots,n \\ j=1,2,\dots,m \end{matrix} \quad (1.47)$$

biçiminde ifade edilebilir.

```
>> zeros(2,2)
ans* =
0     0
0     0
```



**MATLAB uygulaması**

\* “ans” terimi İngilizce’de “answer” kelimesinin kısaltması olup “cevap”, “çözüm” manalarında.

<sup>i</sup> Ill-conditioned

<sup>ii</sup> Well-conditioned

<sup>iii</sup> Square matrix

<sup>iv</sup> Zero/null matrix



#### (4) Köşegen elemanları

2-dizimli bir sayı grubunun köşegeni elemanları, satır ve sütun sayı numarasının aynı olduğu konumdaki matris elemanlarıdır. Köşegen elemanları;

$$a_{k,k}, \quad k = \min(n, m) \quad \begin{matrix} i=1,2,\dots,n \\ j=1,2,\dots,m \end{matrix} \quad (1.48)$$

şeklinde yazılabilir. Bu ifadeden de anlaşılacağı üzere matrisin kare matris olması şart değildir.

```
>> a=[1 2 3;4 5 6;7 8 9]
a =
     1     2     3
     4     5     6
     7     8     9

>> diag(a)
ans =
     1
     5
     9
```



**MATLAB uygulaması**

#### (5) I-matrisi ve birim yöney

I-matrisi, dizine ait köşegen elemanlarının 1, diğer elemanların sıfır olduğu kare-matristir. Söz konusu matris, birim matris<sup>ii</sup> olarak da adlandırılır. Birim matris;

$$I_n = \begin{bmatrix} a_{i,j} = 1 & i = j \\ a_{i,j} = 0 & i \neq j \end{bmatrix} \quad \begin{matrix} i=1,2,\dots,n \\ j=1,2,\dots,n \end{matrix} \quad (1.49)$$

biçiminde yazılabilir. Birim yöney, tüm elemanları 1 olan yöneydir;

$$e_n^T = [a_i = 1]^{i=1,2,\dots,n} \quad (1.50)$$

<sup>i</sup> Main diagonal  
<sup>ii</sup> Identity matrix





```

ans =
>> eye(3,3)
1     0     0
0     1     0
0     0     1
    
```



**MATLAB uygulaması**

### (6) Matrisin devriği

Herhangi bir  $A$  matrisinin devriğini;

$$b_{i,j} = a_{j,i} \quad \begin{matrix} i=1,2,\dots,n \\ j=1,2,\dots,m \end{matrix} \quad (1.51)$$

şeklinde tanımlamak mümkündür. Bu ifade kısaca:

$$B = A^T \quad (1.52)$$

biçiminde de yazılabilir. Burada  $A^T$  ifadesi,  $A$  matrisinin devriğini göstermektedir. İndissel manada satır ve sütun numaralarının yer değiştirilmesi ile elde edilen yeni matrisin anlamı, simetrik matris tanımlaması açısından önem arz eder.

```

A =
>> A=[-2 2 -3;-1 1 3;2 0 -1]
-2     2    -3
-1     1     3
 2     0    -1

ans =
>> A'
-2    -1     2
 2     1     0
-3     3    -1
    
```



**MATLAB uygulaması**





## (7) Matriste simetri

Tanımlı bir  $A$  matrisi ile onun devriği niteliğindeki  $A^T$  matrisi arasında müteakip eşitlik söz konusu ise:

$$a_{ij} = a_{ji} \quad \begin{matrix} i=1,2,\dots,n \\ j=1,2,\dots,m \end{matrix} \quad (1.53)$$

$$A = A^T$$

bahse konu  $A$  matrisine, bakışık veya simetrik matris adı verilir. Benzer şekilde matrisin kendisi ile devriği arasında;

$$A = -A^T \quad (1.54)$$

ilişkisi söz konusu olduğunda, bu takdirde  $A$  matrisine aykırı simetrik<sup>i</sup> matris adı verilir.

Tanımlı  $A$  matrisi;

$$A = \begin{bmatrix} 1 & 2 & 0 \\ 2 & -1 & 2 \\ 0 & 2 & 1 \end{bmatrix}$$

simetrik bir matris olup burada:

$$A = A^T$$

yazılabilir.



Örnek uygulama

## (8) Matrisin belirteci

Bir matrisin belirteci, o matris için önemli bir özellik olup etkisini, lineer denklem takımının çözümünün varlığı veya yokluğu ile ilgili değerlendirmelerde görmek mümkündür. Belirteç ile ilgili diğer bir husus, özdeğer ve özyöney kavramları olup ilerleyen kısımlarda bahse

<sup>i</sup> Skew-symmetric matrix

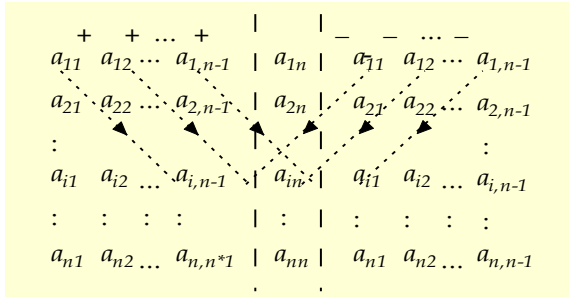


konu terimler detaylandırılmaktadır. Belirteç, sayısal bir büyüklük olup bir matris için sadece bir belirteç söz konusudur. Lineer denklem takımına ait katsayılar arasındaki ilişkiyi betimleyen belirtecin hesaplanmasında, değişik yöntemler ve gösterimler kullanılabilir. Bununla beraber belirteç hesaplamasında dikkate alınan en yaygın yaklaşım, Şekil 1.35'te görülmektedir. Buna göre  $n$  sütuna sahip katsayılar matrisinin son sütununun yanına  $1:n-1$  ve arası sütunlar ilave edilir. Daha sonra  $a_{11}$  elemanında başlayıp  $a_{1n}$  elemanını kapsayacak şekilde, soldan sağa köşegenler üzerindeki  $n$  adet dizin elemanı çarpılarak çarpım sonuçları toplanır. Aynı şekilde, ekleme yapılarak oluşturulan satırlardan  $a_{11}$  elemanı ile başlayıp  $a_{1,n-1}$  elemanını kapsayacak şekilde sağdan sola doğru köşegenler üzerindeki  $n$  adet dizin elemanı çarpılarak çarpım sonuçları toplanır. Daha sonra elde edilen ilk toplamdan ikinci toplam çıkarılarak matrisin belirteci hesaplanmış olur. Belirteç hesabının gösterimini, Denklem (1.55)'teki gibi simgeselleştirmek mümkündür;

$$\det(A) = |A| \quad (1.55)$$

$$|A| = \sum_{i=1:n} (-1)^{i+j} a_{ij} M_{ij}$$

Bu ifadede  $\det(A)$  veya  $|A|$  simgeleri  $A$  matrisinin belirtecini,  $i$  satır sayısını,  $j$  sütun sayısını,  $M_{ij}$   $i$ 'nci satır ve  $j$ 'nci sütun çıkarıldıktan sonra kalan  $(n-1) \times (n-1)$  matrisinin belirtecini temsil etmektedir. Görüldüğü üzere yaklaşımda  $j$  gibi herhangi bir sütun, ama o sütundaki tüm satır elemanları dikkate alınmaktadır. Ayrıca belirteç hesabında bir alt matrise ait başka bir belirteç kullanılmaktadır.



Şekil 1.35 Belirteç hesaplamasında dikkate alınan yöntemin temsili.

<sup>i</sup> Determinant







Örnek uygulama

Tanımlı  $A$  matrisinin belirteci iki farklı şekilde elde edilebilir;

$$A = \begin{bmatrix} 1 & 2 & 0 \\ 3 & -1 & 2 \\ 2 & 4 & -1 \end{bmatrix}$$

Buna göre ilk yöntemde:

$$\begin{array}{ccc|cc} 1 & 2 & 0 & : & 1 & 2 \\ 3 & -1 & 2 & : & 3 & -1 \\ 2 & 4 & -1 & : & 2 & 4 \end{array}$$

düzenlemesi yapıp, önce soldan sağa ve bilahare sağdan sola çarpımlar yapılarak toplanır, elde edilen değerler birbirlerinden çıkarılır:

$$\det(A) = \{(1 \times (-1) \times -1) + (2 \times 2 \times 2) + (0 \times 3 \times 4)\} \\ - \{(0 \times (-1) \times 2) + (1 \times 2 \times 4) + (2 \times 3 \times (-1))\}$$

$$|A| = 7$$

Aynı sonucu Denklem (1.55) ile de bulmak mümkündür. Buna göre  $j = 3$  sütunu dikkate alınarak:

$$\det(A) = \sum_{i=1:n} (-1)^{i+j} a_{ij} M_{ij}$$

$$\det(A) = (-1)^{1+3} a_{13} M_{13} + (-1)^{2+3} a_{23} M_{23} + (-1)^{3+3} a_{33} M_{33}$$

yazılabilir. Bu ifadeler sayısal olarak dikkate alınırsa;

$$\det(A) = (-1)^{1+3} \cdot 0 \cdot \begin{bmatrix} 3 & -1 \\ 2 & 4 \end{bmatrix} + (-1)^{2+3} \cdot 2 \cdot \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} + (-1)^{3+3} \cdot -1 \cdot \begin{bmatrix} 1 & 2 \\ 3 & -1 \end{bmatrix}$$

$$|A| = 7$$

sonucuna ulaşılır.





```
A =
1     2     0
3    -1     2
2     4    -1

>> A=[1 2 0;3 -1 2;2 4 -1]

>> det(A) ans = 7
```



MATLAB uygulaması

## (9) Matrisin normu

Daha önce yöneye ait şiddet kavramı üzerinde durulmuş ve yöneyin şiddetinin, onun büyüklüğünün bir göstergesi olduğu ifade edilmişti. Benzer bir yaklaşım matrisler için de tanımlanabilir. Bir matrisin şiddeti ya da diğer bir ifade ile normu, o matrisin satır ya da sütun sayısı büyüklüğünden ziyade matris elemanlarının büyüklüğünün bir göstergesidir. Değişik norm tanımlamaları mümkün olup genellikle kullanılan dört değişik normdan bahsedilebilir. Bunlardan ilki “Euclidean” norm veya diğer bir adıyla “Frobenius” norm tanımlamasıdır. Herhangi bir  $A$  matrisinin “Euclidean” normu:

$$\|A\|_E = \left\{ \sum_{i,j} (a_{ij})^2 \right\}^{1/2} \quad (1.56)$$

şeklinde tanımlanabilir. Burada  $p$  değeri genellikle 2 olarak seçilir. Dolayısıyla norm, matris elemanlarının karelerinin toplamının karekökü alınarak hesaplanabilir.

Norm-1 olarak adlandırılan diğer norm tipi;

$$\|A\|_1 = \max_j \sum_i |a_{ij}| \quad (1.57)$$

biçimindedir. Bu ifade, sütun elemanlarının mutlak değer toplamlarının azami olduğu değeri dikkate almaktadır. Benzer şekilde Norm- $\infty$  olarak adlandırılan üçüncü norm tipi;

$$\|A\|_\infty = \max_i \sum_j |a_{ij}| \quad (1.58)$$

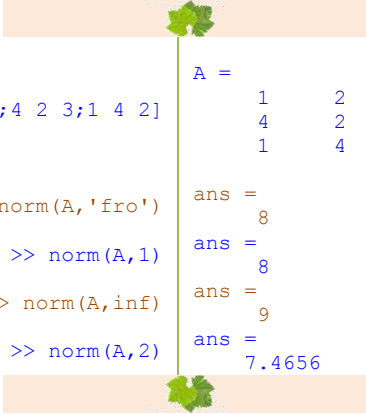
şeklinde dir.



Bu ifade, satır elemanlarının mutlak değer toplamalarının azami olduğu değeri dikkate almaktadır. Son olarak Norm-2 adıyla adlandırılan norm tipi;

$$\|A\|_2 = \sqrt{\lambda_{max}} \quad (1.59)$$

şeklinde tanımlanmaktadır. Burada  $\lambda_{max}$ , daha sonra detaylarını vereceğimiz özdeğer olup  $AA^T$  çarpımı sonucu elde edilen yeni matrisin sahip olduğu özdeğerler arasındaki azami değerdir.



```

>> A=[1 2 3;4 2 3;1 4 2]
A =
     1     2     3
     4     2     3
     1     4     2

>> norm(A,'fro')
ans =
     8

>> norm(A,1)
ans =
     8

>> norm(A,inf)
ans =
     9

>> norm(A,2)
ans =
    7.4656
    
```

MATLAB uygulaması

## (10) Matrislerde toplama ve çıkarma

Matrislerde toplama ve çıkarma işlemleri için farklı yaklaşımlar tanımlanabilir. Bununla beraber yaygın olarak kullanılan yaklaşımda toplama/çıkarma işlemine tabi tutulacak matrislerin aynı ebatta olması gerekir. Buna göre  $[A]_{m \times n}$  boyutlarındaki bir matris ancak  $[B]_{m \times n}$  boyutlarındaki bir matrisle toplama/çıkarma işlemlerine tabi tutulabilir. Söz konusu işlemler:

$$(a \pm b)_{ij} = a_{ij} \pm b_{ij} \quad (1.60)$$

şeklinde tariflenebilir. Böyle bir yaklaşımda dizine ait her bir eleman birebir işleme tabi tutulmaktadır.



```

A =
    1     2     0
    3    -1     2
    2     4    -1

B =
    4     1     3
   -3     0     2
    1     5    -2

ans =
    5     3     3
    0    -1     4
    3     9    -3

ans =
   -3     1    -3
    6    -1     0
    1    -1     1

```



**MATLAB uygulaması**

### (11) Matrislerde çarpma

Matrislerde çarpma işlemleri farklı yaklaşımlarla tanımlanabilir. En basit çarpım işlemi 0-dizin niteliğindeki bir sayı ile 2-dizimli sayı grubu niteliğindeki bir matrisin çarpım işlemi olup;

$$cA = (ca)_{ij} \quad (1.61)$$

$$(ca)_{ij} = ca_{ij}$$

biçiminde tariflenebilir. Burada  $c$ , sayılı ifade etmektedir.

```

A =
    1     2     0
    3    -1     2
    2     4    -1

```



**MATLAB uygulaması**





```

ans =
    2     4     0
    6    -2     4
    4     8    -2
  
```

Müteakip çarpım işlemi,  $n$  boyutlu 1-dizin niteliğindeki yöney ile  $n \times m$  boyutlu matrisin çarpım işlemi olup;

$$\mathbf{x}^T \mathbf{A} = (xa)_{ij} \quad (1.62)$$

$$(xa)_k = \sum_{j=1}^n x_j a_{kj}$$

biçiminde tariflenebilir. Görüldüğü üzere çarpım için öncelikle yöneyin devriğinin dikkate alınması gerekmekte ve çarpım sonucunda da yine bir yöney elde edilmektedir. Hatırlanacağı üzere herhangi bir doğrusal denklem takımı, bir  $A$  matrisi ve bilinmeyen değişkenleri içeren  $\mathbf{x}$  yöneyinin çarpımı ile elde edilmektedir.

```

A =
    1     2     0
    3    -1     2
    2     4    -1

x =
    2
    0
    3

x =
    2     0     3

ans =
    8    16    -3
  
```



MATLAB uygulaması

Çarpım sonucu 2-dizimli bir matrisin elde edilmesi içinse  $n \times m$  boyutlu bir matris ile  $m \times p$  boyutlu bir matrisin çarpımı gerekir. Bu işlem,



$$AB = (ab)_{ij}$$

$$(ab)_{ij} \Big|_{i=1:n}^{j=1:p} = \sum_{k=1}^m a_{ik} b_{kj}$$

(1.63)

şeklinde tariflenebilir. Burada elde edilen yeni matrisin boyutu  $n \times p$  ebadında olacaktır. Dikkat edileceği üzere yeni matrisin  $(ab)_{ij}$  konumundaki sayısının hesaplanmasında  $A$  matrisinin  $i$ 'ninci satır elemanları ile  $B$  matrisinin  $j$ 'ninci sütun elemanları çarpılarak toplanmaktadır.



Tanımlı  $A$  ve  $B$  matrislerinin çarpımı;

$$A = \begin{bmatrix} 1 & 2 \\ 0 & -1 \end{bmatrix}, \quad B = \begin{bmatrix} -1 & 3 \\ 1 & 2 \end{bmatrix}$$

$$AB = \begin{bmatrix} 1 \times (-1) + 2 \times 1 & 1 \times 3 + 2 \times 2 \\ 0 \times (-1) + (-1) \times 1 & 0 \times 3 + (-1) \times 2 \end{bmatrix}$$

$$AB = \begin{bmatrix} 1 & 7 \\ -1 & -2 \end{bmatrix}$$

şeklinde elde edilebilir. Burada görüldüğü üzere  $A$  matrisinin 1'inci satırı ile  $B$  matrisinin 1'inci sütunu, konum bazında çarpılarak toplanmıştır.



```
A =
     1     -2     3
     2      0     4
B =
     1     -2     3
     2      0     4
    -3      1     7

ans =
    -12     1    16
    -10     0    34

>> A*B
```



Örnek uygulama



MATLAB uygulaması





İki matrisin çarpımı ile ilgili diğer bir çarpım işlemi, Hadamard çarpımıdır. Bir önceki matris çarpımından farklı olarak Hadamard çarpımı:

$$(A \circ B)_{ij} \quad (1.64)$$

$$(ab)_{ij} \Big|_{i=1:n}^{j=1:m} = a_{ij}b_{ij}$$

işlemi ile tanımlanır. Söz konusu çarpımın gerçekleşebilmesi için her iki matrisin aynı boyutta olması gerekir.

```

A =
     1     -2     3
     2      0     4
>> A=[1 -2 3;2 0 4]
>> B=[-1 -2 0;1 3 6]
B =
    -1     -2     0
     1      3     6
ans =
    -1      4     0
     2      0    24
>> A.*B

```



**MATLAB uygulaması**

## (12) Matrislerde bölme

2-dizimli sayı gruplarında bölme işlemi, esasen matrisin tersi ve çarpma işlemi kurallarına göre yapılmaktadır. Buna göre  $A$  matrisi ile  $B$  matrisinin bölünmesi işlemi;

$$\begin{aligned} A/B &= AB^{-1} \\ AB^{-1} &= (a/b)_{ij} \end{aligned} \quad (1.65)$$

$$(a/b)_{ij} \Big|_{i=1:n}^{j=1:p} = \sum_{k=1}^m a_{ik}b_{kj}^{-1}$$

<sup>i</sup> Hadamard çarpımı bazen İngilizce'de "element-wise product" olarak geçer.





işlem adımları ile tarif etmek mümkündür. Burada  $B^{-1}$  ifadesi,  $B$  matrisinin tersi manasına gelmektedir. Bir matrisin tersinin olabilmesi için öncelikle:

$$BB^{-1} = I \quad (1.66)$$

ve

$$\det(B) \neq 0 \quad (1.67)$$

şartları sağlanmalıdır. Herhangi bir matrisin tersini farklı yöntemlerle elde etmek mümkündür. Bununla beraber matrisin tersi:

$$B^{-1} = \frac{1}{|B|} C^T \quad (1.68)$$

şeklinde tanımlanabilir. Burada  $C^T$  yine bir matris olup;

$$C = \begin{bmatrix} C_{11} & C_{21} & \cdots & C_{n1} \\ C_{12} & C_{22} & \cdots & C_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ C_{1n} & C_{2n} & \cdots & C_{nn} \end{bmatrix} \quad (1.69)$$

olarak açılabilir. Herhangi bir konumdaki  $C_{ij}$ ;

$$C_{ij} = (-1)^{i+j} M_{ij} \quad (1.70)$$

şeklinde tanımlı olup hatırlanacağı üzere  $M_{ij}$  ifadesi,  $i$ 'nci satır ve  $j$ 'nci sütun çıkarıldıktan sonra kalan  $(n-1) \times (n-1)$  matrisinin belirtecidir. Bir matrisin tersinin bulunmasının diğer bir yöntemi, matris satırları arasında bir takım işlemler yaparak bahse konu matrisin birim matrise dönüşümünü sağlamaktır. Bu dönüşüm esnasında aynı işlemler birim matrise de uygulanır. Tüm işlemler sonunda  $B$  matrisi birim matrise dönüşürken,  $I$  birim matrisi de  $B^{-1}$  matrisine dönüşmüş olur.  $B$  matrisinin tersinin alınması sonrasında  $A$  ve  $B^{-1}$  matrisleri matris çarpımı kurallarına göre çarpım işlemine tabi tutulur ve böylelikle bölme işlemi gerçekleştirilmiş olur.

---

<sup>i</sup> Inverse of matrix





Örnek uygulama

Tanımlı  $B$  matrisinin tersi, uygun satır işlemleri ile elde edilebilir:

$$B = \begin{bmatrix} 2 & 1 \\ 3 & 4 \end{bmatrix}$$

$$\begin{array}{l} \begin{bmatrix} 2 & 1 \\ 3 & 4 \end{bmatrix} \vdots \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\ + \begin{bmatrix} 2 & 1 \\ 3 & 4 \end{bmatrix} \vdots \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 5/4 & 0 \\ 3 & 4 \end{bmatrix} \vdots \begin{bmatrix} 1 & -1/4 \\ 0 & 1 \end{bmatrix} \\ -1/4 \times \begin{bmatrix} 2 & 1 \\ 3 & 4 \end{bmatrix} \vdots \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 0 \\ 3 & 4 \end{bmatrix} \vdots \begin{bmatrix} 4/5 & -1/5 \\ 0 & 1 \end{bmatrix} \\ 4/5 \times \begin{bmatrix} 5/4 & 0 \\ 3 & 4 \end{bmatrix} \vdots \begin{bmatrix} 1 & -1/4 \\ 0 & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 0 \\ 3 & 4 \end{bmatrix} \vdots \begin{bmatrix} 4/5 & -1/5 \\ 0 & 1 \end{bmatrix} \\ -3 \times \begin{bmatrix} 1 & 0 \\ 3 & 4 \end{bmatrix} \vdots \begin{bmatrix} 4/5 & -1/5 \\ 0 & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix} \vdots \begin{bmatrix} 4/5 & -1/5 \\ -12/5 & 8/5 \end{bmatrix} \\ + \begin{bmatrix} 1 & 0 \\ 3 & 4 \end{bmatrix} \vdots \begin{bmatrix} 4/5 & -1/5 \\ 0 & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix} \vdots \begin{bmatrix} 4/5 & -1/5 \\ -12/5 & 8/5 \end{bmatrix} \\ 1/4 \times \begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix} \vdots \begin{bmatrix} 4/5 & -1/5 \\ -12/5 & 8/5 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \vdots \begin{bmatrix} 4/5 & -1/5 \\ -3/5 & 2/5 \end{bmatrix} \end{array}$$

$$B^{-1} = \begin{bmatrix} 0.8 & -0.2 \\ -0.6 & 0.4 \end{bmatrix}$$



MATLAB uygulaması

```
>> B=[2 1;3 4]
B =
     2     1
     3     4

>> inv(B)
ans =
    0.8000   -0.2000
   -0.6000    0.4000

>> A=[1 -3;5 2]
A =
     1    -3
     5     2

>> A/B
ans =
    2.6000   -1.4000
    2.8000   -0.2000
```





```

A =
    1  -2  3
    2   0  4
>> A=[1 -2 3;2 0 4]
>> B=[-1 -2 0;1 3 6]
B =
   -1  -2   0
    1   3   6
ans =
   -1.0000    1.0000    Inf
    2.0000     0     0.6667
>> A./B

```

Daha önce iki matrisin çarpımı ile ilgili olarak ifade edilen Hadamard çarpımının bir benzerini Hadamard bölümü şeklinde tanımlamak mümkündür. Buna göre söz konusu bölme işlemi:

$$(A \oslash B = (a/b)_{ij} \quad (1.71)$$

$$(a/b)_{ij} \Big|_{i=1:n}^{j=1:m} = a_{ij}/b_{ij}$$

şeklinde tanımlanır. Bölme işleminin gerçekleşebilmesi için her iki matrisin aynı boyutta olması gerekir.

### (13) Doğrusal denklem takımlarında çözüm hassasiyeti

Daha önce doğrusal denklem takımlarının çözümü hususuna değinilmiş ve sadece bir çözümün var olabilmesi için doğrusal bağımsızlık, tutarlılık ve iyi-şartlı olma özelliklerinin gerekliliği ifade edilmişti. İyi-şartlı olma özelliği, genellikle matrisin tersi ile ilişkilendirilir. Bu bağlamda herhangi bir  $A$  matrisine ait şartlılık katsayısı<sup>i</sup> olarak tanımlanan katsayı:

$$k(A) = \|A\| \|A^{-1}\| \quad (1.72)$$

$$k(A) \geq 1$$

denklemini ile ifade edilir. Burada dikkate alınan matrisin normu, herhangi bir norm olabilir. Ancak önemli olan, aynı tip norm tanımlamasının her iki terimde de kullanılmasıdır. Genellikle dikkate alınan norm, Norm- $\infty$  normudur. Şartlılık katsayısı, değer olarak dikkate alınmaktan ziyade merteye olarak önem arz eder. Düşük değere sahip şartlılık katsayısı, genellikle denklem

<sup>i</sup> Condition number



takımının hassaslığının da düşük olduğunu gösterir. Dolayısıyla matris, iyi-şartlı matris grubuna girer ve sayısal denklem çözümleri daha güvenilirdir.  $k(A)$  değeri arttıkça, elde edilen çözüm kümesinin güvenilirliği azalır ve matris kötü-şartlı matris olarak tanımlanır. Çözüm kümesinin doğruluk hassasiyetini  $k(A)$  değeri ile ilişkilendirmek mümkündür. Bu bağlamda  $\epsilon$ , donanıma ait hassasiyet değeri olmak üzere söz konusu çözüm kümesi doğruluk hassasiyeti:

$$k(A) \cdot \epsilon \quad (1.73)$$

ifadesi ile elde edilebilir.

Tanımlı  $A$  matrisinin;

$$A = \begin{bmatrix} 2 & 1 \\ 3 & 4 \end{bmatrix}$$

şartlılık katsayısı;

$$k(A) = 7$$

olarak hesaplanabilir. Bu değere göre matris oldukça iyi-şartlı bir matristir. 24 bitlik sayısal hassasiyete sahip bir bilgisayarın  $\epsilon$  değeri  $2^{-23}$  olup bu da  $0.119209 \times 10^{-6}$  değerine karşılık gelir. Dolayısıyla sayısal işlemler neticesinde elde edilecek çözüm kümesi en az;

$$k(A) \cdot \epsilon = 7 \times 0.119209 \times 10^{-6} = 0.83447 \times 10^{-6}$$

$10^{-6}$  mertebesine kadar doğruluk içerir. Aşağıda verilen  $A$  matrisinin

$$A = \begin{bmatrix} 0.835 & 0.667 \\ 0.333 & 0.266 \end{bmatrix}$$

şartlılık katsayısı;

$$k(A) = 1.7543 \times 10^6$$

olarak hesaplanabilir. Bu değere göre matris oldukça kötü-şartlı bir matristir. Dolayısıyla;

$$k(A) \cdot \epsilon = 1.7543 \times 10^6 \times 0.119209 \times 10^{-6} = 0.2091$$

sayısal işlemler neticesinde elde edilecek çözüm kümesi yaklaşık olarak  $10^{-1}$  mertebesine kadar doğruluk içerir. Bu ise sağlıklı bir sonuç değildir.

Örnek uygulama



MATLAB uygulaması

```
>> A=[0.835 0.667;0.333 0.266]
>> e = 1/(2^23)

A =
    0.8350    0.6670
    0.3330    0.2660
e =
    1.1921e-007

>> kA =norm(A,inf) * norm(inv(A),inf)
>> kA*e
ans =
    0.2091

>> kA =norm(A,1) * norm(inv(A),1)
>> 1/kA
ans =
    5.7002e-007

>> rcond(A)
ans =
    5.7002e-007
```

#### (14) Matrislerde dikeygenlik

Bir matrisin dikeygen matris<sup>i</sup> olması için öncelikle kare-matris olması ve ayrıca;

$$A^T = A^{-1} \quad (1.74)$$

şartını sağlaması gerekir. Böyle bir şart aynı zamanda;

$$AA^T = I \quad (1.75)$$

anlamına da gelir. Dikeygen matrisin belirteci daima +1 veya -1 olup +1 olması halinde döndürü<sup>ii</sup>, -1 olması halinde döndürü veya yansıma<sup>iii</sup> dönüşümü söz konusudur.

<sup>i</sup> Orthogonal matrix

<sup>ii</sup> Rotation

<sup>iii</sup> Reflection





Örnek uygulama

İki boyutlu bir ortamda döndürü dönüşümünü gerçekleştiren dikeygen matrise örnek olarak:

$$R = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$$

verilebilir. Burada tanımlanacak herhangi bir  $\theta$  açı değeri için  $R$  matrisi dikeygen matris özelliğine sahiptir. Örneğin  $\theta = 30$  derecelik bir dönü eldesi için:

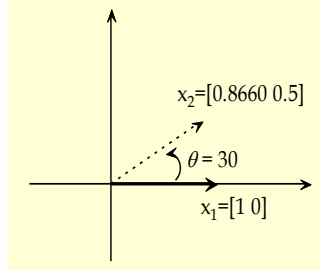
$$R = \begin{bmatrix} 0.8660 & -0.5 \\ 0.5 & 0.8660 \end{bmatrix}$$

dikeygen matrisi yazılabilir. Böyle bir matrisin, örneğin  $x_1$  gibi bir yöney ile çarpımı sonucu elde edilen  $x_2$  yeni yöneyi, önceki yöneyin konumuna göre saatin tersi yönünde  $\theta = 30$  açısız dönüye sahip olacaktır.

$$x_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$x_2 = R x_1 = \begin{bmatrix} 0.8660 & -0.5 \\ 0.5 & 0.8660 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$x_2 = \begin{bmatrix} 0.8660 \\ 0.5 \end{bmatrix}$$



İki boyutlu bir ortamda yansıma dönüşümünü gerçekleştiren dikeygen matrise örnek olarak;

$$R = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ \sin(\theta) & -\cos(\theta) \end{bmatrix}$$

matrisi verilebilir. Burada tanımlanacak herhangi bir  $\theta$  açı değeri için  $R$  matrisi, dikeygen matris özelliğine sahiptir.





```

>> teta=30; R =
>>R=[cosd(teta) -sind(teta); 0.8660 -0.5000
      sind(teta) cosd(teta)] 0.5000 0.8660

>> det(R) ans =
1

>> R*R' ans =
1 0
0 1

>> x=[1;0] x =
1
0

>> R*x ans =
0.8660
0.5000
    
```



MATLAB uygulaması

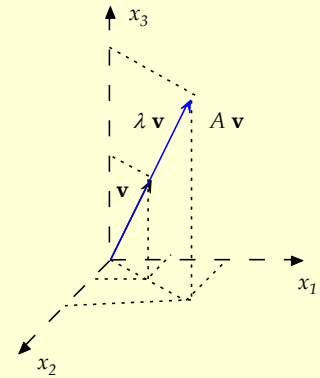
### (15) Matrislerde özdeğer ve özyöney

2-dizimli bir sayı grubu ile 1-dizimli bir sayı grubunun matrisel çarpımı sonucu elde edilen dizinin, yine 1-dizimli bir sayı grubu olduğu daha önce ifade edilmişti. Esasen böyle bir işlemin, doğrusal bir dönüşüm olduğu söylenebilir. Zira yapılan çarpımlar doğrusal çarpım olup çarpılan yöneyden, işlem sonucunda başka bir yöney elde edilmektedir. Özdeğer ve özyöney kavramları da esas itibarı ile doğrusal dönüşüm matrisleriyle ilgilidir. Herhangi bir  $v$  yöneyinin;

$$Av = \lambda v$$

(1.76)

eşitliğini sağlaması durumunda  $v$  yöneyine  $A$  matrisinin özyönevi, sayı niteliğindeki  $\lambda$  değerine ise  $A$  matrisinin özdeğeri denir. Burada  $A$  matrisinin kare-matris olması gerekir. Matrisin yöney ile çarpımı sonucunda aynı yönde ama farklı şiddette başka bir yöney elde edilmektedir. Yazılan ifade tekrar düzenlenir ve  $v$  ortak çarpanına alınırsa;



Şekil 1.36  $Av = \lambda v$  işlemi sonucunda  $v$  yöneyine paralel başka bir yöney elde edilir.







$$\begin{aligned} A\mathbf{v} - \lambda\mathbf{v} &= 0 \\ (A - \lambda I)\mathbf{v} &= 0 \end{aligned} \quad (1.77)$$

elde edilir. Çarpım sonucunda sıfır değerinin elde edilmesi için iki ihtimal söz konusu olup ilki aşikârî çözümdür ve  $\mathbf{v} = 0$  halidir. Sıfır-dışı<sup>ii</sup> çözüm için gerekli olan şart, çarpan durumunda olan ve esasen başka bir matris niteliğindeki  $M = A - \lambda I$  matrisinin belirtecinin sıfıra eşit olmasıdır. Bu durumda:

$$\det(M) = 0 \quad (1.78)$$

yazılabilir. Yazılan bu ifade önemli bir eşitlik olup öncelikle özdeğerlerin bulunmasında kullanılan yaklaşımın temelini oluşturur. Özdeğerlerin bulunması sonrasında ise özyöneyle elde edilebilir.



Örnek uygulama

Tanımlı  $A$  matrisinin özdeğerleri, müteakip cebirsel işlemlerle elde edilebilir:

$$A = \begin{bmatrix} 2 & 1 \\ 3 & 4 \end{bmatrix}$$

$$M = A - \lambda I = \begin{bmatrix} 2 & 1 \\ 3 & 4 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 2 - \lambda & 1 \\ 3 & 4 - \lambda \end{bmatrix}$$

$$\det(M) = (2 - \lambda)(4 - \lambda) - 3$$

$$\lambda^2 - 6\lambda + 5 = 0$$

2'nci derece denklem niteliğindeki denklemin kökleri;

$$\begin{aligned} \lambda_1 &= 1 \\ \lambda_2 &= 5 \end{aligned}$$

olup bu değerler aynı zamanda  $A$  matrisinin özdeğerleridir. Bulunan her bir özdeğer için bir özyöneyle elde edilebilir. Şöyle ki:

<sup>i</sup> Trivial solution

<sup>ii</sup> Non-zero solution



$$A\mathbf{v}_1 = \lambda_1\mathbf{v}_1$$

$$A\mathbf{v}_2 = \lambda_2\mathbf{v}_2$$

ifadelerinden iki ayrı doğrusal denklem takımı oluşturulabilir. Örneğin ilk ifadeden;

$$\begin{bmatrix} 2 & 1 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \lambda_1 \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$2x_1 + x_2 = x_1; x_1 + x_2 = 0$$

$$3x_1 + 4x_2 = x_2; 3x_1 + 3x_2 = 0$$

sonucuna ulaşmak mümkün olup birbirinden bağımsız olmayan bu iki denklem takımını sağlayan her yöney  $A$  matrisinin öz yöneyi olacaktır. Örneğin;

$$\mathbf{v}_1 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

yöneyi söz konusu denklemi sağlamakta olup bir öz yöney niteliğindedir. Benzer yaklaşım  $\lambda_2$  özdeğeri dikkate alınarak  $\mathbf{v}_2$  için de sergilenebilir.

```
>> A=[1 2 0;3 2 4;1 5 2]
A =
     1     2     0
     3     2     4
     1     5     2

V =
   -0.2207   -0.8038    0.3457
   -0.6767    0.0324   -0.7002
   -0.7024    0.5941    0.6247

>> [V D]=eig(A)
D =
    7.1309         0         0
         0    0.9195         0
         0         0   -3.0504
```



**MATLAB uygulaması**

Burada V, her bir sütunda öz yöneyi; D, o öz yöneye ait özdeğeri ifade etmektedir.



Daha önce özdeğerlerin bulunması bağlamında ifade edilen;

$$\det(A - \lambda I) = 0 \quad (1.79)$$

eşitliği beraberinde

$$\det(A) = \prod_{i=1}^n \lambda_i \quad (1.80)$$

denklemini de sağlamaktadır. Dolayısıyla özdeğerlerinin çarpımı,  $A$  matrisinin belirtecinin hesaplanmasında dikkate alınabilmektedir.

#### d. İkilenik form

Matematiksel manada form kavramı daha çok matematiksel gösterim anlamında kullanılır. Aynı matematiksel sonucu farklı formlarda göstermek mümkündür. Bu bağlamda ikilenik<sup>i</sup> form ile simetrik ikili-doğrusal<sup>ii</sup> form arasındaki ilişki örnek olarak verilebilir. Böyle bir ilişkilendirmede ikilenik form, bir fonksiyon tanımlaması olup ikili-doğrusal form ise yöneylerin doğrusal ilişkisini içeren tanımlamadır. Her iki form esasen yöney uzayı ile sayıl uzayı arasındaki bağıntıyı betimler. Buna göre yöney uzayı  $\mathbf{V}$  ve sayıl uzayı  $F$  olmak üzere ikili-doğrusal formu:

$$\mathbf{V} \times \mathbf{V} \rightarrow F \quad (1.81)$$

şeklinde eşleştirebiliriz. Görüldüğü üzere burada iki yöneyin çarpımı söz konusu olup sonuçta bir sayıl elde edilmektedir. Daha fazla sayıda yöneyin çarpımı söz konusu olduğunda form çoklu-doğrusal<sup>iii</sup> form niteliği ( $\mathbf{V}^n \rightarrow F$ ) kazanır. İkili-doğrusal form tanımlaması alâde bir eşleşme olmayıp, böyle bir eşleşmenin geçerli olabilmesi için formun:

$$\begin{aligned} b_f(\mathbf{u} + \mathbf{v}, \mathbf{w}) &= b_f(\mathbf{u}, \mathbf{w}) + b_f(\mathbf{v}, \mathbf{w}) \\ b_f(\mathbf{u}, \mathbf{v} + \mathbf{w}) &= b_f(\mathbf{u}, \mathbf{v}) + b_f(\mathbf{u}, \mathbf{w}) \\ b_f(\lambda \mathbf{u}, \mathbf{v}) &= k b_f(\mathbf{u}, \mathbf{v}) \\ b_f(\mathbf{u}, \lambda \mathbf{v}) &= k b_f(\mathbf{u}, \mathbf{v}) \end{aligned} \quad (1.82)$$

şartlarını sağlaması gerekir. Burada  $\mathbf{u}, \mathbf{v}$  ve  $\mathbf{w}$  herhangi üç yöneyi,  $k$  katsayı niteliğindeki bir sayılı ifade etmektedir. Formun bakışık ikili-doğrusal form olması için ilave bir şart dikkate alınır:

<sup>i</sup> Quadratic form

<sup>ii</sup> Bilinear form

<sup>iii</sup> Multi-linear form





$$b_f(\mathbf{u}, \mathbf{v}) = b_f(\mathbf{v}, \mathbf{u}) \quad (1.83)$$

İkilenik form, ifade edildiği üzere bir fonksiyon olup söz konusu fonksiyonun;

$$q(kv) = k^2 q(v) \quad (1.84)$$

$$\frac{1}{2}\{q(u+v) - q(u) - q(v)\} = b_f(u, v)$$

şartlarını sağlaması gerekir. İkilenik form ile bakışık ikili-doğrusal form arasındaki temel ilişki, şartlardan ikincisinde görülmektedir. İkili-doğrusal forma en iyi örnek:

$$b_f(\mathbf{x}, \mathbf{x}) = \mathbf{x}^T A \mathbf{x} \quad (1.85)$$

şeklindeki doğrusal ilişkidir. Burada  $\mathbf{x}$  terimi,  $x_1, x_2, \dots, x_n$  gibi  $n$  boyutlu bağımsız değişkenleri temsil eden yöneyi;  $A$ ,  $n \times n$  ebatlarındaki bir matrisi ifade etmektedir.  $A$  matrisinin aynı zamanda bakışık matris olduğu da kabul edilebilir. Bahse konu formun ikilenik form karşılığı;

$$q(\mathbf{x}) = \sum_{i,j=1}^n a_{ij} x_i x_j \quad (1.86)$$

biçiminde belirtmek mümkündür. Tabiatıyla burada  $a_{ij}$  katsayıları  $A$  matrisinin sayılarıdır. Daha anlaşılır olması açısından her iki formu beraber ifade edersek:

$$q(\mathbf{x}) = b_f(\mathbf{x}, \mathbf{x}) \quad (1.87)$$

$$\sum_{i,j=1}^n a_{ij} x_i x_j = [x_1 \quad x_2 \quad \dots \quad x_n] \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

şeklindeki açılımı yapmak mümkündür.



İkilenik form dâhilinde dikkate alınan  $A$  matrisi simetrik matris olarak dikkate alınabilir. Zira herhangi bir  $B$  matrisi uygun düzenleme ile simetrik matris şeklinde ifade edilebilir. Bu bağlamda:



Detay açıklama





$$a_{jk} = \frac{1}{2}(b_{jk} + b_{kj})$$

ifadesini yazmak mümkündür. Örneğin  $B$  matrisi aşağıdaki gibi bir matris olsun:

$$B = \begin{bmatrix} 3 & 4 \\ 6 & 1 \end{bmatrix}$$

Bu durumda ikilenik fonksiyon:

$$q(\mathbf{x}) = \mathbf{x}^T B \mathbf{x} = [x_1 \quad x_2] \begin{bmatrix} 3 & 4 \\ 6 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 3x_1^2 + 10x_1x_2 + x_2^2$$

şeklinde elde edilebilir. Söz konusu  $B$  matrisi yerine  $a_{jk}$  yaklaşımı dikkate alınarak  $A$  matrisi yazıldığında:

$$A = \begin{bmatrix} 3 & 5 \\ 5 & 1 \end{bmatrix}$$

$$q(\mathbf{x}) = \mathbf{x}^T A \mathbf{x} = [x_1 \quad x_2] \begin{bmatrix} 3 & 5 \\ 5 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 3x_1^2 + 10x_1x_2 + x_2^2$$

elde edilebilir ki bu ifade daha önce elde edilen ikilenik fonksiyon ile aynıdır. Dolayısıyla ikilenik form dâhilinde dikkate alınan matris simetrik matris olarak kabul edilebilir.



İkili-doğrusal form dâhilindeki matrisin bakışık ya da diğer bir ifade ile simetrik matris olması, matrise ait özdeğerlerin ve dolayısıyla bağımsız değişkenlerin gerçek sayı olmasını sağlar. Diğer taraftan tahmin edileceği üzere  $q(\mathbf{x})$ , 2'nci dereceden homojen çokterimli<sup>i</sup> bir fonksiyon niteliğindedir. Homojen çokterimli fonksiyonlarda katsayısı sıfırdan farklı olan tüm bağımsız değişkenli terimlerin üssel toplamı aynı olup ikilenik formlarda bu sayı ikidir. Dolayısıyla ikilenik form, bu yönüyle ikilenik fonksiyondan ayrılmaktadır. Zira ikilenik fonksiyon<sup>ii</sup>, toplamda derecesi iki ve altındaki üssel ifadeleri içerebilmektedir. İkilenik formlar, doğrusal cebir ile doğrusal olmayan analiz arasında adeta bir köprü görevi görmektedir. Bu sayede doğrusal olmayan ilişkiler lineer işlemlerle tanımlanabilmekte ve çarpım dâhilinde kullanılan matrisin özelliğine göre ikilenik formun yapısal özellikleri yorumlanabilmektedir. Bu meyanda fonksiyonun dışbükey olup olmadığı ve dolayısıyla birden fazla uç noktaya sahip olup olmadığı

<sup>i</sup> Homogeneous polynomial

<sup>ii</sup> Quadratic function



da anlaşılabilir. İkilenik formun yapısal özelliği, ifade edildiği üzere çarpımda dikkate alınan matrisin özelliğine bağlıdır. Matrisin özelliğini, sahip olduğu özdeğerlerin durumu belirler. Buna göre  $q(\mathbf{x})$  ikilenik forma sahip ikilenik fonksiyon,  $A$  matris ve  $\lambda_i, i = 1, 2, \dots, n$  matrisin özdeğerleri olmak üzere:

$$\begin{aligned} \lambda_i > 0 &\rightarrow A \text{ pozitif tanımlıdır ve } q(\mathbf{x}) \text{ kesinlikle dışbükeydir}^i, \\ \lambda_i \geq 0 &\rightarrow A \text{ pozitif sanki-tanımlıdır ve } q(\mathbf{x}) \text{ dışbükeydir}^{ii}, \\ \lambda_i < 0 &\rightarrow A \text{ negatif tanımlıdır ve } q(\mathbf{x}) \text{ kesinlikle içbükeydir}^{iii}, \\ \lambda_i \leq 0 &\rightarrow A \text{ negatif sanki-tanımlıdır ve } q(\mathbf{x}) \text{ içbükeydir}^{iv}, \\ \lambda_i > 0, \lambda_j < 0 &\rightarrow A \text{ tanımsızdır}^v. \end{aligned}$$



Örnek uygulama

Tanımlı  $A$  matrisinin ikilenik form dâhilinde dikkate alınması halinde ikilenik fonksiyon:

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$q(\mathbf{x}) = \mathbf{x}^T A \mathbf{x}$$

$$q(\mathbf{x}) = [x_1 \quad x_2] \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$q(\mathbf{x}) = x_1^2 + x_2^2$$

şeklinde elde edilebilir. Formun matrisinin özdeğerleri ise;

$$\lambda_1 = 1, \quad \lambda_2 = 1$$

olarak bulunabilir. Görüldüğü üzere her iki özdeğer de pozitif olup bu durumda  $A$  matrisi pozitif tanımlı ve  $q(\mathbf{x})$  fonksiyonu da kesinlikle dışbükey bir fonksiyondur. Daha açık bir ifade ile  $q(\mathbf{x})$  fonksiyonu sadece bir tane minimum noktaya sahiptir.



<sup>i</sup> “Positive definite” ve “strictly convex”

<sup>ii</sup> “Positive semi-definite” ve “convex” olarak geçer.

<sup>iii</sup> “Negative definite” ve “strictly concave” olarak geçer.

<sup>iv</sup> “Negative semi-definite” ve “concave” olarak geçer.

<sup>v</sup> Indefinite

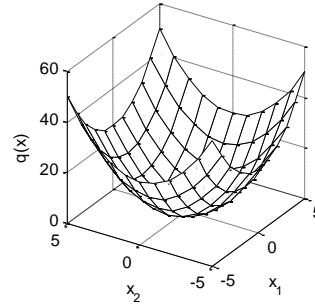




```
>> A=[1 0;0 1]
>> eig(A)
```

```
A =
     1     0
     0     1
ans =
     1
     1
```

```
>> x1=-5:01:5;
>> x2=-5:01:5;
>> [X1 X2]=meshgrid(x1,x2);
>> qX=X1.^2+X2.^2;
>> mesh(X1,X2,qX)
>> axis square
```



MATLAB uygulaması



Tanımlı  $A$  matrisinin ikilenik form dâhilinde dikkate alınması halinde ikilenik fonksiyon:

$$A = \begin{bmatrix} -3 & 0 \\ 0 & -3 \end{bmatrix}$$

$$q(\mathbf{x}) = \mathbf{x}^T A \mathbf{x} = \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} -3 & 0 \\ 0 & -3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = -3x_1^2 - 3x_2^2$$

şeklinde elde edilebilir. Formun matrisinin özdeğerleri;

$$\lambda_1 = -3, \quad \lambda_2 = -3$$

olarak bulunabilir. Görüldüğü üzere her iki özdeğer de negatif olup bu durumda  $A$  matrisi negatif tanımlı ve  $q(\mathbf{x})$  fonksiyonu da kesinlikle içbükey bir fonksiyondur. Daha açık bir ifade ile  $q(\mathbf{x})$  fonksiyonu belirli bir minimuma sahip olmayıp, sadece bir tane maksimum noktaya sahiptir.



Örnek uygulama





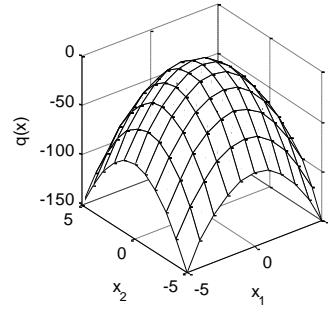


```
>> A=[-3 0;0 -3]
>> eig(A)
```

```
A =
    -3     0
     0    -3

ans =
    -3
    -3
```

```
>> x1=-5:01:5;
>> x2=-5:01:5;
>> [X1 X2]=meshgrid(x1,x2);
>> qX=X1.^2+X2.^2;
>> mesh(X1,X2,qX)
>> axis square
```



MATLAB uygulaması



Örnek uygulama

Tanımlı  $A$  matrisinin ikilenik form dâhilinde dikkate alınması halinde ikilenik fonksiyon:

$$A = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$q(x) = x^T A x = [x_1 \quad x_2] \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = -x_1^2 + x_2^2$$

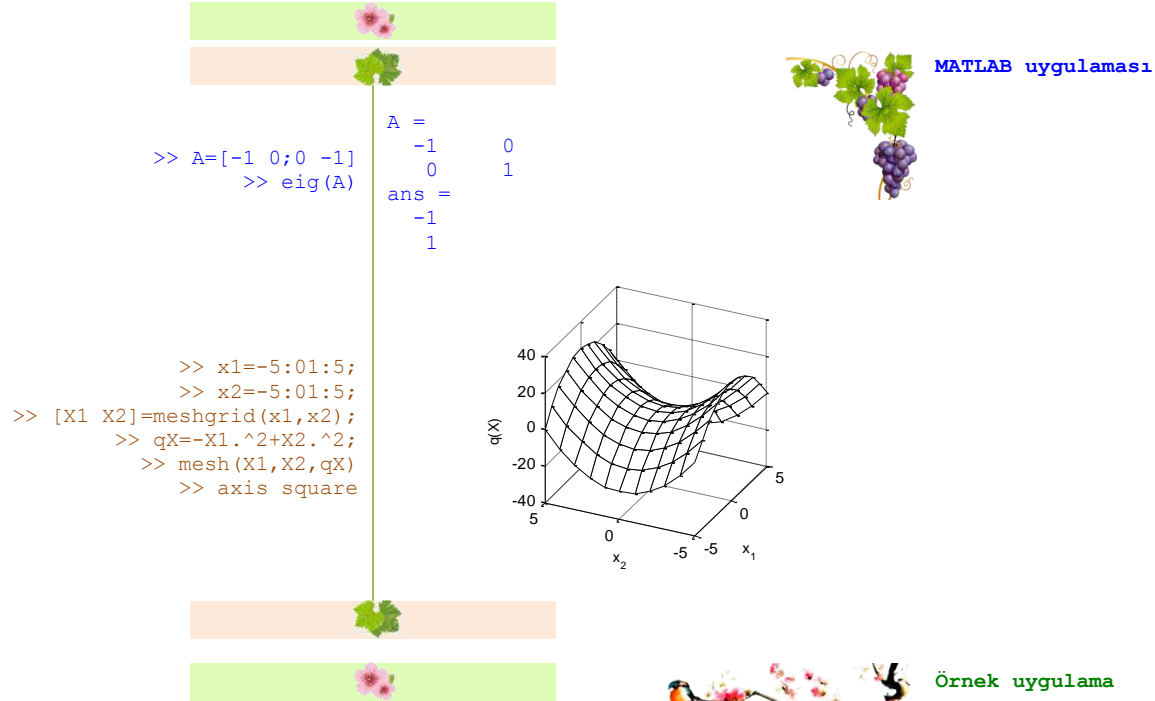
şeklinde elde edilebilir. Formun matrisinin özdeğerleri;

$$\lambda_1 = -1, \quad \lambda_2 = 1$$

olarak bulunabilir.



Görüldüğü üzere özdeğerlerden biri negatif diğeri ise pozitif olup bu durumda  $A$  matrisi tanımsızdır.  $q(x)$  fonksiyonu ne içbükey ve ne de dışbükey olup genellikle semer geometrisi elde edilir. Böyle bir geometride azamilerin aşarisi<sup>i</sup> söz konusudur.



Hatırlanacağı üzere geometri bahsinde eliptik geometri tanımlanmış ve içinde sonsuz sayıda noktayı barındıran  $n$  boyutlu ortamdaki bu kümeye ait noktalar;

$$C = \{x \mid (x - x_c)^T A^{-1} (x - x_c) \leq 1\}$$

$$x \in R^n, \quad A = A^T$$

şeklinde ifade edilmişti. Burada  $x_c$  eliptik eğrinin merkezini,  $A$  kümeye ait noktaların merkezden ne kadar uzakta konumlanabileceğini belirleyen matrisi ifade etmekteydi. Yine matrisin karakterine göre eliptik

<sup>i</sup> Saddle point



eğrinin her eksenindeki en uzak mesafesi farklılık arz ediyordu. Bu mesafeler esasen  $A$  matrisinin özdeğerlerine denk düşmekte ve söz konusu mesafeler;

$$\{\sqrt{\lambda_i}, \quad i = 1, 2, \dots, n\}$$

şeklinde ifade edilebilmekteydi. Örnek bağlamda  $A$  matrisinin ve tersinin;

$$A = \begin{bmatrix} 0.6 & -0.4 \\ -0.4 & 0.6 \end{bmatrix}$$

$$A^{-1} = \begin{bmatrix} 3 & 2 \\ 2 & 3 \end{bmatrix}$$

olduğunu ve merkez noktasının da (0,0) olarak seçildiğini dikkate alırsak, söz konusu  $C$  kümesini;

$$C = \left\{ x \mid x^T \begin{bmatrix} 3 & 2 \\ 2 & 3 \end{bmatrix} x \leq 1 \right\}$$

$$C = \{x \mid 3x_1^2 + 4x_1x_2 + 3x_2^2 \leq 1\}$$

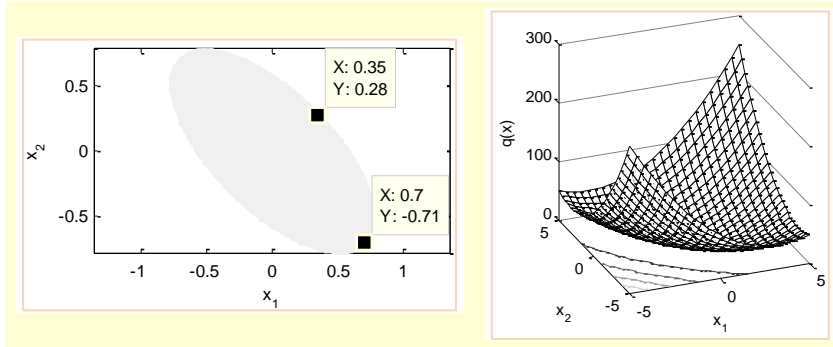
şeklinde ifade edebiliriz.  $A$  matrisinin özdeğerleri;

$$\lambda_1 = 0.2, \quad \lambda_2 = 1$$

olarak bulunabilir.

$$\sqrt{0.2} \cong \sqrt{0.35^2 + 0.28^2}$$

$$\sqrt{1} \cong \sqrt{0.70^2 + 0.71^2}$$



Buna göre elipsin uzun kenarı 1, kısa kenarı ise  $\sqrt{0.2}$  şeklindedir. Nitekim  $C$  kümesinin çizimi ve uç noktaların sayısal eldesi sonucu yapılan değerlendirme önermeyi doğrular niteliktedir. İkilenik fonksiyonun



çizimi bir önceki şekilde görülmektedir. Şekle göre  $q(\mathbf{x})$  yüzeyinde alınacak  $q(\mathbf{x}) \leq 1$  kesiti elipsi vermektedir.



### (e) Doğrusal denklem takımlarında çözüm yöntemleri

Doğrusal denklem takımlarının çözümü bağlamında değişik yöntemler söz konusu olup bu yöntemleri genellikle üç gruba ayırmak mümkündür; eleme yöntemleri, yinelemeli yöntemler ve diğer yöntemler. Eleme yöntemlerinde<sup>i</sup> temel yaklaşım, bağımsız değişken niteliğindeki her bir bilinmeyen  $x_i$  yöney elemanının sırasıyla denklemlerden elenmesi ve nihayetinde her bir denklemin sadece  $x_j = c_j$  gibi bir sayıl değerine indirgenerek çözüm kümesi elemanlarının bulunmasıdır. Eleme yöntemleri arasında Gauss eleme yöntemi veya yerine koyma yöntemleri sayılabilir. Yinelemeli yöntemler; tekrarlı ama aynı işlemleri öngören, başlangıçta  $x_j = rand_j$  gibi rassal olarak değerler atayan ve denklem takımının her bir denklemini sağlayana kadar işlemlere devam eden yöntemlerdir. Bu yöntemlere örnek olarak Jacobi ve Gauss-Seidel yöntemleri verilebilir. Diğer yöntemler başlığı altında ise pek çok yöntem sayılabilir. Bunlar arasında geçmişte eskilere dayanan Cramer kuralı ve matris tersine dayalı yöntem sayılabilir.

İfade edilen yöntemler dâhilinde yapılacak tercih, denklem takımının büyüklüğüne bağlıdır. Çok sayıda bağımsız değişken içeren denklem takımlarının çözümünde yinelemeli yöntemler tercih edilmektedir. Zira bu yöntemler daha az sayıda matematiksel işlem gerektirmekte, dolayısıyla daha kısa sürede sonuca ulaşmaktadırlar. Bahse konu yöntemlerin detaylarına burada girilmeyecek olup sadece matris tersine dayalı yöntem ifade edilecektir. Buna göre;

$$Ax = b \quad (1.88)$$

şeklinde ifade edilen lineer denklem takımının çözümünü;

$$x = A^{-1}b \quad (1.89)$$

işlemi ile elde etmek mümkündür.

---

<sup>i</sup> Elimination methods





```
>> A = [1 2; 4 -1]
A =
     1     2
     4    -1

>> b = [3;3]
b =
     3
     3

>> inv(A)*b
ans =
     1
     1
```



MATLAB uygulaması

### 1.2.3 Analiz

Analiz, değişimin incelenmesi üzerine kurulu matematiğin bir dalıdır. Diğer ana dallardan olan geometri ve cebir, analizin yeşermesine katkıda bulunan dallardır. Geometri şekillerin bilimi, cebir işlem ve ilişkilerin bilimi olarak adlandırılırsa, analiz de geometri ve cebir arasındaki ilişkileri irdeleyen bir daldır denebilir. Analizi dâhilinde değişimin belirlenmesi ve tanımlanması türev<sup>ii</sup> kavramını, niceliğin ve değişimlerin birikimli değerleri ise integral kavramını şekillendirmektedir. Türev ve integral evrik işlemlere<sup>iii</sup> örnek teşkil eder.

Değişim kavramı; neyin, neye göre, nerede ve nasıl değiştiği gibi soruları beraberinde getirir. Aynı zamanda toplamsal değer de nereden nereye veya diğer bir ifade ile başlangıç ve son nokta gibi sınır değerleri gün yüzüne çıkarır. Bu bağlamda öncelikli kavram fonksiyon olup onu sırasıyla türev ve integral takip etmektedir.

#### a. Fonksiyon

Doğada fiziksel etmenler sürekli birbiri ile ilişki içerisindedir. Bazı fiziksel etmenler girdi niteliğindeki bazı fiziksel etmenler de çıktı niteliğindedir. Herhangi bir eylem dâhilindeki fiziksel etmenler arasında var olan ilişkileri betimlemede kullandığımız kurguya, fiziksel model

<sup>i</sup> Analysis/calculus

<sup>ii</sup> Derivative

<sup>iii</sup> Inverse operations



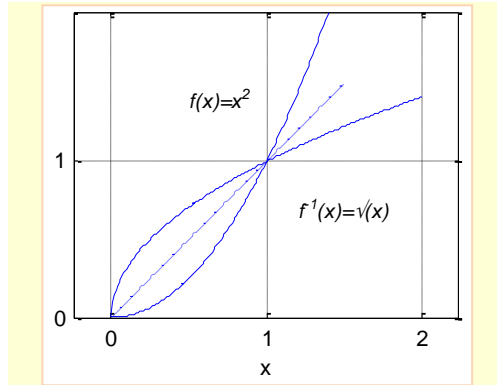
adı verilir. Modelin matematiksel çerçevede karşılığı, iki küme arasındaki ilişkiyi tanımlayan fonksiyon kavramıdır. Herhangi bir fonksiyonu;

$$f: A \rightarrow B \quad (1.90)$$

şeklinde tanımlayabiliriz. Burada  $A$  bir kümeyi,  $B$  ise diğer kümeyi temsil etmektedir. Dolayısıyla  $A$  kümesine ait  $\mathbf{R}^n$  türündeki bir eleman ile  $B$  kümesine ait  $\mathbf{R}^m$  türündeki bir başka eleman arasındaki ilişkiyi  $f$  fonksiyonu betimlemektedir. Diğer taraftan ilişki tek yönlü olmayıp, aynı eşleşme ters yönlü de olabilir. Bu durumda örtüşme:

$$f^{-1}: B \rightarrow A \quad (1.91)$$

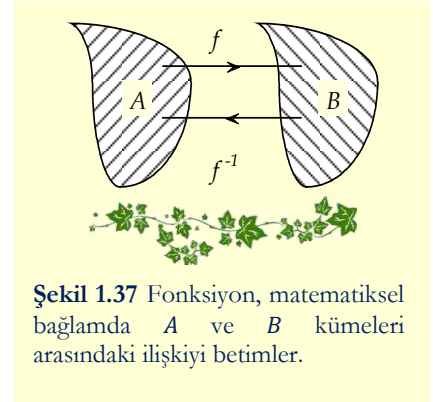
biçiminde ifade edilebilir. Girdi niteliğindeki fiziksel etmenler, bağımsız değişken olarak adlandırılır. Dolayısıyla çıktı niteliğindeki fiziksel olgu da bağımlı değişken niteliği kazanır. Ancak evrik işlemlerin olduğu bir fonksiyon tanımlamasında, çıktı niteliğindeki olgu girdi olarak tanımlanabileceğinden, çıktı veya girdi değişkenleri bağımlı veya bağımsız değişken olarak tanımlanabilir. Bu tanımlama eşleşmenin yönüne bağlıdır.



Şekil 1.38 Evrik işlemlerde temel yaklaşım, yapılan işlemin tersine çevrilmesidir.

## b. Limit

Fonksiyonlarda limit mefhumu, analiz disiplininin en temel kavramlarından biridir. Zira limit kavramı fonksiyonlarda sürekliliği, süreklilik ise fonksiyonun türevinin alınabilirliğini ya da



Şekil 1.37 Fonksiyon, matematiksel bağlamda  $A$  ve  $B$  kümeleri arasındaki ilişkiyi betimler.



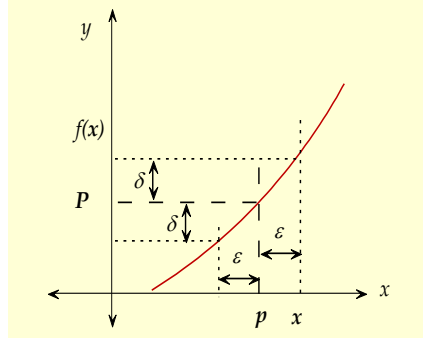
diğer bir ifade ile türetilirliğini sağlar. Türetilirlik, değışimin tanımlanması anlamına gelir. Limit kavramını tanımlamadan önce geometri bahsinde ifade edilen komşuluk kavramının hatırlanması faydalı olacaktır. Bilindiğı üzere komşuluk:

$$|x - p| < \varepsilon \quad (1.92)$$

denklemi ile ifade edilmiştir. Buna göre herhangi bir  $p$  noktasının  $\pm\varepsilon$  civarındaki noktaların bulunduğu bölgeye,  $p$  noktasının komşuluğı denmekteydi. Komşuluk kavramı, fonksiyon kavramı ile beraber düşünülduğünde denilebilir ki;  $p$  noktasına ait fonksiyon değeri  $P$  olmak üzere eğer komşuluk dâhilinde bulunan  $x$  noktası  $p$  noktasına yaklaştığında  $f(x)$  değeri de  $P$  değerine yaklaşıyorsa, fonksiyonun  $o$  noktadaki limiti  $P$  değeridir. Dolayısıyla her  $\varepsilon > 0$  değeri için mutlaka  $\delta > 0$  değeri vardır. Burada  $\delta = |f(x) - P|$  şeklinde tanımlanan bir gerçel sayıdır. Fonksiyonun  $x$  noktasındaki limiti simgesel olarak:

$$\lim_{x \rightarrow p} f(x) = P \quad (1.93)$$

şeklinde gösterilir.



**Şekil 1.39** Limit kavramı,  $p$  noktasının sağından ve solundan yaklaşıldığında fonksiyonun aynı  $P$  değerini almasını gerektirir.

Komşuluk kavramı,  $p$  noktasının ne tarafında olunduğı ile ilgili değildir. Dolayısıyla  $p$  noktasına ister sağından isterse de solundan yaklaşılsın, fonksiyon değerinin aynı  $P$  değerine yaklaşması gerekir. Yaklaşım yönüne bağılı olarak fonksiyon değerinin değışmesi halinde tek-yönlü limiti

<sup>i</sup> One-sided limit





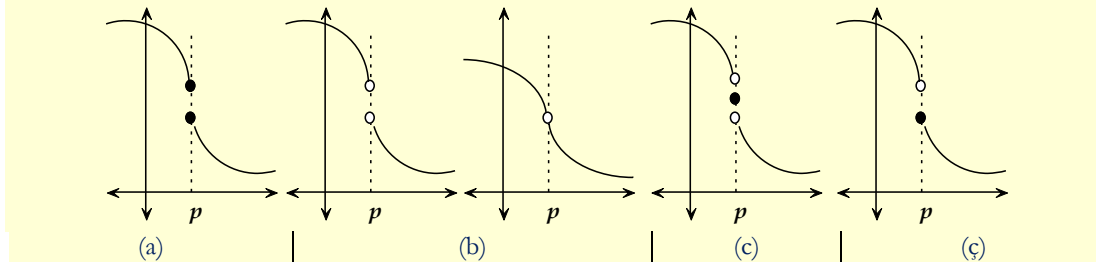
kavramı tanımlanır. Genellikle  $p$  noktasına o eksenin  $+\infty$  yönünden gelinerek yaklaşırsa limit değeri:

$$\lim_{x \rightarrow p^+} f(x) = P_+ \quad (1.94)$$

biçiminde, aynı  $p$  noktasına o eksenin  $-\infty$  yönünden gelinerek yaklaşırsa limit değeri:

$$\lim_{x \rightarrow p^-} f(x) = P_- \quad (1.95)$$

şeklinde ifade edilir. Dolayısıyla fonksiyonun  $p$  noktasında tek bir değeri söz konusu olmayıp, farklı seçenekler söz konusu olabilir. Örneğin  $p$  noktasında fonksiyon değeri olmayabilir. Bu durumda fonksiyon, o noktada tekillik arz etmektedir. Fonksiyonun o noktada değeri olabilir, ancak sağdan veya soldan yaklaşıldığında aynı fonksiyon değerine ulaşamayabilir. Diğer bir seçenekte ise fonksiyonun bir taraftan yaklaşıldığında değeri olmayabilir, ancak diğer taraftan yaklaşıldığında bir değeri olabilir. Bahsedilen tüm seçeneklerde limit kavramı tam bir değere yakınsamadığından dolayı süreklilik kavramı zedelenmektedir.



**Şekil 1.40** (a) Aynı  $p$  noktasına karşılık iki değer olması, (b) değer olmaması, (c) değer olması, ancak limit değerlerin olmaması, (ç) tek taraflı değer olması halleri.

### c. Süreklilik

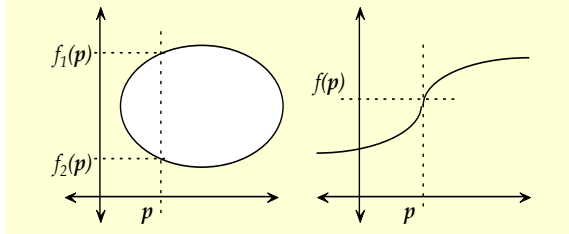
Süreklilik kavramı, limit kavramı üzerine kurulu bir kavram olup seri ya da diğer başka matematiksel kavramlarla da kurgulanabilir. Burada limit kavramı üzerine tanımı yapılacaktır. Özü itibarı ile süreklilik, bir fonksiyonun herhangi bir  $p$  noktasına karşılık limit değere sahip

<sup>i</sup> Tekillik ve tekil nokta kavramları İngilizce’de “singularity” ve “singular point” olarak geçer.





olmasıdır. Buna göre Şekil 1.40 dâhilinde belirtilen hiçbir fonksiyon sürekli fonksiyon<sup>i</sup> olarak tanımlanamaz.



Şekil 1.41 Sürekli fonksiyon örnekleri.

Herhangi bir fonksiyonun  $x$  noktasında sürekli olabilmesi için:

$$\lim_{x \rightarrow p} f(x) = f(p)$$

(1.96)

şartı sağlanmalıdır. Daha açık bir ifade ile  $p$  noktasına ister sağdan yaklaşalım isterse de soldan yaklaşalım, o noktada fonksiyonun alacağı belli bir değer olmalıdır. Ancak bu değer tek bir değer olması şart değildir. Süreklilik, aynı zamanda  $p$  noktasında meydana gelecek küçük bir değişikliğin  $f(p)$  değerinde de küçük bir değişiklik olacağı manasına gelir. Dolayısıyla fonksiyonun her bir  $x$  değerine karşılık belli bir değer alması ve bu fonksiyon değerinde darbesel ya da ani basamak<sup>ii</sup> şeklindeki değişikliklerin olmaması süreklilik için gerekli özelliklerdir.

#### ç. Değişim ve türevlenebilirlik

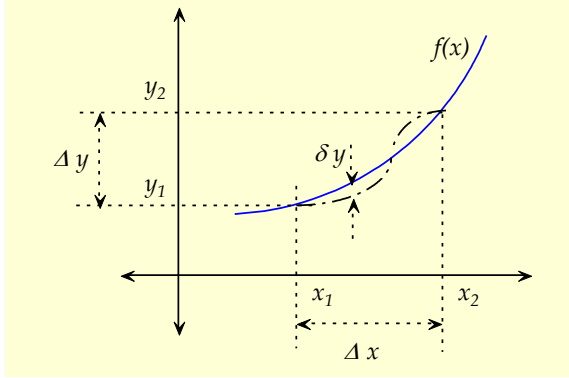
Değişim kavramı, matematiksel anlamda farklı seviyelerde tanımlanabilir. Bu bağlamda nokta bazlı değişim ( $\Delta$ ), diferansiyel değişim ( $d$ ), kısmi değişim ( $\partial$ ), yola bağlı değişim ( $\delta$ ), fonksiyonel değişim ( $\delta$ ) ve nihayetinde toplam değişim ( $D$ ) kavramlarından bahsetmek mümkündür. Tüm değişim kavramları, bağımsız ve ona bağlı olarak değişen bağımlı değişken kavramları üzerine temellendirilir. Bununla beraber daha önce de ifade edildiği üzere evrik işlemlerde, değişkenlerin bağımlılığı kümeler arası ilişkinin yönüne bağlı olarak değişebilir.

<sup>i</sup> Sürekli fonksiyon “Continuous function”, süreksiz fonksiyon ise “Discontinuous function” olarak geçer.

<sup>ii</sup> Darbesel değişime örnek olarak “Dirac delta” veya “Kronecker delta” fonksiyonu verilebilir.



Burada açıklaması yapılacak olan değişim kavramlarında bağımsız değişken olarak  $x$  ve ona bağlı olarak değişen bağımlı değişken ise  $y$  şeklinde simgelenmiştir.



Şekil 1.42 Nokta bazlı değişim ( $\Delta$ ) ile yola bağlı değişim ( $\delta$ ) kavramları.

Nokta bazlı değişim kavramını:

$$\begin{aligned}\Delta x &= x_2 - x_1 \\ \Delta y &= y_2 - y_1\end{aligned}\tag{1.97}$$

ifadeleri ile tanımlamak mümkündür. Dolayısıyla burada değişkenler arası bir ilişki ve değişimden ziyade, konumlar arası bir ilişki ve konuma göre değişim söz konusudur. Konuma göre değişim, bir yönüyle küme içerisinde sıralama anlamına da gelmektedir.

Diferansiyel değişim, bağımsız değişken niteliğindeki  $x$  ile bağımlı değişken niteliğindeki  $y$  arasındaki ilişkiye bağlı olarak kurgulanan “ne kadar değişim” sorusunun cevabını verir. Değişkenler arasındaki ilişki, “değişimin nasıl olduğu” sualine bağlıdır. Diğer bir deyişle “değişimin nasıl olduğu” sorusu,  $y$  değeri  $x$  değerine göre nasıl değişiyor sorusu anlamına gelir. Bu tanımlama türev kavramını doğurur. Türev kavramını limit kavramı ile tanımlamak mümkündür. Buna göre herhangi bir  $y = f(x)$  fonksiyonunun  $x$  noktasındaki türevi:

$$f'(x) = \lim_{\Delta x \rightarrow 0} \frac{f(x+\Delta x) - f(x)}{\Delta x}\tag{1.98}$$

denklemini ile tanımlanır.

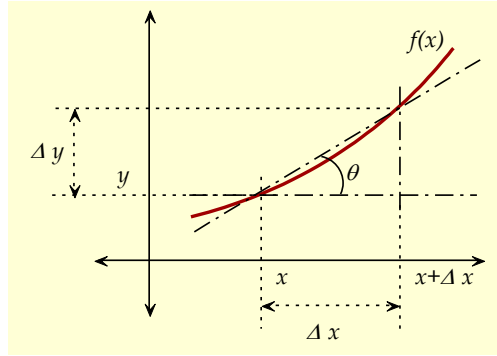




Denklemden geçen  $\Delta x$  değeri, sözel ifadesi ile sonsuz küçük değişimi göstermektedir. Buna karşılık fonksiyonda görülen değişikliğin,  $x$  değişkenindeki sonsuz küçük değişime oranı türevi vermektedir. Esasen söz konusu yaklaşımda değişimin bir anlamda doğrusallaştırılması fikri yatmaktadır. Fonksiyondaki diferansiyel değişim:

$$dy = f'(x) dx \quad (1.99)$$

ifadesi ile tanımlanır. Burada  $dx$  bağımsız nitelikteki  $x$  değişkeninin sonsuz küçük değişimini, bu adıma karşılık  $dy$  ise  $dx$  değişimine göre fonksiyonun ne kadar değiştiğini göstermektedir. Dolayısıyla fonksiyonun ne kadar değiştiği;  $[dy]$ , nasıl değiştiğine;  $[f'(x)]$  ve adım büyüklüğüne;  $[dx]$  bağlıdır.



**Şekil 1.43** Türev kavramının geometrik anlamı.

Türev kavramını, geometri bağlamında eğim olarak dikkate almak mümkündür. Herhangi bir  $x$  noktasının eğimini, o noktadan teğet geçen doğrunun eğimi ile ifade edebiliriz. Buna göre eğim:

$$\tan(\theta) = \frac{\Delta y}{\Delta x} \quad (1.100)$$

şeklinde yazılabilir. Böyle bir yaklaşım, genellikle sayısal analiz çalışmalarında dikkate alınmaktadır. Öte yandan fonksiyonun birden fazla bağımsız değişkeni söz konusu olduğunda diferansiyel değişim, her bir bağımsız değişkene göre meydana gelen değişimlerin toplamı olarak tanımlanır. Bu bağlamda fonksiyonun:



$$f(\mathbf{x}) = f(x_1, x_2, \dots, x_n) \quad (1.101)$$

şeklindeki  $n$  adet bağımsız değişkene bağlı olarak tanımlandığı düşünüldüğünde, sadece  $x_1$  değişkenine göre değişimin nasıl olduğu hususu;

$$\lim_{\Delta x_1 \rightarrow 0} \frac{f(x_1 + \Delta x_1, x_2, \dots, x_n) - f(x_1, x_2, \dots, x_n)}{\Delta x_1} \quad (1.102)$$

limit değeri ile tanımlanır. Tabiatıyla o noktada  $x_1$  değeri hariç diğer eksenlerdeki değerler sabit sayı şeklinde dikkate alınır. Zira buradaki amaç, kısmi olarak  $f(\mathbf{x})$  fonksiyon değerinin sadece  $x_1$  eksenindeki değişimini görmektir. Kısmi veya bağımlı değişim olarak adlandırılan bu değişim kısaca;

$$\frac{\partial f(x)}{\partial x_1} = \lim_{\Delta x_1 \rightarrow 0} \frac{f(x_1 + \Delta x_1, x_2, \dots, x_n) - f(x_1, x_2, \dots, x_n)}{\Delta x_1} \quad (1.103)$$

simgeleri ile gösterilir. Müteakip denklemde olduğu üzere, tüm bağımsız değişkenlerin kısmi değişimlerinin doğrusal bir ilişki bağlamında dikkate alınması sonucunda fonksiyonun diferansiyeli elde edilir;

$$df(x) = \frac{\partial f(x)}{\partial x_1} dx_1 + \frac{\partial f(x)}{\partial x_2} dx_2 + \dots + \frac{\partial f(x)}{\partial x_n} dx_n \quad (1.104)$$

Kısmi türev<sup>i</sup> olarak adlandırılan bağımlı değişim, bazen farklı gösterimlerle de sunulabilmektedir. Yaygın olarak kullanılan bir diğer gösterim de:

$$df = \partial_{x_1} f dx_1 + \partial_{x_2} f dx_2 + \dots + \partial_{x_n} f dx_n \quad (1.105)$$

şeklindeki gösterimdir. Diferansiyel değişim, her ne kadar sonsuz küçük adım sonrası değişim olarak dikkate alınsa da nihayetinde iki durum arası fark olarak tanımlanır. Dolayısıyla değişim, başlangıç ve son arasındaki değişime bağlıdır. Eğer başlangıç ve son nokta arasında fark yoksa değişimden de bahsedilemez. Bununla beraber bazı kavramlar, başlangıç ve son nokta arasındaki farktan ziyade başlangıç ve son nokta arasında izlenen yola bağlı olarak tanımlanabilir. Dolayısıyla fonksiyonun başlangıç ve son noktadaki değerinden ziyade izlediği yola göre aldığı

---

<sup>i</sup> Partial derivative



değerler önem kazanır. Bu durumda dikkate alınan değişim, yola bağlı değişim<sup>i</sup> olarak ifade edilir ve gösterimi de:

$$\delta F = f \cdot dr \quad (1.106)$$

şeklinde. Doğal olarak alınan yol ve bu yol alma esnasında  $f$  değerinin gösterdiği değişim, yola bağlı değişimi etkileyen hususlardır. Denklemdaki diğer bir husus ise  $dr$  ve  $f$  değerlerinin yöneysel büyüklük olmaları, aralarındaki çarpımın sayısal çarpım olmasıdır.



Yola bağlı değişim kavramına en iyi örnek, iş kavramıdır. Termodinamik biliminde göre bir enerji çeşidi olan iş;

$$\delta W = f \cdot dr \quad (1.107)$$

şeklinde tanımlanır. Burada  $f$  değeri uygulanan kuvveti,  $dr$  alınan yolu gösterir.



Yola bağlı değişim ile aynı simgenin kullanıldığı, ancak kavram olarak farklılık arz eden diğer bir değişim türü, fonksiyonel değişimdir. Fonksiyonel değişim<sup>ii</sup> kavramı, içerisinde fonksiyon barındıran bir fonksiyonlar fonksiyonunun, barındırdığı fonksiyondaki değişime bağlı olarak gösterdiği değişimi ifade eder. Dolayısıyla burada bağımsız değişkene göre değişim değil, bağımsız değişkenlerin bir kombinasyonu olan fonksiyona göre değişim söz konusudur. Fonksiyonel değişim:

$$\frac{\delta L}{\delta f} \quad (1.108)$$

ifadesi ile gösterilir. Burada  $L$  fonksiyonlar fonksiyonunu,  $f$  fonksiyonlar fonksiyonunun içinde barındırdığı fonksiyonu ifade etmektedir.

<sup>i</sup> Yola bağlı değişimin önem kazandığı diferansiyel değişim İngilizce’de “inexact differential” olarak geçer.

<sup>ii</sup> Fonksiyonel değişim İngilizce’de “functional derivative” olarak geçer.



Örnek uygulama





Örnek uygulama

Fonksiyonlar fonksiyonu kavramına verilebilecek örneklerden birisi “Lagrange” fonksiyonlar fonksiyonudur. Söz konusu fonksiyonlar fonksiyonu;

$$L[\mathbf{x}, f(\mathbf{x}), f'(\mathbf{x})]$$

şeklinde tanımlanır. Burada  $\mathbf{x}$  bağımsız değişkenleri içeren yöneyi,  $f(\mathbf{x})$  yöneysel fonksiyonu ve  $f'(\mathbf{x})$  ifadesi de kısmi türevleri içermektedir. “Lagrange” fonksiyonlar fonksiyonunun zaman ve konum değişimi için dikkate alınması halinde ifade:

$$L(t, \mathbf{x}, \mathbf{x}')$$

şeklini alır. Fiziksel manada zaman, konum ve zamana göre değişen konum ya da diğer bir ifade ile hız, “Lagrange” fonksiyonlar fonksiyonunu oluşturan öğelerdir.



Değişim bahsinde dikkate alınacak son kavram toplam değişimdir. Toplam değişimi kavramı, herhangi bir yöneysel veya sayısal özelliğin zaman içerisindeki değişimi odaklı bir kavramdır. Dikkat edileceği üzere burada zamana göre değişim ile zaman içerisindeki değişim kavramları birbirinden ayrılmaktadır. Zamana göre değişim, kısmi bir değişimi ifade etmektedir. Zaman içerisindeki değişim ise zaman boyutuna ilave olarak mekânsal değişimleri de dikkate alır. Dört boyutlu uzay-zaman sarmalında  $\varphi(\mathbf{x}, t)$  gibi herhangi bir sayısal özelliğin diferansiyel değişimini, konum ve zamana bağlı olarak:

$$d\varphi = \partial_{x_1}\varphi dx_1 + \partial_{x_2}\varphi dx_2 + \partial_{x_3}\varphi dx_3 + \partial_t\varphi dt \quad (1.109)$$

denklemi ile ifade etmek mümkündür. Bu ifadenin zaman içerisindeki değişimini tanımlamak için eşitliğin her iki tarafı zamanın diferansiyeline bölünebilir. Dolayısıyla:

$$\frac{d\varphi}{dt} = \partial_{x_1}\varphi \frac{dx_1}{dt} + \partial_{x_2}\varphi \frac{dx_2}{dt} + \partial_{x_3}\varphi \frac{dx_3}{dt} + \partial_t\varphi \frac{dt}{dt} \quad (1.110)$$

eşitliği elde edilir.

<sup>i</sup> Toplam değişim İngilizce’de “material derivative” veya “total derivative” olarak geçer.



Bu ifadedeki son terimi sadeleştirmek mümkündür. İlk üç terimdeki  $dx_i/dt$  bölümleri, zaman içerisindeki mekânsal değişimi göstermekte olup kinematik bilimindeki hız kavramını temsil etmektedir:

$$\mathbf{v} = \sum_{i=1}^3 \frac{dx_i}{dt} \mathbf{e}_i \quad (1.111)$$

Mekâna göre kısmi değişimleri içeren  $\partial_{x_i}\varphi$  terimleri nabla operatörü ile örtüştürülebilir. Zira nabla operatörü:

$$\nabla = \sum_{i=1}^3 \mathbf{e}_i \frac{\partial}{\partial x_i} \quad (1.112)$$

şeklindeki mekânsal değişimlerin toplamı olarak ifade edilir. Bu iki kavramın beraber dikkate alınması halinde zaman içerisindeki değişim miktarı:

$$\frac{d\varphi}{dt} = \partial_t \varphi + \mathbf{v} \cdot \nabla \varphi \quad (1.113)$$

denklemi ile toparlanmış olur. Dikkat edileceği üzere buradaki çarpım, yöneylerin sayısal çarpımıdır. Gösterimin klasik diferansiyel değişim kavramından ayrı tutulması amacıyla toplam değişimin simgesi:

$$\frac{DQ}{Dt} = \partial_t Q + \mathbf{v} \cdot \nabla Q \quad (1.114)$$

ifadesi ile genelleştirilebilir. Burada parantez içerisinde gelecek olan özellik, verilen örnekteki  $\varphi$  gibi bir sayı olabileceği gibi,  $\mathbf{p}$  simgeli bir yöney de olabilir. Bu durumda  $\mathbf{p}$  yöneyine ait toplam değişim:

$$\frac{D\mathbf{p}}{Dt} = \partial_t \mathbf{p} + \mathbf{v} \cdot \nabla \mathbf{p} \quad (1.115)$$

şeklinde olacaktır. Denklem ilk terimi zamana göre değişimi ifade ederken, ikinci terimi taşınım kaynaklı değişimi göstermektedir.





#### d. Temel bazı türev alma yöntemleri

Herhangi bir  $f(x)$  fonksiyonunun  $x$  noktasındaki türevini, limit kavramı ile elde etmek mümkündür;

$$f'(x) = \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x) - f(x)}{\Delta x} \quad (1.116)$$

Örneğin  $f(x) = x^2$  şeklindeki bir fonksiyonun türevi;

$$\begin{aligned} f'(x) &= \lim_{\Delta x \rightarrow 0} \frac{(x + \Delta x)^2 - x^2}{\Delta x} = \lim_{\Delta x \rightarrow 0} \frac{x^2 + 2x\Delta x + (\Delta x)^2 - x^2}{\Delta x} \\ f'(x) &= \lim_{\Delta x \rightarrow 0} \left\{ \frac{2x\Delta x}{\Delta x} + \frac{(\Delta x)^2}{\Delta x} \right\} = \lim_{\Delta x \rightarrow 0} \{2x + \Delta x\} \end{aligned} \quad (1.117)$$

işlemleri sonrasında,

$$f'(x) = 2x \quad (1.118)$$

olarak elde edilebilir. Benzer yaklaşımın diğer fonksiyonlar için de sergilenmesi halinde müteakip çizelgedeki gibi bazı türev alma kurallarını tanımlamak mümkündür.

Fonksiyon	Türev alma yöntemi	Fonksiyon	Türev alma yöntemi	Fonksiyon	Türev alma yöntemi
$c$	$0$	$c^{f(x)}$	$c^{f(x)} f'(x) \ln c$	$f(x)^n$	$nf(x)^{n-1} f'(x)$
$[cf(x)]'$	$cf'(x)$	$\ln x$	$\frac{1}{x}$	$c^x$	$c^x \ln c$
$[f(x) \pm g(x)]'$	$f'(x) \pm g'(x)$	$\ln f(x)$	$\frac{f'(x)}{f(x)}$	$\cot(x)$	$-\frac{1}{\sin^2(x)}$
$[f(x)g(x)]'$	$f'(x)g(x) + f(x)g'(x)$	$\sin(x)$	$\cos(x)$	$x^n$	$nx^{n-1}$
$\left[\frac{f(x)}{g(x)}\right]'$	$\frac{f'(x)g(x) - f(x)g'(x)}{[g(x)]^2}$	$\cos(x)$	$-\sin(x)$	$\tan(x)$	$\frac{1}{\cos^2(x)}$

Çizelge 1.2 Temel türev alma kuralları.





MATLAB uygulaması

```

>> syms x
>> f = 2*x^2-3*x+5
>> diff(f,x)

f =
2*x^2 - 3*x + 5
ans =
4*x - 3

>> syms x y
>> f=2*x*exp(-x^2-y^2)
>> diff(f,x)
>> diff(f,y)
>> diff(f,y,2)

ans =
2/exp(x^2 + y^2) - (4*x^2)/exp(x^2 + y^2)
ans =
-(4*x*y)/exp(x^2 + y^2)
ans =
(8*x*y^2)/exp(x^2 + y^2) - (4*x)/exp(x^2 + y^2)

>> syms a x
>> f = [cos(a*x), sin(a*x);
       -sin(a*x), cos(a*x)]
>> diff(f,x)

f =
[ cos(a*x), sin(a*x) ]
[ -sin(a*x), cos(a*x) ]
ans =
[ -a*sin(a*x), a*cos(a*x) ]
[ -a*cos(a*x), -a*sin(a*x) ]

```

### e. Taylor serisi

Literatürde Taylor serisi olarak geçen ve özellikle sayısal analiz çalışmalarında önemli bir yere sahip olan seri açılımı yaklaşımı, herhangi bir fonksiyonun herhangi bir noktadaki değerinin, polinomial denklemler vasıtasıyla tanımlanabileceği savına dayanır. Güç serisi<sup>i</sup> olarak da tanımlanabilen söz konusu yaklaşımın ilk örneklerini Hintli matematikçi Madhava'da (1340–1425) görmek mümkündür<sup>(37)</sup>. Seri açılımının türev kavramı ile bağlantısı, seriye ait katsayıların elde edilmesiyle ilgilidir. Herhangi bir  $f(x)$  fonksiyonunun  $x$  gibi herhangi bir noktadaki değerini;

$$f(x) = \sum_{n=0}^{\infty} b_n x^n \quad (1.119)$$

şeklindeki bir polinomial seri ile ifade etmek mümkündür. Burada  $b_n$  sabit katsayıyı ifade etmektedir. Görüldüğü üzere uygun değerlere sahip katsayılar kullanılarak,  $x$  değerinin üssel değerlerinden oluşan bir bileşimle,  $f(x)$  fonksiyonunun  $x$  noktasındaki değeri elde edilebilmektedir. Aynı fonksiyon değerini belli bir öteleme ile de ifade etmek mümkündür;

<sup>i</sup> Power series





$$f(x) = \sum_{n=0}^{\infty} a_n(x-c)^n \quad (1.120)$$

Burada  $a_n$  sabit katsayıyı,  $c$  ise başka bir sabit sayıyı temsil etmektedir.



2'nci dereceden örnek bir  $f(x)$  fonksiyonunu dikkate alalım;  
 $f(x) = 3x^2 + 2x + 1$ . Bu fonksiyonu, aynı dereceye sahip başka bir polinomiyal denklemle ifade edebiliriz. Örneğin  $x \rightarrow (x - 2)$  dönüşümü ve onun üssel değerleri üzerine kurulu olarak;

$$f(x) = A(x - 2)^2 + B(x - 2) + C$$

şeklindeki yaklaşım, aynı fonksiyonu ifade edebilir. Burada önemli olan  $A, B$  ve  $C$  katsayılarının uygun şekilde seçilmesidir. Nitekim müteakip bazı matematiksel işlemler yapılırsa;

$$\begin{aligned} 3x^2 + 2x + 1 &= A(x - 2)^2 + B(x - 2) + C \\ 3x^2 + 2x + 1 &= Ax^2 + (B - 2A)x + 4A - 2B + C \end{aligned}$$

eşitliği elde edilir. Burada aynı üssel ifadelere sahip katsayılar birbirine eşitlenirse;

$$\begin{aligned} A &= 3 \\ B - 2A &= 2 \\ 4A - 2B + C &= 1 \end{aligned}$$

nihayetinde:

$$A = 3, \quad B = 8, \quad C = 5$$

katsayıları elde edilir. Dolayısıyla aynı  $f(x)$  fonksiyonunu;

$$f(x) = 3(x - 2)^2 + 8(x - 2) + 5$$

ifadesi ile temsil etmek mümkündür.



Güç serisine ait  $a_n$  katsayıları, fonksiyonun kendisi ve türev alma işlemleri ile elde edilebilir. Zira güç serisi;

$$f(x) = a_0 + a_1(x - c) + a_2(x - c)^2 + a_3(x - c)^3 + \dots + a_n(x - c)^n \quad (1.121)$$



Örnek uygulama



şeklinde açılıp,  $x = c$  noktası dikkate alınırsa:

$$a_0 = f(c) \quad (1.122)$$

elde edilir. Daha sonra fonksiyonun 1'inci türevi:

$$f'(x) = a_1 + 2a_2(x - c) + 3a_3(x - c)^2 + \dots + na_n(x - c)^{n-1} \quad (1.123)$$

şeklinde yazılıp, yine  $x = c$  noktası dikkate alınırsa:

$$a_1 = f'(c) \quad (1.124)$$

bulunur. Aynı yaklaşım paralelinde ikinci türev alınır;

$$f''(x) = 2a_2 + 3.2a_3(x - c) + \dots + n.(n - 1)a_n(x - c)^{n-2} \quad (1.125)$$

ve aynı noktadaki  $c$  değeri yerine koyulursa bir sonraki;

$$a_2 = \frac{1}{2}f''(c) \quad (1.126)$$

katsayısı elde edilir. Fonksiyonun 3'üncü türevi alındığında ise;

$$f'''(x) = 3.2a_3 + \dots + n.(n - 1).(n - 2)a_n(x - c)^{n-3} \quad (1.127)$$

yazılıp buradan;

$$a_3 = \frac{1}{3!}f'''(c) \quad (1.128)$$

katsayısına ulaşılır. Tahmin edileceği üzere türev mertebesi artırılarak diğer katsayılara, en nihayetinde son katsayı niteliğindeki  $a_n$  değerine ulaşılabilir:

$$a_n = \frac{1}{n!}f^{(n)}(c) \quad (1.129)$$

Elde edilen katsayı değerleri güç serisinde yerine konulduğunda;

$$f(x) = f(c) + f'(c)(x - c) + \frac{1}{2!}f''(c)(x - c)^2 + \dots + \frac{1}{n!}f^{(n)}(c)(x - c)^n \quad (1.130)$$



açılımına ulaşılır. Bu açılım, Taylor serisi açılımı olarak adlandırılır. Tabiatıyla  $x$  noktası,  $c$  noktasına yakın bir nokta olarak seçildiğinde aralarında;

$$x = c + h \quad (1.131)$$

ilişkisi kurulabilir. Burada  $h$  değeri olarak genellikle küçük bir değer seçilir. Tanımlı  $x$  noktası seri açılımında dikkate alınırsa:

$$f(x) = f(c) + f'(c)h + \frac{1}{2!}f''(c)h^2 + \frac{1}{3!}f'''(c)h^3 + \dots + \frac{1}{n!}f^{(n)}(c)h^n \quad (1.132)$$

şeklindeki Taylor serisi açılımı elde edilmiş olur.

#### f. Gradyan kavramı

Daha önce  $\nabla$  imgesi ile simgelenen nabla operatörünün diğer bir adı da gradyan olup herhangi bir  $f$  fonksiyonunun gradyanı niteliğindeki  $\nabla f$  ifadesinin geometrik bir anlamı da vardır. Zira gradyan yöneysel bir büyüklük olup gösterdiği yön, o noktadaki fonksiyon değerinin en fazla artış göstereceği yöndür. Gradyanın şiddeti, o noktanın eğim değerini ifade eder. Dolayısıyla gradyanın yüksek şiddet değeri, o noktada eğimin büyük olduğunu, buna bağlı olarak fonksiyonun değerinde büyük artışlar meydana geldiğini gösterir. Fonksiyonun gradyanı bilgisi, eniyileme süreçlerinde oldukça işe yarayan bir veridir. Zira gerek doruklaştırma ve gerekse de minimizasyon problemlerinde fonksiyonun değerini en fazla artıran veya azaltan yön bilgisi bu sayede elde edilebilir.



$f(x, y) = 2xe^{-x^2-y^2}$  şeklindeki bir fonksiyonun  $x = 0.8, y = 0.2$  noktasındaki gradyan yöneyi;

$$\nabla f(x, y) = \frac{\partial f}{\partial x} \mathbf{i} + \frac{\partial f}{\partial y} \mathbf{j}$$

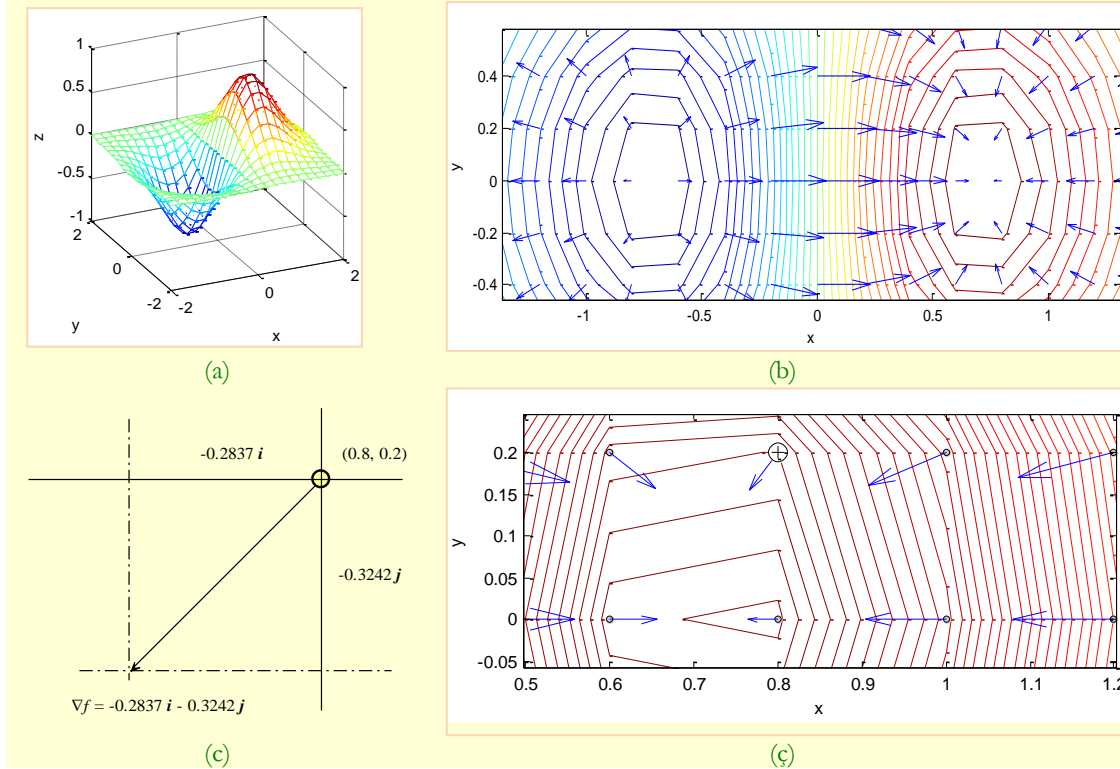
olarak tanımlanan gradyan denklemleri kullanılarak kolayca belirlenebilir.



Örnek uygulama

<sup>i</sup> Gradient





**Şekil 1.44** (a) Yüzey fonksiyonunun perspektif görünümü, (b) Aynı fonksiyonun eşdeğer çizgileri ve gradyan verisine göre çizilen gradyan yönleri, (c) Örnek olarak seçilen  $(0.8, 0.2)$  noktasına ait gradyan yöneyi, (ç) Örnek olarak alınan  $(0.8, 0.2)$  noktasının bulunduğu yerel bölgede diğer bazı noktalara ait gradyan yöneyleri.

Buna göre  $f(x, y)$  fonksiyonunun tanım çerçevesinde kısmi türevler;

$$\frac{\partial f}{\partial x} = 2(1 - 2x^2)e^{-x^2-y^2}, \quad \frac{\partial f}{\partial y} = -4xye^{-x^2-y^2}$$

şeklinde elde edilir ve gradyan tanımında yerine koyulursa;

$$\nabla f(x, y) = 2(1 - 2x^2)e^{-x^2-y^2} \mathbf{i} - 4xye^{-x^2-y^2} \mathbf{j}$$



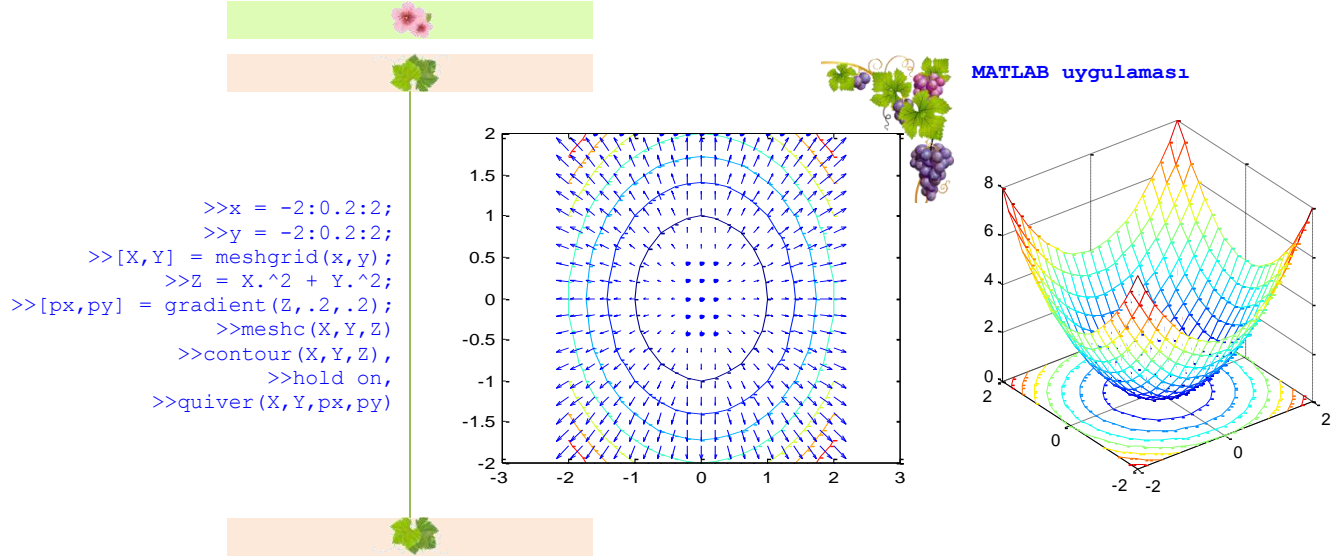


biçimindeki gradyan yöneyi bulunur. Bu ifadede  $x = 0.8$ ,  $y = 0.2$  değerleri dikkate alınırsa, o noktadaki gradyan yöneyi ve eğimi;

$$\nabla f(0.8,0.2) = -0.2837\mathbf{i} - 0.3242\mathbf{j}$$

$$|\nabla f(0.8,0.2)| = [(-0.2837)^2 + (-0.3242)^2]^{1/2} = 0.4308$$

olarak hesaplanır.



MATLAB uygulaması

### g. Birincil ve ikincil türev matrisleri

Gradyan kavramı, görüldüğü üzere sayısal olarak tanımlanabilecek ve bünyesinde  $n$  değişken içeren herhangi bir  $f(\mathbf{x})$  fonksiyonunun kısmi türevlerinden oluşan bir kavramdır. Bazı fonksiyonlar ise yöneysel karakterde olabilir. Bu durumda her bir yöney bileşeni farklı bir fonksiyon içerebilir. Örneğin  $F$  fonksiyonu;

$$F = f_1\mathbf{e}_1 + f_2\mathbf{e}_2 + \dots + f_m\mathbf{e}_m \quad (1.133)$$

şeklinde ve her bir  $f_i$  bileşeni de;

$$f_i(x_1, x_2, \dots, x_n) \quad (1.134)$$





biçiminde tanımlanabilir. Bu durumda  $F$  fonksiyonunun birincil türev tanımı, her bir alt bileşenin kısmi türevlerini içerecek şekilde tanımlanır. Doğal olarak her bir bileşenin kısmi türevleri, satır ve sütun tarzında düzenlenerek bir matris haline getirilebilir. Buna göre birincil türev matrisi<sup>i</sup> olarak adlandırılan söz konusu  $J_F$  matrisi:

$$J_F(x_1, x_2, \dots, x_n) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \dots & \frac{\partial f_m}{\partial x_n} \end{bmatrix} \quad (1.135)$$

olarak yazılabilir. Dikkat edileceği üzere  $m = 1$  olması halinde birincil türev matrisi, gradyan formuna dönüşmektedir. Bu bağlamda birincil türev matrisini,  $F$  fonksiyonunun yerel davranışı (örneğin artan veya azalan  $F$  fonksiyon değeri) hakkında veri içeren bir matris olarak görmek mümkündür.



$$F(r, \theta, \varphi) = r \sin(\theta) \cos(\varphi) \mathbf{e}_r + r \sin(\theta) \sin(\varphi) \mathbf{e}_\theta + r \cos(\theta) \mathbf{e}_\varphi$$

şeklindeki bir fonksiyonun birincil türev matrisini;

$$J_F(r, \theta, \varphi) = \begin{bmatrix} \frac{\partial f_1}{\partial r} & \frac{\partial f_1}{\partial \theta} & \frac{\partial f_1}{\partial \varphi} \\ \frac{\partial f_2}{\partial r} & \frac{\partial f_2}{\partial \theta} & \frac{\partial f_2}{\partial \varphi} \\ \frac{\partial f_3}{\partial r} & \frac{\partial f_3}{\partial \theta} & \frac{\partial f_3}{\partial \varphi} \end{bmatrix} = \begin{bmatrix} \sin(\theta) \cos(\varphi) & r \cos(\theta) \cos(\varphi) & -r \sin(\theta) \sin(\varphi) \\ \sin(\theta) \sin(\varphi) & r \cos(\theta) \sin(\varphi) & r \sin(\theta) \cos(\varphi) \\ \cos(\theta) & -r \sin(\theta) & 0 \end{bmatrix}$$

biçiminde yazmak mümkündür. Burada  $f_i$  fonksiyonları;

$$f_1 = r \sin(\theta) \cos(\varphi), \quad f_2 = r \sin(\theta) \sin(\varphi), \quad f_3 = r \cos(\theta)$$

şeklinde dir.



Örnek uygulama

<sup>i</sup> Jacobean matrix





İkincil türev matrisi<sup>i</sup>, ikinci mertebeden türev bilgisine dayalı olarak türetilen bir matris türüdür. Bununla beraber fonksiyonun yöneysel bir karaktere sahip olması durumunda söz konusu veri, matris olmaktan ziyade 3-dizimli sayı grubu halini alır. Buna göre fonksiyonun sayısal nitelikteki;

$$f(\mathbf{x}) = f(x_1, x_2, \dots, x_n)$$

formunda olması halinde ikincil türev matrisini:

$$H_f = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \dots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \dots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \dots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix} \quad (1.136)$$

olarak yazabiliriz. Bahse konu matris, birincil türev matris formunda da yazılabilir:

$$H_f = J(\nabla f)(\mathbf{x}) \quad (1.137)$$

Bir önceki ifadede dikkat çeken husus, matrisin kare ve aynı zamanda simetrik matris olmasıdır. Zira kısmi türev uygulamasında öncelik-sonralık farkı, sonucu etkilemez:

$$\frac{\partial^2 f}{\partial x_i \partial x_j} = \frac{\partial^2 f}{\partial x_j \partial x_i} \quad (1.138)$$

Öte yandan, ifade edildiği üzere söz konusu fonksiyonun yöneysel karaktere sahip bir fonksiyonlar fonksiyonu olması halinde ikincil türev işlemi uygulaması, matris yerine 3-dizimli sayı grubu ile ifade edilir. Buna göre  $n$  boyutlu değişkene sahip;

$$F(f_1, f_2, \dots, f_m)(\mathbf{x})$$

şeklindeki bir fonksiyonun ikincil türev dizini;

---

<sup>i</sup> Hessian matrix



$$H_F = H_{n \times n \times m} \quad (1.139)$$


biçimindeki bir dizin olacaktır. Burada derinlik konumunu gösteren üçüncü alt indis,  $F$  fonksiyonundaki her bir  $f_k$  fonksiyon bileşeni ifade etmektedir. Örneğin  $H_{2,3,2}$  konumundaki ikincil türev terimini:

$$H_{2,3,2} = \frac{\partial^2 f_2}{\partial x_2 \partial x_3} \quad (1.140)$$

biçiminde ifade etmek mümkündür. Diğer dizin terimleri de benzer şekilde ifade edilebilir:

$$H_{i,j,k} = \frac{\partial^2 f_k}{\partial x_i \partial x_j} \quad (1.141)$$

Birincil türev matrisinin yerel özellikleri ifade etmesi gibi ikincil türev matrisi de fonksiyona ait karakteri tanımlamada yardımcı olur. Zira bu ifade, değişimin değişimini göstermekte olup azalan veya artan bir değişim, bize varılacak fonksiyon değerinin azami veya asgari olup olmadığına dair fikir verir. Örneğin ikincil türev matrisinin, bulunulan  $\mathbf{x}$  noktasında pozitif tanımlı olması, yerel minimuma doğru gidişatı gösterir. Matrisin negatif tanımlı olması ise yerel maksimuma doğru bir gidişatı ifade eder. Matrisin pozitif ve negatif özdeğerlerinin olması halinde bulunulan nokta, bir bükülme noktasıdır.



**MATLAB uygulaması**

```

>> syms x y z
>> f=2*x+y-z
ans =
>> jacobian(f)
[ 2, 1, -1]

f =
2*x^2 + x*y^3 - z*x*y
>> f=2*x^2+x*y^3-z*x*y
ans =
>> jacobian(f)
[ y^3 - z*y + 4*x, 3*x*y^2 - x*z, -x*y]

ans =
[ 4, 3*y^2 - z, -y]
[ 3*y^2 - z, 6*x*y, -x]
[ -y, -x, 0]

```

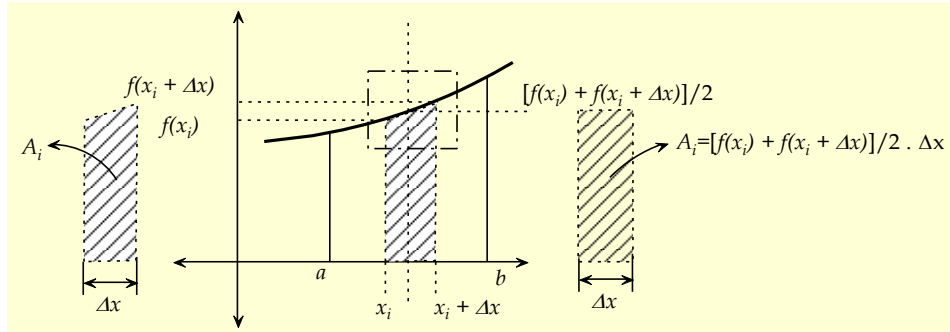
```
ans =
[      4, 3*y^2 - z, -y]
[ 3*y^2 - z, 6*x*y, -x]
[      -y,      -x,  0]
```

## ğ. Analizin en temel önermesi

Analiz biliminin en temel önermesi olarak geçen kanıtsav<sup>i</sup>, esasen türev ile integral arasındaki bağı ifade eder. İlave olarak, sınırları belirli integral kavramının tanımlanmasına da yardımcı olur. Söz konusu önermenin tanımlanması için öncelikle integral kavramının açıklanması yerinde olacaktır.

### (1) İntegral

İntegral kavramını, fonksiyonun temsil ettiği eğrinin altında kalan alan şeklinde ifade etmek mümkündür. Diğer taraftan integral, fonksiyon değerlerinin toplamsal karşılığı olarak da tanımlanabilir. Fonksiyon eğrisinin altında kalan alan tanımlaması çerçevesinde Şekil 1.45'teki gibi bir çizmeyi dikkate alabiliriz. Buna göre  $[a, b]$  aralığında sürekliliği olan fonksiyonun herhangi bir  $x_i$  noktasındaki değeri  $f(x_i)$  olsun. Aynı noktanın  $\Delta x$  kadar uzağındaki fonksiyon değeri ise  $f(x_i + \Delta x)$  olacaktır.



Şekil 1.45 İntegral kavramının geometrik anlamı.

<sup>i</sup>Theorem

Söz konusu  $x_i$  ve  $x_i + \Delta x$  aralığındaki eğrinin altında kalan alan  $A_i$  olarak tanımlanıp;

$$A_i = f(\bar{x}_i)\Delta x \quad (1.142)$$

ifadesi ile elde edilebilir. Burada  $\bar{x}_i$  değeri,  $x_i$  ve  $x_i + \Delta x$  aralığındaki bir değer olup nihayetinde eğrinin altında kalan alanı, dikdörtgen alanı ile eşleştiren bir değerdir. Bu değere karşılık gelen fonksiyon değerini, ortalama değer yaklaşımını dikkate alarak:

$$f(\bar{x}_i) = \frac{1}{2}[f(x_i) + f(x_i + \Delta x)] \quad (1.143)$$

şeklinde açabiliriz. Birim alan niteliğindeki  $A_i$  alanını,  $[a, b]$  aralığında dikkate aldığımızda bu aralıktaki toplam alan değerini;

$$\Delta x = \frac{(b - a)}{n} \quad (1.144)$$

olmak üzere,

$$\sum_{i=1}^n A_i = \sum_{i=1}^n f(\bar{x}_i)\Delta x \quad (1.145)$$

şeklindeki toplamsal değer olarak ifade edebiliriz. Bu ifade  $\Delta x$  ortak parantezine alındığında;

$$\sum_{i=1}^n A_i = \Delta x \sum_{i=1}^n f(\bar{x}_i) \quad (1.146)$$

biçiminde de yazılabilir. Böyle bir yazımın ifade ettiği mana, eğrinin altında kalan alan kavramının, aynı zamanda fonksiyon değerlerinin ölçeklendirilmiş toplamı şeklinde de yorumlanabilmesidir. Diğer taraftan eğrinin altında kalan alanın, sonsuz sayıdaki küçük alanlara bölünmesi halinde elde edilecek değeri;

$$\lim_{n \rightarrow \infty} \sum_{i=1}^n f(\bar{x}_i)\Delta x \quad (1.147)$$

veya

$$\lim_{\Delta x \rightarrow 0} \sum_{i=1}^n f(\bar{x}_i)\Delta x = \int_a^b f(x)dx \quad (1.148)$$





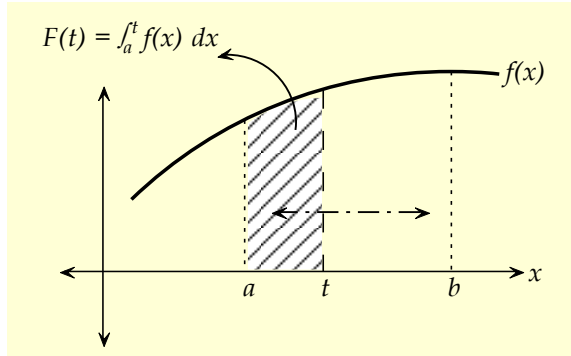
olarak yazabiliriz. Buradaki son simge,  $\int$  integral işareti olup G. W. Leibniz tarafından "summa" kelimesini (toplam) temsilen kullanılmıştır. Sınırları belli olan ve fonksiyon eğrisinin altında kalan alan olarak tanımlanan integral kavramını, biçimsel bağlamda B. Riemann'a da (1826–1866) dayandırabiliriz.

## (2) Önermenin ilk kısmı

Önermeyi açıklamak için öncelikle bağımsız değişkeni  $x$  olan bir  $f(x)$  fonksiyonu tanımlayalım. Ayrıca  $x$  değerinin  $[a, b]$  değer aralığında olduğunu,  $f(x)$  fonksiyonunun da bu aralıkta sürekli ve dolayısıyla da türevlenebilir olduğunu kabul edelim. Daha sonra bahse konu  $[a, b]$  aralığında değeri değişebilen  $t$  gibi ilave bir bağımsız değişken tanımlayalım. Bağımlı değişken niteliğindeki  $f(x)$  fonksiyonunun, tanımlanan  $a$  alt sınırı ile  $t$  üst sınır değeri arasında kalan alan değerini;

$$A = \int_a^t f(x) dx \quad (1.149)$$

şeklinde ifade edebiliriz.



Şekil 1.46 Analizin en temel önermesinin geometrik tanımı.

Buna göre  $t$  değeri değiştikçe üst sınır değeri de değişeceğinden, eğrinin altında kalan alan değeri de değişecektir. Bu bağlamda  $t$  değerini bağımsız değişken, alan değerini de  $t$  değerine bağlı olarak değişen bir bağımlı değişken ya da diğer ifade ile bir  $F(t)$  fonksiyonu olarak tanımlamak mümkündür. Dolayısıyla söz konusu ilişki:



$$F(t) = \int_a^t f(x) dx \quad (1.150)$$

şeklinde ifade edilebilir. Elde edilen bu denklem, analizin en temel önermesinin ilk kısmıdır.

### (3) Önermenin ikinci kısmı

Öte yandan Denklem (1.150) ifadesi kullanılarak belirli integral değerine de ulaşılabilir. Örneğin 0 alt sınır değeri sabit ve  $a, b$  değerleri değişken olmak üzere  $0 < a < b$  gibi sıralamaya sahip üç değer için iki ayrı  $F$  fonksiyon değerini;

$$F(a) = \int_0^a f(x) dx, \quad F(b) = \int_0^b f(x) dx \quad (1.151)$$

biçiminde ifade edebiliriz. İlave olarak  $F(b)$  değerini;

$$F(b) = \int_0^a f(x) dx + \int_a^b f(x) dx \quad (1.152)$$

şeklindeki iki integral değerinin toplamı olarak da yazabiliriz. Bu denklemin son terimi yalnız bırakılır ve  $F(a)$  tanımı da dikkate alınır;

$$F(b) = F(a) + \int_a^b f(x) dx \quad (1.153)$$

$$\int_a^b f(x) dx = F(b) - F(a) \quad (1.154)$$

belirli integral değerine ulaşılır. Bu ifade, önermenin ikinci kısmını göstermektedir.

### (4) İntegral ve türev arasındaki ilişki

Daha önce eğrinin altında kalan alan olarak ifade edilen ve  $t$  değişkenine bağlı olarak değeri değişen  $F(t)$  fonksiyonunun,  $t$  değişkenine göre türetilmesi halinde;



$$\frac{\partial}{\partial t} F(t) = \frac{\partial}{\partial t} \int_a^t f(x) dx \quad (1.155)$$

yazılabilir. İspat edilebilir ki aynı ifadeyi;

$$F'(t) = f(t) \quad (1.156)$$

biçiminde göstermek de mümkündür. Bu ifade,  $F(t)$  integral fonksiyonunun  $t$  noktasındaki türevinin değerinin,  $f(t)$  değerine eşit olduğunu göstermektedir. Buradan çıkarılacak diğer bir anlam da,  $f(x)$  fonksiyonunun  $t$  noktası dikkate alındığında integralinin  $F(t)$  değerine eşit olduğudur. Bu iki sonuç bize türev ve integral arasındaki evrik ilişkiyi gösterir mahiyettedir.



$f(x) = x^3$  şeklindeki bir fonksiyonun  $[0, t]$  aralığındaki integralini alalım:

$$F(t) = \int_0^t f(x) dx = \int_0^t x^3 dx = \frac{x^4}{4} \Big|_0^t = \frac{t^4}{4}$$

Elde edilen bu ifadeyi  $t$  değişkenine göre türetirsek;

$$\frac{\partial}{\partial t} F(t) = 4 \frac{t^3}{4} = t^3$$

sonucu yazılabilir. Bu ise;

$$\begin{aligned} f(t) &= t^3 \\ F'(t) &= f(t) \end{aligned}$$

anlamına gelmektedir.



Analiz biliminin en temel önermesi olarak geçen ve türev ile integral arasındaki ilişkiyi veren yaklaşımın bir sonucu olarak müteakip ifadeyi yazabileceğimizi ifade etmiştik:

$$F'(t) = f(t)$$



Örnek uygulama



Detay açıklama



Bu eşitliği, limit kavramından hareketle ispat etmek mümkündür. Diyelim ki  $[a, b]$  aralığında  $t_1$  ve  $t_1 + \Delta t$  gibi iki değer olsun. Belirtilen ilk nokta için;

$$F(t_1) = \int_a^{t_1} f(x)dx$$

ifadesi ve ikinci nokta için de:

$$F(t_1 + \Delta t) = \int_a^{t_1 + \Delta t} f(x)dx$$

eşitliği yazılabilir. Son yazılan denklemin, bir önceki denklemden taraf tarafa çıkarılması halinde:

$$F(t_1 + \Delta t) - F(t_1) = \int_a^{t_1 + \Delta t} f(x)dx - \int_a^{t_1} f(x)dx$$

eşitliği elde edilir. Eşitliğin sağ tarafındaki ilk terimi, uygun sınır tanımlaması yardımıyla iki ayrı integral şeklinde yazabiliriz:

$$F(t_1 + \Delta t) - F(t_1) = \int_a^{t_1} f(x)dx + \int_{t_1}^{t_1 + \Delta t} f(x)dx - \int_a^{t_1} f(x)dx$$

Dolayısıyla buradan;

$$F(t_1 + \Delta t) - F(t_1) = \int_{t_1}^{t_1 + \Delta t} f(x)dx$$

sonucuna ulaşılır. Öte yandan integralin “eğrinin altında kalan alan” tanımı ve ortalama değer yaklaşımı dikkate alınarak;

$$\int_{t_1}^{t_1 + \Delta t} f(x)dx = f(\bar{t})\Delta t$$

denklemini yazılabilir. Bu ifade, bir öncekinde yerine koyulur ve düzenleme yapılırsa;

$$F(t_1 + \Delta t) - F(t_1) = f(\bar{t})\Delta t$$

$$f(\bar{t}) = \frac{F(t_1 + \Delta t) - F(t_1)}{\Delta t}$$

oranı elde edilir.







Her iki tarafın limiti alınırsa;

$$\lim_{\Delta t \rightarrow 0} f(\bar{t}) = \lim_{\Delta t \rightarrow 0} \frac{F(t_1 + \Delta t) - F(t_1)}{\Delta t}$$

sonuçta;

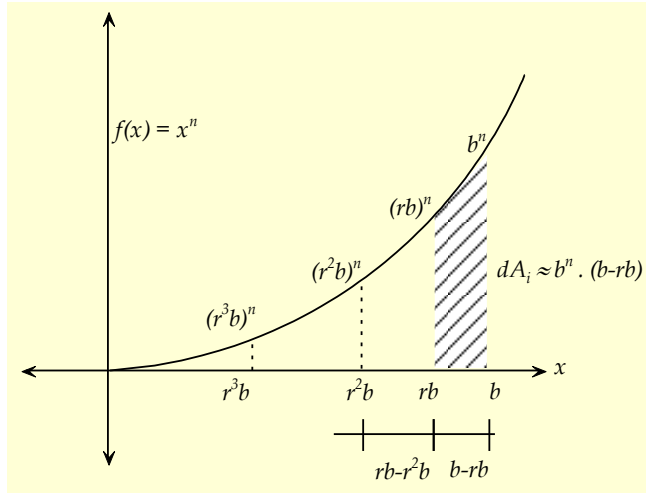
$$f(t_1) = F'(t_1)$$

ifadesine ulaşılır. Dikkat edileceği üzere  $\Delta t \rightarrow 0$  olması halinde ortalama değer olan  $\bar{t}$  değeri  $t_1$  değerine yakınsamakta, eşitliğin sağ tarafı türev halini almaktadır.



### (5) Temel bazı integral alma yöntemleri

İntegral alma kurallarını, türev işlemlerinin tersi şeklinde düşünebileceğimiz gibi, seri açılımları yaklaşımı ile de çözümlenmek mümkündür. Bu meyanda örnek olması açısından,  $f(x) = x^n$  fonksiyonunun integralinin seri ve nihayetinde limit kavramları ile nasıl elde edildiğine bakılabilir. Burada takip edilecek yöntem Fermat'a ait yöntem olup temelinde, eğrinin altında kalan alanın seriye dönüştürülmesi fikri yatmaktadır.



Şekil 1.47 Üssel fonksiyona ait integral hesaplamasında seri yaklaşımı.



Buna göre  $f(x) = x^n$  fonksiyonunun  $[0, b]$  aralığındaki integral ya da diğer ifade ile alan değerini, küçük dikdörtgen alanlarının toplamı şeklinde tanımlayabiliriz. Öncelikle  $[0, b]$  aralığını,  $S = \{0, \dots, r^k b, r^{k-1} b, \dots, r^3 b, r^2 b, r b, b\}$  şeklindeki aralıklara böldüğümüzü düşünelim. Burada  $r$ ,  $0 < r < 1$  ve  $k, k = 0 \rightarrow \infty$  şeklinde tanımlanmaktadır. Bu durumda eğrinin altında kalan alanı:

$$\int_0^b x^n dx = \sum_{i=1}^{\infty} dA_i \quad (1.157)$$

şeklinde ifade edebiliriz. Eşitliğin sağ tarafındaki seri ifadesini  $S_n(r)$  simgesi ile ifade edip, dikdörtgen şeklindeki yaklaşık alan değerlerini dikkate alarak açarsak;

$$S_n(r) = b^n(b - rb) + (rb)^n(rb - r^2b) + (r^2b)^n(r^2b - r^3b) + \dots \quad (1.158)$$

terimlerini yazabiliriz. Her terimdeki ortak çarpanlar dikkate alındığında aynı denklem;

$$S_n(r) = b^{n+1}(1 - r) + (rb)^{n+1}(1 - r) + (r^2b)^{n+1}(1 - r) + \dots \quad (1.159)$$

şeklinde yazılabilir ve bu terimler yeniden düzenlendiğinde:

$$S_n(r) = b^{n+1}(1 - r)\{1 + r^{n+1} + r^{2(n+1)} + r^{3(n+1)} + \dots\} \quad (1.160)$$

sonucu elde edilir. Parantez içerisindeki seri;

$$S_n(r) = b^{n+1}(1 - r) \sum_{k=0}^{\infty} (r^{n+1})^k \quad (1.161)$$

şeklinde yazılabilir. Öte yandan  $r^{n+1}$  değeri nihayetinde sayısal bir değer olup  $t = r^{n+1}$  tanımı yapılabilir ve seri toplamı;

$$\sum_{k=0}^{\infty} (r^{n+1})^k = \sum_{k=0}^{\infty} t^k = 1 + t + t^2 + t^3 + \dots \quad (1.162)$$

olarak açılabilir. Bu serinin toplamı da;



$$\sum_{k=0}^m t^k = \frac{1-t^{m+1}}{1-t} \quad (1.163)$$

$$m = \infty \Rightarrow \frac{1}{1-t}$$

şeklinde ifade edilir. Burada  $t < 1$  olması nedeniyle  $t^\infty = 0$  olarak kabul edilmiştir. Değişken dönüşümünün aslına dönülürse;

$$S_n(r) = b^{n+1}(1-r) \frac{1}{1-r^{n+1}} \quad (1.164)$$

eşitliği elde edilir. Bu denklem tekrar düzenlenirse;

$$S_n(r) = b^{n+1} \frac{1}{(1-r^{n+1})/(1-r)} \quad (1.165)$$

ve elde edilen paydanın esasen daha önce açılımı kullanılan bir seri olduğu düşünülürse;

$$S_n(r) = b^{n+1} \frac{1}{1+r+r^2+r^3+\dots+r^n} \quad (1.166)$$

denklemine ulaşılır. Bu denklemdeki  $r$  değerinin 1 değerine yaklaşması, iki nokta arası mesafenin de  $\Delta x \rightarrow 0$  değerine yaklaşması anlamına gelir. Dolayısıyla payda değerini, toplamda  $n$  sayısına eşitleyebiliriz;

$$S_n(r) = b^{n+1} \frac{1}{1+n} \quad (1.167)$$

Bu sonuç bize, üssel bir fonksiyonun integralinin alınmasında kullanabileceğimiz ve Denklem (1.168)'deki gibi tanımlanabilecek ifadeyi verir:

$$\int_0^b x^n dx = \frac{b^{n+1}}{1+n} \quad (1.168)$$

Benzer seri açılımı yaklaşımlarının diğer fonksiyonlar için de sergilenmesi halinde müteakip çizelgedeki gibi bazı integral alma kurallarını tanımlamak mümkündür.





Fonksiyon	İntegral alma işlemi	Fonksiyon	İntegral alma işlemi
$\int adx$	$ax + C$	$\int \tan(x)dx$	$-\ln \cos(x)  + C$
$\int x^n dx, n \neq -1$	$\frac{x^{n+1}}{n+1} + C$	$\int \cot(x)dx$	$\ln \sin(x)  + C$
$\int \frac{dx}{x}$	$\ln x  + C$	$\int f(x)g'(x)dx$	$f(x)g(x) - \int g(x)f'(x)dx$
$\int a^x dx$	$\frac{a^x}{\ln a} + C$	$\int \frac{f'(x)}{f(x)} dx$	$\ln f(x) + C$
$\int \sin(x)dx$	$-\cos(x) + C$	$\int f(x)^n f'(x)dx, n \neq -1$	$\frac{f(x)^{n+1}}{n+1} + C$
$\int \cos(x)dx$	$\sin(x) + C$	$\frac{\partial}{\partial y} \int_{f(y)}^{g(y)} F(x, y)dx$	$g'(y)F(g(y), y) - f'(y)F(f(y), y) + \int_{f(y)}^{g(y)} \frac{\partial}{\partial y} F(x, y)dx$

**Çizelge 1.3** Bazı yaygın integral alma yöntemleri, ( $C$  simgesi sabit sayıyı temsil etmektedir).



$$\sum_{k=0}^m t^k = \frac{1-t^{m+1}}{1-t}$$



Detay açıklama

eşitliğini, basit bir yaklaşımla kolayca elde etmek mümkündür. Buna göre;

$$S_m(t) = 1 + t + t^2 + t^3 + \dots + t^m$$

$$t \cdot S_m(t) = t + t^2 + t^3 + \dots + t^m + t^{m+1}$$

ifadeleri birbirlerinden çıkarılırsa;

$$S_m(t) - tS_m(t) = 1 - t^{m+1}$$

yazılabilir. Bu denklemdaki toplam değer olan  $S_m(t)$  yalnız bırakılırsa;

$$S_m(t) = \frac{1-t^{m+1}}{1-t}$$

sonucuna ulaşılır.





**MATLAB uygulaması**



$$\int (x^2 - 3x + 5) dx$$

```
>> syms x
>> f=x^2-3*x+5
>> int(f,x)
f =
x^2 - 3*x + 5
ans =
(x*(2*x^2 - 9*x + 30))/6
```

$$\int_0^{\pi/2} \sin(3x) dx$$

```
>> int(sin(3*x), x, 0,
pi/2)
ans =
1/3
```

$$\int \left(x^n + \frac{2}{x}\right) dx$$

```
>> syms x n;
>> f = x^n+2/x
>>int(f,x)
f =
x^n + 2/x
ans =
2*log(x) + x^(n + 1)/(n + 1)
```

$$\int_0^1 \frac{1}{x^3 - 3x - 4} dx$$

```
>> f = @(x)1./(x.^3-3*x-
4)
>> quad(f,0,1)
f =
@(x)1./(x.^3-3*x-4)
ans =
-0.1933
```

$$\int_{\pi/2}^{2\pi} \int_0^{\pi} \left\{2y \sin(x) + \frac{1}{2}x \cos(y)\right\} dx dy$$

```
>> Q = quad2d(@(x,y)
2*y.*sin(x)+1/2*x.*cos(y)
,pi/2,2*pi,0,pi)
Q =
-9.8696
```

## h. Rastsal sayılar

Rastsallık kavramı, belli bir düzen içermeme, ilişki içerisinde olmama şeklinde tanımlanabilir. Dolayısıyla bir küme içerisindeki rastsal sayılar arasında herhangi bağıntı yoktur. Eniyileme süreçlerinde rastsal sayıların kullanımının gittikçe yaygınlaştığı söylenebilir. Bu nedenle, örnek olması bağlamında bazı basit rastsal sayı üretim yöntemleri üzerinde kısaca durmak yararlı olacaktır. Literatür kapsamında bilinen en eski ve basit rastsal sayı üretim yöntemlerinden birisi, doğrusal tabanlı rastsal sayı üreticidir. Bu yöntemde esas itibarı ile tamamen rastsal sayı üretmek söz konusu değildir. Bununla beraber üretilen sayılar, sözde rastsal sayı<sup>ii</sup> olarak tanımlanır. Zira



**Şekil 1.48** Rastsallık, özgür iradenin de temelinde olan bir kavramdır.

<sup>i</sup> Linear congruential generator

<sup>ii</sup> Pseudo-random number



sayılar arasında doğrudan olmasa da serisel bir ilişki vardır. Yöntem, ardı ardına tam sayı üretmekte olup dikkate alınan yaklaşım:

$$x_{n+1} = \{ax_n + c\} \bmod m \quad (1.169)$$

şeklinde. Burada  $m > 0$  olup genlik olarak tanımlanmaktadır. Buna göre çıkan her sayı,  $m$  genlik değerine bölünmekte ve kalan tam sayı değeri dikkate alınmaktadır. Diğer sayılardan  $a$  değeri çarpan olup  $0 < a < m$  şeklinde,  $c$  değeri artım olup  $0 < c < m$  aralığında tanımlanır. Başlangıç değeri niteliğindeki  $x_n$  değeri ise  $0 \leq x_n < m$  aralığında rastgele seçilen ve tohum değer olarak adlandırılan bir sayıdır. Genellikle genlik değeri,  $2^{32}$  veya  $2^{64}$  gibi büyük değerler seçilerek oldukça geniş bir yelpazede rassal sayılar üretmek mümkündür. Ancak genlik nedeniyle, elde edilen sayılar dönemsel ya da diğer bir ifade ile periyodik bir karaktere sahiptir. Doğrusal tabanlı rassal sayı üreticinin bir benzeri yöntem,  $c$  artım değerinin de serisel olarak güncellendiği<sup>ii</sup> yöntemdir. İlkinde olduğu gibi yöntem, sözde rassal sayılar üretmekte olup dönemsel genliği oldukça geniş olabilmektedir. Buna göre yöntemde dikkate alınan örnek bir metrik:

$$x_{n+1} = \{ax_n + c_n\} \bmod m \quad (1.170)$$

$$c_{n+1} = \left\lfloor \frac{ax_n + c_n}{m} \right\rfloor$$

denklemleri ile tanımlanır. Denklemden geçen [...] işareti, bölüm değerinin taban değerine yuvarlanması anlamına gelmektedir<sup>(38)</sup>. Elde edilen sayıların  $0 < x_{n+1} < 1$  aralığında olması istendiğinde, basit bir normalizasyon yapmak yeterli olabilir;

$$u_{n+1} = \frac{x_{n+1}}{m} \quad (1.171)$$

Literatürde diğer başka rassal sayı üretim yöntemleri de mevcut olup bazı yöntemler gerçek manada rassal sayı üretebilmektedir<sup>(39)</sup>. Hesaplamalı eniyileme süreçlerinde kullanılan pek çok bilgisayar programlama dilinde bahse konu rassal sayı üreticileri dikkate alınmaktadır. Sayı üretimi dâhilinde önemli bir husus olan başlangıç değerinin tespiti, genellikle zamana göre yapılır. Zira akıp giden zaman, her an yeni bir başlangıç manasına gelir.

<sup>i</sup> Modulus

<sup>ii</sup> Söz konusu yöntem, İngilizce’de “multiply-with-carry” olarak geçer.





**MATLAB uygulaması**

```
>> x(1) = 1; n = 10;  
a = 2; c = 2; m = 100;  
>> for i=1:n  
    x(i+1) = a*x(i) + c;  
    x(i+1) = mod(x(i+1),m);  
end  
ans =  
1 4 10 22 46 94 90 82 66 34 70  
  
>> n = 10; x(1) = 1; a = 2;  
    c(1) = 2; m = 100;  
>> for i=1:n  
    x(i+1) = a*x(i) + c(i);  
    c(i+1) = floor((x(i+1)/m));  
    x(i+1) = mod(x(i+1),m);  
end  
ans =  
1 4 8 16 32 64 28 57 14 29 58  
  
>> rand  
ans =  
0.8147  
>> rand(1,3)  
ans =  
0.7431 0.3922 0.6555  
>> rand(2,3)  
ans =  
0.0318 0.0462 0.8235  
0.2769 0.0971 0.6948  
  
>> randint(2,3,[0 10])  
ans =  
5 3 2  
10 6 8
```

#### 1.2.4 Analizde sayısal yöntemler

Fiziksel olayların matematiksel olarak tanımlanamaması halinde, ya da tanımlandığı halde matematiksel modelin analitik çözümünün yapılamaması durumunda, çözüm yöntemi olarak sayısal analiz yöntemlerine başvurulur. Matematiksel modelin kurgulanmadığı durumlarda kullanılacak bilgi, deneysel verilere dayanır. Sayısal analiz yöntemleri, oldukça geniş bir yelpazeye yayılmış olup burada sadece sayısal türev ve integral yöntemlerine değinilecektir.



### a. Sayısal türev yöntemleri

Sayısal türev hesaplama yöntemlerinin temelinde Taylor serisi açılımı yatmaktadır. Hatırlanacağı üzere herhangi bir  $x$  noktasındaki  $f(x)$  fonksiyon değeri, yakınındaki bir başka nokta dikkate alarak ifade edilebilmektedir. Buna göre bulunulan nokta  $x_{i+1}$  ve tanımı;

$$x_{i+1} = x_i + h \quad (1.172)$$

olmak üzere serisi açılımı:

$$f(x_{i+1}) = f(x_i) + f'(x_i)h + \frac{1}{2!}f''(x_i)h^2 + \frac{1}{3!}f'''(x_i)h^3 + \dots + \frac{1}{n!}f^{(n)}(x_i)h^n \quad (1.173)$$

biçiminde ifade edilebilir. Bu ifadedeki ikinci ve daha üst mertebeden türevlerin ihmal edilmesi halinde, geriye kalan terimler düzenlendiğinde birinci mertebeden türev değeri:

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{h} \quad (1.174)$$

şeklinde yazılabilir. Bu eşitlik, literatürde ileri fark<sup>i</sup> olarak tanımlanır. Tabiatıyla ihmal edilen terimler nedeniyle bir hata söz konusu olacaktır. Oluşan bu hataya kesme hatası<sup>ii</sup> denir. Nokta ve civarının;

$$x_{i-1} = x_i - h \quad (1.175)$$

biçiminde tanımlanması durumunda Taylor serisi açılımı:

$$f(x_{i-1}) = f(x_i) - f'(x_i)h + \frac{1}{2!}f''(x_i)h^2 - \frac{1}{3!}f'''(x_i)h^3 + \dots + \frac{1}{n!}f^{(n)}(x_i)h^n \quad (1.176)$$

şeklinde dir. Açılımda yüksek mertebedeki terimlerin ihmal edilmesi halinde;

$$f'(x_i) \cong \frac{f(x_i) - f(x_{i-1})}{h} \quad (1.177)$$

şeklindeki birinci mertebeden türev değeri elde edilmiş olur.

<sup>i</sup> Forward differencing

<sup>ii</sup> Truncation error







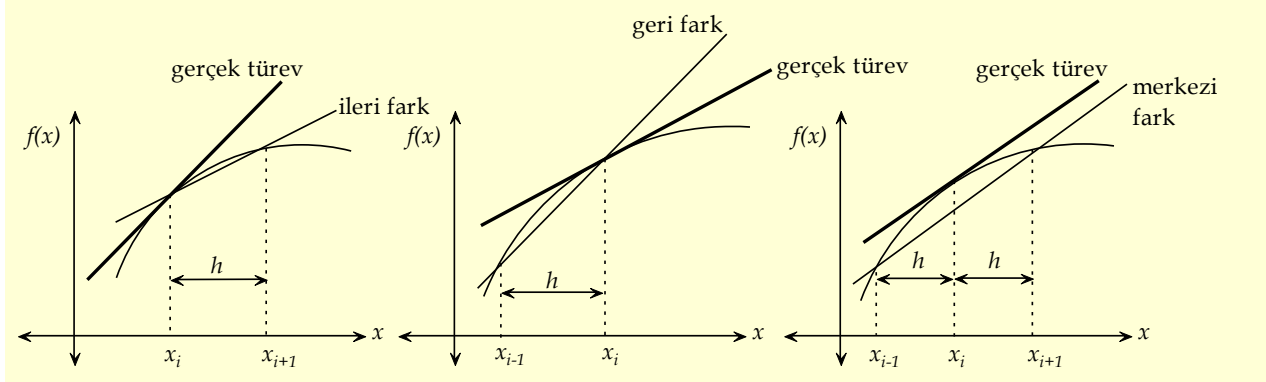
Bu yaklaşım, geri fark<sup>i</sup> yaklaşımıdır. İleri fark yaklaşımı ile geri fark yaklaşımının birbirlerinden çıkarılması durumunda ise elde edilecek türev değeri, merkezi fark<sup>ii</sup> olarak tanımlanır ve yaklaşık olarak;

$$f'(x_i) \cong \frac{f(x_{i+1}) - f(x_{i-1}))}{2h} \quad (1.178)$$

denklemleri ile belirlenir. Öte yandan ileri ve geri fark eldesi için dikkate alınan Taylor serisi açılımlarının toplanması ve elde edilen toplamsal serideki üçüncü mertebeye ve üstü türev içeren terimlerin ihmal edilmesi halinde ikinci mertebeden türev ifadesine ulaşılabilir:

$$f''(x_i) = \frac{f(x_{i+1}) - 2f(x_i) + f(x_{i-1}))}{h^2} \quad (1.179)$$

Elde edilen bu denklem, merkezi farka dayalı ikinci mertebeden türev ifadesi olarak tanımlanır. Benzer güdülemelerle daha yüksek mertebedeki türev terimlerine de ulaşmak mümkündür.



Şekil 1.49 İleri, geri ve merkezi farkın geometrik anlamı.

<sup>i</sup> Backward differencing

<sup>ii</sup> Centered differencing



## b. Sayısal integral yöntemleri

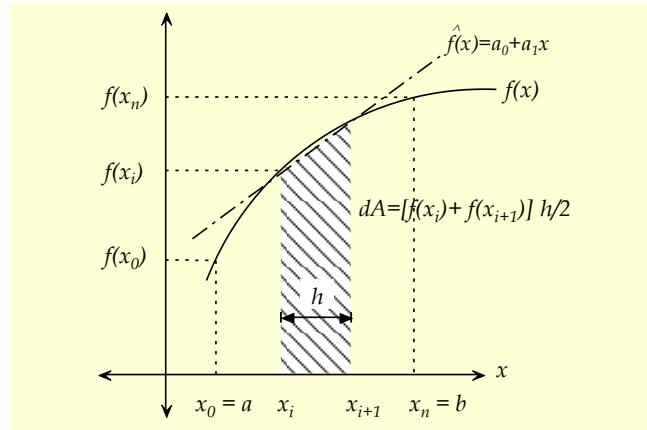
Sayısal integral hesaplama yöntemlerinin temelinde fonksiyonun altında kalan alanın küçük alanlara bölünmesi ve bu alanların hesaplanarak toplamsal alan değerine ulaşılması fikri yatmaktadır. Küçük alanların hesaplanmasında başvurulan yöntem, gerçek fonksiyon yerine daha basit bir fonksiyonu ve altında kalan alanı dikkate almak şeklindedir. Bu meyanda en basit yaklaşım doğrusal denklem yaklaşımıdır. Buna göre  $[a, b]$  aralığında  $f(x)$  fonksiyonunun integralini:

$$I = \int_a^b f(x) dx \quad (1.180)$$

şeklinde ifade edebiliriz. Bu ifadenin analitik yöntemlerle çözülememesi halinde aynı değeri hesaplamak için  $\hat{f}(x)$  gibi daha basit ve integrali alınabilir bir fonksiyon dikkate alınabilir. Dolayısıyla aynı alan değeri:

$$I \cong \int_a^b \hat{f}(x) dx \quad (1.181)$$

denklemini ile ifade edilir.



Şekil 1.50 İntegral hesaplamada 1'inci dereceden polinomial yaklaşım: yamuğun alanı.

Yaklaşık fonksiyon olarak ifade edilen  $\hat{f}(x)$  fonksiyonunun ne olacağı ise tercihe bağlıdır. Genellikle yaklaşık fonksiyon olarak polinomial fonksiyonlar tercih edilir. En basit polinomial



yaklaşım, 1'inci dereceden denklem yaklaşımıdır ve  $n$  adet alt alana bölünmüş bir integralde sayısal integral değeri:

$$I \cong \frac{h}{2} \left[ f(x_0) + 2 \sum_{i=1}^{n-1} f(x_i) + f(x_n) \right] \quad (1.182)$$

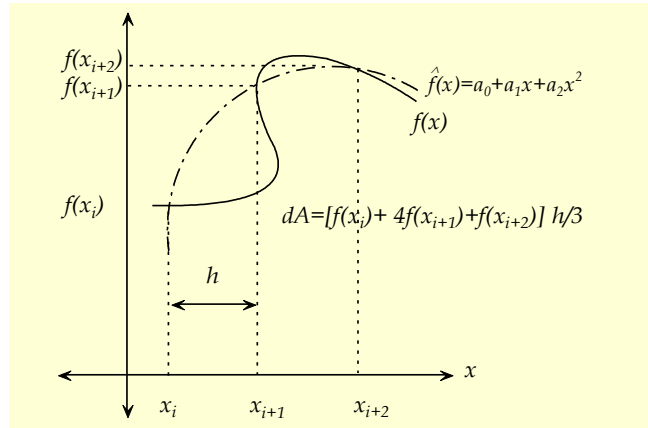
toplamsal değeri ile elde edilebilir. Burada;

$$h = \frac{(b-a)}{n}, \quad x_0 = a, \quad x_n = b \quad (1.183)$$

olarak tanımlanmaktadır. Polinomiyal yaklaşım dâhilinde 1'inci dereceden denklem yerine 2'nci dereceden bir  $\hat{f}(x)$  eğrisel fonksiyonu dikkate alındığında, sayısal integral değeri:

$$I \cong \frac{h}{3} \left[ f(x_0) + 4 \sum_{i=1,3,5,\dots}^{n-1} f(x_i) + 2 \sum_{j=2,4,6,\dots}^{n-2} f(x_j) + f(x_n) \right] \quad (1.184)$$

denkleminde göre hesaplanabilir.



Şekil 1.51 İntegral hesaplamada 2'nci dereceden polinomiyal yaklaşım: parabol alanı.

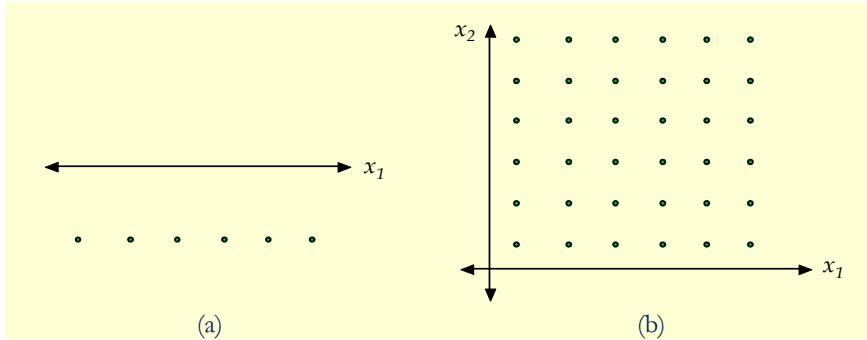


### c. Örnekleme

Örnekleme kavramını, bir  $f(\mathbf{x})$  fonksiyonuna ait bağımsız değişkenlerin, ifade edilen değer aralıklarında örnek değerlerinin alınması şeklinde tanımlayabiliriz. Örnekleme yöntemleri, eniyileme süreçlerinde asgari veya azami fonksiyon değerinin bulunması için gereken başlangıç noktalarını seçmede kullanılabilir. Özellikle sayısal yöntemlerin kullanıldığı eniyileme süreçlerinde örnekleme, bir anlamda ayrıklaştırma işlemi manasına gelmektedir. Örnekleme yöntemlerini üç gruba ayırmak mümkün olup bu gruplar: sistematik örnekleme, rastsal örnekleme ve karma örnekleme şeklindedir. Sistematik örneklemede değişkene ait örnek değerlerin alınması, belli bir yöntemle yapılır. Rastsal örneklemede başlatma etmen rastsallıktır. Bununla beraber rastsallıkta dikkate alınan sayıların sözde rastsal olması, süreci az da olsa belirli bir belirliliğe indirir. Karma yöntemde ise belirli bir sistematik izlenmekle beraber sınırlı çerçevede rastsal sayıların kullanılması söz konusudur.

#### (1) Sistematik örnekleme yöntemleri

Literatürde pek çok sistematik örnekleme yöntemi mevcut olup bu kitap dâhilinde özellikle doğrusal örnekleme, etmensel tasarım ve seçenekleri, merkezi karışım tasarım, Latin çoklu küp tasarım, doğrusal olmayan örnekleme ve Taguchi tasarım örneklemelerinden bahsedilecektir.



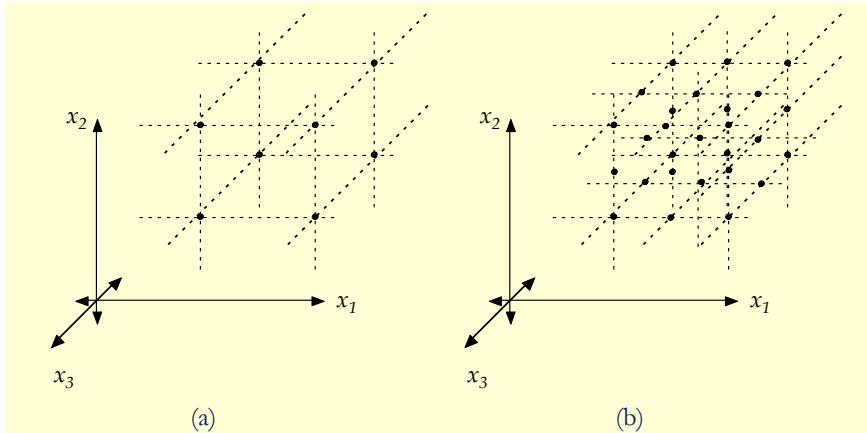
Şekil 1.52 (a) 1-boyutlu, (b) 2-boyutlu ortamda, bağımsız değişken değerlerinin doğrusal olarak 5 eşit aralığa bölünmesi ve ayrıklaştırılması.

<sup>i</sup> Sampling



### (a) Doğrusal örnekleme

Doğrusal örneklemede, bağımsız değişkene ait değer aralığının eşit aralıklara bölünerek örneklenmesi dikkate alınır. Aralığın kaç eşit parçaya bölüneceği, tasarımcının vereceği bir karardır. Değişken sayısının az olduğu fonksiyonlarda doğrusal örnekleme, üzerinde düşünülebilir bir seçenektir. Ancak değişken sayısının artması, örnek noktanın oldukça çok sayıda olmasına sebebiyet verir. Örneğin Şekil 1.52'de görüldüğü üzere, 1-boyutlu bir problemde bağımsız değişken aralığının beş eşit parçaya bölünmesi, sınır değerler dâhil 6 nokta manasına gelmektedir. Aynı bölünmenin 2-boyutlu bir ortamda yapılması halinde ise 36 girdi ve dolayısıyla örnekleme noktası demektir. Bu meyanda  $n$  boyutlu bir ortam uzayında,  $i$  eşit aralığa bölünme söz konusu olduğunda  $(i + 1)^n$  örnekleme noktasından bahsetmek mümkündür. Böyle bir örnekleme büyüklüğü, hesaplama yükü açısından verimli bir yaklaşım olmayabilir.

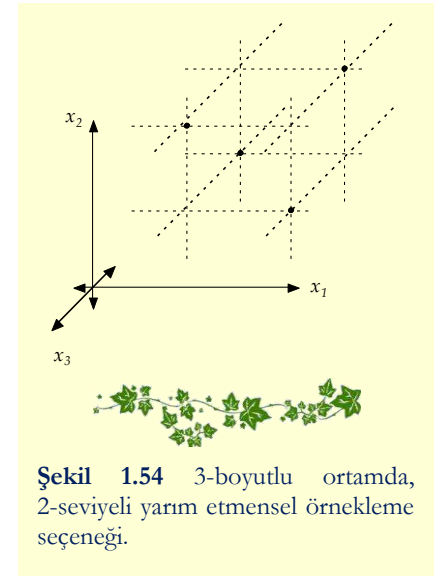


Şekil 1.53 3-boyutlu ortamda; (a) 2-seviyeli, (b) 3-seviyeli etmensele örnekleme.

### (b) Etmensele örnekleme

Etmensele örnekleme ve seçenekleri, esas itibari ile doğrusal örnekleme yöntemleridir. Genellikle 2-seviyeli veya 3-seviyeli<sup>i</sup> etmensele örnekleme yöntemlerinden bahsedilebilir. Terimsel olarak etmeden kastedilen, bağımsız değişkendir. Şekil 1.53 dahilinde görüldüğü üzere, 2-seviyeli ( $2^k$ ,  $k$ : etmen sayısı) etmensele örneklemede, sadece değişkene ait sınır değerler dikkate

<sup>i</sup> “Two-level factorial design” or “three-level factorial design”

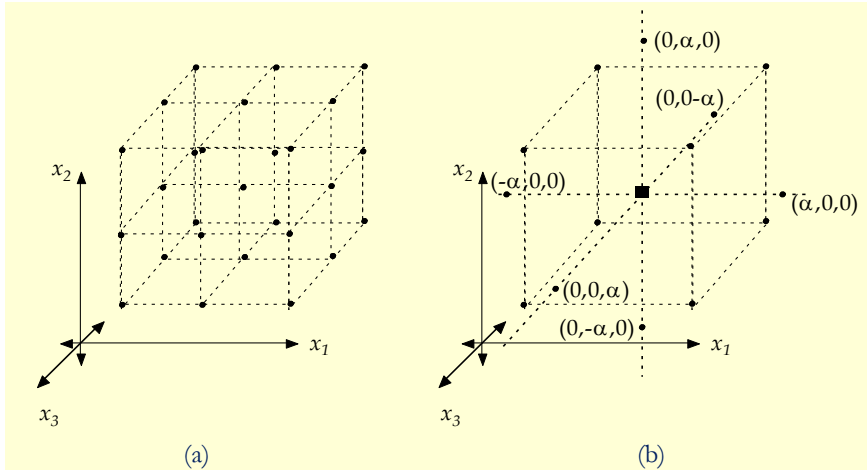


Şekil 1.54 3-boyutlu ortamda, 2-seviyeli yarı etmensele örnekleme seçeneği.



alınarak değer bileşimleri yapılır. 3-seviyeli ( $3^k$ ) etmensel örneklemede ise sınır değerler ve bu değerlerin ortası değer olmak üzere üç nokta ve bileşimleri dikkate alınır.

Etmensel örneklemenin uygulamalarda tercih edilmesinin nedeni, nispeten az sayıda nokta ile örnekleme yapılması ve bu sayede hesaplama yükünün azaltılmasıdır. Bununla beraber her ne kadar az sayıda örnek nokta dikkate alınsa da, etmen sayısının artması örnek nokta sayısında ciddi artışlara neden olabilir. Bu nedenle bazı uygulamalarda tam etmensel örnekleme yerine yarım etmensel örnekleme<sup>i</sup> ( $2^{k-1}$ ) gidilir. Böyle bir yaklaşımda etmen sayısı bir eksik olarak dikkate alınır ve son etmenin etkisi, diğer etmenlerin ortak yansıması olarak düşünülür.



Şekil 1.55 3-boyutlu ortamda; (a) 3-seviyeli tam etmensel, (b) merkezi karışım örnekleme seçenekleri.

### (c) Merkezi karışım örnekleme

Özellikle 3-seviyeli etmensel örneklemede etmen sayısının artması, örnek nokta sayısında ciddi artışa neden olur. Sadece sınır değerlerin dikkate alındığı 2-seviyeli etmensel örnekleme ise doğrusal ve genellikle yetersiz bir örnekleme seçeneği olabilir. Merkezi karışım örnekleme<sup>ii</sup> yaklaşımı, bahse konu her iki yaklaşımı belli oranda içermekte ve az sayıda ama yeteri kadar

<sup>i</sup> One-half fractional factorial design

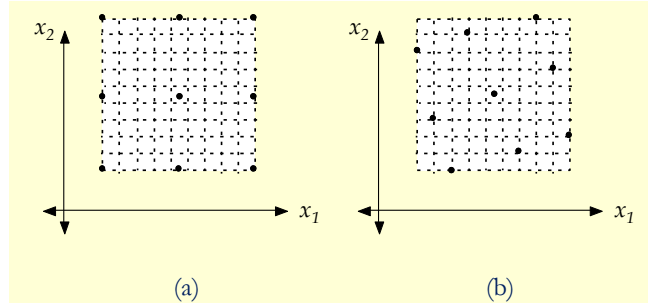
<sup>ii</sup> Central composite design



nokta seçme şansı vermektedir. Buna göre söz konusu yöntem, tam etmensele örnekleme noktaları ile beraber yıldız noktaları ve merkezi noktayı içerir. Yıldız noktadan kastedilen, belli bir  $\alpha$  değerini içerecek şekilde her bir etmen için orta nokta tayinidir. Bahse konu  $\alpha$  değeri,  $k$  etmen sayısı olmak üzere genellikle müteakip ifadeye göre belirlenir;

$$\alpha = 2^{\frac{k}{4}} \quad (1.185)$$

Gerek tam etmensele örnekleme noktaları ve gerekse de yıldız noktalarına ait bağımsız değişkenlerin değerleri ifade edilirken genellikle normalizasyona gidilir. Dolayısıyla üst sınır değeri +1, alt sınır değeri -1, sınır değerlerin orta noktası 0 olarak simgeselleştirilir. Yıldız noktalarda ise etmen değeri  $+\alpha$  veya  $-\alpha$  olarak seçilir, diğer etmenler 0 olarak dikkate alınır. Tüm değerlerin 0 olarak dikkate alındığı nokta, merkez nokta'dır. Bununla beraber farklı yaklaşımların söz konusu olduğu merkezi karışım örnekleme yöntemleri de mevcuttur. Şekil 1.55'te görüldüğü üzere 3 boyutlu bir uzay ortamında,  $3^k$  örnekleme yöntemi ile dikkate alınan nokta sayısı 27 olmaktadır. Aynı ortamda, merkezi karışım örnekleme nokta sayısı ise 15 değerine indirgenebilmektedir.



Şekil 1.56 2-boyutlu ortamda; (a)  $3^k$  örnekleme yöntemi, (b) dikeygen Latin çoklu küp tasarım örneği.

### (ç) Latin çoklu küp tasarım

Örnekleme yapılacak ortamın öncelikle ağ şeklinde öngörülmesi ve ağdaki her bir küp içerisine bir örnek nokta gelecek şekilde seçime gidilmesi sonucu ortaya çıkan örneklemeye, Latin çoklu küp örnekleme<sup>1</sup> adı verilir. İki boyutlu ortamda ağ küçük karelerden, üç boyutlu

<sup>1</sup> Latin hypercube design

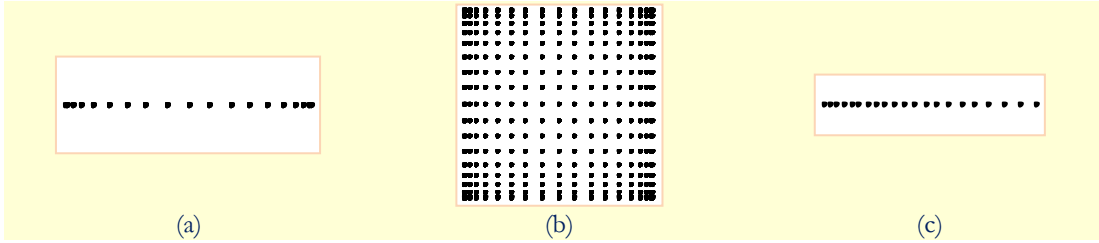


ortamda küplerden, daha yüksek boyutlu ortamlarda ise çoklu küplerden oluşur. Ağ içerisindeki küp uzayında örnek noktanın yerinin tayini için farklı yaklaşımlar yapılabilir. Burada dikeygen Latin çoklu küp tasarımı<sup>(40)</sup> örnek olarak verilebilir. Buna göre  $m$  adet örnek nokta seçilmek üzere söz konusu noktaların dağılımında;

$$\sum_{i=1}^m \sum_{j=i+1}^m \frac{1}{d_{ij}^2} \quad (1.186)$$

toplamsal değerinin asgari olması dikkate alınabilir. Burada  $d_{ij}$  ifadesi,  $i$  ve  $j$  noktaları arasındaki Euclidean mesafesini ifade etmektedir. Şekil 1.56'da verilen örnekte toplam 9 adet nokta; önce 3-seviyeli ( $3^k$ ) tam etmensel örnekleme ve akabinde dikeygen Latin çoklu küp tasarımı dikkate alınarak dağıtılmıştır. İkinci seçeneğin  $3^k$  örnekleme yöntemi ile karşılaştırılması halinde, daha yaygın ve kapsayıcı bir örnekleme imkânı sunulabilir.

#### (d) Doğrusal olmayan örnekleme



Şekil 1.57 (a) 1-boyutlu ortamda her iki sınırdan yoğunlaştırılmış örnekleme, (b) 2-boyutlu ortamda her iki sınırdan yoğunlaştırılmış örnekleme, (c) 1-boyutlu ortamda alt sınırdan yoğunlaştırılmış örnekleme.

Sistematik ancak doğrusal olmayan örnekleme, genellikle belirli bir bölgeye yoğunlaşmak için tercih edilen bir yöntemdir. Örnek noktaların üretiminde genellikle trigonometrik veya üssel işlemlerden faydalanılabilir. Özellikle sınır değerler bölgesinde örnek noktaların çoğaltılması istenirse, *sinus* veya *cosinus* trigonometrik fonksiyonları kullanılabilir. Belli bir sınır bölgesinde yoğunlaşma istendiğinde ise üssel fonksiyon kullanımı daha uygun olabilir. Buna göre  $x_a$  alt sınır ve  $x_u$  üst sınır değer olmak üzere bu sınırlar arasında  $n$  tane örnek nokta;

$$x_i = [x_u - x_a] \cos(\theta_i) \quad (1.187)$$

$$0 \leq \theta_i \leq 360, \quad i = 1, 2, \dots, n$$

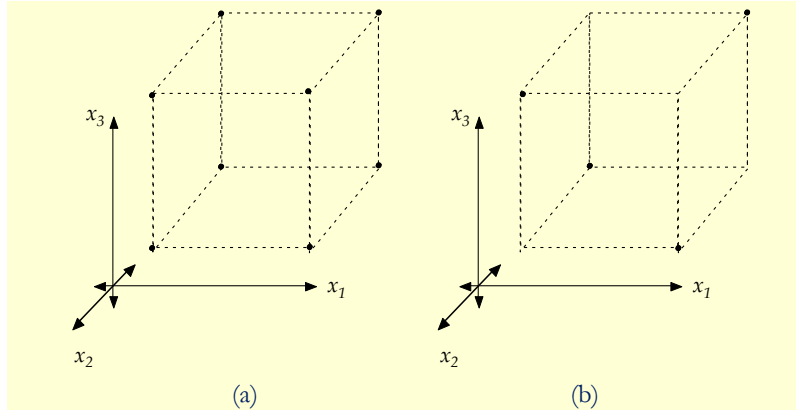
ifadesi ile elde edilebilir.





### (e) Taguchi örnekleme dizinleri

Örnek nokta sayısının azaltılması bağlamında dikkate alınan etkin yöntemlerden birisi de G. Taguchi (1924–2012) tarafından geliştirilen ve Taguchi dizinleri olarak bilinen örnekleme yöntemleridir. Örneklemede, etmen ve bölüm sayısına göre düzenlemeye gidilir. Literatürde kullanılan bazı Taguchi dizinleri<sup>(41)</sup> ve tercih ölçütleri Çizelge 1.4 dâhilinde verilmiştir.



Şekil 1.58 3-boyutlu ortamda; (a)  $2^k$  örnekleme yöntemi ile elde edilen 8 nokta, (b) L4 olarak adlandırılan Taguchi dizini ile elde edilen 4 nokta.

Bölüm sayısı	Etmen sayısı															
	2	3	4	L8	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	L9	$x_1$	$x_2$	$x_3$	$x_4$
2	L4	L4	L8	1	-1	-1	-1	-1	-1	-1	-1	1	-1	-1	-1	-1
3	L9	L9	L9	2	-1	-1	-1	1	1	1	1	2	-1	0	0	0
				3	-1	1	1	-1	-1	1	1	3	-1	1	1	1
				4	-1	1	1	1	1	-1	-1	4	0	-1	0	1
				5	1	-1	1	-1	1	-1	1	5	0	0	1	-1
				6	1	-1	1	1	-1	1	-1	6	0	1	-1	0
				7	1	1	-1	-1	1	1	-1	7	1	-1	1	0
				8	1	1	-1	1	-1	-1	1	8	1	0	-1	1
												9	1	1	0	-1

Çizelge 1.4 (a) Dizin seçimi, (b) L4 Taguchi dizini, (c) L8 Taguchi dizini, (ç) L9 Taguchi dizini.



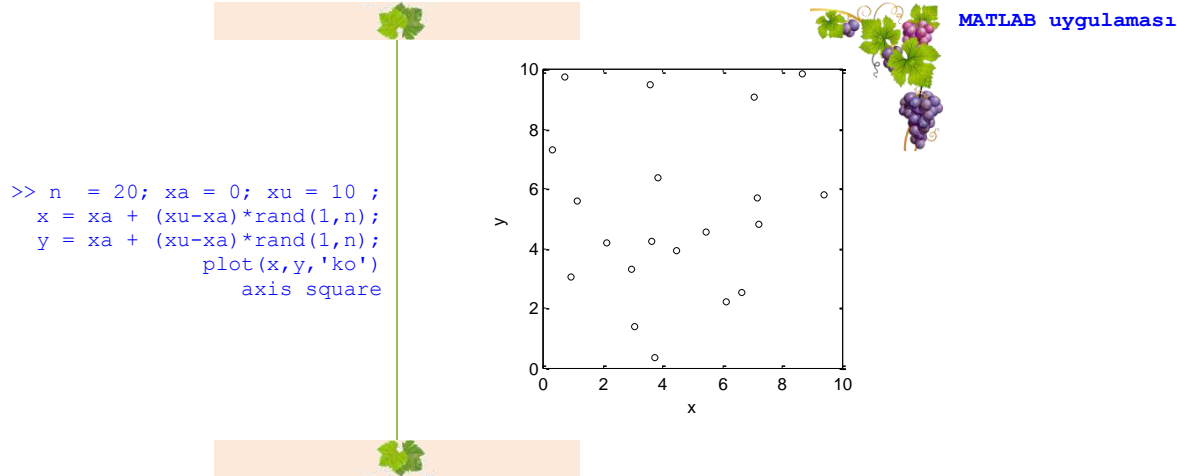


## (2) Rastsal örnekleme yöntemleri

Rastsal örneklemede, bağımsız değişkene ait değer aralığı temel alınarak söz konusu aralıkta rastgele bir şekilde değişken değeri dikkate alınır. Aralığın kaç parçaya bölüneceği, tasarımcının vereceği bir karardır. Daha önce rassal sayılar bahsinde ifade edilen rassal sayı üretim teknikleri, örnekleme adına kullanılan yegâne araçlardır. Doğal olarak, dağılımın belli bir karakter göstermesi beklenemez. Ancak rassal sayı üretiminde dikkate alınan yöntemin karakteri, örneklemin de dağılım karakterini etkileyen birincil unsurdur. Örneğin normal dağılım bazlı bir karaktere sahip rassal sayı üreticisi kullanıldığında, örneklemede dikkate alınan değerlerin de normal dağılım karakteri göstereceği söylenebilir. Değişkenlere ait sınır değerleri  $x_a$  alt sınır ve  $x_u$  üst sınır değer olmak üzere, bu sınırlar arasında  $n$  tane örnek noktanın rassal olarak elde edilmesi istendiğinde;

$$x_i = x_a + [x_u - x_a] \text{rand}, \quad i = 1, 2, \dots, n \quad (1.188)$$

ifadesinin kullanılması mümkündür. Bu ifadede geçen *rand* terimi, rassal sayı üreticini göstermekte olup söz konusu üretim genellikle [0,1] aralığında gerçekleştirilir.



Rassal örneklemin süreklilik arz eden değişkenlerde kullanılması yanında, ayrık düzene sahip sayı dizinlerinde de kullanılması mümkündür. Ancak ayrık konumun tanımlanması için rassal sayı üretiminin tam sayıya evrilmesi gereklidir. Bu meyanda iki dizinli  $D_{ij}$  gibi bir sayı grubu için;  $N_i$



ilk dizinin üst sınırı,  $N_j$  ikinci dizinin üst sınırı olmak üzere  $n$  tane örnek noktanın rassal örnekleme;

$$\begin{aligned} x_k &= D(i, j), & k &= 1, 2, \dots, n \\ i &= \lceil 1 + \text{rand} \times (N_i - 1) \rceil \\ j &= \lceil 1 + \text{rand} \times (N_j - 1) \rceil \end{aligned} \quad (1.189)$$

ifadesi kullanılarak yapılabilir. Burada geçen  $\lceil \cdot \rceil$  ifadesi, elde edilen gerçek sayının en yakın tam sayıya yuvarlanmasını temsil etmektedir.

### (3) Karma örnekleme

Sistemantik örnekleme yöntemleri ile beraber rastsal örnekleme yöntemlerinin aynı örnekleme sürecinde dikkate alınması halinde, belli oranda rassal ve belli oranda da sistemantik karaktere sahip değişken değeri dağılımı elde edilir. Sistemantik örneklemede dikkate alınacak yöntem daha önce bahsedilen yöntemlerden biri olabileceği gibi, çözümü istenen eniyileme sorununun karakterine göre de şekillendirilebilir. Karma örneklemede önemli olan hususlardan ilki, hangi oranda rastsallığın dikkate alınacağıdır. Genellikle yarı yarıya oransal değerler dikkate alınır. Örneğin toplam örnekleme sayısı  $n$  olan bir örnekleme sürecinde rassal örnek sayısı  $n/2$  olarak dikkate alınabilir. Bu oran bazen yüzdesel olarak %40 mertebesine de çekilebilir. Bu durumda örnekleme uzayının çoğunluğunu, sistemantik örnekleme yöntemi ile elde edilen noktalar oluşturacaktır.

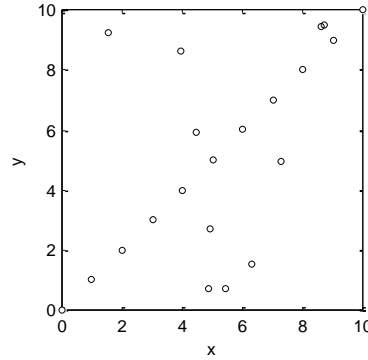
```
>>D = [1 3 5 8;2 4 1 0;8 1 0 3;7 9 6 4]
      D =
      1     3     5     8
      2     4     1     0
      8     1     0     3
      7     9     6     4
>>n = 5; Ni= 4; Nj= 4;
      col1 = randi([1 Ni],1,n);
      col2 = randi([1 Nj],1,n);
>>for i=1:n
      data(i) = D(col1(i),col2(i));
end
data ans =
      6     3     4     8     1
```



**MATLAB uygulaması**



```
>> n = 20; xa = 0; xu = 10;  
xr = xa + (xu-xa)*rand(1,n/2);  
yr = xa + (xu-xa)*rand(1,n/2);  
xs = xa:2*(xu-xa)/n:xu;  
ys = xa:2*(xu-xa)/n:xu;  
plot(xr,yr,'ko')  
hold on  
plot(xs,ys,'ko')  
axis square
```



### 1.2.5 Modelleme yöntemleri

Doğada gerçekleşen eylemlerin neden ve nasıl olduğuna dair düşüncelerimiz bizi; eylemlerin modellenmesine, ya da diğer bir anlatımla eylemlerin açıklanması için ortaya konan ilişki ve bağıntılara yönlendirir. Doğa ile ilgili en temel yaklaşımlarımızdan biri, gerçekleşen eylemler arasında mutlaka bir ilişkinin var olduğu inancıdır. Neden ve nasıl olduğu bazı hallerde açıklanamasa bile bir sistemin ve ilişkinin varlığı kabul edilir. Bu bağlamda herhangi bir eylemle ilgili olarak gerçekte var olan ilişkiler bütününe fiziksel gerçeklik<sup>i</sup> adı verilebilir. Fiziksel gerçekliğin ne olduğu bağlamında yapılan açıklama ve kurgulamaları ikiye ayırmak mümkündür. Geneli itibari ile matematiksel modelleme adı verilen yöntemlerden ilki mekanistik modelleme<sup>ii</sup>, diğeri ise gözlemsel modellemedir<sup>iii</sup>.

Mekanistik modellemede, gerçekleşen eylemin etmenleri arasındaki ilişkiler temel fizik ve kimya bilimlerinden hareketle tespit edilmeye çalışılır. Özellikle korunum yasaları olarak ifade edilen kütle, momentum ve enerjinin korunumu ilkeleri, mekanistik modellemede kullanılan en yaygın fiziksel ilkelerdir. Modelleme sonucu elde edilen ilişkiler, genellikle integral veya diferansiyel ifadeler içerir. Modelin kuruluşu, sağlam bir bilgi birikimi ve entelektüel zekâ

<sup>i</sup> Physical reality

<sup>ii</sup> Mechanistic model

<sup>iii</sup> Empirical model



gerektirir. Ayrıca kurulan modelin analitik çözümü zor ve hatta imkânsızdır. Sayısal çözümleri de genellikle zaman alıcıdır. Gerçek mühendislik sorunlarında çözüm süreci, bazen saatler ve hatta günler alabilmektedir. Kurulan modelin fiziksel gerçekliğe göre sonuçlarının doğruluğu, yapılan deney ve gözlemlerle test edilir. Gerekliğinde model, deneysel verilere göre değişikliklere uğratılır.

Gözlemsel modelleme yöntemlerinin çıkış noktası, deney ve gözlemlerdir. Mekanistik yaklaşımda gözlem ve deneyler, kurgulanan modelin doğruluğunu test etmek için kullanılırken; gözlemsel modellemede doğrudan modelin kurgulanması için kullanılır. Yaklaşımın temelinde, bir nevi eğri uydurma düşüncesi yatmaktadır. Eyleme dair eldeki gözlem ve deney sonuçlarına en yakın değerler veren fonksiyon, model olarak dikkate alınır. Bu fonksiyon, ilave gözlem ve deneylerle test edilir. Basit yapıdaki eğri uydurma yöntemleri genellikle hızlıdır ve mekanistik modele göre çok daha çabuk sonuç verir. Ancak kurgulanan model eldeki verilere göre şekillendirildiğinden, mevcut gözlem ve deneyler dışındaki değişken bölgelerinde modelin vereceği sonuçlar şüpheli olabilir.

Gözlemsel model, bazı çalışmalarda vekil modeli<sup>i</sup> olarak adlandırılır. Bu adlandırmadaki gaye, mekanistik model yerine basit ve hızlı, ancak yaklaşık ve tahmini sonuçlar veren bir model kullanımının öngörülmesidir. Eniyileme sorunlarında sürecin hızlandırılması adına vekil model kullanımı yaygın bir yaklaşım olup kitabın ilerleyen bölümlerinde söz konusu bahis detaylı bir şekilde ele alınmaktadır.

### 1.3 Eniyileme sürecinde temel kavramlar

Bir mühendislik probleminin çözümü sürecinde; öncelikle sorunun teşhisi, teşhis sonrası tanımlanması, müteakiben problemin modellenmesi ve çözüm adına uygun bir yöntemin seçilerek kullanılması gerekir. Tabiatıyla elde edilen çözümün, fiziksel gerçeklik dâhilinde gerçekten çözüm olup olmadığı da sınanması gereken diğer bir husustur<sup>(42)</sup>. Eniyilemede temel kavramlar bahsinde, standart bir optimizasyon probleminin modellenmesinde geçen terimler ve bu terimlerin türleri ifade edilmektedir. Bu bağlamda müteakip paragraflarda tasarım yöneyi, amaç fonksiyonu ile kısıt kavramları ele alınmakta ve örnekler verilerek söz konusu kavramlar açıklanmaktadır. Standart bir optimizasyon problemini Denklem (1.190)'daki gibi bir temsille modellemek mümkündür;

---

<sup>i</sup> Surrogate model



$$\min_{\mathbf{x} \in \mathbb{R}^n, \mathbf{u}(\mathbf{x}) \in \mathbb{R}^m} f(\mathbf{x}, \mathbf{u}(\mathbf{x}))$$

(1.190)

Kısıt

$$\mathbf{h}(\mathbf{x}, \mathbf{u}(\mathbf{x})) = 0$$

$$\mathbf{g}(\mathbf{x}, \mathbf{u}(\mathbf{x})) \leq 0$$

$$\mathbf{x}_L \leq \mathbf{x} \leq \mathbf{x}_U$$

$$\mathbf{G}(\mathbf{x}, \mathbf{u}(\mathbf{x})) = 0$$

Bu ifadede;  $\mathbf{x}$  tasarım yöneyini,  $\mathbf{u}(\mathbf{x})$  hal denklemlerini,  $f(\mathbf{x}, \mathbf{u}(\mathbf{x}))$  amaç fonksiyonunu,  $\mathbf{h}(\mathbf{x}, \mathbf{u}(\mathbf{x}))$  eşitlik kısıtlarını,  $\mathbf{g}(\mathbf{x}, \mathbf{u}(\mathbf{x}))$  eşitsizlik kısıtlarını,  $\mathbf{x}_L$  tasarım yöneyi dâhilindeki değişkenlerin alt sınırlarını,  $\mathbf{x}_U$  tasarım yöneyi dâhilindeki değişkenlerin üst sınırlarını ve  $\mathbf{G}(\mathbf{x}, \mathbf{u}(\mathbf{x}))$  hal denklemlerini içeren diğer eşitlik kısıtlarını temsil etmektedir. Eniyileme sürecindeki temel gaye, üzerinde çalışılan sorunun çözülmesi bağlamında en uygun tasarım yöneyinin bulunmasıdır. Ancak bu arayış yapılırken, düşünülen tasarımın öncelikli bir değerlendirme ölçütü ve ölçüte ilave olarak eldeki imkân ve kısıtların durumu dikkate alınmalıdır.

### 1.3.1 Tasarım yöneyi

Tasarım yöneyi<sup>i</sup>, tanımı yapılan sorunun çözümünde kullanılan değişkenleri içerir. Yöneyin her bir elemanı, farklı bir değişkeni temsil etmekte olup değişkenin değeri değiştikçe tasarımda iyileşme veya kötüleşme olabilir. Tasarım yöneyi, yöneysel büyüklükleri içerebileceği gibi sayısal büyüklükleri de içerebilir. Tasarım yöneyindeki değişkenin alabileceği değere göre tasnife gitmek mümkündür. Bu bağlamda tasarım parametreleri ikili sistemde<sup>ii</sup> temsil edilebilir;

$$\mathbf{x} = \begin{bmatrix} 10110 \\ 01011 \\ 11011 \end{bmatrix}$$



İkili sistem, genellikle bilgisayar bilimlerinde veya detaylarını daha sonra genetik algoritmalar bahsinde ifade edeceğimiz uygulamalarda tercih edilmektedir. Sistemin temelinde 0 ve 1 rakamları olup herhangi bir sayı, uygun dönüşümlerle sadece bu iki sembolle ifade edilebilir. Söz konusu sembollere, “bit” adı verilir.

<sup>i</sup> Design vector

<sup>ii</sup> Binary coding



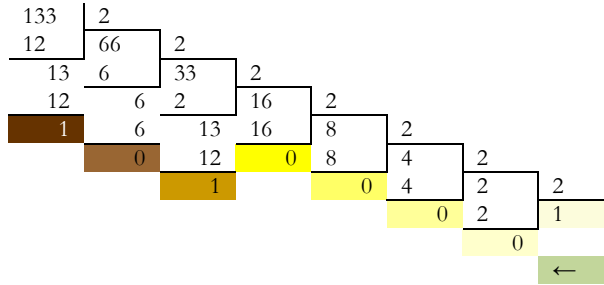
Şekil 1.59 Eldeki soruna göre tasarım yöneyinin, farklı formlarda modellenmesi uygun olabilir.



Detay açıklama



Toplamda sekiz bitin oluşturduğu sayı grubuna sekizli<sup>i</sup> adı verilmektedir. Dikkate alınan sayının ikili forma dönüştürülmesi, esasen sayının iki tabanına göre ifade edilmesidir. Genellikle sayılar 10 tabanına göre ifade edilir. Ancak sayının sürekli 2 sayısına bölünmesi ve kalanların dikkate alınması ile aynı sayı, iki tabanına göre yazılmış olur. Uygulamada tam sayıların dönüşümü oldukça kolay olup sayının büyüklüğü, temsilde ihtiyaç duyulan bit sayısını belirler. Örneğin 133 sayısını ikili formda ifade etmek için;



işlemleri uygulanır ve sonuçta;

$$(133)_{10} = (1000101)_2$$

ifadesi elde edilir. Burada dikkat edilecek husus, on tabanındaki sayının bölümler sonucu kalanlarının, ikili sistemde sağdan başlayarak yazılmasıdır. Aynı yaklaşım, ikili tabana göre ifade edilmiş bir sayının on tabanına çevrilmesi için de geçerlidir. Buna göre ikili tabandaki her bir bit, sağdan başlanarak sırası ile ikinin katları ile çarpılır ve elde edilen sayılar toplanarak on tabanındaki sayı bulunur. Örneğin 1000101 sayısını on tabanına göre ifade etmek için;

$1 \times 2^7$	$0 \times 2^6$	$0 \times 2^5$	$0 \times 2^4$	$0 \times 2^3$	$1 \times 2^2$	$0 \times 2^1$	$1 \times 2^0$
128	0	0	0	0	4	0	1
Toplam							133

işlemleri sonucunda;

$$(1000101)_2 = (133)_{10}$$

değeri elde edilebilir. Ondalık sayıları içeren gerçek sayıların ikili sistemde temsil edilmesi ise dönüşüm uygulamalarını gerektirir. Ayrıca noktadan sonraki hassasiyetin temsil edilmesi için bit sayısının da artması gereklidir. Bir ondalık sayının ikili sistemde gösterilmesine ikili kodlama, ikili koda göre ifade edilen bir sayının ondalık sayıya dönüştürülmesine ise geri kodlama adı verilir. İkili kodlama müteakip denklemlere göre yapılır;

<sup>i</sup> Byte





$$p_{norm} = \frac{p - p_a}{p_{ii} - p_a} \quad (1.191)$$

$$gen(m) = \text{yuvarla} \left[ \left( p_{norm} - \sum_{i=1}^{m-1} gen(i) 2^{-i} \right) 2^{m-1} \right] \quad (1.192)$$

Burada  $p_{norm}$  değeri normalize edilen  $p$  değerini,  $p$  simgesi dönüşüme uğrattılacak ondalık sayıyı,  $p_a$  ondalık sayının içinde bulunduğu aralığın alt değerini,  $p_{ii}$  aynı aralığın üst sınır değerini göstermektedir. Öngörülen aralığın ne olacağı, üzerinde çalışılan tasarım değişkeninin mertbe olarak bulunduğu değerlere bağlıdır. İkinci ifadede geçen  $m$  simgesi, bit numarasını göstermektedir. Örneğin dört bit ile ifade edilmesi istenen bir ondalık sayının kodlanmasında  $m$  değeri sırasıyla 1, 2, 3 ve 4 değerlerini alır. Kodlama sonucu elde edilen ikili sistem ifadesinin geri kodlama ile ondalık sisteme aynen dönüştürülmesi, fazla sayıda bit kullanımına bağlıdır. Dolayısıyla az miktardaki bit sayısı ile yapılan dönüşümlerde aynı ondalık sayının eldesi mümkün olmayabilir. Bu nedenle ikili sistemdeki ifade, yaklaşık bir değeri verecek şekilde geri kodlamaya tabi tutulabilir. Geri kodlamada dikkate alınan denklemler;

$$p_q = \left[ \sum_{i=1}^M gen(i) 2^{-i} \right] + 2^{-(M+1)} \quad (1.193)$$

$$q = p_q (p_{ii} - p_a) + p_a \quad (1.194)$$

ifadeleri olup burada  $M$  simgesi azami bit sayısını göstermektedir. Diğer simgelerden biri olan  $p_q$ , normalize edilen değerin geri kodlanmış hali,  $q$  ise geri kodlama sonucu elde edilen ondalık sayıdır.



Diğer taraftan bazı problemlerde tasarım değişkenlerinin değerinin mutlaka tam sayı<sup>i</sup> olması gerekebilir;

$$\mathbf{x} = \begin{bmatrix} 1 \\ 5 \\ 3 \end{bmatrix}$$

Örneğin bir dişli mekanizmasında diş sayısı, tam sayı olarak dikkate alınır. Tam sayılı tasarım değişkenlerine verilebilecek diğer örnekler ise uçaktaki iniş takımları sayısı, pervane pal sayıları veya tam tur devir sayılarıdır. Bununla beraber genellikle mühendislik problemlerinde karşılaşılan tasarım değişkenleri gerçek sayılardan<sup>ii</sup> oluşur;

<sup>i</sup> Integer number

<sup>ii</sup> Real numbers





$$\mathbf{x} = \begin{bmatrix} 0.25 \\ 5.48 \\ 1.22 \end{bmatrix}$$

Sıralama problemlerindeki tasarım yöneyinin öğeleri, sayısal bir değer içermekten ziyade ayrık öğe<sup>i</sup> diye tanımladığımız nesnelere ya da tanımlama bazlı isimlerden oluşabilir;

$$\mathbf{x} = \begin{bmatrix} \text{Ankara} \\ \text{Bursa} \\ \text{Samsun} \end{bmatrix}$$

Örneğin daha önce gezici satıcı sorunu olarak ifade edilen problemde şehirlerarası yolculuk eden bir satıcının hangi şehirleri hangi sıra ile ziyaret edeceği hususu, ayrık öğeler içeren bir sorundur. Zira bu sorunda şehirlerin ziyaret sırası önem kazanmakta olup sıralamanın herhangi bir sayısal anlamı yoktur. Bununla beraber sıralamaya göre mesafe ölçümü veya daha farklı değer hesaplamalarına gidilebilir. Hangi problemde ne tür bir tasarım yöneyinin dikkate alınacağı, problemin doğasına bağlıdır. Optimizasyon sürecinin ana gayesi, arzu edilen çözümü verecek tasarım değişkenlerinin değerlerini bulmaktır.

Bazı hallerde tasarım yöneyinin bir kısmı veya tümü için alt ve/veya üst sınır değerleri tanımlanabilir;

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix}, \quad \begin{array}{l} x_1^L < x_1 < x_1^U \\ x_2^L < x_2 < x_2^U \\ x_3^L < x_3 < x_3^U \\ \vdots \\ x_n^L < x_n < x_n^U \end{array}$$

Böyle bir tanımlamanın temelinde, dikkate alınması gereken fiziksel veya piyasa kaynaklı nedenler olabilir. Örneğin sıcaklık içeren bir tasarım yöneyinde metale ait sıcaklığın geçemeyeceği bir erime sıcaklık değeri, fiziksel bir zorunluluk olup sıcaklık için üst sınır değeri temsil edebilir. Ya da sistem tasarımında kullanılması düşünülen bir borunun boyutlarının tespitinde, piyasadaki boru standartlarını dikkate almak gerekebilir. Belirli sınırlar içerisinde tasarım parametrelerini kapsayan uzaya, tasarım uzayı adı verilir. Tasarım uzayı, çözümü içinde barındıran bir uzaydır.

---

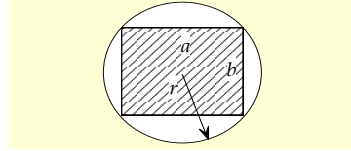
<sup>i</sup> Discrete object





Örnek uygulama

Yarıçapı  $r$  olan bir daire içerisine, alanı maksimum olacak şekilde bir dikdörtgen yerleştirilmek istenmektedir.



Şekil 1.60 Basit bir eniyileme sorunu örneği.

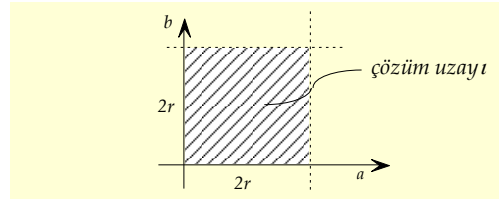
Buna göre dikdörtgen tarafından kapsanacak alanı azami derecede büyük yapabilmek için dikkate alınacak tasarım parametreleri, dikdörtgenin uzun ve kısa kenarlarıdır. Dolayısıyla tasarım yöneyi;

$$\mathbf{x} = \begin{bmatrix} a \\ b \end{bmatrix}$$

şeklinde ifade edilebilir. Yöney dâhilindeki tasarım parametrelerinin alt ve üst sınırları, daire içerisinde kalınması koşulu nedeni ile şu şekilde ifade edilebilir:

$$\begin{aligned} 0 < a < 2r \\ 0 < b < 2r \end{aligned}$$

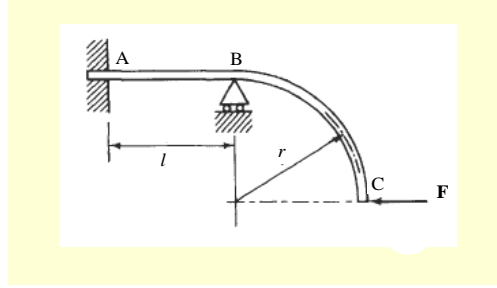
Tasarım parametrelerinin alabileceği değerleri kapsayan çözüm uzayını, müteakip şekilde göstermek mümkündür:



Şekil 1.61 İki boyutlu ortamda çözüm uzayı örneği.



Bünyesinde gerçek sayı barındıran daha karmaşık bir örnek olarak Şekil 1.62'deki gibi bir yapı problemi<sup>(43)</sup> dikkate alınabilir;



Şekil 1.62 Bünyesinde yapısal sorun barındıran eniyileme problemi örneği.

Buna göre  $A$  noktasından ankastre bir şekilde sabitlenen ve  $B$  noktasından desteklenen yarım U şeklindeki yapı,  $C$  noktasından etki eden  $F$  kuvveti nedeni ile esnemektedir. Ancak esneme miktarının, yapının emniyetli sınırını geçmesi durumunda eğri kirişte kalıcı hasar oluşabilir. Eğilmenin makul sınırlar içerisinde kalması adına tasarımda yapılabilecek değişiklikler;  $AB$  arası mesafenin ( $l$ ), dairesel büküme ait yarıçapın ( $R$ ) veya dairesel kesitli kirişin yarıçapının ( $r$ ) değiştirilmesi şeklinde olabilir. Dolayısıyla tasarım yöneyi;

$$\mathbf{x}^T = [l \quad R \quad r]$$

şeklinde ifade edilebilir. Yöney dâhilindeki tasarım parametrelerinin alt ve üst sınırları;

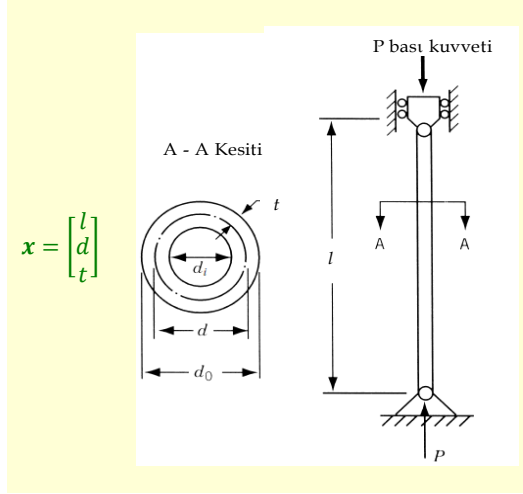
$$\begin{aligned} l^L &< l < l^U \\ R^L &< R < R^U \\ r^L &< r < r^U \end{aligned}$$

şeklinde belirtilebilir.



Yapısal bir problem olarak başka bir düzeneği ele alalım. Buna göre bası kuvvetine maruz kalan içi boş boru tipindeki bir kirişin tasarlandığını düşünelim. Söz konusu kirişin tasarımındaki temel hedefler; etki eden bası kuvvetine karşılık kirişin emniyetli bir şekilde mukavemet göstermesi ve imal edilecek kirişin ucuz olması olarak sıralanabilir. Kirişin maliyeti, harcanan malzemeye ve imalatta sarf edilen işçiliğe bağlıdır. Bu problemde tasarım parametreleri; kirişin boyu ( $l$ ), boru çapı ( $d$ ) ve et kalınlığı ( $t$ ) olarak sıralanabilir.





Şekil 1.63 Diğer bir yapısal eniyeleme sorunu örneği.

Yöneý dâhilindeki tasarım parametrelerinin alt ve üst sınırları, genellikle piyasada bulunan hazır mamul ürünlerin özelliklerine göre ifade edilebilir:

$$\begin{aligned} l^L &< l < l^U \\ d^L &< d < d^U \\ t^L &< t < t^U \end{aligned}$$

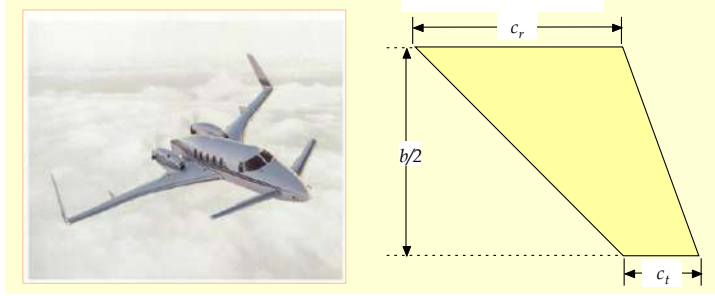


Havacılıkla ilgili örnek bir problemde, hava aracının havada tutunabilmesi için hayati öneme sahip kanat tasarımı düşünülebilir. Hava aracına ait taşıma kuvvetini veren ifade;

$$L = C_L \frac{1}{2} \rho_\infty V_\infty^2 S$$

denklemlerle tanımlanır. Burada  $C_L$  taşıma katsayısını,  $\rho_\infty$  serbest hava akımı yoğunluğunu,  $V_\infty$  serbest hava akımı hızını ve  $S$  kanat alanını ifade eder. Kanat alanını, müteakip şekilde belirtilen temel parametrelerle tanımlamak mümkündür.





**Şekil 1.64** Basit bir uçak<sup>(44)</sup> kanadına ait tasarım parametreleri örneği.

Burada  $c_r$  kök veteri,  $c_t$  uç veteri ve  $b$  kanat açıklığını temsil etmektedir. Mevcut geometriye göre taşıma kuvveti tekrar ifade edilmek istenirse;

$$L = C_L \frac{1}{2} \rho_\infty V_\infty^2 \left[ b \frac{c_r}{2} \left( 1 + \frac{c_t}{c_r} \right) \right]$$

denkleminde ulaşılabilir. Buna göre taşıma kuvvetinin artırılması bağlamında mevcut geometrinin tasarım yöneyini ve dolayısıyla parametrelerini;

$$\mathbf{x} = \begin{bmatrix} b \\ c_r \\ c_t \end{bmatrix}$$

şeklinde tarif etmek mümkündür.

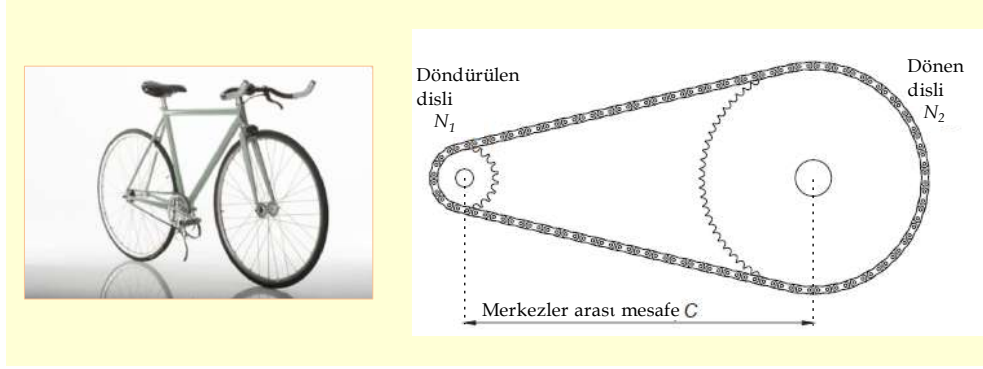


Birbiri ile zincir bağlantısı olan bir dişli mekanizması düşünelim. Bu tür bir zincir mekanizmasında zincir uzunluğu, zincir adım sayısına bağlı olarak<sup>(45)</sup>;

$$L = \frac{N_1 + N_2}{2} p + \frac{N_2 - N_1}{2\pi} \frac{p}{C}$$

denkleminde ifade edilir Burada  $N_1$  ve  $N_2$  sırasıyla döndürülen ve dönen dişliye ait diş sayıları,  $C$  dişli merkezleri arasındaki mesafe ve  $p$  zincir adım mesafesidir.





Şekil 1.65 Dişli-zincir mekanizmasına ait eniyileme sorunu örneği.

Tasarım parametrelerini, denklemde geçen değişkenlere bağlı olarak;

$$\mathbf{x} = \begin{bmatrix} p \\ C \\ N_1 \\ N_2 \end{bmatrix}$$

yöneyi ile tanımlamak mümkündür. Yöneye ait ilk iki tasarım parametresinin değeri, gerçek sayı niteliğindedir. Son iki tasarım parametresi;  $N_1$  ve  $N_2$  dişlilere ait diş sayıları olup uygulamada mutlaka tam sayı değeri almalıdırlar.



Fransa ve Almanya'da bulunan beş farklı şehrin ziyaret edilmesi istenmektedir. Ziyaretin en kısa zamanda gerçekleşmesi için dikkate alınan ölçüt ise kat edilecek yolun kısa olmasıdır. Bu bağlamda şehirlerin ziyaret sırası önem kazanmakta olup tasarım yöneyinin elemanlarını şehirler oluşturacaktır. Sorunun çözümü için dikkate alınabilecek örnek bir tasarım yöneyi;

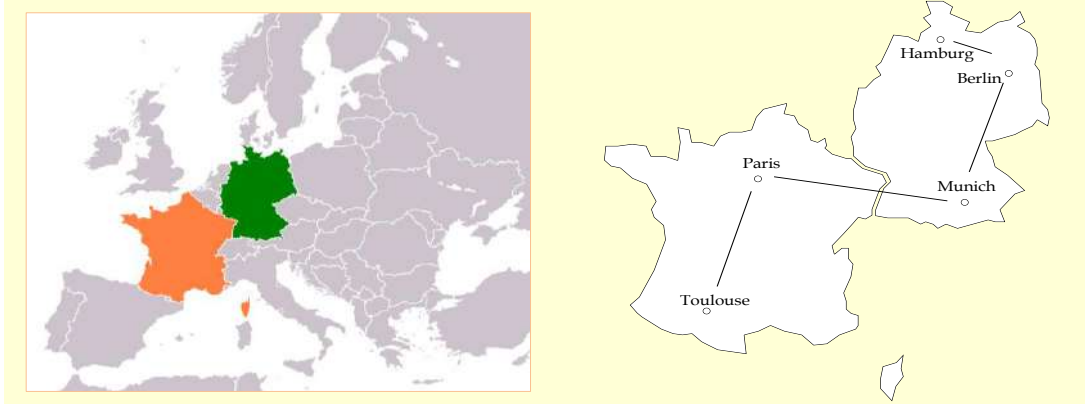
$$\mathbf{x}^T = [Hamburg \ Berlin \ Munich \ Paris \ Toulouse]$$

şeklinde tanımlanabilir. Ayrık yöney niteliğindeki böyle bir tasarım yöneyi için alt ve üst sınırlardan bahsetmek zordur. Bununla beraber tasarım parametrelerinin alabileceği ihtimalleri kapsayan çözüm uzayının büyüklüğü;

$$n!$$



ifadesi ile belirlemek mümkündür Burada  $n$  değeri, şehir sayısını ifade etmektedir. Verilen örnekteki gibi beş şehir dikkate alındığında, muhtemel ziyaret planı sayısı  $5! = 120$  olmaktadır. Şehir sayısının artması durumunda, olasılıklar ciddi miktarda artış gösterebilmekte, bu da sorunun çözümünün makul zamanda bulunmasını engelleyebilmektedir.



Şekil 1.66 Sıralama sorununa örnek bir şehirlerarası ziyaret problemi.

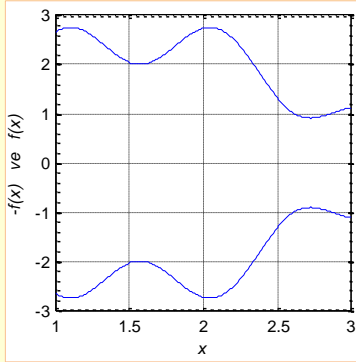


### 1.3.2 Amaç fonksiyonu

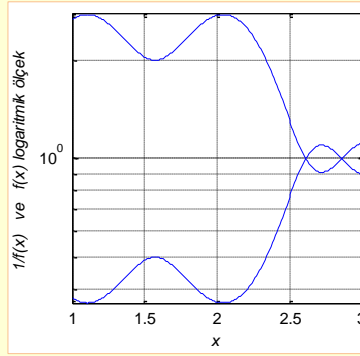
Amaç fonksiyonu<sup>i</sup>, eniyileme sürecinin gidişatını belirleyen en önemli unsur olup problemin tanımlanmasında çok önemli bir yere sahiptir. Tasarım sürecinde ana gaye, istenen işlevsel ve fiziksel özelliklere sahip uygun bir tasarımın elde edilmesidir. Aynı gayeye hizmet eden farklı tasarımları geliştirmek ve bir çözüm olarak değerlendirmek mümkündür. Ancak hangi tasarımın daha iyi olduğunu veya diğer bir ifade ile daha ekonomik, etkin ve güvenilir olduğunu tespit etmek, nesnel ölçütler gerektirir. Tasarımlar arasında ayırt ediciliği içeren ve nesnel ölçütlere bağlı söz konusu tanıma, amaç fonksiyonu adı verilir. Amaç fonksiyonu, problemin doğasına göre tanımlanır. Optimizasyon sürecinde amaç fonksiyonu, maksimize veya minimize edilmeye çalışılır. Standart eniyileme probleminde bu tanım, tamamen minimizasyonu temel alır. Dolayısıyla amaç fonksiyonu, doruklaştırma olarak dikkate alınsa da, uygun bir yöntemle onun standart şekle dönüştürülmesi mümkündür.

<sup>i</sup> Objective function”, “cost function” veya “merit function”





(a)



(b)

**Şekil 1.67** Doruklaştırma problemi olarak tanımlanan sorunun; (a) işaret değiştirme, (b) paydaya alma yaklaşımı ile minimizasyon problemine dönüştürülmesi,  $(f(x) = 2 \sin(x) + \cos(3x))^2$ .

Söz konusu yöntemlerden ilki, amaç fonksiyonunun işaretinin değiştirilmesidir;

$$\max\{f(x)\} \rightarrow \min\{-f(x)\} \quad (1.195)$$

Bu sayede amaç fonksiyonunun değeri arttıkça eniyileme sürecini yönlendiren ifade küçülecektir. Diğer bir yaklaşım ise amaç fonksiyonunun paydaya alınması şeklindedir;

$$\max\{f(x)\} \rightarrow \min\left\{\frac{1}{f(x)}\right\} \quad (1.196)$$

Paydada değeri büyüyen amaç fonksiyonu, eniyileme sürecinde minimizasyonu tetiklemektedir. Şekil 1.67'de örnek bir fonksiyonun grafik çizimleri görülmektedir. Buna göre fonksiyona ait doruk noktaları, dönüşüm sonrasında minimum noktalar olarak kendini göstermektedir.

Bazı eniyileme süreçlerinde birden fazla gaye güdülebilir. Buradaki yaklaşımın temelinde, farklı disiplinlerdeki beklentiler olabileceği gibi aynı disiplin altında farklı amaçlar da olabilir. Bununla beraber farklı amaçları aynı çatı altında toparlayarak tek ama genel bir amaç fonksiyonu tanımlamak mümkündür. Bu tür eniyileme problemlerine, çok-amaçlı optimizasyon problemleri adı verilir. Bünyesinde  $n$  tane farklı amaç taşıyan çok-amaçlı eniyileme problemine ait genel yaklaşım;

$$\min\{f_1(x), f_2(x), \dots, f_i(x), \dots, f_n(x)\} \quad (1.197)$$





şeklinde yazılabilir. Tabiatıyla aynı anda tüm amaçların gerçekleştirilmesi mümkün olmayabilir. Bazı amaçların elde edilmesinde iyileşme gözlenirken, bazı amaçlarda gerileme görülebilir. Dolayısıyla her bir amacın durumundan ziyade genel olarak elde edilen faydaya bakmak daha gerçekçi bir yaklaşım olacaktır. Bu bağlamda amaçların hepsini kapsayacak şekilde tek bir genel amaç fonksiyonu;

$$\min F(x), \quad F(x) = \sum_{i=1}^n w_i f_i(x) \quad (1.198)$$

biçiminde yazılabilir. Görüldüğü üzere bu yaklaşımda, amaçların doğrusal bir birleşimi dikkate alınmaktadır. Bu ifadede geçen  $w_i$  katsayısı, ağırlık katsayısı olup her bir amaca verilen önem derecesini göstermektedir. Zorunlu olmamakla beraber ağırlık katsayıları arasında;

$$\sum_{i=1}^n w_i = 1$$

ilişkisi gözetilebilir. Ancak mutlaka dikkate alınması gereken husus,  $w_i > 0$  şartıdır. Aksi halde amaçlardan biri gözden çıkarılmış demektir.



**Şekil 1.68** Üç parçalı kanat profili örneği<sup>(46)</sup>.

Bir yolcu uçağının kanat tasarımı, neredeyse tüm uçağın tasarımını etkileyen bir husus olup en kritik tasarım süreçlerinden birisidir. Üç boyutlu bir kavram olan kanadın tasarımı, büyük oranda kanat profili denilen ve iki boyutlu bir kavram olan profil şekline bağlıdır. Genellikle yolcu uçaklarında tercih edilen kanat profili, üç parçalı profil geometrisidir. Profil geometrisinden beklenen iki temel husus vardır. Bunlardan ilki, azami taşıma kuvvetini ( $L$ ) üretmesi, ikincisi ise asgari sürüklenme kuvvetine ( $D$ ) neden olmasıdır. Bu iki amacın beraber dikkate alınması genellikle;

$$\min f = 1 / \left( \frac{L}{D} \right)$$

ifadesi ile sağlanır.



Örnek Uygulama





Piyasada üretimi yapılan  $A$  ve  $B$  gibi iki farklı ticari mal olduğu düşünölsün. Üretim sonrası satılan her  $A$  malından  $6 TL$ , her  $B$  malından  $5 TL$  kâr elde ediliyor olsun. Üretim yapan şirketin ana gayesi, kâr elde etmek olduğuna göre, kârın maksimizasyonu sorunu için müteakip amaç fonksiyonunu yazmak mümkündür;

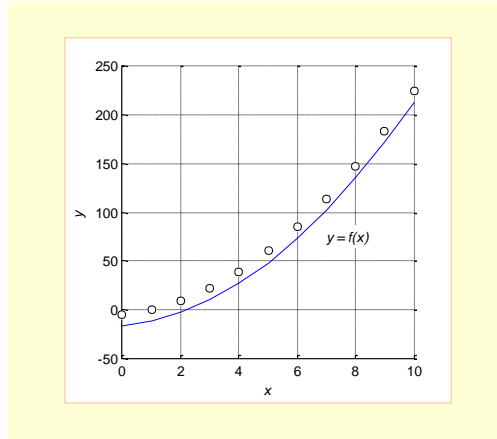
$$\min f = -(6x + 5y)$$

Burada  $x$  ifadesi  $A$  malının üretim miktarını,  $y$  ifadesi  $B$  malının üretim miktarını göstermektedir.



Doğâ bilimlerinde çok sıkça karşılaşılan problemlerden birisi de eğri uydurma<sup>1</sup> uygulamasıdır. Bu tür problemlerde genellikle benzetime veya gözleme dayalı veriler var olup eğri uydurmada ana gaye, bu verilerden geçen analitik denklemi elde etmektir. Örnek olması amacıyla Şekil 1.70’te veri değerleri (siyah dairesel noktalar) ve analitik olarak bilinen bir fonksiyonun çizimi görölmektedir.

$x$	0	1	2	3	4	5	6	7	8	9	10
$y = f(x)$	-17	-12	-3	10	27	48	73	102	135	172	213



Şekil 1.70 Hedef veri noktaları ve örnek eğri.

Eldeki verilerden geçen ikinci dereceden bir polinomiyal denklem dikkate alınmak istendiğinde;

<sup>1</sup> Curve fitting



Şekil 1.69 Ticarete amaç, kârın doruklaştırılmasıdır.





$$\hat{f}(x) = ax^2 + bx + c$$

ifadesi yazılabilir. Bu denklemde bilinmeyen terimler, değişken katsayıları olup buna göre tasarım parametreleri:

$$\mathbf{t} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

yöneyle temsil edilebilir. Buna mukabil, sorundaki ana gayenin, var olan noktalardan geçen en yakın polinomialin bulunması olması nedeniyle amaç fonksiyonu;

$$\min f(x, \mathbf{t}) = \sum (f(x, \mathbf{t}) - \hat{f}(x, \mathbf{t}))^2$$

denklemi ile ifade edilir. Burada  $f(x, \mathbf{t})$ ,  $x$  koordinatına karşılık gelen eldeki verinin değeri,  $\hat{f}(x, \mathbf{t})$  ise uydurulan eğrinin aynı  $x$  koordinatına karşılık gelen değeridir.



### 1.3.3 Tasarım kısıtları

Tasarım kısıtları<sup>i</sup> kavramını, çözüm olarak dikkate alınan parametreler tarafından sağlanması gereken koşullar olarak tanımlayabiliriz. Tasarım kısıtları kavramı tasarım gerekleri<sup>ii</sup> ile karıştırılmamalıdır. Tasarım gerekleri, tasarımın yapılması esnasında dikkate alınan ve tasarımdan beklenen işlevleri ve özellikleri ifade eder. Önerilen bir tasarım, kendisinden beklenen tüm işlevsel ve fiziksel özellikleri taşıyor olabilir, ancak tasarım kısıtlarını karşılamadığı durumda eldeki çözüm, olası bir çözüm<sup>iii</sup> olarak görülmez. Daha önce tasarım yöneyi bahsinde parametrelere ait alt ve üst sınırların olabileceğinden bahsedilmişti. Söz konusu sınırlar esasen tekil olarak o tasarım parametresine ait kısıtları ifade etmektedir. Bireysel sınırlar haricinde tasarım parametrelerinin kendi aralarında da bazı koşulları sağlaması gerekebilir. Bu nedenle tasarım parametrelerini rastgele seçmek mümkün olmayıp, parametrelerin bazı şartları, diğer bir ifade ile kısıtları sağlaması gerekebilir. Eniyileme süreçlerinde söz konusu kısıtları dört grupta tasnif etmek mümkündür. İlk kısıt grubu eşitlik kısıtları<sup>iv</sup> olarak tanımlanmakta olup bir ya da

<sup>i</sup> Design constraints

<sup>ii</sup> Design requirements

<sup>iii</sup> Feasible solution

<sup>iv</sup> Equality constraints



daha fazla tasarım parametresinin arasında müteakip eşitliğin sağlanması gereğini ifade etmektedir;

$$h_i(\mathbf{x}, \mathbf{u}(\mathbf{x})) = 0, \quad i = 1, 2, \dots, p$$

Burada görüldüğü üzere, ifade genel olup birden fazla tasarım parametresi yanında birden fazla,  $p$  tane eşitlik kısıtının varlığı söz konusudur. İkinci kısıt grubu eşitsizlik kısıtları olarak adlandırılır ve müteakip eşitsizliğin sağlanması gereğini ifade eder;

$$g_j(\mathbf{x}, \mathbf{u}(\mathbf{x})) \leq 0, \quad j = 1, 2, \dots, m$$

Bu ifade, doğasında eşitlik ilkesini de barındırmaktadır. Ancak kısıt tanımında eşitlik olması şart olmayıp, genellikle küçük veya büyük şeklinde kısıtlar kullanılır. Üçüncü kısıt grubu, tamsayı olma zorunluluğunu ifade etmektedir. Buna göre tasarım parametrelerinden bazılarının veya tümünün müteakip kısıtı sağlanması şart koşulabilir;

$$\mathbf{x} \in \mathbb{Z}^n$$

Bu ifadede  $\mathbb{Z}^n$ ,  $n$  boyutlu uzayda tam sayılar kümesini temsil etmektedir. Son kısıt grubu ise standart değer olma koşulunu ifade etmektedir. Tasarım aşamasında her ne kadar tasarım gerekleri süreci yönlendirse de bazı tasarım parametrelerinin, piyasa koşulları veya standartları nedeni ile belirli değerler alması gerekebilir. Böyle bir durumda tasarım sonucu elde edilen parametre değerinden ziyade bu değere en yakın standart değer ya da ürün dikkate alınabilir. Söz konusu kısıt;

$$\mathbf{x} \in S^n$$

ifadesi ile temsil edilebilir. Burada  $S^n$ ,  $n$  boyutlu uzayda standart değerler kümesini göstermektedir.

Tasarım kısıtlarını, niteliğine göre de sınıflandırmak mümkündür. Örneğin kısıt, tasarımın işlevselliği veya performansı ile ilgili ise işlevsel kısıt<sup>i</sup> olarak adlandırılır. Ya da kısıt, tasarımın geometrik gerekleri nedeniyle tanımlanmaktaysa geometrik kısıt<sup>iii</sup> olarak ifade edilir.

---

<sup>i</sup> Inequality constraints

<sup>ii</sup> Functional constraint

<sup>iii</sup> Geometrical constraint



Kısıt bahsinde ifade edebileceğimiz diğer bir husus, kısıt denklemlerinin ilave bir özellik daha dikkate alınarak ifade edilebileceği konusudur. Bazı problemlerde kısıt denklemleri, eşitlik veya eşitsizlik içerse de nitelik olarak diferansiyel veya integral formda olabilir. Bu nedenle diferansiyel ya da integral denklem tipinde eşitlik veya eşitsizlik kısıtı olarak bir tanımlamaya gidilebilir. Diferansiyel denklem tipindeki örnek bir eşitlik kısıtını;

$$h_i(x, x', x'', \dots, x^{(k)}, u(x)) = 0, \quad i = 1, 2, \dots, p$$

şeklinde ifade edebiliriz. Burada  $k$  değeri, o mertebedeki türevi göstermektedir. Buna mukabil integral formdaki örnek bir eşitsizlik kısıtını;

$$\int_{x_b}^{x_s} h_j(x, x', x'', \dots, x^{(k)}, u(x)) = 0, \quad j = 1, 2, \dots, m$$

denklemleri ile gösterebiliriz. Tabiiyatıyla integral denklemin iki veya daha fazla katlı integral formu içermesi de mümkün olabilir.



Matematiksel olarak müteakip problemi tanımlayalım;

$$\min f = \frac{1}{y}$$

Kısıt

$$\begin{aligned} -x + y &= 1 \\ 3x + 2y &\leq 12 \\ 2x + 3y &\leq 12 \\ x, y &> 0 \\ x, y &\in \mathbb{Z} \end{aligned}$$



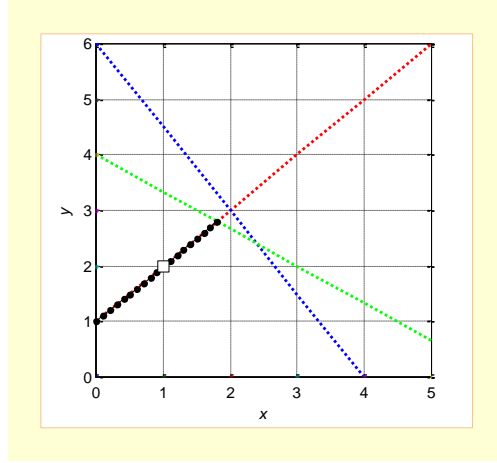
Örnek Uygulama

Görüldüğü üzere problemde  $y$  değerinin belli şartlar altında doruklaştırılması hedeflenmektedir. Kısıtlardan ilki eşitlik kısıtı, sonraki ikisi küçük ve eşitsizlik kısıtı, dördüncüsü eşitsizlik kısıtı ve son kısıt ise tamsayı kısıtıdır. Kısıtların çözüme etkisini görmek için problemi şekilsel olarak değerlendirmek faydalı olabilir. Bu meyanda Şekil 1.71'de verili kısıtlar görülmektedir. Buna göre kırmızı noktalar ile gösterilen doğru,  $y = 1 + x$  denklemi olup ilk kısıtı ifade etmektedir. Kısıt, eşitlik kısıtı olduğu için de çözüm olarak bulunan  $y$  değerinin mutlaka bu doğru üzerinde olması gerekmektedir. Mavi noktalar ile gösterilen doğru,  $y = 6 - 3/2x$  denklemi olup ikinci kısıtı göstermektedir. Kısıt, küçük ifadesini taşıdığı için çözümün bu doğrunun altında kalması gerekir. Yeşil noktaları kapsayan doğru, üçüncü kısıt niteliğindeki  $y = 4 - 2/3x$





doğrusu olup bir öncekinde olduğu gibi küçük ifadesi taşımaktadır. Dolayısıyla çözümün mutlaka bu doğrunun altında kalması gerekir. Siyah renkli yatay ve dikey eksen doğruları,  $x = 0$  ve  $y = 0$  doğruları olup büyüklük kısıtı söz konusudur. Bu nedenle çözüm bu iki doğru değerinden büyük olmalıdır. Son kısıt olan tamsayı olma zorunluluğu ise çözüm noktasındaki  $x$  ve  $y$  değerinin tam sayı olması gereğini ifade eder. Tüm kısıtların sağlandığı yegâne noktanın sadece  $[1,2]$  noktası olduğu görülmektedir. Bu nedenle  $y$  fonksiyonunun alabileceği azami doruk değer 2'dir.



Şekil 1.71 Kısıt doğruları ve en uygun çözüm.



$A$  ve  $B$  gibi iki ticari mal düşünelim.  $A$  maddesinden 1 kg üretebilmek için 0.5 kg  $C$  ve 0.6 kg  $D$  maddesi gereksin. Benzer şekilde  $B$  maddesinden 1 kg üretmek için 0.65 kg  $C$  ve 0.35 kg  $D$  maddesine ihtiyaç duyulsun. Buna mukabil üretimde kullanılan  $C$  ve  $D$  madde miktarları depoda kısıtlı miktarda olsun ve buna göre  $C$  maddesinden 100 kg,  $D$  maddesinden 90 kg bulunsun. Ayrıca, üretilen  $A$  ve  $B$  mallarının piyasada kâr getirebilmesi için  $A$  malından en az 110 adet,  $B$  malından ise en az 40 adet üretme zorunluluğu olsun. Verilen kısıtların matematiksel ifadelerle dönüştürülebilmesi için öncelikle  $A$  malından üretilen miktarı  $x$ ,  $B$  malından üretilen miktarı  $y$  olarak tanımlayalım. Buna göre depoda kısıtlı miktarda bulunan hammaddeler dikkate alınarak aşağıdaki ifadeler yazılabilir;

$$\begin{aligned} 0.5x + 0.65y &\leq 100 \\ 0.6x + 0.35y &\leq 90 \end{aligned}$$





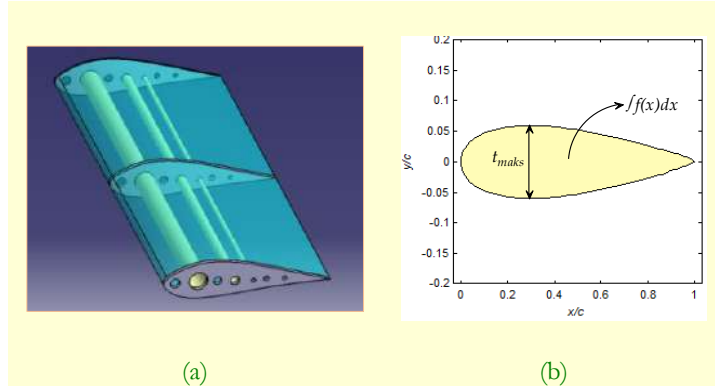
İlave olarak pazarın gerektirdiği asgari üretim miktarlarını da dikkate aldığımızda;

$$\begin{aligned}x &\geq 110 \\y &\geq 40\end{aligned}$$

kısıtlarını yazmak mümkündür.



Şekil 1.72 Örnek depo sistemi ve düzenlemesi.



Şekil 1.73 (a) Örnek kanat tasarımı ve (b) kanat profiline ait kısıt tanımları.

Kanat profiline optimizasyonu kapsamında genellikle taşıma katsayısının artırılması, sürüklenme katsayısının ise azaltılması hedeflenir. Ancak bu işlem esnasında, özellikle yapısal nedenlerden dolayı azami profil kalınlığının belli bir değerin altına inmemesi istenir. İlave olarak, hacimsel kaygılardan dolayı da





profil kesit alanının belli bir limit değer üzerinde tutulması hedeflenebilir. Zira kanat iç hacmi, yakıt deposu olarak kullanılmak istenebilir. Bu meyanda kalınlık ve hacim kısıtları aşağıdaki gibi ifade edilebilir;

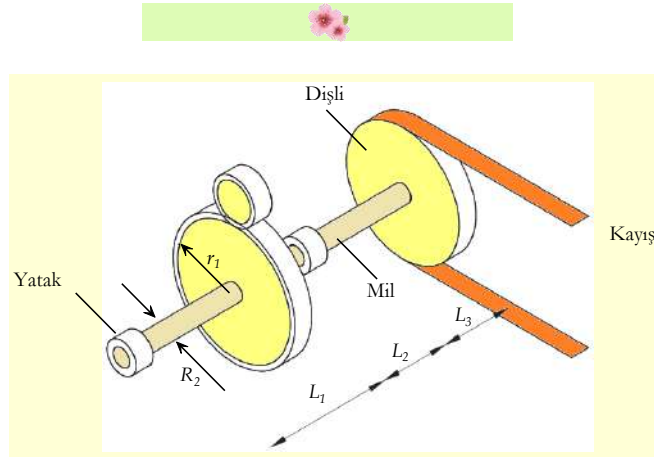
$$t_{max} \leq 0.15c$$

$$\int_0^c f(x)dx \leq A^*$$

Burada  $c$  ifadesi profil veter boyunu,  $f(x)$  fonksiyonu profil eğri denklemini ve  $A^*$  tasarım kesit alan değerini ifade etmektedir. İfade edilen kısıtlara ilave olarak bazen taşıma katsayısını da kapsayan yeni bir kısıt kullanmak mümkündür. Özellikle taşıma ve sürüklenme katsayı oranlarının dikkate alındığı amaç fonksiyonu tanımlamaları, taşıma katsayısının azalması riskini de beraberinde getirebilir. Bu nedenle aşağıdaki gibi bir kısıt genellikle dikkate alınır;

$$C_{L,max} \geq C_{L,max}^*$$

Burada  $C_{L,max}^*$  ifadesi, tasarım azami taşıma katsayısı değerini göstermektedir.



Şekil 1.74 Örnek dişli-kayış mekanizması ve parametreleri.

Dişli, yatak ve kayış mekanizmaları, makine aksamında sıkça kullanılan elemanlar olup tasarımda genellikle kayış tarafından veya kayışa aktarılan moment miktarı girdi verisi olarak dikkate alınır. Aktarılan momenti etkileyen unsurlar; dişli ve yataklar arası mesafeler, milin ve dişlilerin yarıçapı gibi verilerdir.







Diğer taraftan özellikle milin üretildiği malzemeye ait mukavemet değerleri, tasarımı doğrudan şekillendiren bir husustur. Buna göre mekanizmaya ait ilk kısıt olarak;

$$\tau \leq \tau_{emn}$$

ifadesi yazılabilir. Burada  $\tau$  milde meydana gelen kayma gerilmesi değeri,  $\tau_{emn}$  ise mil malzemesine bağlı olarak ifade edilen azami kayma gerilmesi değeridir. Belirtilen bu kısıt, özellikle milin çapını temsil eden  $R_2$  parametresini şekillendirmektedir. Piyasada standart mil değerlerinin olduğu düşünüldüğünde aşağıdaki gibi bir kısıtı yazmak mümkündür;

$$R_2 \in S_{mil}$$

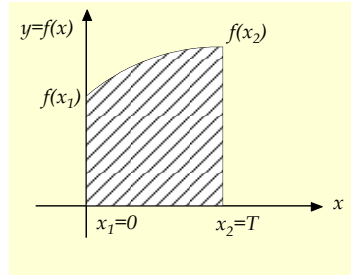
Burada  $S_{mil}$  piyasada satılan standart mil çap değerlerini içeren kümedir. Diğer taraftan tasarımda dişli ve yataklar arası mesafeler de sınırlandırılabilir. Yerleşim kaynaklı nedenlerden dolayı;

$$L_i^L \leq L_i \leq L_i^U, \quad i = 1,2,3$$

şeklindeki kısıtları göz önünde bulundurmak mümkündür.



Herhangi bir  $f(x)$  fonksiyonunun  $x$  eksenini dâhilindeki  $[0, T]$  aralığında kapsadığı alanı doruklaştıran  $f(x)$  fonksiyon denklemini bulmak isteyelim. Bu durumda sorunu kısıtlayan husus,  $f(x)$  fonksiyonuna ait eğri mesafenin  $k$  gibi bir sabite eşit olması gereğidir. Sorunun daha açık halini: “uzunluğu sabit olan bir eğri nasıl olmalı ki altında kalan alan maksimum olsun...” sorusu ile belirtebiliriz.



**Şekil 1.75**  $[0, T]$  aralığında tanımlı  $f(x)$  fonksiyonu ve eğrinin altında kalan alan.





Sorunu matematiksel olarak;

$$\text{maks } f(x) = \int_0^T y dx$$

Kısıt

$$\int_0^T (1 + (y')^2)^{\frac{1}{2}} dx = k$$

$$y(0) = f(0)$$

$$y(T) = f(T)$$

şeklinde tanımlayabiliriz. Görüldüğü gibi kısıtlardan ilki integral formda tanımlanmış bir eşitlik kısıtını göstermekte olup fonksiyon eğrisinin uzunluğunun  $k$  değerine eşit olmasını ifade etmektedir. İlave olarak kısıtın, bünyesinde diferansiyel ifadeler de içerdiğini belirtelim.



### 1.3.4 Hal denklemleri

Eniyileme sürecinde hal denklemleri, kısıt harici bir kavram olup özellikle fiziksel yasaların dikkate alındığı problem çözümlerinde önem arz eder. Tasarım için önerilen çözümün, öngörülen koşulları sağlaması yanında temel fiziksel kanunlara da uygun olması gerekir. Genellikle korunum yasaları bu meyanda dikkate alınan hal denklemlerini içerir. Yaygın olarak dikkate alınan korunum yasaları arasında kütle, momentum ve enerjinin korunumu yasaları sayılabilir.



Örnek Uygulama

Bir kanat profiline ait optimizasyon probleminde genellikle müteakip amaç fonksiyonu tanımlanır;

$$\text{min } f = 1 / (C_L / C_D)$$

Kısıt

$$t = t^*$$

$$C_L \geq C_L^*$$

<sup>i</sup> State equations



Burada  $C_L$  ve  $C_D$  sırasıyla profilden elde edilen taşıma ve sürüklenme katsayısını,  $t$  azami profil kalınlığını,  $t^*$  tasarım azami profil kalınlığını,  $C_L^*$  ise tasarım taşıma katsayısını ifade etmektedir. Gerek taşıma ve gerekse sürüklenme katsayısının değişimi büyük oranda profil şekline, diğer bir ifade ile iki boyutlu ortamda profile ait noktaların  $(x, y)$  geometrisine bağlıdır. Dolayısıyla eniyileme sürecinde ana gaye, dikkate alınan koşullar altında asgari amaç fonksiyon değerini veren geometriyi elde etmektir. Diğer taraftan  $C_L$  taşıma katsayısı, genellikle profil etrafındaki basınç katsayısı dağılıma bağlı olarak hesaplanır. Buna göre;

$$C_L = \int_0^1 (C_{p,l}(x) - C_{p,u}(x)) d \frac{x}{c}$$

yazılabilir. Burada  $C_{p,l}$  profil alt yüzeyine ait basınç katsayısını,  $C_{p,u}$  profil üst yüzeyine ait basınç katsayısını temsil etmektedir. Basınç katsayısı kavramı, serbest akım özellikleri ile profil etrafındaki statik basınç değerine bağlı olarak elde edilir. Söz konusu değer;

$$C_p = \frac{p - p_\infty}{\frac{1}{2} \rho_\infty V_\infty^2}$$

denklemi ile hesaplanır. Burada  $p$  profil üzerindeki statik basınç değerini;  $p_\infty$ ,  $\rho_\infty$  ve  $V_\infty$  ise sırasıyla serbest hava akımı basıncını, yoğunluğunu ve hızını temsil etmektedir. Hesaplama ana girdi niteliğindeki  $p$  statik basınç değeri, akışkanlar mekaniğine ait korunum yasalarından hareketle elde edilir. Buna göre oldukça basitleştirilmiş bir yaklaşımla iki boyutlu ortamda kütle ve momentumun korunumunu;

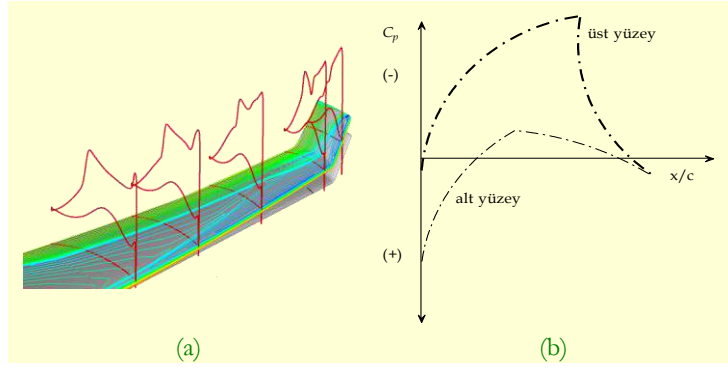
$$\begin{aligned} \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} &= 0 \\ u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} &= -\frac{1}{\rho} \frac{\partial p}{\partial x} \\ u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} &= -\frac{1}{\rho} \frac{\partial p}{\partial y} \end{aligned}$$

denklemleri ile ifade etmek mümkündür. Burada  $u$  ve  $v$  ifadeleri akışkan hızının bileşenleri olup;

$$\begin{aligned} u &= \frac{\partial x}{\partial t} \\ v &= \frac{\partial y}{\partial t} \end{aligned}$$

olarak yazılabilir. Dolayısıyla eniyileme sürecinde elde edilen geometrinin, hal denklemleri olarak adlandırılan ve bu örnekte korunum yasaları adı altında belirtilen yasaları da sağlaması gerekir.





Şekil 1.76 (a) Örnek kanat ve (b) kanat profiline ait basınç katsayısı dağılımları<sup>(47)</sup>.



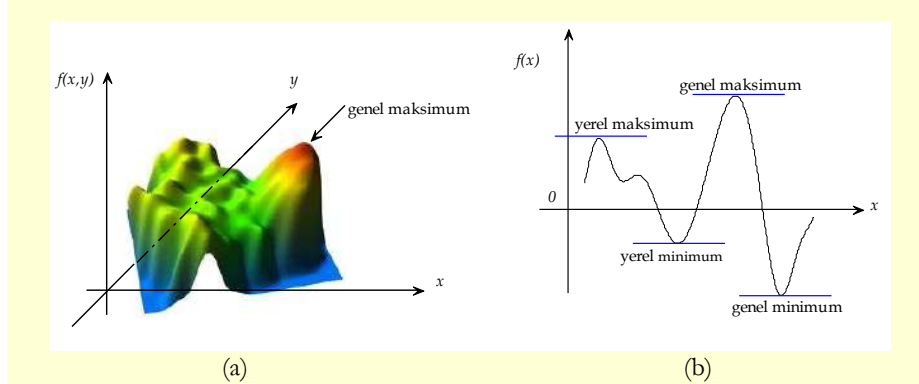
### 1.3.5 Optimal değer tanımı

Daha önce kümede ve bilahare kümelerde sıralama bahsi üzerinde durulmuş, kümeye ait azami ya da asgari değer kavramları açıklanmıştı. Benzer şekilde birden fazla kümenin olduğu birleşme, kesişme veya kapsama gibi hallerde de aynı uç değerlerin nasıl tanımlandığı izah edilmişti. Bu bölümde aynı husus daha somut olarak tekrar ele alınmaktadır. Zira küme kavramı yerine, artık amaç fonksiyonu tanımı dikkate alınabilir. Tanımı gereği amaç fonksiyonu, tasarım parametrelerine bağlı bir fonksiyon olup fonksiyonun farklı mertebelerde asgari veya azami değerleri olabilir. Eğer amaç fonksiyonunun değeri, belli bir  $\mathbf{x}^*$  tasarım yöneyi değerinin çevresinde  $f(\mathbf{x}^*) < f(\mathbf{x})$  şartını sağlıyor ise söz konusu  $\mathbf{x}^*$  yöneyine yerel minimum tasarım yöneyi adı verilir. Buna mukabil eğer amaç fonksiyonunun değeri, belli bir  $\mathbf{x}^*$  tasarım yöneyi değerinin çevresinde  $f(\mathbf{x}^*) > f(\mathbf{x})$  şartını sağlıyor ise söz konusu  $\mathbf{x}^*$  yöneyine yerel maksimum tasarım yöneyi adı verilir. Diğer taraftan amaç fonksiyonunun değeri sadece belli bir bölgede değil de tüm  $\mathbf{x}$  değerleri için  $f(\mathbf{x}^*) < f(\mathbf{x})$  şartını sağlıyorsa  $\mathbf{x}^*$  yöneyine genel minimum tasarım yöneyi adı verilir. Benzer şekilde, amaç fonksiyonunun değeri tüm  $\mathbf{x}$  değerleri için  $f(\mathbf{x}^*) > f(\mathbf{x})$  şartını sağlıyorsa  $\mathbf{x}^*$  yöneyine genel maksimum tasarım yöneyi adı verilir.

Eniyileme sürecinde güdülen amaca göre yerel minimum veya yerel maksimum noktası yerel optimal nokta olarak adlandırılır. Aynı yaklaşımla genel optimum değer, amaca göre genel minimum veya genel maksimum nokta olabilir. İster yerel veya ister genel optimal nokta olsun, böyle noktalarda amaç fonksiyonunun türev değeri sıfırdır. Zira söz konusu noktada fonksiyon

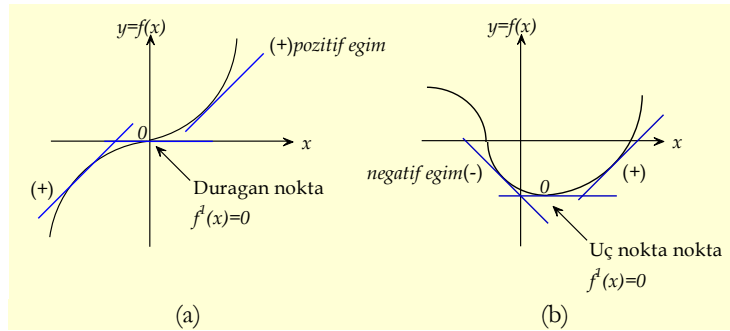


değeri azalma eğiliminden artma trendine veya artma trendinden azalma eğilimine geçmekte ve geçiş noktasında eğim sıfırlanmaktadır. Bu nedenle türevin sıfır olması öncesinde ve sonrasında genellikle türev değeri farklı işarete sahiptir. Dolayısıyla  $x = x^*$  noktasında  $f'(x^*) = 0$  eşitliğini yazmak mümkündür.



Şekil 1.77 (a) Üç boyutlu ortamda, (b) iki boyutlu ortamda uç nokta tanımlamaları.

Bununla beraber birinci mertebeden türevin sıfır olduğu her noktayı uç nokta olarak düşünmek yanıltıcıdır. Zira bazı hallerde türevin sıfır olduğu nokta durağan<sup>i</sup> nokta mahiyetinde olabilir. Örneğin gittikçe azalan bir eğime sahip amaç fonksiyonu, sıfır eğim değerine sahip olduktan sonra yön değiştirmeden artan bir eğime sahip olacak şekilde değer değiştirebilir.



Şekil 1.78 (a) Durağan ve (b) uç nokta tanımları.

<sup>i</sup> Stationary point

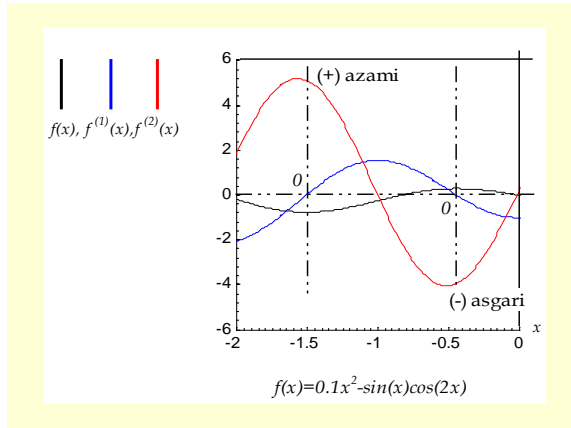


Öte yandan hatırlanacağı üzere uç noktanın azami mi yoksa asgari nokta mı olduğu hususunda ikinci türev önem kazanmaktadır. Buna göre amaç fonksiyonunun uç noktadaki ikinci mertebeden türev değeri pozitif ise söz konusu uç nokta minimum noktadır. Buna mukabil amaç fonksiyonunun ikinci mertebeden türev değerinin negatif olması ise söz konusu noktanın maksimum nokta olduğunun göstergesidir.



Örnek Uygulama

$f(x) = 0.1x^2 - \sin(x)\cos(2x)$  fonksiyonunun  $[-2, 0]$  aralığında aldığı değerler Şekil 1.79 dahilindeki grafikte görülmektedir.



Şekil 1.79 Örnek fonksiyon, fonksiyona ait 1inci ve 2ci mertebeden türev değeri değişimleri.

Buna göre verilen aralıkta, fonksiyonun minimum olduğu noktada, birinci mertebeden türev değeri  $f'(x) = 0$  olmaktadır. Ayrıca aynı noktadaki ikinci mertebeden türev değeri  $f''(x) > 0$  şeklindedir. Benzer şekilde fonksiyonun maksimum olduğu noktada, birinci mertebeden türev değeri  $f'(x) = 0$  olmaktadır. Aynı noktadaki ikinci mertebeden türev değeri ise  $f''(x) < 0$  şeklindedir.



Her nesne, mutlak sıcaklık değeri olan  $0^\circ\text{K}$  üzerindeki her sıcaklıkta çevresine ışıınım yolu ile enerji yayar. Sahip olunan sıcaklık değerine göre yayılan enerjinin ne kadar olduğunu veren yasa, Planck yasası olup;





$$E_{\lambda} = \frac{2\pi hc^2}{\lambda^5 \left( e^{\frac{hc}{kT\lambda}} - 1 \right)}$$

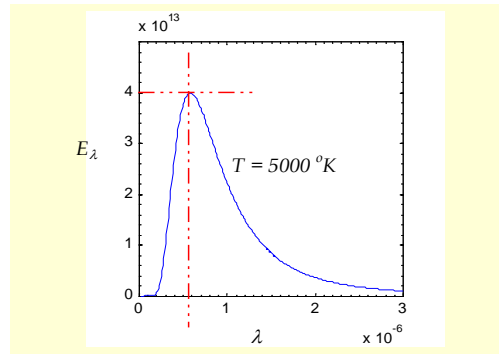
denklemleri ile elde edilir. Burada  $h$  Planck sabiti ( $6.626 \times 10^{-34} \text{ Js}$ ),  $c$  ışık hızı ( $300,000,000 \text{ m/s}$ ) ve  $k$  Boltzmann sabiti ( $1.38 \times 10^{-23} \text{ JK}^{-1}$ ) olarak tanımlanır. Denklemlerde değişken niteliğindeki parametreler ise  $\lambda$  ( $m$ ) dalga boyu ve  $T$  ( $^{\circ}\text{K}$ ) sıcaklıktır.

Belli bir sıcaklık değeri için dalga boyuna göre enerji yayılımı, Şekil 1.80'deki gibi bir karaktere sahiptir. Şekle göre enerji yayılımı belli bir dalga boyunda doruk noktaya ulaşmaktadır. Eğer sıcaklığın sabit olduğu kabulü ile yayılan enerjinin maksimum olduğu dalga boyu elde edilmek istenirse, enerji denkleminin dalga boyuna göre türetilmesi ve elde edilen denklemin sıfıra eşitlenmesi yeterli olacaktır. Zira söz konusu doruk noktada denklemin türevi sıfırdır. Buna göre;

$$\frac{\partial E_{\lambda}}{\partial \lambda} = 0$$

$$\lambda_{maks} = \frac{2.898 \times 10^{-3}}{T}$$

sonucuna ulaşılabilir. Burada dalga boyu  $m$ , sıcaklık ise  $^{\circ}\text{K}$  cinsindedir.



Şekil 1.80 5000  $^{\circ}\text{K}$  sıcaklıktaki enerji yayılımı.

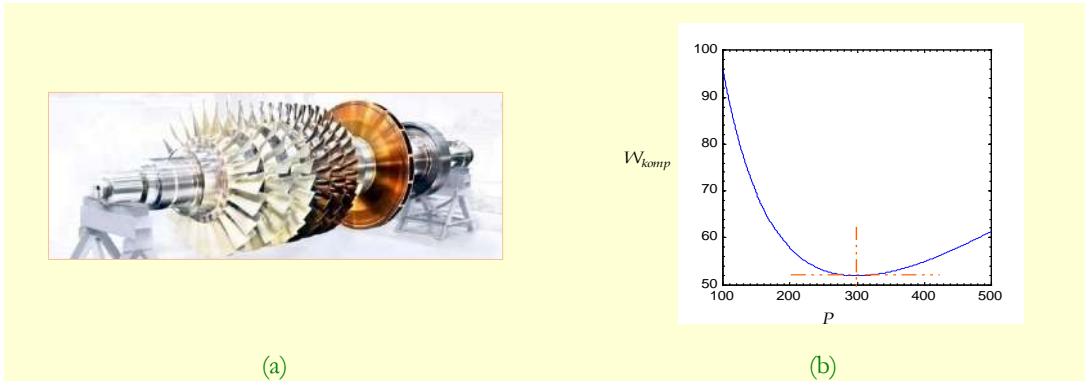




Gaz türbinli motorlarda türbinden elde edilen işin bir kısmı konum olarak daha önde bulunan kompresörlere aktarılır. Buradaki ana gaye, verilen iş ile kompresördeki havanın sıkıştırılmasıdır. İki kademeli bir kompresörde harcanan iş enerjisi;

$$W_{komp} = \frac{nRT_1}{n-1} \left[ \left( \frac{P}{P_1} \right)^{(n-1)/n} - 1 \right] + \frac{nRT_1}{n-1} \left[ \left( \frac{P_2}{P} \right)^{(n-1)/n} - 1 \right]$$

denklemler ile elde edilir<sup>(48)</sup>. Burada  $n$  politropik katsayısı,  $R$  ideal gaz sabiti,  $T_1$  kompresör girişindeki havanın sıcaklığı,  $P_1$  kompresör girişindeki havanın basıncı,  $P_2$  ikinci kademe çıkışındaki havanın basıncı ve  $P$  iki kompresör kademesi arası basınç değeridir.



**Şekil 1.81** (a) Eksenel kompresör<sup>(49)</sup> ve (b) ara kademe basınç değerine göre örnek iş enerjisi değişimi.

Kompresör tarafından harcanan işin mümkün mertebe az olması istenir. Zira türbinden elde edilen enerjinin, itki kuvveti üretmek için kullanılması esastır. Dolayısıyla ara kademe basınç değerine bağlı olarak yapılan işin minimizasyonu amaçlanır. Buna göre;

$$\frac{\partial W_{komp}}{\partial P} = 0$$

işlemi dikkate alınıp, girdi ve çıktı basınç değerlerine göre ara basınç değerinin;

$$P = \left( \frac{P_2}{P_1} \right)^{1/2}$$

olması gerektiği sonucuna varılabilir.





### 1.3.6 Eniyileme problemlerinin sınıflandırılması

Optimizasyon problemlerini farklı ölçütlere göre sınıflandırmak mümkündür. Bununla beraber sınıflandırmada, standart eniyileme problemi tanımından hareket edilebilir. Standart bir optimizasyon probleminin temel öğeleri olan amaç fonksiyonu, tasarım parametreleri ve kısıtlara göre eniyileme problemlerini sınıflandırabiliriz.

Amaç fonksiyonunun gerek niteliği ve gerekse de niceliği, sorunun sınıflandırılması için dikkate alınabilir. Daha önce ifade edildiği üzere eğer amaç fonksiyonu kapsamında sadece bir amacın eniyilemesi yapılıyorsa, bu takdirde sorun tek amaçlı<sup>i</sup> optimizasyon problemi olarak adlandırılır. Amaç fonksiyonu dahilinde birden fazla amacın güdülmesi halinde ise sorun çok amaçlı<sup>ii</sup> optimizasyon problemine dönüşür. Farklı amaçlar aynı mühendislik disiplini dâhilinde olabileceği gibi farklı mühendislik disiplini dâhilinde de olabilir.

Tekli amaç	Çoklu amaç
$\min\{f(x)\}$	$\min\{f_1(x), f_2(x), \dots, f_n(x)\}$



Örnek Uygulama

Hava araçlarının optimizasyonu sürecinde, aerodinamik kaygılar yanında yapısal ve gizelyem özellikleri de dikkate alınabilir. Bahse konu her bir kaygı, ayrı bir mühendislik disiplini içerir. Sadece aerodinamik kaygının dikkate alındığı eniyileme süreci için aşağıdaki gibi bir amaç fonksiyonu yazılabilir;

$$\min f(x) = \frac{1}{(L/D)}$$

Burada  $L$  taşıma kuvvetini,  $D$  ise sürükleme kuvvetini ifade etmektedir. Aerodinamik bazlı amaç fonksiyonuna, diğer bir disiplin olan gizelyem özelliği de katılmak istenirse bu durumda;

$$\min f_1(x) = \frac{1}{(L/D)}, \quad \min f_2(x) = \sigma_{maks}$$

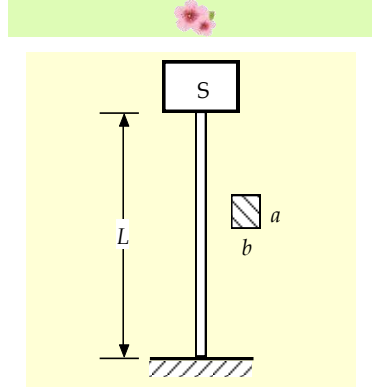
amaç fonksiyonları yazılabilir.

<sup>i</sup> Single objective

<sup>ii</sup> Multi-objective



Burada  $\sigma_{maks}$  belli bir pozisyonda hava aracının radar kesit alanını ifade eden dağılımının azami değeridir. Optimizasyon süreci sonunda elde edilen ve aerodinamik açıdan oldukça iyi sonuçlar veren hava aracı geometrisi, radara yakalanmama özelliği niteliğindeki gizayem seviyesi açısından kötü bir performansa sahip olabilir. Dolayısıyla sürecin her iki özelliği de en iyi yapacak şekilde iletılması gerekir.



Şekil 1.82 Bası ve burkulma etkisinde kalan kiriş sorunu.

Şekil 1.82'deki gibi bir düzenek kapsamında  $S$  kütesine sahip su tankı ile bu tankı taşıyan  $a \times b$  kesit alanına ve  $L$  uzunluğuna sahip bir kirişi dikkate alalım. Kirişi öyle tasarlamak istiyoruz ki kütlesi asgari değerde olsun. Ancak emniyet açısından da herhangi bir sorun yaşanmasın. Buna göre amaç fonksiyonunu;

$$\min f(x) = m$$

şeklinde yazmak mümkündür. Aynı sorunu, aynı disiplin içerisinde ilave bir amaç daha dikkate alarak yeniden düzenlemek mümkündür. Buna göre minimum kütle beklentisi yanında, kirişin burkulmaya karşı doğal frekansının da maksimum olmasını isteyelim. Bu durumda amaç fonksiyonları;

$$\begin{aligned} \min f_1 &= m \\ m &= \rho abL \end{aligned}$$

$$\begin{aligned} \max f_2 &= \omega \\ \omega &= \left[ \frac{Eab^2}{4L^2(M + 0.2357m)} \right]^{1/2} \end{aligned}$$

halini alır. Burada  $\rho$  kirişin özkütlesini ifade etmekte olup  $E$  kirişe ait elastisite modülüdür.



Amaç fonksiyonunun niceliği açısından yapılan değerlendirme yanında, niteliği açısından da bir tasnife gidilebilir. Eğer amaç fonksiyonu ve ilave olarak kısıtların hepsi aşağıdaki gibi bir doğrusal karaktere sahip ise bu takdirde problem doğrusal eniyileme sorunu<sup>i</sup> olarak adlandırılır:

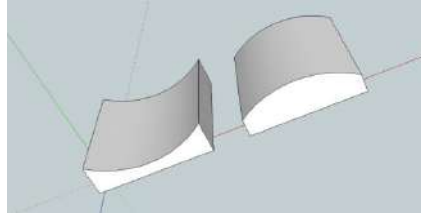
$$\min f(\mathbf{x}) = \sum_{i=1}^n c_i x_i$$

Kısıt

$$g(\mathbf{x}) = \sum_{i=1}^n a_{ij} x_i = b_j, \quad j = 1, 2, \dots, m$$

Burada  $a_{ij}, b_j$  ve  $c_i$  katsayıları sabit sayıları temsil etmektedir. Buna mukabil amaç fonksiyonu veya kısıtlardan herhangi birisinin doğrusal olmayan ifadeler içermesi durumunda, sorun doğrusal olmayan eniyileme problemi<sup>ii</sup> olarak adlandırılır. Doğrusal olmayan sorun için tasarım parametrelerinden herhangi birinin 1'den farklı bir üssel ifade içermesi veya aynı parametrelerin en az ikisinin çarpımının söz konusu olması yeterlidir.

Doğrusal amaç fonksiyonu	Doğrusal olmayan amaç fonksiyonu
$\min f(\mathbf{x}) = \sum_{i=1}^n c_i x_i$	$\min f(\mathbf{x}) = \sum_{i=1}^n c_i x_i^k, \quad k \neq 1$ $\min f(\mathbf{x}) = \sum_{i=1}^n \sum_{j=1}^m c_i x_i^k x_j^t, \quad k \text{ veya } t \neq 0$



Şekil 1.83 3B ortamda dışbükeylik örnekleri<sup>(50)</sup>: dışbükey olmayan ve dışbükey sorun.

Öte yandan doğrusal veya doğrusal olmayan sorun, dışbükey olup olmamasına göre de sınıflandırılabilir. Bazı problemler dışbükey, bazı problemler ise dışbükey olmayan sorun olarak

<sup>i</sup> Linear optimization problem

<sup>ii</sup> Nonlinear optimization problem





adlandırılır. Hatırlanacağı üzere dışbükeylik, kümeler bahsinde detaylandırılmış ve bünyesinde sadece bir uç nokta (azami veya asgari) barındıran kümelere dışbükey küme adı verilmişti. Dışbükey kümelere bir tek optimal çözüm vardır ve o da genel en iyi çözümdür. Dışbükey olmayan problemlerde birden fazla yerel çözüm mümkün olup yerel nitelikli bu çözümlerden sadece bir tanesi genel optimal çözümdür.

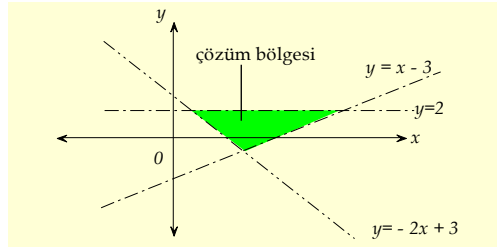


Örnek Uygulama

Aşağıdaki gibi tanımlı bir optimizasyon problemi, doğrusal optimizasyon sorununa örnek olarak gösterilebilir. Buna göre;

$$\begin{aligned} \min f(x) &= y \\ \text{Kısıt} \quad y &\leq 2 \\ y - x &> -3 \\ y + 2x &> 3 \end{aligned}$$

olarak verilen problemde, gerek amaç fonksiyonu ve gerekse de kısıtlar doğrusal denklemleri içermekte olup çözümün dışbükey olduğunu da söylemek mümkündür. Zira çözüm sadedinde sadece bir tek minimum noktanın olduğu görülmektedir.



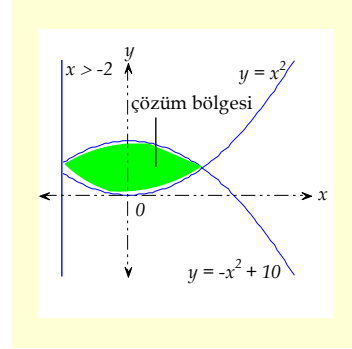
Şekil 1.84 Doğrusal dışbükey eniyileme sorununa bir örnek.

Doğrusal olmayan eniyileme problemi için aşağıdaki gibi bir sorun tanımlanabilir. Buna göre;

$$\begin{aligned} \min f(x) &= y^2 - 5 \\ \text{Kısıt} \quad x &> 2 \\ y - x^2 &> 0 \\ y + x^2 &< 10 \end{aligned}$$



olarak verilen problemde, gerek amaç fonksiyonu ve gerekse de kısıtlar doğrusal olmayan denklemler içermekte olup çözümün burada da dışbükey olduğunu söylemek mümkündür.



Şekil 1.85 Doğrusal olmayan dışbükey eniyileme sorununa bir örnek.



Bir cam imalat atölyesinde, çay ve su bardağı üretim sayılarının belirlenerek kârın doruklaştırılması hedeflenmektedir. Buna göre çay bardağının birim bazında getirdiği kâr 20 kuruş, su bardağının ise 30 kuruştur. Üretim aşamasında çay bardağının ihtiyacı olan cam hammadde miktarı 50 gram iken, su bardağının ihtiyacı olan hammadde miktarı 120 gramdır. Günlük cam hammadde temini en fazla 100 kg olarak verilmektedir.



Şekil 1.86 Su ve çay bardağı örneği.

Buna göre amaç fonksiyonunu;

$$\min f(c, s) = -(20c + 30s)$$

şeklinde yazabiliriz.





Burada  $c$  üretilen çay bardağı sayısını ve  $s$  su bardağı sayısını göstermektedir. Üretilen bardak miktarına göre sarf edilen hammadde kısıtını;

$$50c + 120s \leq 100000$$

denklemleri ile temsil edebiliriz. İfade edilen denklemlerin değerlendirilmesi neticesinde, gerek amaç fonksiyonunun ve gerekse de kısıtların doğrusal denklem niteliğinde tanımlandığı, dolayısıyla sorunun lineer bir optimizasyon problemi olduğu söylenebilir.



**Şekil 1.87** Cam üretim fabrikası.



M.Ö. 4. yüzyılda yaşamış matematikçilerden biri olan Euclid'in tasarladığı geometri problemlerinden biri de bir üçgene yerleştirilecek paralelkenarın alanının doruklaştırılması problemidir. Buna göre Şekil 1.87'deki gibi  $ABC$  noktalarının belirlediği bir üçgen içerisine,  $AEDC$  paralelkenarının yerleştirilmesi istenmektedir. Yerleşimde önemli olan husus, paralelkenar alanının mümkün olan en büyük değeri almasıdır. Paralelkenar ile üçgen arasındaki bağlantıyı,  $E$  noktasının konumu ile ilişkilendirmek mümkündür. Yerleştirilmesi düşünülen paralelkenarın alanını;

$$\frac{Hx(a-x)}{a}$$

denklemleri ile hesaplayabiliriz. Dolayısıyla optimizasyon problemi;

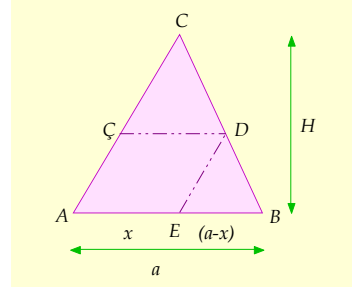
$$\min f(x) = \frac{a}{Hx(a-x)}$$

Kısıt

$$0 < x < a$$

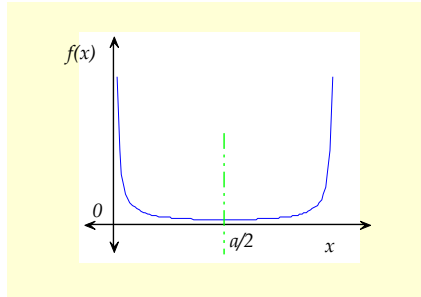
şeklinde tanımlanabilir.





**Şekil 1.88** Doğrusal olmayan dışbükey eniyileme sorununa geometriden bir örnek.

Görüldüğü üzere dikkate alınan kısıt lineer de olsa, amaç fonksiyonu doğrusal olmayan bir karakter taşımaktadır. Amaç fonksiyonunun  $x$  değerine göre değişimi Şekil 1.89 dâhilinde görülmektedir. Grafikten de anlaşılacağı üzere problem doğrusal olmayan bir sorun olup ilave olarak dışbükey bir karakter taşımaktadır. Zira çözüm, tek bir genel uç nokta barındırmakta olup  $x = a/2$  noktası paralelkenarın alanının maksimum olduğu noktadır.



**Şekil 1.89** Amaç fonksiyonunun  $x$  değerine göre değişimi,  $a = H = 10$ .



Optimizasyon probleminde kısıtların niteliğinden ziyade varlığı veya yokluğu dikkate alınarak bir ayrıma gitmek mümkündür. Eğer tasarım değişkenleri için herhangi bir kısıt söz konusu değilse sorun kısıtsız<sup>i</sup> optimizasyon problemi olarak adlandırılır. Diğer taraftan tasarım parametreleri belirli kısıtlar dâhilinde değer alabiliyorsa bu takdirde problem kısıtlı<sup>ii</sup> eniyileme sorunu olarak tasnif edilir.

<sup>i</sup> Unconstrained optimization problem

<sup>ii</sup> Constrained optimization problem





Kısıtsız problem tanımı	Kısıtlı problem tanımı
$\min_{x \in \mathbb{R}^n} f(x)$	$\min_{x \in \mathbb{R}^n} f(x)$
	$h(x) = 0$
	$g(x) \leq 0$
	$x_L \leq x \leq x_U$

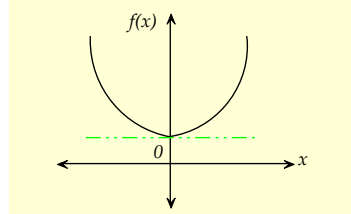


### Örnek Uygulama

Aşağıdaki gibi basit bir matematiksel eniyileme sorununu tanımlayalım:

$$\min f(x) = x^2 + 1$$

Görüldüğü üzere sorunda herhangi bir kısıt verilmemiştir. Buna mukabil tanımlı amaç fonksiyonu doğrusal olmayan bir denklem olup aynı zamanda dışbükey bir denklemdir. Dolayısıyla tek bir çözüm vardır;  $x^* = 0$ .



**Şekil 1.90** Kısıtsız optimizasyon problemi örneği:  $f(x) = x^2 + 1$ .

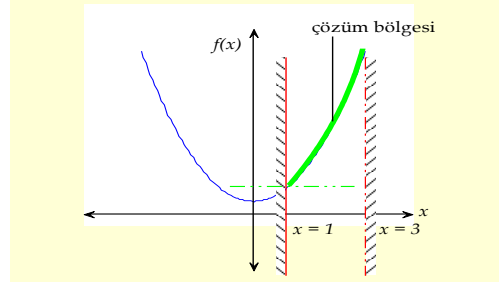
Aynı problemi, tasarım değişkenine bazı kısıtlamalar getirerek kısıtlı optimizasyon sorunu olarak da tanımlayabiliriz. Buna göre;

$$\begin{aligned} \min f(x) &= x^2 + 1 \\ \text{Kısıt} \quad &1 \leq x < 3 \end{aligned}$$

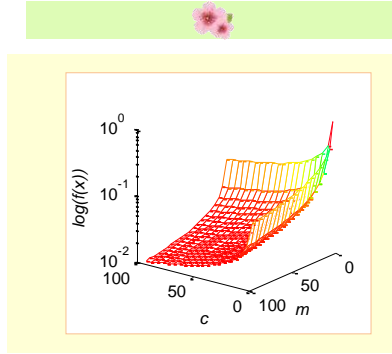
problemi, basit bir kısıtlı optimizasyon problemidir. Müteakip şekilde fonksiyona ait kısıtlanmış çözüm bölgesi ve minimum fonksiyon noktası görülmektedir.







Şekil 1.91 Kısıtlı optimizasyon problemi örneği.



Şekil 1.92 Amaç fonksiyonunun  $c$  ve  $m$  değişkenlerine göre değişimi.

Beslenme programı yapan bir diyetisyenin, cennet elması ile muzdan oluşan bir diyet seçimi yapması gerektiğini düşünelim. Seçim esnasında söz konusu meyvelerden elde edilen besin katkısını;

$$f(c, m) = \sqrt{cm}$$

denklemi ile hesaplandığını düşünelim. Burada  $c$  cennet elması miktarını ve  $m$  muz miktarını göstermektedir. Diğer taraftan tane bazında meyvelerin maliyetinin cennet elması için 50 kuruş, muz için 100 kuruş, eldeki bütçenin ise 100 TL olduğunu kabul edelim. Belirtilen hususlar çerçevesinde optimizasyon problemini;

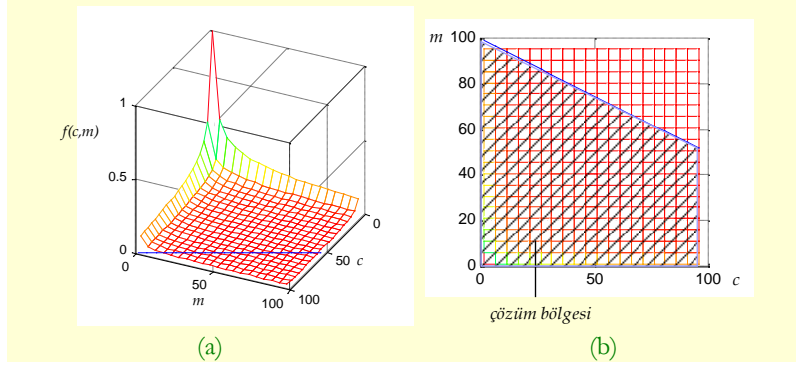
$$\begin{aligned} \text{Kısıt} \quad \min f(c, m) &= \frac{1}{\sqrt{cm}} \\ 0.5c + m &\leq 100 \end{aligned}$$

formunda tanımlamak mümkündür.



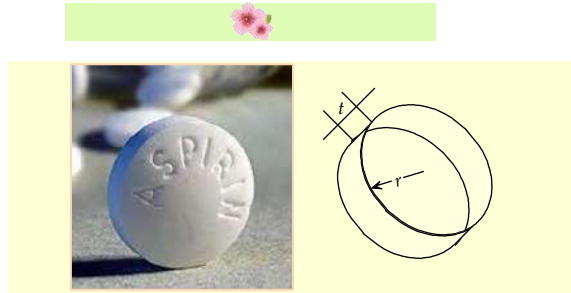


Amaç fonksiyonu, niteliği itibari ile doğrusal olmayan bir karaktere sahiptir. Dolayısıyla problem doğrusal olmayan bir eniyileme sorunudur. Şekil 1.92 dahilinde amaç fonksiyonunun tasarım parametre değerlerine göre değişimi görülmektedir.



**Şekil 1.93** (a) Amaç fonksiyonu ve kısıtın perspektif görünümü, (b) çözüm bölgesinin  $cm$  düzlemindeki görünümü.

Diğer taraftan bütçe kaynaklı kısıtın varlığı, sorunu kısıtlı optimizasyon problemi haline getirmektedir. Tanımlı amaç fonksiyonuna göre muz veya cennet elmasından ne kadar çok alınırsa besin katkısı o kadar artmaktadır. Ancak meyve alım sayısının bütçeyi aşmaması gereği, çözüm uzayını daraltmakta ve bu nedenle çözüm uzayı Şekil 1.93(b)'deki gibi bir bölgeyi kapsamaktadır;



**Şekil 1.94** Aspirine ait geometrik tanımlamalar.

Aspirin imalatı dâhilinde üretilcek aspirinlerin şeklinin silindir olması düşünülmektedir. Ayrıca aspirinin suda çözünmemesi ya da en azından uzun süre suya dayanabilmesi istenmektedir. Çözünmede





geometrik unsur yüzey alanı olup minimum yüzey alanı minimum çözünme anlamına gelmektedir. Buna mukabil ilacın hasta üzerinde etkisinin olabilmesi için belli bir aspirin madde miktarının dikkate alınması gerekir. Madde miktarı kütle ile ve dolayısıyla şeklin kapsadığı hacimle ilgilidir. Buna göre ifade edilen sorunu;

$$\min f(r, t) = 2\pi r^2 + 2\pi r t$$

Kısıt

$$\pi r^2 t = \frac{m}{\rho}$$

formunda tanımlamak mümkündür. Burada  $r$  aspirinin yarıçapını,  $t$  kalınlığını,  $m$  kütle miktarını ve  $\rho$  yoğunluğunu ifade etmektedir. Tabiatıyla kütle ve yoğunluk değerleri sabit olup çözüm sadedinde 1 olarak alındığı varsayılmaktadır. Problemin tanımı dikkate alındığında amaç fonksiyonu ile kısıt fonksiyonunun doğrusal olmayan karakterde olduğu görülür. Dolayısıyla doğrusal olmayan bir kısıtlı optimizasyon sorunu söz konusudur. Diğer taraftan dikkate alınan kısıt, bize yarıçap ile kalınlık arasında bir ilişki kurabilme imkânı sağlamaktadır. Buna göre;

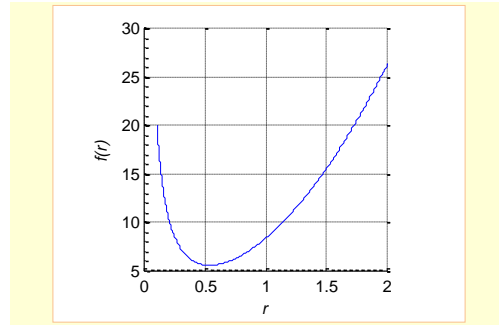
$$\pi r^2 t = 1$$

$$t = \frac{1}{\pi r^2}$$

bağıntısı kolayca yazılabilir. Elde edilen bu bağıntı amaç fonksiyonunda yerine konulduğunda;

$$\min f(r) = 2\pi r^2 + \frac{2}{r}$$

denklemi elde edilir. Denklem yarıçap değerine göre değişimi aşağıdaki şekilde görülmektedir. Buna göre problem dışbükey bir sorun olup tek bir genel minimum nokta söz konusudur.



Şekil 1.95 Amaç fonksiyonunun yarıçapa göre değişimi.



Optimizasyon problemlerinde kullanılan tasarım değişkenlerinin niceliği, niteliği ve ilave olarak fenotipi<sup>i</sup> diyebileceğimiz özelliklerine göre bir tasnife gitmek mümkündür. Nicelik bahsinde dikkate alınan husus, tasarım yöneyinin değişken sayısıdır. Eğer tasarım yöneyi tek bir değişkenden oluşuyorsa, bu takdirde sorun tek değişkenli eniyileme sorunu<sup>ii</sup> olarak adlandırılır. Tasarım değişken sayısının birden fazla olması durumunda sorun, çok değişkenli optimizasyon problemi<sup>iii</sup> olarak tanımlanır. Bununla beraber çok değişkenli eniyileme süreçlerinde de tek değişkenli tarama yöntemleri kullanılır. Zira gidilmesi gereken yön bilgisi yanında, tasarım yöneyi için atılması gereken adım büyüklüğünün de tespit edilmesi gerekir. Adım büyüklüğü, tek değişkenli bir tarama süreci ile belirlenebilir. Standart bir tekrarlı<sup>iv</sup> çözüm yönteminde, bir sonraki konum aşağıdaki denkleme göre belirlenebilir;

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda^{(k)} \mathbf{d}^{(k)} \quad (1.199)$$

Burada  $k$ 'nıncı adımda bulunan  $\mathbf{x}^{(k)}$  konumundan bir sonraki  $\mathbf{x}^{(k+1)}$  konuma gitmek için öncelikle hangi yönde gidilmesi gerektiği,  $\mathbf{d}^{(k)}$  tespit edilir. Yön tayini, tasarım yöneyinin çok değişkenli olması durumunda, her bir değişken için belirlenmesi gereken çok boyutlu bir sorundur. Bununla beraber belirlenen yönde hangi adım büyüklüğünde,  $\lambda^{(k)}$  ilerleme yapılacağı tek boyutlu bir sorun olup genellikle tek değişkenli eniyileme süreci ile elde edilir;

Tasarım yöneyinin niteliği dikkate alınarak yapılan tasnifte iki husus dikkat çekmektedir. İlki, tasarım değişkenlerinin ne tür bir değişken olduğu ile ilgili olup daha önce ikili, tam sayı, gerçek sayı ve ayrık tipte tasarım değişkenlerinden bahsedilmişti. İkinci husus, değişkenin rastsal veya deterministik olup olmadığı ile ilgilidir. Deterministik tasarım parametrelerinde verili değer kesindir. Rastsal tasarım değişkenlerinin söz konusu olduğu durumlarda genellikle dağılım fonksiyonlarından bahsedilir. Dolayısıyla parametre değeri, kesinlik taşımaktan ziyade rastsallık içerir. Gerek tasarım parametresinin doğasında ve gerekse de amaç fonksiyonu veya kısıt fonksiyonu içerisinde rastsallığın söz konusu olduğu optimizasyon problemleri, rastsal eniyileme süreçleri<sup>v</sup> olarak adlandırılır. Tasarım değişkeni, amaç fonksiyonu veya kısıtlardan her hangi birisinin rastsal olmaması durumunda bile eniyileme süreci, rastsal optimizasyon yöntemi olarak adlandırılabilir. Zira bazı eniyileme yöntemlerinde tüm terimler deterministik olsa da çözümün

---

<sup>i</sup> Phenotype

<sup>ii</sup> One-dimensional search

<sup>iii</sup> Multivariable optimization

<sup>iv</sup> Iterative

<sup>v</sup> Stochastic optimization





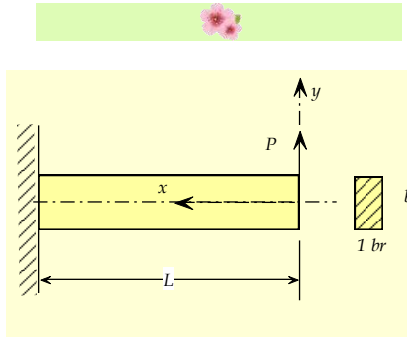
bulunması sürecinde rastsal sayılar kullanılabilir. Bu durumda da yöntem, rastsal yöntem olarak tanımlanır.

Rastsal değişken tanımı beraberinde üç kavramı gündeme getirir: dağılım fonksiyonu, ortalama değer ve varyans değeri. Ortalama değer, değişkenin alması beklenen en muhtemel değeri temsil eder. Varyans, bu değer etrafındaki kümeleşme sıklığını gösterir. Dağılım fonksiyonu, söz konusu kümeleşmenin karakterini tayin eden en önemli unsurdur. Rastsal karaktere sahip örnek bir değişkeni aşağıdaki gibi ifade etmek mümkündür;

$$\omega_i \sim N([\mu, \sigma^2])$$

Burada  $N$  dağılım fonksiyonunun normal dağılım olduğunu,  $\mu$  dağılıma ait ortalama değeri ve  $\sigma^2$  aynı dağılımın varyansını ifade etmektedir.

Deterministik eniyileme problemi	Rastsal eniyileme problemi
$\min_{x \in R^n} f(x)$	$\min_{x \in R^n} f(x, \omega)$ $\omega$ : rastsal değişken



Örnek Uygulama

Şekil 1.96 Ankastré kiriş yapıda rastsal kuvvet ve eğilme problemi.

Şekil 1.96'da verilen basit bir ankastré kirişe  $P$  kuvveti etki etmektedir. Ancak etki eden kuvvetin şiddeti rastsal olarak değişmekte olup değişim;

$$P \sim N([2000, 500]) \text{ Newton}$$

olarak verilmektedir.



Kirişin uzunluğu  $L$  ve genişliği 1 birim olup tasarımda aşağıda tanımları verilen baskı ( $\sigma_x$ ) ve kesme ( $\tau_{xy}$ ) emniyet gerilme değerlerinin geçilmemesi gerekmektedir;

$$\sigma_x = -\frac{Px}{I}y, \quad \tau_{xy} = -\frac{P}{8I}(b^2 - 4y^2)$$

Burada  $I$  eylemsizlik momenti olup  $I = b^3/12$  denklemi ile elde edilir. Buna göre eniyileme sorunu aşağıdaki gibi tanımlanabilir;

$$\begin{aligned} \min f(b, L) &= \rho bL \\ \text{Kısıt} \quad \sigma_x &< 600 \text{ MPa} \\ \tau_{xy} &< 900 \text{ MPa} \end{aligned}$$

Sorunda  $\rho$  yoğunluk olmak üzere minimum malzeme kullanımı eniyilemenin amacını oluşturmaktadır;



$$\begin{aligned} \min f(x, r) &= 3x_1 - 2x_2 + 5x_3 + rx_4 \\ \text{Kısıt} \quad x_1 + 2x_2 &\leq 10 \\ 3x_3 - 2x_4 &< 7 \end{aligned}$$

şeklinde verilen problemde  $r$  olarak ifade edilen terim;

$$r \sim N([2, 0.5])$$

biçimindeki ilave bir rastsal değişkendir. Buna göre sorun, içinde rastsal değişken barındırmasından ötürü rastsal optimizasyon problemidir.



Optimizasyon probleminde kullanılan tasarım değişkenlerinin neyi temsil ettiği önemli bir ölçüt olup tasarım değişkenlerinin fenotipine göre eniyileme problemini dört gruba ayırmak mümkündür. Buna göre parametre, şekil, topoloji ve sıralama optimizasyon sorunundan bahsedilebilir. Parametre optimizasyon sorununda, değişkenler herhangi bir geometrik özellik taşımazlar. Değişken, tasarımda sadece parametrik bir anlam ifade eder. Şekil optimizasyonunda tasarım değişkenleri, geometrik bir şekli temsil ederler ve herhangi bir değişkendeki farklılık temsil edilen geometriye doğrudan yansır. Topoloji optimizasyonunda belirli bir karaktere sahip



şeklin optimize edilmesinden ziyade tasarımdaki ana yaklaşım eniyilemeye çalışılır. Bu meyanda topolojik optimizasyon, şekil optimizasyonundan bir önceki adımı temsil eder. Sıralama veya katışım<sup>i</sup> sorunlarında ise ayrık tasarım değişkenlerinin eşleşmesi, sıralanması veya sayısı belirlenmeye çalışılır.

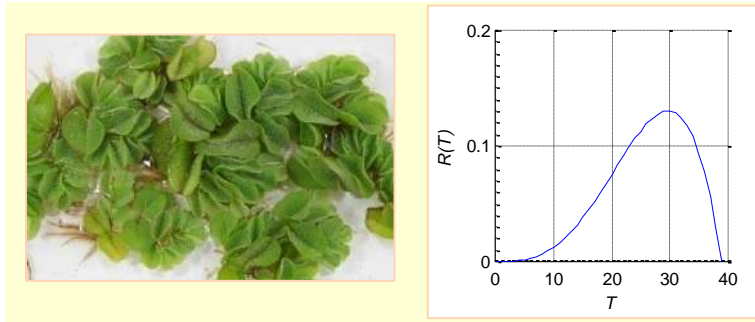


Örnek Uygulama

Makrofit<sup>ii</sup> adı ile anılan deniz bitkileri, su yüzeyinde veya yakın mesafelerde yaşayan ve oksijen üreten, aynı zamanda deniz canlıları için besin kaynağı da olan canlılardır. Makrofitlerin gelişimi ve yayılımında en önemli etmenlerden birisi sıcaklık olup sıcaklık değeri ( $T$ ) ile canlıların gelişim hızı ( $R$ ) arasında;

$$R(T) = 0.13 \left( \frac{39 - T}{9.3} \right) \left( \frac{T}{29.7} \right)^{3.1935}$$

denklemindeki gibi bir ilişki kurgulanabilir<sup>(51)</sup>. Buna göre söz konusu deniz bitkilerinin azami gelişimini sağlayacak optimal su sıcaklığı değeri nedir sorusunda temel değişken sıcaklık olup verilen denkleme parametrik bir anlamı ifade etmektedir.



Şekil 1.97 “Salvinia molesta” deniz bitkisi ve türüne ait sıcaklık değerine göre gelişim hızı.



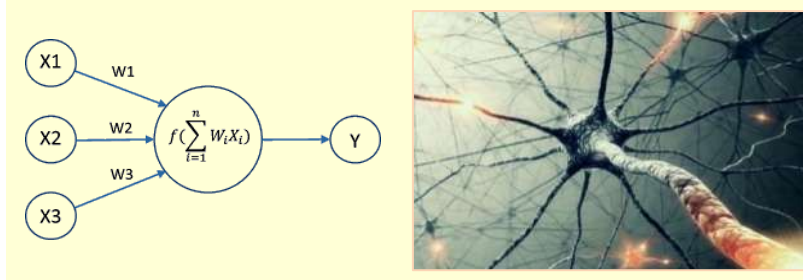
<sup>i</sup> Combinatorial

<sup>ii</sup> Macrophytes





Yapay sinir ağı<sup>i</sup>, sebep-sonuç ilişkilerinin bilinmediği veya bilinse de çözüm sadedinde yoğun hesaplama gereksinimlerinin olduğu durumlarda kullanılan, algoritmik olarak eğri ya da çok boyutlu sorunlarda yüzey uydurma yöntemleridir. Yöntemin temelinde girdi ve çıktı arasına konulan ve sinir olarak adlandırılan elemanlar ile bu elemanlar arasındaki ağırlık katsayıları yatar. Şekil 1.98 dâhilinde üç girdi verisi ve bir çıktı verisinin olduğu basit bir yapay sinir ağı görülmektedir.



Şekil 1.98 Basit bir yapay sinir ağı yapısı.

Buna göre;  $x_i$  girdi verilerini,  $w_i$   $i$ 'nci girdi verisi ile sinir arasında tanımlı ağırlık katsayısını ve  $y$  çıktı verisini ifade etmektedir. Gerçek bir sistemle yapay sinir ağı modeli arasında bir karşılaştırma yapıldığında girdi değerlerinin aynı olduğu, ancak çıktı değerlerinin farklı olabileceği söylenebilir. Ağ çıktı değeri ile gerçek çıktı ( $y_g$ ) verisinin eşit olması için ağırlık katsayılarının uygun değerlerde seçilmesi gerekir. Bu nedenle kurgulanacak bir optimizasyon sorunu;

$$\min f(w_i) = (y_g - y)^2$$

$$y = f \left\{ \sum_{i=1}^3 w_i x_i \right\}$$

denklemleri ile ifade edilebilir. Tanımlanan sorunun tasarım değişkenleri  $w_i$  olup söz konusu değişkenler ağa ait parametrik değerlerdir. Bu nedenle yapay sinir ağlarının kurgulanması, parametre optimizasyon problemi olarak tanımlanabilir.



Muharebe amacıyla kullanılan hava araçlarının radara yakalanmaması veya diğer bir ifade ile radar kesit alanının minimize edilmesi amacıyla üç teknik kullanılmakta olup bunlardan ilki radar dalgalarını emici boya ile yüzeylerin kaplanması, ikincisi radar ışınlarının dağılmasını sağlayan geometrik değişikliklerin

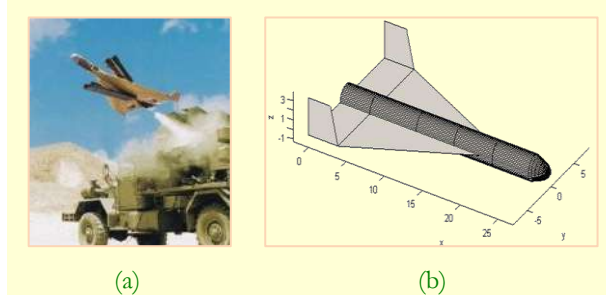
<sup>i</sup> Artificial neural nets





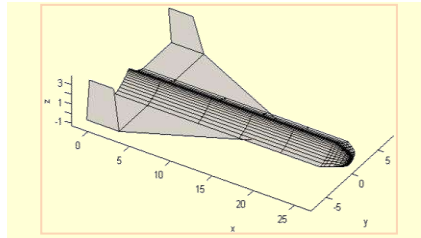


yapılması ve son yöntem ise her iki tekniğin beraber kullanılmasıdır. Radar dalgalarının mümkün mertebe atmosfere dağılmasının sağlanması ve bu sayede düşman radarlarının ekranında uçağın kesit alanının küçük olması için öncelikle uçağın geometrisi optimize edilir. Bu meyanda Şekil 1.99'da verilen ilk çizimde bir insansız hava aracına ait orijinal tasarım görülmektedir.



Şekil 1.99 (a) Örnek insansız hava aracı (İHA) kullanımı ve (b) İHA'ya ait orijinal tasarım görüntüsü.

Radar kesit alanı minimizasyonu problemini; kanat geometrisi, kanat açıları ile gövde geometrisi ve büyüklüğü gibi etmenleri dikkate alarak tanımlamak mümkündür. Bununla beraber verilen örnekte, kanat ucu eklentilerinin dikey eksen ile yaptığı açı ile dairesel gövdenin şekli, tasarım değişkenleri olarak belirlenmiş ve şekil optimizasyonu sonrasında müteakip geometriye ulaşılmıştır.

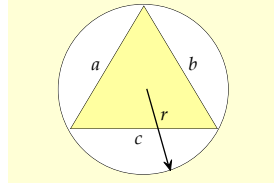


Şekil 1.100 Radar kesit alanının optimizasyonu sonucu elde edilen geometri<sup>(52)</sup>.



Yarıçapı  $r$  olan bir daire içerisine, alanı maksimum olacak şekilde bir üçgen yerleştirilmek istenmektedir.





**Şekil 1.101** Basit bir şekil eniyileme sorunu örneği.

Buna göre üçgen tarafından kapsanacak alanı azami derecede büyük yapabilmek için dikkate alınacak tasarım parametreleri üçgenin kenarları olup söz konusu tasarım yöneyi elemanları geometrik bir anlam ifade etmektedir. Buna göre üçgenin alanını;

$$s = \frac{a + b + c}{2}$$

olmak üzere;

$$A = \sqrt{s(s-a)(s-b)(s-c)}$$

denklemi ile elde etmek mümkündür. Bu durumda eniyileme sorunu;

$$\begin{aligned} \text{Kısıt} \quad \min f(x) &= A \\ 0 < x &< 2r \end{aligned}$$

ifadeleri ile tanımlanabilir. Tasarım yöneyi;

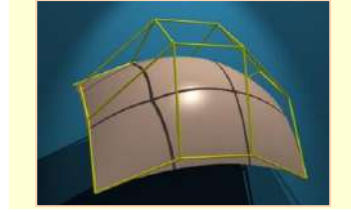
$$x = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

şeklindedir.



İki boyutlu kanat profilinin veya üç boyutlu kanadın geometrik olarak temsil edilmesi noktasında değişik yöntemler kullanılabilir. Yaygın olarak kullanılan yöntemlerden biri de Bézier eğri veya yüzeyleri ile geometrinin temsil edilmesidir.





**Şekil 1.102** Bézier kontrol noktaları (kötşe noktaları) ile temsil edilen üç boyutlu yüzey örneği.

Şekil 1.102'den de görüleceği üzere yeteri kadar kontrol noktasının kullanılmasıyla yumuşak geçişe sahip bir yüzeyi temsil etmek mümkündür. Kontrol noktalarının konumu değıştikçe yüzeyin şekli değışmektedir. Bézier yüzeyi;

$$x(u, v) = \sum_{i=0}^n \sum_{j=0}^m B_i^n(u) B_j^m(v) x_{i,j}$$

$$y(u, v) = \sum_{i=0}^n \sum_{j=0}^m B_i^n(u) B_j^m(v) y_{i,j}$$

$$z(u, v) = \sum_{i=0}^n \sum_{j=0}^m B_i^n(u) B_j^m(v) z_{i,j}$$

$$B_i^n = \frac{n!}{i!(n-i)!} u^i (1-u)^{n-i}$$

$$B_j^m = \frac{m!}{j!(m-j)!} v^j (1-v)^{m-j}$$

denklemleri ile ifade edilebilir. Burada  $x_i, y_i$  ve  $z_i$  terimleri kontrol noktalarını temsil etmektedir. Kontrol noktaları sayısal bir anlam ifade etse de bütünü itibari ile geometrik bir şekli temsil etmektedirler.



**Şekil 1.103** Bézier kontrol noktaları genotip olarak sayısal noktaları ifade etseler de fenotip olarak geometrik bir şekli temsil ederler.



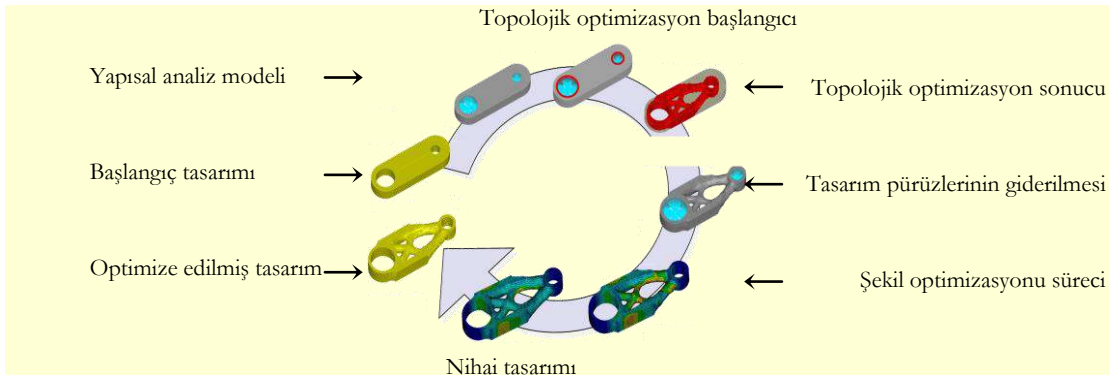
Sandalye tasarımında dikkate alınabilecek temel hususlar; görsellik, işlevsellik, maliyet ve dayanım şeklinde ifade edilebilir. Belirtilen ölçütler arasındaki görsellik ile belli ölçüde işlevselliğin sayısal bir değer ile tanımlanması zor olabilir. Bununla beraber maliyet ve dayanımın, kullanılan malzeme tipi ve miktarı ile doğrudan ilişkisi vardır. Topolojik optimizasyon sürecinde adım adım tasarımın belirlenmesi amaçlanır. Tasarımın belli bir olgunluğa gelmesini müteakip şekil optimizasyonuna geçilir. Şekil 1.104'te sandalye tasarımının başlangıç ve topolojik süreç sonunda ulaşılan son durumu görülmektedir.



Şekil 1.104 Sandalye tasarımının topolojik olarak değişimi.



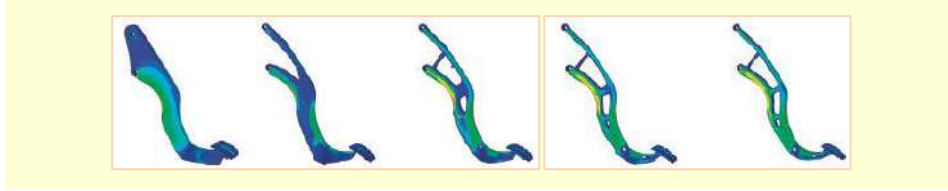
Şekil 1.105'te bir bağlantı elemanına ait tasarım ve eniyileme süreci görülmektedir. Buna göre tasarımın başlangıcında kaba ve hantal bir şekil varken, topolojik optimizasyon sonucunda tamamen farklı bir tasarıma ulaşılmaktadır. Elde edilen tasarım, müteakiben boyutsal sorunları gidermek için pürüzsüzleştirilir ve şekil optimizasyon süreci için hazır hale getirilir. Şekil optimizasyonu sonunda ise nihai tasarıma ulaşılmıştır.



Şekil 1.105 Örnek bir bağlantı elemanının topolojik ve şekil optimizasyon süreci.



Yapısal tasarım kapsamında dikkate alınan diğer bir örnek, fren pedalıdır. Araba içerisindeki fren pedalına ait başlangıç ve topolojik eniyileme süreci sonundaki nihai tasarım Şekil 1.106'da görülmektedir.



Şekil 1.106 Fren pedalı topolojik eniyileme süreci<sup>(53)</sup>.



Şekil 1.107 Sırt çantası problemi katışım sorunlarına verilen iyi bir örnektir.

Katışım problemlerine verilecek iyi bir örnek, sırt çantası sorunudur<sup>i</sup>. Bu problemde bir sırt çantasına yerleştirilecek eşyaların sayısı belirlenmeye çalışılır. İşlemin amacı, taşınabilecek azami çanta ağırlığının dikkate alınması, ama aynı zamanda farklı değerlere sahip eşyaların çantadaki değer toplamının doruklaştırılmasıdır. Bu meyanda  $n$  tane  $e_i$  eşyasının var olduğu bir sorunu;

$$\min \frac{1}{f(x)}, \quad f(x) = \sum_{i=1}^n v_i x_i$$

Kısıt

$$\sum_{i=1}^n w_i x_i \leq W$$

denklemleri ile tanımlayabiliriz. Burada  $x_i$  her bir eşyanın çantaya konulacak sayısını,  $v_i$  maddi değerini,  $w_i$  ağırlığını ve  $W$  çantada taşınabilecek azami ağırlık değerini ifade etmektedir.

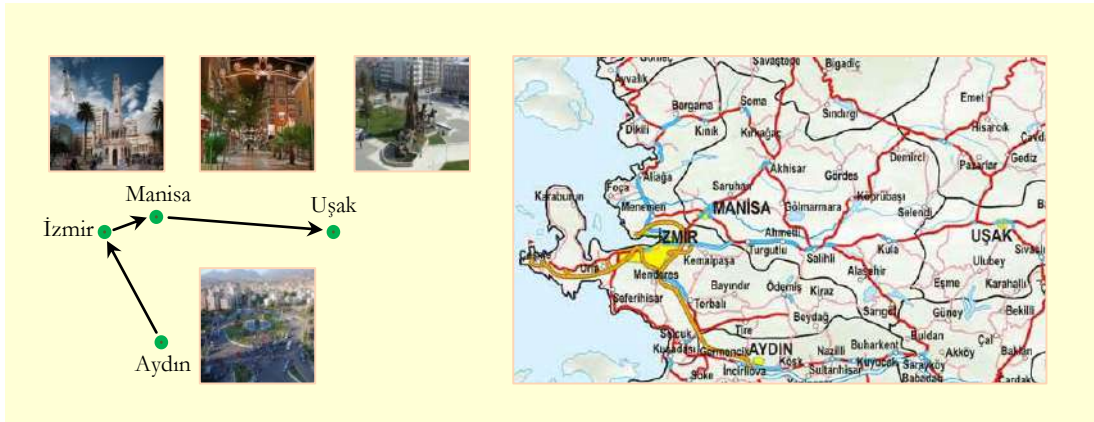
<sup>i</sup> Knapsack problem



Daha önce de bahsi geçen sıralama sorunlarında tasarım yöneyini oluşturan öğeler, genotip olarak sadece bir isimden ibaret olup gerçekte fiziksel bir nesneyi temsil ederler. Dolayısıyla dikkate alınan amaç fonksiyonuna göre objelerin sıralanması optimize edilmeye çalışılır. Örneğin Şekil 1.108'deki gibi dört farklı konumda bulunan şehirlerin, toplamda en kısa mesafe kat edilerek ziyaret edilmesinin planlandığını düşünelim. Dört farklı şehrin ziyaret sıralaması kapsamında farklı sıralama seçenekleri söz konusu olup mesafelerin toplam değeri dikkate alınarak problemin tanımlanması mümkündür. Buna göre her bir şehre verilen numara  $i$  olmak üzere  $x(i)$  her bir kenti temsil edebilir. Bu durumda problem;

$$\min f(x) = \sum_{i=1}^{n-1} d_{x(i),x(i+1)} + d_{x(n),x(1)}$$

denklemler ile tanımlanabilir. Burada  $n$  şehir sayısını,  $x$  sıralamayı gösteren tasarım yöneyini,  $d_{x(i),x(i+1)}$  ise iki şehir arasındaki mesafeyi göstermektedir. Problemden ziyade ziyaret sonrası başlangıç noktasına dönüş esas alınmaktadır.



Şekil 1.108 Şehirlerarası ziyaret planlamasında en uygun sıralamanın Aydın-İzmir-Manisa-Uşak olduğu söylenebilir.

### 1.3.7 Eniyileme problemlerinde zorluk derecesi

Doğada ve hayatta karşılaşılan sorunların her zaman çözümünün olduğunu ya da en azından bilinen bir çözümünün olduğunu söylemek zordur. Bazı sorunların en iyi çözümünü bulmak mümkün olmayabilir. Bulunan bir çözümün de en iyi çözüm olduğu her zaman iddia

edilemez. Çözüm bağlamında karşılaşılan zorluk derecesi, problemlerin sınıflandırılmasında dikkate alınabilecek bir ölçüttür.

Karmaşıklık kuramı<sup>i</sup> olarak adlandırılan yaklaşıma göre problemlerin çözümünde karşılaşılan zorluk derecesi, zaman ve/veya veri karmaşıklığı dikkate alınarak tasnif edilir. Zaman karmaşıklığından kastedilen, çözümün elde edilmesi için gerekli zamansal çabadır. Bu çaba, bazen matematiksel işlem sayısı toplamı, bazen algoritmadaki adım sayısı toplamı, bazen de tekrarlı işlemlerin toplam sayısıdır. Doğal olarak işlem, adım veya tekrar sayıları zamansal bir karşılık ifade ederler. Zamanın uzun olması, çözümü daha zor problem anlamına gelmektedir. Veri karmaşıklığı ise zamandan ziyade mekânsal bir anlam içermektedir. Çözüm dâhilinde özellikle bilgisayar hafızasında kullanılan yerin büyüklüğü, veri karmaşıklığının göstergesidir. Bilgisayar hafızasında kaplanan yerin büyüklüğü arttıkça problemin karmaşıklığı da artar.

Gerek zaman ve gerekse de veri karmaşıklığı, donanım imkânlarının geliştirilmesi ile azaltılabilir. Bununla beraber özellikle zaman karmaşıklığının, donanımdan ziyade sorunun doğasına bağlı olduğu söylenebilir. Bu nedenle problem karmaşıklığı, çözüm yöntemine veya donanımsal imkânlara göre değil, doğrudan doğruya sorunun karakterine göre değerlendirilir. Burada önem arz eden diğer etmen de sorunun içerdiği tasarım değişkeni sayısıdır. Her zaman olmasa da genel olarak tasarım değişkeni sayısının ( $n$ ) artmasının, çözümü daha da zorlaştırdığını söylemek mümkündür.

Zaman karmaşıklığının mertebesini göstermek için  $O(\dots)$  ibaresi kullanılır.  $O$  simgesi mertebeyi<sup>ii</sup>, parantez içerisindeki kısım ise mertebenin büyüklüğünü gösterir. Örneğin  $O(1)$  gösterimi, sabit zaman karmaşıklığını<sup>iii</sup> ifade eder. Burada kastedilen, problemin çözümünün tek seferde elde edilebildiği, çözümün aldığı zamanın tasarım değişkenlerine bağlı olmadığıdır.  $O(n)$  ibaresi doğrusal zaman karmaşıklığını<sup>iv</sup> ifade etmektedir. Böyle bir zaman karmaşıklığına sahip sorunun çözümüne,  $n$  değişken sayısı kadar adım veya işlem sonunda ulaşılabilir.  $O(n^k)$  zaman karmaşıklığı ilk ikisine göre çok daha karmaşık bir sorun tipidir. Polinomial zaman karmaşıklığı<sup>v</sup> olarak ifade edilen bu tür problemlerde çözüme, değişken sayısının üssel çarpımı kadar atılan adım sayısı sonucunda ulaşılabilir. Burada  $k$  değeri 1'den büyük herhangi bir tam sayıyı ifade

---

<sup>i</sup> Complexity theory

<sup>ii</sup> Order

<sup>iii</sup> Constant time complexity

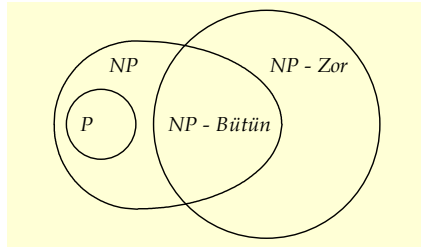
<sup>iv</sup> Linear time complexity

<sup>v</sup> Polynomial time complexity





etmektedir. Tabiatıyla  $k$  değerinin ne olacağı, sorunun tipine ve değişkenin değerine göre değişebilir. Zaman karmaşıklığı ile ilgili olarak bahsedeceğimiz son örnek  $O(m^k)$  olup üstel zaman karmaşıklığı olarak tanımlanır. Burada  $m$  sabit bir tamsayıyı göstermektedir. Polinomial zaman karmaşıklığına sahip sorunlar,  $P^i$  sınıfı problemler olarak gruplandırılır. Bu tür sorunların çözümü, deterministik yöntem ile polinomial zaman dilimi içerisinde elde edilebilir. Diğer taraftan deterministik olmayan algoritmalarla polinomial zaman dilimi içerisinde çözülebilen problemlere ise  $NP^{iii}$  sınıfı problem grubu adı verilir. Ancak bu tanım,  $P$  ve  $NP$  sınıfı problemlerin aynı mı yoksa farklı mı olduğunu tam olarak netleştirmemektedir. Zira her iki problem grubu da polinomial zaman diliminde çözülebilmektedir. Burada önemli olan, algoritmanın deterministik olup olmadığıdır. Bununla beraber deterministik çözüm ile en uygun noktanın elde edildiği sorunları deterministik olmayan algoritmalarla da çözmek mümkündür. Bu nedenle  $NP$  kümesinin  $P$  kümesini kapsadığı söylenebilir,  $P \subseteq NP$ . Öte yandan  $NP$  kümesine ait bir sorunun, yine polinomial zaman dilimi içerisinde başka bir  $NP$  problem türüne dönüştürülebilmesi mümkün ise bu tür problemlerin olduğu gruba  $NP$ -Bütün<sup>iv</sup> problem grubu adı verilir. Dönüşüm sonrası sorunun çözümünün deterministik bir yöntemle polinomial zaman dilimi içerisinde bulunabilmesi halinde söz konusu sorun grubu  $P$  sınıfı soruna dönüşmüş olur. Eğer sorunun çözümü, polinomial zaman dilimi içerisinde elde edilemiyorsa, bu takdirde sorun  $NP$ -Zor<sup>v</sup> problem olarak adlandırılır<sup>(54)</sup>. Şekil 1.109 kapsamında problem karmaşıklığı kümeleri ve aralarındaki ilişkiler görülmektedir.



Şekil 1.109  $P$  ve  $NP$  sorunları.

<sup>i</sup> Exponential time complexity

<sup>ii</sup> Polynomial

<sup>iii</sup> Non-deterministic polynomial

<sup>iv</sup> NP-complete

<sup>v</sup> NP-hard







Örnek Uygulama

Gezici satıcı sorunu, esas itibari ile bir NP-Zor problemidir. Özellikle şehir sayısını aşırı artışı durumunda en uygun çözüme ulaşmak zordur. Kaldı ki elde edilen çözümün de optimal çözüm olduğunu teyit etmek mümkün olmayabilir.



### 1.3.8 Eniyileme sürecinde yakınsama

Yakınsama<sup>i</sup> kavramı, matematiksel manada belli bir değere yaklaşma/ulaşma anlamına gelir. Kavramın tam tersi ise ıraksama<sup>ii</sup> olarak tanımlanır. Daha açık bir ifade ile belli bir değerden uzaklaşma olduğunda ıraksama söz konusudur. Optimizasyon problemlerinin çözümü sadedinde kullanılan algoritmanın yakınsama özelliği önemli bir husus olup en uygun çözüme ulaşma hızını da doğrudan etkileyen bir özelliktir. Yakınsama kavramını matematiksel olarak tanımlamak için  $n = 0$  ile başlayıp  $n \rightarrow \infty$  adedine giden bir  $P$  serisini dikkate alalım. Buna göre serideki herhangi bir eleman  $p_n$  olmak üzere yakınsama ölçütü;

$$\lim_{n \rightarrow \infty} \frac{p_{n+1} - p}{(p_n - p)^q} = k \quad (1.200)$$

$$k \geq 0$$
$$q \geq 1$$

denklemleri ile ifade edilir. Tam yakınsamanın gerçekleşmesi durumunda  $k$  değeri 0 olacaktır. Dikkat edileceği üzere ölçütteki yakınsama hızını belirleyen ana etmen  $q$  değeridir. Bu değer  $q = 1$  olması durumunda  $0 \leq k \leq 1$  olacaktır. Böyle bir yakınsamaya doğrusal yakınsama denir. Üssel etmenin  $q = 2$  olması durumunda, yakınsama ikilenik yakınsama olarak adlandırılır ve ilkinin göre çok daha hızlı bir yakınsama söz konusudur. Yöntemin özelliğine göre bazı algoritmalar doğrusal, bazı algoritmalar ikilenik ve hatta daha hızlı yakınsama özelliğine sahip olabilirler. Yakınsama kavramı, eniyileme süreçlerinde optimal değere yaklaşma olarak algılanır. Dolayısıyla çözüm sürecinde kullanılan algoritmanın en uygun değere ulaşma hızı, bir manada

<sup>i</sup> Convergence

<sup>ii</sup> Divergence



yakınsama hızını gösterir. Tabiatıyla yakınsama hızı yüksek olan çözüm algoritması, daha verimli bir algoritma olarak dikkate alınır. Yakınsamada amaç fonksiyonunun değişimi baz alındığında;

$$\left| \frac{f(x_{i+1}) - f(x_i)}{f(x_i)} \right| \leq \varepsilon \quad (1.201)$$

ölçütü dikkate alınarak bir kanaate ulaşmak mümkündür. Burada  $\varepsilon$  değeri yakınsama toleransı olarak adlandırılır. Zira yakınsamaya yaklaşıldıkça  $\varepsilon$  değeri küçülür. Amaç fonksiyonu yanında tasarım yöneyinin de dikkate alınması durumunda söz konusu yakınsama ölçütü:

$$\left| \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i} \right| \leq \varepsilon \quad (1.202)$$

şeklinde ifade edilir. Bu denklem bir anlamda amaç fonksiyonunun kısmi türevini göstermektedir. Diğer taraftan basit ve kısa bir ölçüt tanımlanmak istendiğinde sadece tasarım yöneyindeki değişim dikkate alınabilir:

$$|x_{i+1} - x_i| \leq \varepsilon \quad (1.203)$$

Bu ifadenin anlamı, çözüme yaklaştıkça çözüm arayışı için yapılan taramada küçük adımların atılmasıdır.

### 1.3.9 Eniyileme yöntemlerinin sınıflandırılması

Optimizasyon problemlerinde görülen çeşitliliğe benzer şekilde, çözüm sadedinde de oldukça geniş bir yöntem yelpazesinden bahsedilebilir. Dolayısıyla çözüm yöntemlerini, net çizgilerle ayırarak gruplandırmak zordur. Bununla beraber bazı genellemelerle optimizasyon yöntemlerini deterministik ve rastsal yöntemler olmak üzere ikiye ayırabiliriz. Deterministik yöntemler, içerisinde rastsal değişken barındırmayan yöntemlerdir. Bu meyanda grafiksel yöntemleri, doğrusal programlama tekniklerini ve tüm gradyan esaslı yöntemleri sayabiliriz. Rastsal yöntemler ise gerek optimizasyon probleminin tanımında ve gerekse de çözüm sürecinde rastsal parametreleri barındıran yöntemlere denir. Rastsal yöntemler arasında yaygın olarak kullanılanlar, akıllı sistemler olarak adlandırılan ve daha çok doğadan esinlenilerek geliştirilen yöntemlerdir.

Akıllı sistemler arasında sayılan bazı yöntemler kendilerine özgü bir isimle, sezgisel yöntemler adıyla anılmaktadırlar. Başlıca sezgisel yöntemler arasında; genetik algoritmalar, parçacık sürü



algoritmaları, karınca kolonisi algoritmaları, benzetimli tavlama algoritmaları, tabu algoritmaları ve yapay arı kolonisi algoritmaları sayılabilir. Akıllı sistemler kapsamındaki diğer yöntemler arasında; yapay sinir ağları ve bulanık mantık sayılabilir. Ancak son iki yöntem eniyileme yöntemi olmaktan ziyade sanal amaç fonksiyonu tanımlama yöntemi olarak kullanılır. Deterministik yöntemler arasında en yaygın olarak kullanılan ve matematiksel olarak da uzun süreden beri üzerinde çalışılan yöntemler gradyan esaslı yöntemler olup söz konusu yöntemlerin temelinde, birinci/ikinci derece türev değerleri kullanılır. Başlıca gradyan esaslı yöntemler arasında; endik azalma<sup>i</sup> yöntemi, eşlenik gradyan<sup>ii</sup> yöntemi, Newton yöntemi, Marquardt yöntemi, sanki Newton<sup>iii</sup> yöntemi, Davidon-Fletcher-Powell yöntemi, Broydon-Fletcher-Goldfarb-Shanno yöntemi ve ardışık ikilenik programlama<sup>iv</sup> yöntemi sayılabilir. Günümüzde gradyan esaslı yöntemler hızlı ve olumlu netice veren özellikleri nedeniyle endüstride yaygın olarak kullanılmaktadır. Ancak sezgisel yöntemlerin de kendine göre bazı avantajları söz konusu olup genel yönelim, sezgisel yöntemlere doğru kaymaktadır.

---

<sup>i</sup> Steepest descent

<sup>ii</sup> Conjugate gradient

<sup>iii</sup> Quasi-Newton

<sup>iv</sup> Sequential quadratic programming



## Bilim Damlası



çemişini 11'inci yüzyıl bilim insanlarından al-Ghazali'ye kadar götürebileceğimiz “evrenin en iyi evren” olduğu anlayışının, evreni yöneten yasaları çıkarsama açısından kullanılması oldukça şaşkınlık verici ve ilginç bir olgudur. Somut halini Hamilton ilkesiyle gördüğümüz bu ilke, aslında evrenin optimal seçeneklerle kurgulandığını göstermektedir. Bahse konu önerme, sadece klasik mekanik için değil kuantum mekaniği için de geçerlidir. Bu nedenle söz konusu ilkenin ve örnek bir doğa yasası çıkarsamanın burada ele alınması eniyileme disiplininin önemini vurgulama açısından kayda değerdir.

Zaman ve mekân sarmalı içerisinde gerçekleşen fiziksel olayların temel karakteristiğini eylem olgusu ile örtüştürmek mümkündür. Genel haliyle eylem<sup>i</sup>, belli bir zaman dilimi içerisinde gerçekleşen fiziksel sürecin kullandığı enerji toplamı olarak tarif edilebilir. Dolayısıyla eylemin birimi “enerji×zaman” şeklinde tarif edilir. Söz konusu tanımı aşağıdaki gibi bir matematiksel denklemle ifade etmek mümkündür:

$$S = \int_{t_1}^{t_2} L(\mathbf{x}, \dot{\mathbf{x}}, t) dt \quad (1.204)$$

Bu denklemde  $S$  eylemi,  $\mathbf{x}(t)$  konum vektörünü,  $\dot{\mathbf{x}}(t)$  konumun zamana göre değişimini ve  $t$  zaman değişkenini göstermektedir.  $L$  ise konuma, konumun zamana göre değişimine ve zamana bağlı olarak değişen ve adına Lagrange fonksiyonu denilen bir fonksiyonlar fonksiyonu olarak tanımlanabilir. İfadeye göre Lagrange fonksiyonunun  $t_1$  ve  $t_2$  zaman dilimi arasındaki integrasyonu, yapılan eylemin miktarını vermektedir. Doğal olarak eylemin miktarı, zaman dilimine ve ondan da öte doğrudan doğruya  $L$  fonksiyonlar fonksiyonuna bağlıdır.

<sup>i</sup> Action



Zaman ve mekân uzayında,  $\mathbf{x}(t_1)$  başlangıç konumundan  $\mathbf{x}(t_2)$  son konumuna ilerleyen bir fiziksel süreç düşünelim. Bu sürecin başlangıç ve bitiş konumları sabit olsun. Söz konusu iki nokta arasında mümkün olan pek çok eylem ve dolayısıyla yol söz konusudur. İzlenen muhtemel yolun  $\mathbf{x}(t)$  ve yapılan eylemin de  $S$  olduğunu varsayalım. İzlenen yolu tanımlayan  $\mathbf{x}(t)$  fonksiyonunda  $\delta\mathbf{x}(t)$  kadar küçük bir değişiklik<sup>i</sup> olduğunu düşünelim. Doğal olarak oluşan yeni  $\mathbf{x}(t) + \delta\mathbf{x}(t)$  yolu, yapılan eylemde de değişikliğe sebep olacaktır. Söz konusu eylem değişikliğini:

$$\Delta S = \int_{t_1}^{t_2} L(\mathbf{x} + \delta\mathbf{x}, \dot{\mathbf{x}} + \delta\dot{\mathbf{x}}, t) dt - \int_{t_1}^{t_2} L(\mathbf{x}, \dot{\mathbf{x}}, t) dt \quad (1.205)$$

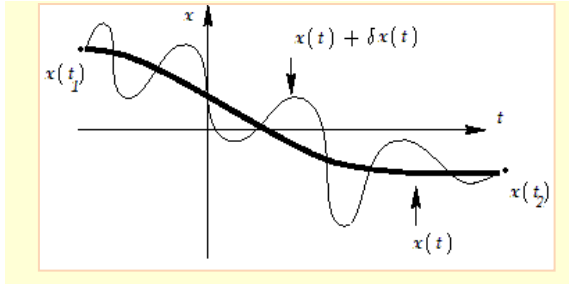
şeklinde ifade edebiliriz. Yazılan bu ifadeyi Taylor serisinin<sup>ii</sup> birinci mertebeden açılımını dikkate alınarak;

$$\Delta S = \int_{t_1}^{t_2} \left\{ \frac{\partial L}{\partial \mathbf{x}} \delta\mathbf{x} + \frac{\partial L}{\partial \dot{\mathbf{x}}} \delta\dot{\mathbf{x}} \right\} dt \quad (1.206)$$

veya parçalara ayırarak;

$$\Delta S = \int_{t_1}^{t_2} \frac{\partial L}{\partial \mathbf{x}} \delta\mathbf{x} dt + \int_{t_1}^{t_2} \frac{\partial L}{\partial \dot{\mathbf{x}}} \delta\dot{\mathbf{x}} dt \quad (1.207)$$

biçiminde yazmak da mümkündür.



**Şekil 1.110** İki nokta arasında dikkate alınan iki farklı yol ve eylem.

<sup>i</sup> A small variation

<sup>ii</sup> Taylor serisi açılımı birinci mertebeden yaklaşımla  $\Delta f = f(x + \delta x) - f(x) = \partial f / \partial x \cdot \delta x$  şeklinde yazılabilir.



Denklem 1.207'de geçen ikinci integral değerini, fonksiyonların çarpımlarının zamana göre türevi ve integrali kuralları ile yeniden yazmak mümkündür. Buna göre;

$$f = \frac{\partial L}{\partial \dot{\mathbf{x}}}$$

$$g = \delta \mathbf{x}$$

olduğu kabul edilir ve;

$$(fg)' = f'g + fg'$$

olduğu hatırlanırsa buradan;

$$fg' = (fg)' - f'g$$

yazılabilir. Bu ifadede  $f$  ve  $g$  yerine karşılıkları yazılırsa;

$$\frac{\partial L}{\partial \dot{\mathbf{x}}} \frac{\partial}{\partial t} (\delta \mathbf{x}) = \frac{\partial}{\partial t} \left( \frac{\partial L}{\partial \dot{\mathbf{x}}} \delta \mathbf{x} \right) - \frac{\partial}{\partial t} \left( \frac{\partial L}{\partial \dot{\mathbf{x}}} \right) \delta \mathbf{x} \quad (1.208)$$

denklemini elde edilir. Eşitliğin her iki tarafının integrali alınır;

$$\int_{t_1}^{t_2} \frac{\partial L}{\partial \dot{\mathbf{x}}} \frac{\partial}{\partial t} (\delta \mathbf{x}) dt = \int_{t_1}^{t_2} \frac{\partial}{\partial t} \left( \frac{\partial L}{\partial \dot{\mathbf{x}}} \delta \mathbf{x} \right) dt - \int_{t_1}^{t_2} \frac{\partial}{\partial t} \left( \frac{\partial L}{\partial \dot{\mathbf{x}}} \right) \delta \mathbf{x} dt \quad (1.209)$$

formuna ulaşılır. Bu denklemin ikinci terimini öncelikle dikkate alırsak;

$$\int_{t_1}^{t_2} \frac{\partial}{\partial t} \left( \frac{\partial L}{\partial \dot{\mathbf{x}}} \delta \mathbf{x} \right) dt = \frac{\partial L}{\partial \dot{\mathbf{x}}} [\delta \mathbf{x}(t_1) - \delta \mathbf{x}(t_2)] \quad (1.210)$$

yazılabilir. Hatırlanacağı üzere daha önce söz konusu eylem sürecinin başlangıç ve bitiş konumlarının sabit olduğu ifade edilmişti. Dolayısıyla Denklem (1.210) ifadesini,  $\delta \mathbf{x}(t_1) = \delta \mathbf{x}(t_2)$  eşitliğini dikkate alarak sıfıra eşitleyebilir ve Denklem (1.209) ifadesini de;

$$\int_{t_1}^{t_2} \frac{\partial L}{\partial \dot{\mathbf{x}}} \frac{\partial}{\partial t} (\delta \mathbf{x}) dt = - \int_{t_1}^{t_2} \frac{\partial}{\partial t} \left( \frac{\partial L}{\partial \dot{\mathbf{x}}} \right) \delta \mathbf{x} dt \quad (1.211)$$

sonucuna bağlayabiliriz. Elde edilen bu sonuç Denklem (1.207) dâhilinde dikkate alınır;



$$\Delta S = \int_{t_1}^{t_2} \frac{\partial L}{\partial x} \delta x dt - \int_{t_1}^{t_2} \frac{\partial}{\partial t} \left( \frac{\partial L}{\partial \dot{x}} \right) \delta x dt \quad (1.212)$$

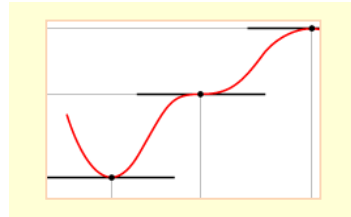
$$\Delta S = \int_{t_1}^{t_2} \left\{ \frac{\partial L}{\partial x} - \frac{\partial}{\partial t} \left( \frac{\partial L}{\partial \dot{x}} \right) \right\} \delta x dt$$

nihai ifadesine ulaşılır. Şimdiye kadar yapılan işlemler esasen matematiksel işlemler olup eylem sürecinin fiziksel durumu henüz mevzu bahis konusu edilmiş değildir. Bu noktada Hamilton, fiziksel eylem sürecinin doğası ile ilgili genel bir ilkeyi beyan eder: “fiziksel süreçler her zaman en az eylem<sup>i</sup> ilkesine göre davranış sergiler.” Bu beyan, sürecin optimal bir noktada karakterize edilmesi manasına da gelir. Optimal nokta ise azami veya asgari ama nihayetinde  $S$  değeri açısından durağan<sup>ii</sup> bir noktayı gösterir. Durağan noktada  $\Delta S$  değeri sıfıra eşittir. Denklem (1.212) eşitliğinin sol tarafının sıfıra eşitlenmesi, integralin ve dolayısıyla integral içinin de sıfıra eşit olması manasına gelir. Buradan hareketle;

$$\frac{\partial L}{\partial x} - \frac{\partial}{\partial t} \left( \frac{\partial L}{\partial \dot{x}} \right) = 0 \quad (1.213)$$

$$\frac{\partial L}{\partial x} = \frac{\partial}{\partial t} \left( \frac{\partial L}{\partial \dot{x}} \right)$$

sonucuna ulaşılır. Elde edilen bu sonuç literatürde Euler-Lagrange denklemi olarak geçer<sup>(56)</sup>. Söz konusu eşitlik, bize süreci belirleyen Lagrange denklemini vermemekle beraber süreci belirleyen söz konusu Lagrange fonksiyonlar fonksiyonunun karakterini göstermektedir. İfade edilmesi gereken diğer bir husus da, elde edilen ölçütün  $x(t_1)$  ve  $x(t_2)$  arasındaki yolun en kısa ya da diğer bir ifade ile doğrusal olduğunu/olması gerektiğini göstermediğidir. Denklem (1.213), sadece söz konusu yol alınırken süreçte etki eden etmenler arasındaki ilişkinin nasıl olması gerektiğini göstermektedir.



**Şekil 1.111** Durağan nokta asgari, bükülme veya azami nokta olabilir.

<sup>i</sup>The principle of least action

<sup>ii</sup> Stationary point



Euler-Lagrange denkleminin doğadaki dinamik sistemlerin oluşumunda belirlediği karakteri elde etmek için Lagrange fonksiyonlar fonksiyonunu aşağıdaki gibi tanımlayabiliriz:

$$L = E_K - E_P \quad (1.214)$$

Burada  $E_K$  eylem içerisinde bulunan cismin kinetik enerjisini,  $E_P$  potansiyel enerjisini ifade eder. Söz konusu enerji değerlerini:

$$\begin{aligned} E_K &= \frac{1}{2} m \dot{x}^2 \\ E_P &= \varphi \end{aligned} \quad (1.215)$$

şeklinde açmak mümkündür. Denklem (1.215) dâhilindeki  $m$  cismin kütesini,  $\varphi$  potansiyel enerji değerini ifade eder. Verilen enerji denklemlerini Denklem (1.213) içerisinde yerlerine koyarsak;

$$\frac{\partial \left( \frac{1}{2} m \dot{x}^2 - \varphi \right)}{\partial x} = \frac{\partial}{\partial t} \left( \frac{\partial \left( \frac{1}{2} m \dot{x}^2 - \varphi \right)}{\partial \dot{x}} \right) \quad (1.216)$$

eşitliği elde edilir. Lagrange fonksiyonlar fonksiyonunun konuma göre türetilmesini;

$$\begin{aligned} \frac{\partial \left( \frac{1}{2} m \dot{x}^2 - \varphi \right)}{\partial x} &= \frac{\partial \left( \frac{1}{2} m \dot{x}^2 \right)}{\partial x} - \frac{\partial \varphi}{\partial x} \\ \frac{\partial \left( \frac{1}{2} m \dot{x}^2 - \varphi \right)}{\partial x} &= - \frac{\partial \varphi}{\partial x} \end{aligned} \quad (1.217)$$

şeklinde açabiliriz. Burada dikkat edilmesi gereken husus, konumun konum ve zamana göre türetilmesinin ifadeyi sonlandırmasıdır:

$$\frac{\partial^2 x}{\partial x \partial t} = \frac{\partial}{\partial t} \left( \frac{\partial x}{\partial x} \right) = 0 \quad (1.218)$$

Denklem (1.216)'nın sağ tarafını ise

$$\frac{\partial}{\partial t} \left( \frac{\partial \left( \frac{1}{2} m \dot{x}^2 - \varphi \right)}{\partial \dot{x}} \right) = \frac{\partial}{\partial t} (m \dot{x}) = m \frac{\partial^2 x}{\partial t^2} \quad (1.219)$$

şeklinde özetleyebiliriz.







Tahmin edilebileceği üzere potansiyel enerji ifadesi zamanın değil konumun bir fonksiyonudur. Denklem (1.217) ile Denklem (1.219) ifadelerinin Denklem (1.216) bağlamında dikkate alınması sonucu;

$$-\nabla\varphi = m\ddot{x} \quad (1.220)$$

şeklindeki nihai denklem elde edilebilir. Bu denklem I. Newton'a ait ünlü ikinci yasadır;

$$\mathbf{F} = m\mathbf{a}$$

Söz konusu yasa, açık haliyle kuvvetin kütle ve ivmenin çarpımına eşit olduğunu gösterir. Burada dikkat edileceği üzere kuvvet potansiyel enerjinin konumsal değişikliğini<sup>i</sup>, ivme ise konumun zamana göre ikinci türevini temsil etmektedir. Şimdiye kadar yapılan işlemler, yaşadığımız evrendeki önemli bir yasanın “en az eylem” ilkesinden hareketle elde edilebileceğini, eniyileme kavramının evrene ait yasaların belirlenmesinde dikkate alındığını göstermektedir.



---

<sup>i</sup> Korunum kuvvetleri İngilizce’de “conservative forces” olarak geçer. Söz konusu kuvvetler ortamdaki potansiyelin mekâna göre değişimi ile örtüştürülebilir.





- [1] Convex Optimization, S. Boyd ve L. Vandenberghe, Cambridge University Press, 7'nci Baskı, 2009.
- [2] The algebraic and geometric theory of quadratic forms; R. Elman, N. Karpenko ve A. Merkurjev, Colloquium Publications, 2008.
- [3] Advanced engineering mathematics, E. Kreyszig, John Wiley & Sons, Inc., 9'uncu Baskı, 2006.
- [4] Calculus concepts: An Applied Approach to the Mathematics of Change, D. R. LaTorre, J. W. Kenelly, I. B. Reed, L. R. Carpenter, C. R. Harris, Houghton Mifflin Comp., 4'üncü Baskı, 2008.
- [5] Response surface methodology: process and product optimization using designed experiments, R. H. Myers, D. C. Montgomery, C. M. Anderson-Cook, 3rd Edition, A John Wiley & Sons, Inc., Publication, 2009.
- [6] Engineering optimization: theory and practice, S.S. Rao, John Wiley and Sons, Inc., 3rd Edition, 1996.
- [7] Practical genetic algorithms, R. L. Haupt and S. E. Haupt, John Wiley & Sons, Inc., 2'nci Baskı, 2004.
- [8] Elements of dynamic optimization, A. C. Chiang, International Editions, McGraw-Hill, Inc., Publication, 1992.





- [1] “Al-Ghazali’s use of Avicenna’s Philosophy,” R. Frank, 2014. Alıntı: <<http://www.ghazali.org/articles/use-avicen-rmf.pdf>> [29 Ocak 2014].
- [2] “Essays of Theodicy on the Goodness of God, the Freedom of Man and the Origin of Evil,” G. Leibniz, 2014. Alıntı: <[http://en.wikipedia.org/wiki/Best\\_of\\_all\\_possible\\_worlds#cite\\_note-1](http://en.wikipedia.org/wiki/Best_of_all_possible_worlds#cite_note-1)> [29 Ocak 2014].
- [3] Merriam-Wenster sözlüğü, 2014. Alıntı: <<http://www.merriam-webster.com/dictionary/optimization>> [03 Şubat 2014].
- [4] Latin Dictionary and Grammer Aid, 2014. Alıntı: < <http://catholic.archives.nd.edu/cgi-bin/lookup.pl?stem=opt&ending=>> [03 Şubat ‘14].
- [5] Marmura, M. E., 1995, Ghazâlian Causes and Intermediaries, *Journal of the American Oriental Society*, 115, pp. 89–100.
- [6] History of Mathematics: Highways and Byways By Amy Dahan-Dalmédico, Jeanne Peiffer, Mathematical Association of America, 2010, sf. 97.
- [7] Pierre de Fermat, Encyclopedia Britannica, Alıntı: <<http://global.britannica.com/EBchecked/topic/204668/Pierre-de-Fermat>> [04 Şubat 2014].
- [8] Secretary Problem, wikipedia, Alıntı: <[http://en.wikipedia.org/wiki/Secretary\\_problem](http://en.wikipedia.org/wiki/Secretary_problem)> [04 Şubat 2014].
- [9] Rolling circles and balls, J. Baez. Alıntı: < [http://math.ucr.edu/home/baez/rolling/rolling\\_3.html](http://math.ucr.edu/home/baez/rolling/rolling_3.html)> [12 Şubat 2014].
- [10] The Bernoulli Family: Their Massive Contributions to Mathematics and Hostility Toward Each Other, Dung (Yom) Bui, Mohamed Allali, e-Research: A Journal of Undergraduate Work: Vol. 2 : No. 2 , Article 6, 2014.
- [11] Theodicy: Essays on the Goodness of God, the Freedom of Man and the Origin of Evil, G. W. Leibniz, Edited with an Introduction by Austin Farrer, Fellow of Trinity College, Oxford, Translated by E.M. Huggard from C.J. Gerhardt's Edition of the Collected Philosophical Works, 1985.
- [12] Newton’s Principia: The Mathematical Principles of Natural Philosophy, I. Newton; İngilizce’ye çeviren: Andrew Motte, New-York, Daniel Adee, 45 Liberty Street, 1845, sf. 328.
- [13] Some Historical Aspects of Honeycomb Mathematics, S. A. Baldwin, Bee Craft Digital, May 2012, sf. 29.
- [14] En Az Eylem İlkesi, E. Özmutlu, Kaygı : Uludağ Üniversitesi Fen Edebiyat Fakültesi Felsefe Dergisi, Bahar 2007, Sayı 8, ss. 173-179.
- [15] Soap Films and Minimal Surfaces, Alıntı: <<https://redlegagenda.com/2015/09/23/soap-films-and-minimal-surfaces/>> [20 Temmuz 2017].
- [16] Cauchy and the Gradient Method, C. Lemarechal, Documenta Mathematica · Extra Volume ISMP (2012) 251–254.
- [17] Josiah Willard Gibbs, Wikipedia, Alıntı: <[http://en.wikipedia.org/wiki/Josiah\\_Willard\\_Gibbs](http://en.wikipedia.org/wiki/Josiah_Willard_Gibbs)> [06 Şubat 2014].



- [18] Theory of Maxima and Minima, H. Hancock, the Atheneum Press, Ginn and Company, USA, 1917.
- [19] On Growth and Form, D. W. Thompson, Cambridge University Press: The Macmillan Company, 1945, sf. 6.
- [20] An illustration from “On growth and form”. Alıntı: < <http://charles.hamel.free.fr/knots-and-cordages/img/Darcy-fishes.jpg>> [13 Şubat 2014].
- [21] Rowland, Todd. "Manifold." From MathWorld--A Wolfram Web Resource, created by Eric W. Weisstein., Alıntı: <<http://mathworld.wolfram.com/Manifold.html>> [10 Şubat 2014].
- [22] Traviling salesman art, A. Westerdiep. Alıntı: < [http://www.drububu.com/illustration/tsp/images/tsp\\_circle\\_different\\_radius.png](http://www.drububu.com/illustration/tsp/images/tsp_circle_different_radius.png)> [13 Şubat 2014].
- [23] Linear Programming and Extensions, G. B. Dantzig, Princeton University Press, 1963, sf. 12.
- [24] The Oxford Handbook of the History of Mathematics, edited by Eleanor Robson, Jacqueline Stedall, Oxford University Press, 2009.
- [25] The eurovisionary: chaos theory, Alıntı: < <http://theurovisionary.blogspot.com.tr/2012/05/chaos-theory.html> > [08 Mart 2017].
- [26] Barricelli's Universe, G. Dyson, Make : Volume 08, 2006, sf. 190.
- [27] History of optimization, M. Kitti, Alıntı: < <http://www.mitrikitti.fi/opthist.html>> [12 Şubat 2014].
- [28] Multiobjective optimization: history and promise, O. L. de Weck, CJK-OSM 3: The Third China-Japan-Korea Joint Symposium on Optimization of Structural and Mechanical Systems ,October 30-November 2, 2004, Kanazawa, Japan, Invited keynote paper.
- [29] Rising fall, Olga Cuttell'e ait sanatsal eser.
- [30] Adaptation in Natural and Artificial Systems, J. H. Holland, 1975, University of Michigan Press, Ann Arbor.
- [31] Optimization, Learning and Natural Algorithms, M. Dorigo, 1992, Ph. D. Thesis, Politecnico di Milano, Italy, 1992.
- [32] An Idea Based On Honey Bee Swarm for Numerical Optimization, D. Karaboga, Technical Report-Tr06, October, 2005.
- [33] Güncel sürü zekası optimizasyon algoritmaları, S. Akyol ve B. Alataş, Nevşehir Üniversitesi Fen Bilimleri Enstitüsü Dergisi 1 (2012) 36–50.
- [34] Algebra, Wikipedia: the free encyclopedia. Alıntı: < <http://en.wikipedia.org/wiki/Algebra> > [19 Mart 2014].
- [35] The Nine Chapters on the History of Chinese Mathematics, Wikipedia: the free encyclopedia, Alıntı: < [http://en.wikipedia.org/wiki/The\\_Nine\\_Chapters\\_on\\_the\\_History\\_of\\_Chinese\\_Mathematics](http://en.wikipedia.org/wiki/The_Nine_Chapters_on_the_History_of_Chinese_Mathematics)> [08 Mart 2017].
- [36] The Nine Chapters on the History of Chinese Mathematics, R. Brandenburg and K. Nevenzeel, 15 July 2007 Alıntı: < [http://www.astro.rug.nl/~nevenzeel/Study/PGvdW\\_t=9C\\_HCM\\_a=RB,KN.pdf](http://www.astro.rug.nl/~nevenzeel/Study/PGvdW_t=9C_HCM_a=RB,KN.pdf)> [08 Mart 2017].
- [37] Madhava of Sangamagrama, From the Wikipedia, the free encyclopedia. Alıntı: < [http://en.wikipedia.org/wiki/Madhava\\_of\\_Sangamagrama](http://en.wikipedia.org/wiki/Madhava_of_Sangamagrama)> [06 Mayıs 2014].
- [38] Project Title: Pseudo-Random Number Generator, S. Murkute. Alıntı: < [http://issuu.com/sheetal\\_murkute/docs/simulation\\_research\\_report](http://issuu.com/sheetal_murkute/docs/simulation_research_report)> [16 Mayıs 2014].



- [39] List of Random Number Generators, From the Wikipedia, the free encyclopedia. Alıntı: < [http://en.wikipedia.org/wiki/List\\_of\\_random\\_number\\_generators](http://en.wikipedia.org/wiki/List_of_random_number_generators)> [15 Mayıs 2014].
- [40] S. Leary, A. Bhaskar, and A. Keane, Optimal orthogonal-array-based Latin hypercubes, Journal of Applied Statistics, Vol. 30, No. 5, 2003, 585–598.
- [41] Design of experiments via Taguchi methods: orthogonal arrays, The Michigan Chemical Process Dynamics and Controls Open Textbook. Alıntı: < [https://controls.engin.umich.edu/wiki/index.php/Design\\_of\\_experiments\\_via\\_taguchi\\_methods:\\_orthogonal\\_arrays](https://controls.engin.umich.edu/wiki/index.php/Design_of_experiments_via_taguchi_methods:_orthogonal_arrays)> [04 Haziran 2014].
- [42] Design of Modern Heuristics Principles and Application, Rothlauf, F., Springer-Verlag Berlin Heidelberg, 2011. Sf. 8-9.
- [43] Aircraft Structures for Engineering Students, T. H. G. Megson, 3üncü Baskı, Butterworth-Heinemann, 2003.
- [44] Alıntı: < [http://www.lakeglennvillemtntop.com/Beech\\_Starship.jpg](http://www.lakeglennvillemtntop.com/Beech_Starship.jpg)> [11 Mayıs 2015].
- [45] Mechanical Design, P. Childs, 2nci Baskı, Elsevier Butterworth-Heinemann, 2004.
- [46] Alıntı: < <http://www.eaa1000.av.org/pix/beeceh18/flapdown.jpg>> [11 Mayıs 2015].
- [47] Alıntı:<<http://www.cats.rwth-aachen.de/research/images/f-hirenasd-wing-winglet-cp3d-wolegend.png>> [11 Mayıs 2015].
- [48] Thermodynamics: An Engineering Approach, Y. A. Çengel, M. A. Boles, 8inci Baskı, McGraw-Hill Company, 2015.
- [49] Alıntı: < <http://turbomachinery.man.eu/products/compressors/axial>> [28 Şubat 2017].
- [50] Alıntı:<[http://www.woodweb.com/knowledge\\_base\\_images/bah/veneer\\_vacuum\\_mold\\_convex\\_or\\_concave\\_02.png](http://www.woodweb.com/knowledge_base_images/bah/veneer_vacuum_mold_convex_or_concave_02.png)> [29 Mayıs 2015].
- [51] “A Simple equation for describing the temperature dependent growth of free-floating macrophytes,” T. Heide, R.M.M. Roijackers, E.H. Nes, ve E.T.H.M. Peeters, Aquatic Botany 84 (2006) 171-175.
- [52] “Vibrational GA enhanced with neural networks in RCS problems,” Y. Volkan Pehlivanoglu ve Oktay Baysal, Aircraft Engineering and Aerospace Technology: An International Journal, Volume 83 · Number 1 · 2011 · 43–48.
- [53] Alıntı: < [http://www.deskeng.com/de/img/2012-simulia-community-conference-wrap-up\\_3.jpg](http://www.deskeng.com/de/img/2012-simulia-community-conference-wrap-up_3.jpg)> [22 Mayıs 2015].
- [54] El Ghazali Talbi, Metaheuristics: from design to implementation, A John Wiley and sons Inc. Publication, 2009, sf. 11-14.
- [55] Principle of least action, Alıntı:<[http://physics-essays.birukou.net/\\_/rsrc/129505635\\_5330/principle-of-least-action/dali\\_clock.jpg](http://physics-essays.birukou.net/_/rsrc/129505635_5330/principle-of-least-action/dali_clock.jpg)> [02 Mart 2017].
- [56] Derivation of the Euler-Lagrange Equation. Alıntı:<<http://www.lepp.cornell.edu/~yuvalg/p3314/E-L.pdf>> [17 Şubat 2014].

Örnek uygulama resmi; Alıntı:<<http://www.michaelheterington.com.au/wp-content/uploads/2017/02/japanesepicture.jpg>> [10 Haziran 2017].





## DETERMINİSTİK YÖNTEMLER

(1)



# 2

“ Özgür iradeye inanmak zorundayız... Başka seçeneğimiz yok.”

Isaac Singer (2)

		Sayfa
2.1	Giriş	176
2.2	Grafik yöntemi	176
2.3	Doğrusal programlama yöntemleri	188
2.4	Doğrusal olmayan programlama yöntemleri	236
2.4.1	Analitik yöntemler	236
2.4.2	Sayısal yöntemler	257
	İrfan Damlası	390
	Kaynakça	391



## 2.1 Giriş



eterminizmi veya diğer adıyla belirlenimcilik kavramı, sebep-sonuç ilişkisi üzerine kurulu bir kavram olup belirli nedenlerin belirli sonuçları doğuracağını kabul etmektedir. Eniyileme süreçlerinde deterministik yöntemden kastedilen, yöntemin içeriğinde rastsal sayı kullanımının söz konusu olmamasıdır. Bu meyanda deterministik yöntemin uygulandığı sorunun çözümü, her defasında aynı sonucu verir. Deterministik yöntemler başlığı altında öncelikle grafik yöntemi üzerinde durulacak, bilahare doğrusal programlama yöntemlerine geçilecektir. Müteakiben doğrusal olmayan problemlerin çözümünde kullanılan yöntemlere değinilecektir. Bu yöntemlerin büyük çoğunluğu gradyan bilgisine ihtiyaç duyarken bazıları gradyan bilgisine gerek duymadan çözümü bulabilirler. Gradyan bilgisine üzerine çözümü arayan yöntemleri, analitik ve sayısal yöntemler olarak tasnif etmek mümkündür. Girişte dikkat çekmek istediğimiz diğer bir husus da grafik yönteminin, basit olmasına rağmen doğrusal veya doğrusal olmayan problem türlerinde kullanılabilir olmasıdır.

## 2.2 Grafik yöntemi

Optimizasyon problemlerinin çözümünden ziyade çözüm yöntemlerinin anlaşılması ve somut çizimlerle gösterilmesi adına grafik yöntemi tercih edilen bir yöntemdir. En fazla iki değişkenin dikkate alınabileceği grafikte çözüm sürecinde, gerek amaç fonksiyonu ve gerekse de kısıtlar aynı eksen takımı dahilinde dikkate alınarak olası çözüm bölgesi ve çözüm bölgesindeki optimal nokta tespit edilmeye çalışılır. Bu meyanda her bir fonksiyon, değişken değerlerine göre grafik üzerinde çizilir ve müteakiben şekil üzerinden minimum veya maksimum nokta tespit edilir. Genellikle optimal nokta, kısıtların amaç fonksiyonu ile kesiştiği yerlerde bulunur.

Tek değişkenin söz konusu olduğu sorunlarda yatay eksen değişken eksenini, dikey eksen ise amaç fonksiyonunu temsil eder. İki değişkenin dikkate alındığı problemlerde yatay düzlemi temsil eden eksenler genellikle değişkenleri, dikey eksen amaç fonksiyonunu gösterir. Daha fazla değişken sayısının olduğu sorunlarda grafik yöntemini kullanmak pek mümkün değildir. Zira içinde yaşadığımız üç boyutlu mekânsal gösterim kısıtı çözümdeki en önemli engeldir. Grafik yönteminin kolay anlaşılır olması yanında diğer bir getirisi, doğrusal olmayan sorunlarda da kullanılabilir olmasıdır. Gerek doğrusal ve gerekse de doğrusal olmayan amaç fonksiyonu ile çok sayıda kısıt fonksiyonu aynı eksen takımında çizilerek gösterilebilir ve insan zekasının görsel

---

<sup>1</sup> Determinism



algılama kapasitesi sayesinde kolayca optimal çözüm noktası tespit edilebilir. Bununla beraber çizim sadedinde kullanılan adım büyüklüğü ve çizim programının özelliği çözümün bulunmasında önemli etmenlerdir. Bazı yazılımlar (örneğin Matlab yazılım programı) ekran üzerinde işaretlenen noktayı doğrudan gösterebilmekte ve nokta koordinatlarını ekrana getirebilmektedir. Yöntemin hassasiyeti, çizim esnasında dikkate alınan adım büyüklüğüne bağlıdır. Dolayısıyla hassas çözüm eldesi için hassas çizim yapmak gerekir. Bu ise çizim için dikkate alınması gereken nokta sayısını bir hayli artırabilir. Grafik yönteminin anlaşılması bağlamında müteakip örnek uygulamalara değinilmektedir.



Örnek uygulama

Bir petrol türevi dönüşüm rafinerisinde, doğadan çıkarılan petrolden “regular” ve “premium” adı altında iki tür yakıt üretildiğini düşünelim.

Söz konusu üretim hattında “regular” gazın tonu  $7 \text{ m}^3$ , “premium” gazın tonu  $11 \text{ m}^3$  hacim kaplamaktadır. Fabrikanın haftalık dönüşüm kapasitesi  $77 \text{ m}^3$ 'tür. Diğer taraftan “regular” gaz üretimi ton başına 10 saat, “premium” gaz üretimi 8 saat sürmektedir. Fabrikanın haftalık çalışma süresi kapasitesi 80 saattir. İlave olarak her bir gaz için depolama kapasitesi de söz konusu olup bu değer “regular” gaz için 10 ton, “premium” gaz için 8 tondur. Fabrikanın ton başına kazandığı kâr ise “regular” gazda 150 TL, “premium” gazda 180 TL'dir<sup>(3)</sup>. Buna göre kârını doruklaştırmak isteyen fabrika haftalık bazda hangi gaz türünden kaç ton üretim yapmalıdır?



Şekil 2.1 Petrol dönüşüm fabrikası<sup>(4)</sup>.

Eldeki veriler ışığı altında tasarım yöneyini tanımlamak, sorunun çözümü için iyi bir başlangıç olarak görülebilir. Buna göre tonaj olarak  $x_1$  “regular” gaz üretim miktarını,  $x_2$  “premium” gaz üretim miktarını göstermek üzere tasarım yöneyini;

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$





şeklinde tanımlayabiliriz. Kârın doruklaştırılması hedefi güdüldüğüne göre amaç fonksiyonunu:

$$\underset{x}{maks} f(x) = [150 \quad 180] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 150x_1 + 180x_2$$

denklemi ile temsil etmek mümkündür. Diğer taraftan fabrikanın haftalık dönüşüm kapasitesi dikkate alındığında aşağıdaki gibi bir kısıt yazılabilir:

$$7x_1 + 11x_2 = 77$$

Benzer şekilde çalışma süresinin dikkate alındığı kısıt ise;

$$10x_1 + 8x_2 = 80$$

denklemi ile ifade edilebilir. İlave olarak her bir gaz türü için depolama kapasiteleri ayrı ayrı düşünüldüğünde;

$$\begin{aligned} 0 &\leq x_1 \leq 10 \\ 0 &\leq x_2 \leq 8 \end{aligned}$$

eşitsizlik kısıt denklemlerini yazabiliriz. Amaç ve kısıtlar dâhil yazılan tüm denklemlerin beraberce dikkate alınması halinde eniyileme sorununu;

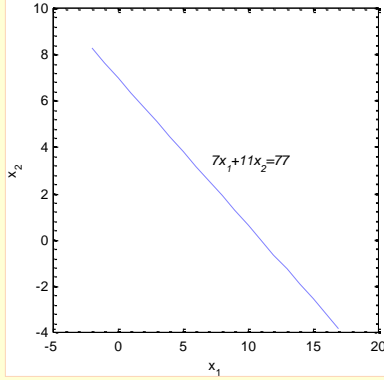
$$\underset{x}{maks} f(x) = 150x_1 + 180x_2$$

Kısıt

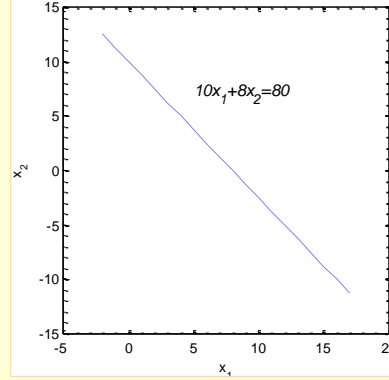
$$\begin{aligned} 7x_1 + 11x_2 &= 77 \\ 10x_1 + 8x_2 &= 80 \\ 0 &\leq x_1 \leq 10 \\ 0 &\leq x_2 \leq 8 \end{aligned}$$

şeklinde tanımlamak mümkündür. Görüldüğü üzere tüm denklemler doğrusal denklem niteliğindedir. Burada dikkat edilmesi gereken diğer bir konu da denklemlerin üç boyutlu bir uzayda düzlemleri temsil etmesidir. Dolayısıyla geometrik bağlamda ortaya çıkan şeklin bir çokyüzlü olduğu söylenebilir. Söz konusu çokyüzlünün sınırlarını kısıtsal düzlemler oluşturmaktadır. Amaç fonksiyonu dâhil doğrusal denklem takımı içeren bu problemi grafiğe dökerek çözmek mümkündür. Buna göre verilen kısıtların grafik çizimleri müteakip şekillerde görülmektedir:

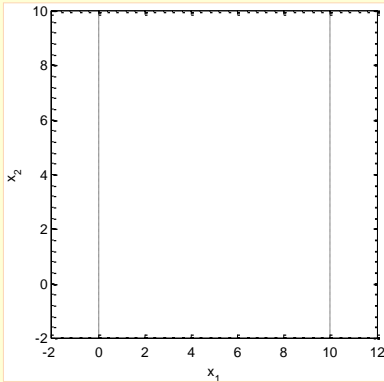




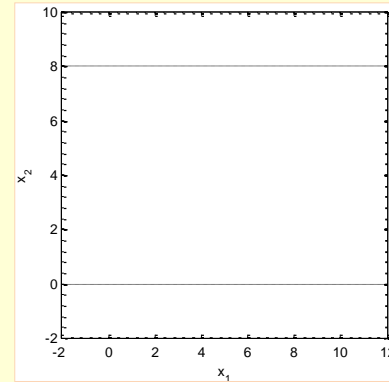
(a)  $7x_1 + 11x_2 = 77$  kısıtına ait grafik.



(b)  $10x_1 + 8x_2 = 80$  kısıtına ait grafik.



(c)  $0 \leq x_1 \leq 10$  kısıtına ait grafik.



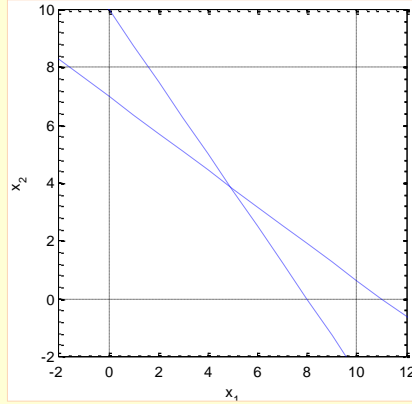
(ç)  $0 \leq x_2 \leq 8$  kısıtına ait grafik.

**Şekil 2.2** Kısıtların grafiksel gösterimi.

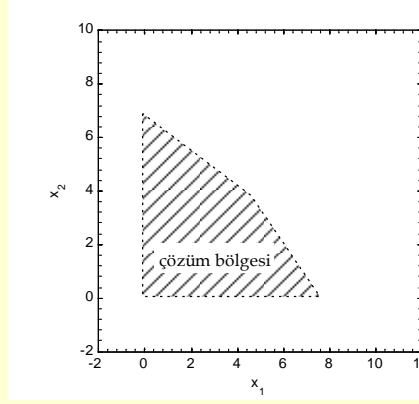
Çözüm bağlamında mevcut kısıtların beraber dikkate alınması halinde, Şekil 2.3(a)'daki gibi bir grafik ve Şekil 2.3(b)'deki gibi bir çözüm bölgesi elde edilebilir. Buna mukabil olası tasarım yöneyi değerlerine göre elde edilmesi muhtemel amaç fonksiyonu değerlerini gösteren yüzey ise Şekil 2.3(c)'de görülmektedir. Verili kısıtlar ile amaç fonksiyonu yüzeyinin beraber değerlendirilmesi neticesinde elde edilebilecek azami kârın [4.88 3.88] noktası olduğu kolayca bulunabilir. Zira bu nokta ilk iki kısıtın kesiştiği nokta olup aynı zamanda tasarım yöneyine ait alt ve üst değer gereklilerini de karşılamaktadır. Söz konusu noktaya karşılık gelen kâr değeri, nokta koordinatlarının amaç fonksiyonunda dikkate alınması ile 1433.3 TL olarak



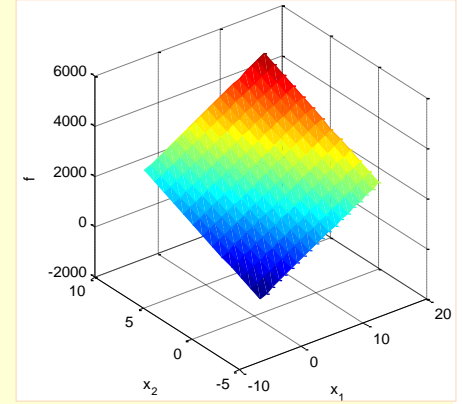
bulunabilir. Çözüm noktası ve onun amaç fonksiyonu yüzeyindeki yerini gösteren grafik çizimleri müteakip şekillerde verilmiştir.



(a) Tüm kısıtların dikkate alındığı grafik.

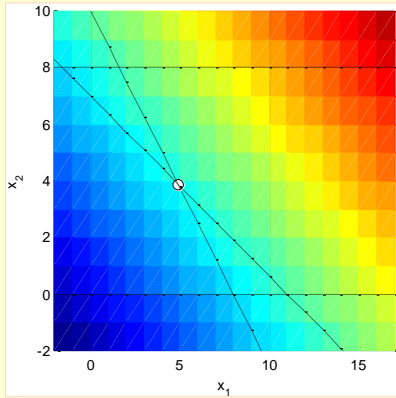


(b) Kısıtların çerçevelediği çözüm bölgesi.

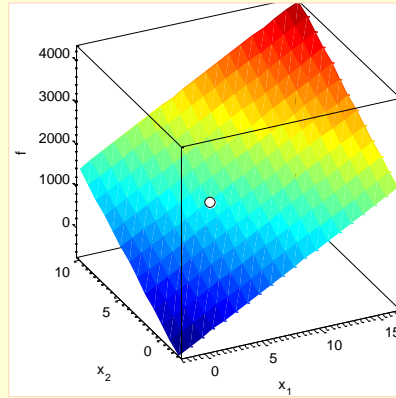


(c) Amaç fonksiyonu yüzeyi.

Şekil 2.3 Olası çözüm uzayı gösterimi.



(a) Kısıtların çevrelediği çözüm bölgesi ve azami kârı veren nokta (üsttten görünüş).



(b) Azami kârın elde edildiği çözüm noktasının amaç fonksiyonu yüzeyindeki konumu.

Şekil 2.4 Amaç fonksiyonu yüzeyi dahilinde optimal çözüm noktası.



Şekil 2.5 Mikrodalga fırın üretim fabrikası<sup>(5)</sup>

Mikrodalga fırın üreten bir fabrikada iki tür fırın imalatı yapılmakta olup fırın tiplerinin ilki standart fırın, diğeri lüks tip fırındır. Üretim hatlarında fırınların mekanik ve elektronik aksamları için ayrı ayrı zaman harcanmakta ve ayrıca alt sistemlerin montajı için de ilave süre gerekmektedir. Müteakip çizelgede aylık bazda her bir fırın tipi için harcanan işgücü süreleri ve kapasite değerleri görülmektedir<sup>(6)</sup>. Şirketin maliye bölümü, standart tipteki her bir fırından yaklaşık 50 TL, lüks tipteki her bir fırından 60 TL kâr elde edildiğini beyan etmektedir. Buna göre kârın doruklaştırılması amacıyla her bir fırın tipinden kaçar adet üretim yapılmalıdır?

	Standart (hr <sup>i</sup> )	Lüks (hr)	Kapasite (hr)
Mekanik aksam	4	4	560
Elektronik aksam	3	2	400
Genel montaj	2	4	400

Bir önceki örnekte olduğu gibi  $x_1$  adet olarak standart tip fırın üretim miktarını,  $x_2$  lüks tip fırın üretim miktarını göstermek üzere tasarım yöneyini;

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

şeklinde tanımlayabiliriz. Kârın maksimizasyonu amacı güdüldüğüne göre amaç fonksiyonunu:

$$\max_{\mathbf{x}} f(\mathbf{x}) = [50 \quad 60] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 50x_1 + 60x_2$$

denklemleri ile temsil etmek mümkündür. Diğer taraftan fabrikanın mekanik aksam için harçayabileceği aylık işgücü kapasitesi dikkate alındığında aşağıdaki gibi bir kısıt yazılabilir:

$$4x_1 + 4x_2 = 560$$

<sup>i</sup> İngilizce’de “hour” kelimesinin kısaltması olup saat süresi olarak geçer.





Benzer şekilde elektronik aksam için harcanabilecek aylık işgücü kapasitesinin dikkate alındığı kısıt ise;

$$3x_1 + 2x_2 = 400$$

denklemini ile ifade edilebilir. İlave olarak her bir fırın türü için montaj kapasitesi düşünüldüğünde;

$$2x_1 + 4x_2 = 400$$

eşitlik kısıt denklemini yazabiliriz. Amaç ve kısıtlar dâhil yazılan tüm denklemlerin beraberce dikkate alınması halinde eniyileme sorununu;

$$\max_x f(x) = 50x_1 + 60x_2$$

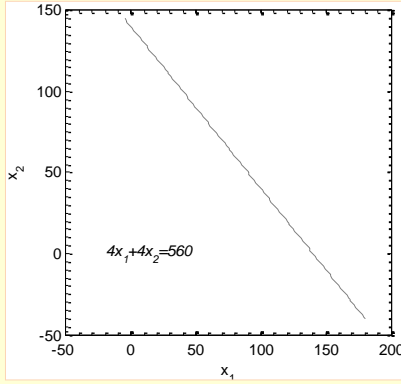
Kısıt

$$4x_1 + 4x_2 = 560$$

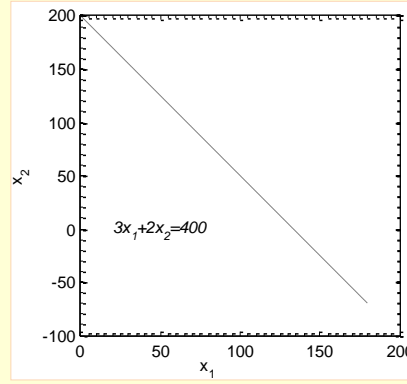
$$3x_1 + 2x_2 = 400$$

$$2x_1 + 4x_2 = 400$$

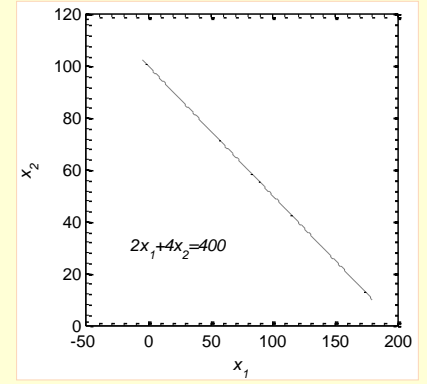
şeklinde tanımlamak mümkündür.



(a)  $4x_1 + 4x_2 = 560$  kısıtına ait grafik.



(b)  $3x_1 + 2x_2 = 400$  kısıtına ait grafik.



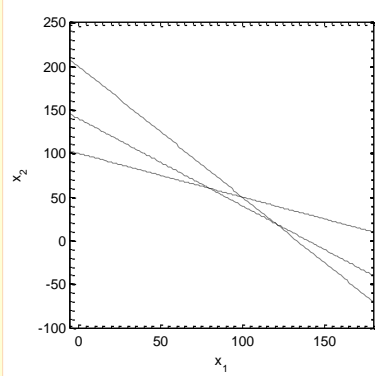
(c)  $2x_1 + 4x_2 = 400$  kısıtına ait grafik.

**Şekil 2.6** Kısıtların grafiksel gösterimi.

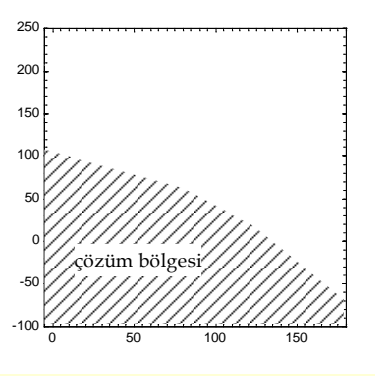
Görüldüğü üzere tüm denklemler doğrusal denklem niteliğindedir. Dolayısıyla geometrik bağlamda bir çokyüzlü söz konusu olup amaç fonksiyonu dâhil doğrusal denklem takımı içeren bu problemi grafiğe dökerek çözmek mümkündür. Çözümün eldesi sadedinde bütün kısıtların dikkate alınması gerekir. Bu nedenle tüm kısıtların beraberce oluşturduğu çözüm uzayını Şekil 2.7(a) ve Şekil 2.7(b)'de verilen



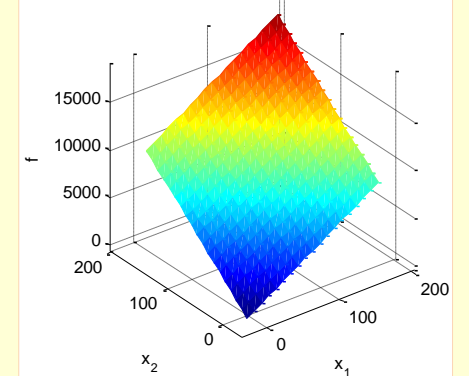
grafiklerle görselleştirebiliriz. Benzer şekilde olası tasarım yöneyi değerlerine göre amaç fonksiyonu değerleri de Şekil 2.7(c)'de görülmektedir.



(a) Tüm kısıtların dikkate alındığı grafik.

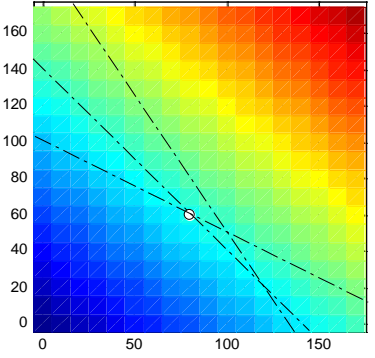


(b) Kısıtların çerçevelediği çözüm bölgesi.

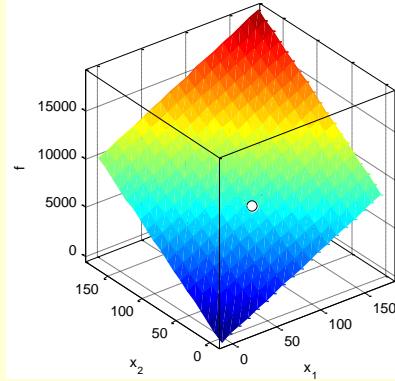


(c)  $f(\mathbf{x}) = 50x_1 + 60x_2$  amaç fonksiyonu yüzeyi.

**Şekil 2.7** Olası çözüm uzayı ve amaç fonksiyonu gösterimleri.



(a) Kısıtların çevrelediği çözüm bölgesi ve azami kârı veren nokta (üstten görünüş).



(b) Azami kârın elde edildiği çözüm noktasının amaç fonksiyonu yüzeyindeki konumu.

**Şekil 2.8** Amaç fonksiyonu yüzeyi dahilinde optimal çözüm noktası.

Çözüm bölgesi ile olası amaç fonksiyonu yüzeyinin beraberce değerlendirilmesi neticesinde Şekil 2.8'deki gibi bir optimal nokta tayini yapılabilir. Buna göre çözüm uzayı dâhilinde en fazla kârın elde edilmesini



sağlayan nokta, ilk ve üçüncü kısıtın kesiştiği nokta olup kârı doruklaştıran üretim değerleri sırası ile (80, 60) adettir. Söz konusu üretim miktarlarına karşılık gelen kâr ise 7600 TL'dir.



Kil esaslı malzeme kullanarak tamamen el becerisine dayalı kâse ve kahve kupası üreten bir imalathanede, kâse ve kupa üretim miktarları arttıkça her bir üründen elde edilen kârın düştüğü belirtilmektedir. Kâse üretim miktarı  $x_1$  olmak üzere her bir kâsedan elde edilen kâr;

$$5 + 0.1x_1^2$$

denklemlerle tanımlanmaktadır. Benzer şekilde kupa üretim miktarı  $x_2$  olmak üzere her bir kupadan elde edilen kâr;

$$6 + 0.2x_2$$

denklemlerle ifade edilmektedir. Burada belirtilen denklemlere TL olarak değer atfedilmektedir. Diğer taraftan üretim kapasitesi, işçi işgücüne göre sınırlı olup kâse üretimi için 1 saat, kupa üretimi için 2 saat işgücü harcanmakta, günlük imalathane toplam işgücü kapasitesi 50 saat olarak verilmektedir<sup>(7)</sup>. Verilenlere göre imalathanenin kârını doruklaştırmak için günlük üretim miktarları ne olmalıdır?



Şekil 2.9 Çömlek üretim imalathanesi<sup>(8)</sup>

Çözüm dâhilinde tasarım yöneyini;

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

şeklinde tanımlayabiliriz. Kârın maksimizasyonu hedefi güdüldüğüne göre amaç fonksiyonunu:





$$\max_x f(x) = [5 + 0.1x_1^2 \quad 6 + 0.2x_2] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 5x_1 + 0.1x_1^3 + 6x_2 + 0.2x_2^2$$

denklemi ile temsil etmek mümkündür. Diğer taraftan imalathanenin günlük işgücü kapasitesi dikkate alındığında aşağıdaki gibi bir kısıt yazılabilir:

$$x_1 + 2x_2 = 50$$

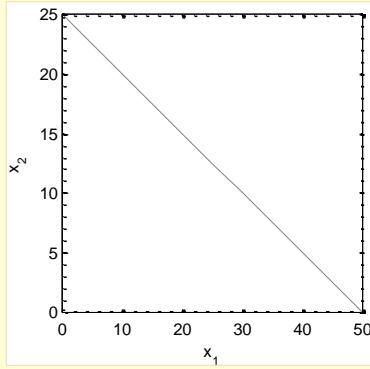
Amaç ve kısıt dâhil yazılan denklemlerin beraberce dikkate alınması halinde eniyileme sorununu;

$$\max_x f(x) = 5x_1 + 0.1x_1^3 + 6x_2 + 0.2x_2^2$$

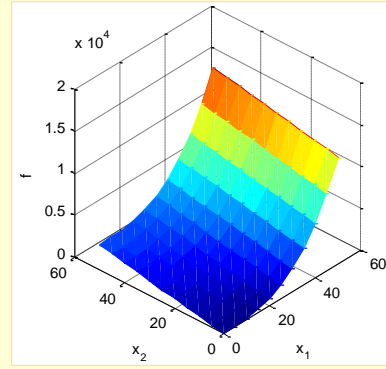
Kısıt

$$x_1 + 2x_2 = 50$$

şeklinde tanımlamak mümkündür. Daha önce çözülen iki sorunun aksine, buradaki sorunun tanımından da görüldüğü üzere amaç fonksiyonu doğrusal olmayan denklem niteliğindedir. Dolayısıyla amaç fonksiyonu ve kısıtlar bir çokyüzlüden ziyade sınırlandırılmış eğrisel bir yüzeyi göstermektedir. Doğrusal olmayan denklem içeren bu sorunu grafiğe dökerek çözmek mümkündür. Buna göre eldeki tek kısıtı ve olası amaç fonksiyonu yüzeyini aşağıdaki gibi grafik ortamında gösterebiliriz;



(a)  $x_1 + 2x_2 = 50$  kısıtına ait grafik.



(b)  $f(x) = 5x_1 + 0.1x_1^3 + 6x_2 + 0.2x_2^2$  amaç fonksiyonu.

**Şekil 2.10** Kısıt doğrusu ve amaç fonksiyonu yüzeyi gösterimi.

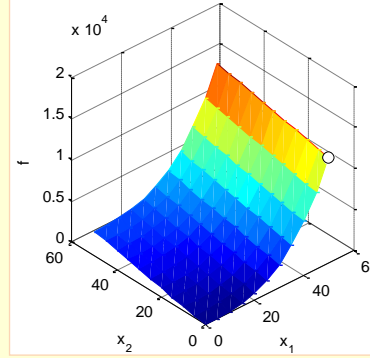
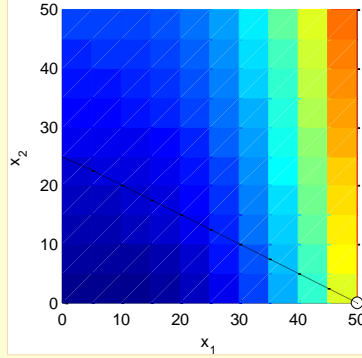
Tanımlı kısıt ile amaç fonksiyonu yüzeyinin değerlendirilmesi halinde Şekil 2.11(a)'daki gibi bir bölge ve bu bölgedeki azami kâr noktası tespit edilmiş olur. Buna göre üretim dâhilinde (50, 0) sayılarını dikkate alarak







azami 12750 TL'lik bir kâr elde etmek mümkündür. Söz konusu noktanın amaç fonksiyonu üzerindeki konumu Şekil 2.11(b)'de görülmektedir.



(a) Kısıtın çevrelediği çözüm bölgesi ve azami kârı veren nokta, (üsttten görünüş). (b) Azami kârın elde edildiği çözüm noktasının amaç fonksiyonu yüzeyindeki konumu.

Şekil 2.11 Amaç fonksiyonu yüzeyi dahilinde optimal çözüm noktası.

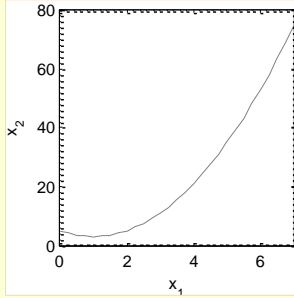


Kısıtları doğrusal olmayan nitelikte örnek bir eniyileme sorununu;

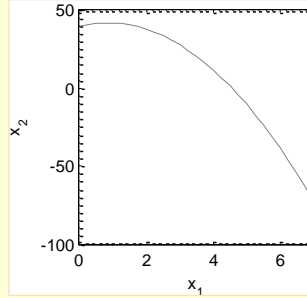
$$\begin{aligned} \max_x f(x) &= 5x_1 + 7x_2 \\ \text{Kısıt} \quad 2x_2 - 8x_1 - 4x_1^2 &\leq 10 \\ x_2 - 5x_1 + 3x_1^2 &\geq 40 \\ x_1 &\geq 1 \end{aligned}$$

şeklinde tanımlamak mümkündür. Görüldüğü üzere denklemlerde iki değişken olup verili kısıtlardan ilk ikisi 2<sup>nci</sup> dereceden denklemleri içermektedir. Buna mukabil amaç fonksiyonu ile verili 3<sup>üncü</sup> kısıt ise doğrusal denklem niteliğindedir. Kısıtların grafik gösterimi Müteakip şekillerde görülmektedir;

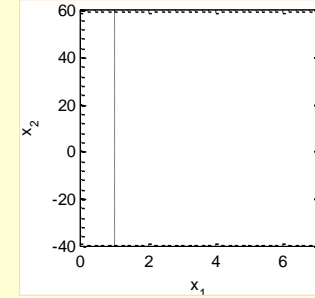




(a)  $2x_2 - 8x_1 - 4x_1^2 \leq 10$  kısıtına ait grafik.



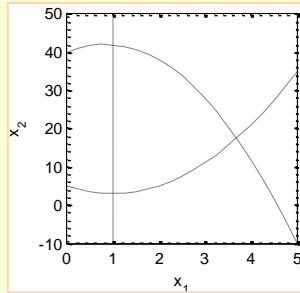
(b)  $x_2 - 5x_1 + 3x_1^2 \geq 40$  kısıtına ait grafik.



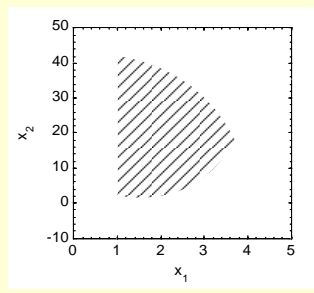
(c)  $x_1 \geq 1$  kısıtına ait grafik.

**Şekil 2.12** Kısıt fonksiyonlarının gösterimi.

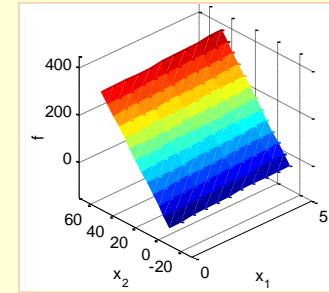
Tanımlı kısıtların çevrelediği çözüm uzayı Şekil 2.13(a) ve (b)'de görülmektedir. Buna mukabil olası amaç fonksiyonu yüzeyi de Şekil 2.13(c)'de çizilmiştir. Verili kısıtların ve amaç fonksiyonu değerlerinin beraberce dikkate alınması halinde azami değerini (1, 42) noktasında elde edilebileceği Şekil 2.14(a)'da görülmektedir. Buna karşılık gelen amaç fonksiyonu değeri ise 299 olup Şekil 2.14(b) dâhilinde söz konusu noktanın amaç fonksiyonu yüzeyindeki konumu görülmektedir.



(a) Tüm kısıtların dikkate alındığı grafik.



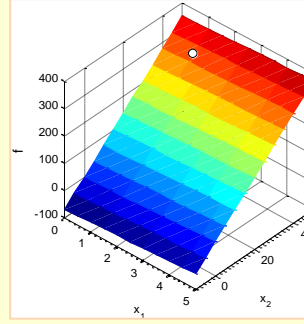
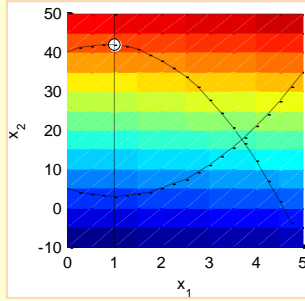
(b) Kısıtlarla çevrili çözüm uzayı.



(c)  $f(\mathbf{x}) = 5x_1 + 7x_2$  amaç fonksiyonu.

**Şekil 2.13** Olası çözüm uzayı ve amaç fonksiyonu yüzeyi gösterimi.





(a) Kısıtın çevrelediği çözüm bölgesi ve azami değeri veren nokta, (üstttten görünüş).

(b) Azami değerin elde edildiği çözüm noktasının amaç fonksiyonu yüzeyindeki konumu.

Şekil 2.14 Çözüm bölgesi ve amaç fonksiyonu yüzeyi dâhilinde en uygun çözüm noktası.



### 2.3 Doğrusal programlama yöntemleri

Daha önce amaç fonksiyonu ile kısıtların doğrusal karaktere sahip olması halinde söz konusu sorunun doğrusal eniyileme sorunu olarak tanımlanabileceği ifade edilmişti. Buna göre bir doğrusal programlama problemini;

$$\begin{aligned} & \min_{\mathbf{x}} f(\mathbf{x}) = \mathbf{c}^T \mathbf{x} & (2.1) \\ \text{Kısıt} & \\ & A\mathbf{x} \begin{cases} <, > \\ = \\ <=, >= \end{cases} \mathbf{b} \\ & \mathbf{x} \begin{cases} <, > \\ = \\ <=, >= \end{cases} 0 \end{aligned}$$

şeklinde ifade etmek mümkündür. Burada  $\mathbf{c}$  amaç fonksiyonuna ait tasarım parametre katsayılarını,  $\mathbf{x}$  tasarım yöneyini,  $A$  kısıtlar dâhilinde bulunan sabit tasarım parametre katsayılarının oluşturduğu katsayılar matrisini ve  $\mathbf{b}$  kısıtlara ait sabit değerleri temsil etmektedir.



Denklem 2.1 dâhilinde kısaca ifade edilen denklemleri,  $n$  deęişkene sahip tasarım yöneyi ile  $m$  tane kısıtın dikkate alındığı sorunda daha açık bir şekilde göstermek gerekirse;

$$\begin{aligned} \min_x f(x) &= c_1x_1 + c_2x_2 + \dots + c_nx_n && (2.2) \\ \text{Kısıt} & && \\ a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n & \begin{cases} <, > \\ = \\ <, > \end{cases} b_1 && \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n & \begin{cases} <, > \\ = \\ <, > \end{cases} b_2 && \\ \vdots & && \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n & \begin{cases} <, > \\ = \\ <, > \end{cases} b_m && \\ & && \\ & && \\ & && \\ & && \\ x_i & \begin{cases} <, > \\ = \\ <, > \end{cases} 0 && \end{aligned}$$

yazılabilir. Doğrusal eniyileme sorunlarının çözümünde kullanılan yöntemlere literatürde doğrusal programlama<sup>i</sup> yöntemleri adı verilmektedir. Doğrusal programlama terimi, bu konuda önemli çalışmaları olan George B. Dantzig'e aittir. Dantzig'in ilk dönemlerinde “doğrusal yapılarda programlama<sup>ii</sup>” olarak adlandırdığı sorun, daha sonra kısaltılarak doğrusal programlama biçiminde kullanılmaya başlandı. Doğrusal programlama, matematiksel programlamanın bir alt dalı olup oldukça geniş bir çözüm yelpazesinden bahsedilebilir. Söz konusu çözüm yöntemlerini başlıca iki grupta incelemek mümkündür. Bunlardan ilki “taban deęişimi<sup>iii</sup>” yöntemleri, dięer grup ise “içsel nokta deęişimi<sup>iiii</sup>” yöntemleridir.

Taban deęişimi yöntemlerindeki yaklaşım, olası tasarım uzayını çevreleyen noktalardan hareketle amaç fonksiyonunu minimize eden optimal noktanın bulunmasıdır. Dolayısıyla kısıtlarla sınırlandırılmış çokyüzlü niteliğindeki geometrik şeklin kesişim noktaları, her defasında belli bir sistematik dâhilinde denenerek en iyi çözüme ulaşılmaya çalışılır. Böyle bir yaklaşımın olumsuz olarak görülebilecek tarafı, çokyüzlünün dış yüzeyinden hareket edilmesi, bu nedenle de çözüm

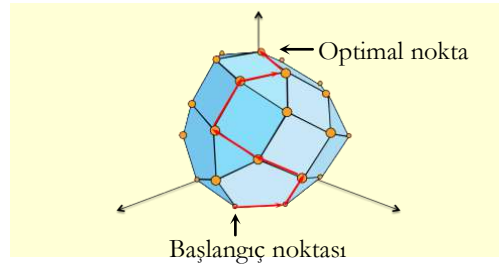
<sup>i</sup> Linear programming: LP

<sup>ii</sup> Basis exchange methods

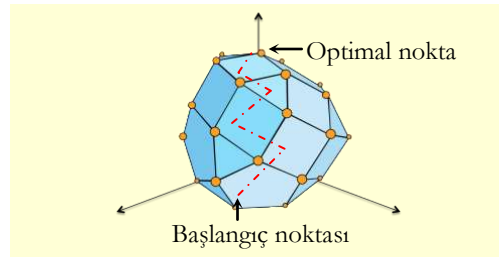
<sup>iii</sup> Interior point methods



için alınacak yolun uzun olmasıdır. Nitekim taban değişimi bazı yöntemler üssel zaman karmaşıklığına sahip yöntemlerdir. Söz konusu yöntemlerin doğrusal programlama konulu sorunlarda çözüm yöntemi olarak kullanımını 2nci Dünya Savaşı yıllarında görmekteyiz. 1939 yılında L. Kantorovich tarafından kaleme alınan “**Üretim planlama ve organizasyonunda matematiksel yöntemler**” isimli eser, özellikle ticari bağlamda doğrusal programlama sorunlarına çözüm önerilerini içermektedir<sup>(11)</sup>. 1947 yazında gündeme getirilen ve doğrusal programlama problemlerinin çözümünde oldukça popüler bir yöntem olan “simplex” yöntemi ise Dantzig’e aittir<sup>(12)</sup>.



**Şekil 2.15** Üç boyutlu ortamda kısıtların çevrelediği yüzeyler üzerinden optimal çözüme yönelik adımları<sup>(10)</sup>.



**Şekil 2.16** Üç boyutlu ortamda kısıtların çevrelediği yüzeyler dâhilinde en uygun çözüme yönelik adımları<sup>(13)</sup>.

İçsel nokta değişimi yöntemlerinde çokyüzlünün yüzeyinden ziyade, olası tasarım uzayının içinden geçen ve daha kısa görünen yol takip edilir. Modern manada içsel nokta değişimi



**Şekil 2.17** Karmarkar'a ait 1984 tarihli makalenin ilk sayfası<sup>(15)</sup>.

<sup>i</sup> “Mathematical methods of organizing and planning production”



yöntemleri popülerliğini L. Khachiyan'a borçludur. 1979 yılında yayınladığı<sup>(14)</sup> “Doğrusal programlamada polinomiyal algoritma” adlı makalesiyle polinomiyal zaman dilimi içerisinde, diğer bir ifade ile ilk gruba göre daha kısa sürede lineer optimizasyon problemlerinin çözülebileceğini gösterdi. Bazı çalışmalarda eliptik algoritma olarak adlandırılan yöntemin kuramsal olarak kabul gördüğü belirtilse de pratikte yaygınlaştığı söylenemez. İçsel nokta değişimi yöntemlerinin pratikte etkin olarak kullanılması N. Karmarkar tarafından gerçekleştirildi. 1984 yılında yayımlanan “Doğrusal programlama için yeni polinomiyal zamanlı algoritma” isimli makalesi ile Karmarkar<sup>(16)</sup> önerdiği yöntemin “simplex” yönteminden daha hızlı olduğunu belirtti. Günümüzde gerek taban değişimi ve gerekse de içsel nokta bazlı yöntemlerin daha kısa sürede sonuç verecek şekilde geliştirilmesine devam edilmektedir.

Doğrusal programlama yöntemleri bahsinde değinilecek bir diğer konu da, özellikle eşleşme sorunlarının<sup>iii</sup> çözümü dâhilinde dikkate alınan Macar algoritmasıdır<sup>iv</sup>. Eşleşme sorunu, bir doğrusal programlama problemi olarak aşağıdaki gibi tanımlanabilir;

$$\begin{aligned} \min_{\mathbf{c}} f(\mathbf{c}) &= \sum_{i=1}^n \sum_{j=1}^m c_{i,j} x_{i,j} & (2.3) \\ \text{Kısıt} \quad & \\ & \sum_{i=1}^n x_{i,j} = 1 \\ & \sum_{j=1}^m x_{i,j} = 1 \\ & 0 \leq x_{i,j} \leq 1 \end{aligned}$$

Burada  $\mathbf{c}$  maliyet verisini içeren yöneydir. Konuyu, anlaşılması açısından biraz daha açabiliriz. Bazı mühendislik problemlerinde  $n$  farklı kişi veya makine,  $m$  farklı işi, zamana veya paraya dayalı ama farklı bir maliyetle yapabilir. Böyle durumlarda önemli olan husus, kişi/makineyi yapılacak işle ve en az maliyetle eşleştirmektir. Daha net bir ifade ile hangi işin kimin tarafından yapılacağı belirlenmelidir. Çözüm bağlamında dikkate alınan kısıt ise her bir makine veya kişiye sadece bir işin verilebileceğidir. Nitekim kısıtlardan ilki, her bir kişinin sadece bir kez görevlendirilebileceğini, ikincisi her bir işin sadece bir kez verilebileceğini göstermektedir.

<sup>i</sup> “A polynomial algorithm in linear programming”

<sup>ii</sup> “A new polynomial-time algorithm for linear programming”

<sup>iii</sup> Assignment problem

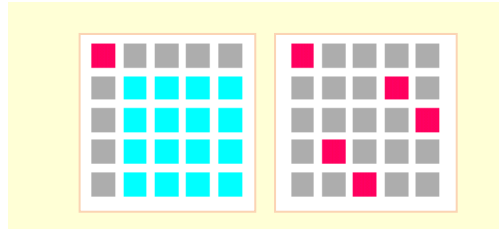
<sup>iv</sup> Hungarian algorithm



Özellikle  $n = m$  koşulunun geçerli olduğu problemlerde optimal çözüme ulaşmak mümkündür. Çizge kuramı üzerine kurulu eşleşme sorunu, bazı kayıtlarda kümelere dayalı olarak da tanımlanır. Buna göre  $A$  ve  $T$  aynı büyüklükte iki küme olmak üzere,  $C: A \times T$  şeklinde bir maliyet fonksiyonu tanımlanabilir. Buna göre ana gaye;

$$\min_C f(C) = \sum_{i=1}^n C(a, f(a)) \quad (2.4)$$

denklemi ile de ifade edilir. Denklemde geçen  $a$  ilk kümeye ait eleman,  $f(a)$  ise ikinci kümeye ait elemandır. Sorunun tanıma uygun olarak belirlenmesi için iki küme arasındaki ilişkiyi sağlayan  $f(a)$  fonksiyonunun birebir örten<sup>ii</sup> fonksiyon olması gerekir. Daha açık bir ifade ile söz konusu koşul, her bir makine veya kişiye sadece bir işin verilebileceği kısıtını temsil etmektedir.



**Şekil 2.18** Macar algoritmasında ana gaye, mevcut imkânlar ile yapılması gereken işlerin en uygun şekilde eşleştirilmesidir<sup>(17)</sup>.

Bahse konu eşleşme sorunlarında kullanılacak yöntemlerden biri olan Macar algoritması 1955 yılında H. W. Kuhn tarafından kaleme alınan “Eşleşme problemleri için Macar algoritması<sup>iii</sup>” başlıklı makale ile gündeme geldi. Yaklaşımın kuramsal altyapısı, 1910’lu ve 1930’lu yıllarda D. König ve E. Egervary adlı Macar matematikçilerin çalışmalarına dayanmakta olup<sup>(18)</sup>, esas itibari ile taban değişimi bazlı bir yöntemdir. Ayrıca yöntemin polinomiyal zaman karmaşıklığına sahip bir yöntem olduğu da söylenebilir<sup>(19)</sup>.



**Şekil 2.19** Kuhn’a ait 1955 tarihli makalenin ilk sayfası.

<sup>i</sup> Graph theory

<sup>ii</sup> Bisection function

<sup>iii</sup> “The Hungarian method for the assignment problem.”



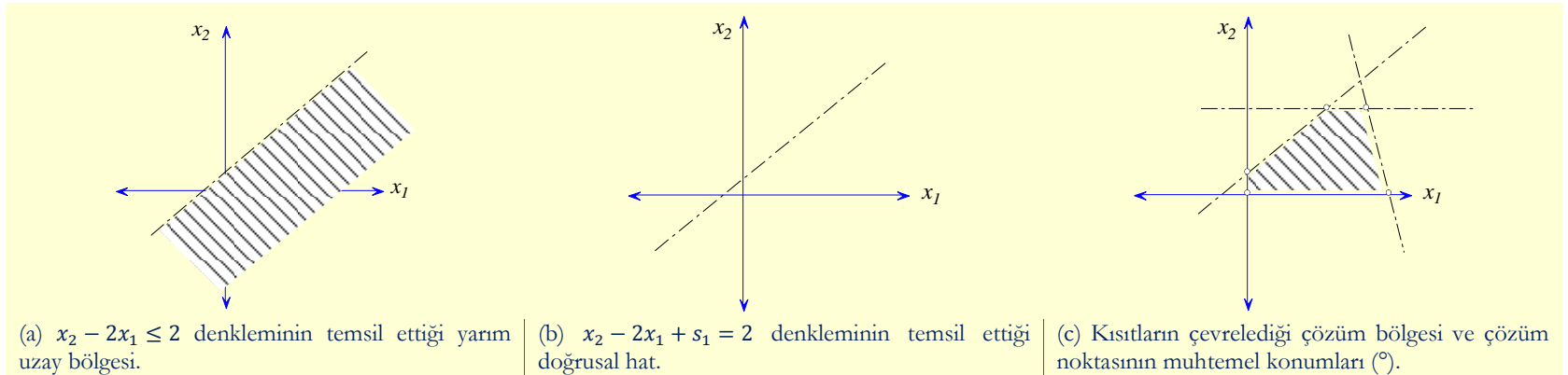


### 2.3.1 “Simplex” yöntemi

Doğrusal eniyileme problemlerinde oldukça yaygın olarak kullanılan “simplex” yönteminin temelinde, kısıtların oluşturduğu denklem takımının dikkate alınması ve çözümün bu çerçeve üzerinde aranması yatar. Çözüm arayışına girmek için öncelikle çözüm bölgesinin sınırları dikkate alınır. Daha sonra bu sınırlar üzerindeki köşe noktaları, belli bir sistematik çerçevede deneyerek maliyet açısından en düşük değeri veren noktaya ulaşılır. Çözüm bölgesinin sınırlarının dikkate alınması için tanımlanan sorunun standart forma dönüştürülmesi gerekir. Bir doğrusal probleminin standart formda olabilmesi için tanımlı tüm kısıtların eşitlik kısıtı, her bir eşitliğin sağ tarafındaki sabit değerlerin ve tasarım değişkenlerinin de pozitif olması gerekir. Buna göre standart formdaki doğrusal bir programlama problemini;

$$\begin{aligned} \min_x f(x) &= c^T x && (2.5) \\ \text{Kısıt} \quad Ax &= b \\ x &\geq 0 \\ b &\geq 0 \end{aligned}$$

şeklinde yazmak mümkündür.



Şekil 2.20 Standart forma dönüşümün çözüm bölgesine etkisi.

Doğal olarak gerçek probleminin her zaman standart formda olması beklenemez. Dolayısıyla eldeki tanımlı sorunun, çözüm arayışı öncesinde standart forma dönüştürülmesi gerekir.





Eşitsizlik kısıtlarının eşitlik kısıtlarına dönüştürülmesi çözüm bölgesinden çözüm hattına geçişi sağlar. Zira optimal çözüm noktası, çözüm bölgesini çevreleyen hat üzerinde ve köşe noktalarındadır. Örneğin  $x_2 - 2x_1 \leq 2$  şeklinde verilen basit bir iki boyutlu eşitsizlik kısıtı geometrik anlamda bir yarımuzayı temsil eder. Söz konusu yarımuzay Şekil 2.20(a)'da görüldüğü üzere bir doğru değil, bir alandır. Eşitsizlik kısıtını,  $s_1$  gibi ilave yapay bir değişken<sup>i</sup> tanımlayıp denkleme ekleyerek eşitlik kısıtı haline getirmek mümkündür:  $x_2 - 2x_1 + s_1 = 2$ . Bu durumda denklemin geometrik anlamı Şekil 2.20(b)'de görüldüğü gibi yarımuzaydan ziyade bir doğrusal hat haline gelir.

Dönüşümün diğer bir gayesi, başlangıçtaki çözümüm orijin olarak belirlenmesidir. Dolayısıyla çözüm arayışındaki ilk adımda  $\mathbf{x} = 0$  ve amaç fonksiyon değeri de  $f = 0$  olur. Problemin standart forma dönüştürülmesi sonrasında çözüm noktası arayışı süreci başlatılır. Bu meyanda yapılan işlemler pivotlama işlemleridir. Pivotlama işlemi ile çözüm hattındaki köşe noktaları sırası ile denenecek optimal çözümü veren noktanın bulunması amaçlanır. Şekil 2.20(c)'de görüldüğü üzere köşe noktasının özelliği, iki boyutlu ortamda iki kısıtın kesişim noktası olmasıdır. Dolayısıyla her iki kısıt denklemini sağlayan çözüm noktası, olası bir optimal nokta olup bu noktanın amaç fonksiyonundaki değerine bakılarak bir sonraki adıma geçilir. Eğer bulunulan konum optimal nokta değilse bir sonraki kesişim noktasına geçilir. Kesişim noktası, mevcut kısıt denklemlerinden biri ile diğer başka bir kısıt denkleminin ortak çözüm noktasıdır.

Doğrusal programlama sorununun standart forma dönüştürülmesi ile elde edilen sorun, çözümü itibarı ile orijinal sorunla aynı çözüme sahiptir. Diğer taraftan pivotlama işlemleri sorunun değişimine neden olmaz, zira eşitliğin sağına ve soluna aynı işlemlerin uygulanması ile eşitlik bozulmaz.

#### a. Standart forma dönüşüm

Tanımlanan herhangi bir doğrusal eniyileme sorununun çözümünün “simplex” yöntemi ile elde edilmesi için yapılması gereken dönüşüm adımlarını; kısıtların, tasarım değişkenlerinin ve amaç fonksiyonunun dönüşümü şeklindeki üç grupta derleyebiliriz<sup>(20)</sup>.

Yöntemin temelinde minimizasyon amacı yatmaktadır. Dolayısıyla doruklaştırma olarak tanımlı bir sorunun minimizasyon sorununa dönüşümü gerekir. Hatırlanacağı üzere daha önce Alt Bölüm 1.3.2 Amaç fonksiyonu başlığı altında dönüşümün nasıl yapılacağı konusu işlenmişti.

<sup>i</sup> Slack variable



Buna göre verili amaç fonksiyonunun  $(-1)$  ile çarpılması neticesinde doruklaştırma sorunu, çözümde herhangi bir değişikliğe yol açılmadan minimizasyon problemine dönüştürülmüş olur. Örneğin  $\max_x f(x) = 4x_1 + 2x_2$  amaç fonksiyonunu,  $\min_x -f(x) = -4x_1 - 2x_2$  formuna dönüştürebiliriz.

Kısıtların dönüşümünde dört durumdan bahsedilebilir. İlk durum kısıtın " $\leq$ " olarak verilmesidir. Bu durumda kısıt denkleminin sol tarafına yapay bir  $s_i$  değişkeni eklenerek denklem eşitlik denklemini haline gelir. Tabiatıyla eklenen değişkenin  $s_i \geq 0$  şartını sağlaması gerekir. Örneğin;  $x_1 + 2x_2 \leq 4$  kısıtını;

$$\begin{aligned} x_1 + 2x_2 + s_1 &= 4 \\ s_1 &\geq 0 \end{aligned}$$

şeklindeki eşitlik kısıtı haline getirebiliriz.

Kısıt denkleminin " $\geq$ " şartını taşıması halinde denkleme iki ilave yapay değişken eklenir. Bunlardan ilki eksiltme amaçlı  $e_i$  yapay değişkenidir. Örneğin;  $2x_1 + 3x_2 \geq 6$  kısıtını;

$$\begin{aligned} 2x_1 + 3x_2 - e_1 &= 6 \\ e_1 &\geq 0 \end{aligned}$$

formunda yazabiliriz. Ancak başlangıçta dikkate alınan  $x = 0$  çözümü yerine koyulduğunda,  $e_1 = -6$  olmaktadır ki bu da  $e_1 \geq 0$  şartını sağlamamaktadır. Bu nedenle kısıt denklemine, daha önce tanımlanan  $s_i$  değişkeni ilave edilir. Dolayısıyla kısıt;

$$\begin{aligned} 2x_1 + 3x_2 - e_1 + s_1 &= 6 \\ e_1 &\geq 0 \\ s_1 &\geq 0 \end{aligned}$$

şekline dönüşür. Dikkat edilecek olursa  $x = 0$  çözümü yerine konulduğunda, eşitliği ve aynı zamanda  $e_1 \geq 0$  ve  $s_1 \geq 0$  şartlarını sağlayabilecek değişkenlerin var olabileceği görülür. Kısıtın " $=$ " olması durumunda kısıt açısından bir sorun yok gibi görünebilir. Ancak yöntem gereği dikkate alınan  $x = 0$  çözümü yerine koyulduğunda, eşitliğin sağlanamaması sorunu ile karşılaşılabilir. Bu nedenle tanımlı kısıta  $s_i$  değişkeni eklenerek dönüşüm sağlanmış olur. Örneğin;

---

<sup>i</sup> Surplus variable



$x_1 + 3x_2 = 4$  kısıtında başlangıç çözümü ( $x = 0$ ) dikkate alınrsa,  $0 = 4$  gibi bir tutarsızlık görülür. Bu nedenle aynı kısıtı;

$$\begin{aligned} x_1 + 3x_2 + s_1 &= 4 \\ s_1 &\geq 0 \end{aligned}$$

formunda yazabiliriz.

Kısıt dönüşümüne ait son durum, denklemin sağ tarafındaki sabit değerin negatif olması halidir. Bu durumda eşitliğin her iki tarafı  $(-1)$  ile çarpılarak ilk dönüşüm gerçekleşmiş olur. Sonraki adımda ise eldeki duruma göre daha önce belirtilen ilk iki halden biri dikkate alınır. Örneğin;  $4x_1 + 2x_2 \leq -1$  kısıtı;

$$-4x_1 - 2x_2 \geq 1$$

şeklinde yazılabilir. Dikkat edileceği üzere çarpım sonrası ilişki değişikliği olmaktadır. Değişen kısıt, ilave değişkenler dikkate alınarak aşağıdaki gibi bir eşitlik kısıtına dönüştürülmüş olur;

$$\begin{aligned} -4x_1 - 2x_2 - e_1 + s_1 &= 1 \\ e_1 &\geq 0 \\ s_1 &\geq 0 \end{aligned}$$

Dönüşümlerde dikkat edilmesi gereken diğer bir adım, tasarım değişkenleri ile ilgilidir. Bazı hallerde tasarım değişkeni  $x_i \leq 0$  şeklinde tanımlı olabilir. Bu durumda denkleme, kısıtlarda olduğu gibi  $s_i$  değişkeninin eklenmesi ile  $x_i + s_i \leq 0$  yazılıp istenen sonuca varılamaz. Zira  $s_i \geq 0$  şartı nedeniyle hala  $x_i \leq 0$  durumu devam etmektedir. Bu nedenle sorun  $y_i = -x_i$  ve dolayısıyla  $y_i \geq 0$  tanımı ile çözülür. Tabiatıyla  $x_i = -y_i$  dönüşümünün, amaç fonksiyonu dâhil tüm denklemlerde dikkate alınması gerekir. Örneğin;

$$\begin{aligned} \min_x f(x) &= 4x_1 - 3x_2 \\ \text{Kısıt} \quad & \\ & 3x_1 + 2x_2 \leq 5 \\ & 2x_1 + x_2 \leq 4 \\ & x_1 \geq 0 \\ & x_2 \leq 0 \end{aligned}$$

şeklinde tanımlı bir sorun için  $x_2 \leq 0$  ifadesinin standart forma dönüşümünde  $y_2 = -x_2$  tanımı dikkate alınarak aşağıdaki değişiklikler yapılabilir;





$$\min_x f(x) = 4x_1 + 3y_2$$

Kısıt

$$3x_1 - 2y_2 + s_1 = 5$$

$$2x_1 - y_2 + s_2 = 4$$

$$x_1, y_2, s_1, s_2 \geq 0$$

Tasarım değişkenleri ile ilgili son durum, değişkenin işareti konusunda herhangi bir kısıtın olmaması halidir. Oysa hatırlanacağı üzere standart formda tüm tasarım değişkenlerinin  $x \geq 0$  şartını sağlaması gerekmektedir. Örneğin aşağıdaki gibi tanımlı bir sorunda;

$$\min_x f(x) = x_1 - x_2$$

Kısıt

$$x_1 + 2x_2 \leq 3$$

$$2x_1 + 3x_2 \leq 2$$

$$x_1 \geq 0$$

$x_2$  değişkeninin yönüne ait herhangi bir tanım verilmemektedir. Böyle bir durumda dönüşüm,  $x_2 = y_2 - w_2$  gibi ilave iki değişkenin dikkate alınması ile çözülür. Dolayısıyla standart forma dönüşen sorun;

$$\min_x f(x) = x_1 - y_2 + w_2$$

Kısıt

$$x_1 + 2(y_2 - w_2) + s_1 = 3$$

$$2x_1 + 3(y_2 - w_2) + s_2 = 2$$

$$x_1, s_1, s_2, y_2, w_2 \geq 0$$

şeklinde yazılabilir. Dikkat edileceği üzere  $y_2 \geq 0$  ve  $w_2 \geq 0$  olsa da, orijinal sorun tanımındaki  $x_2 \geq 0$  veya  $x_2 \leq 0$  olasılıkları hala sağlanabilmektedir.

## b. “Simplex” tablosu ve pivot işlemleri

Doğrusal eniyileme sorununun standart forma dönüştürülmesi sonrasında, lineer programlama tanımına uyacak şekilde çizelgeleme ve çizelge elemanları üzerinde işlem icra etme safhası başlar. Konu işlemler arasında değişkenlerin yer değiştirmesi ve katsayılarının sıfırlanması gibi aritmetik operasyonlardan bahsedilebilir. Çizelgenin her yenilenme işleminde değişkenlerden





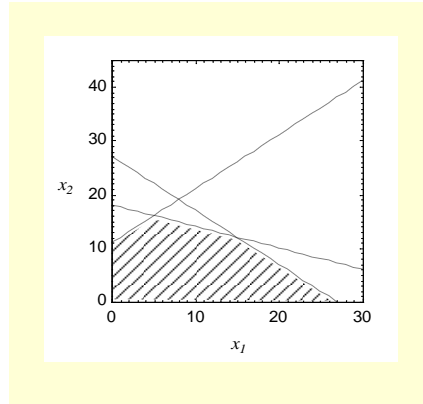
birisinin çizelgeden çıkarılması<sup>i</sup>, buna mukabil diğer bir değişkenin ise çizelgeye dâhil edilmesi<sup>ii</sup> söz konusudur. Çizelge oluşumunda dikkat edilen ilk husus, en üst satıra tüm değişkenlerin sıralanması ve sütun olarak en sağa veya sola bağımlı değişkenlerin<sup>iii</sup> yerleştirilmesidir. Çizelgenin oluşturması ve sonrasında en uygun çözüm noktasının bulunması için yapılan işlemlerin somut bir sorun sürecinde örneklendirilmesi daha uygun olacaktır<sup>(21)</sup>. Buna göre müteakip doğrusal eniyileme sorununu ele alalım;

$$\max_x f(x) = 4x_1 + 6x_2$$

Kısıt

$$\begin{aligned} -x_1 + x_2 &\leq 11 \\ x_1 + x_2 &\leq 27 \\ 2x_1 + 5x_2 &\leq 90 \\ x_1, x_2 &\geq 0 \end{aligned}$$

Veriler ışığında sorunun bir doruklaştırma sorunu olduğu, ilave olarak üç adet kısıtın tanımlandığı anlaşılmakta olup kısıtların çevrelediği çözüm bölgesini müteakip şekilde görmek mümkündür;



Şekil 2.21 Kısıtların çevrelediği çözüm bölgesi.

Yapılması gereken ilk işlem, problemin standart forma dönüştürülmesidir. Sorun, amaç fonksiyonunun minimizasyonu ve eşitsizlik kısıtlarının eşitlik kısıtlarına çevrilmesi sayesinde;

<sup>i</sup> Departing variable

<sup>ii</sup> Entering variable

<sup>iii</sup> Bağımlı değişken İngilizce’de “basic variable”, bağımsız değişken ise “non-basic variable” olarak geçer.





$$\min_x z = -f(x) = -4x_1 - 6x_2$$

Kısıt

$$-x_1 + x_2 + s_1 = 11$$

$$x_1 + x_2 + s_2 = 27$$

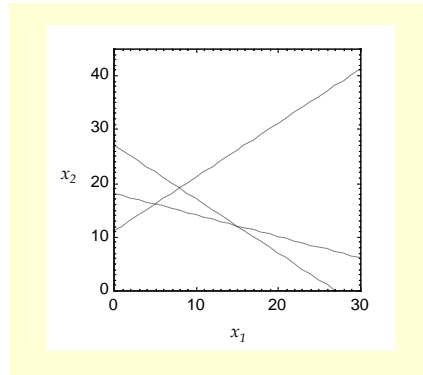
$$2x_1 + 5x_2 + s_3 = 90$$

$$x_1, x_2, s_1, s_2, s_3 \geq 0$$

şeklindeki standart formda ifade edilebilir. Buna mukabil başlangıçtaki “simplex” çizelgesi;

$x_1$	$x_2$	$s_1$	$s_2$	$s_3$	$b$	Bağımlı değişkenler
-1	1	1	0	0	11	$s_1$
1	1	0	1	0	27	$s_2$
2	5	0	0	1	90	$s_3$
-4	-6	0	0	0	0	$z$

şeklinde oluşturulabilir.



Şekil 2.22 Başlangıçtaki çözüm noktası: (0,0).

Dikkat edileceği üzere çizelgenin oluşturulmasında kısıt denklemlerindeki değişkenlerin katsayıları, eşitliklerin sağ tarafındaki sabit değerler ve son satırda kısıt denklemleri haricindeki amaç fonksiyonu denklemi dikkate alınmaktadır. Bağımlı değişkenler, tasarım değişkenleri haricindeki yapay değişkenler olup her birine ait sütun dâhilinde kendisi haricindeki diğer değişkenlerin katsayıları sıfırdır. Standart formun doğası gereği, başlangıçtaki çözüm  $(x_1, x_2, s_1, s_2, s_3) = (0, 0, 11, 27, 90)$  ve buna karşılık gelen amaç fonksiyonu değeri  $z = 0$  olarak ifade edilebilir. Şekil 2.22’de çözüm başlangıç noktasının konumu görülmektedir.



Çizelgenin oluşturulmasını müteakiben yenilenme işlemlerine başlanır. Bu işlemlerin yapılması esnasında dikkate edilecek husus, en alt satırda negatif katsayının olmamasıdır. Daha açık bir ifade ile amaç fonksiyonuna ait satırda negatif sayı kalmayınca kadar çizelgenin yenilenmesi işlemlerine devam edilir. Söz konusu satırdaki tüm sayıların pozitif olması halinde, gelinen noktanın optimal nokta olduğu söylenebilir. Başlangıç çizelgesinde öncelikle en alt satırdaki en küçük sayısal değerin yeri tespit edilir. Bu tespit ile çizelgenin en sağındaki bağımlı değişkenler sütununa girecek değişken de belirlenmiş olur. Buna göre son satırdaki minimum değer  $-6$  olup bağımlı değişken sütununa dâhil edilecek değişken  $x_2$  değişkenidir. Bağımlı değişkenler sütunundan çıkarılacak değişken ise sabit değerlerin bulunduğu  $b$  sütunu ile giren değişkenin bulunduğu sütunun birbirine bölünmesi ile tespit edilir. Buna göre;

$$11/1 = 11$$

$$27/1 = 27$$

$$90/5 = 18$$

işlemleri yapılarak işlerinde pozitif işarete sahip en küçük değer seçilir. Dolayısıyla  $11/1 = 11$  değeri en küçük pozitif sayı olup onun bulunduğu satırdaki  $s_1$  değişkeni bağımlı değişken sütunundan ayrılacak değişkendir.

$x_1$	$x_2$	$s_1$	$s_2$	$s_3$	$b$	Bağımlı değişkenler
-1	1	1	0	0	11	$s_1$
1	1	0	1	0	27	$s_2$
2	5	0	0	1	90	$s_3$
-4	-6	0	0	0	0	$z$

← Çıkarılan değişken

↑ Dâhil edilen değişken

Değişkenlerin tespiti sonrasında pivotlama işlemleri yapılır. Bu işlemde dahil edilen değişken sütunu ile çıkarılan değişken satırının kesiştiği sayı (burada 1 sayısı) pivot olarak kullanılır ve o sütundaki diğer tüm sayıların sıfır olması için satır işlemleri uygulanır. Bu arada kendi değeri de 1 sayısından farklı ise öncelikle o satıra uygun bölme işlemi uygulanır. Örneğin ikinci satırdan birinci satır çıkarılarak ikinci satır tekrar yazılır ve bu sayede ikinci satır ikinci sütundaki katsayı sıfırlanmış olur. Benzer işlemler aynı sütundaki diğer katsayılar için uygulanırsa;

$$\begin{array}{l}
 [1 \ 1 \ 0 \ 1 \ 0 \ 27]^{2\text{nci satır}} \quad - \quad [-1 \ 1 \ 1 \ 0 \ 0 \ 11]^{pivot satır} = [2 \ 0 \ -1 \ 1 \ 0 \ 16]^{2\text{nci satır}} \\
 [2 \ 5 \ 0 \ 0 \ 1 \ 90]^{3\text{üncü satır}} \quad - \quad 5 \times [-1 \ 1 \ 1 \ 0 \ 0 \ 11]^{pivot satır} = [7 \ 0 \ -5 \ 0 \ 1 \ 35]^{3\text{üncü satır}} \\
 [-4 \ -6 \ 0 \ 0 \ 0 \ 0]^{4\text{üncü satır}} \quad + \quad 6 \times [-1 \ 1 \ 1 \ 0 \ 0 \ 11]^{pivot satır} = [-10 \ 0 \ 6 \ 0 \ 0 \ 66]^{4\text{üncü satır}}
 \end{array}$$

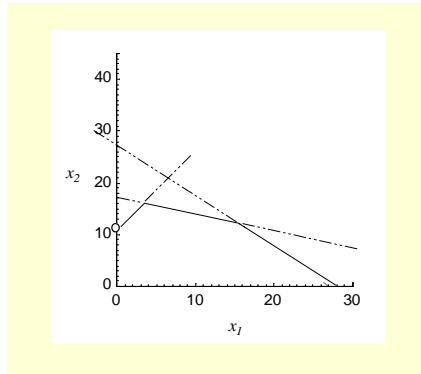




sayıları ve müteakip “simplex” çizelgesi elde edilmiş olur:

$x_1$	$x_2$	$s_1$	$s_2$	$s_3$	$b$	Bağımlı değişkenler
-1	1	1	0	0	11	$x_2$
2	0	-1	1	0	16	$s_2$
7	0	-5	0	1	35	$s_3$
-10	0	6	0	0	66	$z$

Yenilenmiş çizelgeye göre yeni çözüm noktası  $(x_1, x_2, s_1, s_2, s_3) = (0, 11, 0, 16, 35)$  ve buna mukabil gelen amaç fonksiyonu değeri de  $z = 66$  olarak ifade edilebilir. Dikkat edileceği üzere  $x_2$  değeri 11 olarak dikkate alınmıştır. Şekil 2.23 dâhilinde yeni çözüm noktasının konumu görülmektedir.



Şekil 2.23 Yeni çözüm noktası: (0,11).

Oluşan yeni çizelgenin en alt satırı kontrol edildiğinde hala negatif sayının var olduğu görülmektedir. Bu nedenle çizelge yenileme işlemlerine aynı işlem adımları çerçevesinde devam edilir. Buna göre son satırdaki minimum değer  $-10$  olup bağımlı değişken sütununa dâhil edilecek değişken  $x_1$  değişkenidir. Bağımlı değişkenler sütunundan çıkarılacak değişken ise sabit değerlerin bulunduğu  $b$  sütunu ile giren değişkenin bulunduğu sütunun birbirine bölünmesi sonucu elde edilen ve içlerinde pozitif işarete sahip en küçük değer seçilir;

$$\begin{aligned} 11 / -1 &= -11 \\ 16 / 2 &= 8 \\ 35 / 7 &= 5 \end{aligned}$$







Dolayısıyla,  $35/7 = 5$  değeri en küçük pozitif sayı olup onun bulunduğu satırdaki  $s_3$  değişkeni bağımlı değişken sütunundan ayrılacak değişkendir.

$x_1$	$x_2$	$s_1$	$s_2$	$s_3$	$b$	Bağımlı değişkenler
-1	1	1	0	0	11	$x_2$
2	0	-1	1	0	16	$s_2$
7	0	-5	0	1	35	$s_3$
-10	0	6	0	0	66	$z$

↑ Dâhil edilen değişken

← Çıkarılan değişken

Değişkenlerin tespiti sonrasında pivotlama işlemleri yapılır. Bu işlemde dahil edilen değişken sütunu ile çıkarılan değişken satırının kesiştiği sayı (burada 7 sayısı) pivot olarak kullanılır ve o sütundaki diğer tüm sayıların sıfır olması için satır işlemleri uygulanır. Ancak pivot sayısının 7 olması nedeniyle öncelikle o satır  $1/7$  ile çarpılır. Daha önce bahsedilen benzer işlemler uygulanırsa;

$$\begin{array}{l}
 [-1 \ 1 \ 1 \ 0 \ 0 \ 11]^{1\text{inci satır}} + \quad 1/7 \times [7 \ 0 \ -5 \ 0 \ 1 \ 35]^{pivot \ satır} = [1 \ 0 \ -5/7 \ 0 \ 1/7 \ 5]^{pivot \ satır} \\
 [2 \ 0 \ -1 \ 1 \ 0 \ 16]^{2\text{nci satır}} - \quad [1 \ 0 \ -5/7 \ 0 \ 1/7 \ 5]^{pivot \ satır} = [0 \ 1 \ 2/7 \ 0 \ 1/7 \ 16]^{1\text{inci satır}} \\
 [2 \ 0 \ -1 \ 1 \ 0 \ 16]^{2\text{nci satır}} - \quad 2 \times [1 \ 0 \ -5/7 \ 0 \ 1/7 \ 5]^{pivot \ satır} = [0 \ 0 \ 3/7 \ 1 \ -2/7 \ 6]^{2\text{nci satır}} \\
 [-10 \ 0 \ 6 \ 0 \ 0 \ 66]^{4\text{üçü satır}} + \quad 10 \times [1 \ 0 \ -5/7 \ 0 \ 1/7 \ 5]^{pivot \ satır} = [0 \ 0 \ -8/7 \ 0 \ 10/7 \ 116]^{4\text{üncü satır}}
 \end{array}$$

sayıları ve müteakip yeni “simplex” çizelgesi elde edilmiş olur:

$x_1$	$x_2$	$s_1$	$s_2$	$s_3$	$b$	Bağımlı değişkenler
0	1	2/7	0	1/7	16	$x_2$
0	0	3/7	1	-2/7	6	$s_2$
1	0	-5/7	0	1/7	5	$x_1$
0	0	-8/7	0	10/7	116	$z$

Yenilenmiş çizelgeye göre yeni çözüm noktası  $(x_1, x_2, s_1, s_2, s_3) = (5, 16, 0, 6, 0)$  ve buna mukabil gelen amaç fonksiyonu değeri  $z = 116$  olarak ifade edilebilir. Şekil 2.24 dâhilinde yeni çözüm noktasının konumu görülmektedir. Oluşan yeni çizelgenin en alt satırı kontrol edildiğinde bir adet negatif sayının var olduğu görülmektedir. Bu nedenle çizelge yenileme işlemlerine aynı işlem adımları çerçevesinde devam edilir. Buna göre son satırdaki minimum değer  $-8/7$  olup bağımlı değişken sütununa dâhil edilecek değişken  $s_1$  değişkenidir. Bağımlı değişkenler sütunundan



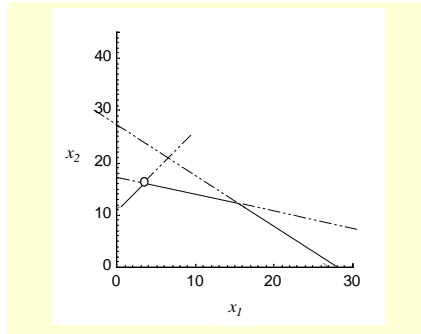
çıkarılacak değişken ise sabit değerlerin bulunduğu  $b$  sütunu ile giren değişkenin bulunduğu sütunun birbirine bölünmesi sonucu;

$$16/(2/7) = 56$$

$$6/(3/7) = 14$$

$$5/(-5/7) = -7$$

işlemleri yazılarak içlerinden pozitif işarete sahip en küçük değer seçilir. Buna göre  $6/(3/7) = 14$  değeri en küçük pozitif sayı olup onun bulunduğu satırdaki  $s_2$  değişkeni bağımlı değişken sütunundan ayrılacak değişkendir.



Şekil 2.24 Yeni çözüm noktası (5,16).

$x_1$	$x_2$	$s_1$	$s_2$	$s_3$	$b$	Bağımlı değişkenler
0	1	2/7	0	1/7	16	$x_2$
0	0	3/7	1	-2/7	6	$s_2$
1	0	-5/7	0	1/7	5	$x_1$
0	0	-8/7	0	10/7	116	$z$

↑ Dâhil edilen değişken

← Çıkarılan değişken

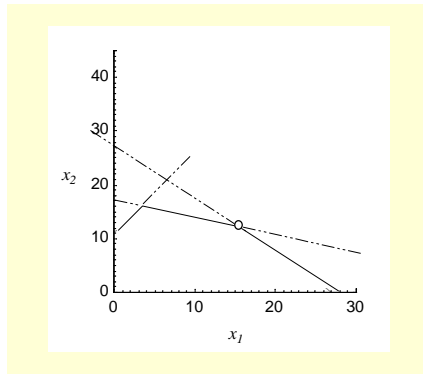
Değişkenlerin tespiti sonrasında pivotlama işlemleri yapılır. Bu işlemde dahil edilen değişken sütunu ile çıkarılan değişken satırının kesiştiği sayı (burada  $3/7$  sayısı) pivot olarak kullanılır ve o sütundaki diğer tüm sayıların sıfır olması için satır işlemleri uygulanır. Ancak pivot sayısının  $3/7$  olması nedeniyle öncelikle o satır  $7/3$  ile çarpılır. Daha önce bahsedilen benzer işlemler uygulanırsa;



$$\begin{array}{l}
 [0 \ 1 \ 2/7 \ 0 \ 1/7 \ 16] \text{1inci satır} \\
 [1 \ 0 \ -5/7 \ 0 \ 1/7 \ 5] \text{3üncü satır} \\
 [0 \ 0 \ -8/7 \ 0 \ 10/7 \ 116] \text{4üncü satır}
 \end{array}
 \begin{array}{l}
 - \\
 + \\
 +
 \end{array}
 \begin{array}{l}
 7/3 \times [0 \ 0 \ 3/7 \ 1 \ -2/7 \ 6] \text{pivot satır} \\
 2/7 \times [0 \ 0 \ 1 \ 7/3 \ -2/3 \ 14] \text{pivot satır} \\
 5/7 \times [0 \ 0 \ 1 \ 7/3 \ -2/3 \ 14] \text{pivot satır} \\
 8/7 \times [0 \ 0 \ 1 \ 7/3 \ -2/3 \ 14] \text{pivot satır}
 \end{array}
 =
 \begin{array}{l}
 [0 \ 0 \ 1 \ 7/3 \ -2/3 \ 14] \text{pivot satır} \\
 [0 \ 1 \ 0 \ -2/3 \ 1/3 \ 12] \text{1inci satır} \\
 [1 \ 0 \ 0 \ 5/3 \ -1/3 \ 15] \text{3üncü satır} \\
 [0 \ 0 \ 0 \ 8/3 \ 2/3 \ 132] \text{4üncü satır}
 \end{array}$$

sayıları ve müteakip “simplex” çizelgesi elde edilmiş olur:

$x_1$	$x_2$	$s_1$	$s_2$	$s_3$	$b$	Bağımlı değişkenler
0	1	0	-2/3	1/3	12	$x_2$
0	0	1	7/3	-2/3	14	$s_1$
1	0	0	5/3	-1/3	15	$x_1$
0	0	0	8/3	2/3	132	$z$

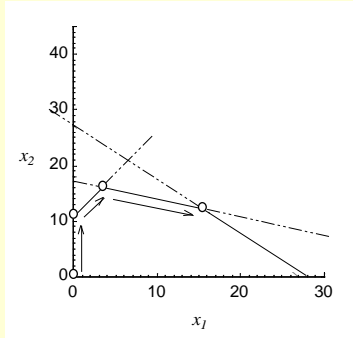


Şekil 2.25 Yeni çözüm noktası (15,12).

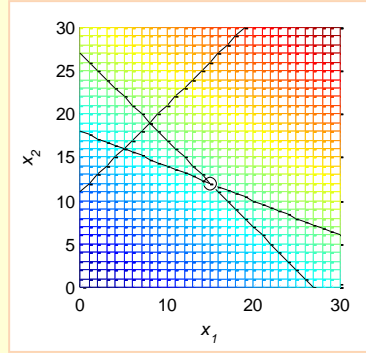
Yenilenmiş çizelgeye göre yeni çözüm noktası  $(x_1, x_2, s_1, s_2, s_3) = (15, 12, 14, 0, 0)$  ve buna mukabil gelen amaç fonksiyonu değeri  $z = 132$  olarak ifade edilebilir. Şekil 2.25'te yeni çözüm noktasının konumu görülmektedir. Oluşan yeni çizelgenin en alt satırı kontrol edildiğinde herhangi bir negatif değer var olmadığı anlaşılmaktadır. Bu nedenle bulunulan nokta en uygun nokta olup eniyileme süreci bu noktada sonlandırılabilir.

Görüldüğü üzere “simplex” yöntemi ile dört adımda optimal noktaya ulaşılabilmiştir. Şekil 2.26'dan da gözlemlenebileceği gibi, varılan her bir nokta çözüm bölgesini çevreleyen hat üzerindeki köşe, ya da diğer bir ifade ile kısıtların kesiştiği nokta olup çizelge yenileme işlemleri ile bir kesişim noktasından başka bir kesişim noktasına geçilmekte ve bu sayede optimal nokta taraması yapılmaktadır.





(a)



(b)

Şekil 2.26 (a) Her adımda ziyaret edilen kesişim noktaları, (b) amaç fonksiyonu üzerindeki optimal çözüm noktası (15,12).

### 2.3.2 “Big-M Simplex” yöntemi

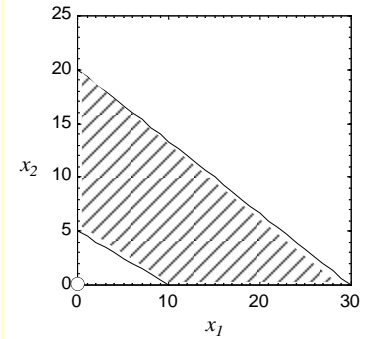
“Simplex” yönteminin temelinde çözüm bölgesinin çevrenmesi ve başlangıç noktasının da çözüm bölgesi içinde kalacak şekilde  $x = 0$  olması yatmaktaydı. Ancak bazı problemlerde başlangıç noktası olarak seçilen  $x = 0$  noktası olası çözüm uzayının dışında olabilir. Bu durumda yöntemin uygulanmasında sorunlar çıkabilir. Örneğin müteakip sorunu<sup>(22)</sup> dikkate alalım;

$$\min_x f(x) = 4x_1 + 5x_2$$

Kısıt

$$\begin{aligned} x_1 + 2x_2 &\geq 10 \\ 2x_1 + 3x_2 &\leq 60 \\ x_1, x_2 &\geq 0 \end{aligned}$$

Veriler ışığında çözüm bölgesi ve kısıtlar Şekil 2.27’deki gibi resmedilebilir. Buna göre başlangıç noktası olarak seçilen  $x_1, x_2 = 0$  noktası, tasarım değişkenleri için dikkate alınan  $x_1, x_2 \geq 0$  kısıtını karşılamakta, ancak olası çözüm bölgesinin dışında kalmaktadır. Hatırlanacağı üzere tanımlanan sorunun standart forma dönüşümü kapsamında aynı sorun;



Şekil 2.27 Olası çözüm bölgesi.



$$\min_x f(x) = 4x_1 + 5x_2$$

Kısıt

$$\begin{aligned} x_1 + 2x_2 - s_2 &= 10 \\ 2x_1 + 3x_2 + s_1 &= 60 \\ x_1, x_2, s_1, s_2 &\geq 0 \end{aligned}$$

şeklinde yazılabilmekteydi. Ancak bu formda  $x_1, x_2 = 0$  noktası başlangıç olarak dikkate alındığında  $s_2 \geq 0$  şartı sağlanamamaktadır, ( $s_2 = -10$ ). Dolayısıyla eklenen yapay değişkenlere ilave olarak  $e_1$  gibi ilave bir değişken, kısıt denkleminde dâhil edilmekte ve bu durumda problem;

$$\min_x f(x) = 4x_1 + 5x_2$$

Kısıt

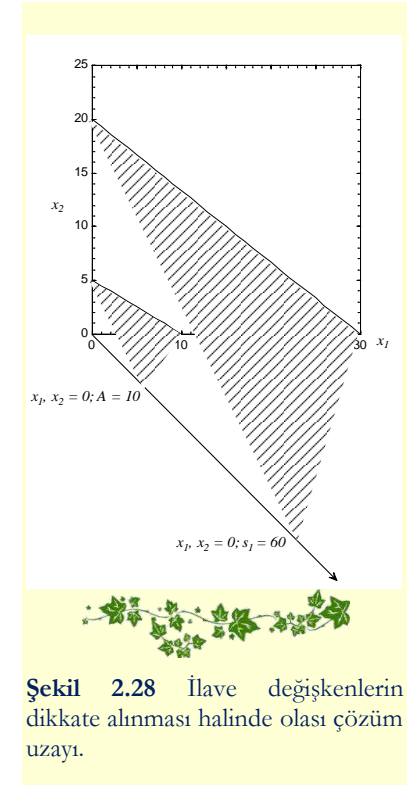
$$\begin{aligned} x_1 + 2x_2 - s_2 + e_1 &= 10 \\ 2x_1 + 3x_2 + s_1 &= 60 \\ x_1, x_2, s_1, s_2, e_1 &\geq 0 \end{aligned}$$

$$2x_1 + 3x_2 + s_1 = 60$$

haline dönüştürülmektedir. Buradaki ilk kısıt denkleminde bakılacak olursa  $e_1 - s_2$  ifadesinin esas itibari ile  $A = e_1 - s_2$  gibi bir ilave boyut olduğu düşünülebilir. Ayrıca söz konusu  $A$  değerinin  $x_1, x_2 = 0$  başlangıç noktası için azami 10 olması gerektiği de eşitlik kısıtından çıkarılabilir. Standart forma dönüşmüş haldeki olası çözüm uzayı Şekil 2.28'deki gibi grafiğe dökülebilir.

Şekilden de görüldüğü üzere  $(x_1, x_2, s_1, A) = (0, 0, 10, 60)$  noktası, olası çözüm uzayında çözümün aranması için gerekli başlangıç çözüm noktası özelliklerini taşımaktadır. Ancak söz konusu nokta, gerçek sorunda olası çözüm bölgesi dışında kalmaktadır. Bu nedenle çözüm arayışında  $A$  değerinin mümkün mertebeye sıfır değerine yakınsaması için önerilen yaklaşım, minimizasyon gayeli sorunda amaç fonksiyonunun içine söz konusu değişkeni dâhil etmektir. Bu yaklaşımda aynı değişken,  $M$  gibi değeri oldukça büyük kabul edilen bir katsayı ile çarpılarak dikkate alınır. Dolayısıyla amaç fonksiyonu  $\min_x f(x) = 4x_1 + 5x_2 + M \times A$  şeklinde yazılabilir. Bununla beraber pratik uygulamalarda  $A$  yerine,  $s_2 = 0$  değeri dikkate alınarak  $A = e_1$  ve buradan  $\min_x f(x) = 4x_1 + 5x_2 + M e_1$  amaç fonksiyonu dikkate alınır. Sonraki işlem adımlarında  $M$  değerinin etkisi, özellikle bir sonraki olası çözüm noktasının seçiminde önem arz eder. Amaç fonksiyonunun  $e_1$  değerine göre bir anlamda cezalandırıldığı bu uygulamadan dolayı yöntem, "Big-M simplex" yöntemi olarak adlandırılır.

"Big-M simplex" yönteminin daha iyi anlaşılabilmesi için, değişken sayısı daha fazla olacak şekilde müteakip örneği dikkate alalım:



Şekil 2.28 İlave değişkenlerin dikkate alınması halinde olası çözüm uzayı.





$$\max_x f(x) = 6x_1 + 5x_2$$

Kısıt

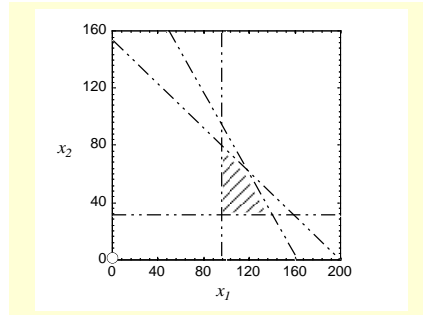
$$0.5x_1 + 0.65x_2 \leq 100$$

$$0.5x_1 + 0.35x_2 \leq 80$$

$$x_1 \geq 100$$

$$x_2 \geq 30$$

Buna göre sorun, doruklaştırma sorunu olup olası çözüm bölgesini müteakip şekildeki gibi resmedebiliriz;



Şekil 2.29 Olası çözüm bölgesi.

Tahmin edileceği üzere sorunun çözümünde yapılacak ilk işlem standart forma dönüşüm olup daha önce belirtilen dönüşüm yöntemleri dikkate alınarak sorun;

$$\min_x z = -f(x) = -6x_1 - 5x_2 + M(e_1 + e_2)$$

Kısıt

$$0.5x_1 + 0.65x_2 + s_1 = 100$$

$$0.5x_1 + 0.35x_2 + s_2 = 80$$

$$x_1 - s_3 + e_1 = 100$$

$$x_2 - s_4 + e_2 = 30$$

$$x_1, x_2, s_1, s_2, s_3, s_4, e_1, e_2 \geq 0$$

şeklinde yeniden yazılabilir. Son iki kısıt denklemini dikkate alınarak özellikle amaç fonksiyonu bünyesindeki  $e_i$  değişkenleri yerine  $x_i$  ve  $s_i$  değişkenleri kullanılabilir. Bu durumda;

$$e_1 = 100 - x_1 + s_3$$

$$e_2 = 30 - x_2 + s_4$$

eşitlikleri yazılarak amaç fonksiyonunda yerlerine konulup, uygun düzenlemeler yapılırsa;



$$z = -6x_1 - 5x_2 + M(100 - x_1 + s_3 + 30 - x_2 + s_4) = -(6 + M)x_1 - (5 + M)x_2 + Ms_3 + Ms_4 + 130M$$

$$z - 130M = -(6 + M)x_1 - (5 + M)x_2 + Ms_3 + Ms_4$$

denkleme ulaşılır. Elde edilen amaç fonksiyonu denklemleri ve kısıt denklemlerinin beraber dikkate alınması sonucu başlangıçtaki “simplex” çizelgesini;

$x_1$	$x_2$	$s_1$	$s_2$	$s_3$	$s_4$	$e_1$	$e_2$	$b$	Bağımlı değişkenler
1	0	0	0	-1	0	1	0	100	$e_1$
0	1	0	0	0	-1	0	1	30	$e_2$
0.5	0.65	1	0	0	0	0	0	100	$s_1$
0.5	0.35	0	1	0	0	0	0	80	$s_2$
$-(6+M)$	$-(5+M)$	0	0	M	M	0	0	$z-130M$	Z

şeklinde oluşturmak mümkündür. Müteakip işlem adımları kapsamında öncelikle en alt satırdaki amaç fonksiyonuna ait katsayılara bakılır ve bu değerlerden negatif işaretli ve en küçük değeri barındıran sütun (burada  $-(6 + M)$ ) dikkate alınır. Bu sütuna ait değerler en sağda bulunan  $b$  sütunundaki değerlere (amaç fonksiyonu satırı hariç) bölünür ve içlerinden en küçük pozitif sayıyı veren satır belirlenir. Burada dikkate alınacak diğer bir husus, değeri sıfır olan katsayıların bölme işlemine tabi olmayacağıdır. Buna göre en küçük pozitif sayı  $(100/1)$  olup pivot satırı ilk satır, çizelgenin bağımlı değişkenler sütunundan çıkarılacak değişken  $e_1$  değişkeni ve buna mukabil aynı konuma girecek değişken  $x_1$  değişkenidir. Bilahare pivot satırı haricindeki diğer satırlara pivotlama işlemleri yapılır;

$x_1$	$x_2$	$s_1$	$s_2$	$s_3$	$s_4$	$e_1$	$e_2$	$b$	Bağımlı değişkenler
1	0	0	0	-1	0	-	0	100	$x_1$
0	1	0	0	0	-1	-	1	30	$e_2$
0	0.65	1	0	0.5	0	-	0	50	$s_1$
0	0.35	0	1	0.5	0	-	0	30	$s_2$
0	$-5-M$	0	0	-6	M	-	0	$Z+600-30M$	Z

Yenilenen çizelgede amaç fonksiyonu satırındaki negatif en küçük değer,  $M$  değerinin çok büyük olduğu düşünüldüğünde,  $-(5 + M)$  olduğu söylenebilir. Buna mukabil  $b$  sütunundaki değerlere bölüm sonrası en küçük pozitif sayının,  $(30/1)$  ile ikinci satırda olduğu görülmektedir. Dolayısıyla  $x_2$  değişkeni, bağımlı değişken olarak  $e_2$  değişkeni ile yer değiştirir. Pivot elemanının bulunduğu sütundaki diğer katsayıların sıfırlanması için pivotlama işlemleri yapılır. İkinci pivotlama işlemleri sonrasında müteakip çizelgeyi yazmak mümkündür;



$x_1$	$x_2$	$s_1$	$s_2$	$s_3$	$s_4$	$e_1$	$e_2$	$b$	Bağımlı değişkenler
1	0	0	0	-1	0	-	-	100	$x_1$
0	1	0	0	0	-1	-	-	30	$x_2$
0	0	1	0	0.5	0.65	-	-	30.5	$s_1$
0	0	0	1	0.5	0.35	-	-	19.5	$s_2$
0	0	0	0	-6	-5	-	-	Z+750	Z

Bu çizelgenin amaç fonksiyonu satırındaki negatif en küçük değerin  $-6$  olduğu görülmektedir. Aynı sütundaki en küçük pozitif bölüm değeri ise  $(19.5/0.5 = 39)$  olup buna göre bağımlı değişken listesinden  $s_2$  değişkeni çıkmakta ve yerine  $s_3$  değişkeni girmektedir. Pivotlama işlemlerine, önce pivot elemanının (burada 0.5) bir olması için tüm satırın 2 ile çarpılmasıyla başlanır. Bilahare diğer sütun elemanları pivotlama işlemlerine tabi tutularak sıfırlanır. Üçüncü pivotlama işlemleri nihayetinde müteakip çizelgeye ulaşılabilir;

$x_1$	$x_2$	$s_1$	$s_2$	$s_3$	$s_4$	$e_1$	$e_2$	$b$	Bağımlı değişkenler
1	0	0	2	0	0.7	-	-	139	$x_1$
0	1	0	0	0	-1	-	-	30	$x_2$
0	0	1	-1	0	0.3	-	-	11	$s_1$
0	0	0	2	1	0.7	-	-	39	$s_3$
0	0	0	12	0	-0.8	-	-	Z+984	Z

Yeni çizelgenin amaç fonksiyonu satırında hala negatif değer var olup  $s_4$  sütununda  $-0.8$  değerinin olduğu görülmektedir. Aynı sütundaki en küçük pozitif bölüm değeri  $(11/0.3 = 36.6)$  olup buna göre bağımlı değişken listesinden  $s_1$  değişkeni çıkmakta ve yerine  $s_4$  değişkeni girmektedir. Pivotlama işlemlerine, önce pivot elemanının (burada 0.3) bir olması için tüm satırın  $10/3$  ile çarpılmasıyla başlanır. Bilahare diğer sütun elemanları pivotlama işlemlerine tabi tutularak sıfırlanır. Dördüncü pivotlama işlemleri nihayetinde müteakip çizelge elde edilir;

$x_1$	$x_2$	$s_1$	$s_2$	$s_3$	$s_4$	$e_1$	$e_2$	$b$	Bağımlı değişkenler
1	0	-2.33	4.33	0	0	-	-	113.33	$x_1$
0	1	10/3	-10/3	0	0	-	-	66.66	$x_2$
0	0	10/3	-10/3	0	1	-	-	36.66	$s_4$
0	0	-2.33	4.33	1	0	-	-	13.338	$s_3$
0	0	2.66	9.33	0	0	-	-	Z+1013.32	Z

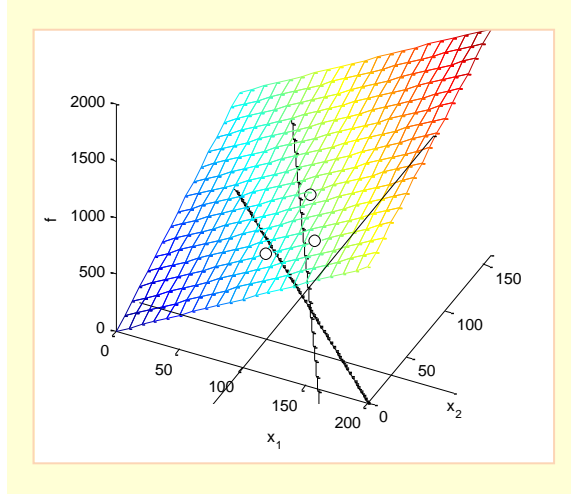




Çizelgeye ait son satır kontrol edildiğinde herhangi bir negatif değerin bulunmadığı görülür. Dolayısıyla optimal çözüme ulaşılmış olup en uygun çözümün;

$$(x_1, x_2, s_1, s_2, s_3, s_4, e_1, e_2) = (113.33, 66.66, 0, 0, 13.338, 36.66, 0, 0)$$

noktası ve buna karşılık gelen amaç fonksiyonu değerinin ise  $z = 1013.32$  olduğu söylenebilir. Müteakip şekilde üçüncü çizelgeden itibaren olası çözüm bölgesini çevreleyen hat üzerindeki çözüm noktaları görülmektedir.



Şekil 2.30 Üçüncü adımdan itibaren ziyaret edilen amaç fonksiyonu üzerindeki kesişim noktaları.

### 2.3.3 “Karmarkar” yöntemi

Doğrusal eniyileme sorununun, kısıtların çevrelediği hat üzerindeki kesişim noktalarından hareketle çözülmesi yaklaşımı, kısıt sayısının artması ile oldukça uzun zaman alabilir. Bu durumda dış yüzeyden hareket etmek yerine, doğrudan iç bölgede hareket ederek çözüme ulaşmak daha kısa bir zamanda gerçekleştirilebilir. İçsel nokta değişimi bazlı söz konusu yöntemlerden biri olan Karmarkar’ın izdüşüm ölçeklendirme yöntemi, literatürde Dantzig’in “simplex” yönteminden daha etkin bir yöntem olarak görülmektedir<sup>(23)</sup>. Bununla beraber

<sup>i</sup> Projective scaling method



nispeten az sayıdaki kısıtın söz konusu olduğu sorunlarda, “simplex” yönteminin çok daha verimli olduğunu gözlemek mümkündür. İlave olarak Karmarkar’ın yönteminin, hesaplama yükü açısından değilse de programlama zorluğu açısından bazı dezavantajlar taşıdığı da söylenebilir. İzdüşüm ölçeklendirme yönteminin uygulanabilmesi için öncelikle sorunun Denklem 2.6’daki gibi tanımlanması veya bu tanıma dönüştürülmesi gerekir;

$$\begin{aligned} \min_x z &= \mathbf{c}^T \mathbf{x} && (2.6) \\ \text{Kısıt} \quad & && \\ & \mathbf{A}\mathbf{x} = \mathbf{0} && \\ & \mathbf{e}_n^T \mathbf{x} = 1 && \\ & \mathbf{x} \geq \mathbf{0} && \end{aligned}$$

Burada  $n$  değişken sayısını ( $n \geq 2$ ),  $\mathbf{e}_n$  birim matrisi ifade etmektedir. Böyle bir tanımda sorunun çözümü adına dikkate alınan ilk nokta  $\mathbf{x}_{(0)} = 1/n \mathbf{e}_n$  ve sorunun, çözüm noktasında  $z = 0$  olduğu varsayılır. Yöntem esas itibari ile doğrusal bir denklem sisteminde gradyan bilgisine dayanarak optimal çözümün eldesi fikrine dayanmaktadır. Dolayısıyla  $\mathbf{x}^{(0)}$  başlangıç noktası olmak üzere her adımda  $\mathbf{x}^{(i)}$  gibi yeni bir olası çözüm noktası elde edilir. Yeni noktanın eldesinde yol gösterici amaç fonksiyonunun gradyan bilgisi, adım büyüklüğünü belirleyen ise belli bir çerçevede kullanıcının kendisidir.

Adım büyüklüğü, sorunun çözümünde esasen önemli bir rol oynamaktadır. Zira gradyan bilgisi amaç fonksiyonunda azami azalışı sağlayan yön bilgisini vermekle beraber, atılan adım sonrasında olası çözüm bölgesinden uzaklaşılabilir. Bunun anlamı, dikte edilen kısıtların karşılanamamasıdır. Bu nedenle adım büyüklüğü belli bir sistematik çerçevesinde ölçeklendirilerek olası çözüm uzayında kalınmasını sağlayacak şekilde hesaplanır. Dikkat edileceği üzere  $\mathbf{e}_n^T \mathbf{x} = 1$  ifadesi  $\mathbf{x} \geq \mathbf{0}$  kısıtı ile beraber düşünüldüğünde, geometrik bağlamda “simplex” şeklini ifade etmektedir. Örneğin üç değişkenin söz konusu olduğu bir ortamda söz konusu denklem takımı üçgen prizmayı temsil eder. Prizmanın hipotenüs yüzeyinin merkezi aynı zamanda  $\mathbf{x}^{(0)} = 1/n \mathbf{e}_n$  noktasıdır. Başlangıç noktası merkez olmak üzere böyle bir yüzeyin çevrelediği en büyük kürenin yarıçapı;

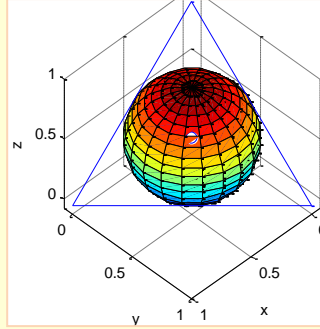
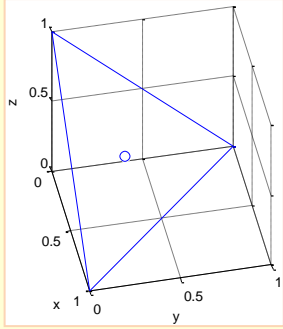
$$r = \frac{1}{\sqrt{n(n-1)}} \quad (2.7)$$

ifadesi ile bulunabilir.





Şekil 2.31'de  $n = 3$  için dikkate alınan  $x_1 + x_2 + x_3 = 1$  yüzeyi ve bu yüzeyin merkezi konumundaki  $(1/3, 1/3, 1/3)$  noktası, aynı yüzey çevresinin çevrelediği, yarıçapı  $r = 1/\sqrt{6}$  olan küre yerleşimi görülmektedir. Dikkate alınan yarıçap değerinin önemi, adım büyüklüğünü azami kılmasıdır. Burada önemli olan, hangi doğrultuda gidilirse gidilsin, olası çözüm uzayı içerisinde kalınarak en büyük adımı atmaktır.



(a)  $I_n x = 1$  ifadesi ve  $x \geq 0$  kısıtı ile beraber çevrelenen uzay ve hipotenüs yüzeyine ait merkezi nokta. (b) Hipotenüs yüzeyinin çevreleyebileceği en büyük küre.

Şekil 2.31 “Simplex” geometrisi ve çevrelenebilecek küre geometrisi.

Diğer önemli husus, ifade edildiği üzere bir sonraki adımda gidilecek yönün belirlenmesidir. Doğal olarak yön tayininde, amaç fonksiyonunun gradyan yönü dikkate alınır. Ancak önemli olan bu yönün olası çözüm uzayındaki izdüşümüdür. Bu nedenle adım büyüklüğü yanında gradyan yönünün izdüşümünün de hesaplanması gerekir. Her iki bilginin eldesi sonrasında  $x^{(i+1)} = x^{(i)} + \alpha d^{(i)}$  denklemi ile bir sonraki çözüm noktasına geçilir. Denklemden geçen  $d^{(i)}$  gidilecek yönü belirten doğrultu yöneyi,  $\alpha$  adım büyüklüğü adına ilave bir esneklik sağlayan sabit katsayıyı ifade etmektedir. Karmarkar'ın izdüşüm ölçeklendirme yönteminin işlem adımlarının daha iyi özümsemesi adına müteakip örnek sorun dikkate alınabilir<sup>(24)</sup>.

Buna göre;

$$\min_x f(x) = 3x_1 + 3x_2 - x_3$$

Kısıt

$$\begin{aligned} 2x_1 - 3x_2 + x_3 &= 0 \\ x_1 + x_2 + x_3 &= 1 \\ x_1, x_2, x_3 &\geq 0 \end{aligned}$$



şeklinde tanımlanan doğrusal eniyileme sorununda verili kısıt ve amaç fonksiyonu dikkate alınarak;

$$\begin{aligned} x^T &= [x_1 \ x_2 \ x_3] & A &= [2 \ -3 \ 1] \\ c^T &= [3 \ 3 \ -1] & e^T &= [1 \ 1 \ 1] \end{aligned}$$

yazılabilir. Sorunun çözüm uzayının daha da belirginleşmesi adına aşağıdaki ifadeleri yazmak mümkündür:

$$\begin{aligned} \Omega &= \{x \in \mathbb{R}^3 | Ax = 0\} \\ \Delta &= \{x \in \mathbb{R}^3 | I_3^T x = 1, x \geq 0\} \end{aligned}$$

Burada  $\Omega$  kümesi kısıt kaynaklı kümeyi,  $\Delta$  yöntem kaynaklı kısıt kümesini içermektedir. Olası çözüm kümesini  $\Omega \cap \Delta$  şeklinde bir kesişim kümesi olarak tanımlamak mümkündür. Daha formsal bir ifadeyle;

$$\begin{aligned} S &= \Omega \cap \Delta \\ \Omega \cap \Delta &= \left\{x \in \mathbb{R}^3 \mid B_{(0)}x = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, x \geq 0\right\} \\ B_{(0)} &= \begin{bmatrix} A \\ e^T \end{bmatrix} \end{aligned}$$

yazılabilir. Şekil 2.32'de olası çözüm noktalarını barındıran  $S$  kümesi görülmektedir. Olası çözüm hattı üzerinde bulunulan  $x^{(0)} = (1/3, 1/3, 1/3)$  noktasından itibaren hangi yönde ve hangi adım büyüklüğünde ilerleneceği hususunda yapılacak ilk işlem, amaç fonksiyonu temel alınarak belirlenen gradyanın olası çözüm hattındaki normalize edilmiş iz düşümünü tespit etmektir. Bu meyanda;

$$c^{(0)} = \frac{P^{(0)}c}{\|P^{(0)}c\|} \quad (2.8)$$

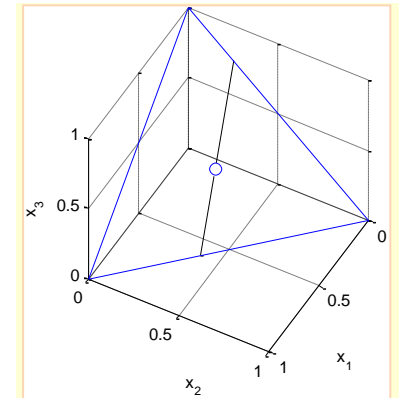
denklemleri dikkate alınır. Burada  $c^{(0)}$  normalize edilmiş izdüşüm yöneyini,  $P^{(0)}$  izdüşüm operatörünü temsil etmektedir. Söz konusu izdüşüm operatörü;

$$P^{(0)} = I_3 - B^{T,(0)}(B^{(0)}B^{T,(0)})^{-1}B^{(0)} \quad (2.9)$$

denklemleri ile elde edilebilir. Buna göre;

$$B^{(0)} = \begin{bmatrix} 2 & -3 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

olmak üzere izdüşüm operatörü,



Şekil 2.32  $x^{(0)}$  noktası (o) ve olası çözüm noktaları.





$$P^{(0)} = \begin{bmatrix} 0.3810 & 0.0952 & -0.4762 \\ 0.0952 & 0.0238 & -0.1190 \\ -0.4762 & -0.1190 & 0.5952 \end{bmatrix}$$

şeklinde hesaplanabilir. İzdüşüm operatörünün gradyan yöneyine uygulanması neticesinde;

$$c^{(0)} = \begin{bmatrix} 0.3810 & 0.0952 & -0.4762 \\ 0.0952 & 0.0238 & -0.1190 \\ -0.4762 & -0.1190 & 0.5952 \end{bmatrix} \begin{bmatrix} 3 \\ 3 \\ -1 \end{bmatrix} / 3.0861 = \begin{bmatrix} 0.6172 \\ 0.1543 \\ -0.7715 \end{bmatrix}$$

biçimindeki normalize edilmiş izdüşüm yöneyine ulaşılmış olur. Şekil 2.33'te izdüşüm yöneyinin geometrik anlamı görülmektedir. Buna göre olası çözüm hattına paralel olan ve  $\{x \in \mathbb{R}^3 | B^{(0)}x = 0, x \geq 0\}$  şartını sağlayan  $N(B^{(0)})$  hattı kullanılarak önce izdüşüm operatörü ve akabinde de  $P^{(0)}c$  izdüşüm yöneyi bulunmakta ve bilahare yöney normalize edilmektedir. Normalize edilen yöney, daha önce adım büyüklüğünü belirlemek için dikkate alınan  $r$  yarıçapı ile beraber dikkate alındığında, başlangıç noktasına eklenecek adım olan  $d^{(0)}$  elde edilir;

$$d^{(0)} = -rc^{(0)}$$

$$d^{(0)} = -1/\sqrt{6} \begin{bmatrix} 0.6172 \\ 0.1543 \\ -0.7715 \end{bmatrix} = \begin{bmatrix} -0.2520 \\ -0.0630 \\ 0.3150 \end{bmatrix}$$

Amaç fonksiyonunda azami azalmayı sağlayan  $d^{(0)}$  yöneyinin başlangıç noktasına eklenmesi neticesinde  $x^{(1)}$  noktasına ulaşılabilir;

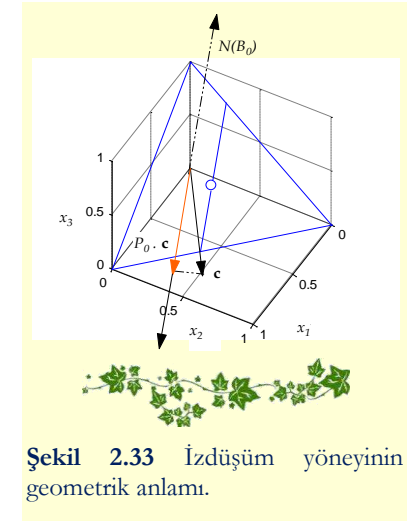
$$x^{(1)} = x^{(0)} + \alpha d^{(0)}$$

$$x^{(1)} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix} + \begin{bmatrix} -0.2520 \\ -0.0630 \\ 0.3150 \end{bmatrix} = \begin{bmatrix} 0.0814 \\ 0.2703 \\ 0.6483 \end{bmatrix}$$

Uygulamada doğrultu yöneyinin katsayısı için  $0 < \alpha \leq 1$  aralığında farklı değerlerin alınması mümkün olup burada görüldüğü üzere  $\alpha = 1$  değeri seçilmiştir. Ulaşılan yeni noktanın optimal nokta olup olmadığı müteakip ölçüte göre belirlenir;

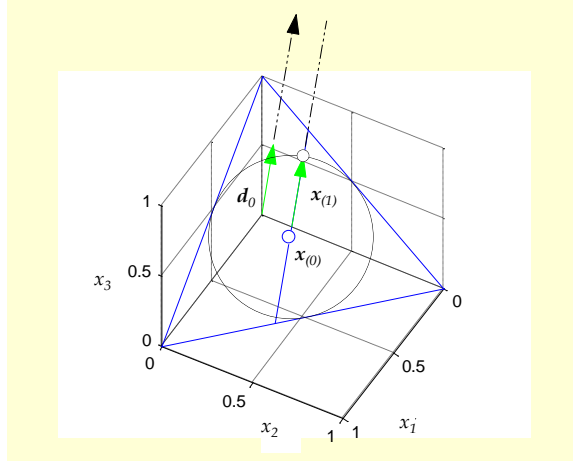
$$c^T x^{(k)} \leq \epsilon \quad (2.10)$$

Burada  $\epsilon$ , hassasiyet derecesi olup en uygun noktaya ne kadar hassasiyetle yaklaşılmak istendiğine bağlı olarak kullanıcı tarafından belirlenen bir değerdir.



Şekil 2.33 İzdüşüm yöneyinin geometrik anlamı.





**Şekil 2.34** Amaç fonksiyonunda azami azalmayı sağlayan  $\mathbf{d}^{(0)}$  yöneyi ve ilk hesaplama sonrası elde edilen  $\mathbf{x}^{(1)}$  yöneyi.

Hatırlanacağı üzere yöntemin standart formunda optimal amaç fonksiyonu değerinin 0 olduğu kabul edilmekteydi. Dolayısıyla sıfır civarı uygun bir hassasiyet değerinin seçilmesi mümkündür. Çok hassas yaklaşımların iterasyon sayısını artıracığı da unutulmamalıdır. Mevcut sorunda söz konusu çarpım değeri;

$$[3 \quad 3 \quad -1] \begin{bmatrix} 0.0814 \\ 0.2703 \\ 0.6483 \end{bmatrix} = 0.4068$$

olup seçilecek tolerans değerine göre (örneğin burada  $\epsilon = 0.01$ ) ikinci iterasyona geçilebilir. Sonraki iterasyonlarda işlem adımları, bir önceki iterasyondaki işlem adımları ile benzerlik gösterir. Ancak elde edilen yeni  $\mathbf{x}^{(1)}$  noktasının, “simplex” geometrisinin merkezinde olmaması nedeniyle uygun bir dönüşümle nokta kaydırması yapılır. Söz konusu dönüşümde ilk yapılan işlem,  $k$ 'ninci adıma ait  $D^{(k)}$  matrisini tanımlamaktır;

$$D^{(k)} = \begin{bmatrix} x_{k,1} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & x_{k,n} \end{bmatrix} \quad (2.11)$$

Müteakiben izdüşüm operatörünün elde edilmesi amacıyla;



$$B^{(k)} = \begin{bmatrix} AD^{(k)} \\ \mathbf{e}^T \end{bmatrix} \quad (2.12)$$

$$P^{(k)} = I_n - B^{T,(k)}(B^{(k)}B^{T,(k)})^{-1}B^{(k)} \quad (2.13)$$

değerleri hesaplanır. Buna göre eldeki sorun için söz konusu değerler;

$$D^{(1)} = \begin{bmatrix} 0.0814 & 0 & 0 \\ 0 & 0.2703 & 0 \\ 0 & 0 & 0.6483 \end{bmatrix}$$

$$B^{(1)} = \begin{bmatrix} 2 & -3 & 1 \\ 0.0814 & 0 & 0 \\ 0 & 0.2703 & 0 \\ 0 & 0 & 0.6483 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0.1628 & -0.8109 & 0.6483 \\ 1 & 1 & 1 \end{bmatrix}$$

$$P^{(1)} = \begin{bmatrix} 0.6427 & -0.2138 & -0.4289 \\ -0.2118 & 0.0711 & 0.1427 \\ -0.4289 & 0.1427 & 0.2862 \end{bmatrix}$$

şeklinde dir. Normalize edilmiş izdüşüm yöneyini;

$$\mathbf{c}^{(k)} = \frac{P^{(k)}D^{(k)}\mathbf{c}}{\|P^{(k)}D^{(k)}\mathbf{c}\|} \quad (2.14)$$

denklemini dikkate alarak,

$$\mathbf{c}^{(1)} = \frac{\begin{bmatrix} 0.6427 & -0.2138 & -0.4289 \\ -0.2118 & 0.0711 & 0.1427 \\ -0.4289 & 0.1427 & 0.2862 \end{bmatrix} \begin{bmatrix} 0.0814 & 0 & 0 \\ 0 & 0.2703 & 0 \\ 0 & 0 & 0.6483 \end{bmatrix} \begin{bmatrix} 3 \\ 3 \\ -1 \end{bmatrix}}{0.3263} = \begin{bmatrix} 0.8016 \\ -0.2667 \\ -0.5349 \end{bmatrix}$$

biçiminde bulmak mümkündür. Müteakiben bulunulan noktadan itibaren atılacak adım büyüklüğü hesaplanarak yeni noktaya ulaşılabilir;

$$\mathbf{d}^{(k)} = -r\mathbf{c}^{(1)} \quad (2.15)$$

$$\bar{\mathbf{x}}^{(k+1)} = \mathbf{x}^{(0)} + \alpha\mathbf{d}^{(k)} \quad (2.16)$$



Dolayısıyla;

$$\mathbf{a}^{(1)} = -1/\sqrt{6} \begin{bmatrix} 0.8016 \\ -0.2667 \\ -0.5349 \end{bmatrix} = \begin{bmatrix} -0.3273 \\ 0.1089 \\ 0.2184 \end{bmatrix}$$

$$\bar{\mathbf{x}}^{(2)} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix} + \begin{bmatrix} -0.3273 \\ 0.1089 \\ 0.2184 \end{bmatrix} = \begin{bmatrix} 0.0061 \\ 0.4422 \\ 0.5517 \end{bmatrix}$$

değerleri elde edilir. Burada dikkat edilmesi gereken husus, bulunan noktanın kaydırlmış nokta olmasıdır. Bu nedenle orijinal noktanın elde edilmesi için ters dönüşüm uygulanır;

$$\mathbf{x}^{(k+1)} = \frac{D^{(k)}\bar{\mathbf{x}}^{(k+1)}}{e^T D^{(k)}\bar{\mathbf{x}}^{(k+1)}} \quad (2.17)$$

$$\mathbf{x}^{(2)} = \frac{\begin{bmatrix} 0.0814 & 0 & 0 \\ 0 & 0.2703 & 0 \\ 0 & 0 & 0.6483 \end{bmatrix} \begin{bmatrix} 0.0061 \\ 0.4422 \\ 0.5517 \end{bmatrix}}{\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0.2703 & 0 \\ 0 & 0 & 0.6483 \end{bmatrix} \begin{bmatrix} 0.0061 \\ 0.4422 \\ 0.5517 \end{bmatrix}} = \begin{bmatrix} 0.0010 \\ 0.2502 \\ 0.7487 \end{bmatrix}$$

Elde edilen yeni noktanın optimal nokta olup olmadığı kontrol edilirse;

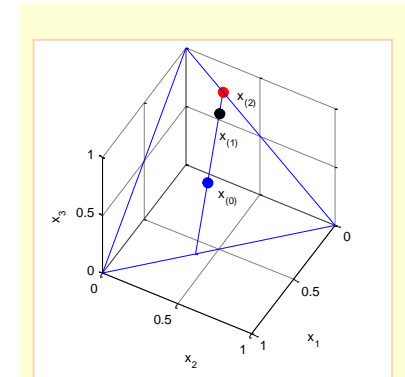
$$c^T \mathbf{x}^{(k)} \leq \epsilon \quad (2.18)$$

$$\begin{bmatrix} 3 & 3 & -1 \end{bmatrix} \begin{bmatrix} 0.0010 \\ 0.2502 \\ 0.7487 \end{bmatrix} = 0.005 < 0.01$$

değeri elde edilebilir ve uygun hassasiyete sahip optimal noktaya ulaşılmış olduğu görülür. Şekil 2.35'te başlangıç noktası dâhil iki iterasyon sonrasında ulaşılan noktalar görülmektedir.

#### 2.3.4 “Karmarkar” yönteminin ilgin seçeneği

Karmarkar yönteminde çözüm için kısıtların  $A\mathbf{x} = 0$  ve  $e_n^T \mathbf{x} = 1$  şartlarına uydurulması, dolayısıyla tanımlı doğrusal eniyileme probleminin istenen forma dönüşümü gerekir. Burada işlenecek ilgin yöntem bazı seçeneklerde ise sorunun;



Şekil 2.35 Karmarkar yöntemi ile çözülen sorunda ziyaret edilen noktalar.



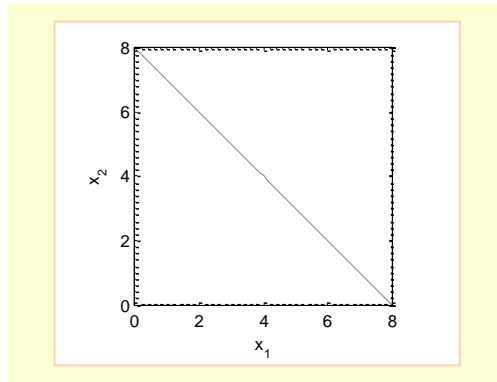


$$\begin{aligned} \text{maks}_x z &= c^T x & (2.19) \\ \text{Kısıt} \quad Ax &= b \\ x &\geq 0 \end{aligned}$$

şeklinde tanımlanması yeterlidir. Hatırlanacağı üzere kısıt denklemlerinin eşitsizlik halinden eşitlik denklemine dönüşümünde ilave değişken kullanımı öngörülmekteydi. Söz konusu yaklaşım burada da geçerlidir. Yöntemin işlem adımlarını, örnek bir sorun üzerinde göstermenin uygun olacağı değerlendirilmektedir. Buna göre aşağıdaki gibi bir doğrusal eniyileme sorununu tanımlayalım<sup>(25)</sup>;

$$\begin{aligned} \text{maks}_x f(x) &= x_1 + 2x_2 \\ \text{Kısıt} \quad x_1 + x_2 &\leq 8 \\ x_1, x_2 &\geq 0 \end{aligned}$$

Tanımlı sorunun kısıtlarının çevrelediği çözüm bölgesini müteakip grafikte gözlemek mümkündür;



Şekil 2.36 Kısıtların çevrelediği olası çözüm bölgesi.

Çözüm bağlamında öncelikle tanımlı sorunun öngörülen forma dönüştürülmesi gerekir. Buna göre sorunu;



$$\underset{x}{\text{maks}} f(x) = x_1 + 2x_2$$

Kısıt

$$\begin{aligned} x_1 + x_2 + x_3 &= 8 \\ x_1, x_2, x_3 &\geq 0 \end{aligned}$$

şeklinde yeniden yazabiliriz. Görüldüğü üzere kısıtların eşitlik denklemlerine dönüştürülmesi sonrasında ilave bir değişken ( $x_3$ ) soruna dâhil edilmiştir. Yeni sorun tanımını dikkate alarak;

$$\begin{aligned} A &= [1 \quad 1 \quad 1] \\ b &= [8] \\ c^T &= [1 \quad 2 \quad 0] \end{aligned}$$

ifadeleri yazılabilir. Ayrıca sorunun çözümü adına iterasyonlar için uygun bir başlangıç noktasının tayin edilmesi gerekir. Tahminde önemli olan, başlangıç noktasının olası çözüm uzayında olmasıdır. Bu meyanda başlangıç noktası olarak  $\mathbf{x}^{T,(0)} = [2, 2, 4]$  noktası alınabilir. Dikkat edileceği üzere  $x_1, x_2$  noktaları olası çözüm bölgesinde rastgele seçilirken, diğer değişken değeri, kısıtın dikkate alınması ile tespit edilmiştir. Başlangıç noktası, olası çözüm uzayı dâhilinde bulunmasına rağmen mevcut “simplex” geometrisinin merkezinde değildir. Bu nedenle söz konusu noktanın merkeze alınması için aşağıdaki dönüşümlerin yapılması gerekir;

$$D = \text{diag}(\mathbf{x}^{(0)}) = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 4 \end{bmatrix}$$

$$\bar{\mathbf{x}}^{(0)} = D^{-1}\mathbf{x}^{(0)} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Benzer şekilde katsayı matrisleri de dönüşüme uğrattılır;

$$\begin{aligned} \bar{A} &= AD = [2 \quad 2 \quad 4] \\ \bar{c}^T &= Dc = [2 \quad 4 \quad 0] \end{aligned}$$

Bilahare izdüşüm operatörü ile izdüşüm yöneyi;

$$P^{(1)} = I_3 - \bar{A}^T \cdot (1) (\bar{A}^{(1)} \bar{A}^T \cdot (1))^{-1} \bar{A}^{(1)} \quad (2.20)$$

$$\mathbf{c}^{(1)} = P^{(1)} \bar{\mathbf{c}} \quad (2.21)$$



denklemlerinden hareketle;

$$P^{(1)} = \begin{bmatrix} 0.8333 & -0.1667 & -0.3333 \\ -0.1667 & 0.8333 & -0.3333 \\ -0.3333 & -0.3333 & 0.3333 \end{bmatrix}$$

$$c^{T,(1)} = [1 \quad 3 \quad -2]$$

şeklinde elde edilir. Gradyan bilgisine dayalı izdüşüm yöneyinin dikkate alınmasıyla bir sonraki tasarım noktası;

$$\bar{x}^{(1)} = \bar{x}^{(0)} + \frac{\alpha}{v} c^{(1)} \quad (2.22)$$

denklemine göre bulunur. Dikkat edileceği üzere sorun, doruklaştırma sorunu olduğu için izdüşüm yöneyi doğrudan ilk noktaya eklenmektedir. Ayrıca burada  $v$  değeri, atılacak adım büyüklüğünün izdüşüm yöneyine göre şekillenmesini sağlayan bir etmen olup genellikle yöneyin negatif işaretli, mutlak değer manasında en büyük değeri olarak dikkate alınır. Denklemde geçen  $\alpha$  değeri 0.5 olarak seçilmiştir. Buna göre;

$$\bar{x}^{(1)} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + \frac{0.5}{2} \begin{bmatrix} 1 \\ 3 \\ -2 \end{bmatrix} = \begin{bmatrix} 1.25 \\ 1.75 \\ 0.5 \end{bmatrix}$$

noktası elde edilir. Ancak bu nokta, kaydırma sonrası erişilen nokta olup orijinal koordinat sistemine göre geri dönüşümü yapılırsa;

$$x^{(1)} = D\bar{x}^{(1)} \quad (2.23)$$

$$x^{(1)} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 4 \end{bmatrix} \begin{bmatrix} 1.25 \\ 1.75 \\ 0.5 \end{bmatrix} = \begin{bmatrix} 2.5 \\ 3.5 \\ 2 \end{bmatrix}$$

sonucuna varılır. Bulunan sonucun tüm kısıtları sağladığı ve amaç fonksiyonu değerini de başlangıçtaki 6 değerinden 9.5 değerine çıkardığı gözlemlenmektedir. Benzer işlem adımlarının müteaddit defalar yapılması neticesinde uğranılan olası çözümler ve optimal çözüm noktası Şekil 2.37 ve Çizelge 2.1'de görülmektedir. Buna göre on iterasyon sonunda ulaşılan nokta (0.01,7.98) olup bu noktaya karşılık gelen amaç fonksiyon değeri  $f = 16$ 'dır. İterasyonların durdurulması ölçütü olarak müteakip denklem dikkate alınabilir;

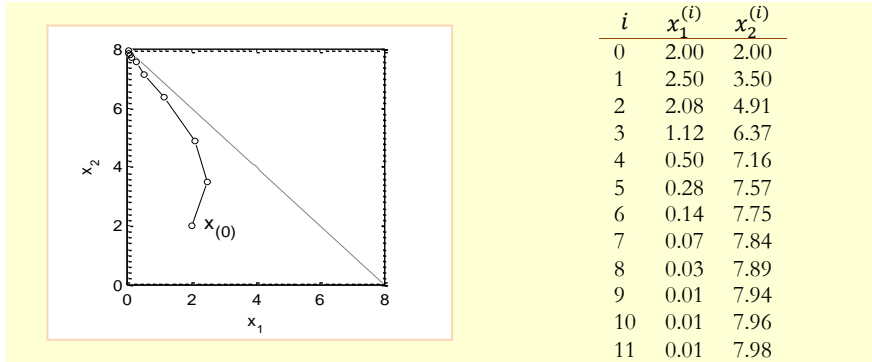


$$f^{(i+1)} - f^{(i)} \leq \epsilon \quad (2.24)$$

Örnek dâhilinde  $\epsilon$  olarak 0.05 değeri dikkate alındığında,  $f^{(11)} - f^{(10)} \leq \epsilon$  şartının sağlandığı kolayca görülebilir.

Karmarkar yöntemi, yapılan örneklerden de görüleceği üzere az sayıdaki değişken ve kısıtların söz konusu olduğu sorunlarda verimli ve hassas çözüm üreten bir yöntem değildir. Hatta istenilen hassasiyetin artması durumunda iterasyon sayısı da artmaktadır. Bunun yanında adım büyüklüğü önemli bir etmen olup seçilen  $\alpha$  değerine göre tekrar sayısı değişebilmektedir. Bazı hallerde adım büyüklüğünün büyük olması, olası çözüm uzayından dışarı çıkmak gibi ciddi bir soruna neden olabilir.

Diğer bir husus, başlangıç noktasının seçilmesidir. Yönteme göre başlangıç noktası, olası çözüm uzayının içerisinde olmalıdır. Orijinal yöntemde, sorunun istenen forma dönüştürülmesi halinde başlangıç noktasının tespiti bir sorun yaratmaz. Ancak dönüşümün olmadığı, örneğin ilgin seçeneğinde olduğu gibi, sorunlarda olası başlangıç noktasının tespiti kolay olmayabilir. Orijinal yöntemin dönüşümü de bazı hallerde ilave kabulleri ve hesaplama yükünü beraberinde getirir. Bununla beraber literatürde Karmarkar'ın yönteminin, değişken ve kısıt sayılarının çok büyük olduğu sorunlarda daha hızlı sonuçlar verdiği beyan edilmektedir<sup>(26)</sup>.



**Şekil 2.37** Karmarkar yöntemi ile on adımda ulaşılan noktalar ve optimal nokta; (0.01, 7.98).

**Çizelge 2.1** İterasyonlarda ulaşılan noktalar.



### 2.3.5 Macar algoritması

Eşleşme sorunlarında kullanılabilir yöntemlerden biri olan Macar algoritmasının H. W. Kuhn tarafından dile getirildiği, algoritmanın temellerinin ise D. König ve E. Egervary adlı Macar matematikçilerin çalışmalarından alındığı daha önce dile getirilmiştir. Burada ilave bir bilginin daha paylaşılmasında fayda görüyoruz. Son dönemde yapılan yayınlarda, eşleşme sorunlarının çözümüne yönelik daha eski çalışmaların da var olduğu ve algoritmanın bir benzerinin C. G. Jacob (1804-1851) tarafından önerildiği belirtilmektedir<sup>(27)</sup>. Her ne kadar Jacob'ın üzerinde çalıştığı sorun  $n$  tane bilinmeyen içeren  $n$  tane adi diferansiyel denklemin çözümü odaklı olsa da, problemin formülasyonu  $n \times n$  ebatlı matriste azami toplamı veren sayıların tespitine dayanmaktadır.

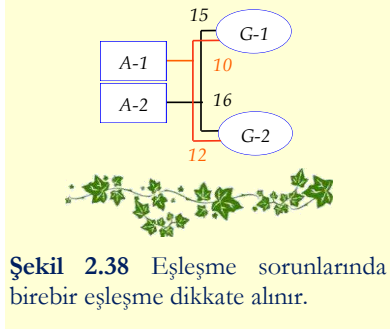
Bazı mühendislik sorunlarında  $n$  farklı kişi ya da makine,  $n$  farklı işi farklı maliyet ile yapabilir. Böyle bir durumda önemli olan, en az maliyetle kişi/makineyi işle eşleştirmektir. Örneğin iki işçi (Ali ve Veli gibi) ve yapılması gereken iki iş (cam silme ve paspaslama gibi) olduğunu dikkate alalım. Ali cam silme işini 10 TL'ye, paspaslama işlemini 15 TL'ye yapmaktadır. Buna karşılık Veli cam silme işini 12 TL'ye, paspaslama işlemini 16 TL'ye yapmaktadır. Mevcut iş yükü ile işgücü kapasitesini eşleştirme sorununda cevaplanması gereken soru, hangi işin kime verileceğidir. İş taksiminde dikkat edilecek kısıt her bir işçiye sadece bir işin verilebileceğidir. İfade edilen sorun maliyet matrisi şeklinde aşağıdaki gibi çözümlenebilir;

İşçi/İş	Cam silme	Paspaslama
Ali	10	12
Veli	15	16

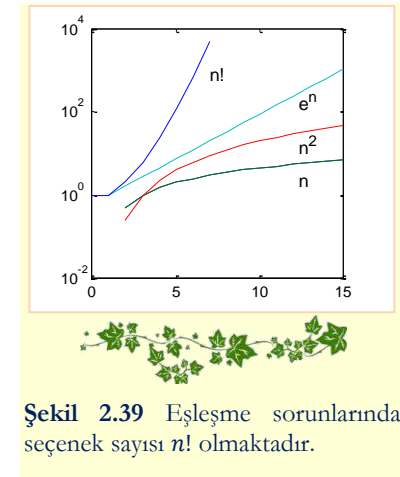
Çok az değişkenin söz konusu olduğu böyle bir sorunda cevap, tahmin edilebileceği üzere;

İşçi/İş	Cam silme	Paspaslama
Ali	10	12
Veli	15	16

şeklinindedir. Buna göre yapılacak işlerin toplam maliyeti  $10 + 16 = 26$  TL'dir. Problem dâhilinde iş/işçi sayısının artması söz konusu olduğunda, zaman karmaşıklığı ciddi manada artar. Zira  $n$  işçi ve  $n$  işin olduğu bir problemde eşleşme adına yapılabilecek seçenek sayısı  $n!$  olup böyle bir durumda zaman karmaşıklığı üstel zaman karmaşıklığından daha karmaşıktır. Daha kısa sürede çözüm bulma adına kullanılabilir en yaygın algoritmalarından birisi Macar algoritmasıdır. Algoritmik yaklaşımın temelinde müteakip önerme yatmaktadır;



Şekil 2.38 Eşleşme sorunlarında birbir eşleşme dikkate alınır.



Şekil 2.39 Eşleşme sorunlarında seçenek sayısı  $n!$  olmaktadır.

- Maliyet matrisi dâhilinde herhangi bir satır veya sütuna ait tüm matris elemanlarına yapılacak ekleme veya çıkarma işlemi, oluşan yeni matris ile orijinal matris arasında, en uygun çözüm açısından bir fark teşkil etmez.

Söz konusu önermenin doğruluğu basit bir yaklaşımla izah edilebilir. Hatırlanacağı üzere eşleşme sorunu;

$$\min_{\mathbf{c}} f(\mathbf{c}) = \sum_{i=1}^n \sum_{j=1}^n c_{i,j} x_{i,j} \quad (2.25)$$

Kısıt

$$\begin{aligned} \sum_{i=1}^n x_{i,j} &= 1 \\ \sum_{j=1}^n x_{i,j} &= 1 \\ 0 &\leq x_{i,j} \leq 1 \end{aligned}$$

formunda tanımlanmıştı. Burada  $c_{i,j}$  maliyet çizelgesindeki her bir değeri ifade etmekteydi. Herhangi bir  $q$  satırındaki maliyet matrisi elemanlarından  $k$  gibi bir sayıyı çıkardığımızı düşünelim. Bu durumda amaç fonksiyonu;

$$\tilde{f}(\mathbf{c}) = \sum_{j=1}^n (c_{q,j} - k) x_{q,j} + \sum_{i=1, i \neq q}^n \sum_{j=1}^n c_{i,j} x_{i,j} \quad (2.26)$$

şeklinde yeniden yazılabilir. Aynı ifade uygun bir düzenlemeyle;

$$\tilde{f}(\mathbf{c}) = \sum_{i=1}^n \sum_{j=1}^n c_{i,j} x_{i,j} + k \sum_{j=1}^n x_{q,j} \quad (2.27)$$

haline getirilebilir. Hatırlanacağı üzere kısıtlar çerçevesinde;

$$\sum_{j=1}^n x_{q,j} = 1$$

olmalıydı. Dolayısıyla değişen amaç fonksiyonunu;

$$\tilde{f}(\mathbf{c}) = f(\mathbf{c}) - k \quad (2.28)$$



eşitliği ile ifade etmek mümkündür. Bu ifade, minimizasyon çerçevesinde asgari  $\bar{f}(c)$  değerinin aynı zamanda asgari  $f(c)$  değeri anlamına geldiğini, dolayısıyla eşleşme meyanında aynı eşleşmenin geçerli olduğunu göstermektedir. Benzer yaklaşım, sütun işlemleri için de düşünülebilir. İfade edilen önerme haricinde dikkate alınan diğer bir husus, çözüm süreci sonrasında ulaşılabilecek optimal noktada maliyetin sıfır olacağıdır.

Polinomiyal zaman dilimi içerisinde çözüm üretebilen Macar algoritmasının işlem adımlarını, yukarıda ifade edilen örnek üzerinde göstermenin faydalı olacağı değerlendirilmektedir. Buna göre iş ve işçi sayısı  $n = 2$  olan ve işçiye göre iş maliyetleri değişen yukarıdaki sorunu tekrar dikkate alalım. Algoritmanın ilk işlem adımında her bir satıra ait asgari değer tespit edilerek çizelgenin en sağına ilave bir sütun halinde sıralanır;

İşçi/İş	Cam silme	Paspaslama	Minimum satır
Ali	10	12	10
Veli	15	16	15

Müteakiben oluşturulan yeni sütun diğer her bir sütundan çıkarılarak yeni çizelge oluşturulur;

İşçi/İş	Cam silme	Paspaslama
Ali	0	2
Veli	0	1

Algoritmanın ikinci işlem adımında, her bir sütuna ait asgari değerler tespit edilir ve çizelgenin en altına ilave bir satır eklenerek tespit edilen değerler sıralanır;

İşçi/İş	Cam silme	Paspaslama
Ali	0	2
Veli	0	1
Minimum sütun	0	1

Akabinde oluşturulan yeni satır, diğer her bir satırdan çıkarılarak yeni çizelge oluşturulur;

İşçi/İş	Cam silme	Paspaslama
Ali	0	1
Veli	0	0



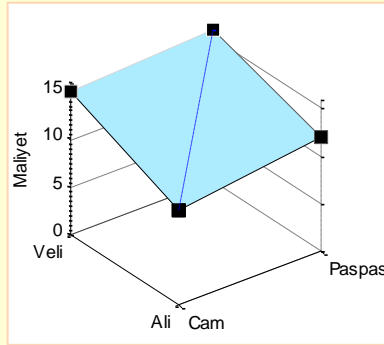


Oluşan yeni çizelgede, değeri sıfır olan matris elemanları kapsanacak şekilde, tüm hat boyunca  $k$  adet yatay veya dikey çizgiler çizilir. Bu süreçte önemli olan husus, en az sayıda çizgi ile bu işlemi gerçekleştirmektir. Buna göre toplamda  $k = 2$  adet hat ile tüm sıfır değerli matris elemanları kapsanabilmektedir;

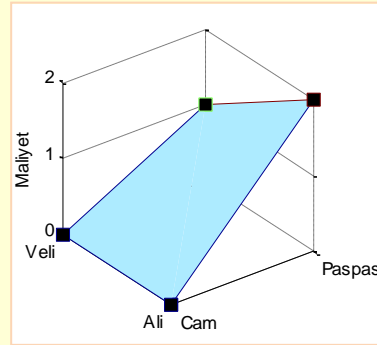
İşçi/İş	Cam silme	Paspaslama
Ali	0	1
Veli	0	0

Elde edilen çözümün en uygun çözüm olup olmadığı  $k = n$  eşitliğinin, ya da diğer bir ifade ile işçi sayısı kadar işin eşleştirilip eşleştirilmediğine bağlıdır. Dolayısıyla yapılan işlemler neticesinde optimal çözüme ulaşılmış olup elde edilen yeni çizelgeye göre minimum maliyet sıfırdır;

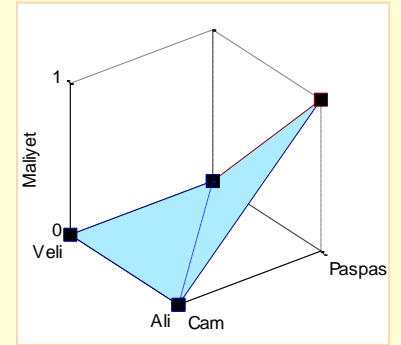
İşçi/İş	Cam silme	Paspaslama
Ali	0	1
Veli	0	0



(a) Başlangıç hali.



(b) İlk işlem adımı sonrası durum.



(c) İkinci işlem adımı sonrası durum.

Şekil 2.40 Macar algoritması yaklaşımının grafiksel gösterimi.

Yöntemin başlangıcında ifade edilen “elde edilen son maliyet matrisinin optimal çözümü ile orijinal maliyet matrisinin en uygun çözümü arasında fark olmadığı” önermesi hatırlanacak olursa, sorunun optimal çözümünün;





İşçi/İş	Cam silme	Paspaslama
Ali	10	12
Veli	15	16

eşleşmesi olduğu söylenebilir. Buna göre belirtilen iş-işçi eşleşmesi sonucu asgari maliyet  $10 + 16 = 26 TL$ 'dir. Şekil 2.40 dâhilinde işçi-iş ve maliyet grafiği, ilk ve ikinci adım sonrası durum görülmektedir. Dikkat edileceği üzere Şekil 2.40(a) seçeneğindeki iş-işçi eşleşmesi ile Şekil 2.40(c) seçeneğindeki iş-işçi eşleşmesi aynıdır. Ayrıca optimal çözümde maliyetin sıfır olduğu, Şekil 2.40(c)'de net bir biçimde görülmektedir.

Algoritmanın ikinci adımında,  $k = n$  eşitliğinin sağlanamaması halinde yapılması gereken işlem adımlarını başka bir örnek sorun üzerinde gösterebiliriz. Buna göre  $n = 4$  iş-işçinin söz konusu olduğu başka bir maliyet matrisini dikkate alalım<sup>(28)</sup>;

İşçi/İş	G - 1	G - 2	G - 3	G - 4
A - 1	90	75	75	80
A - 2	35	85	55	65
A - 3	125	95	90	105
A - 4	45	110	95	115

Algoritmanın ilk iki işlem adımı uygulandığında müteakip çizelgeye ulaşılabilir;

İşçi/İş	G - 1	G - 2	G - 3	G - 4
A - 1	15	0	0	0
A - 2	0	50	20	25
A - 3	35	5	0	10
A - 4	0	65	50	65

Oluşan yeni çizelgede değeri sıfır olan matris elemanları kapsanacak şekilde, tüm hat boyunca yatay veya dikey çizgiler çizildiğinde müteakip tablo ile karşılaşılır;

İşçi/İş	G - 1	G - 2	G - 3	G - 4
A - 1	15	0	0	0
A - 2	0	50	20	25
A - 3	35	5	0	10
A - 4	0	65	50	65



Buna göre toplamda  $k = 3$  adet hat ile tüm sıfır değerli matris elemanları kapsanabilmektedir. Ancak  $n = 4$  ve  $k \neq n$  olması nedeniyle ulaşılan nokta optimal nokta değildir. Bu aşamada yapılacak olan işlem, herhangi bir çizgi ile kapsanmayan matris elemanları arasından asgari değeri tespit ederek bu değeri tamamen yatay çizgi ile kapsanmamış satır elemanlarından çıkarmaktır:

İşçi/İş	G - 1	G - 2	G - 3	G - 4
A - 1	15	0	0	0
A - 2	0	50	20	25
A - 3	35	5	0	10
A - 4	0	65	50	65

İşçi/İş	G - 1	G - 2	G - 3	G - 4
A - 1	15	0	0	0
A - 2	0-5=-5	50-5=45	20-5=15	25-5=20
A - 3	35-5=30	5-5=0	0-5=-5	10-5=5
A - 4	0-5=-5	65-5=60	50-5=45	65-5=60

Akabinde aynı değer, dikey çizgi ile kapsanmış tüm sütun elemanlarına eklenir;

İşçi/İş	G - 1	G - 2	G - 3	G - 4
A - 1	15+5=20	0	0+5=5	0
A - 2	-5+5=0	45	15+5=20	20
A - 3	30+5=35	0	-5+5=0	5
A - 4	-5+5=0	60	45+5=50	60

Elde edilen yeni çizelgede değeri sıfır olan matris elemanları kapsanacak şekilde, tüm hat boyunca tekrar yatay veya dikey çizgiler çizildiğinde müteakip tablo ile karşılaşılır;

İşçi/İş	G - 1	G - 2	G - 3	G - 4
A - 1	20	0	5	0
A - 2	0	45	20	20
A - 3	35	0	0	5
A - 4	0	60	50	60

Görüldüğü üzere toplamda  $k = 3$  adet hat ile tüm sıfır değerli matris elemanları kapsanabilmektedir. Ancak  $k \neq n$  olması nedeniyle ulaşılan nokta hala optimal nokta değildir. Bu



nedenle bir önceki adımda yapılan işlem adımları tekrar edilir. Buna göre herhangi bir çizgi ile kapsanmayan matris elemanları arasındaki asgari değer 20 olup tamamen yatay çizgi ile kapsanmamış satır elemanlarından çıkarılır:

İşçi/İş	G - 1	G - 2	G - 3	G - 4
A - 1	20	0	5	0
A - 2	0	45	20	20
A - 3	35	0	0	5
A - 4	0	60	50	60

İşçi/İş	G - 1	G - 2	G - 3	G - 4
A - 1	20	0	5	0
A - 2	0-20=-20	45-20=25	20-20=0	20-20=0
A - 3	35	0	0	5
A - 4	0-20=-20	60-20=40	50-20=30	60-20=40

Müteakiben aynı sayı, dikey çizgi ile kapsanmış tüm sütun elemanlarına eklenir;

İşçi/İş	G - 1	G - 2	G - 3	G - 4
A - 1	20+20=40	0	5	0
A - 2	-20+20=0	25	0	0
A - 3	35+20=55	0	0	5
A - 4	-20+20=0	40	30	40

Elde edilen yeni çizelgede, değeri sıfır olan matris elemanları kapsanacak şekilde, tüm hat boyunca tekrar yatay veya dikey çizgiler çizildiğinde müteakip tablo ile karşılaşılır;

İşçi/İş	G - 1	G - 2	G - 3	G - 4
A - 1	40	0	5	0
A - 2	0	25	0	0
A - 3	55	0	0	5
A - 4	0	40	30	40

Elde edilen son çözümde,  $k = n$  eşitliğinin sağlanması nedeniyle optimal çözüme ulaşılmış olup elde edilen yeni çizelgeye göre minimum maliyet sıfırdır;



İşçi/İş	G - 1	G - 2	G - 3	G - 4
A - 1	40	0	5	0
A - 2	0	25	0	0
A - 3	55	0	0	5
A - 4	0	40	30	40

Bulunan çözüm, orijinal sorun için de geçerli olup aşağıdaki iş-işçi eşleşmesi dâhilinde asgari maliyetin  $45 + 95 + 55 + 80 = 275$  TL olduğu görülür;

İşçi/İş	G - 1	G - 2	G - 3	G - 4
A - 1	90	75	75	80
A - 2	35	85	55	65
A - 3	125	95	90	105
A - 4	45	110	95	115



Şekil 2.41 Örnek iş makineleri ve inşaat bölgesi.

Bir inşaat firmasının dört farklı yerde araç parkı olduğunu ve sorumluluğu bünyesinde dört farklı inşaat sahasının bulunduğunu düşünelim<sup>(29)</sup>. Araç park sahaları ile inşaat şantiyelerinin arasındaki mesafe kilometre olarak aşağıdaki gibi verilmektedir. Buna göre hangi araç park sahası hangi inşaat şantiyesi için görevlendirilmelidir?

Park/Şantiye	Ş - 1	Ş - 2	Ş - 3	Ş - 4
P - 1	14	5	8	7
P - 2	2	12	6	5
P - 3	7	8	3	9
P - 4	2	4	6	10



Örnek uygulama



Algoritmanın ilk işlem adımında, her bir satıra ait asgari değer tespit edilerek çizelgenin en sağına ilave bir sütun halinde sıralanır; müteakiben oluşturulan yeni sütun diğer her bir sütundan çıkarılarak yeni çizelge oluşturulur;

Park/Şantiye	Ş - 1	Ş - 2	Ş - 3	Ş - 4
P - 1	9	0	3	2
P - 2	0	10	4	3
P - 3	4	5	0	6
P - 4	0	2	4	8

Algoritmanın ikinci işlem adımında, çizelgenin son hali dikkate alınarak her bir sütuna ait asgari değerler tespit edilir ve çizelgenin en altına ilave bir satır eklenerek tespit edilen değerler sıralanır; akabinde oluşturulan yeni satır diğer her bir satırdan çıkarılarak müteakip çizelge oluşturulur;

Park/Şantiye	Ş - 1	Ş - 2	Ş - 3	Ş - 4
P - 1	9	0	3	0
P - 2	0	10	4	1
P - 3	0	5	0	4
P - 4	4	2	4	6

Oluşan yeni çizelgede, değeri sıfır olan matris elemanları kapsanacak şekilde, tüm hat boyunca  $k$  adet yatay veya dikey çizgiler çizilir. Buna göre toplamda  $k = 3$  adet hat ile tüm sıfır değerli matris elemanları kapsanabilmektedir;

Park/Şantiye	Ş - 1	Ş - 2	Ş - 3	Ş - 4
P - 1	9	0	3	0
P - 2	0	10	4	1
P - 3	0	5	0	4
P - 4	4	2	4	6

Ancak  $n = 4$  ve  $k \neq n$  olması nedeniyle ulaşılan nokta optimal nokta değildir. Bu aşamada yapılacak olan işlem, herhangi bir çizgi ile kapsanmayan matris elemanları arasından asgari değeri tespit ederek tamamen yatay çizgi ile kapsanmamış satır elemanlarından çıkarmaktır;

Park/Şantiye	Ş - 1	Ş - 2	Ş - 3	Ş - 4
P - 1	10	0	3	0
P - 2	0	9	3	0
P - 3	5	5	0	4
P - 4	0	1	3	5





Elde edilen son çizelgede, değeri sıfır olan matris elemanlarının asgari 4 hat ile kapsanabildiği görülmektedir. Dolayısıyla en uygun çözüme ulaşılmış olup park sahası ile inşaat şantiyesi eşleşmesi aşağıdaki gibidir;

Park/Şantiye	Ş - 1	Ş - 2	Ş - 3	Ş - 4
P - 1	10	0	3	0
P - 2	0	9	3	0
P - 3	5	5	0	4
P - 4	0	1	3	5

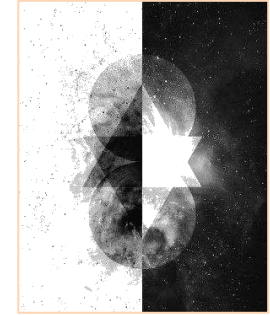
Buna mukabil, park sahaları ile şantiye konumları arasındaki toplam mesafe değeri  $5 + 5 + 3 + 2 = 15$  Km olarak tespit edilmiş olur.

Park	Şantiye	Mesafe
P - 1	Ş - 2	5
P - 2	Ş - 4	5
P - 3	Ş - 3	3
P - 4	Ş - 1	2
Toplam		15 km



### 2.3.6 Eşleniklik

Doğrusal eniyileme sorunlarının çözümü sadedinde dikkate alınan bir diğer yaklaşım, sorunun farklı bir bakış açısı ile analiz edilmesi ve daha basit bir forma getirilerek çözüme gidilmesidir. Bu bağlamda eşleniklik yaklaşımı, sıkça dikkate alınan bir yaklaşımdır. Herhangi bir doğrusal sorun, farklı bir bakış açısı ile tekrar ifade edilebilir. Söz konusu yaklaşımda sorunun orijinal haline asli<sup>i</sup> sorun, yeni formattaki tanıma eşlenik<sup>ii</sup> sorun adı verilir. Asli sorunda amaç fonksiyonunu minimize eden tasarım değerleri, eşlenik sorunda doruklaştırma amacını da gerçekleştiren değerlerdir. Bu anlamda eşleniklik, kendi içersinde bir zıtlığı da içerir. Asli ve eşlenik sorun ikiliğini basit bir örnekle biraz daha açabiliriz<sup>(30)</sup>. İzmir'de ofis mobilyaları üretimi yapan bir "A" firmasını düşünelim. Firmanın üretim hattında masa, sıra ve sandalye bulunsun. Bir masanın üretimi için 8 kilogram ahşap ve 5 kilogram metal malzeme harcanmaktadır. Masanın



Şekil 2.42 Eşleniklik, bünyesinde zıtlığı da barındırır.

<sup>i</sup> Primal

<sup>ii</sup> Dual



satış fiyatı ise 80 TL'dir. Bir sıranın üretimi için 6 kilogram ahşap ve 4 kilogram metal malzeme harcanmaktadır. Sıranın satış fiyatı 60 TL'dir. Bir sandalyenin üretimi için 4 kilogram ahşap ve 4 kilogram metal malzeme harcanmaktadır. Sandalyenin satış fiyatı 50 TL'dir. Firmanın sahip olduğu ham malzeme miktarları ahşapta 100 kg ve metalde 60 kg ile sınırlıdır. Buna göre firmanın cirosunu doruklaştıracak üretim miktarlarını konu edinen sorun, müteakip formdaki gibi ifade edilebilir;

$$\begin{aligned} \text{maks}_x f(x) &= 80x_1 + 60x_2 + 50x_3 \\ \text{Kısıt} \quad & 8x_1 + 6x_2 + 4x_3 \leq 100 \\ & 5x_1 + 4x_2 + 4x_3 \leq 60 \\ & x_i \geq 0 \end{aligned}$$

Diğer taraftan aynı şehirde rakip başka bir "B" şirketini düşünelim. "B" Şirketi oldukça hırslı bir politika gütmekte olsun. Öyle ki "A" firmasının üretim pazarında devamlılığını sonlandırmak için şirkete ait hammadde kaynaklarının tümünü satın almak istesin. Tabiiyatıyla "B" şirketinin hammadde satın alma sürecindeki teklifi "A" firması için üretimden vazgeçilecek kadar makul, kendisi içinse minimum seviyede olmalıdır. Böyle bir sorunu aşağıdaki gibi tarif etmek mümkündür;

$$\begin{aligned} \text{min}_y f(y) &= 100y_1 + 60y_2 \\ \text{Kısıt} \quad & 8y_1 + 5y_2 \geq 80 \\ & 6y_1 + 4y_2 \geq 60 \\ & 4y_1 + 4y_2 \geq 50 \\ & y_i \geq 0 \end{aligned}$$

Görüldüğü üzere burada  $y$  değerleri hammadde birim fiyatını göstermektedir. Kısıtlar ise "A" firmasının elde edeceği satış değerlerine göre düzenlenmektedir. Dolayısıyla "A" firması herhangi bir ürün için üretim yerine hammadde satışı ile de aynı ciroyu yapabilir görünmektedir. Buna göre verilen örneğin asli sorununun;

$$\begin{aligned} \text{maks}_x f(x) &= cx && (2.29) \\ \text{Kısıt} \quad & Ax \leq b \\ & x \geq 0 \end{aligned}$$

şeklinde tanımlanması halinde eşlenik sorun da;





$$\min_{\mathbf{y}} z(\mathbf{y}) = \mathbf{y}\mathbf{b} \quad (2.30)$$

Kısıt

$$\begin{aligned} \mathbf{y}A &\geq \mathbf{c} \\ \mathbf{y} &\geq 0 \end{aligned}$$

formunda yazılabilir. Dolayısıyla  $n$  değişkene ve  $m$  kısıta sahip bir asli soruna karşılık,  $m$  değişkene ve  $n$  kısıta sahip bir eşlenik sorun tanımlamak mümkündür. Bu durum, eşlenik sorunda değişken sayısının azaltılmasına ve dolayısıyla sorunun çözümünde daha kolay bir sürecin eldesine vesile olabilir. Öte yandan asli sorunun bir minimizasyon sorunu olması halinde eşlenik sorun bir doruklaştırma sorunu olarak öngörülür. Tersisi durum da geçerlidir. Dolayısıyla asli sorun, dışbükey sorun olmasa bile eşlenik sorun dışbükey bir sorun olarak tanımlanmış olabilir<sup>(31)</sup>. İki sorun arasındaki ana bağ, katsayılar matrisi  $A$  tarafından sağlanır. Genel bir yaklaşım olarak, eşlenik sorunun eşleniği asli sorun ile aynıdır. Elde edilen amaç fonksiyon değerine göre eşleniklik, zayıf ve güçlü eşleniklik olmak üzere ikiye ayrılabilir. Zayıf eşlenikliğin tanımı; “Eğer  $\mathbf{x}$  asli sorun için ve  $\mathbf{y}$  de eşlenik sorun için olası bir çözüm ise  $\mathbf{c}\mathbf{x} \leq \mathbf{y}\mathbf{b}$  eşitsizliği geçerlidir.” şeklindedir. Zayıf eşlenikliğin temelinde eşitsizlik ifadesi yatmaktadır. Esasen iddia edilen sav kolayca ispat edilebilir. Zira;

$$\sum_j c_j x_j \leq \sum_i b_i y_i$$

eşitsizliğinde  $c_j$  yerine, eşlenik sorundaki küçük eşiti dikkate alınırsa;

$$\begin{aligned} \sum_j c_j x_j &\leq \sum_j \left\{ \sum_i y_i a_{i,j} \right\} x_j \\ \sum_j c_j x_j &\leq \sum_{i,j} y_i a_{i,j} x_j \\ \sum_j c_j x_j &\leq \sum_i \left\{ \sum_j a_{i,j} x_j \right\} y_i \end{aligned}$$

ifadeleri yazılabilir. Burada parantez içersindeki ifadenin  $\mathbf{b}$  olduğu hatırlanırsa;

<sup>i</sup> Söz konusu ifade İngilizce’de “the dual of the dual problem is the primal problem” olarak geçer.





$$\sum_j c_j x_j \leq \sum_i b_i y_i$$

sonucuna varılır. Buna mukabil güçlü eşlenikliğin tanımı, eşitsizlikten eşitliğe geçiş ile yapılır. Daha açık bir ifade ile güçlü eşleniklik; “Eğer  $\mathbf{x}^*$  asli sorun için ve  $\mathbf{y}^*$  da eşlenik sorun için en uygun çözüm ise  $\mathbf{c}\mathbf{x}^* = \mathbf{y}^*\mathbf{b}$  eşitliği geçerlidir.” önermesi ile tanımlanır<sup>(32)</sup>. Eşlenikliğin güçlü veya zayıf olması, asli sorunun optimal çözüme sahip olup olmamasına bağlıdır. Dolayısıyla sınırları belirli bir olası çözüm uzayında optimal çözüm noktasının varlığı, güçlü bir eşleniklik manasına gelir. Eşlenik sorunun çözümü sonrasında asli soruna ait tasarım değişkenlerinin eldesi, doğrusal denklem takımının çözümü sorununa indirgenebilir. Zira eldeki  $\mathbf{A}\mathbf{x} \leq \mathbf{b}$  denklemlerine ilave olarak  $\mathbf{c}\mathbf{x}^* = \mathbf{y}^*\mathbf{b}$  eşitliği dikkate alınır ve böylece  $\mathbf{x}$  yöneyi kolayca elde edilebilir. Eşleniklik kavramını geometrik meyanda örneklemek için görselliği mümkün olan, dolayısıyla grafik yöntemi ile çözülebilecek aşağıdaki gibi bir sorunu ele alalım;

$$\begin{aligned} \underset{\mathbf{x}}{\text{maks}} f(\mathbf{x}) &= 40x_1 + 35x_2 \\ \text{Kısıt} \quad & 2x_1 + 3x_2 \leq 60 \\ & 4x_1 + 3x_2 \leq 96 \\ & x_i \geq 0 \end{aligned}$$

Tanımlı sorunun çözümü (18,8) noktası olup grafik gösterimi Şekil 2.43(a) dâhilinde gözlemlenmektedir. Optimal noktada elde edilen amaç fonksiyon değeri ise 1000’dir. Diğer taraftan aynı asli sorunun eşleniğini, müteakip formda temsil etmek mümkündür;

$$\begin{aligned} \underset{\mathbf{y}}{\text{min}} z(\mathbf{y}) &= 60y_1 + 96y_2 \\ \text{Kısıt} \quad & 2y_1 + 4y_2 \geq 40 \\ & 3y_1 + 3y_2 \geq 35 \\ & y_i \geq 0 \end{aligned}$$

Eşlenik sorunun çözümü, (10/3,25/3) noktası olup grafik gösterimi Şekil 2.43(b) dâhilinde görülmektedir. Optimal noktada elde edilen amaç fonksiyon değeri, bir önceki asli sorun ile aynı olup 1000’dir. Şekil 2.43(c) ve (ç) ise asli sorunda amaç fonksiyonunun doruklaşmasını, eşlenik sorunda minimizasyonunu göstermektedir. Eşlenik sorunda elde edilen optimal değerler kullanılarak;

$$\begin{aligned} \mathbf{c}\mathbf{x}^* &= \mathbf{y}^*\mathbf{b} \\ 40x_1 + 35x_2 &= 1000 \end{aligned}$$

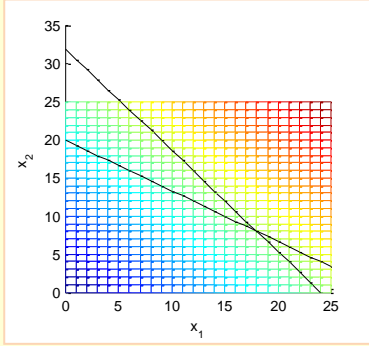




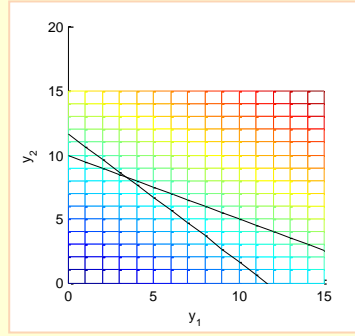
yazılabilir. Ayrıca asli soruna ait kısıtlardan herhangi biri ve mesela ilki dikkate alınarak müteakip doğrusal denklem takımı oluşturulabilir;

$$40x_1 + 35x_2 = 1000$$

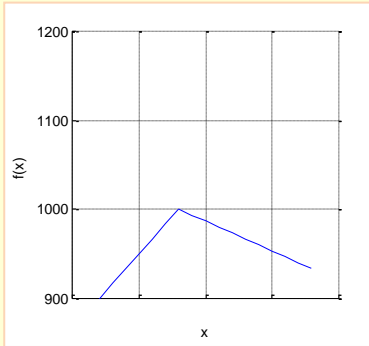
$$2x_1 + 3x_2 = 60$$



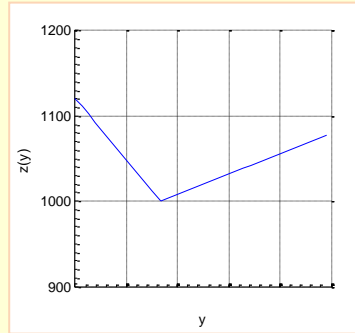
(a) Asli sorunda kısıtlar ve amaç fonksiyonu.



(b) Eşlenik sorunda kısıtlar ve amaç fonksiyonu.



(c) Asli sorunda, kısıt hattı boyunca amaç fonksiyonu değişimi.



(ç) Eşlenik sorunda, kısıt hattı boyunca amaç fonksiyonu değişimi.

Şekil 2.43 Asli ve eşlenik sorunda çözüm ortaklığı.



Oluşturulan bu denklem takımının çözümü sonrasında asli soruna ait optimal tasarım yöneyi;

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 40 & 35 \\ 2 & 3 \end{bmatrix}^{-1} \begin{bmatrix} 1000 \\ 60 \end{bmatrix} = \begin{bmatrix} 18 \\ 8 \end{bmatrix}$$

olarak elde edilir. Görüldüğü üzere bu sonuç, grafik yöntemi kullanılarak elde edilen asli sorun çözüm değerleri ile aynıdır.

## 2.4 Doğrusal olmayan programlama yöntemleri

Şimdiye kadar dikkate alınan yöntemler, esas itibari ile doğrusal problemlerin çözümü odaklıydı. Herhangi bir kısıtın olmadığı doğrusal problemde çözüm aramanın anlamı olmayabilir, zira her bir tasarım değişkeninin artması veya azalması belli bir yöndedir. Dolayısıyla tasarım değişkenine bağlı olarak azami veya asgari değer birliktelikleri doğrudan doğruya amaç fonksiyonuna ait azami veya asgari noktanın belirlenmesinde kullanılabilir. Diğer taraftan doğada ve pek çok mühendislik probleminde doğrusal olmayan sorunlar yaygın olarak görülmektedir. Doğrusal sorunların aksine doğrusal olmayan amaç fonksiyonlarında tasarım değişkenlerinin değerine göre yer yer artma veya azalma olabilir. Bu nedenle, herhangi bir kısıt olmasa dahi, azami veya asgari değer bulunması kolay olmayabilir. Hatırlanacağı üzere amaç fonksiyonunda sadece bir tepe veya çukur ya da diğer bir ifade ile engebe olduğunda, sorun dışbükey problem olarak adlandırılmaktaydı. Bu tür sorunlarda uç değer bulunması nispeten kolay olup sadece zaman ve hassasiyet hususları önem kazanır. Ancak sorunun dışbükey olmaması halinde, birden fazla uç nokta ve dolayısıyla engebenin varlığı söz konusudur. Bu durumda sadece uç noktanın değeri değil, hangi uç noktanın en uygun nokta olduğu da araştırılacak diğer bir mesele haline gelir. Kısıtlı veya kısıtsız doğrusal olmayan sorunların çözümünde pek çok yöntem kullanılmakta olup bu yöntemleri öncelikle analitik ve sayısal yöntemler şeklinde ikiye ayırabiliriz. Sayısal yöntemleri de kendi içerisinde gradyan bilgisine ihtiyaç duyulan ve duyulmayan yöntemler olarak ikiye ayırmak mümkündür.

### 2.4.1 Analitik yöntemler

Doğrusal olmayan bir amaç fonksiyonunun, herhangi bir kısıt söz konusu değilken uç noktasının/noktalarının bulunması, doğrudan doğruya fonksiyonun birinci mertebeden türevi ile ilişkilendirilebilir. Zira uç noktalarda eğimsel olarak bir dönüşüm söz konusu olup artmadan azalmaya veya azalmadan artmaya geçiş olur. Geçiş noktasının bulunduğu yerde eğimin sıfır olduğunu, dolayısıyla bir uç noktanın var olduğunu söyleyebiliriz. Bununla beraber analitik yöntemin kullanılabilmesi için öncelikle amaç fonksiyonunun matematiksel bir model/fonksiyon





olarak tanımlanabilmesi, fonksiyonun türevlenebilir olması ve asgari birinci mertebeden türevinin alınabilmesi gerekir. Çözümde dikkate alınan önermeyi aşağıdaki gibi ifade etmek mümkündür;

- ▶ Herhangi bir  $a \leq x \leq b$  aralığında tanımlı  $f(x)$  fonksiyonunun birinci mertebeden türevinin sıfır olduğu nokta, kesinlikle bir dönüşüm noktası, bununla beraber muhtemel bir uç noktasıdır.

Dolayısıyla  $f'(x) = 0$  olan noktanın  $x^* = x$  olabileceği söylenebilir. Bununla beraber aynı noktanın, daha önce de değinildiği üzere bir durağan nokta olma ihtimali de söz konusudur. Diğer taraftan çözüm sadedinde karşılaşılabilecek başka bir sorun da,  $f'(x) = 0$  ifadesi sonrasında elde edilen denklemin yine analitik olarak çözülüp çözülemeyeceği hususudur. Zira yüksek dereceden denklemlerin analitik çözüm yöntemleri olmayabilir. Bu durumda başvurulacak çare, başlangıçta belirtildiği gibi sayısal yöntemlerin kullanılmasıdır. Amaç fonksiyonuna ait birinci mertebeden türev bilgisine dayanılarak bulunan noktanın azami veya asgari nokta olup olmadığı, aynı fonksiyonun ikinci mertebeden türevine bakılarak netleştirilebilir. Buna göre  $f''(x) > 0$  olması durumunda eldeki uç nokta bir minimum noktadır. Aksi halde,  $f''(x) < 0$ , eldeki nokta maksimum noktadır.



$$f(x) = x^2 - 5x + 6$$



Örnek uygulama

şeklinde tanımlanan amaç fonksiyonunun optimal noktasını, aynı fonksiyona ait birinci mertebeden türev verisine bakarak tespit etmek mümkündür. Buna göre optimal nokta;

$$\begin{aligned} f'(x^*) &= 0 \\ f'(x) &= 2x - 5 = 0 \\ x^* &= 2.5 \end{aligned}$$

olarak elde edilebilir. Buna mukabil, bulunan  $x^*$  noktasının bir tepe veya çukur noktası olup olmadığının tespiti ise;

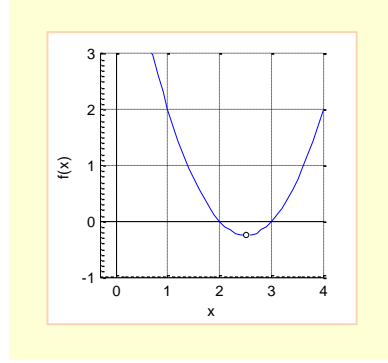
$$f''(x^*) = 2$$

eşitliği dikkate alınarak netleştirilebilir.





İkinci mertebeden türevin  $f''(x^*) > 0$  olması nedeniyle, eldeki optimal nokta bir asgari noktadır.



Şekil 2.44  $f(x) = x^2 - 5x + 6$  fonksiyonuna ait optimal nokta (o) çözümü.



$$f(x) = x^4 - 8x^3 + 22x^2 - 24x + 20$$

şeklindeki amaç fonksiyonunun uç nokta değerinin ne olduğunu tespit etmek amacıyla birinci mertebeden türev verisine bakalım;

$$f'(x^*) = 0$$

ifadesi dikkate alınarak;

$$\begin{aligned} f'(x^*) &= 4x^3 - 24x^2 + 44x - 24 = 0 \\ 4(x^3 - 6x^2 + 11x - 6) &= 0 \\ x^3 - 6x^2 + 11x - 6 &= 0 \end{aligned}$$

denklemleri kolayca elde edilir. Türev verisinden hareketle elde edilen denklem, 3üncü dereceden doğrusal olmayan denklemdir. Söz konusu denklemin analitik çözümü, uygun bir çarpanlara ayırma yaklaşımıyla;

$$\begin{aligned} (x - 1)(x - 2)(x - 3) &= 0 \\ x &= 1 \\ x &= 2 \\ x &= 3 \end{aligned}$$

şeklinde bulunabilir.





Görüldüğü üzere amaç fonksiyonunun üç farklı uç noktası bulunmaktadır. Hangi noktanın ne tür bir nokta olduğunu netleştirmek adına amaç fonksiyonunun ikinci mertebeden türevi alınırsa;

$$f'(x^*) = 12x^2 - 48x + 44$$

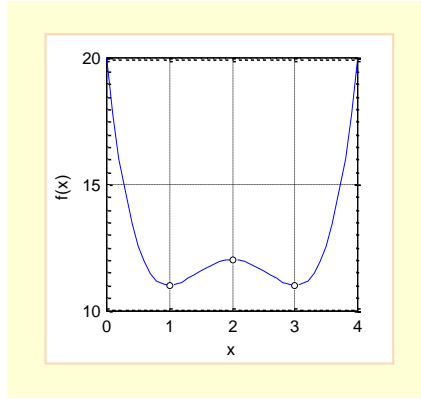
denklemini ve bu ifadede her bir nokta değeri dikkate alınırsa;

$$f''(1) = 8 > 0$$

$$f''(2) = -4 < 0$$

$$f''(3) = 8 > 0$$

değerlerine ulaşılır.



**Şekil 2.45**  $f(x) = x^4 - 8x^3 + 22x^2 - 24x + 20$  fonksiyonuna ait optimal nokta (o) çözümü.

Elde edilen bu değerler ışığında  $x = 1$  noktasının bir minimum noktası,  $x = 2$  noktasının bir maksimum noktası ve  $x = 3$  noktasının da yine bir minimum noktası olduğu görülür. Eldeki iki minimum noktadan hangisinin optimal nokta olduğu, bizzat amaç fonksiyonunun o noktadaki değerine bakılarak belirlenir. Dolayısıyla;  $f(1) = 11$  ve  $f(3) = 11$  değerlerine göre her iki noktanın, amaç fonksiyonu açısından aynı özelliğe sahip olduğu söylenebilir. Şekil 2.45'te uç noktalar ve fonksiyon değerleri görülmektedir.





Tasarım yöneyinin çok boyutlu olması halinde azami veya asgari nokta tespiti;

$$\frac{\partial f(x^*)}{\partial x_i} = 0, \quad i = 1, 2, \dots, n \quad (2.31)$$

denklemlerinin çözümü ile elde edilebilir. Burada  $n$ , tasarım yöneyinin değişken sayısını ifade etmektedir. Elde edilen uç noktanın azami veya asgari nokta olup olmadığı, ikincil türev bilgisine göre netleştirilir. Hatırlanacağı üzere ikincil türev, bir kare matris niteliğindedir. Bulunulan nokta koordinatlarının dikkate alındığı ikincil türev matrisinin pozitif tanımlı olması halinde uç nokta, bir minimum noktasıdır. Aynı matrisin negatif tanımlı olması durumunda uç nokta, bir maksimum noktasıdır. Matrisin pozitif veya negatif tanımlı olmaması halinde, diğer bir ifade ile tanımsız olması durumunda geometrinin bir semer geometrisi olduğu ve bulunulan noktanın da bir azamilerin asgarisi noktası olduğu söylenebilir.



$$f(x) = x_1^2 + (x_2 - x_1)^2 + x_2^2 - 3x_2$$



Örnek uygulama

şeklindeki amaç fonksiyonunun uç nokta değerinin ne olduğunu tespit etmek amacıyla öncelikle birinci mertebeden kısmi türev verilerine bakalım;

$$\frac{\partial f(x^*)}{\partial x_1} = 2x_1 - x_2 = 0, \quad \frac{\partial f(x^*)}{\partial x_2} = 4x_2 - 2x_1 = 3$$

ifadeleri dikkate alınarak elde edilen iki bilinmeyenli iki denklem eşanlı olarak çözülmüşür;

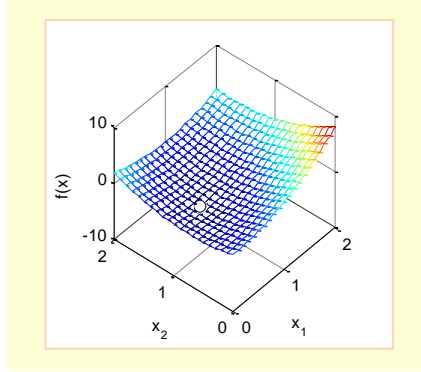
$$x_1 = \frac{1}{2}, \quad x_2 = 1$$

noktasına ulaşılır. Bulunan noktanın karakterinin tespiti amacıyla ikincil türev matrisi teşkil edilir ve özdeğerleri hesaplanırsa;

$$\begin{bmatrix} 2 & -1 \\ -2 & 4 \end{bmatrix}, \quad \begin{aligned} \lambda_1 &= 1.268 \\ \lambda_2 &= 4.732 \end{aligned}$$

değerleri bulunur. Görüldüğü üzere  $\lambda_i > 0$  olup matris pozitif tanımlıdır. Dolayısıyla bulunulan nokta, bir minimum noktasıdır. Şekil 2.46'da amaç fonksiyonu yüzeyi ve minimum noktası görülmektedir.





Şekil 2.46  $f(x) = x_1^2 + (x_2 - x_1)^2 + x_2^2 - 3x_2$  fonksiyonuna ait en uygun nokta çözümü (o).



$$f(x) = x_1^2 - x_2^2$$

şeklindeki amaç fonksiyonunun uç nokta değerinin ne olduğunu tespit etmek amacıyla birinci mertebeden kısmi türev verilerine bakalım;

$$\frac{\partial f(x^*)}{\partial x_1} = 2x_1 = 0, \quad \frac{\partial f(x^*)}{\partial x_2} = -2x_2 = 0$$

ifadeleri dikkate alınarak;

$$x_1 = 0, \quad x_2 = 0$$

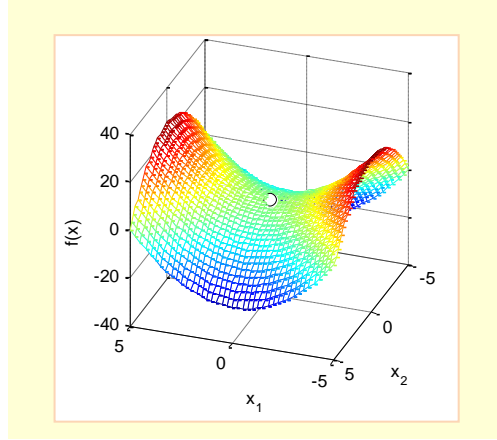
noktasına ulaşılır. Bulunan noktanın karakterinin tespiti amacıyla ikincil türev matrisi teşkil edilir ve özdeğerleri hesaplanırsa;

$$\begin{bmatrix} 2 & 0 \\ 0 & -2 \end{bmatrix}, \quad \begin{array}{l} \lambda_1 = -2 \\ \lambda_2 = +2 \end{array}$$

değerleri bulunur. Görüldüğü üzere  $\lambda_1 < 0$  ve  $\lambda_2 > 0$  olup matris tanımsızdır. Bu nedenle elde edilen nokta, semer geometrisi dâhilindeki azamilerin asgarisi noktasıdır. Şekil 2.47'de amaç fonksiyonu yüzeyi ve en uygun konum görülmektedir.







Şekil 2.47  $f(x) = x_1^2 - x_2^2$  fonksiyonuna ait optimal nokta çözümü (o).



#### a. Analitik yöntemlerde eşitlik kısıtlı eniyileme süreci

Doğrusal olmayan eniyileme süreçlerinde analitik olarak tanımlanan amaç fonksiyonuna ilave olarak yine analitik olarak tanımlanan kısıt denklemleri dikkate alınabilir. Söz konusu kısıt denklemleri, eşitlik kısıtı olabileceği gibi eşitsizlik kısıtı niteliğinde de olabilir. Eşitlik kısıtının var olduğu sorunların çözümü, eşitsizlik kısıtlı sorunlara göre daha kolaydır. Genel olarak eşitlik kısıtlarını dikkate alan eniyileme sorununu;

$$\begin{aligned} & \min_{\mathbf{x}} f(\mathbf{x}) \\ \text{Kısıt} \quad & h_i(\mathbf{x}) = 0, \quad i = 1, \dots, m \end{aligned}$$

2.32

formunda ifade edebiliriz. Burada  $f(\mathbf{x})$ , doğrusal olmayan karakterdeki amaç fonksiyonunu,  $\mathbf{x}$  tasarım yöneyini temsil etmekte olup  $n$  değişkenden ibarettir. Eşitlik kısıtlı eniyileme sorununun analitik yöntemlerle çözümü sadedinde başlıca iki yöntemden bahsedilecek olup yöntemlerden ilki yerine koyma yöntemi, diğeri Lagrange çarpanları yöntemidir.





## (1) Yerine koyma yöntemi

Yerine koyma yönteminin temelinde, kısıt denklemlerinden faydalanılarak bir değişkenin diğer bir değişken cinsinden ifade edilmesi ve amaç fonksiyonunda yerlerine konularak çözüme gidilmesi yatmaktadır. Böylelikle kısıtlar, amaç fonksiyonuna dâhil edilerek kısıtsız sorun çözümü ile aynı çözüm zeminine dönüşüm sağlanmış olur. Hatırlanacağı üzere kısıtları barındıran amaç fonksiyonu kullanılarak optimalite şartını barındıran kısmi türev bazlı denklemler elde edilebilmekte ve bu denklemler eşanlı olarak çözülüp sonuca gidilmekteydi. Yöntemin daha iyi anlaşılabilmesi için Şekil 2.48'de belirtildiği gibi bir sorun ele alınabilir; buna göre yarıçapı  $1$   $br$  olan dairenin içerisine, merkezi daire merkezi ve alanı maksimum olacak şekilde bir dikdörtgen yerleştirilim. Sorun, dikdörtgenin kenar uzunluklarının belirlenmesi sorunudur. Söz konusu sorunu müteakip formda ifade etmek mümkündür;

$$\begin{aligned} \text{maks}_x f(x) &= 4ab \\ \text{Kısıt} \quad a^2 + b^2 - r^2 &= 0 \end{aligned}$$

Burada  $4ab$  dikdörtgenin alanını,  $a^2 + b^2 = r^2$  kısıtı ise dikdörtgenin daire içerisinde kalması gereğini göstermektedir. Verilen kısıttan hareketle  $b$  değişkeni  $a$  değişkenine bağlı olarak ifade edilebilir;

$$b = \sqrt{1 - a^2}$$

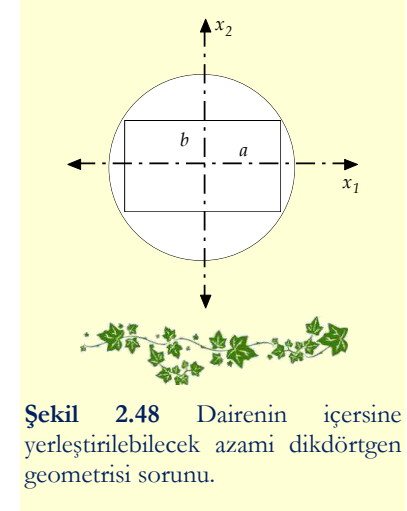
Bu eşitlik, amaç fonksiyonunda dikkate alınırsa;

$$f(x) = 4a\sqrt{1 - a^2}$$

denklemi elde edilir. Görüldüğü üzere bu denklem verili kısıtı da ihtiva etmektedir. Çözüm amacıyla fonksiyonun birinci mertebeden türevi alınarak sıfıra eşitlenirse;

$$\begin{aligned} \frac{\partial f(a^*)}{\partial a} &= 4\sqrt{1 - a^2} - \frac{4a^2}{\sqrt{1 - a^2}} = 0 \\ a^* &= \frac{1}{\sqrt{2}} \end{aligned}$$

elde edilir. Bu değer  $b$  değişkeninde dikkate alınırsa;



Şekil 2.48 Dairenin içerisine yerleştirilebilecek azami dikdörtgen geometrisi sorunu.





$$b^* = \sqrt{1 - \left(\frac{1}{\sqrt{2}}\right)^2} = \frac{1}{\sqrt{2}}$$

sonucuna ulaşılır. Buna mukabil amaç fonksiyonu değeri  $f(x^*) = 2br^2$  olarak bulunur. Bulunan noktanın azami veya asgari olup olmadığını tespit etmek için ikincil türev değerine bakılırsa;

$$\frac{\partial^2 f(a)}{\partial a^2} = \frac{-12a}{\sqrt{1-a^2}} - \frac{4a^3}{\sqrt{1-a^2}(1-a^2)}$$

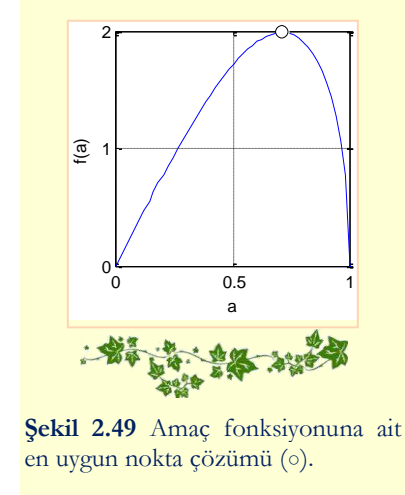
$$f''(a^*) = -16$$

değeri elde edilir. Dolayısıyla eldeki nokta bir maksimum noktadır. Şekil 2.49'da amaç fonksiyonu ve en uygun nokta görülmektedir. Yerine koyma yöntemi analitik fonksiyonlarda etkin bir yöntem gibi görünmesine rağmen, bazı hallerde değişkenlerin birbirleri cinsinden ifade edilmesi mümkün olmayabilir. Özellikle iç içe girmiş fonksiyonlarda değişkenlerden birinin yalnız bırakılması ve diğer değişkenler tarafından analitik olarak ifade edilmesi zor, hatta bazı hallerde imkânsızdır.

## (2) “Lagrange” çarpanları yöntemi

Doğrusal olmayan sorunlarda kısıtların dikkate alınması ile çözüme giden yöntemlerden bir diğeri, Lagrange çarpanları yöntemidir. Yöntemin temelinde kısıt fonksiyonlarının amaç fonksiyonuna dâhil edilmesi ve bu sayede birinci mertebeden türev bilgisine dayanılarak çözüme gidilmesi yatmaktadır. Dolayısıyla daha önceden bahsedilen yerine koyma yöntemi ile esas itibarı ile benzerlik arz etmektedir. Yöntemin anlaşılması için amaç fonksiyonu ve sadece bir kısıtın söz konusu olduğu Şekil 2.50'deki gibi nispeten basit bir sorunu dikkate alalım. Şekilden de görüleceği üzere optimal nokta öyle bir noktadır ki hem  $f(x)$  ve hem de  $h(x)$  fonksiyonu üzerinde bulunmakta, ayrıca mümkün olan en küçük veya büyük amaç fonksiyonu değerini vermektedir. Dolayısıyla  $f(x)$  amaç fonksiyonu ile  $h(x)$  kısıt fonksiyonu bu noktada birbirine teğettir. Teğet doğruya dik olarak var olan doğrultuya, hatırlanacağı üzere gradyan doğrultusu adı verilmekteydi. Bu nedenle en uygun noktada amaç fonksiyonu ile kısıt fonksiyonlarının gradyanları, şiddetleri farklı da olsa doğrultu bağlamında birbirlerine paralellik arz eder. Buradan hareketle;

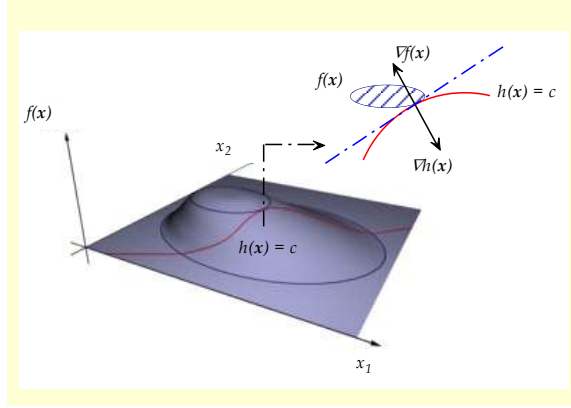
$$\nabla f(x) = -\lambda \nabla h(x) \quad (2.33)$$



Şekil 2.49 Amaç fonksiyonuna ait en uygun nokta çözümü (o).



yazılabilir. Denklemden  $\lambda$ , bir katsayı olup Lagrange çarpanı adıyla anılır.



Şekil 2.50 Lagrange çarpanları yönteminde temel yaklaşımın gösterimi.

Bu ifadeden hareketle hem amaç fonksiyonunu ve hem de kısıt denklemini içersinde barındıran daha genel bir fonksiyon tanımlamak mümkündür;

$$L(\mathbf{x}, \lambda) = f(\mathbf{x}) + \lambda h(\mathbf{x}) \quad (2.34)$$

Burada  $L$ , Lagrange fonksiyonu olarak adlandırılır. Kısıt sayısının artması durumunda Lagrange fonksiyonu;

$$L(\mathbf{x}, \lambda) = f(\mathbf{x}) + \sum_{i=1}^m \lambda_i h_i(\mathbf{x}) \quad (2.35)$$

şeklinde genişletilebilir. Diğer taraftan optimal noktada Lagrange fonksiyonunun birinci mertebeden türev değerleri her bir değişken için sıfır olacaktır;

$$\nabla L(\mathbf{x}, \lambda) = 0 \quad (2.36)$$

Bununla beraber Lagrange fonksiyonunun tasarım yöneyi değişkenlerine göre türetilmesi veya yöney değişkenlerine ilave olarak Lagrange çarpanlarına göre de türetilmesi düşünülebilir. İlk yaklaşımda sadece tasarım değişkenlerine göre türetim dikkate alınırsa,  $n$  adet denklem elde edilebilir. Oysa Lagrange fonksiyonunda bilinmeyen sayısı  $n + m$  adettir. Dolayısıyla ilave  $m$  adet



denkleme ihtiyaç duyulur. İhtiyaç duyulan ilave denklemler, bizzat eşitlik kısıtları dikkate alınarak elde edilebilir. Dolayısıyla;

$$\begin{aligned} \frac{\partial L}{\partial x_j}(\mathbf{x}^*, \boldsymbol{\lambda}) &= 0, & i = 1, 2, \dots, n \\ h_i(\mathbf{x}) &= 0, & i = 1, 2, \dots, m \end{aligned} \quad (2.37)$$

ifadelerinden elde edilen denklem takımının çözümü ile optimal noktaya ulaşılmış olur. Diğer seçenekte Lagrange fonksiyonu, tasarım yöneyi değişkenleri ile Lagrange çarpanlarına göre türetilebilir ve optimal nokta şartı dikkate alınabilir;

$$\frac{\partial L}{\partial x_j}(\mathbf{x}^*, \boldsymbol{\lambda}^*) = 0, \quad i = 1, 2, \dots, n \quad (2.38)$$

$$\frac{\partial L}{\partial \lambda_j}(\mathbf{x}^*, \boldsymbol{\lambda}^*) = 0, \quad j = 1, 2, \dots, m \quad (2.39)$$

Her iki yaklaşımda da elde edilen çözüm aynıdır. Bulunan optimal noktanın azami veya asgari nokta olup olmadığı, oluşturulacak ikincil türev matrisi ve bu matrisin küçümen matrislerinin<sup>i</sup> belirteç değerlerine bakılarak belirlenir. İkincil türev matrisi müteakip formda oluşturulur;

$$\begin{bmatrix} L_{11} & L_{12} & \dots & L_{1n} & h_{11} & h_{21} & \dots & h_{m1} \\ L_{21} & L_{22} & \dots & L_{2n} & h_{12} & h_{22} & \dots & h_{m2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ L_{n1} & L_{n1} & \dots & L_{nn} & h_{1n} & h_{2n} & \dots & h_{mn} \\ h_{11} & h_{12} & \dots & h_{1n} & 0 & 0 & \dots & 0 \\ h_{21} & h_{22} & \dots & h_{2n} & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ h_{m1} & h_{m2} & \dots & h_{mn} & 0 & 0 & \dots & 0 \end{bmatrix} \quad (2.40)$$

Burada;

$$L_{ij} = \frac{\partial^2 L}{\partial x_i \partial x_j}(\mathbf{x}^*, \boldsymbol{\lambda}^*), h_{ij} = \frac{\partial h_i}{\partial x_j}(\mathbf{x}^*) \quad (2.41)$$

şeklinde tanımlanmaktadır.

---

<sup>i</sup> Minor of matrix



Oluşturan bu matrisin küçümen matrislerinin belirteci dikkate alınarak aşağıdaki koşul sağlanıyorsa;

$$(-1)^k \det(k^{\text{ninci}} \text{ küçümen matris}) < 0 \quad (2.42)$$

elde edilen noktanın azami nokta olduğu söylenebilir<sup>(33)</sup>. Bu ifade, belirteç değerinin işaret olarak değiştiğini göstermektedir. Dikkat edileceği üzere ilk küçümen matrisin belirteci her zaman negatiftir. Dolayısıyla ikinci küçümen matrisin belirtecinin pozitif olması noktanın niteliğini belirler. Aksi halde elde edilen noktanın asgari nokta olduğu öngörülebilir. Küçümen matris tanımına sağ alt köşeden başlandığını da burada ifade etmek gerekir.

Yöntemin daha iyi anlaşılabilmesi için Şekil 2.51'de belirtilen sorunu ele alalım; buna göre yarı kenar uzunlukları  $x_1, x_2$  ve  $x_3$  olan dikdörtgenler prizmasının, yarıçapı  $1$   $br$  olan bir küre içersine yerleştirildiğini düşünelim. Yerleştirme yapılırken dikkate alınan husus, dikdörtgen prizmanın hacminin doruklaştırılmasıdır. Belirtilen sorunu aşağıdaki gibi tanımlamak mümkündür;

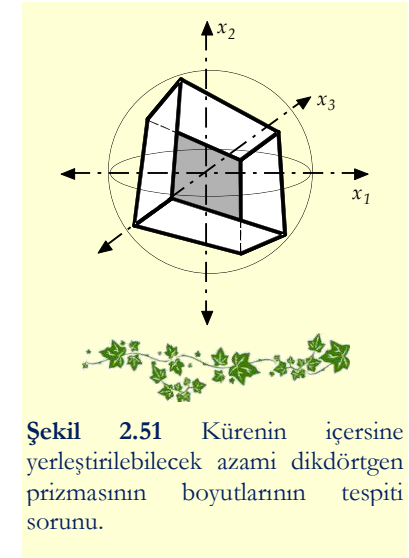
$$\begin{aligned} \text{maks}_x f(x) &= 8x_1x_2x_3 \\ \text{Kısıt} \quad x_1^2 + x_2^2 + x_3^2 - r^2 &= 0 \end{aligned}$$

Burada  $8x_1x_2x_3$  çarpımı prizmanın hacmini,  $x_1^2 + x_2^2 + x_3^2 = r^2$  denklemi prizmanın küre içersinde kalması gereğini göstermektedir. Verili amaç fonksiyonu ile kısıt denklemini kullanarak Lagrange fonksiyonunu;

$$L(x_1, x_2, x_3, \lambda) = 8x_1x_2x_3 + \lambda(x_1^2 + x_2^2 + x_3^2 - 1)$$

şeklinde yazabiliriz. Bu fonksiyonun birinci mertebeden kısmi türevleri alınır ve sıfıra eşitlenirse;

$$\begin{aligned} \frac{\partial L}{\partial x_1} &= 8x_2x_3 + 2\lambda x_1 = 0 \\ \frac{\partial L}{\partial x_2} &= 8x_1x_3 + 2\lambda x_2 = 0 \\ \frac{\partial L}{\partial x_3} &= 8x_1x_2 + 2\lambda x_3 = 0 \\ x_1^2 + x_2^2 + x_3^2 - 1 &= 0 \end{aligned}$$



Şekil 2.51 Kürenin içersine yerleştirilebilecek azami dikdörtgen prizmasının boyutlarının tespiti sorunu.



denklemleri elde edilir. İlk üç denklem yeniden düzenlenerek tasarım değişkenleri arasındaki ilişkiler kolayca elde edilebilir;

$$\begin{aligned} 2\lambda &= -\frac{8x_2x_3}{x_1} \\ 2\lambda &= -\frac{8x_1x_3}{x_2} \\ 2\lambda &= -\frac{8x_1x_2}{x_3} \\ x_1^2 &= x_2^2 = x_3^2 \end{aligned}$$

Değişkenler arasındaki bu ilişki, Lagrange çarpanına ait son denklemde yerine koyulursa;

$$\begin{aligned} x_1^2 + x_1^2 + x_1^2 &= 1 \\ 3x_1^2 &= 1 \\ x_1^* &= x_2^* = x_3^* = \frac{1}{\sqrt{3}} \end{aligned}$$

sonucuna ulaşılır. Buna mukabil Lagrange çarpanı ve amaç fonksiyonu değeri;

$$\begin{aligned} \lambda^* &= -4/\sqrt{3} \\ f(x^*) &= \frac{8}{3\sqrt{3}} \end{aligned}$$

olarak hesaplanabilir. Bulunan noktanın azami veya asgari olup olmadığını tespit etmek için ikincil türev matrisi ve küçümen matris belirteçleri dikkate alınabilir. Buna göre ikincil türev matrisi;

$$\begin{bmatrix} \frac{\partial^2 L}{\partial x_1^2} & \frac{\partial^2 L}{\partial x_1 \partial x_2} & \frac{\partial^2 L}{\partial x_1 \partial x_3} & \frac{\partial h_1}{\partial x_1} \\ \frac{\partial^2 L}{\partial x_2 \partial x_1} & \frac{\partial^2 L}{\partial x_2^2} & \frac{\partial^2 L}{\partial x_2 \partial x_3} & \frac{\partial h_1}{\partial x_2} \\ \frac{\partial^2 L}{\partial x_3 \partial x_1} & \frac{\partial^2 L}{\partial x_3 \partial x_2} & \frac{\partial^2 L}{\partial x_3^2} & \frac{\partial h_1}{\partial x_3} \\ \frac{\partial h_1}{\partial x_1} & \frac{\partial h_1}{\partial x_2} & \frac{\partial h_1}{\partial x_3} & 0 \end{bmatrix} = \begin{bmatrix} 2\lambda & 8x_3 & 8x_2 & 2x_1 \\ 8x_3 & 2\lambda & 8x_1 & 2x_2 \\ 8x_2 & 8x_1 & 2\lambda & 2x_3 \\ 2x_1 & 2x_2 & 2x_3 & 0 \end{bmatrix}$$

şeklinde oluşturulabilir. Bulunan optimal nokta değerleri  $(x^*, \lambda^*)$ , oluşturulan matriste dikkate alınır;



$$\begin{bmatrix} \frac{8}{\sqrt{3}} & \frac{8}{\sqrt{3}} & \frac{8}{\sqrt{3}} & \frac{2}{\sqrt{3}} \\ 8 & 8 & 8 & 2 \\ \frac{8}{\sqrt{3}} & -\frac{8}{\sqrt{3}} & \frac{8}{\sqrt{3}} & \frac{2}{\sqrt{3}} \\ \frac{8}{\sqrt{3}} & \frac{8}{\sqrt{3}} & -\frac{8}{\sqrt{3}} & \frac{2}{\sqrt{3}} \\ 2 & 2 & 2 & 0 \\ \frac{2}{\sqrt{3}} & \frac{2}{\sqrt{3}} & \frac{2}{\sqrt{3}} & 0 \end{bmatrix}$$

matrisi elde edilir. Bu matrisin küçümen matrisleri ve o matrislere ait belirteç değerleri ise;

$$k = 2 \quad K_2 = \begin{bmatrix} \frac{8}{\sqrt{3}} & \frac{2}{\sqrt{3}} \\ 2 & 0 \end{bmatrix} \quad \det(K_2) = -1.33$$

$$k = 3 \quad K_3 = \begin{bmatrix} \frac{8}{\sqrt{3}} & \frac{8}{\sqrt{3}} & \frac{2}{\sqrt{3}} \\ 8 & 8 & 2 \\ \frac{8}{\sqrt{3}} & -\frac{8}{\sqrt{3}} & \frac{2}{\sqrt{3}} \\ 2 & 2 & 0 \end{bmatrix} \quad \det(K_3) = 24.62$$

$$k = 4 \quad K_4 = \begin{bmatrix} \frac{8}{\sqrt{3}} & \frac{8}{\sqrt{3}} & \frac{8}{\sqrt{3}} & \frac{2}{\sqrt{3}} \\ -\frac{8}{\sqrt{3}} & \frac{8}{\sqrt{3}} & \frac{8}{\sqrt{3}} & \frac{2}{\sqrt{3}} \\ \frac{8}{\sqrt{3}} & -\frac{8}{\sqrt{3}} & \frac{8}{\sqrt{3}} & \frac{2}{\sqrt{3}} \\ \frac{8}{\sqrt{3}} & \frac{8}{\sqrt{3}} & -\frac{8}{\sqrt{3}} & \frac{2}{\sqrt{3}} \\ 2 & 2 & 2 & 0 \\ \frac{2}{\sqrt{3}} & \frac{2}{\sqrt{3}} & \frac{2}{\sqrt{3}} & 0 \end{bmatrix} \quad \det(K_4) = -341.33$$

şeklinde hesaplanabilir. Optimal noktanın azami nokta olması için karşılanması gereken şart dikkate alınırsa;

$$\begin{aligned} (-1)^2 \times -1.33 &< 0 \\ (-1)^3 \times 24.62 &< 0 \\ (-1)^4 \times -341.33 &< 0 \end{aligned}$$

eşitsizliklerinin geçerli olduğu, dolayısıyla optimal noktanın bir maksimum nokta niteliği taşıdığı söylenebilir.





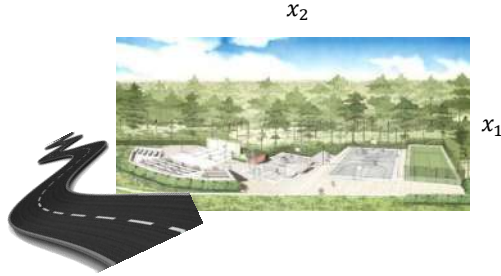


Örnek uygulama

Karayolu üzerinde  $5000 \text{ m}^2$  alana sahip bir dinlenme alanının inşa edilmesi planlanmaktadır. Alanın dikdörtgen geometrisinde olması, karayoluna bakan yüzünün çevrelenmemesi ve diğer kenarları kapsayan çit maliyetinin de minimum olması istenmektedir. Buna göre dikdörtgenin optimal mesafe değerlerini Lagrange çarpanı yöntemi ile bulalım. İfade edilen problem;

$$\begin{aligned} \min_x f(x) &= x_1 + 2x_2 \\ \text{Kısıt} \quad x_1 x_2 &= 5000 \end{aligned}$$

şeklinde tanımlanabilir. Burada  $x_1 + 2x_2$  toplamı, dinlenme alanının çitle çevrilecek mesafesini,  $x_1 x_2$  çarpımı, alan değerini göstermektedir.



Şekil 2.52 Karayolu kenarındaki dinlenme alanı geometrisi.

Amaç fonksiyonu ile kısıtı bir arada bulunduran Lagrange fonksiyonunu;

$$L(x_1, x_2, \lambda) = x_1 + 2x_2 + \lambda(x_1 x_2 - 5000)$$

şeklinde yazabiliriz. Bu fonksiyonun optimal konumda birinci mertebeden kısmi türevleri alınır ve sıfıra eşitlenirse;

$$\begin{aligned} \frac{\partial L}{\partial x_1} &= 1 + \lambda x_2 = 0 \\ \frac{\partial L}{\partial x_2} &= 2 + \lambda x_1 = 0 \\ x_1 x_2 - 5000 &= 0 \end{aligned}$$



denklemleri elde edilir. İlk iki denklem yeniden düzenlenerek tasarım değişkenleri arasındaki ilişki kolayca elde edilebilir;

$$\lambda = -\frac{1}{x_2} = -\frac{2}{x_1}$$

$$x_1 = 2x_2$$

Değişkenler arasındaki bu ilişki, Lagrange çarpanına ait son denklemde yerine koyulursa;

$$2x_2 \times x_2 - 5000 = 0$$

$$2x_2^2 = 5000$$

$$x_2^* = 50$$

$$x_1^* = 100$$

sonucuna ulaşılır. Buna mukabil Lagrange çarpanı ve amaç fonksiyonu değeri;

$$\lambda^* = -1/50$$

$$f(\mathbf{x}^*) = 200 \text{ m}$$

olarak hesaplanabilir. Bulunan noktanın azami veya asgari olup olmadığını tespit etmek için ikincil türev matrisi ve küçümen matris belirteçleri dikkate alınabilir. Buna göre ikincil türev matrisi;

$$\begin{bmatrix} \frac{\partial^2 L}{\partial x_1^2} & \frac{\partial^2 L}{\partial x_1 \partial x_2} & \frac{\partial h_1}{\partial x_1} \\ \frac{\partial^2 L}{\partial x_2 \partial x_1} & \frac{\partial^2 L}{\partial x_2^2} & \frac{\partial h_1}{\partial x_2} \\ \frac{\partial h_1}{\partial x_1} & \frac{\partial h_1}{\partial x_2} & 0 \end{bmatrix} = \begin{bmatrix} 0 & \lambda & x_2 \\ \lambda & 0 & x_1 \\ x_2 & x_1 & 0 \end{bmatrix}$$

şeklinde oluşturulabilir. Bulunan optimal nokta değerleri  $(\mathbf{x}^*, \lambda^*)$ , oluşturulan matriste dikkate alınır;

$$\begin{bmatrix} 0 & -\frac{1}{50} & 50 \\ -\frac{1}{50} & 0 & 100 \\ 50 & 100 & 0 \end{bmatrix}$$

matrisi elde edilir. Bu matrisin küçümen matrisleri ve o matrislere ait belirteç değerleri;



$$\begin{aligned}
 k = 2 \quad K_2 &= \begin{bmatrix} 0 & 100 \\ 100 & 0 \end{bmatrix} & \det(K_2) &= -1000 \\
 k = 3 \quad K_3 &= \begin{bmatrix} 0 & -\frac{1}{50} & 50 \\ -\frac{1}{50} & 0 & 100 \\ 50 & 100 & 0 \end{bmatrix} & \det(K_3) &= -200
 \end{aligned}$$

şeklinde hesaplanabilir. Optimal noktanın asgari nokta olması için karşılanması gereken şart dikkate alınır;

$$\begin{aligned}
 (-1)^2 \times -1000 &< 0 \\
 (-1)^3 \times -200 &> 0
 \end{aligned}$$

eşitsizliklerinin geçerli olduğu, dolayısıyla optimal noktanın bir minimum nokta niteliği taşıdığı söylenebilir.



## b. Analitik yöntemlerde eşitsizlik kısıtlı eniyileme süreci

Genel olarak eşitsizlik kısıtlarını dikkate alan eniyileme sorununu;

$$\begin{aligned}
 &\min_{\mathbf{x}} f(\mathbf{x}) && (2.43) \\
 \text{Kısıt} \quad &g_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, m
 \end{aligned}$$

formunda ifade edebiliriz. İfade edilen bu sorun, Lagrange çarpanları yöntemi dikkate alınarak çözülebilir. Burada dikkat edilmesi gereken husus, eşitsizliğin eşitlik kısıtına dönüştürülmesi ve akabinde daha önce belirtilen işlem adımlarının uygulanarak çözüme gidilmesidir. Eşitsizlik kısıtının eşitlik kısıtına dönüştürülmesi, ilave değişken tanımı ile yapılır. Şöyle ki;

$$g_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, m$$

şeklinde tanımlanan eşitsizlik  $y_i^2$  geçici değişken ilavesi ile,

$$g_i(\mathbf{x}) + y_i^2 = 0, \quad i = 1, \dots, m \quad (2.44)$$

eşitliği haline getirilebilir. Bu durumda yeni kısıt denklemini  $G_i(\mathbf{x}, \mathbf{y})$  olarak tanımlayabiliriz. Tanımlanan eşitlik kısıtını dikkate alarak Lagrange fonksiyonu;



$$L(\mathbf{x}, \mathbf{y}, \boldsymbol{\mu}) = f(\mathbf{x}) + \sum_{i=1}^m \mu_i G_i(\mathbf{x}, \mathbf{y}) \quad (2.45)$$

şeklinde yazılabilir. Burada  $\mu_i$ , bir diğer Lagrange çarpanı olarak tanımlanmaktadır. Tabiatıyla ilave değişken tanımı, belirlenmesi gereken toplam değişken sayısını da artırmaktadır. Bununla beraber optimal noktada ilk türev değerinin sıfır olması şartı dikkate alınarak,  $(n + 2m)$  bilinmeyene karşılık  $(n + 2m)$  denklem yazılabilir ve bu denklemler eşanlı olarak çözülebilir. Dolayısıyla;

$$\frac{\partial L}{\partial x_i}(\mathbf{x}^*, \mathbf{y}^*, \boldsymbol{\mu}^*) = 0 \quad (2.46)$$

$$\frac{\partial L}{\partial y_i}(\mathbf{x}^*, \mathbf{y}^*, \boldsymbol{\mu}^*) = 0 \quad (2.47)$$

$$\frac{\partial L}{\partial \lambda_i}(\mathbf{x}^*, \mathbf{y}^*, \boldsymbol{\mu}^*) = 0 \quad (2.48)$$

denklemleri çözümde dikkate alınan denklemlerdir. Çözüm sadedinde kullanılacak diğer bir yaklaşım, Kuhn-Tucker şartlarının dikkate alınmasıdır. 1951 yılında Kuhn ve Tucker tarafından gündeme getirilen ve daha sonraki dönemde KKT (Karush-Kuhn-Tucker) şartları olarak belirtilen<sup>(34)</sup> yaklaşım, eşitsizlik kısıtının etkisi üzerine dayandırılmaktadır. İfade edildiği üzere eşitsizlik kısıtı,  $g_i(\mathbf{x}) \leq c$ ,  $i = 1, \dots, m$  şeklinde tanımlanmaktaydı. Lagrange fonksiyonu ise amaç fonksiyonuna ilave olarak  $\mu_i g_i(\mathbf{x})$  terimleri içermekteydi. Tahmin edileceği üzere optimal noktanın  $g_i(\mathbf{x}) < 0$  bölgesinde olması durumunda, tanımlı eşitsizliğin herhangi bir etkisinin olmayacağı, dolayısıyla Lagrange denkleminde  $\mu_i$  değerinin sıfır olacağı düşünülebilir. Buna mukabil optimal noktanın  $g_i(\mathbf{x}) = 0$  üzerinde olması durumunda kısıt gereği  $g_i(\mathbf{x}^*) = 0$  olacağı aşıkardır. Buradan hareketle optimal noktada Lagrange çarpanı veya kısıt denklemleri kaynaklı, ama her halükarda geçerli  $\mu_i g_i(\mathbf{x}) = 0$  eşitliğinin varlığından söz edilebilir. Dolayısıyla çözüm sadedinde;

$$\frac{\partial L}{\partial x_i} = \frac{\partial f}{\partial x_i} + \sum_{j=1}^m \mu_j \frac{\partial g_j}{\partial x_i} = 0, \quad i = 1, \dots, n \quad (2.49)$$

$$\mu_j g_j(\mathbf{x}) = 0, \quad j = 1, \dots, m \quad (2.50)$$

denklemleri dikkate alınabilir. Görüldüğü üzere Lagrange denklemindeki  $n + m$  adet bilinmeyen değişken, birinci mertebeden türev şartı nedeni ile  $n$  denklem ve kısıt kaynaklı  $m$  denklem yardımı ile çözülebilmektedir.



### c. Genelleştirilmiş KKT şartları

Genelleştirilmiş KKT şartları, bünyesinde eşitlik ve eşitsizlik kısıtları barındıran sorunların çözümünde dikkate alınmaktadır. Daha açık bir ifade ile doğrusal olmayan eniyileme sorununun optimal noktasının sağlaması gereken şartları ifade etmektedir. Eşitlik ve eşitsizlik kısıtlarını dikkate alan eniyileme sorununu;

$$\begin{array}{l} \min_{\mathbf{x}} f(\mathbf{x}) \\ \text{Kısıt} \\ h_i(\mathbf{x}) = 0, \quad i = 1, \dots, m \\ g_j(\mathbf{x}) \leq 0, \quad j = 1, \dots, p \end{array} \quad (2.51)$$

formunda ifade edebiliriz. Amaç fonksiyonu ile kısıt denklemlerinin beraber dikkate alınması halinde Lagrange fonksiyonunu;

$$L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f(\mathbf{x}) + \sum_{i=1}^m \lambda_i h_i(\mathbf{x}) + \sum_{j=1}^p \mu_j g_j(\mathbf{x}) \quad (2.52)$$

şeklinde tanımlayabiliriz. Lagrange fonksiyonu, optimizasyon probleminin eşlenik sorununun tanımlanmasında kullanılır. Zira amaç fonksiyonunda ana gaye minimizasyon olarak tanımlanırken, Lagrange eşlenik sorununda<sup>i</sup> ana gaye;

$$\begin{array}{l} \max_{\boldsymbol{\mu}} \theta(\boldsymbol{\lambda}, \boldsymbol{\mu}) \\ \text{Kısıt} \\ \boldsymbol{\mu} \geq 0, \quad i = 1, \dots, m \end{array} \quad (2.53)$$

şeklinde tanımlanır. Burada  $\theta(\boldsymbol{\lambda}, \boldsymbol{\mu})$  fonksiyonu Lagrange fonksiyonunun uç değerleri olup;

$$\theta(\boldsymbol{\lambda}, \boldsymbol{\mu}) = \inf_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \quad (2.54)$$

denklemleri ile ifade edilir. Dolayısıyla her bir  $(\boldsymbol{\lambda}, \boldsymbol{\mu})$  değeri için Lagrange fonksiyonunun asgari değeri dikkate alınarak, asgari değerler arasından bir azami uç nokta seçimi yapılmaktadır. Buradan hareketle  $\theta(\boldsymbol{\lambda}, \boldsymbol{\mu})$  eşlenik fonksiyonunu içbükey, Lagrange fonksiyonunu ise dışbükey

<sup>i</sup> Lagrangian dual problem



olarak tanımlamak mümkündür<sup>(35)</sup>. Lagrange fonksiyonunun birinci mertebeden türevinin sıfır olması şartı dikkate alındığında;

$$\frac{\partial L}{\partial x_k}(x^*, \lambda^*, \mu^*) = \frac{\partial f(x^*)}{\partial x_k} + \sum_{i=1}^m \lambda_i^* \frac{\partial h_i(x^*)}{\partial x_k} + \sum_{j=1}^p \mu_j^* \frac{\partial g_j(x^*)}{\partial x_k} = 0, \quad k = 1, \dots, n \quad (2.55)$$

denklemleri elde edilebilir. Görüldüğü üzere toplamda  $(n + m + p)$  bilinmeyen için  $n$  denklem bu sayede elde edilebilmektedir. Sorunda verili olan eşitlik kısıtları dikkate alınarak hâlihazırda  $m$  denklem yazılabilir;

$$h_i(x^*) = 0, \quad i = 1, \dots, m \quad (2.56)$$

Denklem takımının çözümünde ihtiyaç duyulan ilave  $p$  denklem ise eşitsizlik kısıtları dikkate alınarak yazılabilir;

$$\begin{aligned} \mu_j^* g_j(x^*) &= 0, \quad j = 1, \dots, p \\ \mu_j^* &\geq 0 \end{aligned} \quad (2.57)$$

Dolayısıyla  $(n + m + p)$  bilinmeyen değişkene karşılık  $(n + m + p)$  denklem dikkate alınarak doğrusal veya doğrusal olmayan denklem takımı çözümüne gidilebilir. Bahse konu yaklaşım, analitik olarak ifade edilebilen sorunların çözümünde etkin bir yöntem gibi görünse de bazı zorlukları içerir. Çözüm sürecinde karşılaşılan en ciddi sorun, elde edilen denklem takımının analitik olarak çözülmesidir. Özellikle doğrusal olmayan terimler içeren denklemlerin varlığı analitik çözümü zor, hatta bazı hallerde imkânsız hale getirebilir. Böyle bir durumda dikkate alınan yaklaşım, doğrusal olmayan denklem takımının sayısal yöntemlerle çözülmesidir. Diğer bir husus ise elde edilen noktanın gerçekten bir minimum nokta olup olmadığıdır. Her ne kadar optimalite şartları sağlansa da, bulunan noktanın durağan nokta olması mümkündür. Bu nedenle çözüm olarak elde edilen nokta ya da noktaların denenmesi ve amaç fonksiyon değerlerinin görülmesi ya da yorumlanması gerekebilir.

Kısıt

$$\min_x f(x) = (x_1 - 1)^2 + (x_2 - \frac{1}{2})^2$$

$$2x_1 + x_2 = 2$$

$$4x_1 + x_2 \leq 12$$



Örnek uygulama



şeklinde belirtilen doğrusal olmayan eniyileme problemini, KKT şartlarını dikkate alarak çözelim. Öncelikle Lagrange fonksiyonunu;

$$L(x, \lambda, \mu) = (x_1 - 1)^2 + (x_2 - \frac{1}{2})^2 + \lambda(2x_1 + x_2 - 2) + \mu(4x_1 + x_2 - 12)$$

şeklinde tanımlayabiliriz. Bu fonksiyonun optimal noktada birinci mertebeden türevinin sıfır olması şartı dikkate alındığında;

$$\begin{aligned} \frac{\partial L}{\partial x_1} &= 2(x_1 - 1) + 2\lambda + 4\mu = 0 \\ \frac{\partial L}{\partial x_2} &= 2\left(x_2 - \frac{1}{2}\right) + \lambda + \mu = 0 \end{aligned}$$

yazılabilir. İlave olarak kısıtlar dikkate alınırrsa;

$$\begin{aligned} 2x_1 + x_2 &= 2 \\ \mu(4x_1 + x_2 - 12) &= 0 \end{aligned}$$

denklemleri elde edilir. Son yazılan denklemde  $\mu = 0$  olduğu kabul edilirse;

$$\begin{aligned} x_1 + \lambda &= 1 \\ 2x_2 + \lambda &= 1 \\ 2x_1 + x_2 &= 2 \end{aligned}$$

doğrusal denklem takımı oluşturulmuş olur. Bu denklem takımı eşanlı olarak çözümlerse;

$$\begin{aligned} x_1^* &= 0.8 \\ x_2^* &= 0.4 \\ \lambda^* &= 0.2 \end{aligned}$$

değerleri ve bulunan bu noktaya karşılık gelen amaç fonksiyon değeri de  $f(x^*, \lambda^*, \mu^*) = 0.05$  olarak bulunur. Diğer taraftan eşitsizlik kısıtı kaynaklı denklemde  $\mu \neq 0$  ve dolayısıyla  $4x_1 + x_2 - 12 = 0$  olduğu kabul edilirse;

$$\begin{aligned} 2x_1 + 2\lambda + 4\mu &= 2 \\ 2x_2 + \lambda + \mu &= 1 \\ 2x_1 + x_2 &= 2 \\ 4x_1 + x_2 &= 12 \end{aligned}$$

eşitlikleri elde edilir. Lineer denklem takımı niteliğindeki bu denklemler eşanlı olarak çözümlerse;



$$\begin{aligned} x_1^* &= 5, & x_2^* &= -8 \\ \lambda^* &= 35.5, & \mu^* &= -18.5 \end{aligned}$$

verileri ve bulunan noktaya karşılık gelen amaç fonksiyon değeri de  $f(\mathbf{x}^*, \lambda^*, \mu^*) = 88.25$  olarak belirlenir. Elde edilen bu çözüm, daha önce yöntemin anlatılması esnasında ifade edilen  $\mu_j^* \geq 0$  şartını sağlamamaktadır. Ayrıca minimizasyon gayeli sorun için elde edilen amaç fonksiyon değeri de zaten oldukça büyüktür. Dolayısıyla optimal çözüm  $f(0.8, 0.4, 0.2, 0) = 0.05$  olarak bulunabilir.



## 2.4.2 Sayısal yöntemler

Matematiksel modelin analitik çözümünün yapılamaması durumunda sayısal analiz yöntemlerine başvurulur. Analitik çözüm yöntemlerinin aksine sayısal yöntemlerde işlem tekrarı ön plandadır. Bu nedenle sayısal yöntemlerde bilgisayar destekli çözüm yöntemleri yaygın olarak kullanılmaktadır. Yaklaşımın temelinde bir başlangıç noktası seçimi, sonrasında da gidilecek yön ile o yönde atılacak adım büyüklüğünün tespiti ve bu sayede optimal noktaya veya yakın bir bölgeye ulaşmaya kadar ilerleme düşüncesi yatmaktadır. Böylelikle atılan her bir adımla en uygun noktaya biraz daha yakınlaşma hedeflenir. Başlangıç noktası  $n$  değişkene sahip  $\mathbf{x}_{(0)}$  olmak üzere bir sonraki nokta  $\mathbf{x}_{(1)} = \mathbf{x}_{(0)} + \alpha \mathbf{d}_{(0)}$  denklemi ile belirlenir. Burada  $\alpha$  adım büyüklüğünü,  $\mathbf{d}_{(0)}$  başlangıç noktasından itibaren ilerlenecek yönü göstermektedir. Atılan ilk adım sonrası takip eden adımlar ise;

$$\mathbf{x}_{(i+1)} = \mathbf{x}_{(i)} + \alpha_{(i)} \mathbf{d}_{(i)} \quad (2.58)$$

ifadesi ile elde edilir. Çözüm sadedinde dikkate alınan algoritmanın öncelikli hedefi, optimal noktaya en kısa sürede ulaşmaktır. Bu meyanda başlangıç noktasının, adım büyüklüğünün ve doğru yönün seçimi ayrı ayrı öneme sahiptir. Diğer taraftan dikkat edileceği üzere denklemdeki konum ve yön terimleri yöneysel büyüklükler, adım büyüklüğü ise sayısal bir büyüklüktür. Dolayısıyla adım büyüklüğünün bulunması bir anlamda tek değişkenli tarama süreci olarak düşünülebilir.

### a. Tek boyutlu arama yöntemleri

Tek boyutlu arama yöntemlerinin ana gayesi amaç fonksiyonunda azami azalma sağlayan adım büyüklüğünün tespitidir. Diğer bir ifade ile  $f(\mathbf{x}_{(i)} + \alpha_{(i)} \mathbf{d}_{(i)})$  değerini,  $(\mathbf{x}_{(i)}, \mathbf{d}_{(i)})$  değerleri dikkate alındığında minimum yapacak adım büyüklüğünün bulunmasıdır;





$$\alpha_{(i)}^* = \min_{\alpha} f(\mathbf{x}_{(i)} + \alpha_i \mathbf{d}_{(i)}) \quad (2.59)$$

Gidilecek yönün bulunması, çok değişkenli bir tarama süreci olarak düşünülebilir. Yaklaşım, sayısal bir yaklaşım olsa da özellikle bünye denkleminde bulunan  $\alpha_{(i)}$  adım büyüklüğü ifadesi analitik yolla da bulunabilir. Analitik yöntemin kullanılmasının mümkün olmadığı durumlarda sayısal yönteme başvurulur. Sayısal yöntem, gradyan bilgisine dayalı veya eleme bazlı yöntem olabilir. Eleme bazlı yöntemlerde gradyan bilgisine, dolayısıyla birinci mertebeden türevlenebilirlik özelliğine ihtiyaç duyulmaz. Onun yerine yerel olarak belirlenen bir aralıkta aralık daraltma yaklaşımı ile optimal değere ulaşmaya çalışılır. Gradyan esaslı yaklaşımda eğime bağlı olarak adım adım optimal noktaya ulaşma amaçlanır. Adım büyüklüğünün analitik olarak bulunması hususu önceki bölümlerde ele alınmıştı. Bununla beraber konunun netlik kazanması amacıyla aşağıdaki gibi bir sorun dikkate alınabilir;

$$f(\mathbf{x}) = x_1^2 - x_2^2$$

fonksiyonunun asgari değerini bulmak için yapılacak tarama faaliyetinde, başlangıç noktası olarak  $\mathbf{x}^T = [0 \ 4]$  ve yön olarak da  $\mathbf{d}^T = [0 \ -8]$  değerlerini dikkate alalım. Buna göre yeni nokta;

$$\mathbf{x}_{(1)} = \mathbf{x}_{(0)} + \alpha \mathbf{d}_{(0)} = \begin{bmatrix} 0 \\ 4 \end{bmatrix} + \alpha \begin{bmatrix} 0 \\ -8 \end{bmatrix} = \begin{bmatrix} 0 \\ 4 - 8\alpha \end{bmatrix}$$

olacaktır. İfadedeki adım büyüklüğünü bulmak için önce yeni nokta koordinatlarının amaç fonksiyonunda yerine konulması gerekir;

$$f(\mathbf{x}_{(1)}) = 0^2 - (4 - 8\alpha)^2 = -(4 - 8\alpha)^2$$

Elde edilen bu fonksiyon görüldüğü üzere sadece  $\alpha$  değişkenine bağlıdır. Fonksiyonun minimum değerini veren  $\alpha$  değeri;

$$\begin{aligned} \frac{\partial f}{\partial \alpha} &= 4 - 8\alpha = 0 \\ \alpha &= 0.5 \end{aligned}$$

olarak kolayca bulunabilir. Bununla beraber analitik olarak çözümün bulunamayacağı sorunlar da olabilir. Örneğin aşağıdaki gibi bir diğer sorunu dikkate alalım;



$$f(\mathbf{x}) = \sin^2 x_1 + \frac{1 - x_1}{\cos x_2}$$

fonksiyonu için başlangıç noktası olarak  $\mathbf{x}^T = [0 \ 1]$  ve yön olarak da  $\mathbf{d}^T = [1 \ 1]$  koordinatlarını dikkate alalım. Buna göre yeni nokta;

$$\mathbf{x}_{(1)} = \mathbf{x}_{(0)} + \alpha \mathbf{d}_{(0)} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} + \alpha \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha \\ 1 + \alpha \end{bmatrix}$$

olarak belirlenebilir. Bu yöneyin amaç fonksiyonunda yerine konulması halinde;

$$f(\mathbf{x}) = \sin^2 \alpha + \frac{1 - \alpha}{\cos(1 + \alpha)}$$

denklemini yazılabilir. Amaç fonksiyonunun asgari değerinin eldesi için gereken  $\alpha$  değerinin bulunması gayesiyle birinci mertebeden türev alındığında;

$$\frac{\partial f}{\partial \alpha} = 2 \sin \alpha \cos \alpha - \frac{1}{\cos(\alpha + 1)} - \frac{(\alpha - 1) \sin(\alpha + 1)}{\cos^2(\alpha + 1)} = 0$$

denklemine ulaşılır. Görüldüğü üzere bu denklemin analitik olarak çözülmesi ve adım büyüklüğünün doğrudan elde edilmesi zordur. Bu nedenle başvuru çözümlenmesi sayısal yöntemlerden biridir.

Gerek gradyan bilgisine dayalı ve gerekse de eleme bazlı yöntemlerin kullanılması halinde karşılaşılabilecek bir diğer husus, en yakın optimal değere yakınsama hususudur. Analitik çözüme dayalı yaklaşımda birinci mertebeden türev alınması ve akabinde eğimin sıfıra eşit olduğu noktaların tespiti neticesinde amaç fonksiyonunun derecesine bağlı olarak pek çok kök bulunabilir. Bulunan bu noktadaki amaç fonksiyonu değerine bakılarak hangisinin yerel, hangisinin genel optimum nokta olduğu tespit edilebilir. Sayısal yöntemlerde ise başlangıç noktasına bağlı olarak adım adım gidildiğinde en yakın optimal noktaya ulaşılır. Ancak bu noktanın yerel ya da genel optimal nokta olup olmadığını tespit etmek mümkün değildir. En yakın optimal noktaya ulaşım sadedinde yeni bir kavramdan daha bahsetmekte yarar görmekteyiz. Söz konusu kavram tek engebeli fonksiyon<sup>1</sup> kavramıdır. Örnekleri Şekil 2.53'te verilen tek engebeli fonksiyon, belirli bir aralıkta tepe veya çukur ama sadece bir engebeye sahip

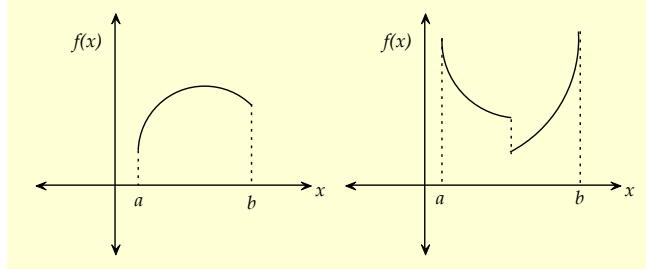
---

<sup>1</sup> Unimodal function



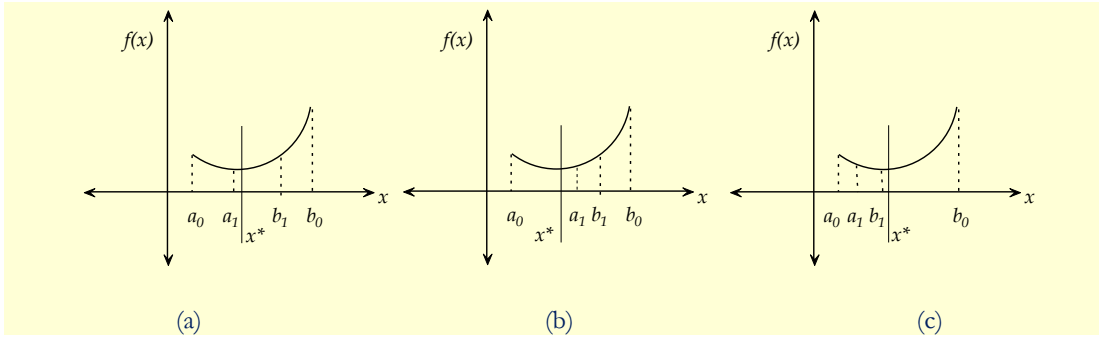


olan fonksiyondur. Tanımı gereği fonksiyonun belirtilen aralıkta süreklilik arz etmesi, diğer bir ifade ile türevlenebilir olması şart değildir. Özellikle eleme bazı yöntemlerde, değişken aralığının sürekli daraltılması ve bu sayede optimal noktaya ulaşılması hedeflendiği için dikkate alınan aralığın tek engebeye sahip bir bölge olması gerekir.



Şekil 2.53 Tek engebeli fonksiyon örnekleri.

Müteakip paragraflarda eleme bazı yöntemlerden “altın kesit” ve “Fibonacci” yöntemleri, gradyan bilgisine dayalı yöntemler arasında ise “quasi-Newton”, “Secant” ve “Newton” yöntemlerinden bahsedilecektir.



Şekil 2.54 Eleme yöntemlerinde yeni aralığın tespiti.



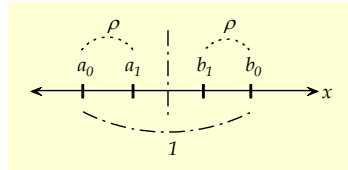


## (1) Altın kesit yöntemi

Altın kesit yöntemi, tek değişkenli tarama sorunlarının çözümünde kullanılabilir sayısal bazı bir eleme yöntemidir. İfade edildiği üzere eleme yöntemlerinde belirli bir aralığın her adımında daraltılması ve bu sayede optimal noktanın bulunduğu bölgenin belirlenmesi hedeflenir. Yeteri kadar daralan bölge aynı zamanda belirli bir hassasiyetle optimal noktanın bulunması anlamına da gelir. Altın Kesit yöntemi, ilk defa J. Kiefer (1924-1981)<sup>(36)</sup> tarafından gündeme getirilen bir yöntem olup<sup>(37)</sup> tekrarlı işlemler içeren basit ama etkili bir yöntemdir. Optimal noktası  $[a_0, b_0]$  aralığında olan tek engibeli bir  $f(x)$  fonksiyonunu dikkate alalım. Bir sonraki adımda, içinde optimal noktayı da barındıran daha dar bir aralığın belirlenmesi için  $[a_1, b_1]$  gibi en az iki yeni noktanın tespiti gerekir. Zira tek bir noktanın tayini optimal noktanın konumunun belirlenmesi için yeterli değildir. İki yeni noktanın tespiti sonrasında aynı noktadaki amaç fonksiyonu değerlerine göre kesit aralığı tespit edilebilir. Örneğin aşağıdaki Şekil 2.54(a)'da görüldüğü üzere  $[a_1, b_1]$  aralığına ait sınır noktalarının amaç fonksiyon değerleri arasında  $f(a_1) < f(b_1)$  ilişkisi geçerli ise  $x^*$  optimal noktasının  $[a_1, b_1]$  aralığında olduğu söylenebilir. Ancak aynı ilişkinin geçerli olduğu Şekil 2.54(b)'deki durumdan da görüleceği üzere  $x^*$  optimal noktasının  $[a_1, b_1]$  aralığında olduğu söylenemez. Bununla beraber her iki durumda da kesinlikle söylenebilecek öngörü,  $x^*$  optimal noktasının  $[a_0, b_1]$  aralığında olduğudur. Benzer durum  $f(a_1) > f(b_1)$  koşulu dikkate alınarak da değerlendirilebilir. Şekil 2.54(c)'de olduğu gibi söz konusu koşulun geçerli olduğu durumda,  $x^*$  optimal noktasının kesinlikle  $[a_1, b_0]$  aralığında olduğu söylenebilir. Görüldüğü üzere aralığın tahmini, amaç fonksiyonu değerlerine göre yapılabilmektedir. Aralığın sınırlarının belirlenmesi bağlamında simetrik bölmelere ayırma yaklaşımı dikkate alınabilir. Buna göre başlangıçta dikkate alınan  $[a_0, b_0]$  aralığının 1 birim mesafe değerine sahip olduğunu düşünelim. Söz konusu aralık, ortası simetri hattı olmak üzere  $[a_1, b_1]$  noktaları ile simetrik bir bölme işlemine tabi tutulabilir. Müteakip şekilden de görüleceği üzere bölünme sonrası;

$$a_1 - a_0 = b_0 - b_1 = \rho(b_0 - a_0)$$

$$\rho < \frac{1}{2}$$



yazılabilir.

<sup>i</sup> Golden section



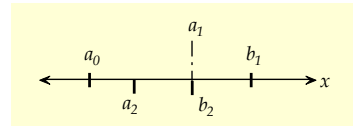
Denklemdaki  $\rho$  ifadesi, yeni aralığın belirlenmesinde dikkate alınan kesit oranı olarak tanımlanabilir. Daha öncede ifade edildiği üzere belirlenen  $[a_1 \ b_1]$  noktalarına karşılık  $f(a_1) < f(b_1)$  ilişkisi geçerli ise  $x^*$  optimal noktasının  $[a_0 \ b_1]$  aralığında olduğu söylenebilir. Sonraki adımda,  $[a_2 \ b_2]$  noktalarının tespit adına iki seçeneekli bir yol izlenebilir. İlk seçenekte, başlangıçta olduğu gibi iki ayrı nokta tespit edilebilir ve bu noktalardaki amaç fonksiyonu değerlerine bakılabilir. Ancak bu yöntemde iki ayrı nokta için iki ayrı amaç fonksiyonu hesabı gerekir. Diğer seçenekte ise noktalardan birinin daha önceki adımda bilinen noktalardan biri ( $a_1$ ) olarak seçilmesidir. Bu yaklaşımda, her adımda sadece bir kere amaç fonksiyonu hesabı gerekir. Dolayısıyla hesaplama yükü açısından ikinci seçeneğin tercih edilmesi etkin bir yöntemdir. Ancak bu sefer  $[a_2 \ b_2]$  noktalarından hangisinin  $a_1$  noktası olarak dikkate alınacağını belirlenmesi gerekir. Eldeki  $a_1$  noktasının  $a_2$  noktası olarak dikkate alınması halinde;

$$\begin{aligned} a_2 - a_0 &= \rho(b_1 - a_0) = \rho(1 - \rho) \\ a_2 - a_0 &= a_1 - a_0 = \rho \\ \rho &= \rho(1 - \rho) \\ \rho &= 1 \end{aligned}$$

elde edilir. Hatırlanacağı üzere  $\rho < 0.5$  şartı gereği bu seçenek geçerli değildir. Diğer taraftan eldeki  $a_1$  noktasının  $b_2$  noktası olarak dikkate alınması halinde;

$$\begin{aligned} b_2 - b_1 &= \rho(b_1 - a_0) = \rho(1 - \rho) \\ b_2 - b_1 &= a_1 - b_1 = 1 - 2\rho \\ \rho(1 - \rho) &= \rho(1 - \rho) \\ \rho^2 - 3\rho + 1 &= 0 \\ \rho &= \frac{3 \pm \sqrt{5}}{2} \end{aligned}$$

sonucuna ulaşılır. İki değer dikkate alındığında,  $\rho < 0.5$  şartını sağlayan  $\rho \cong 0.382$  değerinin uygun olacağı söylenebilir:



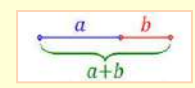
Diğer taraftan kesit oranının elde edilmesi sadedinde kullanılan ikinci dereceden denklem aşağıdaki gibi uygun şekilde düzenlendiğinde;





$$\begin{aligned}\rho^2 - 3\rho + 1 &= 0 \\ \rho^2 - 2\rho + 1 &= \rho \\ (\rho - 1)^2 &= \rho \\ \frac{1 - \rho}{1} &= \frac{\rho}{1 - \rho}\end{aligned}$$

oranı elde edilebilir. Bu ifade, bir aralığın bölünme sonrası kısa kenarının ( $\rho$ ) uzun kenarına ( $1 - \rho$ ) oranının, uzun kenarının ( $1 - \rho$ ) aralık mesafesine (1) oranına eşit olması anlamına gelmektedir. Söz konusu tanımı aşağıdaki gibi genelleştirmek mümkündür. Buna göre kısa kenar için  $b$  ve uzun kenar için  $a$  simgesi kullanılırsa;

$$\frac{b}{a} = \frac{a}{a+b}$$


yazılabilir. İfadede geçen  $a/b$  oranı  $\varphi$  olarak tanımlanır ve uygun düzenlemeler yapılırsa;

$$\begin{aligned}\varphi &= \frac{\varphi + 1}{\varphi} \\ \varphi^2 - \varphi - 1 &= 0 \\ \varphi &= \frac{1 \pm \sqrt{5}}{2}\end{aligned}$$

sonucuna ulaşılır. Tanımlı oranın pozitif olması gereği dikkate alındığında,  $\varphi \cong 1.618$  değeri elde edilir. Literatürde gerek  $\rho \cong 0.382$  ve gerekse de  $\varphi \cong 1.618$ <sup>i</sup> değerleri “altın oran” terimi ile bilinir. Doğadan sanata pek çok uygulamaları olan altın oranın eniyileme süreçlerindeki uygulamasını müteakip örnekle göstermek mümkündür. Buna göre minimizasyon bazlı optimizasyon problemini;

$$\min f(x) = x^3 - 3x^2 - x$$

amaç fonksiyonu ile tanımlayalım. Sorunun altın kesit yöntemi ile çözümü sadedinde yapılacak ilk işlem, başlangıç aralığının ve çözüm hassasiyetinin tespitidir. Çözüm hassasiyetinden kastedilen, optimal noktanın bulunduğu aralığın hassasiyetidir. Başlangıç aralığı olarak [1.5 2.5] ve

<sup>i</sup> İki değer arasında  $\rho = 1 - 1/\varphi$  ilişkisi yazılabilir.

<sup>ii</sup> Golden ratio



hassasiyet olarak da  $\varepsilon = 0.05$  değerlerini dikkate alalım. Başlangıç aralığını ve altın kesit oranını dikkate alarak yeni aralığı;

$$\begin{aligned} a_1 &= a_0 + \rho(b_0 - a_0) \\ a_1 &= 1.5 + 0.382(2.5 - 0) \\ a_1 &= 1.882 \end{aligned}$$

$$\begin{aligned} b_1 &= a_0 + (1 - \rho)(b_0 - a_0) \\ b_1 &= 1.5 + (1 - 0.382)(2.5 - 1.5) \\ b_1 &= 2.118 \end{aligned}$$

şeklinde elde edebiliriz. Bu değerler amaç fonksiyonunda yerine konulduğunda;

$$\begin{aligned} f(1.882) &= -5.84 \\ f(2.118) &= -6.07 \end{aligned}$$

değerleri elde edilir. Bulunan bu değerlerin karşılaştırılması neticesinde;  $f(1.882) > f(2.118)$  eşitsizliği dikkate alınarak yeni aralığın  $[a_1 b_0] = [1.882 2.5]$  olduğu gözlemlenir. Söz konusu aralıkta hassasiyetin  $\varepsilon = 2.5 - 1.882 = 0.618 > 0.05$  olması nedeniyle bir sonraki adıma geçilir. İkinci aralığın alt sınırı olarak  $a_2 = 2.118$  değeri dikkate alınırsa, yeni aralık;

$$\begin{aligned} a_2 &= b_1 \\ a_2 &= 2.118 \end{aligned}$$

$$\begin{aligned} b_2 &= a_1 + (1 - \rho)(b_0 - a_1) \\ b_2 &= 2.2639 \end{aligned}$$

biçiminde bulunur. Bu değerler amaç fonksiyonunda dikkate alındığında;

$$\begin{aligned} f(2.118) &= -6.074 \\ f(2.2639) &= -6.0366 \end{aligned}$$

değerleri elde edilir. Bulunan bu değerlerin karşılaştırılması neticesinde;  $f(2.118) < f(2.2639)$ , yeni aralığın  $[a_1 b_2] = [1.882 2.2639]$  olduğu görülür. Söz konusu aralıkta hassasiyetin  $\varepsilon = 2.2639 - 1.882 = 0.3819 > 0.05$  olması nedeniyle bir sonraki adıma geçilir. İkinci aralığın üst sınırı olarak  $b_3 = 2.118$  değeri dikkate alınırsa, yeni aralık;

$$\begin{aligned} a_3 &= a_1 + \rho(b_2 - a_1) \\ a_3 &= 1.882 + 0.382(2.2639 - 1.882) \\ a_3 &= 2.0279 \end{aligned}$$

$$\begin{aligned} b_3 &= a_2 \\ b_3 &= 2.118 \end{aligned}$$

olarak belirlenir. Bu değerler amaç fonksiyonunda dikkate alındığında;

$$\begin{aligned} f(2.0279) &= -6.0255 \\ f(2.118) &= -6.074 \end{aligned}$$





değerleri elde edilir. Bulunan bu değerlerin karşılaştırılması neticesinde;  $f(2.0279) > f(2.118)$ , yeni aralığın  $[a_3 \ b_2] = [2.0279 \ 2.2639]$  olduğu görülür. Söz konusu aralıkta hassasiyetin  $\varepsilon = 2.2639 - 2.0279 = 0.236 > 0.05$  olması nedeniyle bir sonraki adıma geçilir. İkinci aralığın alt sınırı olarak  $a_4 = 2.118$  değeri dikkate alınır, yeni aralık;

$$\begin{aligned} a_4 &= b_3 \\ a_4 &= 2.118 \end{aligned}$$

$$\begin{aligned} b_4 &= a_3 + (1 - \rho)(b_2 - a_3) \\ b_4 &= 2.0279 + (1 - 0.382)(2.2639 - 2.0279) \\ b_4 &= 2.1738 \end{aligned}$$

biçiminde bulunur. Bu değerler amaç fonksiyonunda dikkate alındığında;

$$\begin{aligned} f(2.118) &= -6.074 \\ f(2.1738) &= -6.0779 \end{aligned}$$

değerleri elde edilir. Bulunan bu değerlerin karşılaştırılması neticesinde;  $f(2.118) > f(2.1738)$ , yeni aralığın  $[a_4 \ b_2] = [2.118 \ 2.2639]$  olduğu görülür. Söz konusu aralıkta hassasiyetin  $\varepsilon = 2.2639 - 2.118 = 0.1459 > 0.05$  olması nedeniyle bir sonraki adıma geçilir. Müteakip çizelgede, sonraki adımlara ait sınır değerleri ve istenen hassasiyeti veren aralık değeri görülmektedir.

$a_i$	$b_i$	$a_{i+1}$	$b_{i+1}$	$\varepsilon$
2.1180	2.2082	2.1525	2.1738	0.0902
2.1180	2.1738	2.1393	2.1525	0.0557
2.1393	2.1738	2.1525	2.1606	0.0344

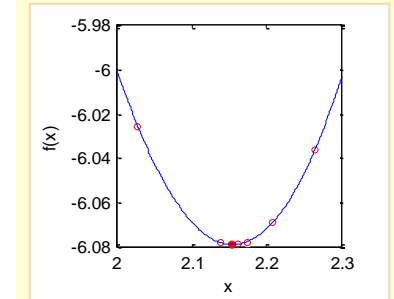
Buna göre  $x = 2.1525$  değeri  $f(x) = -6.079$  amaç fonksiyonu değeri ile istenen hassasiyette sonuç vermektedir. Bu sonuca ulaşmak için başlangıç aralığı dâhil yedi adım atılmış olup Şekil 2.55 dâhilinde analitik çizim ve her adımda ulaşılan asgari nokta (o) görülmektedir.



Şekil 2.56'daki gibi bir yamağın alanını doruklaştıran açı değerini, altın kesit eniyileme yöntemini kullanarak belirleyelim. Geometrik bağıntılar kullanarak yamağın alanını;

$$f(\theta) = a^2 \sin \theta (1 + \cos \theta)$$

denklemleri ile tanımlamak mümkündür.



Şekil 2.55 Altın kesit yöntemi ile azami nokta taraması.



Örnek uygulama



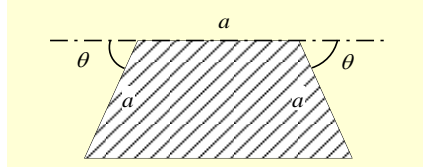




Sorunun somutlaştırılması adına  $a = 2$  değeri dikkate alınır ve minimizasyon formuna dönüşüm sağlanırsa aynı sorun;

$$\min f(\theta) = \frac{1}{4 \sin \theta (1 + \cos \theta)}$$

şeklinde ifade edilebilir.



**Şekil 2.56** Açıya bağlı olarak yamuğun azami alanının tespiti sorunu.

Problemin altın kesit yöntemi ile çözümü sadedinde yapılacak ilk işlem, başlangıç aralığının ve çözüm hassasiyetinin tespitidir. Başlangıç aralığı olarak  $[\pi/4 \ \pi/2]$  ve hassasiyet olarak da  $\varepsilon = 0.05$  değerlerini dikkate alalım. Tanımlanan aralığı ve altın kesit oranını kullanarak yeni aralığı;

$$\begin{aligned} a_1 &= a_0 + \rho(b_0 - a_0) \\ a_1 &= 0.7854 + 0.382(1.5708 - 0.7854) \\ a_1 &= 1.0854 \end{aligned}$$

$$\begin{aligned} b_1 &= a_0 + (1 - \rho)(b_0 - a_0) \\ b_1 &= 0.7854(1 - 0.382)(1.5708 - 0.7854) \\ b_1 &= 1.2708 \end{aligned}$$

şeklinde elde edebiliriz. Bu değerler amaç fonksiyonunda yerine konulduğunda;

$$\begin{aligned} f(1.0854) &= 0.1927 \\ f(1.2708) &= 0.2020 \end{aligned}$$

değerleri elde edilir. Bulunan bu değerlerin karşılaştırılması neticesinde;  $f(1.0854) < f(1.2708)$ , yeni aralığın  $[a_0 \ b_1] = [0.7854 \ 1.2708]$  olduğu gözlemlenir. Söz konusu aralıkta hassasiyetin  $\varepsilon = 1.2708 - 0.7854 = 0.4854 > 0.05$  olması nedeniyle bir sonraki adıma geçilir. İkinci aralığın üst sınırı olarak  $b_2 = 1.0854$  değeri dikkate alınır, yeni aralık;

$$\begin{aligned} a_2 &= a_0 + \rho(b_1 - a_0) \\ a_2 &= 0.7854 + 0.382(1.2708 - 0.7854) \\ a_2 &= 0.9708 \end{aligned}$$

$$\begin{aligned} b_2 &= a_1 \\ b_2 &= 1.0854 \end{aligned}$$

biçiminde bulunur. Bu değerler amaç fonksiyonunda dikkate alındığında;

$$\begin{aligned} f(0.9708) &= 0.1936 \\ f(1.0854) &= 0.1927 \end{aligned}$$





değerleri elde edilir. Bulunan bu değerlerin karşılaştırılması neticesinde;  $f(0.9708) > f(1.0854)$ , yeni aralığın  $[a_2 \ b_1] = [0.9708 \ 1.2708]$  olduğu görülür. Bahse konu aralıkta hassasiyetin  $\varepsilon = 1.2708 - 0.9708 = 0.3 > 0.05$  olması nedeniyle bir sonraki adıma geçilir. İkinci aralığın alt sınırı olarak  $a_3 = 1.0854$  değeri dikkate alınrsa, yeni aralık;

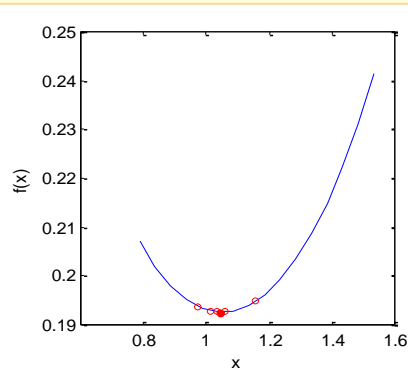
$$a_3 = b_2 \\ a_3 = 1.0854$$

$$b_3 = a_2 + (1 - \rho)(b_1 - a_2) \\ b_3 = 0.9708 + (1 - 0.382)(1.2708 - 0.9708) \\ b_3 = 1.1562$$

olarak belirlenir. Müteakip çizelgede, sonraki adımlara ait sınır değerleri ve istenen hassasiyeti veren aralık değeri görülmektedir;

$a_i$	$b_i$	$a_{i+1}$	$b_{i+1}$	$\varepsilon$
0.9708	1.1562	1.0416	1.0854	0.1854
0.9708	1.0854	1.0146	1.0416	0.1146
1.0146	1.0854	1.0416	1.0583	0.0708
1.0146	1.0583	1.0313	1.0416	0.0438

Buna göre  $x = 1.0416$  radyan değeri,  $f(x) = 0.1924$  amaç fonksiyonu değeri ile istenen hassasiyette sonuç vermektedir. Bu sonuca ulaşmak için başlangıç aralığı dâhil yedi adım atılmış olup takip eden grafikte analitik çizim ve her adımda ulaşılan asgari nokta (o) görülmektedir. Problemin orijinal hali dikkate alındığında, açışal olarak yaklaşık  $60^\circ$  ve alan değeri olarak da  $5.19 \text{ br}^2$  sonucu elde edilir.



Şekil 2.57 Altın kesit yöntemi ile taranan aralıklar ve ulaşılan optimal nokta.



$$\min f(\theta) = 1/[4 \sin \theta (1 + \cos \theta)]$$

```

clc; clear
a = pi/4; b = pi/2; epsilon = 0.05; iter = 50;
ro = double(3-sqrt(5))/2; k = 1;
x1 = a+ro*(b-a) ;
x2 = a+(1-ro)*(b-a);
data(k,:) = [a b x1 x2 b-a];
f_x1 = 1/(4*sin(x1)*(1+cos(x1)));
f_x2 = 1/(4*sin(x2)*(1+cos(x2)));

while ((abs(b-a)>epsilon) && (k<iter))
    k=k+1;
    if f_x1<f_x2
        b = x2; x2 = x1; x1 = a+ro*(b-a);
        f_x1 = 1/(4*sin(x1)*(1+cos(x1)));
        f_x2 = 1/(4*sin(x2)*(1+cos(x2)));
    else
        a = x1; x1 = x2; x2 = a+(1-ro)*(b-a);
        f_x1=1/(4*sin(x1)*(1+cos(x1)));
        f_x2=1/(4*sin(x2)*(1+cos(x2)));
    end
    data(k,:) = [a b x1 x2 b-a];
end
data

```

data =

0.7854	1.5708	1.0854	1.2708	0.7854
0.7854	1.2708	0.9708	1.0854	0.4854
0.9708	1.2708	1.0854	1.1562	0.3000
0.9708	1.1562	1.0416	1.0854	0.1854
0.9708	1.0854	1.0146	1.0416	0.1146
1.0146	1.0854	1.0416	1.0583	0.0708
1.0146	1.0583	1.0313	1.0416	0.0438

**MATLAB uygulaması**

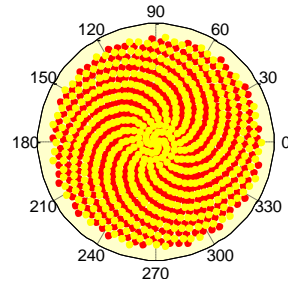


Altın oran ile ayçiçeği <sup>(38)</sup>

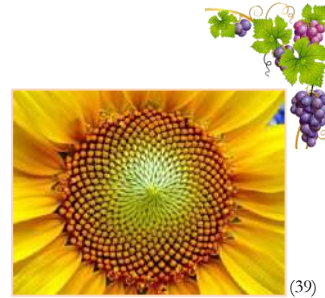
```

phi = (sqrt(5)-1)/2;
n = 1000;
rho = (0:n-1).^phi;
theta = (0:n-1)*2*pi*phi;
theta1 = theta(1,1:2:n); rho1 =
rho(1,1:2:n);
theta2 = theta(1,2:2:n); rho2 =
rho(1,2:2:n);
polar(theta1, rho1, 'r. '); hold
on;
polar(theta2, rho2, 'y. ');

```



**MATLAB uygulaması**



(39)





### Örnek uygulama

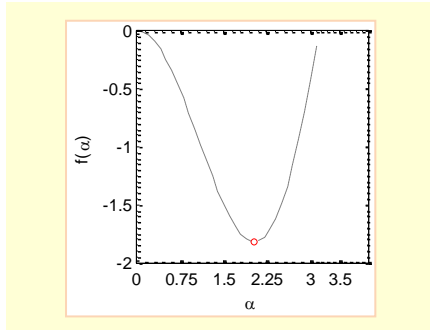
$f(\mathbf{x}) = \sin x_1 (1 - x_2)$  fonksiyonu için başlangıç noktası olarak  $\mathbf{x}^T = [0 \ 1]$  ve yön olarak da  $\mathbf{d}^T = [1 \ 1]$  koordinatlarını dikkate alalım. Buna göre yeni nokta;

$$\mathbf{x}_{(1)} = \mathbf{x}_{(0)} + \alpha \mathbf{d}_{(0)} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} + \alpha \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha \\ 1 + \alpha \end{bmatrix}$$

olarak belirlenebilir. Bu yöneyin amaç fonksiyonunda yerine konulması halinde;

$$f(\mathbf{x}_{(1)}) = -\alpha \sin \alpha$$

denklemi yazılabilir. Amaç fonksiyonunun aşgari değerinin eldesi için gereken  $\alpha$  değerinin bulunması sadedinde altın kesit yöntemi uygulanabilir. Buna göre başlangıç aralığı olarak  $[1.5 \ 2.5]$  ve hassasiyet olarak  $\varepsilon = 0.05$  değerlerini dikkate alalım. Tanımlı hassasiyet ve sınır değerlerine göre işlem adımları takip edildiğinde müteakip sonuçlar elde edilebilir;



data =

1.5000	2.5000	1.8820	2.1180	1.0000
1.8820	2.5000	2.1180	2.2639	0.6180
1.8820	2.2639	2.0279	2.1180	0.3820
1.8820	2.1180	1.9721	2.0279	0.2361
1.9721	2.1180	2.0279	2.0623	0.1459
1.9721	2.0623	2.0066	2.0279	0.0902
2.0066	2.0623	2.0279	2.0410	0.0557
2.0066	2.0410	2.0197	2.0279	0.0344

Şekil 2.58 Altın kesit yöntemi ile ulaşılan optimal nokta ve aralık değerleri.

Bu sonuca göre algoritma sekiz adım sonunda  $\alpha = 2.0279$  ve  $f(\alpha) = -1.8197$  amaç fonksiyonu değeri ile istenen hassasiyette sonuç vermektedir. Grafikte analitik çizim ve altın kesit yöntemi ile ulaşılan aşgari nokta (o) görülmektedir. Öte yandan elde edilen optimal  $\alpha$  değerinin  $\mathbf{x}_{(1)}$  noktasının tayininde dikkate alınması halinde yeni nokta;

$$\mathbf{x}_{(1)} = \begin{bmatrix} \alpha \\ 1 + \alpha \end{bmatrix} = \begin{bmatrix} 2.0279 \\ 3.0279 \end{bmatrix}$$

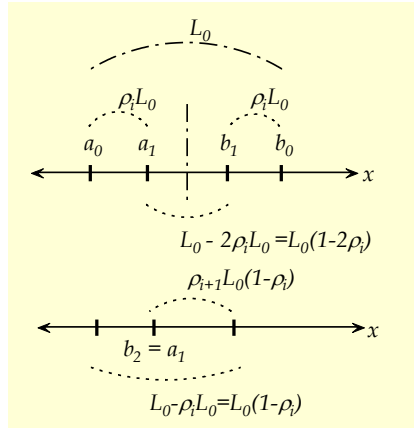
olarak elde edilir.



## (2) “Fibonacci” yöntemi

Altın kesit yöntemine göre daha etkin bir diğer eleme bazlı yöntem, “Fibonacci” eniyileme yöntemidir. Diğerinde olduğu gibi yöntemin ilk uygulamasını J. Kiefer (1924-1981)<sup>(38)</sup> tarafından kaleme alınan “Azami değer için silsileli asgari-azami taraması” adlı makalede görmekteyiz. “Fibonacci” eniyileme yönteminin çıkış kaynağı, her adımda aralığın sabit bir oran ile belirlenmesi yerine minimizasyonu daha etkin kılacak şekilde değişken bir oran ile elde edilmesi gayesidir. Hatırlanacağı üzere altın kesit yönteminde, başlangıçta dikkate alınan  $[a_0 \ b_0]$  aralığı simetrik ve eşit bölme yaklaşımı ile üç aralığa bölünmekteydi. Bir sonraki aşamada ise yeni aralık tayini ve eldeki noktalardan birisinin bölmelemede dikkate alınması hedeflenmişti. Müteakip şekilden de görüleceği üzere başlangıç aralığı mesafesi  $L_0$  olmak üzere değişken nitelikteki  $\rho_i$  oranının dikkate alınması halinde;

$$L_0(1 - 2\rho_i) = \rho_{i+1}L_0(1 - \rho_i)$$



denklemi kolayca yazılabilir. Bu ifadede  $\rho_{i+1}$  terimi yalnız bırakılırsa;

$$\rho_{i+1} = 1 - \frac{\rho_i}{1 - \rho_i} \quad (2.60)$$

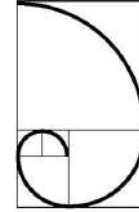
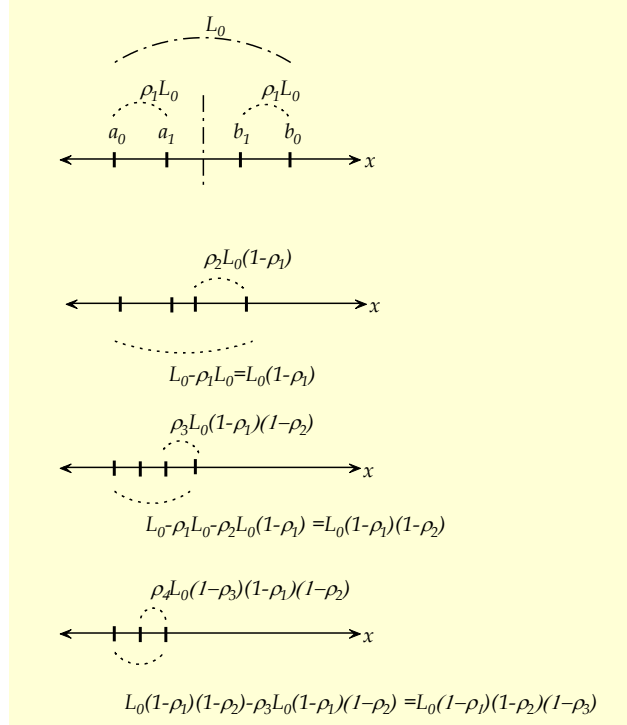
elde edilir.

<sup>i</sup> “Sequential minimax search for a maximum”



Diğer taraftan her aşamada dikkate alınan  $\rho_i$  oranının  $\rho_i < 1/2$  şartını sağlaması gerektiğini de hatırlatalım. Öte yandan aşağıdaki şekilden de görüleceği üzere her yeni adımda aralığın;

$$L_0(1 - \rho_1)(1 - \rho_2)(1 - \rho_3) \dots (1 - \rho_i)$$



(43)



(44)



**Şekil 2.59** Doğada Fibonacci sayılarına göre şekillenen bazı düzenlemeler.

değerine indirgeneceği söylenebilir. Süreçteki ana gaye, atılan  $N$  adıma karşılık söz konusu aralığın minimize edilmesidir. Dolayısıyla her adımda dikkate alınacak oranın cevabı;

$$\min_{\rho} f(\rho) = (1 - \rho_1)(1 - \rho_2)(1 - \rho_3) \dots (1 - \rho_N) \quad (2.61)$$

Kısıt

$$\rho_{i+1} = 1 - \frac{\rho_i}{1 - \rho_i}, \quad i = 1, 2, \dots, N - 1$$

$$0 < \rho_i < 1/2, \quad i = 1, 2, \dots, N$$

şeklinde tanımlanan doğrusal olmayan eniyileme probleminin çözümüdür. Detayları belirtilen referansta<sup>(40)</sup> bulunabilecek söz konusu sorunun çözümlerinden birisi de Fibonacci serisidir.





Fibonacci serisine ait sayıların geçmişi oldukça eskilere, M.Ö. 200 yıllarına kadar gider. Kayıtlara göre eski Hint matematikçisi Pingala'nın yazıtlarında söz konusu seriden bahsedilmektedir<sup>(41)</sup>. Bununla beraber serinin Batı dünyasına tanıtımı, İtalyan matematikçi L. Bonacci (1170-1250) veya nam-ı diğer Fibonacci tarafından gerçekleştirildi<sup>(42)</sup>. Fibonacci sayılarının diğer bir özelliği, doğadaki bazı yapılarda rastlanılmasıdır. Fibonacci serisini aşağıdaki gibi tanımlamak mümkündür;

$$F_{i+1} = F_i + F_{i-1}, \quad i = 1, 2, \dots \quad (2.62)$$

Buna göre bir sonraki sayı, bir önceki ardıl iki sayının toplamına eşittir. Genellikle  $F_0 = F_1 = 1$  olarak dikkate alınır ve serinin devam eden sayıları ifade edilen denkleme göre belirlenir. Aşağıdaki çizelgede ilk on Fibonacci sayısı listelenmiştir;

$F_1$	$F_2$	$F_3$	$F_4$	$F_5$	$F_6$	$F_7$	$F_8$	$F_9$	$F_{10}$
1	2	3	5	8	13	21	34	55	89

Adım sayısı  $N$  olmak üzere Fibonacci sayısı ile her adımda dikkate alınacak küçültme oranı arasındaki ilişkiyi;

$$\rho_i = 1 - \frac{F_{N-i+1}}{F_{N-i+2}} \quad (2.63)$$

denklemini ifade etmek mümkündür.



$$\rho_i = 1 - \frac{F_{N-i+1}}{F_{N-i+2}}$$



Detay açıklama

ilişkisinin, yöntemin çıkış noktası olan eniyileme sorunundaki kısıtlar ile örtüştüğünü, basit bazı düzenlemeler ile göstermek mümkündür. Hatırlanacağı üzere kısıtlardan ilki;

$$\rho_{i+1} = 1 - \frac{\rho_i}{1 - \rho_i}$$

şeklindeydi. İfadede geçen  $\rho_{i+1}$  oranını;





$$\rho_{i+1} = 1 - \frac{F_{N-(i+1)+1}}{F_{N-(i+1)+2}} = 1 - \frac{F_{N-i}}{F_{N-i+1}}$$

denklemini kullanarak Fibonacci sayılarıyla ilişkilendirmek mümkündür. Oran ilişkisi ile Fibonacci sayıları arasındaki ilişkinin beraber dikkate alınması halinde;

$$1 - \frac{F_{N-i}}{F_{N-i+1}} = 1 - \frac{1 - \frac{F_{N-i+1}}{F_{N-i+2}}}{1 - \left(1 - \frac{F_{N-i+1}}{F_{N-i+2}}\right)}$$

eşitliğine ulaşılır. Bu eşitlikte uygun sadeleştirmeler yapıldığında;

$$\frac{F_{N-i}}{F_{N-i+1}} = \frac{1 - \frac{F_{N-i+1}}{F_{N-i+2}}}{\frac{F_{N-i+1}}{F_{N-i+2}}} = \left(\frac{F_{N-i+2} - F_{N-i+1}}{F_{N-i+1}}\right)$$

$$F_{N-i} = F_{N-i+2} - F_{N-i+1}$$

elde edilir. Görüldüğü üzere bu ifade,  $k = N - i + 1$  dönüşümü dikkate alındığında;

$$F_{k+1} = F_k + F_{k-1}$$

şeklindeki Fibonacci serisinin tanımını vermektedir.



Küçültme oranı ile yöntemin  $N$  adım sonrasında sağlayacağı  $\Delta$  aralık değeri beraber dikkate alındığında;

$$L_0(1 - \rho_1)(1 - \rho_2)(1 - \rho_3) \dots (1 - \rho_N) \quad (2.64)$$

$$\Delta = L_0 \left[1 - \left(1 - \frac{F_N}{F_{N+1}}\right)\right] \left[1 - \left(1 - \frac{F_{N-1}}{F_N}\right)\right] \left[1 - \left(1 - \frac{F_{N-2}}{F_{N-1}}\right)\right] \dots \left[1 - \left(1 - \frac{F_1}{F_2}\right)\right] = \frac{L_0}{F_{N+1}} \quad (2.65)$$

ifadesine ulaşılır. Ancak Fibonacci sayılarının tanımı gereği, aralığın son adımında  $\rho_N = 1 - (1/2) = 1/2$  oranında küçültülmesi hususu, başlangıçta dikkate alınan  $0 < \rho_i < 1/2$  şartını





sağlamamaktadır. Bu nedenle son adımda  $\varepsilon$  gibi bir oran için geçerli olan belirli bir hassasiyet tanımı yapılarak son aralıkta küçültme oranı;

$$1 - \rho_N \rightarrow 1 - (\rho_N - \varepsilon) = \frac{1}{2} + \varepsilon = \frac{1 + 2\varepsilon}{2} \quad (2.66)$$

değeri ile dikkate alınır, son aralık değeri;

$$\Delta = L_0 \frac{F_N}{F_{N+1}} \frac{F_{N-1}}{F_N} \frac{F_{N-2}}{F_{N-1}} \dots \frac{1 + 2\varepsilon}{2} = L_0 \frac{1 + 2\varepsilon}{F_{N+1}} \quad (2.67)$$

şeklinde elde edilmiş olur. Daraltma mertebesi, eniyileme sorunlarında istenen aralık hassasiyeti değerine göre atılması gereken adım sayısını belirlemekte kullanılabilir. Zira başlangıçtaki aralık değeri ( $L_0$ ) ile son aralık değeri ( $\Delta$ ) arasındaki oranı;

$$\frac{\Delta}{L_0} = \frac{1 + 2\varepsilon}{F_{N+1}} \quad (2.68)$$

şeklinde ifade etmek mümkündür. Yöntemin uygulamadaki işlem adımlarını daha somut bir şekilde gözlemlemek için daha önce altın kesit yöntemi ile çözülen aynı sorunu Fibonacci yöntemi ile çözelim. Hatırlanacağı üzere sorun, bir yamağın alanını doruklaştıran açı değerinin bulunması şeklindeydi. Buna göre amaç fonksiyonu;

$$\min f(\theta) = \frac{1}{4 \sin \theta (1 + \cos \theta)}$$

şeklinde ifade edilmekteydi. Sorunun Fibonacci yöntemi ile çözümü bağlamında yapılacak ilk işlem, tıpkı altın kesit yönteminde olduğu gibi başlangıç aralığının ve son adımda ulaşılması istenen aralık değerinin, diğer bir ifade ile çözüm hassasiyetinin tespitidir. Ayrıca, oran için geçerli olan hassasiyet de belirlenmelidir. Buna göre başlangıç aralığı olarak  $[\pi/4 \ \pi/2]$  aralığını, çözüm hassasiyetini  $\Delta = 0.04$  ve oran hassasiyetini de  $\varepsilon = 0.05$  olarak dikkate alalım. İstlenen çözüm aralığı değerinin sağlanabilmesi için atılması gereken adım sayısı değerini;

$$F_{N+1} \leq \frac{1 + 2\varepsilon}{\frac{\Delta}{L_0}} = \frac{1 + 2 \times 0.05}{\frac{0.04}{\left(\frac{\pi}{2} - \frac{\pi}{4}\right)}} = 21.59$$



eşitsizliğini dikkate alarak  $N = 6$  şeklinde belirlemek mümkündür. Zira  $F_7 = 21$  olup belirtilen eşitsizliği sağlamaktadır. Başlangıçta tanımlanan aralığı ve Fibonacci sayısı bazlı kesit oranını kullanarak yeni aralığı;

$a_1 = a_0 + \rho_1(b_0 - a_0)$ $a_1 = b_0 - \frac{F_6}{F_7}(b_0 - a_0)$ $a_1 = 1.5708 - \frac{13}{21}(1.5708 - 0.7854)$ $a_1 = 1.0846$	$b_1 = a_0 + (1 - \rho_1)(b_0 - a_0)$ $b_1 = a_0 + \frac{F_6}{F_7}(b_0 - a_0)$ $b_1 = 0.7854 + \frac{13}{21}(1.5708 - 0.7854)$ $b_1 = 1.2716$
---	---

şeklinde elde edebiliriz. Bu değerler amaç fonksiyonunda yerine konulduğunda;

$$f(1.0846) = 0.1927$$

$$f(1.2716) = 0.2021$$

değerleri elde edilir. Bulunan bu değerlerin karşılaştırılması neticesinde;  $f(1.0846) < f(1.2716)$  olduğundan yeni aralığın  $[a_0 \ b_1] = [0.7854 \ 1.2716]$  olacağı gözlemlenir. Söz konusu aralıkta hassasiyetin  $\Delta = 1.2716 - 0.7854 = 0.4862 > 0.04$  olması nedeniyle bir sonraki adıma geçilir. İkinci aralığın üst sınırı olarak  $b_2 = 1.0846$  değeri dikkate alınır, yeni aralık;

$a_2 = b_1 - \frac{F_5}{F_6}(b_1 - a_0)$ $a_2 = 1.2716 - \frac{8}{13}(1.2716 - 0.7854)$ $a_2 = 0.9724$	$b_2 = a_1$ $b_2 = 1.0846$
--	----------------------------

biçiminde bulunur. Bu değerler amaç fonksiyonunda dikkate alındığında;

$$f(0.9724) = 0.1935$$

$$f(1.0846) = 0.1927$$

değerleri elde edilir. Bulunan bu değerlerin karşılaştırılması neticesinde;  $f(0.9724) > f(1.0846)$  olduğundan yeni aralığın  $[a_2 \ b_1] = [0.9724 \ 1.2716]$  olacağı görülür. Bahse konu aralıktaki hassasiyetin  $\Delta = 1.2716 - 0.9724 = 0.2992 > 0.04$  olması nedeniyle bir sonraki adıma geçilir. Üçüncü aralığın alt sınırı olarak  $a_3 = 1.0846$  değeri dikkate alınır, yeni aralık;



$$a_3 = b_2$$

$$a_3 = 1.0846$$

$$b_3 = a_2 + \frac{F_4}{F_5}(b_1 - a_2)$$

$$b_3 = 0.9724 + \frac{5}{8}(1.2716 - 0.9724)$$

$$b_3 = 1.1594$$

olarak belirlenir. Bu değerler amaç fonksiyonunda yerine konulduğunda;

$$f(1.0846) = 0.1927, \quad f(1.1594) = 0.1948$$

değerleri elde edilir. Bulunan bu değerlerin karşılaştırılması neticesinde;  $f(1.0846) < f(1.1594)$  olduğundan yeni aralığın  $[a_2 \ b_3] = [0.9724 \ 1.1594]$  şeklinde olduğu gözlemlenir. Söz konusu aralıkta hassasiyetin  $\Delta = 1.1594 - 0.9724 = 0.1870 > 0.04$  olması nedeniyle bir sonraki adıma geçilir. Dördüncü aralığın üst sınırı olarak  $b_4 = 1.0846$  değeri dikkate alınırsa, yeni aralık;

$$a_4 = a_2 - \frac{F_3}{F_4}(b_3 - a_2)$$

$$a_4 = 0.9724 - \frac{3}{5}(1.1594 - 0.9724)$$

$$a_4 = 1.0472$$

$$b_4 = a_3$$

$$b_4 = 1.0846$$

biçiminde bulunur. Söz konusu değerler amaç fonksiyonunda yerine konulduğunda;

$$f(1.0472) = 0.1925$$

$$f(1.0846) = 0.1927$$

değerleri elde edilir. Bu değerlerin karşılaştırılması neticesinde;  $f(1.0472) < f(1.0846)$  olduğundan yeni aralığın  $[a_2 \ b_4] = [0.9724 \ 1.0846]$  biçiminde olduğu gözlemlenir. Bahse konu aralıkta hassasiyetin  $\Delta = 1.0846 - 0.9724 = 0.1122 > 0.04$  olması nedeniyle bir sonraki adıma geçilir. Beşinci aralığın üst sınırı olarak  $b_5 = 1.0472$  değeri dikkate alınırsa, yeni aralık;

$$a_5 = a_2 - \frac{F_2}{F_3}(b_4 - a_2)$$

$$a_5 = 0.9724 - \frac{2}{3}(1.0846 - 0.9724)$$

$$a_5 = 1.0098$$

$$b_5 = a_4$$

$$b_5 = 1.0472$$

olarak hesaplanır. Bu değerler amaç fonksiyonunda yerine konulduğunda;





$$f(1.0098) = 0.1927$$

$$f(1.0472) = 0.1925$$

değerleri elde edilir. Bulunan bu değerlerin karşılaştırılması neticesinde;  $f(1.0098) > f(1.0472)$  olduğundan yeni aralığın  $[a_5 \ b_4] = [1.0098 \ 1.0846]$  olduğu görülür. Bahse konu aralıktaki hassasiyetin  $\Delta = 1.0846 - 1.0098 = 0.0748 > 0.04$  olması nedeniyle bir sonraki adıma geçilir. Altıncı aralığın alt sınırı olarak  $a_6 = 1.0472$  değeri dikkate alınabilir. Ancak sonraki adımda Fibonacci sayıları  $F_1/F_2 = 1/2$  değerlerini almaktadır. Dolayısıyla yeni aralığın hesaplanmasında oran hassasiyeti değerinin de hesaba katılması gerekir. Bu nedenle yeni aralık değerleri;

$$a_6 = b_5$$

$$a_6 = 1.0472$$

$$b_6 = a_5 + \left(\frac{F_1}{F_2} + \varepsilon\right)(b_4 - a_5)$$

$$b_6 = 1.0098 + \left(\frac{1}{2} + 0.05\right)(1.0098 - 1.0846)$$

$$b_6 = 1.0435$$

olarak belirlenir. Daha önceki adımlarda olduğu üzere değerler amaç fonksiyonunda yerine konulduğunda;

$$f(1.0472) = 0.1924500$$

$$f(1.0435) = 0.1924527$$

değerleri elde edilir. Bu değerlerin karşılaştırılması neticesinde;  $f(1.0472) < f(1.0435)$  olduğundan yeni aralığın  $[a_5 \ b_6] = [1.0098 \ 1.0435]$  olduğu gözlemlenir. Bahse konu aralıkta hassasiyetin  $\Delta = 1.0435 - 1.0098 = 0.0337 < 0.04$  olması nedeniyle tarama süreci durdurulur. Buna göre  $x = 1.0472$  radyan değeri,  $f(x) = 0.19245$  amaç fonksiyonu değeri ile istenen hassasiyette sonuç vermektedir. Bu sonuca ulaşmak için başlangıç aralığı dâhil altı adım atılmış olup açılmal olarak yaklaşık  $60^\circ$  ve alan değeri olarak da  $5.19 \text{ br}^2$  sonucu elde edilir. Dikkat çeken diğer bir husus ise altın kesit yöntemine göre daha az adım ile sonuca ulaşılmasıdır.

$$\min f(\theta) = 1/[4 \sin \theta (1 + \cos \theta)]$$



**MATLAB uygulaması**

```
clc; clear; b = pi/2; a = pi/4; tolerance = 0.05; distance = 0.04; n = 1;
while fibonacci(n+1) <= (1+2*tolerance)/(distance/(b-a));
    n = n+1;
end
p = b - (fibonacci(n-1)/fibonacci(n))*(b-a);
q = a + (fibonacci(n-1)/fibonacci(n))*(b-a);
fp = 1/(4*sin(p)*(1+cos(p))); fq = 1/(4*sin(q)*(1+cos(q)));
data(1,:) = [a b p q (b-a)]; i = 2;
```



```

for k = n-1:-1:2
    if k>=3
        if (fp <= fq)
            b = q; q = p; fq = fp;
            p = b - (fibonacci(k-1)/fibonacci(k))*(b-a);
            fp = 1/(4*sin(p)*(1+cos(p)));
        else
            a = p; p = q; fp = fq;
            q = a + (fibonacci(k-1)/fibonacci(k))*(b-a);
            fq = 1/(4*sin(q)*(1+cos(q)));
        end
    else
        if (fp <= fq)
            b = q; q = p; fq = fp;
            p = b - ((fibonacci(k-1)/fibonacci(k))+tolerance)*(b-a);
            fp = 1/(4*sin(p)*(1+cos(p)));
        else
            a = p; p = q;
            fq = fp; q = a + ((fibonacci(k-1)/fibonacci(k))-tolerance)*(b-a);
            fp = 1/(4*sin(q)*(1+cos(q)));
        end
    end
    data(i,:) = [a b p q (b-a)]; i = i + 1;
end
if (fp <= fq);
    b = q; x_opt = p; f_opt = fp ;
else;
    a = p; x_opt = q; f_opt = fq ;
end
data
opt = [x_opt f_opt (b-a)]
function y = fibonacci(x)
F = [1 2];
for i = 2:x
    F(i+1) = F(i)+F(i-1);
end
y = F(x);

```



```

data =
    0.7854    1.5708    1.0846    1.2716    0.7854
    0.7854    1.2716    0.9724    1.0846    0.4862
    0.9724    1.2716    1.0846    1.1594    0.2992
    0.9724    1.1594    1.0472    1.0846    0.1870
    0.9724    1.0846    1.0098    1.0472    0.1122
    1.0098    1.0846    1.0472    1.0435    0.0748
opt =
    1.0472    0.1925    0.0337

```





### (3) “Quasi-Newton” yöntemi

Eleme bazı yöntemlerin haricinde, yaygın olarak kullanılan diğer yöntem gruplarından belki de en geniş yelpazeye sahip olanı, gradyan esaslı yöntemlerdir. Eleme bazı yöntemlerde, belirli bir aralığın tanımlanması ile çözüm sürecine başlanırken, gradyan esaslı yöntemlerde aralıktan ziyade bir başlangıç noktası ile işlemlere başlanır. Haliyle ilk yöntem grubunda, çözümün belirtilen aralıkta olduğu düşünülürken, gradyan esaslı yöntemlerde en yakın çözüm noktasına ulaşmak hedeflenir. Gradyan esaslı yöntemlerin temelinde fonksiyonun Taylor serisi açılımı yatar. Hatırlanacağı üzere her hangi bir fonksiyonun  $\alpha_{i+1}$  noktasındaki değerini, bu noktaya yakın seçilen bir  $\alpha_i$  noktasını dikkate alarak;

$$f(\alpha_{i+1}) = f(\alpha_i) + f'(\alpha_i)(\alpha_{i+1} - \alpha_i) + \frac{1}{2!}f''(\alpha_i)(\alpha_{i+1} - \alpha_i)^2 + \frac{1}{3!}f'''(\alpha_i)(\alpha_{i+1} - \alpha_i)^3 + \dots + \frac{1}{n!}f^{(n)}(\alpha_i)(\alpha_{i+1} - \alpha_i)^n \quad (2.69)$$

şeklinde seriye açmak mümkündür. Bu açılımının hassasiyet olarak sadece ilk üç teriminin dikkate alınması halinde, ikilenik yaklaşımla;

$$f(\alpha_{i+1}) = f(\alpha_i) + f'(\alpha_i)(\alpha_{i+1} - \alpha_i) + \frac{1}{2!}f''(\alpha_i)(\alpha_{i+1} - \alpha_i)^2 \quad (2.70)$$

denklemine ulaşılabilir. Diğer taraftan  $\alpha_{i+1}$  noktasının uç nokta olması halinde fonksiyonun türevinin sıfıra eşit olacağı düşünüldüğünde;

$$\frac{\partial f(\alpha_{i+1})}{\partial \alpha_{i+1}} = 0 \quad (2.71)$$

$$f'(\alpha_i) + f''(\alpha_i)(\alpha_{i+1} - \alpha_i) = 0 \quad (2.72)$$

eşitliği elde edilir. Bu ifadeye  $\alpha_{i+1}$  yalnız bırakılırsa;

$$\alpha_{i+1} = \alpha_i - \frac{f'(\alpha_i)}{f''(\alpha_i)} \quad (2.73)$$

denklemini yazılabilir. Görüldüğü üzere fonksiyona ait birinci ve ikinci mertebeden türev bilgisinin kullanılması ile adım adım optimal noktaya ulaşmak mümkündür. Fonksiyonun türevlenebilirliği konusunda sorunlar yaşandığında, eldeki denklem daha da sayısal hale getirilebilir. Hatırlanacağı



üzere herhangi bir fonksiyonun birinci ve ikinci mertebeden türevleri,  $\Delta\alpha$  seviyesinde birbirine yakın noktaların seçilmesi ve o noktalardaki fonksiyon değerlerinin dikkate alınması ile ifade edilebilmekteydi;

$$f'(\alpha_i) \cong \frac{f(\alpha_i + \Delta\alpha) - f(\alpha_i - \Delta\alpha)}{2\Delta\alpha} \quad (2.74)$$

$$f''(\alpha_i) \cong \frac{f(\alpha_i + \Delta\alpha) - 2f(\alpha_i) + f(\alpha_i - \Delta\alpha)}{\Delta\alpha^2} \quad (2.75)$$

Bu yaklaşımların bir önceki denklemde dikkate alınması ve uygun sadeleştirmelerin yapılması halinde;

$$\alpha_{i+1} = \alpha_i - \frac{\Delta\alpha[f(\alpha_i + \Delta\alpha) - f(\alpha_i - \Delta\alpha)]}{2[f(\alpha_i + \Delta\alpha) - 2f(\alpha_i) + f(\alpha_i - \Delta\alpha)]} \quad (2.76)$$

ardışık denklem tanımına ulaşılır. Her bir adım sonrası ulaşılan noktanın optimal nokta olup olmadığı, fonksiyonun o noktadaki türevinin sıfır ya da sayısal anlamda  $\varepsilon$  değerindeki hassasiyete ulaşıp ulaşmadığına bakılarak belirlenir;

$$|f'(\alpha_{i+1})| \cong \left| \frac{f(\alpha_{i+1} + \Delta\alpha) - f(\alpha_{i+1} - \Delta\alpha)}{2\Delta\alpha} \right| \leq \varepsilon \quad (2.77)$$

“Quasi-Newton” yöntemi olarak adlandırılan bu yöntemin daha iyi anlaşılması için örnek bir sorunun çözümünü ele alabiliriz;

$$f(\mathbf{x}) = 0.1 (x_1 + x_2)^2 \sin x_2$$

fonksiyonu için başlangıç noktası olarak  $\mathbf{x}^T = [0 \ 0]$  ve yön olarak  $\mathbf{d}^T = [0 \ 1]$  koordinatlarını dikkate alalım. Buna göre yeni nokta;

$$\mathbf{x}_{(1)} = \mathbf{x}_{(0)} + \alpha \mathbf{d}_{(0)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \alpha \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ \alpha \end{bmatrix}$$

olarak belirlenebilir. Bu yöneyin amaç fonksiyonunda yerine konulması halinde;

$$f(\mathbf{x}_{(1)}) = 0.1\alpha^2 \sin \alpha$$

denklemini yazılabilir.





“Quasi-Newton” yöntemi ile çözüme gidilmesi için öncelikle başlangıç noktasının ve bu noktadan başlayarak atılacak adım büyüklüğünün öngörülmesi gerekir. Sorunun çözümü sadedinde başlangıç noktası olarak  $\alpha_1 = 1$  ve adım büyüklüğü olarak  $\Delta\alpha = 0.1$  değerlerini dikkate alalım. Ayrıca işlemlerin sonlandırılması noktasında dikkate alınacak hassasiyet değerini de  $\varepsilon = 0.01$  olarak belirleyelim. Verilen değerlerin bir sonraki nokta tayininde dikkate alınması halinde;

$$\alpha_2 = \alpha_1 - \frac{\Delta\alpha[f(\alpha_1 + \Delta\alpha) - f(\alpha_1 - \Delta\alpha)]}{2[f(\alpha_1 + \Delta\alpha) - 2f(\alpha_1) + f(\alpha_1 - \Delta\alpha)]} = 1 - \frac{0.1(f(1.1) - f(0.9))}{2[f(1.1) - 2f(1) + f(0.9)]}$$

$$\alpha_2 = 1.3261$$

değerine ulaşılır. Ulaşılan bu noktanın, istenen çözüm hassasiyetini sağlayıp sağlamadığı kontrol edildiğinde;

$$|f'(\alpha_{i+1})| \cong \left| \frac{f(\alpha_{i+1} + \Delta\alpha) - f(\alpha_{i+1} - \Delta\alpha)}{2\Delta\alpha} \right| \leq \varepsilon$$

$$|f'(0.3389)| \cong \left| \frac{f(1.3261 + 0.1) - f(1.3261 - 0.1)}{2 \times 0.1} \right| \leq 0.01$$

$$0.3394 > 0.01$$

eşitsizliği tespit edilebilir. Dolayısıyla bir sonraki adıma geçilir;

$$\alpha_3 = \alpha_2 - \frac{\Delta\alpha[f(\alpha_2 + \Delta\alpha) - f(\alpha_2 - \Delta\alpha)]}{2[f(\alpha_2 + \Delta\alpha) - 2f(\alpha_2) + f(\alpha_2 - \Delta\alpha)]} = 1 - \frac{0.1(f(1.3361) - f(1.2261))}{2[f(1.3361) - 2f(1.3261) + f(1.2261)]}$$

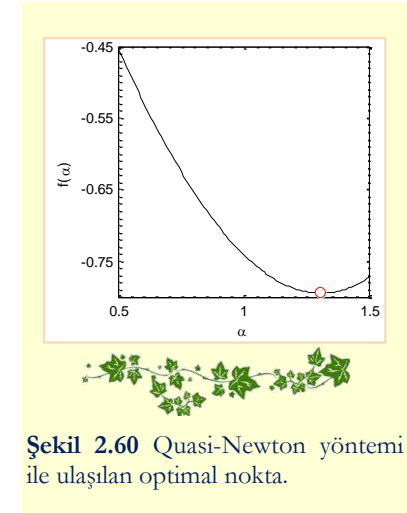
$$\alpha_2 = 0.3061$$

Bu noktadaki çözüm hassasiyeti ise;

$$|f'(0.6101)| \cong \left| \frac{f(0.4061) - f(0.2061)}{2 \times 0.1} \right| \leq 0.01$$

$$6.57 \times 10^{-5} \leq 0.01$$

değerinde olup istenen çözüm hassasiyetine ulaşılması nedeniyle, optimal noktanın  $\alpha = 1.3061$  ve bu noktadaki amaç fonksiyonunun  $f(1.3061) = -0.7946$  değerine sahip olduğu görülür. Şekil 2.60 dâhilinde sorunun analitik çizimi ve elde edilen çözüm noktası görülmektedir.



Şekil 2.60 Quasi-Newton yöntemi ile ulaşılan optimal nokta.





Öte yandan elde edilen optimal  $\alpha$  değerinin,  $\mathbf{x}_{(1)}$  noktasının belirlenmesinde dikkate alınması halinde yeni nokta;

$$\mathbf{x}_{(1)} = \begin{bmatrix} 0 \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ 1.3061 \end{bmatrix}$$

olarak elde edilir.

$$f(\alpha) = 0.1\alpha^2 \sin \alpha$$

**MATLAB uygulaması**



```

clc; clear;
a(1) = 1; da = 0.1;
eps = 0.01;
f0 = 0.1*a(1).^2-sin(a(1));
f10 = 0.1*(a(1)-da).^2-sin(a(1)-da);
f01 = 0.1*(a(1)+da).^2-sin(a(1)+da);
a(2) = a(1)-(da*(f01-f10))/(2*(f01-2*f0+f10)); i = 2;
df = abs(((0.1*(a(1)+da).^2-sin(a(1)+da))-(0.1*(a(1)-da).^2-sin(a(1)-da)))/(2*da));
data(1,:) = [a(2) df];

while (df>eps)
    f0 = 0.1*a(i).^2-sin(a(i));
    f10 = 0.1*(a(i)-da).^2-sin(a(i)-da);
    f01 = 0.1*(a(i)+da).^2-sin(a(i)+da);
    a(i+1) = a(i)-(da*(f01-f10))/(2*(f01-2*f0+f10)) ;
    df = abs(((0.1*(a(i+1)+da).^2-sin(a(i+1)+da))-(0.1*(a(i+1)-da).^2-sin(a(i+1)-da)))/(2*da));
    data(i,:) = [a(i+1) df] ; i = i+1 ;
end

data

```

Iteration	a	df
1	1.3261	0.3394
2	1.3061	0.0001

“Quasi-Newton” yönteminin etkinliği, fonksiyonun ikinci mertebeden türevine bağlı olarak değişir. Taramanın yapıldığı bölgede;

$$f''(\alpha_i) \geq 0$$



şartının sağlandığı durumlarda yöntem, yerel minimumum bulunmasında etkin olarak kullanılabilir. Ancak taramanın yapıldığı bölgeye yakın bir yerde, azami veya durağan nokta bulunuyorsa bu takdirde yöntem, azami veya durağan noktaya doğru çözüm arayışına yönelir ve gerçek çözümden uzaklaşılır. Bununla beraber yöntemin iyi yönlerinden birisi de sadece fonksiyon değerlerine bağlı olarak ilerleme sağlanmasıdır. Zira bu sayede amaç fonksiyonunun birinci veya ikinci mertebeden türev bilgisine gerek duyulmadan adım büyüklüğü taraması yapılabilmektedir. Müteakip örnekte minimizasyon yerine doruklaştırma sürecine yönelik bahse konu sorun görülmektedir.



$$f(\mathbf{x}) = \sin x_1 \cos x_2 (x_1 - x_2)$$



Örnek uygulama

fonksiyonu için başlangıç noktası olarak  $\mathbf{x}^T = [1 \ 1]$  ve yön olarak  $\mathbf{d}^T = [1 \ 0.5]$  koordinatlarını dikkate alalım. Buna göre yeni nokta;

$$\mathbf{x}_{(1)} = \mathbf{x}_{(0)} + \alpha \mathbf{d}_{(0)} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \alpha \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}$$

$$\mathbf{x}_{(1)} = \begin{bmatrix} 1 + \alpha \\ 1 + \alpha/2 \end{bmatrix}$$

olarak belirlenebilir. Bu yöneyin amaç fonksiyonunda yerine konulması halinde;

$$f(\mathbf{x}_{(1)}) = \frac{\alpha}{2} \sin(1 + \alpha) \cos\left(1 + \frac{\alpha}{2}\right)$$

denklemi yazılabilir. “Quasi-Newton” yöntemi ile çözüme gidilmesi için öncelikle başlangıç noktasının ve bu noktadan başlayarak atılacak adım büyüklüğünün öngörülmesi gerekir. Sorunun çözümü sadedinde başlangıç noktası olarak  $\alpha_1 = 1$  ve adım büyüklüğü olarak  $\Delta\alpha = 0.1$  değerlerini dikkate alalım. Ayrıca işlemlerin sonlandırılması noktasında dikkate alınacak hassasiyet değerini de  $\varepsilon = 0.01$  olarak belirleyelim. Verilen değerlerin bir sonraki nokta tayininde dikkate alınması halinde;

$$\alpha_2 = \alpha_1 - \frac{\Delta\alpha [f(\alpha_1 + \Delta\alpha) - f(\alpha_1 - \Delta\alpha)]}{2[f(\alpha_1 + \Delta\alpha) - 2f(\alpha_1) + f(\alpha_1 - \Delta\alpha)]} = 1 - \frac{0.1(f(1.1) - f(0.9))}{2[f(1.1) - 2f(1) + f(0.9)]}$$

$$\alpha_2 = 0.3389$$

değerine ulaşılır. Ulaşılan bu noktanın istenen çözüm hassasiyetini sağlayıp sağlamadığı kontrol edildiğinde;



$$|f'(\alpha_{i+1})| \cong \left| \frac{f(\alpha_{i+1} + \Delta\alpha) - f(\alpha_{i+1} - \Delta\alpha)}{2\Delta\alpha} \right| \leq \varepsilon$$

$$|f'(0.3389)| \cong \left| \frac{f(0.3389 + 0.1) - f(0.3389 - 0.1)}{2 \times 0.1} \right| \leq 0.01$$

$$0.1281 > 0.01$$

eşitsizliği tespit edilebilir. Dolayısıyla bir sonraki adıma geçilir;

$$\alpha_3 = \alpha_2 - \frac{\Delta\alpha[f(\alpha_2 + \Delta\alpha) - f(\alpha_2 - \Delta\alpha)]}{2[f(\alpha_2 + \Delta\alpha) - 2f(\alpha_2) + f(\alpha_2 - \Delta\alpha)]} = 1 - \frac{0.1(f(0.4389) - f(0.2389))}{2[f(0.4389) - 2f(0.3389) + f(0.2389)]}$$

$$\alpha_2 = 0.6101$$

Bu noktadaki çözüm hassasiyeti;

$$|f'(0.6101)| \cong \left| \frac{f(0.7101) - f(0.5101)}{2 \times 0.1} \right| \leq 0.01$$

$$0.0189 > 0.01$$

değerinde olup üçüncü adım atılır;

$$\alpha_4 = \alpha_2 - \frac{\Delta\alpha[f(\alpha_3 + \Delta\alpha) - f(\alpha_3 - \Delta\alpha)]}{2[f(\alpha_3 + \Delta\alpha) - 2f(\alpha_3) + f(\alpha_3 - \Delta\alpha)]} = 1 - \frac{0.1(f(0.7101) - f(0.5101))}{2[f(0.7101) - 2f(0.6101) + f(0.5101)]}$$

$$\alpha_4 = 0.5774$$

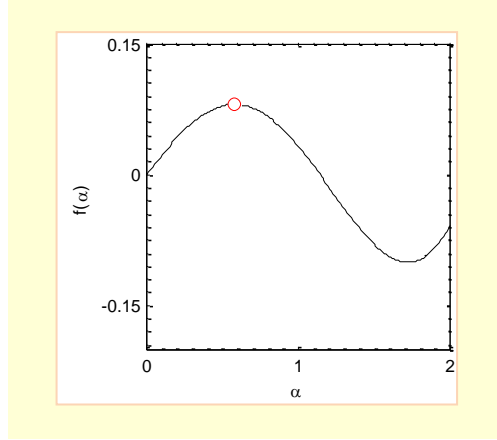
Elde edilen bu noktanın çözüme yakınlığı test edildiğinde;

$$|f'(0.5774)| \cong \left| \frac{f(0.6774) - f(0.4774)}{2 \times 0.1} \right| \leq 0.01$$

$$8.43 \times 10^{-5} \leq 0.01$$

olduğu gözlemlenir. İstenen çözüm hassasiyetine ulaşılması nedeniyle optimal noktanın  $\alpha = 0.5774$  ve bu noktadaki amaç fonksiyonunun  $f(0.5774) = 0.0804$  değerine sahip olduğu görülür. Ancak elde edilen nokta minimum nokta olmayıp, gerçekte maksimum noktadır.





Şekil 2.61 “Quasi-Newton” yöntemi ile ulaşılan azami nokta.



#### (4) “Secant” yöntemi

“Quasi-Newton” yönteminin tanımı esnasında dikkate alınan ifade;

$$\alpha_{i+1} = \alpha_i - \frac{f'(\alpha_i)}{f''(\alpha_i)}$$

denklemi şeklindeki. Bu ifadedeki ikinci mertebeden türev terimi, sayısal olarak birinci mertebeden türev cinsinden ifade edilirse;

$$f''(\alpha_i) = \frac{f'(\alpha_i) - f'(\alpha_{i-1})}{\alpha_i - \alpha_{i-1}} \quad (2.78)$$

eşitliği yazılabilir. Elde edilen bu denklem bir önceki denklemde yerine konulup, uygun düzenlemeler yapıldığında;

$$\alpha_{i+1} = \alpha_i - \frac{f'(\alpha_i)}{\frac{f'(\alpha_i) - f'(\alpha_{i-1})}{\alpha_i - \alpha_{i-1}}} \quad (2.79)$$





$$\alpha_{i+1} = \frac{f'(\alpha_i)\alpha_{i-1} - f'(\alpha_{i-1})\alpha_i}{f'(\alpha_i) - f'(\alpha_{i-1})} \quad (2.80)$$

ifadesine ulaşılır. Görüldüğü üzere bu denklemde sadece birinci mertebeden türev bilgisine ihtiyaç duyulmaktadır. Dolayısıyla “Secant” yöntemi olarak adlandırılan söz konusu yöntemde tarama, “Quasi-Newton” yönteminde olduğu gibi sadece amaç fonksiyonu değerine göre değil, amaç fonksiyonunun birinci mertebeden türev değerine göre yapılmaktadır. Bu meyanda “Secant” yöntemi sayısal yaklaşım özelliğini az da olsa kaybetmekte, belli oranda analitik yaklaşım özelliği göstermektedir. Diğer taraftan yöntemin uygulanabilmesi için sadece bir başlangıç noktası yeterli olmayıp, en az iki noktada amaç fonksiyonuna ait eğim değerinin verilmesi gerekir. Tekrarlı işlemlerin sonlandırılması ise birinci mertebeden türev değerinin sıfıra yakınlığına ya da sayısal anlamda  $\varepsilon$  değerindeki hassasiyete ulaşmış olduğuna bakılarak belirlenir;

$$|f'(\alpha_{i+1})| \leq \varepsilon \quad (2.81)$$

Yöntemin daha iyi anlaşılması için müteakip örneği ele alalım. Buna göre tanımlanan;

$$f(\alpha) = -3\alpha^2 + 2\alpha^3 + 5 \sin \alpha \cos \alpha$$

amaç fonksiyonunun optimal noktasını “Secant” yöntemi ile elde edelim. Bu meyanda başlangıç noktası ve yakın komşu nokta olarak [0.20 0.25] noktalarını ve hassasiyet değeri olarak da  $\varepsilon = 0.05$  değerini dikkate alalım. Yöntemin uygulanması için amaç fonksiyonunun birinci mertebeden türev bilgisine ihtiyaç duyulmaktadır;

$$\frac{\partial f(\alpha)}{\partial \alpha} = 6(\alpha^2 - \alpha^2) + 5(\cos^2 \alpha - \sin^2 \alpha)$$

Bu ifade kullanılarak komşu iki noktanın eğim değerleri hesaplanırsa;

$$\begin{aligned} f'(0.20) &= 3.6453 \\ f'(0.25) &= 3.2629 \end{aligned}$$

elde edilir. Bu değerler bir sonraki noktanın tespitinde yerine koyulursa;

$$\alpha_3 = \frac{f'(0.25)0.20 - f'(0.20)0.25}{f'(0.25) - f'(0.20)} = 0.6766$$





sonucuna ulaşılır. Bulunulan noktadaki eğim değeri ile hassasiyet değeri karşılaştırılması yapılırsa;

$$|f'(0.6766)| = 0.2338 > \varepsilon = 0.05$$

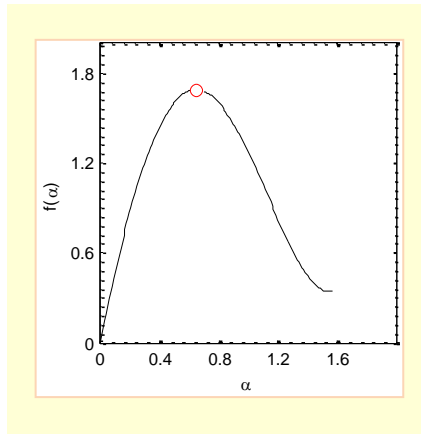
eşitsizliği nedeniyle bir sonraki adıma geçilir.

$$\alpha_4 = \frac{f'(0.6766)0.25 - f'(0.25)0.6766}{f'(0.6766) - f'(0.25)} = 0.6481$$

Akabinde yeni noktadaki eğim değeri ile hassasiyet değeri karşılaştırılması yapıldığında;

$$|f'(0.6481)| = 0.0127 \leq \varepsilon = 0.05$$

eşitsizliğinin sağlandığı gözlemlenebilir. İstenen hassasiyet değerine ulaşılması nedeniyle tarama işlemi sonlandırılabilir. Buna göre optimal nokta  $\alpha^* = 0.6481$  ve karşılık gelen amaç fonksiyon değeri  $f(0.6481) = 1.6907$  şeklindedir. Şekil 2.62 dâhilinde sorunun analitik çizimi ve elde edilen azami çözüm noktası görülmektedir.



Şekil 2.62 “Secant” yöntemi ile ulaşılan azami nokta.





$$f(\alpha) = -3\alpha^2 + 2\alpha^3 + 5 \sin \alpha \cos \alpha$$

MATLAB uygulaması

```
clc; clear; eps = 0.05;
a(1) = 0.20; a(2) = 0.25;
for i = 1:2
    df(i) = 6 * (a(i)^2 - a(i)) + 5 * ((cos(a(i)))^2 - (sin(a(i)))^2);
end
a(i+1) = (df(2)*a(1) - df(1)*a(2)) / (df(2) - df(1));
Df = 6 * (a(i+1)^2 - a(i+1)) + 5 * ((cos(a(i+1)))^2 - (sin(a(i+1)))^2);
df(i+1) = Df; k = 3;
while abs(Df) > eps
    a(k+1) = (df(k)*a(k-1) - df(k-1)*a(k)) / (df(k) - df(k-1));
    df(k+1) = 6 * (a(k+1)^2 - a(k+1)) + 5 * ((cos(a(k+1)))^2 - (sin(a(k+1)))^2);
    Df = df(k+1); k = k + 1;
end
a
```

```
a =
    0.2000    0.2500    0.6766    0.6481
```

### (5) “Newton” yöntemi

Hatırlanacağı üzere “Quasi-Newton” yönteminin geliştirilmesi safhasında her hangi bir fonksiyonun  $\alpha_{i+1}$  noktasındaki değeri, Taylor serisine açma yöntemi ile ifade edilmiş ve ikilenik hassasiyet yaklaşımı ile söz konusu seri basitleştirilerek;

$$\alpha_{i+1} = \alpha_i - \frac{f'(\alpha_i)}{f''(\alpha_i)} \quad (2.82)$$

denklemi elde edilmişti. Elde edilen bu denklem, amaç fonksiyonunun birinci ve ikinci mertebeden türev verilerine dayalı olup fonksiyonun analitik olarak türevlenebilir olması halinde oldukça etkin bir şekilde tarama işlemi yapabilir. Bununla beraber “Quasi-Newton” yönteminde olduğu gibi, tarama bölgesinde durağan veya azami noktanın varlığı söz konusu olduğunda atılan adımlar minimum noktadan ziyade bahse konu noktalara doğru ilerler. Dolayısıyla minimizasyon amaçlı bir sorunda çözüme ulaşamayabilir. Diğer taraftan yöntem, büyük oranda analitik verilere dayalı olup bu yönüyle sayısal yöntem olarak adlandırmak zordur. İşlem adımlarının sonlandırılması, birinci mertebeden türev değerinin sıfıra yakınlığına ya da sayısal anlamda  $\varepsilon$  değerindeki hassasiyete ulaşıp ulaşmadığına bakılarak belirlenir;



$$|f'(\alpha_{i+1})| \leq \varepsilon \quad (2.83)$$

Yöntemin daha iyi anlaşılması için müteakip örneği ele alalım. Buna göre;

$$f(\alpha) = -3\alpha - 5 \sin^2 \alpha \cos \alpha$$

amaç fonksiyonunun optimal noktasını “Newton” yöntemini kullanarak elde edelim. Bu meyanda başlangıç noktası olarak  $\alpha_1 = 1$  noktasını ve hassasiyet değeri olarak  $\varepsilon = 0.01$  değerini dikkate alalım. Yöntemin uygulanması için amaç fonksiyonunun birinci ve ikinci mertebeden türev bilgilerine ihtiyaç duyulmaktadır;

$$\frac{\partial f(\alpha)}{\partial \alpha} = -3 + 5 \sin^3 \alpha - 10 \sin \alpha \cos^2 \alpha$$

$$\frac{\partial^2 f(\alpha)}{\partial \alpha^2} = 35 \cos \alpha \sin^2 \alpha - 10 \cos^3 \alpha$$

Bu ifadelerde başlangıç noktası dikkate alınırsa;

$$\begin{aligned} f'(1) &= -2.4774 \\ f''(1) &= 11.8128 \end{aligned}$$

değerleri elde edilir. Bu değerler, bir sonraki noktanın tespitinde yerine koyulursa;

$$\alpha_2 = \alpha_1 - \frac{f'(\alpha_1)}{f''(\alpha_1)} = 1 - \frac{-2.4774}{11.8128} = 1.2097$$

noktasına ulaşılır. Ulaşılan bu noktadaki eğim değeri, istenen hassasiyet değeri ile karşılaştırılırsa;

$$|f'(1.2097)| = 0.0738 > 0.01$$

eşitsizliği gözlemlenebilir. Akabinde bir sonraki adıma geçilir;

$$\alpha_3 = \alpha_2 - \frac{f'(\alpha_2)}{f''(\alpha_2)} = 1.2097 - \frac{-0.0738}{10.3807} = 1.2168$$

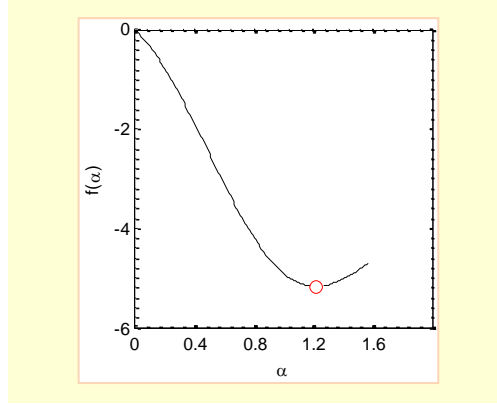
Bu noktadaki eğim;

$$|f'(1.2168)| = 4.34 \times 10^{-4} \leq 0.01$$





olarak elde edilir ve istenen hassasiyetteki çözüm noktasına ulaşıldığı görülür. Buna göre optimal nokta  $\alpha = 1.2168$  ve karşılık gelen amaç fonksiyon değeri  $f(1.2168) = -5.1754$  şeklindedir.



Şekil 2.63 “Newton” yöntemi ile ulaşılan asgari nokta.

$$f(\alpha) = -3\alpha - 5 \sin^2 \alpha \cos \alpha$$

MATLAB uygulaması

```
clc; clear; i = 1;
a(i) = 1;
eps = 0.01;
df = 5*sin(a(i))^3 - 10*cos(a(i))^2*sin(a(i)) - 3;
dff = 35*cos(a(i))*sin(a(i))^2 - 10*cos(a(i))^3;
while (abs(df)>eps)
    a(i+1) = a(i)-df/dff;
    df = 5*sin(a(i+1))^3 - 10*cos(a(i+1))^2*sin(a(i+1)) - 3;
    dff = 35*cos(a(i+1))*sin(a(i+1))^2 - 10*cos(a(i+1))^3;
    i = i + 1;
end
```

```
data = [a' (-3*a-5*(sin(a)).^2.*cos(a))']
```

```
data =
    1.0000    -4.9129
    1.2097    -5.1751
    1.2168    -5.1754
```

## Tarihten Bir Sayfa



(45)

**T**

ek değişkenli sorunların çözümünde dikkate alınan yöntemlerin geçmişi çok eskilere dayanır. Her ne kadar yöntemlerin ismi veya çıkış noktası “Newton” yöntemi olarak adlandırılrsa da esasen yöntemlerin, bahse konu isimle veya çalışmaları ile doğrudan ilişkilendirilmesi doğru olmaz. Kayıtlara göre “Secant” yönteminin ilk uygulama örneklerini M.Ö. 1800’lü yıllarda Mısır’da görmekteyiz. İskoç gezgin A. Henry Rhind’in 1858 yılındaki Mısır seyahati esnasında antika eşyalar satan bir dükkândan tesadüfen satın aldığı “Ahmes” papirüsleri üzerindeki kayıtlara göre “Secant” yönteminin ilk örneği, tarihte “çift hata konumu kuralı” olarak bilinen bir yönetime dayanmakta.

“Ahmes” papirüsleri hiyeroglif alfabesi kullanılarak kaleme alınan ve büyük oranda lafız ile çizimlerden oluşan bir eser. İçerisinde çok sayıda problem ve bu problemlerin çözümüne yönelik yaklaşımlar mevcut. Bahse konu papirüste geçen ve “Secant” yöntemi ile alakalı olduğu düşünülen ilk sorun, “tek hata konumu kuralı<sup>i</sup>” olarak bilinen ve aşağıdaki gibi tanımlanan problemdir;

$$a_1x + a_2x + \dots + a_nx = c$$

denkleme göre  $x$  değeri nedir? Bu sorunun çözümü için Mısırlılar, doğrusal karakterdeki katsayıları bilinen olarak dikkate alıp, bilinmeyen  $x$  için bir  $x_0$  tahmini yapmakta ve buradan;

$$ax_0 = c_0, \quad a = \sum_{i=1}^n a_i$$

<sup>i</sup> The rule of double false positions

<sup>ii</sup> The rule of a single false position



ifadesini dikkate alarak,

$$x = \frac{c}{c_0/x_0} = c \frac{x_0}{c_0}$$

eşitliğini elde etmektedir. Bir sonraki problem, mevcut sorunu bir aşama daha ileri götürmektedir;

$$a_1x + a_2x + \dots + a_nx + b = c, \quad b \neq 0$$

ifadelerine göre  $x$  değeri nedir? Belirtilen sorun, Mısırlı âlimlerce “çift hata konumu kuralı” yöntemi ile çözülmektedir. Buna göre önce bilinmeyen  $x$  için bir  $x_0$  tahmini yapmakta ve buradan;

$$ax_0 + b - c = e_0$$

hata değeri hesaplanmaktadır. Benzer şekilde ikinci bir  $x_1$  tahmini yapılmakta ve buradan;

$$ax_1 + b - c = e_1$$

hata değeri hesaplanmaktadır. Bu iki ifade birbirinden çıkarılarak;

$$ax_1 - ax_0 = e_1 - e_0$$

$$a = \frac{e_1 - e_0}{x_1 - x_0}$$

ifadesine ulaşılmaktadır. Bu denklem  $ax_0 + b - c = e_0$  eşitliğinde yerine konularak;

$$\frac{e_1 - e_0}{x_1 - x_0}x_0 + b - c = e_0$$

$$c - b = \frac{e_1x_0 - e_0x_1}{x_1 - x_0}$$

eşitliği elde edilmektedir. Bu eşitlik başlangıçta ifade edilen sorunun tanımında dikkate alınırsa;

$$ax + b = c$$

$$x = \frac{c - b}{a} = \frac{\frac{e_1x_0 - e_0x_1}{x_1 - x_0}}{\frac{e_1 - e_0}{x_1 - x_0}} = \frac{e_1x_0 - e_0x_1}{e_1 - e_0}$$

sonucuna varılır. Hata değerlerinin dikkate alınan  $x$  değerine göre değişeceği düşünüldüğünde;

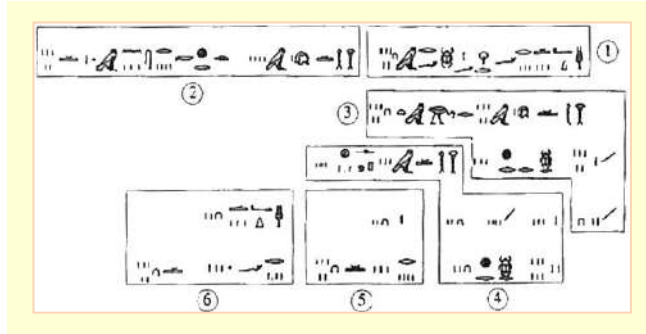


$$e_0 = f(x_0)$$

$$e_1 = f(x_1)$$

$$x = \frac{f(x_1)x_0 - f(x_0)x_1}{f(x_1) - f(x_0)}$$

denklemleri elde edilir. Dikkat edileceği üzere elde edilen denklemler “Secant” yönteminin birebir uygulaması olup sadece türev kavramı, ikinci bir fonksiyon olarak dikkate alınmaktadır. Belirtilen sorunun “Ahmes” papirüslerinde geçen hiyeroglif metinleri, Şekil 2.64’te görülmektedir.



Şekil 2.64 “Ahmes” papirüsünde geçen “çift hata konumu kuralı” içerikli metin<sup>(46)</sup>





## b. Çok boyutlu arama yöntemleri

Sayısal yöntemlerde başlangıç noktası  $n$  değişkene sahip  $\mathbf{x}_{(0)}$  olmak üzere bir sonraki nokta  $\mathbf{x}_{(1)} = \mathbf{x}_{(0)} + \alpha \mathbf{d}_{(0)}$  denklemi ile belirlenmekteydi. Burada  $\alpha$  adım büyüklüğünü ve  $\mathbf{d}_{(0)}$  başlangıç noktasından itibaren ilerlenecek yönü göstermekteydi. Tek boyutlu arama yöntemleri kullanılarak  $\alpha$  değeri tespit edilmekte ve akabinde elde edilen değer ana denklemde yerine konularak bir sonraki noktaya ulaşılmaktaydı. Tasarım yöneyinin tek değişkenli olması halinde adım büyüklüğü için dikkate alınan yöntemler doğrudan tasarım yöneyi için de dikkate alınabilir ve sorun kolayca çözülebilir. Ancak tasarım yöneyinin çok boyutlu olması durumunda mevcut tek boyutlu yöntemler yeterli olmayabilir. Bu durumda yeni yöntemlerin bulunması ya da mevcut tek boyutlu arama yöntemlerinin çok boyutlu arama için geliştirilmesi gerekir.

Çok boyutlu arama yöntemlerini, taramada kullanılan bilgi kaynağına göre ikiye ayırmak mümkündür. Doğrudan tarama yöntemleri olarak adlandırılan ilk grubun genel özelliği  $f(\mathbf{x})$  amaç fonksiyon değerine göre adım atılacak doğrultunun tespit edilmesidir. Dolayısıyla gradyan verisi olmadan tarama yapma imkânı söz konusudur. Bu tür yöntemlerin iyi tarafı, birincil veya ikincil türev bilgisine ihtiyaç duyulmaması, ilave olarak gradyan verisinin yönlendirmesi sonucu ortaya çıkan yerel çözümlere saplanıp kalma riskinin azalmasıdır. Daha açık bir ifade ile yerel çözümlerden ziyade genel çözüme ulaşma şansı daha yüksektir. Ancak sadece amaç fonksiyonu değerine göre istenen hassasiyette çözüme ulaşmak kolay olmayabilir.

İkinci grup yöntemler gradyan esaslı yöntemlerdir. Bu yöntemlerden bazıları zaten tek boyutlu arama yöntemleri bahsinde ifade edilmişti. Hatırlanacağı üzere gidilecek yön ve atılacak adım büyüklüğü için birincil ve bazı yöntemlerde ikincil türev verisine ihtiyaç duyulmaktaydı. Söz konusu verilerle hassas çözümlere ulaşmak neredeyse kesindir. Ancak ulaşılan çözümün genel çözüm olup olmaması, büyük oranda başlangıç noktasına bağlı olup yerel çözüme saplanma riski söz konusudur.

Müteakip paragraflarda doğrudan tarama bazlı yöntemlerden “Powell” yöntemine değinilecek, gradyan bilgisine dayalı yöntemler arasında ise “Endik azalma”, “Eşlenik gradyan”, “Newton”, “Davidon-Fletcher-Powell (DFP)”, “Broydon-Fletcher-Goldfarb-Shanno (BFGS)” ve “Ardışık ikilenik programlama” yöntemlerinden bahsedilecektir.

---

<sup>i</sup> Direct search methods

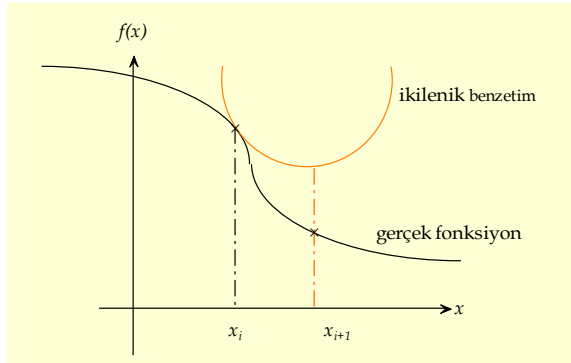
<sup>ii</sup> Sequential quadratic programming





## (1) “Powell” yöntemi

“Powell” yöntemi, 1964 yılında Michael J.D. Powell tarafından kaleme alınan ve ikilenik fonksiyonların türev verisine ihtiyaç duyulmadan minimum noktasının bulunması konusunu<sup>i</sup> içeren makaleye dayanır. Yöntem, özellikle ikilenik fonksiyonlarda başlangıç noktasına bağlı kalmadan aşgari noktanın nispeten kolayca bulunabilmesine imkân vermektedir. Bununla beraber ikilenik formda olmayan sorunlarda da kullanılabilir. Zira Şekil 2.65’ten de görüleceği üzere doğrusal olmayan bir fonksiyonun yerel olarak ikilenik fonksiyon ile benzeşiminin kurulması ve bu sayede ikilenik fonksiyona ait azami noktanın yerinin gerçek fonksiyondaki izdüşümünün dikkate alınması ile tarama yapılması mümkündür. Ancak bu durumda “Powell” yönteminin, ikilenik fonksiyonda olduğu gibi iyi bir performans sergilemesi beklenmemelidir.



Şekil 2.65 Herhangi bir fonksiyon ile eşlenik fonksiyon arasında benzeşim kurulması.

İkilenik fonksiyon yaklaşımı yanında yöntemde dikkate alınan diğer bir husus, eşlenik doğrultu kullanımıdır. Eşlenik doğrultu kavramını açıklamak için öncelikle ikilenik fonksiyon formuna tekrar bakmakta yarar görülmektedir. Hatırlanacağı üzere ikilenik fonksiyon matris formunda;

$$f(x) = x^T A x + B^T x + c \quad (2.84)$$

şeklinde ifade edilmektedir.

<sup>i</sup> Makalenin özgün hali “An efficient method for finding the minimum of a function of several variables without calculating derivatives” şeklindedir.

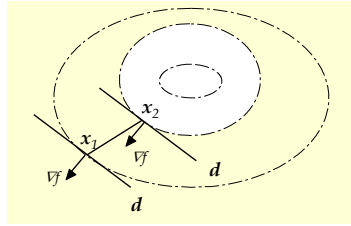




Burada  $A$  terimi,  $n$  boyutlu tasarım yöneyi tanımını dahilinde  $n \times n$  ebadındaki matrisi,  $B$  doğrusal terimlerin katsayı yöneyini ve  $c$  sabit sayıyı temsil etmektedir.  $A$  matrisinin birim matris olması halinde herhangi iki boyuttaki birim yöneylerin (örneğin  $A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$  olmak üzere iki boyutlu ortamda  $\begin{bmatrix} 1 & 0 \end{bmatrix}^T$  ve  $\begin{bmatrix} 0 & 1 \end{bmatrix}^T$  yöneyleri) çarpımının doğal olarak sifıra eşitleneceği, aralarında da 90 derecelik bir açı olacağı ve bu yöneylerin eşlenik yöneyler olarak tanımlanabileceği kolayca öngörülebilir. Ancak açının farklı bir değerinde ve çarpımın da hala sifıra eşit olması halinde tanım  $A$ -eşlenik olarak ifade edilir. Buna göre  $\mathbf{d}_1$  ve  $\mathbf{d}_2$  gibi herhangi iki doğrultuyu ifade eden yöneyler arasında;

$$\mathbf{d}_1^T A \mathbf{d}_2 = 0 \quad (2.85)$$

ilişkisinin söz konusu olması halinde bu iki yöneyin  $A$ -eşlenik olduğu söylenebilir. Dolayısıyla tanıma göre uygun iki yöneyin seçilmesi halinde belirtilen çarpım terimi sifıra eşitlenebilir.



Şekil 2.66 Eşlenik yöneyler kavramı:  $\mathbf{d}$  ve  $\mathbf{x}_2 - \mathbf{x}_1$ .

Diğer taraftan Şekil 2.66'da gösterildiği üzere eşlenik formdaki  $f(\mathbf{x})$  fonksiyonu üzerinde birbirine paralel iki nokta ve bu noktalara doğal olarak teğet geçen bir  $\mathbf{d}$  doğrultusu düşünelim. Fonksiyonun her bir noktadaki gradyan değeri dikkate alındığında;

$$\nabla f(\mathbf{x}_1) = A \mathbf{x}_1 + B \quad (2.86)$$

$$\nabla f(\mathbf{x}_2) = A \mathbf{x}_2 + B \quad (2.87)$$

eşitlikleri yazılabilir. Bu iki ifade birbirinden çıkarıldığında;

$$\nabla f(\mathbf{x}_2) - \nabla f(\mathbf{x}_1) = A(\mathbf{x}_2 - \mathbf{x}_1) \quad (2.88)$$

eşitliği elde edilebilir. Elde edilen eşitliğin her iki tarafının  $\mathbf{d}$  doğrultusu ile çarpılması halinde;



$$d[\nabla f(x_2) - \nabla f(x_1)] = dA(x_2 - x_1) \quad (2.89)$$

ifadesi ve buradan;

$$d\nabla f(x_1) = 0 \quad (2.90)$$

$$d\nabla f(x_2) = 0 \quad (2.91)$$

eşitlikleri dikkate alınarak;

$$dA(x_2 - x_1) = 0 \quad (2.92)$$

denklemleri kolayca yazılabilir. Dikkat edileceği üzere  $x_i$  noktasındaki gradyan ile  $d$  doğrultusu birbirine dik yöneylerdir. Elde edilen son denklemden hareketle  $x_2 - x_1$  yöneyi ile  $d$  yöneyinin  $A$ -eşlenik olduğu kolayca görülebilir. Powell'ın ifade ettiği üzere<sup>(47)</sup>, birbiri ile ardışık olarak  $A$ -eşlenik olan bir seri  $d_i$  doğrultusu kullanılarak ikilenik formdaki  $f(x)$  fonksiyonunun asgari noktasını,  $n^2$  adım veya öncesinde elde etmek mümkündür. Buna göre taramada başlangıç noktası  $x_0$  olmak üzere optimal noktayı;

$$x^* = x_0 + \sum_{i=1}^n \lambda_i d_i \quad (2.93)$$

şeklinde ifade edebiliriz. Burada  $d_i$  gidilen eşlenik yönü ve  $\lambda_i$  o yöndeki adım büyüklüğünü göstermektedir. Ulaşılan optimal noktada gradyan değerinin sıfıra eşit olacağı düşünüldüğünde;

$$\nabla f(x^*) = Ax^* + B = 0 \quad (2.94)$$

$$A \left[ x_0 + \sum_{i=1}^n \lambda_i d_i \right] + B = 0 \quad (2.95)$$

elde edilebilir. Bu denklemde çarpma işlemi dağıtılsa;

$$B + Ax_0 + A \sum_{i=1}^n \lambda_i d_i = 0 \quad (2.96)$$

terimleri yazılabilir. İfadedeki son terim, nihayetinde katsayı ile beraber bir yöneyi ifade etmektedir. Dolayısıyla bu denklem;

$$B + Ax_0 + A\lambda_j d_j = 0 \quad (2.97)$$





şeklinde kısaltılarak irdelenebilir. Burada  $\lambda_j \mathbf{d}_j$  terimleri,  $\lambda_i \mathbf{d}_i$  çarpımlarının toplamı sonucu elde edilen değeri göstermektedir. Son denklem güdüleme amaçlı olarak  $\mathbf{d}_j$  ile çarpılıp elde edilen ifadede  $\lambda_j$  katsayısı yalnız bırakılırsa;

$$\lambda_j = -\frac{(\mathbf{B} + A\mathbf{x}_0)^T \mathbf{d}_j}{\mathbf{d}_j^T A \mathbf{d}_j} \quad (2.98)$$

sonucuna ulaşılabilir. Öte yandan sayısal yöntemin bir gereği olarak ardı ardına elde edilen noktalar arasında;

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha_i^* \mathbf{d}_i \quad (2.99)$$

ilişkisi dikkate alınmaktaydı. Burada  $\alpha_i^*$  terimi, optimal adım büyüklüğünü göstermekteydi. Optimal adım büyüklüğünün elde edilmesi bağlamında;

$$\frac{\partial f(\mathbf{x}_{i+1})}{\partial \alpha_i} = \frac{\partial f(\mathbf{x}_{i+1})}{\partial \mathbf{x}_{i+1}} \frac{\partial \mathbf{x}_{i+1}}{\partial \alpha_i} = 0 \quad (2.100)$$

yaklaşımı ile;

$$\frac{\partial f(\mathbf{x}_{i+1})}{\partial \mathbf{x}_{i+1}} = \nabla f(\mathbf{x}_{i+1}) \quad (2.101)$$

$$\frac{\partial \mathbf{x}_{i+1}}{\partial \alpha_i} = \mathbf{d}_i \quad (2.102)$$

eşitlikleri yazılıp, bu ifadelerden;

$$\mathbf{d}_i^T \nabla f(\mathbf{x}_{i+1}) = 0 \quad (2.103)$$

eşitliğine varılabilir. Diğer taraftan gradyan verisi için ikilenik yaklaşımla  $\nabla f(\mathbf{x}_{i+1}) = \mathbf{B} + A\mathbf{x}_{i+1}$  denklemini, bir sonraki adım için de başlangıçtaki  $\mathbf{x}_i + \alpha_i^* \mathbf{d}_i$  tanımını dikkate alırsak;

$$\mathbf{d}_i^T [\mathbf{B} + A(\mathbf{x}_i + \alpha_i^* \mathbf{d}_i)] = 0 \quad (2.104)$$

eşitliği elde edilir. Denklemden optimal adım büyüklüğü yalnız bırakıldığında;

$$\alpha_i^* = -\frac{(\mathbf{B} + A\mathbf{x}_i)^T \mathbf{d}_i}{\mathbf{d}_i^T A \mathbf{d}_i} \quad (2.105)$$

sonucuna varılır.



Ardışık noktaların birbiri ile ve hatta başlangıç noktası ile ilişkisi dikkate alındığında;

$$\mathbf{x}_i = \mathbf{x}_0 + \sum_{j=1}^{i-1} \alpha_j^* \mathbf{d}_j \quad (2.106)$$

yazılabilir. Bu eşitliğin her iki tarafı  $A\mathbf{d}_i$  ile çarpılırsa;

$$\mathbf{x}_i^T A\mathbf{d}_i = \mathbf{x}_0^T A\mathbf{d}_i + \sum_{j=1}^{i-1} \alpha_j^* \mathbf{d}_j^T A\mathbf{d}_i \quad (2.107)$$

elde edilir. Hatırlanacağı üzere ardı ardına seçilen  $A$  –eşlenik yöneylerden dolayı  $\mathbf{d}_j^T A\mathbf{d}_i = 0$  olacaktır. Dolayısıyla Denklem 2.107 daha basit haliyle;

$$\mathbf{x}_i^T A\mathbf{d}_i = \mathbf{x}_0^T A\mathbf{d}_i \quad (2.108)$$

olarak dikkate alınabilir. Bu sonuç, optimal adım büyüklüğü denkleminde yerine koyulursa;

$$\alpha_i^* = - \frac{(\mathbf{B} + A\mathbf{x}_0)^T \mathbf{d}_i}{\mathbf{d}_i^T A\mathbf{d}_i} \quad (2.109)$$

elde edilir. Dikkat edileceği üzere bu denklem daha önce elde edilen  $\lambda_j$  terimi ile birebir aynıdır. Dolayısıyla  $f(\mathbf{x})$  fonksiyonu ikilenik ve ardı ardına gelen  $\mathbf{d}_i$  yöneyleri  $A$  –eşlenik olmak kaydıyla,  $n$  adımda optimal noktaya ulaşmak mümkün görünmektedir. Ancak her adımda  $n$  boyutta işlem yapılması nedeniyle yakınsama  $n^2$  işlem adımı sonrasında gerçekleşebilecektir. Yaklaşımın ana dayanağını oluşturan  $A$  –eşlenik yöneylerin eldesi noktasında Powell’ın teklifi, öncelikle her bir eksen dâhilindeki birim yöneylerin kullanılması, akabinde elde edilen sonuca göre ilk ve son nokta arası doğrultuyu temel alan birim yöneyin dikkate alınmasıdır.

Geliştirilmiş Powell yöntemi işlem basamakları dâhilinde başlangıç noktası;

$$\mathbf{x}_i^{(k)}, \quad i = 0, \quad k = 1$$

olmak üzere her bir eksenindeki birim yöney doğrultusu dikkate alınarak;

$$\mathbf{x}_i^{(k)} = \mathbf{x}_{i-1}^{(k)} + \lambda_i^* \mathbf{d}_i, \quad i = 1, 2, \dots, n$$



noktaları hesaplanır. Tarama esnasında amaç fonksiyonunda meydana gelen en büyük azalış miktarı;

$$\text{maks} [f(\mathbf{x}_{m-1}^{(k)}) - f(\mathbf{x}_m^{(k)})]$$

ifadesi çerçevesinde tespit edilerek bu azalışın elde edildiği  $m$  indisi not edilir. Akabinde elde edilen son nokta ile ilk nokta dikkate alınarak eşlenik doğrultu hesaplanır;

$$\mathbf{d}_e^{(k)} = \mathbf{x}_n^{(k)} - \mathbf{x}_0^{(k)}.$$

Bu doğrultu temel alınarak yeni nokta elde edilir;

$$\mathbf{x}_0^{(k+1)} = \mathbf{x}_0^{(k)} + \lambda^* \mathbf{d}_e^{(k)}.$$

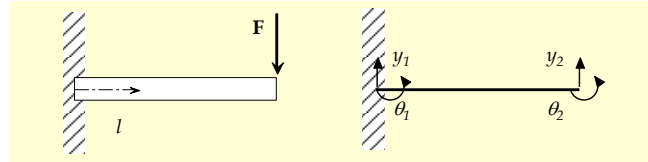
Hesaplanan adım büyüklüğünün;

$$|\lambda^*| < \left[ \frac{f(\mathbf{x}_0^{(k)}) - f(\mathbf{x}_0^{(k+1)})}{f(\mathbf{x}_{m-1}^{(k)}) - f(\mathbf{x}_m^{(k)})} \right]^{1/2} \quad (2.110)$$

ölçütünü sağlaması halinde daha önce dikkate alınan ilerleme doğrultuları aynen kullanılmaya devam edilir. Aksi halde  $m$  indisi doğrultu yöneyi, yeni elde edilen  $\mathbf{d}_e^{(k)}$  eşlenik yöneyiyle değiştirilir. Değişim sonrası başlangıçta ifade edilen işlem adımları tekrar edilir. Bu arada dikkate alınan yakınsama ölçütü ise;

$$\|\mathbf{x}_i^{(k+1)} - \mathbf{x}_i^{(k)}\| \leq \epsilon$$

şeklindedir. Yöntemde dikkat çekmek istediğimiz bir diğer husus, elde edilen ardışık  $\mathbf{d}_e^{(k)}$  yöneylerinin  $A$ -eşlenik yöney olmasıdır. İfade edilen işlem basamaklarının daha somut bir şekilde anlaşılabilmesi için yapı disiplinine ait müteakip sorunu, Powell yöntemini kullanarak çözelim.



Şekil 2.67 Ankastre kirişte azami eğilmenin tespiti.



Buna göre ankastre olarak duvara sabitlenmiş bir kirişin uç noktasına  $\mathbf{F}$  kuvvetinin etki ettiğini düşünelim. Etki eden kuvvetin neden olacağı azami eğilme değerini, kirişin toplam potansiyel enerjisinin minimize edilmesi yolu ile bulmak mümkündür. Kirişin potansiyel enerji denklemini;

$$P = \frac{EI}{2L^3} \int_0^L \left( \frac{\partial^2 y}{\partial \beta^2} \right)^2 d\beta + Fy_2$$

şeklinde ifade edebiliriz. Burada  $L$  kirişin boyunu,  $\beta = l/L$  normalize edilmiş mesafeyi,  $E$  kirişe ait elastisite modülünü ve  $I$  eylemsizlik momentini temsil etmektedir. Bu denklem uygun güdüleme işlemleri yapılarak;

$$f(y_2, \theta_2) = 12y_2^2 + 4\theta_2L^2 - 12y_2\theta_2L + 2y_2$$

formuna çevrilebilir<sup>(48)</sup>. Denklemin daha formal bir hale sokulması adına;

$$x_1 = y_2, \quad x_2 = \theta_2L, \quad \frac{FL^3}{EI} = 1$$

kabul ve tanımları yapılarak aynı denklem;

$$f(\mathbf{x}) = 12x_1^2 + 4x_2^2 - 12x_1x_2 + 2x_1$$

şeklindeki ikilenik ifadeye dönüştürülebilir. Amaç fonksiyonunun ikilenik formda olması Powell yönteminin uygulanması için uygun bir zemin hazırlamaktadır. Çözüm adına yapılacak ilk işlem, başlangıç noktasının ve hassasiyet ölçütünün tespiti ile her bir eksendeki birim yöneylerin ifade edilmesidir. Başlangıç yöneyi  $\mathbf{x}_0^{(1)} = [-1 \quad -2]^T$  ve  $\epsilon = 0.3$  olarak dikkate alınabilir. Birim yöneyler;

$$\mathbf{d}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \mathbf{d}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

şeklinde ifade edilebilir. İlk eksen dâhilindeki yeni noktayı;

$$\mathbf{x}_1^{(1)} = \mathbf{x}_0^{(1)} + \lambda_1 \mathbf{d}_1$$

$$\mathbf{x}_1^{(1)} = \begin{bmatrix} -1 \\ -2 \end{bmatrix} + \lambda_1 \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} -1 + \lambda_1 \\ -2 \end{bmatrix}$$

olarak ifade edebiliriz.



Bilinen doğrultuya karşılık o doğrultudaki optimal adım büyüklüğünü, bu noktanın amaç fonksiyonunda yerine konması ve adım büyüklüğüne göre birinci mertebeden türevinin alınarak sıfıra eşitlenmesi yolu ile elde edebiliriz. Dolayısıyla;

$$f(\mathbf{x}_1^{(1)}) = 12(\lambda_1 - 1)^2 + 26(\lambda_1 - 1) + 16$$

$$\frac{\partial f(\mathbf{x}_1^{(1)})}{\partial \lambda_1} = 24\lambda_1^* + 2 = 0$$

$$\lambda_1^* = -0.083$$

olarak elde edilen optimal adım büyüklüğü dikkate alınarak yeni nokta ve amaç fonksiyonu değerini;

$$\mathbf{x}_1^{(1)} = \begin{bmatrix} -1 + \lambda_1^* \\ -2 \end{bmatrix} = \begin{bmatrix} -1.083 \\ -2 \end{bmatrix}$$

$$f(\mathbf{x}_1^{(1)}) = 1.9166$$

şeklinde hesaplayabiliriz. İkinci eksen dahilindeki yeni noktayı, benzer yaklaşımla;

$$\mathbf{x}_2^{(1)} = \mathbf{x}_1^{(1)} + \lambda_2 \mathbf{d}_2$$

$$\mathbf{x}_2^{(1)} = \begin{bmatrix} -1.083 \\ -2 \end{bmatrix} + \lambda_2 \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -1.083 \\ -2 + \lambda_2 \end{bmatrix}$$

olarak ifade edebiliriz. Optimal adım büyüklüğünü;

$$\frac{\partial f(\mathbf{x}_2^{(1)})}{\partial \lambda_2} = 8\lambda_2 - 3 = 0$$

$$\lambda_2 = 0.375$$

olarak dikkate alıp, yeni nokta ve amaç fonksiyonu değerini;

$$\mathbf{x}_2^{(1)} = \begin{bmatrix} -1.083 \\ -2 + \lambda_2 \end{bmatrix} = \begin{bmatrix} -1.083 \\ -1.625 \end{bmatrix}$$

$$f(\mathbf{x}_2^{(1)}) = 1.3541$$

biçiminde hesaplayabiliriz. Elde edilen amaç fonksiyon değerlerini dikkate alarak en büyük azalmanın;



$$f(\mathbf{x}_0^{(1)}) = 2.0000$$

$$f(\mathbf{x}_1^{(1)}) = 1.9166$$

$$f(\mathbf{x}_2^{(1)}) = 1.3541$$

değerleri çerçevesinde ikinci noktaya geçişte gerçekleştiği söylenebilir;

$$\Delta f = f(\mathbf{x}_1^{(1)}) - f(\mathbf{x}_2^{(1)}) = 0.5625$$

Dolayısıyla  $m = 2$  olarak kayıt altına alınabilir. Akabinde elde edilen son nokta ile ilk nokta dikkate alınarak eşlenik doğrultu hesaplanır;

$$\mathbf{d}_e^{(1)} = \mathbf{x}_2^{(1)} - \mathbf{x}_0^{(1)}$$

$$\mathbf{d}_e^{(1)} = \begin{bmatrix} -0.083 \\ 0.375 \end{bmatrix}$$

Bu doğrultu dikkate alınarak yeni döngünün başlangıç noktası;

$$\mathbf{x}_0^{(2)} = \mathbf{x}_0^{(1)} + \lambda \mathbf{d}_e^{(1)}$$

$$\mathbf{x}_0^{(2)} = \begin{bmatrix} -1 \\ -2 \end{bmatrix} + \lambda \begin{bmatrix} -0.083 \\ 0.375 \end{bmatrix} = \begin{bmatrix} -1 - 0.083\lambda \\ -2 + 0.375\lambda \end{bmatrix}$$

olarak ifade edilebilir. Bu doğrultudaki optimal adım büyüklüğünün eldesi amacıyla;

$$\frac{\partial f(\mathbf{x}_0^{(1)})}{\partial \lambda} = 49\lambda - 40 = 0$$

$$\lambda^* = 0.816$$

işlemleri yapılır ve yerine koyulursa;

$$\mathbf{x}_0^{(2)} = \begin{bmatrix} -1.068 \\ -1.693 \end{bmatrix}$$

$$f(\mathbf{x}_0^{(2)}) = 1.3197$$

noktasına ve amaç fonksiyonu değerine ulaşılır. Başlangıçtaki nokta ile elde edilen son nokta arasındaki fark dikkate alınarak yakınsama kontrolü yapılırsa;



$$\| \mathbf{x}_0^{(2)} - \mathbf{x}_0^{(1)} \| = \left\| \begin{bmatrix} -1.068 \\ -1.693 \end{bmatrix} - \begin{bmatrix} -1 \\ -2 \end{bmatrix} \right\| = 0.3134 > 0.3$$

sonucu gözlemlenir ve bir sonraki döngüye geçilir. Döngünün başlangıcında, doğrultu değişikliği yapıp yapılmayacağı belirlenir. Bu amaçla daha önce ifade edilen ölçüt dikkate alınarak;

$$|\lambda^*| = 0.816 < \frac{\left[ f(\mathbf{x}_0^{(1)}) - f(\mathbf{x}_0^{(2)}) \right]^{1/2}}{\left[ f(\mathbf{x}_1^{(1)}) - f(\mathbf{x}_2^{(1)}) \right]} = 1.0997$$

eşitsizliğinin halen geçerli olduğu görülür. Dolayısıyla ikinci döngüde herhangi bir doğrultu değişikliğine gidilmesine gerek yoktur. İkinci döngüde;

$$\mathbf{d}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \mathbf{d}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

doğrultuları dikkate alınarak ilk eksen dahilindeki yeni noktayı;

$$\begin{aligned} \mathbf{x}_1^{(2)} &= \mathbf{x}_0^{(2)} + \lambda_1 \mathbf{d}_1 \\ \mathbf{x}_1^{(2)} &= \begin{bmatrix} -1.068 \\ -1.693 \end{bmatrix} + \lambda_1 \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} -1.068 + \lambda_1 \\ -1.693 \end{bmatrix} \end{aligned}$$

olarak ifade edebiliriz. Optimal adım büyüklüğünü;

$$\begin{aligned} \frac{\partial f(\mathbf{x}_1^{(2)})}{\partial \lambda_1} &= 24\lambda_1^* - 3.312 = 0 \\ \lambda_1^* &= 0.138 \end{aligned}$$

şeklinde hesaplayıp yeni nokta ve amaç fonksiyonu değerini;

$$\begin{aligned} \mathbf{x}_1^{(2)} &= \begin{bmatrix} -1.068 + \lambda_1 \\ -1.693 \end{bmatrix} = \begin{bmatrix} -0.93 \\ -1.693 \end{bmatrix} \\ f(\mathbf{x}_1^{(2)}) &= 1.0918 \end{aligned}$$

şeklinde elde edebiliriz. İkinci eksen dâhilindeki yeni noktayı ise;

$$\mathbf{x}_2^{(2)} = \mathbf{x}_1^{(2)} + \lambda_2 \mathbf{d}_2$$



$$\mathbf{x}_2^{(2)} = \begin{bmatrix} -0.93 \\ -1.6938 \end{bmatrix} + \lambda_2 \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -0.93 \\ \lambda_2 - 1.6938 \end{bmatrix}$$

olarak ifade edebiliriz. Bu doğrultudaki optimal adım büyüklüğünü;

$$\frac{\partial f(\mathbf{x}_2^{(2)})}{\partial \lambda_2} = 8\lambda_2 - 2.3904 = 0$$

$$\lambda_2^* = 0.2988$$

olarak dikkate alıp, yeni nokta ve amaç fonksiyonu değerini;

$$\mathbf{x}_2^{(2)} = \begin{bmatrix} -0.93 \\ \lambda_2^* - 1.6938 \end{bmatrix} = \begin{bmatrix} -0.93 \\ -1.395 \end{bmatrix}$$

$$f(\mathbf{x}_2^{(2)}) = 0.7347$$

biçiminde hesaplayabiliriz. Elde edilen amaç fonksiyon değerlerini dikkate alarak en büyük azalmanın;

$$f(\mathbf{x}_0^{(2)}) = 1.3197$$

$$f(\mathbf{x}_1^{(2)}) = 1.0918$$

$$f(\mathbf{x}_2^{(2)}) = 0.7347$$

değerleri çerçevesinde ikinci noktaya geçişte gerçekleştiği söylenebilir;

$$\Delta f = f(\mathbf{x}_1^{(2)}) - f(\mathbf{x}_2^{(2)}) = 0.3569$$

Dolayısıyla  $m = 2$  olarak kayıt altına alınabilir. Akabinde elde edilen son nokta ile ilk nokta dikkate alınarak eşlenik doğrultu hesaplanır;

$$\mathbf{d}_e^{(2)} = \mathbf{x}_2^{(2)} - \mathbf{x}_0^{(2)}$$

$$\mathbf{d}_e^{(2)} = \begin{bmatrix} 0.1380 \\ 0.2988 \end{bmatrix}$$

Bu doğrultu dikkate alınarak yeni döngünün başlangıç noktası;





$$\mathbf{x}_0^{(3)} = \mathbf{x}_0^{(2)} + \lambda \mathbf{d}_e^{(2)}$$

$$\mathbf{x}_0^{(3)} = \begin{bmatrix} -1.068 \\ -1.693 \end{bmatrix} + \lambda \begin{bmatrix} 0.1380 \\ 0.2988 \end{bmatrix} = \begin{bmatrix} -1.068 + 0.1380\lambda \\ -1.693 + 0.2988\lambda \end{bmatrix}$$

olarak ifade edilebilir. Bu doğrultudaki optimal adım büyüklüğünün eldesi amacıyla;

$$\frac{\partial f(\mathbf{x}_0^{(3)})}{\partial \lambda} = 0.1816\lambda - 0.6749 = 0$$

$$\lambda^* = 3.7164$$

işlemleri yapılır ve yerine koyulursa;

$$\mathbf{x}_0^{(3)} = \begin{bmatrix} -0.555 \\ -0.583 \end{bmatrix}$$

$$f(\mathbf{x}_0^{(3)}) = 0.063$$

değerlerine ulaşılır. Başlangıçtaki nokta ile elde edilen son nokta arasındaki fark dikkate alınarak yakınsama kontrolü yapılırsa;

$$\|\mathbf{x}_0^{(3)} - \mathbf{x}_0^{(2)}\| = \left\| \begin{bmatrix} -0.555 \\ -0.583 \end{bmatrix} - \begin{bmatrix} -1.068 \\ -1.693 \end{bmatrix} \right\| = 1.22 > 0.3$$

sonucu gözlemlenir ve bir sonraki döngüye geçilir. Doğrultu değişikliği yapıp yapılmayacağını belirlemek amacıyla adım büyüklüğü kontrol edilirse;

$$|\lambda^*| = 3.7164 > \left[ \frac{f(\mathbf{x}_0^{(2)}) - f(\mathbf{x}_0^{(3)})}{f(\mathbf{x}_1^{(2)}) - f(\mathbf{x}_2^{(2)})} \right]^{1/2} = 3.519$$

olduğu görülür. Dolayısıyla ikinci döngüde doğrultu değişikliğine gitmek gerekmektedir. Daha önce azami azalma sağlayan yön dâhilinde  $m = 2$  değerini kayıt altına almıştık. Doğrultu değişikliğini, bu alt indisi dikkate alarak yapabiliriz;

$$\mathbf{d}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$\mathbf{d}_2 = \mathbf{d}_e^{(2)} = \begin{bmatrix} 0.1380 \\ 0.2988 \end{bmatrix}$$



Müteakiben yeni doğrultuları dikkate alarak ilk doğrultu dâhilindeki yeni noktayı;

$$\mathbf{x}_1^{(3)} = \mathbf{x}_0^{(3)} + \lambda_1 \mathbf{d}_1$$

$$\mathbf{x}_1^{(3)} = \begin{bmatrix} -0.555 \\ -0.583 \end{bmatrix} + \lambda_1 \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} -0.555 + \lambda_1 \\ -0.583 \end{bmatrix}$$

biçiminde ifade edebiliriz. Optimal adım büyüklüğünü;

$$\frac{\partial f(\mathbf{x}_1^{(3)})}{\partial \lambda_1} = 24\lambda_1^* - 4.324 = 0$$

$$\lambda_1^* = 0.18$$

şeklinde hesaplayıp yeni nokta ve amaç fonksiyonu değerini;

$$\mathbf{x}_1^{(3)} = \begin{bmatrix} -0.555 + \lambda_1 \\ -0.583 \end{bmatrix} = \begin{bmatrix} -0.375 \\ -0.583 \end{bmatrix}$$

$$f(\mathbf{x}_1^{(3)}) = -0.3264$$

şeklinde elde edebiliriz. İkinci eksen dâhilindeki yeni noktayı ise;

$$\mathbf{x}_2^{(3)} = \mathbf{x}_1^{(3)} + \lambda_2 \mathbf{d}_2$$

$$\mathbf{x}_2^{(3)} = \begin{bmatrix} -0.375 \\ -0.583 \end{bmatrix} + \lambda_2 \begin{bmatrix} 0.1380 \\ 0.2988 \end{bmatrix} = \begin{bmatrix} -0.375 + 0.138\lambda_2 \\ -0.583 + 0.2988\lambda_2 \end{bmatrix}$$

olarak ifade edebiliriz. Bu doğrultudaki optimal adım büyüklüğünü;

$$\frac{\partial f(\mathbf{x}_2^{(3)})}{\partial \lambda_2} = 0.1818\lambda_2 - 0.0494 = 0$$

$$\lambda_2^* = 0.271$$

şeklinde hesaplayıp, yeni nokta ve amaç fonksiyonu değerini;

$$\mathbf{x}_2^{(3)} = \begin{bmatrix} -0.375 + 0.138\lambda_2 \\ -0.583 + 0.2988\lambda_2 \end{bmatrix} = \begin{bmatrix} -0.337 \\ -0.502 \end{bmatrix}$$

$$f(\mathbf{x}_2^{(3)}) = -0.598$$

olarak elde ederiz. Hesaplanan amaç fonksiyon değerlerini dikkate alarak en büyük azalmanın;



$$\begin{aligned} f(\mathbf{x}_0^{(3)}) &= 0.063 \\ f(\mathbf{x}_1^{(3)}) &= -0.3264 \\ f(\mathbf{x}_2^{(3)}) &= -0.598 \end{aligned}$$

değerlerine göre, önceki döngülerden farklı olarak ilk noktaya geçişte gerçekleştiği söylenebilir;

$$\Delta f = f(\mathbf{x}_0^{(3)}) - f(\mathbf{x}_1^{(3)}) = 0.3894$$

Dolayısıyla  $m = 1$  olarak kayıt altına alınabilir. Müteakiben elde edilen son nokta ile ilk nokta dikkate alınarak eşlenik doğrultu hesaplanır;

$$\begin{aligned} \mathbf{d}_e^{(3)} &= \mathbf{x}_2^{(3)} - \mathbf{x}_0^{(3)} \\ \mathbf{d}_e^{(3)} &= \begin{bmatrix} 0.218 \\ 0.081 \end{bmatrix} \end{aligned}$$

Bu doğrultu dikkate alınarak yeni döngünün başlangıç noktası;

$$\mathbf{x}_0^{(4)} = \begin{bmatrix} -0.555 \\ -0.583 \end{bmatrix} + \lambda \begin{bmatrix} 0.218 \\ 0.081 \end{bmatrix} = \begin{bmatrix} -0.555 + 0.218\lambda \\ -0.583 + 0.081\lambda \end{bmatrix}$$

şeklinde ifade edilebilir. Bu doğrultudaki optimal adım büyüklüğünün eldesi amacıyla;

$$\begin{aligned} \frac{\partial f(\mathbf{x}_0^{(4)})}{\partial \lambda} &= 0.7692\lambda - 0.7809 = 0 \\ \lambda^* &= 1.015 \end{aligned}$$

işlemleri yapılır ve yeni adım büyüklüğü yerine koyulursa;

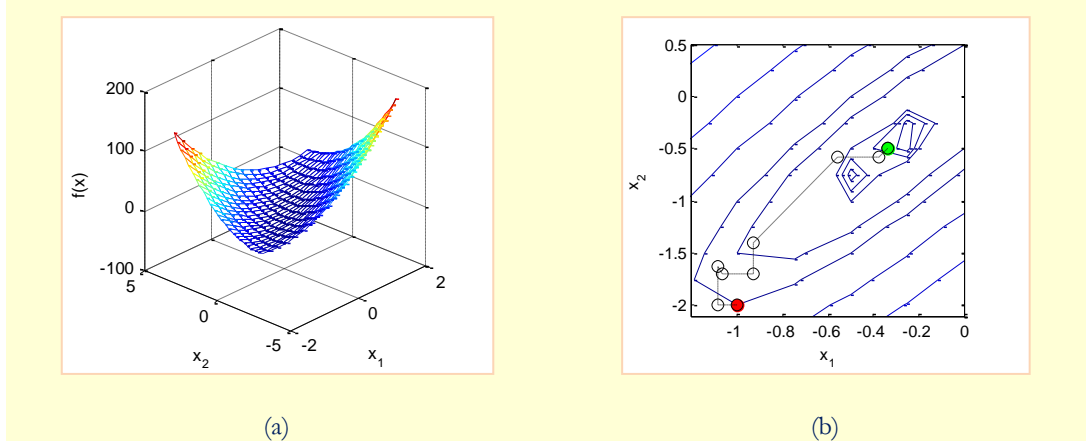
$$\mathbf{x}_0^{(4)} = \begin{bmatrix} -0.333 \\ -0.500 \end{bmatrix}, \quad f(\mathbf{x}_0^{(4)}) = -0.334$$

değerlerine ulaşılır. Başlangıçtaki nokta ile elde edilen son arasındaki fark dikkate alınarak yakınsama kontrolü yapılırsa;

$$\|\mathbf{x}_0^{(4)} - \mathbf{x}_0^{(3)}\| = \left\| \begin{bmatrix} -0.333 \\ -0.500 \end{bmatrix} - \begin{bmatrix} -0.555 \\ -0.583 \end{bmatrix} \right\| = 0.287 < 0.3$$



sonucu gözlemlenir.



**Şekil 2.68** Amaç fonksiyonunun 3B görüntüsü (solda), amaç fonksiyonu kesit değerleri ve taranan noktalar (başlangıç noktası  $\circ$  ve optimal nokta  $\bullet$ ).

Elde edilen bu eşitsizlik, istenen hassasiyet değerindeki çözüm noktasına ulaşıldığını göstermektedir. Dolayısıyla tarama işlemi sonucunda en uygun noktaya varılmış olup işlemler sonlandırılabilir. Buna göre azami eğilme değeri aşağı yönde  $0.333 br$  mesafede olmakta, aynı noktadaki eğilme değeri de saatin dönüş yönünde  $0.5/L$  açıda gerçekleşmektedir. Sorunun analitik çözümü dikkate alındığında söz konusu azami eğilme değerinin  $FL^3/(3EI)$  olduğu<sup>(49)</sup>, başlangıçta dikkate alınan  $FL^3/(EI) = 1$  kabulü nedeniyle elde edilen  $0.333$  sayısal değerinin, analitik olarak elde edilen  $1/3$  değeriyle gayet iyi bir şekilde örtüştüğü görülmektedir. Şekil 2.68 dahilinde amaç fonksiyonu yüzeyi üzerindeki her adımda varılan nokta ve ulaşılan optimal konum görülmektedir. Diğer taraftan Powell yöntemine göre elde edilen  $d_e^{(k)}$  yöneylerinin,  $A$ -eşlenik yöney özelliği taşıdığını, son iki yöneyi dikkate alarak gözlemek mümkündür. Amaç fonksiyonuna ait katsayıların dikkate alınması neticesinde aynı fonksiyonu ikilenik formda;

$$f(x) = \frac{1}{2} [x_1 \ x_2]^T \begin{bmatrix} 24 & -12 \\ -12 & 8 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + [2 \ 0]^T \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

şeklinde yazabiliriz. Bu denklemdeki  $A$  matrisini, son iki eşlenik doğrultu yöneyi ile beraber dikkate aldığımızda;

$$\mathbf{d}_e^{(k)T} A \mathbf{d}_e^{(k+1)}$$

$$\mathbf{d}_e^{(2)T} A \mathbf{d}_e^{(3)} = [0.138 \quad 0.2988] \begin{bmatrix} 24 & -12 \\ -12 & 8 \end{bmatrix} \begin{bmatrix} 0.218 \\ 0.081 \end{bmatrix} \cong -0.0001$$

sonucuna ulaşabiliriz. Elde edilen bu değer yaklaşık olarak 0 olup eşlenik doğrultuların  $A$  –eşlenik yöney özelliğini teyit etmektedir. “Powell” yöntemi her ne kadar türev verisinden bağımsız bir yöntem<sup>i</sup> gibi görünse de bünyesinde tek değişkenli tarama işlemlerini barındırmaktadır. Dolayısıyla ikincil nitelikteki söz konusu taramanın, etkin bir şekilde yapılması istendiğinde, tek boyutlu da olsa gradyan bilgisine başvurulmakta ve tamamen sayısal ve türevsiz bir yöntem olma özelliği bu meyanda zaafa uğramaktadır. Ancak türev bilgisine ihtiyaç duyulmayan eleme bazlı yöntemlerin dikkate alınması halinde “Powell” yönteminin tamamen amaç fonksiyonu değerine bağlı bir yöntem olduğu söylenebilir.

## (2) “Endik azalma” yöntemi

Endik azalma<sup>ii</sup> ve müteakip paragraflarda ifade edilen yöntemlerin temeli, değişik mertebelerdeki türev verisine dayanır. Daha önceki bölümlerde gradyan kavramı tanımlanmış, onun yöneysel bir büyüklük olduğu ve fonksiyonun en fazla artış gösterdiği yönü içerdiği ifade edilmişti. 1847 tarihli makalesi ile M. A. Cauchy tarafından geliştirilen endik azalma yöntemi de esasen gradyan verisine dayanır<sup>(50)</sup>. Yöntemin işlem adımlarına geçmeden önce gradyan yönünü içeren ve eniyileme süreçlerinde oldukça önemli bir yere sahip olan müteakip önermeyi irdelemekte fayda görülmektedir;

- ▶ Fonksiyona ait gradyan yöneyi, o fonksiyonun değerinin azami artış gösterdiği yönedir.

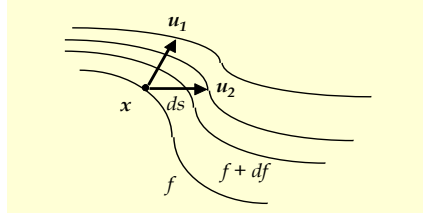
Söz konusu önermenin doğruluğu, basit bir yaklaşımla izah edilebilir. Buna göre Şekil 2.69’da belirtildiği üzere başlangıç noktası olarak  $\mathbf{x}$  noktasını, bir sonraki nokta olarak dikkate alınacak konuma olan uzaklığı  $ds$  ve gidilecek yönü de  $\mathbf{u}_i$  olarak dikkate alalım. Konuma göre amaç fonksiyon değeri değişecek olup aynı şekil dâhilinde amaç fonksiyonu eşdeğer çizgileri de görülmektedir. Buna göre bulunulan  $\mathbf{x}$  noktasından  $\mathbf{u}_1$  istikametinde  $ds$  kadar mesafe gidildiğinde

<sup>i</sup> Derivative-free method

<sup>ii</sup> Steepest descend



elde edilen amaç fonksiyon değeri, aynı mesafe kat edilerek  $\mathbf{u}_2$  istikametinde gidildiğinde elde edilecek amaç fonksiyon değerinden farklı olacaktır. Dolayısıyla adım büyüklüğü aynı olsa dahi amaç fonksiyon değeri açısından adımın atılacağı yön büyük önem arz etmektedir.



Şekil 2.69 Başlangıç noktası ve yön tanımlaması.

Başlangıç noktası ile ulaşılan nokta arasındaki yer değiştirme yöneyini;

$$dx = ds \mathbf{u} \quad (2.111)$$

denklemi ile ifade edebiliriz. Burada  $\mathbf{u}$  yöneyi, gidilecek yöndeki birim yöneyi ifade etmektedir. Diğer taraftan atılan adım sonrasında amaç fonksiyonu değerindeki değişimi;

$$df = \sum_{i=1}^n \frac{\partial f}{\partial x_i} dx_i = \nabla f^T \cdot dx \quad (2.112)$$

biçiminde yazabiliriz. Amaç fonksiyonundaki değişimin  $ds$  adım büyüklüğüne göre nasıl değiştiğini ise;

$$\frac{df}{ds} = \frac{\nabla f^T \cdot dx}{ds} \quad (2.113)$$

oranı ile temsil edebiliriz. Bu ifadede yer değiştirme yöneyinin tanımı dikkate alınır;

$$\frac{df}{ds} = \frac{\nabla f^T \cdot ds \mathbf{u}}{ds} = \nabla f^T \cdot \mathbf{u} \quad (2.114)$$

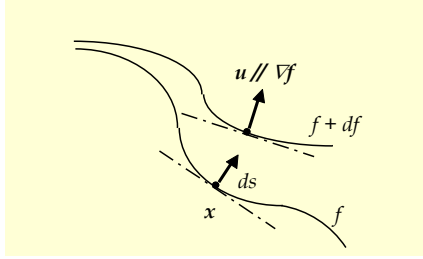
sonucuna ulaşılır. Görüldüğü üzere denklem, sayısal çarpım işlemi içermekte olup yöney şiddetleri ve aradaki açı dikkate alınarak;





$$\nabla f^T \cdot \mathbf{u} = |\nabla f^T| \cdot |\mathbf{u}| \cdot \cos \theta \quad (2.115)$$

denkleme dönüşürülebilir.



Şekil 2.70 Başlangıç noktası ve gradyan yönü.

Amaç fonksiyon değerinin, atılan  $ds$  adımı sonrasında azami değişiminin sağlanabilmesi için bu çarpımın maksimum değeri alması gerekir. Dolayısıyla iki yöney arasındaki açı değerinin  $\theta = 0$  olması, değişimi doruklaştıracaktır. Bunun anlamı, gidilen yön ile gradyanın birbirlerine paralel olmasıdır. Hülusa, amaç fonksiyonunda azami değişimin elde edilebilmesi için atılacak adımın, gradyan doğrultusunda olması gerekir. Şekil 2.70'ten de görüleceği üzere gradyan doğrultusu, bulunulan noktaya teğet olan doğrultu ile dikeygenlik arz etmektedir. Endik azalma yönteminin işlem adımlarını aşağıdaki gibi sıralayabiliriz;

1. Öncelikle mevcut tecrübe dâhilinde en iyi başlangıç noktası tahmini,  $\mathbf{x}_0$  yapılır.
2. Akabinde gidilecek yönün tayin edilmesi için amaç fonksiyonunun  $\mathbf{x}_0$  noktasındaki gradyanı elde edilir. Bulunan bu değer aksistikameti, gidilecek yön olarak tayin edilir. Aksistikametin seçilmesinin nedeni, minimizasyon süreci dâhilinde amaç fonksiyonundaki azami azalmanın hedeflenmesidir;

$$\mathbf{d}_i = -\nabla f(\mathbf{x}_i), \quad i = 0 \quad (2.116)$$

3. Gidilecek yönün belirlenmesi sonrasında atılacak adım büyüklüğünün tespiti gerekir. Daha önceki bölümlerde söz konusu tek değişkenli tarama sorunları üzerinde durulmuş ve değişik yöntemlerle  $\alpha^*$  optimal adım büyüklüğünün belirlenebileceği ifade edilmişti. Dolayısıyla en uygun adım büyüklüğünün eldesini müteakip yeni konum;

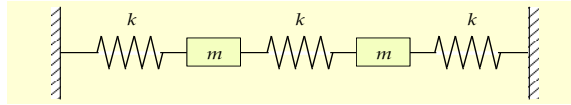


$$\begin{aligned} \mathbf{x}_{i+1} &= \mathbf{x}_i + \alpha^* \mathbf{d} \\ \mathbf{x}_{i+1} &= \mathbf{x}_i - \alpha^* \nabla f(\mathbf{x}_i) \end{aligned} \quad (2.117)$$

ifadesi ile belirlenir. Bununla beraber yaklaşımın basitleştirilmesi adına uygun mertebede ama sabit bir değerin adım büyüklüğü olarak dikkate alınması da mümkündür. Yeni konumun eldesi sonrasında istenilen hassasiyette amaç fonksiyon değerinin elde edilip edilmediği kontrol edilir. Bu meyanda daha önceki bölümlerde değişik yakınsama ölçütleri ifade edilmişti. Yakınsama kontrolü için tasarım yöneyi değişimi baz alındığında;

$$|\mathbf{x}_{i+1} - \mathbf{x}_i| \leq \varepsilon \quad (2.118)$$

ölçütü dikkate alınabilir. Burada  $\varepsilon$  değeri hassasiyet mertebesidir. İstenilen hassasiyet mertebesine ulaşılması durumunda tarama işlemi sonlandırılır. Aksi halde 2nci işlem maddesine gidilerek tekrarlı işlemlere devam edilir.



Şekil 2.71 Kütle-yay düzeneği.

“Endik azalma” yönteminin daha iyi anlaşılabilmesi için Şekil 2.71’deki gibi basit bir mekanik düzenek sorununu, belirtilen işlem adımlarını takip ederek çözebiliriz. Buna göre üç yay ve iki kütleli hareketli sistemin doğal frekansını;

$$f(\omega) = \frac{m^2}{5} \omega^5 - \frac{4km}{3} \omega^3 + 3k^2 \omega$$

şeklinde tanımlanan denklemin asgari değerini elde ederek bulmak mümkündür<sup>(51)</sup>. Burada  $m$  kütle değerini,  $k$  yay katsayısını ve  $\omega$  doğal frekansı ifade etmektedir. Sorunun tanımlanması çerçevesinde  $m = 1 \text{ kg}$  ve  $k = 2 \text{ N/m}$  olarak verilmektedir. Yöntemin uygulanması kapsamında başlangıç noktasını  $\omega_0 = 1.5 \text{ Hz}$  ve ulaşılması istenilen hassasiyet değerini  $\varepsilon = 0.001$  olarak dikkate alabiliriz. Akabinde gidilecek yönün tayin edilmesi için amaç fonksiyonunun  $\omega_0$  noktasındaki gradyanı elde edilir;

$$\begin{aligned} \nabla f(\omega) &= m^2 \omega^4 - 4km\omega^2 + 3k^2 \\ d_0 &= -\nabla f(\omega_0) = 1^2(1.5)^4 - 4.2.1.(1.5)^2 + 3.2^2 \\ d_0 &= 0.9375 \end{aligned}$$







Her ne kadar çözümün eldesinde adım büyüklüğünün etkisi büyük de olsa, çözümün basitleştirilmesi adına adım büyüklüğünü sabit kabul edip,  $\alpha = 0.01$  değerini dikkate alabiliriz. Buna göre yeni noktamız;

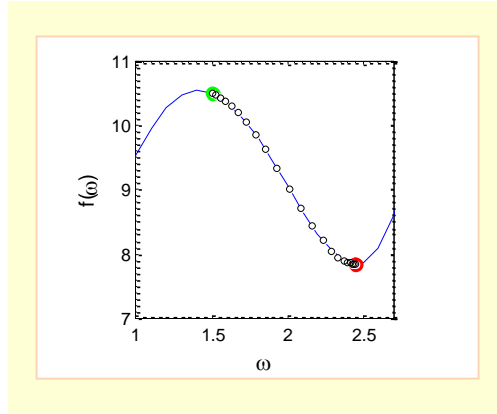
$$\begin{aligned}\omega_1 &= \omega_0 + \alpha d_0 = 1.5 + 0.01 \times 0.9375 \\ \omega_1 &= 1.5094\end{aligned}$$

olarak belirlenir. Yakınsama kontrolü için tasarım yöneyi değerleri baz alındığında;

$$|\omega_1 - \omega_0| = |1.5094 - 1.5| = 0.0094 > \varepsilon = 0.001$$

eşitsizliği gözlemlenebilir. Dolayısıyla bir sonraki adıma geçmek için yeni doğrultu hesaplanır;

$$\begin{aligned}d_1 &= -\nabla f(\omega_1) = 1^2 1.5094^4 - 4.2.1. (1.5094)^2 + 3.2^2 \\ d_1 &= 1.0354\end{aligned}$$



**Şekil 2.72** Tarama süreci kapsamında başlangıç (○), tarama adımları ve optimal (○) nokta.

Bu doğrultuda yeni konum;

$$\begin{aligned}\omega_2 &= \omega_1 + \alpha d_1 = 1.5094 - 0.001 \times 19.7602 \\ \omega_2 &= 1.5197\end{aligned}$$

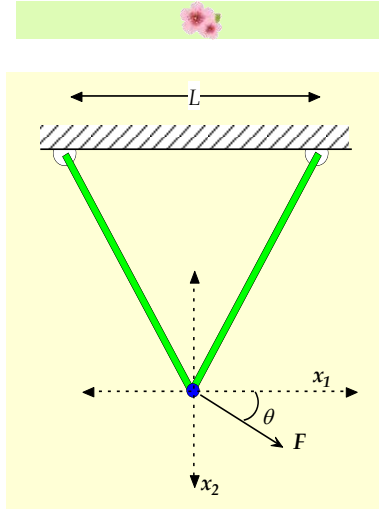
olarak belirlenir. Yakınsama kontrolü yapıldığında;





$$|\omega_2 - \omega_1| = |1.5197 - 1.5094| = 0.0104 > \varepsilon = 0.001$$

eşitsizliği nedeniyle sonraki adımlara geçilir. Şekil 2.72 dâhilinde, istenen hassasiyete ulaşılması amacıyla taranan noktalar ve nihayetinde ulaşılan en uygun nokta görülmektedir. Buna göre kütle-yay düzeneğinin doğal frekanslarından birisi  $\omega = 2.4461$  olup bu sonuç analitik çözümlerden biri olan  $\sqrt{3k/m} = 2.4495$  sonucu ile örtüşmektedir.



Örnek uygulama

Şekil 2.73 İki-kiriş düzeneği ve uç noktada oluşan yer değiştirme mesafelerinin tespiti.

Kenarları ve mafsal aralığı eşit mesafede olan Şekil 2.73'deki gibi iki-kirişli bir düzeneği dikkate alalım. Böyle bir yapının potansiyel enerji denklemi;

$$f(x) = \frac{EA}{4L} x_1^2 + \frac{3EA}{4L} x_2^2 - Fx_1 \cos \theta - Fx_2 \sin \theta$$

şeklinde yazılabilir. Burada  $E$  elastisite modülü,  $A$  kirişe ait kesit alan değeri,  $L$  mesafe değeri,  $F$  uç noktaya  $\theta$  açısı ile etki eden kuvvet,  $x_1$  ve  $x_2$  uç noktanın yer değiştirme miktarlarıdır. Uygulanan kuvvet nedeniyle kirişlerin birleşim noktasında meydana gelen yer değiştirme değerlerini, potansiyel enerji denkleminin

<sup>i</sup> Two-bar truss



minimizasyonu yoluyla elde etmek mümkündür. Denklemdaki katsayı değerlerini;  $E = 200 \times 10^9 Pa$ ,  $A = 10^{-5} m^2$ ,  $L = 5 m$ ,  $F = 10000 N$  ve  $\theta = 30^\circ$  olarak dikkate aldığımızda aynı denklemi;

$$f(\mathbf{x}) = 10^5 x_1^2 + 3 \times 10^5 x_2^2 - 8660 x_1 - 5000 x_2$$

şeklinde ifade edebiliriz. Çözüm dâhilinde başlangıç noktası olarak  $\mathbf{x}_0^T = [0 \ 0]$  ve yakınsama ölçütü olarak  $\epsilon = 0.3$  değerini dikkate alalım. Endik azalma yöntemi gereği öncelikle gidilecek yönün tayin edilmesi için amaç fonksiyonunun  $\mathbf{x}_0$  noktasındaki gradyanı elde edilir;

$$\nabla f(\mathbf{x}_i) = \begin{bmatrix} \partial f / \partial x_{i,1} \\ \partial f / \partial x_{i,2} \end{bmatrix} = \begin{bmatrix} 2 \times 10^5 x_{i,1} - 8660 \\ 6 \times 10^5 x_{i,2} - 5000 \end{bmatrix}$$

$$\mathbf{d}_0 = -\nabla f(\mathbf{x}_0) = \begin{bmatrix} 8660 \\ 5000 \end{bmatrix}$$

Buna göre yeni nokta;

$$\begin{aligned} \mathbf{x}_1 &= \mathbf{x}_0 + \alpha_0 \mathbf{d}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \alpha_0 \times \begin{bmatrix} 8660 \\ 5000 \end{bmatrix} \\ \mathbf{x}_1 &= \begin{bmatrix} 8660 \alpha_0 \\ 5000 \alpha_0 \end{bmatrix} \end{aligned}$$

olarak belirlenir. Optimal adım büyüklüğünün elde edilmesi kapsamında, adım büyüklüğü değerine bağlı  $\mathbf{x}_1$  değeri, amaç fonksiyonunda yerine konulduğunda;

$$\begin{aligned} f(\mathbf{x}_1) &= 10^5 x_{1,1}^2 + 3 \times 10^5 x_{1,2}^2 - 8660 x_{1,1} - 5000 x_{1,2} \\ f(\alpha) &= 14.9996 \times 10^{12} \alpha_0^2 - 9.99956 \times 10^7 \alpha_0 \end{aligned}$$

ifadesi elde edilebilir ve en uygun adım büyüklüğü;

$$\begin{aligned} \frac{\partial f}{\partial \alpha_0} &= 2.99992 \times 10^{13} \alpha_0 - 9.99956 \times 10^7 = 0 \\ \alpha_0 &= 3.3333 \times 10^{-6} \end{aligned}$$

şeklinde bulunur. Bulunan adım büyüklüğü, yeni noktada dikkate alındığında;

$$\mathbf{x}_1 = \begin{bmatrix} 0.0289 \\ 0.0167 \end{bmatrix}$$

elde edilir. Yakınsama kontrolü için amaç fonksiyon değerleri baz alındığında;



$$|f(\mathbf{x}_1) - f(\mathbf{x}_0)| = |-166.5860| = 166.5860 > \varepsilon = 0.3$$

eşitsizliği gözlemlenebilir. Dolayısıyla bir sonraki adıma geçmek için yeni doğrultu hesaplanır;

$$\mathbf{d}_1 = -\nabla f(\mathbf{x}_1) = - \begin{bmatrix} 2 \times 10^5 x_{1,1} - 8660 \\ 6 \times 10^5 x_{1,2} - 5000 \end{bmatrix} = - \begin{bmatrix} 2 \times 10^5 \times 0.0289 - 8660 \\ 6 \times 10^5 \times 0.0167 - 5000 \end{bmatrix}$$

$$\mathbf{d}_1 = \begin{bmatrix} -2880 \\ 5020 \end{bmatrix}$$

Belirlenen doğrultuda yeni konum;

$$\mathbf{x}_2 = \mathbf{x}_1 + \alpha \mathbf{d}_1 = \begin{bmatrix} 0.0289 \\ 0.0167 \end{bmatrix} + \alpha \times \begin{bmatrix} -2880 \\ 5020 \end{bmatrix}$$

$$\mathbf{x}_2 = \begin{bmatrix} 0.0289 - 2880\alpha \\ 0.0167 + 5020\alpha \end{bmatrix}$$

olarak elde edilir. Optimal adım büyüklüğünün elde edilmesi kapsamında, adım büyüklüğü değerine bağlı  $\mathbf{x}_2$  değeri amaç fonksiyonunda yerinde kullanılarak en uygun adım büyüklüğü;

$$\frac{\partial f}{\partial \alpha_1} = 1.6779 \times 10^{13} \alpha_1 + 3.34948 \times 10^7 = 0$$

$$\alpha_1 = -1.9962 \times 10^{-6}$$

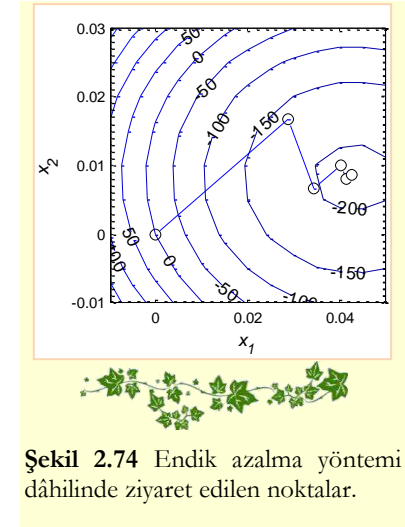
şeklinde elde edilebilir. Bulunan yeni adım büyüklüğü bir sonraki nokta tanımında dikkate alındığında;

$$\mathbf{x}_2 = \begin{bmatrix} 0.0346 \\ 0.0067 \end{bmatrix}$$

değerine ulaşılır. Yakınsama kontrolü için amaç fonksiyon değerleri baz alındığında;

$$|f(\mathbf{x}_2) - f(\mathbf{x}_1)| = |-199.9530 + 166.5860| = 33.3670 > \varepsilon = 0.3$$

eşitsizliği gözlemlenebilir. Dolayısıyla bir sonraki adıma geçmek için yeni doğrultunun hesaplanması gerekir. Müteakip çizelgede her bir adımda ulaşılan nokta, gidilecek doğrultu ve yakınsama ölçüt değeri, Şekil 2.74'te ise ziyaret edilen noktalar görülmektedir.



Şekil 2.74 Endik azalma yöntemi dâhilinde ziyaret edilen noktalar.





$d_i$	$\alpha_i$	$x_{i+1}$	$ f(x_{i+1}) - f(x_i)  < \varepsilon$
[8660; 5000]	$3.3333 \times 10^{-6}$	[0.0289; 0.0167]	166.586
[-2880; 5020]	$-1.9962 \times 10^{-6}$	[0.0346; 0.0067]	33.367
[1740; 980]	$3.3746 \times 10^{-6}$	[0.0405; 0.0100]	6.7520
[560; -1000]	$1.9821 \times 10^{-6}$	[0.0416; 0.0080]	1.2950
[340; 200]	$3.3022 \times 10^{-6}$	[0.0427; 0.0087]	0.2460

Buna göre söz konusu iki-kiriş düzeneğinin uç noktasında meydana gelen yer değiştirme değerleri  $x = [0.0427 \ 0.0087] m$  olup minimum potansiyel enerji değeri  $208.2460 Nm$  olarak elde edilmektedir. Bulunan değerler, birinci mertebeden türev alınarak elde edilebilecek analitik değer ile örtüşmektedir.



**MATLAB uygulaması**

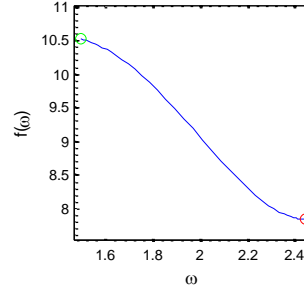
```

clc; clear;
m = 1; k = 2; w(1) = 1.5;
eps = 0.001; alp = 0.01; i = 1;
d = -( m^2*w(1)^4-4*k*m*w(1)^2+3*k^2);
w(i+1) = w(i) + alp * d;
f = 1/5*m^2*w.^5-4/3*k*m*w.^3+3*w*k^2;
yak = abs((w(i+1)-w(i)));

while yak > eps; i = i +1;
    d = -( m^2*w(i)^4-
    4*k*m*w(i)^2+3*k^2);
    w(i+1) = w(i) + alp * d;
    f(i+1) = 1/5*m^2*w(i+1).^5-
    4/3*k*m*w(i+1).^3+3*w(i+1)*k^2;
    yak = abs((w(i+1)-w(i)));
end ;

plot(w,f); hold on;
plot(w(1),f(1), 'go',w(i+1),f(i+1), 'ro')

```





### (3) Eşlenik gradyan yöntemi

Eşlenik gradyan<sup>i</sup> yöntemi, 1950'li yıllarda doğrusal denklem takımının tekrarlı yöntemlerle çözülmesi için geliştirilmiş bir yöntemdir. Hatırlanacağı üzere doğrusal denklem takımı,  $Ax = b$  denklemi ile ifade edilmekteydi. Burada eşitliği sağlayan  $x$  yöneyi çözüm yöneyidir. Sorun her ne kadar doğrusal denklem takımı sorunu gibi görünse de ikilenik formdaki bir denklemin optimal nokta çözümü ile de ilişkilidir. Herhangi bir ikilenik denklem;

$$f(x) = \frac{1}{2}x^T Ax - x^T b + c \quad (2.119)$$

şeklinde ifade edilebilir. Tanımlı denklemin optimal çözümünü analitik yöntemle, diğer bir ifade ile birinci mertebeden türev ifadesinin sıfıra eşitlenmesi ile elde etmek mümkündür;

$$\begin{aligned} \nabla f(x) &= Ax - b = 0 \\ Ax &= b \end{aligned} \quad (2.120)$$

Görüldüğü üzere doğrusal olmayan ikilenik denklemin optimal çözümü, aynı zamanda doğrusal denklem takımının çözümüdür. Buradan hareketle doğrusal denklem takımının tekrarlı yöntemlerle çözümü adına kullanılan eşlenik gradyan yöntemi, doğrusal olmayan eniyileme süreçlerinin çözümünde de kullanılabilir. Önerilen yöntem, her ne kadar ikilenik formdaki sorunlarda geçerli olsa da doğrusal olmayan bir fonksiyonun, yerel olarak ikilenik fonksiyon ile benzeşiminin kurulması ve bu sayede ikilenik fonksiyona ait azami noktanın yerinin gerçek fonksiyondaki izdüşümünün dikkate alınması ile tarama yapılması mümkündür. Eşlenik gradyan yönteminin temelinde, verimli bir yön tayini yatmaktadır. Bu sayede birden fazla adım atmak yerine, sadece bir adımla aynı çözüme erişim hedeflenmektedir. Gradyan esaslı yöntemlerin çözüm yaklaşımı;

$$x_{i+1} = x_i + \alpha_i d_i$$

biçimindeki tekrarlı işlemlerini içermekteydi. Daha önce bahsedilen endik eniyileme yöntemi ile doğrultu yöneyi,  $d_i = -\nabla f(x_i)$  denklemi kullanılarak elde edilmekteydi. Her seferinde birbirinden bağımsız olan doğrultu yöneyleri yerine eşlenik gradyan yönteminde, birbirleri ile basit şekilde doğrusal ilişki içerisinde olan yöneylerin kullanılması öngörülmektedir. Buna göre ardışık doğrultu yöneyleri arasında doğrusal karakterli;

<sup>i</sup> Conjugate gradient method



$$\mathbf{d}_i = -\nabla f_i + \beta_i \mathbf{d}_{i-1} \quad (2.121)$$

ifadesi yazılabilir. Burada  $\beta_i$  terimi her seferinde tekrar hesaplanan ve  $\mathbf{d}_i$  ile  $\mathbf{d}_{i-1}$  doğrultularının eşlenik doğrultu olmalarını sağlayan bir katsayıdır. Hatırlanacağı üzere birbirleri işe  $A$  – eşleniklik özelliği gösteren doğrultular arasında;

$$\mathbf{d}_i^T \mathbf{A} \mathbf{d}_j = 0, \quad i \neq j$$

eşitliği söz konusuydu. Öngörülen eşleniklik şartının sağlanması için  $\beta_i$  katsayısının ne olması gerektiği ise yapılan bazı güdüleme işlemleri ile kolayca elde edilebilir. Doğrultular arasındaki doğrusal bağıntının her iki tarafı  $\mathbf{A} \mathbf{d}_{i-1}$  ifadesi ile çarpılırsa;

$$\mathbf{d}_i^T \mathbf{A} \mathbf{d}_{i-1} = [-\nabla f_i + \beta_i \mathbf{d}_{i-1}]^T \mathbf{A} \mathbf{d}_{i-1} \quad (2.122)$$

eşitliği elde edilir. Bu ifadenin sol tarafı  $A$  – eşleniklik gereği sıfıra eşitlenirse;

$$-\nabla f_i^T \mathbf{A} \mathbf{d}_{i-1} + \beta_i \mathbf{d}_{i-1}^T \mathbf{A} \mathbf{d}_{i-1} = 0 \quad (2.123)$$

ifadesi ve buradan  $\beta_i$  katsayısı çekilirse;

$$\beta_i = \frac{\nabla f_i^T \mathbf{A} \mathbf{d}_{i-1}}{\mathbf{d}_{i-1}^T \mathbf{A} \mathbf{d}_{i-1}} \quad (2.124)$$

denklemi bulunur. Uygulamalarda genellikle  $A$  matrisini elemine edecek farklı güdülemeler yapılır. Örneğin Hestenes-Stiefel yaklaşımına göre gradyan esaslı yöntemlerin genel denkleminde hareketle;

$$\mathbf{x}_i = \mathbf{x}_{i-1} + \alpha_{i-1} \mathbf{d}_{i-1} \quad (2.125)$$

eşitliği  $A$  matrisi ile çarpılırsa;

$$\mathbf{A} \mathbf{x}_i = \mathbf{A} \mathbf{x}_{i-1} + \alpha_{i-1} \mathbf{A} \mathbf{d}_{i-1} \quad (2.126)$$

sonucu elde edilir. İkilenik formun türevi alındığında elde edilen denklem dikkate alınır;

$$\begin{aligned} \nabla f_{i-1} &= \mathbf{A} \mathbf{x}_{i-1} - \mathbf{b} \\ \nabla f_i &= \mathbf{A} \mathbf{x}_i - \mathbf{b} \end{aligned} \quad (2.127)$$

yazılabilir.



Bu ifadelerden  $\mathbf{A}$  matrisi içeren terimler çekilir ve bir önceki denklemde yerine koyularak sadeleştirmeye gidilir;

$$\begin{aligned} \mathbf{A}\mathbf{x}_i &= \nabla f_i + \mathbf{b} \\ \mathbf{A}\mathbf{x}_{i-1} &= \nabla f_{i-1} + \mathbf{b} \\ \nabla f_i + \mathbf{b} &= \nabla f_{i-1} + \mathbf{b} + \alpha_{i-1}\mathbf{A}\mathbf{d}_{i-1} \end{aligned} \quad (2.128)$$

sonucuna varılır. Buradan  $\mathbf{A}\mathbf{d}_{i-1}$  ifadesi yerine;

$$\mathbf{A}\mathbf{d}_{i-1} = \frac{\nabla f_i - \nabla f_{i-1}}{\alpha_{i-1}} \quad (2.129)$$

ifadesi yazılabilir. Elde edilen bu denklem,  $\beta_i$  katsayısıyla ilgili Denklem 2.124'te dikkate alınır;

$$\beta_i = \frac{\nabla f_i^T \mathbf{A}\mathbf{d}_{i-1}}{\mathbf{d}_{i-1}^T \mathbf{A}\mathbf{d}_{i-1}} = \frac{\nabla f_i^T \frac{\nabla f_i - \nabla f_{i-1}}{\alpha_{i-1}}}{\mathbf{d}_{i-1}^T \frac{\nabla f_i - \nabla f_{i-1}}{\alpha_{i-1}}} = \frac{\nabla f_i^T (\nabla f_i - \nabla f_{i-1})}{\mathbf{d}_{i-1}^T (\nabla f_i - \nabla f_{i-1})} \quad (2.130)$$

sonucuna ulaşılır. Benzer bir yaklaşım, Fletcher-Reeves denklemi olarak ifade edilen;

$$\beta_i = \frac{\nabla f_i^T \nabla f_i}{\nabla f_{i-1}^T \nabla f_{i-1}} \quad (2.131)$$

denklemdir. Literatürde sıkça karşılaşılan diğer tanımlar, Polak-Ribiere ve Dai-Yuan denklemleri olup müteakip denklemlerde ifade edilmişlerdir<sup>(52)</sup>;

$$\beta_i = \frac{\nabla f_i^T (\nabla f_i - \nabla f_{i-1})}{\nabla f_{i-1}^T \nabla f_{i-1}} \quad (2.132)$$

$$\beta_i = \frac{\nabla f_i^T \nabla f_i}{-\mathbf{d}_{i-1}^T (\nabla f_i - \nabla f_{i-1})} \quad (2.133)$$

Uygulamada hangi tanımın daha iyi sonuç verdiği hususunda genel bir kanı mevcut olmayıp, çözüm sürecinin verimliliği büyük oranda sorunun doğasına ve ana denklemde yer alan adım büyüklüğü değerine bağlıdır. Eşlenik gradyan yöntemi, ikilenik formdaki  $n$  değişkenli fonksiyonlarda  $n$  adımda optimal çözüme ulaşmayı sağlar. İfade edilen bu önermeyi  $A$  -eşleniklik







özelliğini kullanarak göstermek mümkündür. Hatırlanacağı üzere söz konusu önerme Powell yönteminde dikkate alınmış ve  $n$  adım sonrasında ikilenik formdaki bir fonksiyonun optimal noktasına ulaşılabildiği görülmüştü.

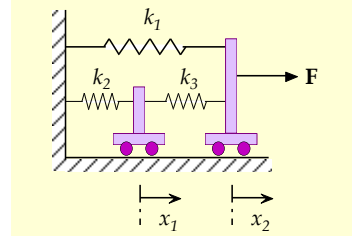
Eşlenik gradyan yönteminin daha iyi anlaşılabilmesi için Şekil 2.73 dâhilinde tanımlanan üç yaylı bir kütle düzenegi düşünelim. Üzerine  $F$  kuvveti uygulandığında etkilenen kütle-yay düzeneginin yer değiştirme değerlerini ( $x$  yöneyi), potansiyel enerjinin minimizasyonu süreci ile elde etmek mümkündür. Buna göre sisteme ait potansiyel enerji değerini, yaylardaki gerilme enerjisi ile yapılan iş arasındaki farka dayandırabiliriz;

$$f(x) = \frac{1}{2}k_2x_1^2 + \frac{1}{2}k_3(x_2 - x_1)^2 + \frac{1}{2}k_1x_2^2 - Fx_2$$

Denklemdede;  $k_i$  yay katsayılarını,  $x_i$  yer değiştirme mesafelerini göstermektedir. Sorunun somutlaştırılması adına yay katsayısı değerlerini;  $k_1 = 200 \text{ N/cm}$ ,  $k_2 = 300 \text{ N/cm}$ ,  $k_3 = 400 \text{ N/cm}$  ve uygulanan kuvveti  $F = 2000 \text{ N}$  olarak dikkate alalım. Verilen değerler enerji denkleminde dikkate alınıp uygun düzenlemeler yapılırsa;

$$f(x) = 350x_1^2 + 300x_2^2 - 400x_1x_2 - 2000x_2$$

denklemini elde edilir.



Şekil 2.75 Üç yay ve iki hareketli kütleyle sahip kütle-yay düzenegi.

İfade edilen potansiyel enerji denklemini minimize etmek için nispeten basit nitelikteki Fletcher-Reeves denklemi bazlı eşlenik gradyan yöntemini kullanabiliriz. Bu meyanda başlangıç noktası olarak  $x_0^T = [0 \ 0]$ , çözüm hassasiyet değeri olarak  $\varepsilon = 0.02$  ve ayrıca yakınsama ölçütü olarak da  $|x_{i+1} - x_i| \leq \varepsilon$  ifadesi dikkate alınabilir. Yöntem dâhilinde yapılacak ilk işlem, başlangıç noktasındaki gradyanın ve dolayısıyla ilk doğrultunun belirlenmesidir;





$$\mathbf{d}_0 = -\nabla f_0 = - \begin{bmatrix} \frac{\partial f}{\partial x_{0,1}} \\ \frac{\partial f}{\partial x_{0,2}} \end{bmatrix} = - \begin{bmatrix} 700x_{0,1} - 400x_{0,2} \\ 600x_{0,2} - 400x_{0,1} - 2000 \end{bmatrix} = \begin{bmatrix} 0 \\ 2000 \end{bmatrix}$$

Bu doğrultu dâhilindeki ilk nokta;

$$\mathbf{x}_1 = \mathbf{x}_0 + \alpha_0 \mathbf{d}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \alpha_0 \begin{bmatrix} 0 \\ 2000 \end{bmatrix} = \begin{bmatrix} 0 \\ 2000\alpha_0 \end{bmatrix}$$

olarak belirlenir. Elde edilen yöney, optimal adım büyüklüğünün tespiti amacıyla amaç fonksiyonunda yerine konularak birinci mertebeden türev alınır;

$$f(\alpha_0) = 300(2000\alpha_0)^2 - 2000(2000\alpha_0)$$

$$\frac{\partial f(\alpha_0)}{\partial \alpha_0} = 2.4 \times 10^9 \alpha_0 - 4 \times 10^6 = 0$$

$$\alpha_0^* = 1.66 \times 10^{-3}$$

Belirlenen adım büyüklüğünün dikkate alınmasıyla ulaşılan yeni nokta;

$$\mathbf{x}_1 = \begin{bmatrix} 0 \\ 3.32 \end{bmatrix}$$

şeklindedir. İlk adım sonrası yakınsama kontrolü yapılırsa;

$$|\mathbf{x}_1 - \mathbf{x}_0| = \left| \begin{bmatrix} 0 \\ 3.32 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right| = 3.32 > 0.02$$

olduğu görülür. İstenen yakınsama ölçütünün karşılanmaması nedeniyle tarama işlemine devam edilir. Bir sonraki noktanın tespiti amacıyla yön tayininin yapılması gerekmekte olup Fletcher-Reeves denklemi bazlı eşlenik gradyan yöntemine göre yeni yön;

$$\mathbf{d}_1 = -\nabla f_1 + \beta_1 \mathbf{d}_0$$

$$\beta_1 = \frac{\nabla f_1^T \nabla f_1}{\nabla f_0^T \nabla f_0}$$

denklemleri ile elde edilebilir. Dolayısıyla bir sonraki yön;





$$\nabla f_1 = \begin{bmatrix} 700x_{1,1} - 400x_{1,2} \\ 600x_{1,2} - 400x_{1,1} - 2000 \end{bmatrix} = \begin{bmatrix} -1328 \\ -8 \end{bmatrix}$$

$$\beta_1 = \frac{\begin{bmatrix} -1328 & -8 \end{bmatrix} \begin{bmatrix} -1328 \\ -8 \end{bmatrix}}{\begin{bmatrix} 0 & -2000 \end{bmatrix} \begin{bmatrix} 0 \\ -2000 \end{bmatrix}} = 0.4409$$

$$\mathbf{d}_1 = -\begin{bmatrix} -1328 \\ -8 \end{bmatrix} + (0.4409) \begin{bmatrix} 0 \\ 2000 \end{bmatrix} = \begin{bmatrix} 1328 \\ 889.8 \end{bmatrix}$$

olarak bulunur. Elde edilen yönün yeni noktada dikkate alınması neticesinde;

$$\mathbf{x}_2 = \mathbf{x}_1 + \alpha_1 \mathbf{d}_1 = \begin{bmatrix} 0 \\ 3.32 \end{bmatrix} + \alpha_1 \begin{bmatrix} 1328 \\ 889.8 \end{bmatrix}$$

$$\mathbf{x}_2 = \begin{bmatrix} 1328\alpha_1 \\ 3.32 + 889.8\alpha_1 \end{bmatrix}$$

ifadesine ulaşılır. Optimal adım büyüklüğünün tespiti amacıyla, elde edilen yöney amaç fonksiyonunda yerine koyularak birinci mertebeden türev alınır;

$$\frac{\partial f(\alpha_1)}{\partial \alpha_1} = 7.6423 \times 10^8 \alpha_1 - 1.7707 \times 10^6 = 0$$

$$\alpha_1^* = 0.0023$$

Bulunan yeni adım büyüklüğünün dikkate alınmasıyla bir sonraki nokta;

$$\mathbf{x}_2 = \begin{bmatrix} 3.0544 \\ 5.3665 \end{bmatrix}$$

şeklinde bulunur. İkinci adım sonrası yakınsama kontrolü yapılırsa;

$$|\mathbf{x}_2 - \mathbf{x}_1| = \left| \begin{bmatrix} 3.0544 \\ 5.3665 \end{bmatrix} - \begin{bmatrix} 0 \\ 3.32 \end{bmatrix} \right| = 3.67 > 0.02$$

olduğu görülür. İstenen yakınsama ölçütünün karşılanmaması nedeniyle tarama işlemine devam edilir. Üçüncü noktanın tespiti amacıyla yeni yön tayininin yapılması gerekmekte olup yine Fletcher-Reeves denklemi bazlı eşlenik gradyan yöntemini dikkate alarak;



$$\nabla f_2 = \begin{bmatrix} 700x_{2,1} - 400x_{2,2} \\ 600x_{2,2} - 400x_{2,1} - 2000 \end{bmatrix} = \begin{bmatrix} -8.52 \\ -1.86 \end{bmatrix}$$

$$\beta_2 = \frac{\nabla f_2^T \nabla f_2}{\nabla f_1^T \nabla f_1} = \frac{\begin{bmatrix} -8.52 & -1.86 \end{bmatrix} \begin{bmatrix} -8.52 \\ -1.86 \end{bmatrix}}{\begin{bmatrix} -1328 & -8 \end{bmatrix} \begin{bmatrix} -1328 \\ -8 \end{bmatrix}} = 4.3121 \times 10^{-5}$$

$$\mathbf{d}_2 = -\nabla f_2 + \beta_2 \mathbf{d}_1 = -\begin{bmatrix} -8.52 \\ -1.86 \end{bmatrix} + 4.3121 \times 10^{-5} \begin{bmatrix} 1328 \\ 889.8 \end{bmatrix} = \begin{bmatrix} 8.5772 \\ 1.8983 \end{bmatrix}$$

işlemleri sonrasında yeni doğrultu bulunur. Elde edilen yönün bir sonraki noktada dikkate alınması neticesinde;

$$\mathbf{x}_3 = \mathbf{x}_2 + \alpha_2 \mathbf{d}_2 = \begin{bmatrix} 3.0544 \\ 5.3665 \end{bmatrix} + \alpha_2 \begin{bmatrix} 8.5772 \\ 1.8983 \end{bmatrix} = \begin{bmatrix} 3.0544 + 8.5772\alpha_2 \\ 5.3665 + 1.8983\alpha_2 \end{bmatrix}$$

ifadesine ulaşılır. Optimal adım büyüklüğünün tespiti amacıyla gerekli türev işlemleri yapılırsa;

$$\frac{\partial f(\alpha_2)}{\partial \alpha_2} = 3.5342 \times 10^4 \alpha_2 - 72.1543 = 0$$

$$\alpha_2^* = 0.0020$$

sonucuna varılır. Bu adım büyüklüğünün dikkate alınmasıyla yeni nokta;

$$\mathbf{x}_3 = \begin{bmatrix} 3.071 \\ 5.370 \end{bmatrix}$$

olarak elde edilir. Üçüncü adım sonrası yakınsama kontrolü yapılırsa;

$$|\mathbf{x}_3 - \mathbf{x}_2| = \left| \begin{bmatrix} 3.071 \\ 5.370 \end{bmatrix} - \begin{bmatrix} 3.0544 \\ 5.3665 \end{bmatrix} \right| = 0.017 < 0.02$$

olduğu görülür. Bu adımda, istenen yakınsama ölçütünün karşılanması nedeniyle tarama işlemi sonlandırılır. Buna göre kütle–yay düzeneğinde yer değiştirme değerleri  $\mathbf{x}^T = [3.07 \ 5.36] \text{ cm}$  olup elde edilen bu değerler analitik çözüm verileri ile de gayet uyumludur. Düzeneğin analitik çözümü;



$$\mathbf{x} = \begin{bmatrix} Fk_3/(k_1k_2 + k_1k_3 + k_2k_3) \\ F(k_2 + k_3)/(k_1k_2 + k_1k_3 + k_2k_3) \end{bmatrix}$$

ifadesi ile elde edilebilir.

#### (4) “Newton” yöntemi

Hatırlanacağı üzere daha önce tek boyutlu arama yöntemleri bahsinde Newton yöntemine değinilmişti. Herhangi bir fonksiyonun Taylor serisi açılımı ve bu açılımın ilk üç terimi dikkate alınarak, yaklaşık da olsa ikilenik formu yazılabilmekteydi. Tek değişkenli fonksiyonlardaki bu ifade, çok değişkenli fonksiyonlar için de geçerlidir. Değişen sadece 1inci ve 2nci mertebeden türev değerleri yerine, çok değişkenli yapıya uygun olarak gradyan ve ikincil türev matrisi değerlerinin kullanılmasıdır. Buna göre çok değişkenli herhangi bir fonksiyonu;

$$f(\mathbf{x}) = f(\mathbf{x}_i) + \nabla f_i^T (\mathbf{x} - \mathbf{x}_i) + \frac{1}{2} (\mathbf{x} - \mathbf{x}_i)^T H_i (\mathbf{x} - \mathbf{x}_i) \quad (2.134)$$

formunda yazmak mümkündür. Burada  $H_i$  matrisi ikincil türev matrisini göstermektedir. Bu ifadenin  $\mathbf{x}$  değişkenine göre birinci mertebeden türevi alınıp sifıra eşitlenirse optimal nokta değeri elde edilebilir. Buna göre;

$$\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} = \nabla f_i + H_i (\mathbf{x} - \mathbf{x}_i) = 0 \quad (2.135)$$

denklemden hareketle  $\mathbf{x}$  değeri, ya da diğer bir ifade ile bir sonraki nokta değeri yalnız bırakılırsa;

$$\mathbf{x}_{i+1} = \mathbf{x}_i - \frac{\nabla f_i}{H_i} \quad (2.136)$$

ifadesine ulaşılır. Newton yönteminin sonucunda azami veya asgari nokta değerine ulaşılması, ikincil türev matrisine bağlıdır. Tekrarlı işlemler esnasında ikincil türev matrisinin pozitif tanımlı olması durumunda, yerel asgari nokta niteliğindeki optimal nokta değerine ulaşmak mümkündür. Matrisin negatif tanımlı olması halinde ise azami noktaya ulaşılabilir. İkincil türev matrisinin karakterindeki sorunların yakınsama sürecini doğrudan etkilediği de söylenebilir. Bununla beraber ikilenik formdaki amaç fonksiyonlarında optimal noktaya tek adımda ulaşılabilir.



Müteakip şekilde tanımlanmış bir ikilenik fonksiyonu dikkate alarak en uygun noktayı kolayca ifade edebiliriz;

$$f(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T \mathbf{A} \mathbf{x} - \mathbf{x}^T \mathbf{b} + c$$

Zira bu ifadenin optimal noktası;

$$\nabla f(\mathbf{x}) = \mathbf{A} \mathbf{x} - \mathbf{b} = 0 \quad (2.137)$$

$$\mathbf{x}^* = \mathbf{A}^{-1} \mathbf{b} \quad (2.138)$$

şeklinde yazılabilir. Diğer taraftan  $\mathbf{x}_0$  başlangıç noktası baz alınarak bir sonraki nokta;

$$\mathbf{x}_1 = \mathbf{x}_0 - \frac{\nabla f_0}{H_0} \quad (2.139)$$

$$\begin{aligned} \nabla f_0 &= \mathbf{A} \mathbf{x}_0 - \mathbf{b} \\ H_0 &= \mathbf{A} \end{aligned} \quad (2.140)$$

eşitlikleri dahilinde;

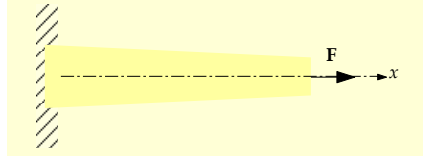
$$\mathbf{x}_1 = \mathbf{x}_0 - \frac{\mathbf{A} \mathbf{x}_0 - \mathbf{b}}{\mathbf{A}} = \mathbf{A}^{-1} \mathbf{b} \quad (2.141)$$

$$\mathbf{x}_1 = \mathbf{x}^*$$

olarak elde edilir. Görüldüğü üzere ikilenik bir fonksiyonda optimal nokta tek seferde elde edilebilmektedir. Bununla beraber ikilenik olmayan amaç fonksiyonunun söz konusu olduğu sorunlarda yöntem aynı etkinliği göstermez. Çözümüne ulaşmada atılacak adım sayısı, sorunun doğasına göre değişiklik gösterebilir.

Newton yönteminin daha iyi anlaşılabilmesi için Şekil 2.76'daki gibi tanımlanmış bir sorunu adım adım çözebiliriz. Buna göre bir ucu ankastre, diğer ucu doğrusal olarak daralan bir kiriş düzeneği düşünelim.





Şekil 2.76 Ucu köreltilmiş kiriş düzeneği.

Kirişin ucuna 200 N'luk  $F$  kuvveti etki etmekte olup yüksekliği kök kısmında  $y = 500 \text{ mm}$ , uç kısmında  $y = 250 \text{ mm}$  olmaktadır. Kirişin genişliği sabit olup  $250 \text{ mm}$  ve uzunluğu  $L = 2 \text{ m}$ 'dir. Kirişe etki eden kuvvet nedeniyle  $x$  eksenı boyunca meydana gelen uzama değerlerini, potansiyel enerjinin minimizasyonu yolu ile elde etmek mümkündür. Buna göre potansiyel enerjiyi;

$$E_p = \int_0^L \frac{EA}{2} \left( \frac{\partial u}{\partial x} \right)^2 dx - Fu|_{x=L}$$

denklemi ile tanımlamak mümkündür. Burada  $E$  elastisite modülünü,  $A$  kesit alanı ve  $u$  eksen boyunca değişen uzama değerini göstermektedir. Doğrusal olarak değişen yükseklik nedeniyle kesit alan değeri;

$$A = 0.25(0.5 - 0.125x)$$

denklemi ile temsil edilebilir. Diğer taraftan uzama değişimi için Rayleigh-Ritz yaklaşımı ile ikinci dereceden bir denklem kabulü yapılabilir;

$$u(x) = a_1x + a_2x^2$$

İfade edilen denklemlerin enerji denkleminde dikkate alınması neticesinde;

$$f = 9375a_1^2 + \frac{100000}{3}a_1a_2 + \frac{125000}{3}a_2^2 - 200(2a_1 + 4a_2)$$

şeklindeki amaç fonksiyonunu elde edebiliriz<sup>(53)</sup>. Amaç fonksiyonunun minimizasyonu kapsamında başlangıç noktası olarak  $\mathbf{a}_0^T = [0 \ 0]$  değerini dikkate alalım. Diğer taraftan  $\mathbf{a}_0$  başlangıç noktası baz alınarak Newton yöntemine göre bir sonraki nokta;

$$\mathbf{a}_1 = \mathbf{a}_0 - \frac{\nabla f_0}{H_0};$$



eşitliği ile elde edilebilir. Dolayısıyla gradyan ve ikincil türev matrisleri;

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial a_1} \\ \frac{\partial f}{\partial a_2} \end{bmatrix} = \begin{bmatrix} 18750a_1 + 33333.3a_2 - 400 \\ 33333.3a_1 + 83333.3a_2 - 800 \end{bmatrix}$$

$$\nabla f(\mathbf{a}_0) = \begin{bmatrix} -400 \\ -800 \end{bmatrix}$$

$$H = \begin{bmatrix} \frac{\partial^2 f}{\partial a_1^2} & \frac{\partial^2 f}{\partial a_1 a_2} \\ \frac{\partial^2 f}{\partial a_2 a_1} & \frac{\partial^2 f}{\partial a_2^2} \end{bmatrix} = \begin{bmatrix} 18750 & 33333.3 \\ 33333.3 & 83333.3 \end{bmatrix}$$

şeklinde elde edilip önceki eşitlikte yerine koyulursa;

$$\mathbf{a}_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} - \frac{\begin{bmatrix} -400 \\ -800 \end{bmatrix}}{\begin{bmatrix} 18750 & 33333.3 \\ 33333.3 & 83333.3 \end{bmatrix}} = \begin{bmatrix} 0.01476 \\ 0.003692 \end{bmatrix}$$

sonucuna ulaşılır. Bu değerlerin amaç fonksiyonunda yerine konulması neticesinde elde edilen değer;

$$f(\mathbf{a}_1) = -1.4788$$

olup bu noktadaki gradyan değeri ise;

$$\nabla f(\mathbf{a}_1) \cong \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

olarak bulunur. Bu değer, elde edilen noktanın yerel optimal nokta olduğunu göstermektedir. Eniyileme süreci sonunda ulaşılan noktadaki katsayı değerlerinin uzama fonksiyonunda yerine koyulması ile konuma bağlı denkleme erişilmiş olur;

$$u(x) = 0.01476x + 0.003692x^2$$

Buna göre en uç noktadaki uzama miktarı  $u(2) = 0.04428 \text{ m}$  olarak elde edilir. Bulunan sonuçlar, gradyanın sifıra eşitlenmesi sayesinde elde edilebilecek analitik değerlerle uyumlu görünmektedir.







$$f(x) = x_1^4 - 2x_1x_2 + x_2^2 + 3x_2^4$$



MATLAB uygulaması



```
clc; clear; i = 2; x(:,1) = [1;1]; eps = 0.01;
grad = [4*x(1,1)^3 - 2*x(2,1); 12*x(2,1)^3 + 2*x(2,1) - 2*x(1,1)];
hessian = [ 12*x(1,1)^2 -2;-2 36*x(2,1)^2+2];
x(:,2) = x(:,1)-hessian\grad;
yak = norm(x(:,2) - x(:,1));

while yak>eps
    grad = [4*x(1,i)^3 - 2*x(2,i); 12*x(2,i)^3 + 2*x(2,i) - 2*x(1,i)];
    hessian = [ 12*x(1,i)^2 -2;-2 36*x(2,i)^2+2];
    x(:,i+1) = x(:,i)-hessian\grad;
    yak = norm(x(:,i+1) - x(:,i)); i = i + 1;
end

f = x(1,:).^4 - 2*x(1,:).*x(2,:) + x(2,:).^2+3*x(2,:).^4 ;
[x' f']
```



ans =

1.0000	1.0000	3.0000
0.7788	0.6726	0.3865
0.6484	0.4703	-0.0652
0.5776	0.3665	-0.1236
0.5521	0.3344	-0.1270
0.5492	0.3312	-0.1270



## (5) Yaklaşık “Newton” yöntemi

“Newton” yöntemi, özellikle ikilenik formdaki amaç fonksiyonlarının uç noktalarının bulunmasında oldukça etkili bir yöntemdir. Ancak yöntemin ciddi manada sayısal zorluklarından da bahsetmek gerekir. Zira birincil ve ikincil türev matrislerinin hesaplanması yanında ilave olarak ikincil türev matrisinin tersinin de kullanılması söz konusudur. Özellikle payda durumundaki matrisin özelliğine göre yöntemin yakınsama veya ıraksama ihtimali mevcuttur. Bazı amaç fonksiyonu tanımlamalarında zaman zaman türev matrislerinin alınması zor veya imkânsız olabilir. Bu nedenle hesaplama maliyeti açısından daha az zahmetli ve kolay yöntemler geliştirilmiş olup bu tür yöntemlere yaklaşık “Newton” yöntemleri denmektedir. Hatırlanacağı üzere “Newton” yönteminde;

$$x_{i+1} = x_i - H_i^{-1} \nabla f_i$$



denklemini dikkate alınmaktaydı. Bu denklemdeki  $H_i^{-1}$  ifadesi yerine,  $B_i$  gibi yaklaşık olarak onu simgeleyen başka bir matrisi kullanmak mümkündür. Yeni tanımlanan matrisin temel özelliği, pozitif tanımlı ve kolay hesaplanabilir olmasıdır. Genellikle hesaplamada birincil türev matrisi değeri dikkate alınır. Buradan hareketle yaklaşık “Newton” yöntemlerinde dikkate alınan iterasyon denklemini;

$$\mathbf{x}_{i+1} = \mathbf{x}_i - \alpha_i^* B_i \nabla f_i \quad (2.142)$$

şeklinde. Denklemde bir öncekine göre ilave olarak  $\alpha_i^*$  katsayısı bulunmakta olup gidilen yön niteliğindeki  $B_i \nabla f_i$  yönünde elde edilebilecek azami azalmayı sağlayan optimal adım büyüklüğünü göstermektedir. Bu meydana yöntemin tek değişkenli de olsa ek bir tarama süreci getirdiği söylenebilir, ancak bu hesaplama yükü ikincil türev işlemlerine göre daha tercih edilebilir bir durumdur.

Yaklaşım matrisi niteliğindeki  $B_i$  değeri için değişik yöntemler önerilmektedir. Bunlardan ilki, 1959 tarihli DFP (Davidon-Fletcher-Powell) yöntemidir. Bu yönteme göre söz konusu yaklaşım matrisi;

$$B_{i+1} = B_i + \frac{\alpha_i^* \mathbf{d}_i^T \mathbf{d}_i}{\mathbf{d}_i^T \mathbf{g}_i} - \frac{(B_i \mathbf{g}_i)^T B_i \mathbf{g}_i}{\mathbf{g}_i^T B_i \mathbf{g}_i} \quad (2.143)$$

ifadesi ile elde edilir<sup>(54)</sup>. Burada;

$$\mathbf{d}_i = -B_i \nabla f_i \quad (2.144)$$

$$\mathbf{g}_i = \nabla f_{i+1} - \nabla f_i \quad (2.145)$$

$$\alpha_i^* = \min_{\alpha \geq 0} f(\mathbf{x}_i + \alpha \mathbf{d}_i) \quad (2.146)$$

şeklinde. Diğer bir yaklaşım matrisi tanımı, 1970 tarihli BFGS (Broyden-Fletcher-Goldfarb-Shanno) yöntemidir. Bu yönteme göre aynı matris;

$$B_{i+1} = B_i + \left[ 1 + \frac{\mathbf{g}_i^T B_i \mathbf{g}_i}{\mathbf{s}_i^T \mathbf{g}_i} \right] \frac{\mathbf{s}_i^T \mathbf{s}_i}{\mathbf{s}_i^T \mathbf{g}_i} - \frac{\mathbf{s}_i \mathbf{g}_i^T B_i}{\mathbf{s}_i^T \mathbf{g}_i} - \frac{B_i \mathbf{g}_i \mathbf{s}_i^T}{\mathbf{s}_i^T \mathbf{g}_i} \quad (2.147)$$

denkleminde göre hesaplanır<sup>(55)</sup>. Burada;



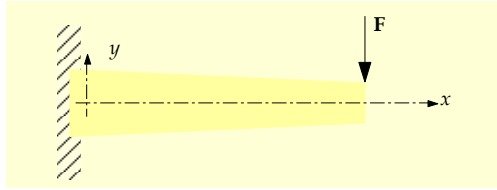
$$s_i = x_{i+1} - x_i \quad (2.148)$$

ifadesi, gidilecek optimal yöneyi göstermektedir. DFP ve BFGS tanımlamaları haricinde daha basit veya daha karmaşık denklemlerden de bahsedilebilir<sup>(56)</sup>. Bu meyanda bahsedebileceğimiz tanımlamalardan biri de Broyden ailesi tanımı olup söz konusu tanım DFP ve BFGS tanımlamalarının ortak kullanıldığı bir denklem niteliğindedir;

$$T_{i+1} = (1 - \varphi_i)(B_{i+1}^{BFGS})^{-1} + \varphi_i(B_{i+1}^{DFP})^{-1}, \varphi_i \in [0, 1] \quad (2.149)$$

$$B_{i+1} = (T_{i+1})^{-1} \quad (2.150)$$

Günümüzde en yaygın olarak kullanılan yaklaşım matrisi ifadesi BFGS tanımlamasıdır<sup>(57)</sup>. Diğer taraftan yaklaşık matris niteliğindeki  $B_i$  için başlangıçta genellikle birim matris dikkate alınır. Yaklaşık “Newton” yönteminin uygulamasının daha iyi anlaşılabilmesi için Şekil 2.77’deki gibi bir problemi, BFGS tanımlamasını dikkate alarak adım adım çözebiliriz.



Şekil 2.77 Ucu köreltilmiş kiriş düzeneği.

Buna göre bir ucu duvara sabitlenmiş ve diğer ucu doğrusal olarak daralan bir kiriş düzeneği düşünelim. Kirişin ucuna  $10 \text{ kN}$ 'luk  $F$  kuvveti etki etmekte olup kirişin yüksekliği doğrusal olarak değişmekte ve kök kısmında  $h = 500 \text{ mm}$ , uç kısmında  $h = 250 \text{ mm}$  olmaktadır. Kirişin genişliği sabit olup  $g = 250 \text{ mm}$ , uzunluğu  $L = 8 \text{ m}$ 'dir. Kirişe etki eden kuvvet nedeniyle dikey eksen boyunca meydana gelen yer değiştirme ya da diğer bir ifade ile eğilme mesafe değerleri, potansiyel enerjinin minimizasyonu yolu ile elde edilebilir. Buna göre potansiyel enerjiyi;

$$E_p = \int_0^L \frac{EI}{2} \left( \frac{\partial^2 y}{\partial x^2} \right)^2 dx - Fy|_{x=L}$$

denklemini ile tanımlamak mümkündür.



Burada  $E$  elastisite modülünü,  $I$  eylemsizlik momentini ve  $y$  dikey eksen boyunca değişen eğilme mesafe değerini göstermektedir. Doğrusal olarak değişen yükseklik nedeniyle eylemsizlik moment değeri;

$$I = \frac{gh^3}{12} = \frac{(16-x)^3}{1572864}$$

denklemi ile temsil edilebilir. Ayrıca uzama değişimi için Rayleigh-Ritz yaklaşımı ile üçüncü dereceden bir denklem kabulü yapılabilir;

$$y(x) = a_0 + a_1x + a_2x^2 + a_3x^3$$

Öte yandan kökteki ankastre sınır şartlarının varlığı nedeniyle  $a_0$  ve  $a_1$  katsayılarının değerinin sıfır olduğu söylenebilir. Dolayısıyla kalan terimleri içeren eğilme denklemi ile eylemsizlik moment denkleminin enerji denkleminde dikkate alınması neticesinde;

$$f = 390625 \times 10^3 a_2^2 + 65 \times 10^8 a_2 a_3 + 42 \times 10^9 a_3^2 + 10000(64a_2 + 512a_3)$$

şeklindeki amaç fonksiyonunu elde edebiliriz. Amaç fonksiyonunun minimizasyonu kapsamında başlangıç noktası olarak  $\mathbf{a}_0^T = [0 \ 0]$  değerini, ayrıca yakınsama kontrolü kapsamında gradyan değerine bağlı olarak  $|\nabla f| \leq \epsilon = 0.001$  ölçütünü dikkate alalım. Başlangıç noktası  $\mathbf{a}_0$  baz alınarak, yaklaşık Newton yöntemine göre bir sonraki nokta;

$$\mathbf{a}_1 = \mathbf{a}_0 - \alpha_0^* B_0 \nabla f_0$$

şeklinde dir. Denklemdaki gradyan değeri ve başlangıçtaki  $B_0$  matrisi;

$$\nabla f_0 = \begin{bmatrix} \frac{\partial f}{\partial a_2} \\ \frac{\partial f}{\partial a_3} \end{bmatrix} = \begin{bmatrix} 78125 \times 10^4 a_{0,2} + 65 \times 10^8 a_{0,3} + 640000 \\ 65 \times 10^8 a_{0,2} + 84 \times 10^9 a_{0,3} + 5120000 \end{bmatrix}$$

$$\nabla f(\mathbf{a}_0) = \begin{bmatrix} 640000 \\ 5120000 \end{bmatrix}, \quad B_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

olarak elde edilebilir. Bu ifadeler bir önceki denklemde yerine konulursa;

$$\mathbf{a}_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} - \alpha_0^* \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 640000 \\ 5120000 \end{bmatrix} = \begin{bmatrix} -640000\alpha_0 \\ -5120000\alpha_0 \end{bmatrix}$$



eşitliği yazılabilir. Bu eşitlikteki optimal adım büyüklüğünün bulunması için amaç fonksiyonu ve onun adım büyüklüğüne göre 1inci mertebeden türev denklemi dikkate alınrsa;

$$\frac{\partial f(\alpha_0)}{\partial \alpha_0} = 2244928 \times 10^{18} \alpha_0 - 26624 \times 10^9 = 0$$

$$\alpha_0^* = 1.186 \times 10^{-11}$$

elde edilir. Bu adım büyüklüğünün dikkate alınmasıyla yeni nokta;

$$\mathbf{a}_1 = 1 \times 10^{-4} \begin{bmatrix} -0.0759 \\ -0.6072 \end{bmatrix}$$

olarak bulunur. Bir sonraki adımın hesaplanması için öncelikle gradyan verisine bakılabilir. Zira gradyan değerinin küçük olması, yerel en uygun noktaya ulaşıldığı anlamına gelir. Dolayısıyla yakınsama kontrolü kapsamında;

$$\nabla f_1 = \begin{bmatrix} 78125 \times 10^4 a_{1,2} + 65 \times 10^8 a_{1,3} + 640000 \\ 65 \times 10^8 a_{1,2} + 84 \times 10^9 a_{1,3} + 5120000 \end{bmatrix}$$

$$f(\mathbf{a}_1) = \begin{bmatrix} 2.3938 \times 10^5 \\ -2.9923 \times 10^4 \end{bmatrix}$$

değeri gözlemlenir. Buna göre gradyan değeri büyük olduğundan bir sonraki adıma geçilir ve müteakip ara işlemlerin yapılması gerekir;

$$\mathbf{g}_0 = \nabla f_1 - \nabla f_0 = \begin{bmatrix} 2.3938 \times 10^5 \\ -2.9923 \times 10^4 \end{bmatrix} - \begin{bmatrix} 640000 \\ 5120000 \end{bmatrix} = \begin{bmatrix} -4.0062 \times 10^5 \\ -5.1499 \times 10^6 \end{bmatrix}$$

$$\mathbf{s}_0 = \mathbf{a}_1 - \mathbf{a}_0 = 1 \times 10^{-4} \begin{bmatrix} -0.0759 \\ -0.6072 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -7.59 \times 10^{-6} \\ -6.072 \times 10^{-5} \end{bmatrix}$$

$$\mathbf{g}_0^T B_0 \mathbf{g}_0 = 2.6682 \times 10^{13}, \quad \mathbf{s}_0 \mathbf{g}_0^T B_0 = \begin{bmatrix} 3.0408 & 39.0887 \\ 24.3260 & 312.7099 \end{bmatrix}, \quad B_0 \mathbf{g}_0 \mathbf{s}_0^T = \begin{bmatrix} 3.0408 & 24.3260 \\ 39.0887 & 312.7099 \end{bmatrix}$$

$$\mathbf{s}_0 \mathbf{s}_0^T = 1 \times 10^{-8} \begin{bmatrix} 0.0058 & 0.0461 \\ 0.0461 & 0.3687 \end{bmatrix}, \quad \mathbf{s}_0^T \mathbf{g}_0 = 315.7506, \quad \mathbf{s}_0 \mathbf{g}_0^T = \begin{bmatrix} 3.0408 & 39.0887 \\ 24.3260 & 312.7099 \end{bmatrix}, \quad \mathbf{g}_0 \mathbf{s}_0^T = \begin{bmatrix} 3.0408 & 24.3260 \\ 39.0887 & 312.7099 \end{bmatrix}$$



$$B_1 = B_0 + \left[ 1 + \frac{\mathbf{g}_0^T B_0 \mathbf{g}_0}{\mathbf{s}_0^T \mathbf{g}_0} \right] \frac{\mathbf{s}_0 \mathbf{s}_0^T}{\mathbf{s}_0^T \mathbf{g}_0} - \frac{\mathbf{s}_0 \mathbf{g}_0^T B_0}{\mathbf{s}_0^T \mathbf{g}_0} - \frac{B_0 \mathbf{g}_0 \mathbf{s}_0^T}{\mathbf{s}_0^T \mathbf{g}_0}$$

$$B_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \left[ 1 + \frac{2.6682 \times 10^{13}}{315.7506} \right] \frac{1 \times 10^{-8}}{315.7506} \begin{bmatrix} 0.0058 & 0.0461 \\ 0.0461 & 0.3687 \end{bmatrix} - \frac{1}{315.7506} \begin{bmatrix} 3.0408 & 39.0887 \\ 24.3260 & 312.7099 \end{bmatrix} - \frac{1}{315.7506} \begin{bmatrix} 3.0408 & 24.3260 \\ 39.0887 & 312.7099 \end{bmatrix}$$

$$B_1 = \begin{bmatrix} 0.9962 & -0.0775 \\ -0.0775 & 0.0060 \end{bmatrix}$$

Elde edilen yeni yaklaşım matrisi ana iterasyon denkleminde yerine koyulursa;

$$\mathbf{a}_2 = \mathbf{a}_1 - \alpha_1 * B_1 \nabla f_1 = 1 \times 10^{-4} \begin{bmatrix} -0.0759 \\ -0.6072 \end{bmatrix} - \alpha_1 * \begin{bmatrix} 0.9960 & -0.0777 \\ -0.0777 & 0.0057 \end{bmatrix} \begin{bmatrix} 2.3938 \times 10^5 \\ -2.9923 \times 10^4 \end{bmatrix}$$

$$\mathbf{a}_2 = \begin{bmatrix} -7.59 \times 10^{-6} - 2.4075 \times 10^5 \alpha_1 \\ -6.072 \times 10^{-5} + 0.1877 \times 10^5 \alpha_1 \end{bmatrix}$$

eşitliği yazılabilir. Bu eşitlikteki optimal adım büyüklüğünün bulunması için amaç fonksiyonu ve onun adım büyüklüğüne göre 1inci mertebeden türev denklemi dikkate alınırsa;

$$\frac{\partial f(\alpha_1)}{\partial \alpha_1} = 1.6134 \times 10^{19} \alpha_1 - 5.8199 \times 10^{10} = 0$$

$$\alpha_1^* = 3.6073 \times 10^{-9}$$

elde edilir. Bu adım büyüklüğünün dikkate alınmasıyla yeni nokta;

$$\mathbf{a}_2 = \begin{bmatrix} -8.7615 \times 10^{-4} \\ 6.8449 \times 10^{-6} \end{bmatrix}$$

olarak hesaplanır. Bir önceki adımda olduğu üzere, sonraki adımın hesaplanması öncesinde yakınsama kontrolü yapılabilir. Bu meyanda gradyan değeri hesaplanırsa;

$$\nabla f_2 = \begin{bmatrix} 78125 \times 10^4 a_{2,2} + 65 \times 10^8 a_{2,3} + 640000 \\ 65 \times 10^8 a_{2,2} + 84 \times 10^9 a_{2,3} + 5120000 \end{bmatrix}$$

$$\nabla f(\mathbf{a}_2) = \begin{bmatrix} 0 \\ 9.31 \times 10^{-10} \end{bmatrix}$$

elde edilir.

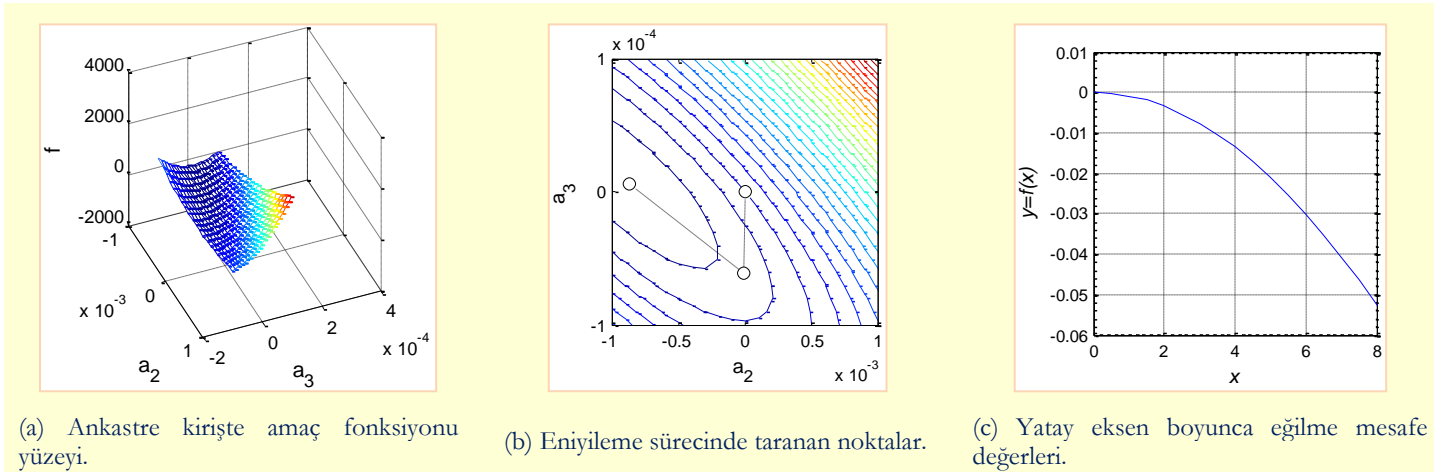


Bu değerin yöneysel şiddeti ile yakınsama ölçütü karşılaştırıldığında yerel optimal noktaya ulaşıldığı görülür;

$$|\nabla f| = 9.31 \times 10^{-10} < \epsilon = 0.001$$

Uygulanan **F** kuvvetine karşılık  $x$  eksenı boyunca eğilme değeri, elde edilen optimal noktanın eğilme denkleminde yerine koyulması ile kolayca bulunabilir;

$$y(x) = -8.7615 \times 10^{-4}x^2 + 6.8449 \times 10^{-6}x^3$$



Şekil 2.78 Amaç fonksiyon yüzeyi, taranan noktalar ve elde edilen eğilme eğrisi.

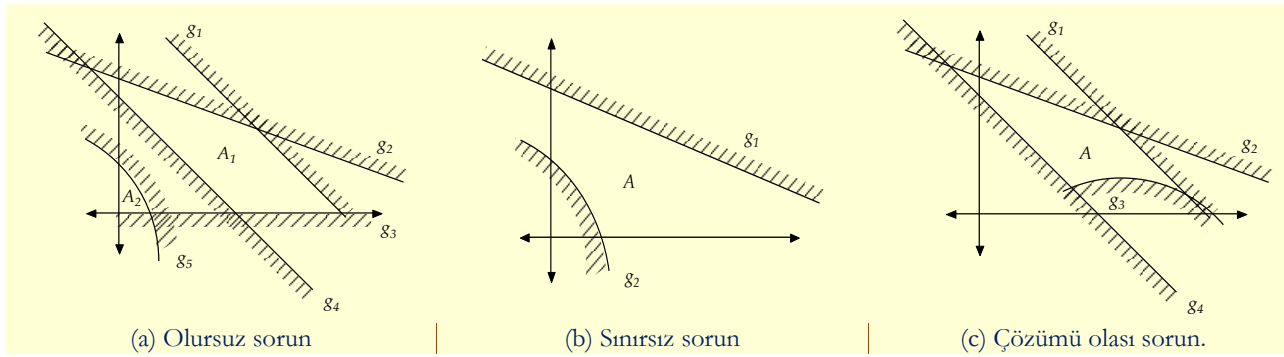
Buna göre en uç noktada eğilme değeri  $y(8) = -0.0526 m$  olarak bulunur. Tıpkı Newton yönteminde olduğu gibi yaklaşık Newton yöntemi de ikilenik formdaki amaç fonksiyonlarında oldukça etkin bir yöntemdir. Bu örnekten de anlaşılacağı üzere çözüme yakın bir noktadan başlandığı takdirde birkaç adımda en uygun noktaya ulaşmak mümkün olmaktadır. Bununla beraber ikilenik formda olmayan sorunlarda yöntemin etkinliği değişkenlik gösterebilir.

### c. Sayısal yöntemlerde kısıtların dikkate alınması

Mühendislik problemlerinin çoğu, doğrusal olmayan karakterdeki sorunlar olup genellikle bu tür sorunlar, bünyelerinde eşitlik veya eşitsizlik kısıtları barındırır. Analitik olarak ifade



edilmesi kolay ve ilk iki mertebedeki türev değerleri de erişim açısından sorun oluşturmayan problemlerde, çözüm yöntemi olarak genelleştirilmiş KKT şartlarını dikkate almak ve analitik bir çözüme gitmek tercih edilebilir bir durumdur. Ancak ifade edilen hususlarda sorun yaşanması durumunda sayısal yöntemlerin kullanılması kaçınılmaz olmakta ve bu meyanda tekrarlı işlemler grubunun içerisine bir şekilde kısıt denklemlerinin dâhil edilmesi gerekmektedir. Çözüm sürecini etkileyen ana unsur, tahmin edilebileceği üzere kısıtların niteliği, daha da ötesi en uygun noktayı etkileyip etkilemediğidir. Ancak optimal noktanın kısıtlardan etkilenip etkilenmediğini önceden bilmek pek mümkün değildir. Bu nedenle kullanılan yöntemin kapsayıcı ve genel bir yöntem olması tavsiye edilir.



Şekil 2.79 Doğrusal olmayan kısıtlı eniyileme sorunlarına örnek problemler.

Kısıtların durumuna göre sorun ve sorunun çözümü değişik özellikler gösterebilir. Örneğin öngörülen kısıtları karşılayan bir tasarım yöneyi, belirtilen amaç fonksiyonu çerçevesinde mümkün olmayabilir. Bu tür sorunlar olursuz sorun<sup>i</sup> olarak adlandırılabilir. Bazı sorunlarda çözüm mümkün olabilir, ancak mevcut kısıtlar kapsamında çözümün alt veya üst sınırı tanımlanamayabilir. Öyle ki tasarım yöneyi değerleri değiştikçe daha iyi çözümler elde edilebilir. Bu tür sorunlar her ne kadar kısıt içerse de sınırsız sorun<sup>ii</sup> olarak nitelendirilebilir. Bizim üzerinde duracağımız sorunlar, belirtilen kısıtları karşılayan ve sınırlı sayıda çözüm içeren sorunlardır. Bu tür sorunlar literatürde çözümü olası<sup>iii</sup> sorun olarak adlandırılır. Şekil 2.79'daki grafiklerde örnek sorun gösterimleri görülmektedir. Buna göre şekildeki ilk seçenekte mevcut şartlar altında

<sup>i</sup> An infeasible problem

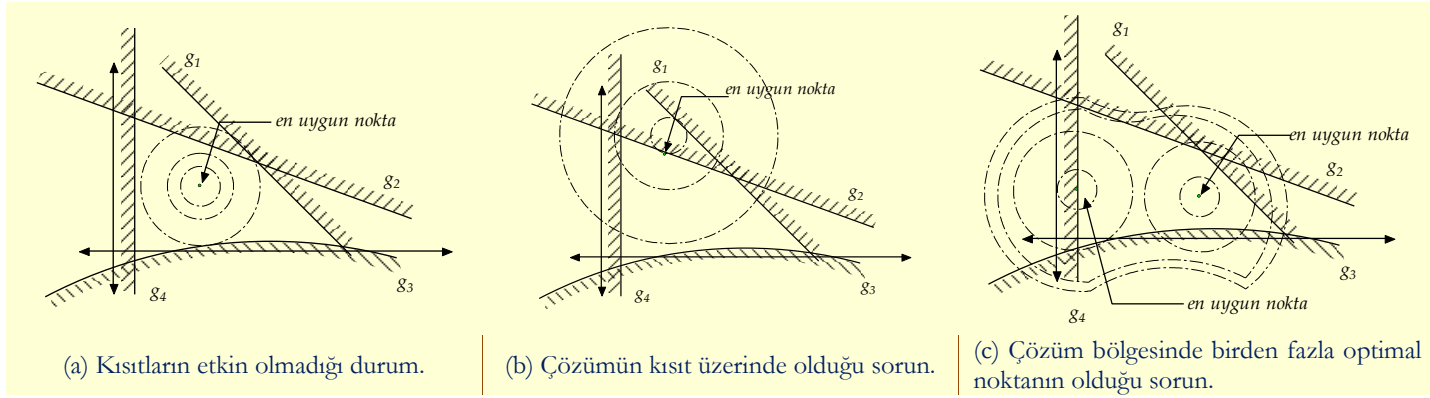
<sup>ii</sup> An unbounded problem

<sup>iii</sup> A feasible problem





çözümün mümkün olmadığı görülmektedir. Zira  $A_1$  bölgesi  $g_i, i = 1,2,3,4$  kısıtlarını karşılarken  $g_5$  kısıtını karşılayamamaktadır. Buna mukabil  $A_2$  bölgesi  $g_i, i = 1,2,3,5$  kısıtlarını karşılarken  $g_4$  kısıtını karşılayamamaktadır. Tüm kısıtları karşılayan bir bölge veya nokta olası görünmemektedir. İkinci seçenekte  $A$  bölgesi, belirtilen her iki kısıtı da rahatlıkla karşılamaktadır. Ancak bu bölgenin sınırlarının kapalı olmaması nedeniyle, amaç fonksiyonunun niteliğine bağlı olarak sonsuz sayıda olası çözüm söz konusu olabilir. Son seçenekte  $A$  bölgesi belirtilen tüm kısıtların şartlarını karşılamakta ve sınırlı sayıda olası çözüm noktası barındırmaktadır. Kısıtların etkinliği hususu, bu meyanda diğer bir ölçüttür. Şekil 2.80(a)'dan da görüleceği üzere mevcut kısıtlar ve amaç fonksiyonu beraber değerlendirildiğinde, kısıtların amaç fonksiyonuna bir etkisinin olmadığı görülmektedir.



Şekil 2.80 Çözüm üzerinde kısıtların etkisi.

Aynı şeklin (b) seçeneğinde çözümün kısıt hattı üzerinde olduğunu gözlemlemekteyiz. Buna göre kısıt denklemi, sorunun çözümünde aktif olup en uygun noktayı belirleyici unsur olarak görünmektedir. Son seçenekte kısıtların çözüm bölgesini daralttığı, ancak çözüm bölgesinde birden fazla yerel en uygun nokta olduğu müşahede edilmektedir. Bu sorunda göze çarpan diğer bir özellik de çözümlerden birinin kısıt üzerinde olması, diğerinin ise kısıttan bağımsız bir konumda bulunmasıdır.

Doğrusal olmayan kısıtlı eniyileme problemlerinin çözümü ile ilgili literatürde pek çok yöntemden bahsedilebilir. Bu yöntemler genellikle iki ana gruba ayrılmakta olup bunlar doğrudan ve dolaylı çözüm yöntemleridir. İlk grup olarak ifade edilen doğrudan çözüm yöntemleri, kısıtların açık formlarının dikkate alınması ve amaç fonksiyonuna entegre edilmesi yaklaşımını, ikinci grup olarak ifade edilen dolaylı çözüm yöntemleri ise kısıtların uygun dönüşümlerle kısıtsız



optimizasyon sorununa evrilmesi ve kademeli olarak kısıtsız eniyileme yöntemlerinin kullanılması yaklaşımını içermektedir. Bazı çalışmalarda doğrudan çözüm yöntemleri aktif set yöntemleri olarak da bilinir. Zira bu yöntemler bulunulan konum itibari ile etkin olan kısıtların dikkate alınması ve akabinde doğrusallaştırma üzerine yapılandırılmaktadır. Kitabın bu bölümünde doğrudan çözüm yöntemleri adı altında ardışık doğrusal programlama<sup>i</sup>, ikilenik programlama<sup>ii</sup>, uygun doğrultu<sup>iii</sup>, genelleştirilmiş-azaltılmış gradyan<sup>iv</sup> ve ardışık ikilenik programlama<sup>v</sup> yöntemlerinden bahsedilmektedir. Dolaylı yöntemler dâhilinde dönüşüm yaklaşımı ve ceza fonksiyonu yöntemleri<sup>vi</sup> dikkate alınmaktadır.

### (1) Ardışık doğrusal programlama yöntemi

Ardışık doğrusal programlama yönteminin temelinde doğrusal olmayan sorunun alt problem niteliğindeki doğrusal problemlere dönüştürülmesi ve bu sayede analitik sonuçlar veren doğrusal yöntemler kullanılarak çözüme gidilmesi düşüncesi yatmaktadır. Yöntemin diğer isimleri arasında yaklaşık programlama yöntemi veya kesik düzlem yöntemi<sup>vii</sup> sayılabilir. 1950'li yılların sonlarına doğru literatüre girmiş olan yöntemin ilk uygulamaları endüstri kaynaklı olup örnekleri arasında petrol rafinerisi optimizasyonu<sup>(58)</sup> ile ısı değiştiricisi eniyilemesi<sup>(59)</sup> sayılabilir. Doğrusal olmayan karakterdeki amaç veya kısıtların doğrusal denklem haline getirilmesinde izlenen yaklaşım oldukça basit olup 1inci mertebeden Taylor serisi açılımı dikkate alınmaktadır. Adından da anlaşılacağı üzere yöntem, yaklaşık bir yöntem olup alt eniyileme süreci sonunda ulaşılan noktanın gerçek manada optimal nokta olması beklenmemelidir. Ancak elde edilen nokta, bir sonraki alt problem için referans nokta olarak kullanılabilir. Yöntemle ilgili diğer bir husus, mevcut kısıtlara ilave olarak, eğer tanımlanmamış ise, her bir değişken için alt ve üst sınır belirlenmesi gereğidir. Zira doğrusal eniyileme süreci sonunda elde edilen noktanın gerçek kısıtları karşılama ihtimali zayıftır. Zayıf yönlerine rağmen yöntemin ikilenik formdaki sorunların çözümünde etkin bir yaklaşım olduğu söylenebilir. Yöntemin uygulamada takip edilecek işlem adımları müteakip maddelerde belirtilmektedir;

<sup>i</sup> Sequential linear programming

<sup>ii</sup> Quadratic programming

<sup>iii</sup> Methods of feasible directions

<sup>iv</sup> Generalized reduced gradient method

<sup>v</sup> Sequential or successive quadratic programming

<sup>vi</sup> Penalty functions

<sup>vii</sup> “Method of approximate programming” or “Cutting plane method”



- ▶ 1. Öncelikle  $\mathbf{x}_0$  başlangıç noktası tayin edilir. Başlangıç noktasının olası çözüm uzayı içerisinde olması zorunlu değildir.
- ▶ 2. Akabinde amaç fonksiyonu ve kısıt denklemleri Taylor serisi yaklaşımı ile doğrusal denklem haline dönüştürülür. Özü itibari ile doğrusal karakter taşıyan amaç veya kısıt denklemlerine herhangi bir işlem yapılmasına gerek yoktur. Her bir fonksiyonun seri açılımında sadece 1inci merteben seri terimleri dikkate alınır;

$$f(\mathbf{x}) \cong f(\mathbf{x}_i) + \nabla f(\mathbf{x}_i)^T (\mathbf{x} - \mathbf{x}_i) \quad (2.151)$$

$$g_j(\mathbf{x}) \cong g_j(\mathbf{x}_i) + \nabla g_j(\mathbf{x}_i)^T (\mathbf{x} - \mathbf{x}_i) \quad (2.152)$$

$$h_k(\mathbf{x}) \cong h_k(\mathbf{x}_i) + \nabla h_k(\mathbf{x}_i)^T (\mathbf{x} - \mathbf{x}_i) \quad (2.153)$$

Açılım denklemlerinde gradyan ifadesinin bulunması ve bu ifadenin de fonksiyonun kendisine bağlı olarak doğrusal olmayan terimler içerebileceği düşünülebilir. Ancak unutulmamalıdır ki bahse konu terimler denklem niteliğinde olmayıp, sayısal olarak  $\mathbf{x}_i$  noktasındaki eğim değerini göstermektedir.

- ▶ 3. Doğrusallaştırılan amaç ile kısıt denklemlerine ilave olarak, eğer orijinal sorunda tanımlanmamış ise, her bir değişken için alt ve üst değer kısıtları belirlenir ve orijinal sorun doğrusal karaktere dönüştürülmüş olarak aşağıdaki gibi tekrar tanımlanır;

$$\min_{\mathbf{x}} f(\mathbf{x}_i) + \nabla f(\mathbf{x}_i)^T (\mathbf{x} - \mathbf{x}_i) \quad (2.154)$$

Kısıt

$$\begin{aligned} g_j(\mathbf{x}_i) + \nabla g_j(\mathbf{x}_i)^T (\mathbf{x} - \mathbf{x}_i) &\leq 0, & j = 1, 2, \dots, n \\ h_k(\mathbf{x}_i) + \nabla h_k(\mathbf{x}_i)^T (\mathbf{x} - \mathbf{x}_i) &= 0, & k = 1, 2, \dots, m \\ \mathbf{x}_i^a &\leq \mathbf{x}_i \leq \mathbf{x}_i^u \end{aligned}$$

- ▶ 4. Tanımlanan alt problem, uygun bir doğrusal programlama yöntemiyle çözümlenerek olası bir çözüm noktasına ulaşılır. Genellikle çözümde “simplex” yöntemi kullanılır.
- ▶ 5. Eniyileme süreci nihayetinde elde edilen en uygun nokta  $\mathbf{x}_{i+1}$  olarak dikkate alınır ve bu noktanın kısıtları sağlayıp sağlamadığı kontrol edilir. Unutulmamalıdır ki yöntem sayısal bir yöntem olup diğer sayısal yöntemlerde olduğu üzere kısıt şartlarının sağlanıp sağlanmadığı belli bir hassasiyet değerine göre karşılaştırılır. Eğer yeni nokta;



$$g_j(\mathbf{x}_{i+1}) \leq \epsilon \quad (2.155)$$

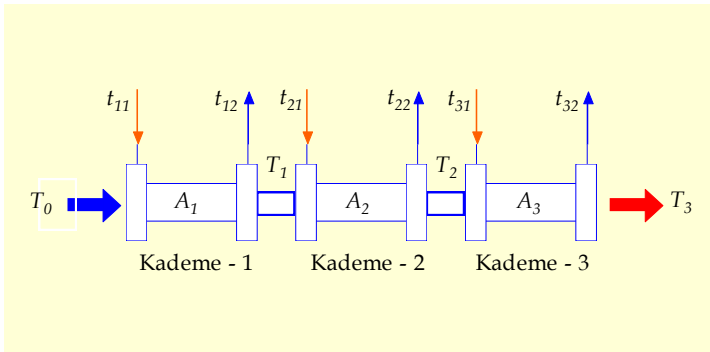
$$h_k(\mathbf{x}_{i+1}) \leq \epsilon \quad (2.156)$$

şartlarını sağlıyor ise ulaşılan nokta optimal nokta olarak dikkate alınır ve süreç sonlandırılır.

- 6. Yeni noktanın, kısıt şartlarını istenen hassasiyette sağlamaması halinde süreç yeniden başlatılır. Ancak tüm denklemlerin tekrar dönüştürülmesi yerine, en fazla ihlale neden olan kısıt belirlenir ve seri yaklaşımı ile yeni bir kısıt denklemini teşkil edilir. Yeni kısıt denkleminin teşkilinde  $\mathbf{x}_{i+1}$  noktası dikkate alınır. Azami soruna yol açan kısıtın, örneğin  $p$ 'ncinci kısıt, eşitlik kısıtı olması halinde  $m + 1$ 'inci kısıt denklemini olarak;

$$h_{m+1}(\mathbf{x}) = h_p(\mathbf{x}_{i+1}) + \nabla h_p(\mathbf{x}_{i+1})^T (\mathbf{x} - \mathbf{x}_{i+1}) = 0 \quad (2.157)$$

ifadesi teşkil edilir. İlave kısıtın da dikkate alındığı doğrusal programlama sorunu, 4üncü adımdan itibaren tekrar çözülür ve bu işlemler istenen hassasiyette çözüme ulaşıncaya kadar devam ettirilir. İfade edilen yöntemin daha iyi anlaşılabilmesi için Şekil 2.81'deki gibi çok kademeli bir ısı değiştiricisinin eniyileme sürecini, ardışık doğrusal programlama işlem adımlarını takip ederek çözmek mümkündür.



Şekil 2.81 Üç kademeli ısı değiştiricisi diyagramı.



Şekil 2.82 Örnek ısı değiştiricisi <sup>(60)</sup>.

<sup>i</sup> Multistage heat exchanger



Buna göre belli bir debi ( $D$ ) ve özgül ısı ( $c_p$ ) değerine sahip soğuk akışkan, birbirine seri olarak bağlanan üç kademeli bir ısı değiştiricisine  $T_0$  sıcaklığında girmekte ve çıkışta da aynı akışkanın  $T_3$  sıcaklığında olması istenmektedir. Isı değiştiricisine ait her kademeye ısıtma amacıyla gönderilen sıcak akışkanın giriş sıcaklıkları  $t_{i1}, i = 1,2,3$  ve kademeye ait ısı iletim katsayıları  $U_i, i = 1,2,3$  bilinmektedir. Isı değiştiricisinin tasarımındaki amaç, istenen  $T_3$  sıcaklığını sağlayan en ucuz ısı değiştiricisinin üretilmesidir. Üretim maliyetini etkileyen en önemli unsur, her bir kademedenin ebadını belirleyen ısı iletim yüzey alanıdır. Buradan hareketle eniyileme sürecine ait amaç fonksiyonunu;

$$\min f = A_1 + A_2 + A_3$$

şeklinde ifade edebiliriz. Fonksiyonda  $A_i$ , her bir kademeye ait yüzey alan değeridir. Tasarıma ait verili değerler müteakip çizelgede ifade edilmektedir;

$T_0$	100 °F	$t_{11}$	300 °F	$U_1$	120 B. t. u./hr - ft <sup>2</sup> - °F
$T_3$	500 °F	$t_{21}$	400 °F	$U_2$	80 B. t. u./hr - ft <sup>2</sup> - °F
$Dc_p$	10 <sup>5</sup> B. t. u./hr - °F	$t_{31}$	600 °F	$U_3$	40 B. t. u./hr - ft <sup>2</sup> - °F

Diğer taraftan tasarıma ait kısıtları, enerjinin korunumu ilkesi ve ısı iletimine ait denklemleri kullanarak yazabiliriz. Buna göre soğuk akışkanın soğurduğu ısı akısı, sıcak akışkanın yaydığı ısı akısından küçük veya ideal halde ona eşit olmalıdır. Bu ilke her bir kademe için yazılırsa;

$$Dc_p(T_i - T_{i-1}) \leq Dc_p(t_{i1} - t_{i2}), \quad i = 1,2,3$$

şeklinde üç kısıt elde edilebilir. Benzer şekilde ısı iletim denklemlerinin her kademe için dikkate alınması durumunda;

$$Dc_p(T_i - T_{i-1}) \leq U_i A_i (t_{i2} - T_{i-1}), \quad i = 1,2,3$$

şeklindeki ilave üç kısıt denklemi daha yazılabilir. İfade edilen altı kısıt denklemi, bilinen veriler dikkate alınarak kolayca açık halde yazılabilir. Örneğin ilk ısı akısı denkleminde hareketle;

$$\begin{aligned} Dc_p(T_1 - T_0) &\leq Dc_p(t_{11} - t_{12}) \\ (T_1 + t_{12}) &\leq (t_{11} + T_0) \end{aligned}$$



$$\frac{(T_1 + t_{12})}{(t_{11} + T_0)} = \frac{(T_1 + t_{12})}{(300 + 100)} \leq 1$$

işlemleri sonucunda:

$$g_1 = 0.0025(T_1 + t_{12}) - 1 \leq 0$$

kısıt denklemi elde edilmiş olur. Benzer işlemler diğer denklemlere uygulandığında aşağıdaki kısıt denklemleri kolayca yazılabilir;

$$\begin{aligned} g_2 &= 0.0025(-T_1 + T_2 + t_{22}) - 1 \leq 0 \\ g_3 &= 0.01(-T_2 + t_{32}) - 1 \leq 0 \\ g_4 &= 100A_1 - A_1t_{12} + 833.332T_1 - 83,333.33 \leq 0 \\ g_5 &= A_2T_1 - A_2t_{22} - 1250T_1 + 1250T_2 \leq 0 \\ g_6 &= A_3T_2 - A_3t_{32} - 2500T_2 + 1,250,000 \leq 0 \end{aligned}$$

Tanımlanan soruna göre toplamda sekiz tasarım değişkeni bulunmakta olup bu parametreler;  $A_1, A_2, A_3, T_1, T_2, t_{12}, t_{22}$  ve  $t_{32}$  şeklindedir. Problemin tanımlanması sonrasında çözüm adına yapılacak ilk işlem, başlangıç noktasının tayin edilmesidir. Başlangıç noktası olarak;

$$\mathbf{x}_0^T = [5000 \ 5000 \ 5000 \ 200 \ 350 \ 150 \ 225 \ 425]$$

yönevi dikkate alınabilir. Ayrıca çözüm hassasiyeti olarak  $\epsilon = 0.01$  değeri kullanılabilir. Akabinde amaç fonksiyonu ve kısıt denklemleri, Taylor serisi yaklaşımı ile doğrusal denklem haline dönüştürülür. Burada amaç fonksiyonu ve bazı kısıt denklemleri orijinal hali ile zaten doğrusaldır. Dolayısıyla doğrusal olmayan  $g_4, g_5$  ve  $g_6$  kısıtlarının başlangıç noktası baz alınarak seri açılımı yapılabilir. Buna göre;

$$g_4(\mathbf{x}) \cong g_4(\mathbf{x}_0) + \nabla g_4(\mathbf{x}_0)^T(\mathbf{x} - \mathbf{x}_0)$$

$$g_4(\mathbf{x}_0) = 100 \times 5000 - 5000 \times 150 + 833.332 \times 200 - 83,333.33 = -0.166 \times 10^6$$

$$\nabla g_4(\mathbf{x}_0)^T = \left[ \frac{\partial g_4}{\partial A_1} \frac{\partial g_4}{\partial A_2} \frac{\partial g_4}{\partial A_3} \frac{\partial g_4}{\partial T_1} \frac{\partial g_4}{\partial T_2} \frac{\partial g_4}{\partial t_{12}} \frac{\partial g_4}{\partial t_{22}} \frac{\partial g_4}{\partial t_{32}} \right]$$

$$\nabla g_4(\mathbf{x}_0)^T = [100 - t_{12} \ 0 \ 0 \ 833.332 \ 0 - A_1 \ 0 \ 0] = [-50 \ 0 \ 0 \ 833.332 \ 0 - 5000 \ 0 \ 0]$$

$$g_4(\mathbf{x}) \cong 833.332T_1 - 50A_1 - 5000t_{12} + 0.6666 \times 10^6$$

elde edilir.



Benzer işlemler diğer kısıt denklemleri için uygulanırsa;

$$\begin{aligned} g_5(\mathbf{x}) &\cong 3750T_1 - 25A_2 + 1250T_2 - 5000t_{22} + 125000 \\ g_6(\mathbf{x}) &\cong 2500T_2 - 75A_3 - 5000t_{32} + 1.625 \times 10^6 \end{aligned}$$

denklemlerine ulaşılır. Doğrusallaştırılan kısıt denklemlerine ilave olarak, her bir değişken için alt ve üst değer kısıtları belirlenebilir. Buna göre tasarım değişkenleri için müteakip sınır değerler dikkate alınabilir;

$$\begin{aligned} 100 &\leq A_1 \leq 10,000 \\ 1,000 &\leq A_i \leq 10,000 \quad i = 2,3 \\ 10 &\leq T_i \leq 1,000 \quad i = 1,2,3 \\ 10 &\leq t_{i2} \leq 1,000 \quad i = 1,2,3 \end{aligned}$$

Yapılan değişiklikler ve belirlemeler sonrası orijinal sorun, doğrusal karaktere dönüştürülmüş olarak tekrar tanımlanır;

$$\min f = A_1 + A_2 + A_3$$

Kısıt

$$\begin{aligned} g_1 &= 0.0025(T_1 + t_{12}) - 1 \leq 0 & g_7(A_1) &\geq 100 & g_{13}(T_1) &\geq 10 & g_{19}(t_{22}) &\geq 10 \\ g_2 &= 0.0025(-T_1 + T_2 + t_{22}) - 1 \leq 0 & g_8(A_1) &\leq 10,000 & g_{14}(T_1) &\leq 1,000 & g_{20}(t_{22}) &\leq 1,000 \\ g_3 &= 0.01(-T_2 + t_{32}) - 1 \leq 0 & g_9(A_2) &\geq 1,000 & g_{15}(T_2) &\geq 10 & g_{21}(t_{32}) &\geq 10 \\ g_4(\mathbf{x}) &\cong 833.332T_1 - 50A_1 - 5000t_{12} + 0.6666 \times 10^6 \leq 0 & g_{10}(A_2) &\leq 10,000 & g_{16}(T_2) &\leq 1,000 & g_{22}(t_{32}) &\leq 1,000 \\ g_5(\mathbf{x}) &\cong 3,750T_1 - 25A_2 + 1,250T_2 - 5,000t_{22} + 125,000 \leq 0 & g_{11}(A_3) &\geq 1,000 & g_{17}(t_{12}) &\geq 10 & & \\ g_6(\mathbf{x}) &\cong 2,500T_2 - 75A_3 - 5,000t_{32} + 1.625 \times 10^6 \leq 0 & g_{12}(A_3) &\leq 10,000 & g_{18}(t_{12}) &\leq 1,000 & & \end{aligned}$$

Tanımlanan bu problemi uygun bir “simplex” yöntemi aracını kullanarak çözmek mümkündür<sup>(61)</sup>. Buna göre sorunun ilk çözümü sonucunda;

$$\mathbf{x}_1^T = [100 \quad 1000 \quad 3337.142 \quad 229.42 \quad 349.88 \quad 170.57 \quad 279.54 \quad 449.885]$$

noktasına ulaşılabilir. Bu noktanın amaç fonksiyon değeri 4437.142 ft<sup>2</sup> olarak elde edilir. Elde edilen noktanın, orijinal kısıtları istenen hassasiyette karşılama durumu değerlendirildiğinde ilk üç kısıtın karşılandığı görülmektedir;

$$\begin{aligned} g_1 &= 0.0025(T_1 + t_{12}) - 1 = -2.5 \times 10^{-5} \leq \epsilon = 0.01 \\ g_2 &= 0.0025(-T_1 + T_2 + t_{22}) - 1 = 0 \leq \epsilon = 0.01 \\ g_3 &= 0.01(-T_2 + t_{32}) - 1 = 18.8 \times 10^{-4} \leq \epsilon = 0.01 \end{aligned}$$



Buna mukabil, beklendiği üzere sonraki üç kısıttan sadece  $g_4(\mathbf{x})$  kısıtının karşılandığı, diğer iki kısıtın ise karşılanmadığı gözlemlenmektedir;

$$\begin{aligned} g_4 &= 100A_1 - A_1t_{12} + 833.332T_1 - 83,333.33 = -7,057.13 \leq \epsilon = 0.01 \\ g_5 &= A_2T_1 - A_2t_{22} - 1250T_1 + 1250T_2 = 100,455 \geq \epsilon = 0.01 \\ g_6 &= A_3T_2 - A_3t_{32} - 2500T_2 + 1,250,000 = 40,100.77 \geq \epsilon = 0.01 \end{aligned}$$

İlave olarak geri kalan sınır değer kısıtlarının karşılandığı söylenebilir. Hâlihazırdaki kısıt ihlalleri arasında en büyük ihlale neden olan kısıt,  $g_5(\mathbf{x})$  kısıtı olup bu kısıt yeni noktada tekrar birinci mertebeden Taylor serisine açılarak yeni bir kısıt teşkiline gidilir. Dolayısıyla;

$$g_{23}(\mathbf{x}) \cong g_5(\mathbf{x}_1) + \nabla g_5(\mathbf{x}_1)^T(\mathbf{x} - \mathbf{x}_1)$$

yaklaşımı dikkate alınarak;

$$g_{23}(\mathbf{x}) \cong 1250T_2 - 250T_1 - 50.12A_2 - 1000t_{22} + 50,120$$

denklemleri doğrusal eniyileme sorununa eklenir ve yeni alt problem tekrar tanımlanır;

$$\min f = A_1 + A_2 + A_3$$

Kısıt

$$\begin{array}{llll} g_1 = 0.0025(T_1 + t_{12}) - 1 \leq 0 & g_7(A_1) \geq 100 & g_{13}(T_1) \geq 10 & g_{19}(t_{22}) \geq 10 \\ g_2 = 0.0025(-T_1 + T_2 + t_{22}) - 1 \leq 0 & g_8(A_1) \leq 10,000 & g_{14}(T_1) \leq 1,000 & g_{20}(t_{22}) \leq 1,000 \\ g_3 = 0.01(-T_2 + t_{32}) - 1 \leq 0 & g_9(A_2) \geq 1,000 & g_{15}(T_2) \geq 10 & g_{21}(t_{32}) \geq 10 \\ g_4(\mathbf{x}) \cong 833.332T_1 - 50A_1 - 5000t_{12} + 0.6666 \times 10^6 \leq 0 & g_{10}(A_2) \leq 10,000 & g_{16}(T_2) \leq 1,000 & g_{22}(t_{32}) \leq 1,000 \\ g_5(\mathbf{x}) \cong 3,750T_1 - 25A_2 + 1,250T_2 - 5,000t_{22} + 125,000 \leq 0 & g_{11}(A_3) \geq 1,000 & g_{17}(t_{12}) \geq 10 & \\ g_6(\mathbf{x}) \cong 2,500T_2 - 75A_3 - 5,000t_{32} + 1.625 \times 10^6 \leq 0 & g_{12}(A_3) \leq 10,000 & g_{18}(t_{12}) \leq 1,000 & \\ g_{23}(\mathbf{x}) \cong 1250T_2 - 250T_1 - 50.12A_2 - 1000t_{22} + 50,120 \leq 0 & & & \end{array}$$

Tanımlanan yeni problemin “simplex” yöntemi ile çözümü sonucunda;

$$\mathbf{x}_2^T = [100 \quad 1000 \quad 4825.395 \quad 229.42 \quad 305.23 \quad 170.57 \quad 324.19 \quad 405.23]$$

noktasına ulaşılabilir. Bu noktanın amaç fonksiyon değeri  $5925.395 \text{ ft}^2$  olarak elde edilir. İlk taramada olduğu gibi benzer işlemler takip edilerek müteakip optimal noktaya ve amaç fonksiyonu değerine ulaşılabilir<sup>(59)</sup>;

$$\mathbf{x}^{*,T} = [567 \quad 1357 \quad 5125 \quad 181 \quad 295 \quad 219 \quad 286 \quad 395], f^* = 7049 \text{ ft}^2$$





## (2) İkilenik programlama

İkilenik programlama yöntemi, ikilenik formdaki her hangi bir amaç fonksiyonu ile doğrusal karakterdeki eşitlik veya eşitsizlik kısıtlarını içeren eniyileme sorunlarını kapsar. Yöntem, her ne kadar ikilenik formdaki fonksiyonları içerse de daha yüksek dereceden fonksiyonların ikinci mertebeden Taylor serisine açılımı yolu ile ikilenik formda yaklaşık olarak ifade edilebilmeleri nedeniyle başka sorunların çözümünde veya yöntemlerin geliştirilmesinde dikkate alınabilmektedir. Benzer husus, birinci mertebeden Taylor serisine açılımı yolu ile kısıtlar için de geçerlidir. Genel olarak ikilenik programlama sorunu;

$$\min f(x) = \frac{1}{2}x^T Ax + b^T x \quad (2.158)$$

Kısıt

$$\begin{aligned} Cx &\leq d \\ x &\geq 0 \end{aligned}$$

şeklinde ifade edilebilir. Bu ifadelerdeki yöney veya matrisleri aşağıdaki gibi daha açık halde yazabiliriz;

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \quad d = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_m \end{bmatrix}$$

$$C = \begin{bmatrix} c_{11} & c_{12} & \dots & c_{1n} \\ c_{21} & c_{22} & \dots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m1} & c_{n2} & \dots & c_{mn} \end{bmatrix} \quad x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

İkilenik programlama sorununu değişik yöntemler kullanarak çözmek mümkündür. Bununla beraber genellikle Lagrange çarpanları yöntemi kullanılarak çözüme gidilmesi yaygındır. Çözüm sürecine başlamadan önce dikkat edilmesi gereken diğer bir husus, eşitsizlik kısıtlarının eşitlik kısıtlarına dönüştürülmesidir. Zira sonraki aşamalara ait çözüm arayışında uygun bir doğrusal programlama yöntemi, örneğin “simplex” yöntemi kullanılabilir. Bu meyanda eniyileme sorunu;

$$\min f(x) = \frac{1}{2}x^T Ax + b^T x \quad (2.159)$$

Kısıt

$$\begin{aligned} C^T x + s^T s &= d \\ -x + t^T t &= 0 \end{aligned}$$



formunda yeniden yazılabilir. Burada  $\mathbf{s}$  arttırma amaçlı,  $\mathbf{t}$  eksiltme amaçlı olarak denklemlere eklenen yapay değişkenlerdir. Söz konusu değişkenlerin kare şeklinde tanımlanmalarının nedeni, değerlerinin kendilerinden bağımsız olarak her zaman pozitif olmaları ve ayrıca türetilmeleri halinde kendi terimlerini muhafaza etmeleridir. Lagrange çarpanları yöntemi dikkate alındığında Lagrange fonksiyonunu;

$$L(\mathbf{x}, \mathbf{s}, \mathbf{t}, \boldsymbol{\lambda}, \boldsymbol{\theta}) = \frac{1}{2} \mathbf{x}^T \mathbf{A} \mathbf{x} + \mathbf{b}^T \mathbf{x} + \sum_{i=1}^m \lambda_i (C_{i,1:n}^T \mathbf{x} + s_i^2 - d_i) + \sum_{j=1}^n \theta_j (-x_j + t_j^2) \quad (2.160)$$

şeklinde yazmak mümkündür. Tanımlanan bu fonksiyonun optimal noktada birinci mertebeden türevinin sıfıra eşitlenmesi gereği düşünüldüğünde;

$$\frac{\partial L}{\partial x_j} = \sum_{i=1}^n a_{i,j} x_i + b_j + \sum_{i=1}^m (\lambda_i c_{i,j} - \theta_j) = 0, \quad j = 1:n \quad (2.161)$$

$$\frac{\partial L}{\partial s_i} = 2\lambda_i s_i = 0, \quad i = 1:m \quad (2.162)$$

$$\frac{\partial L}{\partial t_j} = 2\theta_j t_j = 0, \quad j = 1:n \quad (2.163)$$

$$\frac{\partial L}{\partial \lambda_i} = C_{i,1:n}^T \mathbf{x} + s_i^2 - d_i = 0, \quad i = 1:m \quad (2.164)$$

$$\frac{\partial L}{\partial \theta_j} = -x_j + t_j^2 = 0, \quad j = 1:n \quad (2.165)$$

denklemleri yazılabilir. Bu adımda elde edilen denklemlerden faydalanılarak bazı güdüleme işlemlerine gidilebilir. Örneğin  $y_i = s_i^2, y_i \geq 0$  tanımlaması sonrasında;

$$C_{i,1:n}^T \mathbf{x} - d_i = -y_i \quad (2.166)$$

elde edilir. Koşullardan ikinci denklemin her iki tarafının  $s_i$  ile çarpılması sonucu;

$$s_i \cdot \left\{ \begin{array}{l} \frac{\partial L}{\partial s_i} = 2\lambda_i s_i = 0 \\ \lambda_i s_i^2 = 0 \\ \lambda_i y_i = 0 \end{array} \right\} \quad (2.167)$$



koşulu yazılabilir. Benzer işlem diğer yapay değişken için tekrarlandığında;

$$t_i \cdot \left\{ \frac{\partial L}{\partial t_j} = 2\theta_j t_j = 0 \right\} \quad (2.168)$$

$$\theta_j t_i^2 = 0$$

eşitliğine ulaşılır. Elde edilen bu tür güdüleme denklemleri ve benzer güdüleme teknikleri diğer temel kısıt denklemlerinde kullanılırsa;

$$\lambda_i \cdot \left\{ \frac{\partial L}{\partial \lambda_i} = C_{i,1:n}^T \mathbf{x} + s_i^2 - d_i = 0 \right\} \quad (2.169)$$

$$\lambda_i (C_{i,1:n}^T \mathbf{x} - d_i) + \lambda_i s_i^2 = 0, \quad \lambda_i s_i^2 = 0$$

$$C_{i,1:n}^T \mathbf{x} - d_i = 0$$

eşitlikleri ve ilave olarak;

$$\theta_j \cdot \left\{ \frac{\partial L}{\partial \theta_j} = -x_j + t_j^2 = 0 \right\} \quad (2.170)$$

$$-\theta_j x_j + \theta_j t_j^2 = 0, \quad \theta_j t_j^2 = 0$$

$$\theta_j x_j = 0$$

denklemleri elde edilir. Bu dönüşümlerden hareketle optimal nokta şartları yeniden yazılırsa;

$$b_j - \theta_j + \sum_{i=1}^n a_{i,j} x_i + \sum_{i=1}^m \lambda_i c_{i,j} = 0, \quad j = 1:n \quad (2.171)$$

$$C_{i,1:n}^T \mathbf{x} - d_i = 0$$

$$\lambda_i y_i = 0$$

$$\theta_j x_j = 0$$

$$x_j, y_i, \lambda_i, \theta_j \geq 0$$

denklemler grubuna ulaşılır.



Bu denklemlere yakinen bakıldığında, toplam  $2(n + m)$  adet değişkenin bulunduğu, buna mukabil  $2(n + m)$  adet denklemin yazılabildiği görülmektedir. Ayrıca temel kısıt denklemlerinin doğrusal karakterde olduğu da söylenebilir. Buradan hareketle, istenen koşulların sağlanabildiği bir doğrusal programlama sorununu tanımlamak ve elde edilen problemi yaygın olarak kullanılan “simplex” algoritması ile çözümlenmek mümkündür. Bu amaçla orijinal amaç fonksiyonundaki değişken sayısı kadar sanal değişken tanımlanarak yeni üretilen doğrusal programlama problemine esas bir amaç fonksiyonu tanımlanır;

$$\min F(\mathbf{z}) = \sum_{j=1}^n z_j \quad (2.172)$$

Buna mukabil, tanımlı  $\mathbf{z}$  değişkenlerinin kısıtlarda görünürlüğüne sağlanması adına ilk kısıt koşuluna eklemeye işlemi gerçekleştirilir;

$$b_j - \theta_j + \sum_{i=1}^n a_{i,j}x_i + \sum_{i=1}^m \lambda_i c_{i,j} + z_j = 0, \quad j = 1:n \quad (2.173)$$

Hülasa, ikilenik programlama problemi Denklem 2.174'deki gibi bir doğrusal programlama sorununa dönüştürülerek çözüm yoluna gidilmiş olur;

$$\min F(\mathbf{z}) = \sum_{j=1}^n z_j \quad (2.174)$$

Kısıt

$$\begin{aligned} b_j - \theta_j + \sum_{i=1}^n a_{i,j}x_i + \sum_{i=1}^m \lambda_i c_{i,j} + z_j &= 0, \quad j = 1:n \\ C_{i,1:n}^T \mathbf{x} - d_i &= 0 \\ x_j, y_i, \lambda_i, \theta_j &\geq 0 \\ \lambda_i y_i &= 0 \\ \theta_j x_j &= 0 \end{aligned}$$

Burada doğrusal olmayan karakterdeki  $\lambda_i y_i = 0$  ve  $\theta_j x_j = 0$  denklemleri, doğrusal programlama probleminin çözümünde doğrudan değil ama dolaylı olarak dikkate alınır. Nihayetinde doğrusal programlama sorununun çözüm yöneyi, aynı zamanda ikilenik programlama sorununun çözüm yöneyidir.

Lagrange çarpanları yöntemi ile ikilenik programlama sorununun çözümü adına takip edilen işlem adımlarını daha iyi kavrayabilmek için müteakip problemi ele alabiliriz. Buna göre;



$$\min f(\mathbf{x}) = x_1^2 + 2x_2^2 - 2x_1x_2 - 4x_1 + 5$$

Kısıt

$$\begin{aligned} 2x_1 + x_2 &\leq 6 \\ x_1 - 4x_2 &\leq 0 \\ x_1, x_2 &\geq 0 \end{aligned}$$

sorununun çözümü bağlamında yapılacak ilk işlem, gerekli yöney ve matrislerin teşkilidir. Verili amaç ve kısıt fonksiyonları dikkate alınarak;

$$f(\mathbf{x}) = [-4 \quad 0] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \frac{1}{2} [x_1 \quad x_2] \begin{bmatrix} 2 & -2 \\ -2 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + 5$$

düzenlemesi ve akabinde kısıt ifadeleri beraber değerlendirildiğinde;

$$\begin{aligned} A &= \begin{bmatrix} 2 & -2 \\ -2 & 4 \end{bmatrix} & \mathbf{b} &= \begin{bmatrix} -4 \\ 0 \end{bmatrix} \\ C &= \begin{bmatrix} 2 & 1 \\ 1 & -4 \end{bmatrix} & \mathbf{d} &= \begin{bmatrix} 6 \\ 0 \end{bmatrix} & \mathbf{x} &= \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \end{aligned}$$

tanımlamaları kolayca yazılabilir. Müteakiben eşitsizlik kısıtlarının eşitlik kısıtı şekline dönüştürülmesi işlemine geçilir. Bu maksatla ilk iki kısıt denklemine  $s_i$  ve sonraki kısıtlara  $t_i$  yapay değişkenleri eklenerek sorun yeniden tanımlanır;

$$\min f(\mathbf{x}) = x_1^2 + 2x_2^2 - 2x_1x_2 - 4x_1 + 5$$

Kısıt

$$\begin{aligned} 2x_1 + x_2 + s_1^2 &= 6 \\ x_1 - 4x_2 + s_2^2 &= 0 \\ -x_1 + t_1^2 &= 0 \\ -x_2 + t_2^2 &= 0 \end{aligned}$$

Yeniden tanımlanan ikilenik programlama problemi, bir doğrusal programlama sorununa dönüştürülmüş olur. Şöyle ki;

$$\min F(\mathbf{z}) = z_1 + z_2$$

Kısıt

$$b_j - \theta_j + \sum_{i=1}^2 a_{i,j}x_i + \sum_{i=1}^2 \lambda_i c_{i,j} + z_j = 0, \quad j = 1:2$$

ifadesinden hareketle;



$$\begin{aligned} -4 - \theta_1 + 2x_1 - 2x_2 + 2\lambda_1 + \lambda_2 &= 0 \\ -\theta_2 - 2x_1 + 4x_2 + \lambda_1 - 4\lambda_2 &= 0 \end{aligned}$$

denklemleri,

$$C_{i,1:n}^T \mathbf{x} - d_i = 0$$

ifadesinden hareketle;

$$\begin{aligned} 2x_1 + x_2 + y_1 &= 6 \\ x_1 - 4x_2 + y_2 &= 0 \end{aligned}$$

denklemleri elde edilmiş olur. Hatırlanacağı üzere burada  $y_1 = s_1^2, y_2 = s_2^2$  olup  $\lambda_i$  ve  $\theta_j$  Lagrange çarpanlarını göstermektedir. Hülasa, doğrusal programlama problemini;

$$\begin{aligned} \text{Kısıt} \quad \min F(\mathbf{z}) &= z_1 + z_2 \\ -\theta_1 + 2x_1 - 2x_2 + 2\lambda_1 + \lambda_2 + z_1 &= 4 \\ -\theta_2 - 2x_1 + 4x_2 + \lambda_1 - 4\lambda_2 + z_2 &= 0 \\ 2x_1 + x_2 + y_1 &= 6 \\ x_1 - 4x_2 + y_2 &= 0 \\ x_1, x_2, y_1, y_2, \lambda_1, \lambda_2, \theta_1, \theta_2 &\geq 0 \\ \lambda_1 y_1 &= 0 \\ \lambda_2 y_2 &= 0 \\ \theta_1 x_1 &= 0 \\ \theta_2 x_2 &= 0 \end{aligned}$$

olarak yazmak mümkündür. Bu problem, “simplex” algoritması kullanılarak çözüldüğünde optimal değerler;

$$\{x_1, x_2, y_1, y_2, \lambda_1, \lambda_2, \theta_1, \theta_2, z_1, z_2\}^* = \{2.46, 1.07, 0, 1.84, 0.61, 0, 0, 0, 0, 0\}$$

$$\begin{aligned} F^* &= 0 \\ f^* &= -1.763 \end{aligned}$$

şeklinde hesaplanmış olur.



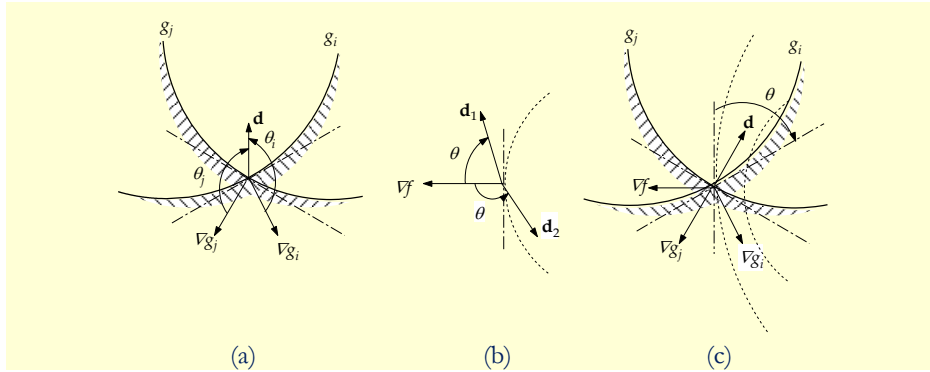


### (3) Uygun doğrultu yöntemleri

Uygun doğrultu yöntemleri, çözüm uzayında bulunan bir başlangıç noktasından hareketle daha iyi bir noktaya ulaşmayı hedefleyen yöntemlerdir. Tekrarlı işlem adımlarını içeren bu tür yöntemlerde, başlangıç noktası önemli olup çözüm uzayının dışında seçilen başlangıç noktası yöntemin temelini oluşturan ölçütleri etkisiz kılar. Bu nedenle uygun bir başlangıç noktasının seçimi önem arz eder. Başlangıç noktasından hareketle bir sonraki uygun nokta;

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha \mathbf{d}_i$$

denklemleri ile elde edilir. Hatırlanacağı üzere burada  $\alpha$  adım büyüklüğünü,  $\mathbf{d}_i$  ise gidilecek doğrultuyu ifade etmekteydi. Eniyileme sürecinde öncelik, uygun doğrultunun tespit edilmesidir. Uygun doğrultunun tespiti, bazı özelliklerin tanımlanması ile mümkündür. Bunlardan ilki seçilen doğrultunun, istenen kısıtları karşılaması gereğidir.



Şekil 2.83 Uygun doğrultunun sağlanması gereken şartların şekilsel gösterimi.

Şekil 2.83(a)'daki çizimden de görüleceği üzere  $g_i$  ve  $g_j$  gibi örnek iki farklı kısıtın etkin olduğu bir konumu dikkate alalım. Bulunulan konuma göre kısıtın gradyan yönü, koşulun sağlanamadığı, tersi ise istenen koşulun sağlanabildiği yön olarak dikkate alınır. Bu durumda kesikli çizgi ile gösterilen hatların teğet hatlar olduğu, bu hatlar ile gradyan hatlarının arasında kalan bölgenin, istenen kısıtları sağlayamadığı kolayca gözlemlenebilir. Dolayısıyla gradyan hattı ile gidilmesi düşünülen doğrultu arasında tanımlanan  $\theta$  açısının, doksan dereceden fazla olması gerektiği söylenebilir. Bu ifade dikkate alındığında;



$$\begin{aligned} \mathbf{d}^T \nabla g_i &\leq 0 \\ \mathbf{d}^T \nabla g_j &\leq 0 \end{aligned} \quad (2.175)$$

şartları yazılabilir. Hatırlanacağı üzere iki yöneyin sayısal çarpımı, şiddet değerleri ile aradaki açısının kosinüs değerine bağlıdır,  $\mathbf{d}^T \nabla g_i = |\mathbf{d}| \cdot |\nabla g_i| \cos \theta_i$ . Dolayısıyla açının  $90 \leq \theta \leq 270$  olması halinde elde edilecek çarpım değeri, sıfır veya negatif olacaktır. Buradan hareketle gidilecek doğrultunun istenen kısıtları karşılaması için;

$$\mathbf{d}^T \nabla \mathbf{g} \leq 0 \quad (2.176)$$

koşulunu sağlaması gerektiği söylenebilir. Uygun doğrultu seçiminde dikkate alınan diğer bir husus, amaç fonksiyon değerindeki değişimdir. Minimizasyon amaçlı bir eniyileme sürecinde gidilecek doğrultunun, amaç fonksiyon değerinde bir azalmaya neden olması istenir. Şekil 2.83(b)'deki çizimden de görüleceği üzere amaç fonksiyonunun gradyan doğrultusu, daima fonksiyon değerinde büyümeye neden olur. Örneğin şekildeki  $\mathbf{d}_1$  doğrultusu dikkate alındığında, gradyan doğrultusu ile teğet doğrultu arasında bulunması nedeniyle, atılacak adımın amaç fonksiyonunda büyümeye neden olacağı görülebilir. Buna mukabil,  $\mathbf{d}_2$  doğrultusunun dikkate alınması halinde aradaki açının, doksan dereceden büyük olması ( $\theta > 90$ ) nedeni ile amaç fonksiyonunda azalmaya neden olacağı gözlemlenebilir. Hülasa, seçilen doğrultunun amaç fonksiyonunda azalmaya neden olması isteniyorsa aradaki açı değerinin, tıpkı kısıt şartında olduğu gibi,  $90 < \theta < 270$  aralığında bulunması, dolayısıyla doğrultu ile amaç fonksiyonu gradyanı arasında;

$$\mathbf{d}^T \nabla f < 0 \quad (2.177)$$

koşulunun sağlanması gerekir. Şunu da ifade edelim ki, kısıtların aktif olmadığı durumlarda yön olarak doğrudan doğruya amaç fonksiyonunun gradyanının tersi yön dikkate alınabilir. Diğer taraftan geometrik olarak izah edilen amaç ve kısıt fonksiyonu şartlarını, Taylor serisinin 1inci mertebeden açılımı yoluyla da gözlemlemek mümkündür. Amaç fonksiyonu ile kısıt denklemlerinin 1inci mertebeden seri açılımları;

$$f(\mathbf{x}_{i+1}) = f(\mathbf{x}_i) + \nabla f(\mathbf{x}_i)^T (\mathbf{x}_{i+1} - \mathbf{x}_i) \quad (2.178)$$

$$g_j(\mathbf{x}_{i+1}) = g_j(\mathbf{x}_i) + \nabla g_j(\mathbf{x}_i)^T (\mathbf{x}_{i+1} - \mathbf{x}_i) \leq 0 \quad (2.179)$$

denklemleri ile ifade edilebilir. Doğal olarak sorunun minimizasyon amaçlı olması durumunda her bir denklem için;





$$f(\mathbf{x}_{i+1}) - f(\mathbf{x}_i) < 0 \quad (2.180)$$

$$g_j(\mathbf{x}_{i+1}) - g_j(\mathbf{x}_i) \leq 0 \quad (2.181)$$

şartlarının sağlanması beklenir. Bu beklenti;

$$\nabla f(\mathbf{x}_i)^T (\mathbf{x}_{i+1} - \mathbf{x}_i) < 0 \quad (2.182)$$

$$\nabla g_j(\mathbf{x}_i)^T (\mathbf{x}_{i+1} - \mathbf{x}_i) \leq 0 \quad (2.183)$$

koşullarını da beraberinde getirir. Dikkat edileceği üzere burada  $(\mathbf{x}_{i+1} - \mathbf{x}_i)$  ifadesi aynı zamanda iki nokta arasındaki doğrultu yöneyini ifade etmektedir. Şekil 2.83(c)'deki çizimde görülen  $\mathbf{d}$  doğrultu ve  $\theta$  açı değer bölgesi, gerek kısıt ve gerekse de amaç fonksiyonu bazlı şartları sağlayabilen olası çözüm uzayını göstermektedir.

Doğrultunun belirlenmesi sonrasında açığa kavuşturulması gereken diğer önemli konu da adım büyüklüğüdür. Genellikle adım büyüklüğü, seçilen doğrultuda amaç fonksiyonundaki azami azalmayı sağlayacak değer olarak dikkate alınır. Ancak önemli olan, atılan adım sonrasında hala kısıtları sağlayabilmek ve dolayısıyla çözüm uzayında kalabilmektir.

Uygun doğrultu ile adım büyüklüğünün bulunması sadedinde değişik yöntemlerden bahsedilebilir. Burada Zoutendijk'in uygun doğrultu yöntemine değinilmektedir. Yöntemin kökeni 1960'lı yıllara dayanmakta olup Zoutendijk'in yayımladığı “Uygun doğrultulara dayalı yöntemler” isimli kitabı ile literatüre kazandırılmıştır<sup>(62)</sup>. Yaklaşımın temelinde, daraltılmış çözüm uzayının doğrusallaştırılması ve uygun bir doğrusal programlama yöntemi kullanarak normalize edilmiş doğrultunun elde edilmesi fikri bulunmaktadır. Amaç fonksiyonunda azalmayı ve aynı zamanda kısıt şartlarını da sağlayan doğrultuda atılacak adım büyüklüğü, yine amaç fonksiyonu ve kısıtlara göre tek değişkenli tarama yöntemi kullanılarak bulunur. Zoutendijk'in uygun doğrultu yöntemine ait işlem adımları aşağıdaki gibi sıralanabilir;

- ▶ 1. Öncelikle olası çözüm uzayı içinde bulunan  $n$  boyutlu bir  $\mathbf{x}_1$  başlangıç noktası seçilir. Başlangıç noktasına ilave olarak  $\epsilon_1$  yakınsama ölçütü ve  $\epsilon_2$  kısıt fonksiyonları için aktif olup olmama hassasiyeti değerleri öngörülür.

---

<sup>i</sup> Methods of feasible directions





- 2. Başlangıç noktasının amaç fonksiyon değeri ile kısıt fonksiyonlarının bu noktada etkin olup olmadığı kontrol edilir. Eğer öngörülen hassasiyet çerçevesinde hiçbir kısıt fonksiyonu aktif değilse gidilecek yön, amaç fonksiyonunun gradyanının tersi istikamet olarak belirlenir. Ayrıca bu istikamet normalizasyonu gerçekleştirilir. Normalizasyonda azami yöney değeri, bölen olarak dikkate alınabilir;

$$\mathbf{d}_i = -\frac{1}{\max(|\nabla f(x_i)|)} \nabla f(x_i), \quad i = 1 \quad (2.184)$$

- 3. Herhangi bir kısıt etkinliği söz konusu ise aktif kısıt seti,  $J$  oluşturulur;

$$g_k(\mathbf{x}_i) \leq \epsilon_2, \quad k \in J \quad (2.185)$$

- 4. Akabinde doğrultu tayini için doğrusal programlama problemi teşkil edilir. Bu teşkilde temel çıkış noktası, daha önce ifade edilen olası çözüm uzayının sınırlarıdır. Doğrusal programlama sorunu Denklem 2.186'daki gibi teşkil edilir;

$$\begin{aligned} & \min -\beta \\ \text{Kısıt} & \sum_{p=1}^n t_p \frac{\partial f(\mathbf{x})}{\partial x_p} + \beta \leq \sum_{p=1}^n \frac{\partial f(\mathbf{x})}{\partial x_p} \\ & \sum_{p=1}^n t_p \frac{\partial g_k(\mathbf{x})}{\partial x_p} + \theta_k \beta \leq \sum_{p=1}^n \frac{\partial g_k(\mathbf{x})}{\partial x_p}, \quad k \in J \\ & 0 \leq t_p \leq 2 \\ & \beta > 0 \end{aligned} \quad (2.186)$$

Burada tasarım yöneyi  $[\mathbf{t} \ \beta]$  olup  $\theta_k$  olarak genellikle 1 alınır. İfade edilen sorunun çözümü sonucu elde edilen optimal yöney kullanılarak yön tayini yapılır;

$$\mathbf{d}_i = \mathbf{t}^* - 1 \quad (2.187)$$

- 5. Yön tayini sonrasında adım büyüklüğünün öngörülmesi adımına geçilir. Burada öncelikle kısıt fonksiyonları dikkate alınır. Tahmin edileceği üzere çok sayıda kısıtın dolması nedeniyle çok sayıda adım büyüklüğü değeri elde edilir. Bu değerler arasından asgari adım büyüklüğü değeri dikkate alınır;





$$\bar{\alpha} = \min(\alpha: g_j(\mathbf{x}_i + \alpha \mathbf{d}_i) = 0, \alpha \geq 0) \quad (2.188)$$

Bu değer, esasen adım büyüklüğünün üst sınırını ifade etmektedir. Dolayısıyla  $0 \leq \alpha \leq \bar{\alpha}$  olacak şekilde, amaç fonksiyonu dikkate alınarak adım büyüklüğü elde edilir;

$$\frac{\partial f(\alpha)}{\partial \alpha} = 0$$

- 6. Gidilecek yönün ve adım büyüklüğünün eldesi sonrasında, bir sonraki nokta kolayca hesaplanabilir;

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha \mathbf{d}_i$$

- 7. Ulaşılan noktanın en uygun nokta olup olmadığını kontrol etmek amacıyla değişik yakınsama ölçütlerini dikkate almak mümkündür. Genellikle amaç fonksiyonu veya tasarım yöneynindeki değişiklik mertebesine göre yakınsama kontrolü yapılabilir;

$$\left| \frac{f(\mathbf{x}_i) - f(\mathbf{x}_{i+1})}{f(\mathbf{x}_i)} \right| \leq \epsilon_1$$

veya

$$\|\mathbf{x}_{i+1} - \mathbf{x}_i\| \leq \epsilon_1$$

- 8. İstenilen hassasiyette çözüme ulaşıldığı durumda tarama süreci sonlandırılır. Aksi halde 2nci adıma gidilerek tekrarlı işlem adımlarına devam edilir.

İfade edilen yöntemin daha iyi anlaşılabilmesi için örnek bir eniyileme sürecini<sup>(63)</sup> Zoutendijk'in uygun doğrultu yöntemi işlem adımlarını takip ederek çözmek mümkündür. Buna göre;

$$\begin{aligned} \min f(\mathbf{x}) &= x_1^2 + x_2^2 - 4x_1 - 4x_2 + 8 \\ \text{Kısıt} \quad g(\mathbf{x}) &= x_1 + 2x_2 - 4 \leq 0 \end{aligned}$$

sorununun çözümü bağlamında, başlangıç noktası olarak  $\mathbf{x}^{(1)} = [0 \ 0]^T$  noktasını, yakınsama ölçütü olarak  $\epsilon_1 = 0.01$  ve kısıt aktivitesinin değerlendirilmesi ölçütü olarak da  $\epsilon_2 = \pm 0.05$  aralık değerlerini dikkate alalım. Başlangıç noktasının amaç fonksiyon değeri ile kısıtı sağlayıp sağlamadığını öncelikle değerlendirebiliriz;





$$\begin{aligned} f(\mathbf{x}^{(1)}) &= 8 \\ g(\mathbf{x}^{(1)}) &= -4 \leq 0 \end{aligned}$$

Buna göre başlangıç noktasında kısıt koşulu sağlanmakta olup kısıt aktif değildir. Dolayısıyla gidilecek yönü, doğrudan doğruya amaç fonksiyonunun gradyan değerine göre şekillendirmek mümkündür;

$$\nabla f(x) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 2x_1 - 4 \\ 2x_2 - 4 \end{bmatrix}$$

$$\mathbf{d}^{(1)} = -\frac{1}{\max(|\nabla f(\mathbf{x}_1)|)} \nabla f(\mathbf{x}_1) = -\frac{1}{4} \begin{bmatrix} -4 \\ -4 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Yön tayini sonrasında adım büyüklüğünün öngörülmesi adımına geçilir. Bu işlemde önemli olan, kısıt fonksiyonudur;

$$\begin{aligned} \mathbf{x}^{(2)} &= \mathbf{x}^{(1)} + \alpha \mathbf{d}^{(1)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \alpha \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha \\ \alpha \end{bmatrix} \\ g(\mathbf{x}^{(2)}) &= 3\alpha - 4 = 0 \\ \bar{\alpha} &= \frac{4}{3} \end{aligned}$$

Adım büyüklüğü için üst sınırın belirlenmesi sonrasında, amaç fonksiyonu dikkate alınarak adım büyüklüğü değeri gözlemlenir;

$$\begin{aligned} f(\mathbf{x}^{(2)}) &= 2\alpha^2 - 8\alpha + 8 \\ \frac{\partial f(\alpha)}{\partial \alpha} &= 4\alpha - 8 = 0 \\ \alpha &= 2 \end{aligned}$$

Görüldüğü üzere bu değer, kısıt koşulu bazlı adım büyüklüğü değerine göre daha büyüktür. Dolayısıyla adım büyüklüğü olarak üst sınır olan 4/3 değeri dikkate alınmalıdır. Yeni adım büyüklüğüne göre yeni tasarım noktası;

$$\mathbf{x}^{(2)} = \begin{bmatrix} \alpha \\ \alpha \end{bmatrix} = \begin{bmatrix} 4/3 \\ 4/3 \end{bmatrix}$$

şeklinde elde edilir.



Ulaşılan yeni noktanın en uygun nokta olup olmadığı, yakınsama kontrolü ile denetlenebilir;

$$\left| \frac{f(\mathbf{x}^{(1)}) - f(\mathbf{x}^{(2)})}{f(\mathbf{x}^{(1)})} \right| = \left| \frac{8 - 0.888}{8} \right| \cong 0.888 \geq \epsilon_1 = 0.01$$

İstenen hassasiyette yakınsama söz konusu olmayıp, bir sonraki iterasyona geçilir. İkinci noktanın  $g(\mathbf{x}^{(2)}) = 0 \in \epsilon_2 = \pm 0.05$  olması nedeniyle kısıtın bu noktada etkin, diğer bir ifade ile aktif olduğu görülmektedir. Bu durumda doğrultu tayini için doğrusal programlama problemi teşkil edilir. Doğrusal programlama sorunu;

$$\begin{aligned} \frac{\partial f(\mathbf{x}^{(2)})}{\partial x_1} &= -\frac{4}{3} & \frac{\partial f(\mathbf{x}^{(2)})}{\partial x_2} &= -\frac{4}{3} \\ \frac{\partial g(\mathbf{x}^{(2)})}{\partial x_1} &= 1 & \frac{\partial g(\mathbf{x}^{(2)})}{\partial x_2} &= 2 \end{aligned}$$

değerlerinden hareketle;

$$\begin{aligned} & \min -\beta \\ \text{Kısıt} & \\ & -\frac{4}{3}t_1 - \frac{4}{3}t_2 + \beta \leq -\frac{8}{3} \\ & t_1 + 2t_2 + \beta \leq 3 \\ & 0 \leq t_{1,2} \leq 2 \\ & \beta > 0 \end{aligned}$$

şeklinde tanımlanır. Burada  $\theta_k = 1$  değeri dikkate alınmıştır. Oluşturulan doğrusal programlama sorunu “simplex” algoritması kullanılarak çözüldüğünde;

$$\begin{bmatrix} t_1 \\ t_2 \\ \beta \end{bmatrix} = \begin{bmatrix} 2 \\ 0.3 \\ 0.4 \end{bmatrix}$$

şeklindeki çözüm yöneyi elde edilir. Bu yöney kullanılarak adım atılacak doğrultu yöneyini;

$$\mathbf{d}^{(2)} = \mathbf{t}^* - \mathbf{1} = \begin{bmatrix} 2 \\ 0.3 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ -0.7 \end{bmatrix}$$

biçiminde elde edilebilir. Buna göre yeni nokta;



$$\mathbf{x}^{(3)} = \mathbf{x}^{(2)} + \alpha \mathbf{d}^{(2)} = \begin{bmatrix} 4/3 \\ 4/3 \end{bmatrix} + \alpha \begin{bmatrix} 1 \\ -0.7 \end{bmatrix} = \begin{bmatrix} \frac{4}{3} + \alpha \\ \frac{4}{3} - 0.7\alpha \end{bmatrix}$$

şeklindedir. Yön tayini sonrasında adım büyüklüğünün öngörülmesi adımına geçilir. Üzerinde bulunulan noktanın kısıt hattı üzerinde olması nedeniyle adım büyüklüğü tayini hassas bir işlem gerektirmektedir. Kısıt koşulu için belirlenen  $\epsilon_2 = \pm 0.05$  aralığı dikkate alınarak adım büyüklüğü değeri hesaplanabilir;

$$g(\mathbf{x}^{(3)}) = \frac{12}{3} + \frac{4}{10}\alpha - 4 \leq -0.05$$

$$\bar{\alpha} = 0.125$$

Öte yandan amaç fonksiyonu baz alındığında adım büyüklüğü;

$$\frac{\partial f(\alpha)}{\partial \alpha} = 1.49\alpha - 0.4 = 0, \quad \alpha^* = 0.268$$

olarak bulunur. Görüldüğü üzere bu değer, kısıt temelli adım büyüklüğü değerinden büyük olup atılacak adımda dikkate alınacak adım büyüklüğü  $\alpha = 0.125$  değeridir. Buna göre yeni nokta;

$$\mathbf{x}^{(3)} = \begin{bmatrix} \frac{4}{3} + \alpha \\ \frac{4}{3} - 0.7\alpha \end{bmatrix} = \begin{bmatrix} 1.458 \\ 1.246 \end{bmatrix}$$

olarak hesaplanır. Varılan noktanın en uygun nokta olup olmadığı yakınsama kontrolü ile denetlenebilir;

$$\left| \frac{f(\mathbf{x}^{(2)}) - f(\mathbf{x}^{(3)})}{f(\mathbf{x}^{(2)})} \right| = \left| \frac{0.888 - 0.862}{0.888} \right| \cong 0.029 \geq \epsilon_1 = 0.01$$

İstenen hassasiyette yakınsama söz konusu olmayıp, bir sonraki iterasyona geçilir. Üçüncü noktada kısıtın aktif olması nedeniyle ( $\epsilon_2 = \pm 0.05$ ) doğrusal programlama sorunu;

$$\frac{\partial f(\mathbf{x}^{(3)})}{\partial x_1} = -1.084 \quad \frac{\partial f(\mathbf{x}^{(3)})}{\partial x_2} = -1.508$$



$$\frac{\partial g(\mathbf{x}^{(2)})}{\partial x_1} = 1 \quad \frac{\partial g(\mathbf{x}^{(2)})}{\partial x_2} = 2$$

değerlerinden hareketle;

$$\begin{aligned} & \text{Kısıt} \quad \min -\beta \\ & -1.084t_1 - 1.508t_2 + \beta \leq -2.592 \\ & t_1 + 2t_2 + \beta \leq 3 \\ & 0 \leq t_{1,2} \leq 2 \\ & \beta > 0 \end{aligned}$$

şeklinde teşkil edilir. Oluşturulan doğrusal programlama sorunu “simplex” algoritması kullanılarak çözüldüğünde;

$$\begin{bmatrix} t_1 \\ t_2 \\ \beta \end{bmatrix} = \begin{bmatrix} 2 \\ 0.406 \\ 0.108 \end{bmatrix}$$

şeklindeki çözüm yöneyi elde edilir. Bu yöney vasıtasıyla adım atılacak doğrultu yöneyini;

$$\mathbf{d}^{(3)} = \mathbf{t}^* - \mathbf{1} = \begin{bmatrix} 2 \\ 0.406 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ -0.594 \end{bmatrix}$$

olarak elde edebiliriz. Buna göre yeni nokta;

$$\mathbf{x}^{(4)} = \mathbf{x}^{(3)} + \alpha \mathbf{d}^{(3)} = \begin{bmatrix} 1.458 \\ 1.246 \end{bmatrix} + \alpha \begin{bmatrix} 1 \\ -0.594 \end{bmatrix} = \begin{bmatrix} 1.458 + \alpha \\ 1.246 - 0.594\alpha \end{bmatrix}$$

biçimindedir. Yön tayini sonrasında yine adım büyüklüğünün öngörülmesi adımına geçilir. Üzerinde bulunulan noktanın hâla kısıt aralığı üzerinde olması nedeniyle daha önce dikkate alınan adım büyüklüğü üst değeri burada da geçerlidir;

$$\bar{\alpha} = 0.125$$

Öte yandan amaç fonksiyonu baz alındığında adım büyüklüğü;

$$\frac{\partial f(\alpha)}{\partial \alpha} = 2.705\alpha - 0.188 = 0$$





$$\alpha^* = 0.069$$

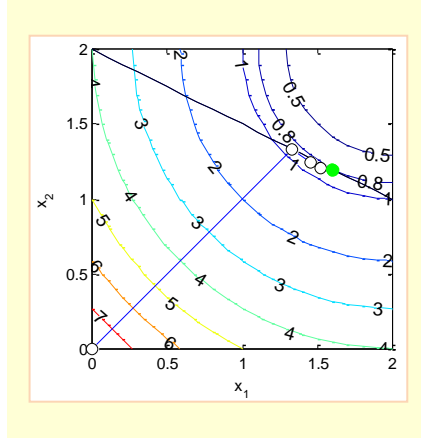
olarak bulunur. Elde edilen bu değerin, üst sınırdan daha düşük olması nedeniyle yeni adım büyüklüğü  $\alpha^* = 0.069$  değeridir. Buna göre yeni nokta;

$$\mathbf{x}^{(4)} = \begin{bmatrix} 1.458 + \alpha \\ 1.246 - 0.594\alpha \end{bmatrix} = \begin{bmatrix} 1.5275 \\ 1.2046 \end{bmatrix}$$

olarak hesaplanır. Varılan noktanın en uygun nokta olup olmadığı yakınsama kontrolü ile denetlenebilir;

$$\left| \frac{f(\mathbf{x}^{(3)}) - f(\mathbf{x}^{(4)})}{f(\mathbf{x}^{(3)})} \right| = \left| \frac{0.862 - 0.855}{0.862} \right| \cong 0.007 < \epsilon_1 = 0.01$$

Görüldüğü üzere istenen hassasiyette yakınsama yakalandığı için tarama süreci sonlandırılabilir. Şekil 2.84 dâhilinde amaç fonksiyonu yüzeyi ile taranan ve ulaşılan noktalar görülmektedir.



Şekil 2.84 Amaç fonksiyonu yüzeyi ve taranan noktalar, (yeşil nokta optimal noktadır.)

Zoutendijk'in uygun doğrultu yöntemi, nispeten basit bir yöntemdir. Bununla beraber hassas çözüm arayışlarında, fazla sayıda tekrarlı işleme ihtiyaç duyduğu söylenebilir. Zira kısıt koşullarının etkin olduğu bölgelerde adım büyüklüğünün fazla olması, kısıt koşulunun ihlal





edilmesine neden olabilir. Dolayısıyla küçük adımlarla ilerleme sağlanabilir. Bu ise tekrarlı işlem sayısını doğal olarak artıracaktır.

#### (4) Genelleştirilmiş-azaltılmış gradyan yöntemi

Genelleştirilmiş-azaltılmış gradyan yöntemi, azaltılmış gradyan yönteminin genişletilmiş bir seçeneğidir. Azaltılmış-gradyan yöntemi, Wolfe'nin 1962 tarihli yayımlanmamış raporuna dayanır<sup>(64)</sup>. Yöntemin genelleştirilmesi hâli, “Wolfe'nin azaltılmış gradyan yönteminin doğrusal olmayan kısıtlar için genelleştirilmesi” başlıklı makalede geçmektedir<sup>(65)</sup>. Hatırlanacağı üzere daha önce analitik çözüm yöntemleri bahsinde yerine koyma yöntemine değinilmiş ve söz konusu yöntemin temelinde, kısıt denklemlerinden faydalanılarak bir değişkenin diğer bir değişken cinsinden ifade edilmesi ve amaç fonksiyonunda yerlerine konularak çözüme gidilmesi fikrinin yattığı ifade edilmişti. Böylelikle kısıtlar, amaç fonksiyonuna dâhil edilerek kısıtsız sorun çözümü ile aynı çözüm zeminine dönüşüm sağlanmış olmaktadır. Benzer yaklaşım, genelleştirilmiş-azaltılmış gradyan yönteminin çıkış kaynağıdır. Buna göre kısıt denklemlerinden hareketle bazı değişkenler diğer değişkenler cinsinden ifade edilerek amaç fonksiyonunda yerine konulmakta ve bu sayede sorun boyutsal olarak daha az değişken içeren, diğer bir ifade ile değişken sayısı açısından azaltılmış bir sorun haline getirilmektedir. İlave olarak, yeniden tanımlanan sorunun, yerel çözüm bulmada oldukça etkin olan gradyan esaslı yöntemlerle çözümü mümkün hale gelmektedir. Bununla beraber, tıpkı yerine koyma yönteminde olduğu gibi burada da doğrusal olmayan kısıt denklemlerinden değişken azaltmak zor ve hatta mümkün olmayabilir. Yöntemin bu durumda başvurduğu işlem, kısıt fonksiyonlarının birinci mertebeden “Taylor” serisi açılımlarını kullanarak doğrusallaştırma sürecine gitmek ve bu sayede değişkenleri, amaç fonksiyonunda yerine koymaktır. Tabiatıyla bulunulan konumda doğrusallaştırma, gerçek sorundan belli oranda uzaklaşmaya neden olmaktadır. Ancak her seferinde elde edilen yeni nokta dikkate alınarak sorun tekrar tanımlanmakta ve istenen hassasiyette çözüme ulaşıncaya kadar tarama sürecine devam edilmektedir.

Optimizasyon sorunundaki kısıtların hepsinin eşitlik kısıtı olması durumunda çözüme gitmek daha kolaydır. Zira eşitlik ifadelerinde belirli değişkenler diğer değişkenler cinsinden ifade edilebilir. Kısıt denklemlerinde, yalnız bırakılacak değişkenlere bağımlı değişken, bu değişkenlerin ifade edilmesi için kullanılacak ve dolayısıyla amaç fonksiyonunda dikkate alınacak değişkenlere bağımsız değişken adı verilir. Tahmin edileceği üzere her bir kısıt denklemi ile sadece bir bağımlı değişken tanımlanabilir. Diğer taraftan yöntemin başarılı olabilmesi için gereken diğer bir husus da başlangıç noktasının olası çözüm uzayında olması gereğidir.



Kısıtların içerisinde eşitsizlik kısıtı olması durumunda, tıpkı “simplex” algoritmasında olduğu gibi, arttırma veya eksiltme amaçlı ilave yapay değişkenlerin kısıt denklemlerine ilave edilmesi ve eşitlik kısıtı denklemlerine dönüştürme işlemleri gerçekleştirilir. Dolayısıyla orijinal haliyle doğrusal olmayan kısıtlı optimizasyon sorunu;

$$\begin{array}{l} \text{Kısıt} \\ \min f(\mathbf{x}) \\ h_k(\mathbf{x}) = 0, \quad k = 1, 2, \dots, l \\ g_j(\mathbf{x}) \leq 0, \quad j = 1, 2, \dots, m \\ x_i^a \leq x_i \leq x_i^u, \quad i = 1, 2, \dots, n \end{array} \quad (2.189)$$

eklenen ilave değişkenlerle;

$$\begin{array}{l} \text{Kısıt} \\ \min f(\mathbf{x}) \\ h_k(\mathbf{x}) = 0, \quad k = 1, 2, \dots, l \\ g_j(\mathbf{x}) + x_{n+j} = 0, \quad j = 1, 2, \dots, m \\ x_i^a \leq x_i \leq x_i^u, \quad i = 1, 2, \dots, n \\ x_{n+j} \geq 0, \end{array} \quad (2.190)$$

haline dönüşür. Burada  $x_{n+j} \geq 0$  ifadesi, ilave değişkeni temsil etmektedir. Eşitsizlik kısıt sayısına göre eklenen değişkenlerin dikkate alınmasıyla tasarım yöneyi;

$$\mathbf{x} = [x_1 \ x_2 \ \dots \ x_n \ x_{n+1} \ x_{n+2} \ \dots \ x_{n+m}]^T \quad (2.191)$$

şeklinde ifade edilebilir. Bu yöneyde bazı değişkenler bağımsız ve geri kalan da bağımlı değişken olarak seçilip iki gruba ayırma işlemi uygulanırsa aynı yöney;

$$\mathbf{x} = \begin{bmatrix} \mathbf{y} \\ \mathbf{z} \end{bmatrix} = [y_1 \ y_2 \ \dots \ y_{n-l} \ z_1 \ z_2 \ \dots \ z_{l+m}]^T \quad (2.192)$$

şeklinde yazılabilir. Burada  $\mathbf{y}$  bağımlı,  $\mathbf{z}$  bağımsız değişken grubunu göstermektedir. Benzer yaklaşım kısıtlar için de yapılabilir;

$$\mathbf{H} = \begin{bmatrix} \mathbf{h} \\ \mathbf{g} \end{bmatrix} \quad (2.193)$$

Burada  $\mathbf{h}$  mevcut eşitlik kısıtlarını,  $\mathbf{g}$  eşitsizlik halinden dönüşüme uğramış eşitlik kısıtlarını göstermektedir. Amaç fonksiyonu ile eşitlik kısıtı mahiyetindeki kısıt denklemleri kısmi gradyan değerleri dikkate alınarak 1inci mertebeden “Taylor” serisine açılırsa;



$$f(\mathbf{z}, \mathbf{y}) = f(\mathbf{z}^{(p)}, \mathbf{y}^{(p)}) + \nabla_y f(\mathbf{z}^{(p)}, \mathbf{y}^{(p)})^T d\mathbf{y} + \nabla_z f(\mathbf{z}^{(p)}, \mathbf{y}^{(p)})^T d\mathbf{z} \quad (2.194)$$

$$\mathbf{H}(\mathbf{z}, \mathbf{y}) = \mathbf{H}(\mathbf{z}^{(p)}, \mathbf{y}^{(p)}) + \nabla_y \mathbf{H}(\mathbf{z}^{(p)}, \mathbf{y}^{(p)})^T d\mathbf{y} + \nabla_z \mathbf{H}(\mathbf{z}^{(p)}, \mathbf{y}^{(p)})^T d\mathbf{z} \quad (2.195)$$

denklemleri elde edilir. Burada  $\mathbf{z}^{(p)}, \mathbf{y}^{(p)}$  noktası, o adımda bulunulan konumu göstermektedir. Diğer taraftan gerek başlangıç ve gerekse de hâlihazırda bulunulan konumun olası çözüm uzayı içerisinde olması koşulu dikkate alındığında eşitlik kısıtlarının değerinin sıfır olması gerektiği söylenebilir. Bu durumda;

$$\mathbf{H}(\mathbf{z}, \mathbf{y}) = \mathbf{H}(\mathbf{z}^{(p)}, \mathbf{y}^{(p)}) = 0 \quad (2.196)$$

yazılabilir. Dolayısıyla eşitlik kısıt denklemlerinden hareketle;

$$\nabla_y \mathbf{H}(\mathbf{z}^{(p)}, \mathbf{y}^{(p)})^T d\mathbf{y} + \nabla_z \mathbf{H}(\mathbf{z}^{(p)}, \mathbf{y}^{(p)})^T d\mathbf{z} = 0 \quad (2.197)$$

$$d\mathbf{y} = -[\nabla_y \mathbf{H}(\mathbf{z}^{(p)}, \mathbf{y}^{(p)})]^{-1} \nabla_z \mathbf{H}(\mathbf{z}^{(p)}, \mathbf{y}^{(p)}) d\mathbf{z} \quad (2.198)$$

denklemleri elde edilebilir. Bağımsız değişkenleri içeren bu ifade amaç fonksiyonunda yerine konulursa;

$$f(\mathbf{z}, \mathbf{y}) = f(\mathbf{z}^{(p)}, \mathbf{y}^{(p)}) + \nabla_y f(\mathbf{z}^{(p)}, \mathbf{y}^{(p)})^T \left[ -\nabla_y \mathbf{H}(\mathbf{z}^{(p)}, \mathbf{y}^{(p)}) \right]^{-1} \nabla_z \mathbf{H}(\mathbf{z}^{(p)}, \mathbf{y}^{(p)}) d\mathbf{z} + \nabla_z f(\mathbf{z}^{(p)}, \mathbf{y}^{(p)})^T d\mathbf{z} \quad (2.199)$$

ve elde edilen bu denklemde aynı terimler aynı çatı altında düzenlenirse;

$$f(\mathbf{z}, \mathbf{y}) = f(\mathbf{z}^{(p)}, \mathbf{y}^{(p)}) + \left[ \nabla_z f(\mathbf{z}^{(p)}, \mathbf{y}^{(p)}) - [\nabla_y \mathbf{H}(\mathbf{z}^{(p)}, \mathbf{y}^{(p)})]^{-1} \nabla_z \mathbf{H}(\mathbf{z}^{(p)}, \mathbf{y}^{(p)}) \nabla_y f(\mathbf{z}^{(p)}, \mathbf{y}^{(p)}) \right]^T d\mathbf{z} \quad (2.200)$$

ifadesi elde edilir. Bu ifadenin, bağımsız değişkenleri içeren  $\mathbf{z}$  yöneyine göre değişimi;

$$\frac{df(\mathbf{z}, \mathbf{y})}{d\mathbf{z}} = \nabla_z f(\mathbf{z}^{(p)}, \mathbf{y}^{(p)}) - \left[ \nabla_y \mathbf{H}(\mathbf{z}^{(p)}, \mathbf{y}^{(p)}) \right]^{-1} \nabla_z \mathbf{H}(\mathbf{z}^{(p)}, \mathbf{y}^{(p)}) \nabla_y f(\mathbf{z}^{(p)}, \mathbf{y}^{(p)}) \quad (2.201)$$

denklemleri ile temsil edilebilir.



Bu ifadenin eldesi sonucunda azaltılmış gradyan yöneyi ( $\mathbf{G}_R$ ) tespit edilmiş olur. Hatırlanacağı üzere amaç fonksiyonunda azalma elde etmek için gradyanın tersi yönünde hareket edilmeliydi. Dolayısıyla gidilecek yön olarak;

$$\mathbf{G}_R = \nabla_z f(\mathbf{z}^{(p)}, \mathbf{y}^{(p)}) - \left[ [\nabla_y \mathbf{H}(\mathbf{z}^{(p)}, \mathbf{y}^{(p)})]^{-1} \nabla_z \mathbf{H}(\mathbf{z}^{(p)}, \mathbf{y}^{(p)}) \right]^T \nabla_y f(\mathbf{z}^{(p)}, \mathbf{y}^{(p)}) \quad (2.202)$$

$$\mathbf{d}^{(p)} = -\mathbf{G}_R \quad (2.203)$$

İfadeleri dikkate alınabilir. Gradyan yönünün tespit sonrasında yakınsama kontrolüne gitmek mümkündür. Zira gradyanın sıfır veya hassasiyete bağlı olarak  $\epsilon$  değerine göre daha küçük bir değer alması durumunda istenilen çözüm noktasına ulaşılmış olur;

$$\|\mathbf{G}_R\| \leq \epsilon \quad (2.204)$$

Aksi halde yapılacak işlem, belirlenen yöndeki adım büyüklüğünün tayinidir. Bu amaçla, adım büyüklüğüne bağlı olarak yeni nokta tanımlanır;

$$\mathbf{x}^{(p+1)} = \mathbf{x}^{(p)} + \alpha \mathbf{d}^{(p)} \quad (2.205)$$

Daha sonra bu nokta, amaç fonksiyonunda yerine konularak fonksiyondaki azalmayı doruklaştıracak adım büyüklüğü tespit edilir;

$$\frac{df(\mathbf{x}^{(p+1)})}{d\alpha} = 0 \quad (2.206)$$

Adım büyüklüğünün tespiti sonrasında yeni nokta kolayca bulunmuş olur. Ancak burada dikkat edilmesi gereken husus, adım büyüklüğünün kısıtları ihlal etmemesi gereğidir. Dolayısıyla yeni bulunan  $\mathbf{x}^{(p+1)}$  noktasının, kısıt koşullarını sağlayıp sağlamadığı kontrol edilir. Eğer bir ihlal söz konusu ise  $\mathbf{d}^{(p)}$  yönünde atılan adımın uygun olmadığı anlaşılır ve dolayısıyla adım büyüklüğünün ihlal edilen kısıt denklemine göre tespit edilmesi gerekir;

$$\mathbf{H}_q(\mathbf{x}^{(p+1)}) = 0 \quad (2.207)$$

Burada  $\mathbf{H}_q$  fonksiyonu, ihlal edilen kısıt fonksiyonunu göstermektedir.



Genelleştirilmiş-azaltılmış gradyan yönteminin daha iyi anlaşılabilmesi amacıyla örnek bir sorunu, belirtilen işlem adımlarına göre çözmek mümkündür. Buna göre;

$$\begin{aligned} \text{Kısıt} \quad \min f(\mathbf{x}) &= -2x_1 - 4x_2 + x_1^2 + x_2^2 + 5 && (2.208) \\ & -x_1 + 2x_2 - 2 \leq 0 \\ & x_1 + x_2 - 4 \leq 0 \end{aligned}$$

sorununun  $\mathbf{x}^{(1)} = [0 \ 0]^T$  başlangıç noktası ve  $\epsilon = 0.01$  çözüm hassasiyeti değerine göre en uygun tasarım yöneyini elde etmek için öncelikle eşitsizlik kısıtlarını eşitlik kısıtı haline dönüştürülmesi gerekir;

$$\begin{aligned} \text{Kısıt} \quad \min f(\mathbf{x}) &= -2x_1 - 4x_2 + x_1^2 + x_2^2 + 5 \\ g_1(\mathbf{x}) &= -x_1 + 2x_2 - 2 + x_3 = 0 \\ g_2(\mathbf{x}) &= x_1 + x_2 - 4 + x_4 = 0 \\ x_3, x_4 &\geq 0 \end{aligned}$$

Burada  $x_3$  ve  $x_4$  değişkenleri, yapay değişken niteliğinde olup ayrıca mevcut kısıtlara ilave iki kısıt daha tanımlanmış olmaktadır. Yapay değişkenlerin başlangıç noktası itibari ile değerlerini, kısıt denklemlerinden faydalanarak elde etmek mümkündür;

$$\begin{aligned} x_3 &= x_1^{(1)} - 2x_2^{(1)} + 2 = 2 \\ x_4 &= -x_1^{(1)} - x_2^{(1)} + 4 = 4 \\ \mathbf{x}^{(1)} &= [0 \ 0 \ 2 \ 4]^T \end{aligned}$$

Oluşan yeni tasarım yöneyinin bağımlı ve bağımsız değişken ayrımı, ilave yapay değişkenlerin bağımsız değişken olarak tanımlanması dikkate alınarak yapılabilir;

$$\begin{aligned} \mathbf{y} &= \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\ \mathbf{z} &= \begin{bmatrix} x_3 \\ x_4 \end{bmatrix} \\ H &= \begin{bmatrix} g_1(\mathbf{x}) \\ g_2(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} -x_1 + 2x_2 - 2 + x_3 \\ x_1 + x_2 - 4 + x_4 \end{bmatrix} \end{aligned}$$

Diğer taraftan yöntemle göre gidilecek yön, azaltılmış gradyan yöneyine bağlı olarak hesaplanmaktadır;



$$\mathbf{G}_R = \nabla_z f(\mathbf{z}^{(1)}, \mathbf{y}^{(1)}) - \left[ \nabla_y \mathbf{H}(\mathbf{z}^{(1)}, \mathbf{y}^{(1)}) \right]^{-1} \nabla_z \mathbf{H}(\mathbf{z}^{(1)}, \mathbf{y}^{(1)}) \right]^T \nabla_y f(\mathbf{z}^{(1)}, \mathbf{y}^{(1)})$$

Dolayısıyla;

$$\nabla_z f(\mathbf{z}, \mathbf{y}) = \begin{bmatrix} \frac{\partial f(\mathbf{x})}{\partial x_3} \\ \frac{\partial f(\mathbf{x})}{\partial x_4} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \nabla_y f(\mathbf{z}, \mathbf{y}) = \begin{bmatrix} \frac{\partial f(\mathbf{x})}{\partial x_1} \\ \frac{\partial f(\mathbf{x})}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 2x_1 - 2 \\ 2x_2 - 4 \end{bmatrix} \quad \nabla_y f(\mathbf{x}^{(1)}) = \begin{bmatrix} -2 \\ -4 \end{bmatrix}$$

$$\nabla_z \mathbf{H}(\mathbf{x}) = \begin{bmatrix} \frac{\partial g_1(\mathbf{x})}{\partial x_3} & \frac{\partial g_1(\mathbf{x})}{\partial x_4} \\ \frac{\partial g_2(\mathbf{x})}{\partial x_3} & \frac{\partial g_2(\mathbf{x})}{\partial x_4} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \nabla_y \mathbf{H}(\mathbf{x}) = \begin{bmatrix} \frac{\partial g_1(\mathbf{x})}{\partial x_1} & \frac{\partial g_1(\mathbf{x})}{\partial x_2} \\ \frac{\partial g_2(\mathbf{x})}{\partial x_1} & \frac{\partial g_2(\mathbf{x})}{\partial x_2} \end{bmatrix} = \begin{bmatrix} -1 & 2 \\ 1 & 1 \end{bmatrix}$$

ifadelerinden hareketle azaltılmış gradyan yöneyi;

$$\mathbf{G}_R = \begin{bmatrix} 0 \\ 0 \end{bmatrix} - \left[ \begin{bmatrix} -1 & 2 \\ 1 & 1 \end{bmatrix} \right]^{-1} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} -2 \\ -4 \end{bmatrix} = \begin{bmatrix} 2/3 \\ 8/3 \end{bmatrix}$$

şeklinde elde edilir. Tarama sürecinin yakınsama kontrolü yapıldığında;

$$\|\mathbf{G}_R\| = 2.748 \geq \epsilon = 0.01$$

sonucu nedeniyle bir sonraki adıma geçilir.

Yeni noktanın, gradyan yönü dikkate alınarak belirlenmesi mümkündür. Ancak öncelikle bağımsız değişkenlerin teşkili ve akabinde de kısıt denklemleri kullanılarak bağımlı değişkenlerin eldesi yapılmalıdır. Belirlenen yön doğrultusunda bağımsız değişkenler;

$$\mathbf{d}^{(1)} = -\mathbf{G}_R = - \begin{bmatrix} 2/3 \\ 8/3 \end{bmatrix}$$

$$\mathbf{x}^{(2)} = \mathbf{x}^{(1)} + \alpha \mathbf{d}^{(1)} \Rightarrow \begin{bmatrix} x_3^{(2)} \\ x_4^{(2)} \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \end{bmatrix} - \alpha \begin{bmatrix} 2/3 \\ 8/3 \end{bmatrix} = \begin{bmatrix} 2 - 2/3\alpha \\ 4 - 8/3\alpha \end{bmatrix}$$

şeklinde elde edilip, bağımlı değişkenler;





$$x_1^{(2)} = \frac{1}{3}(x_3^{(2)} - 2x_4^{(2)}) + 2 = \frac{14}{9}\alpha$$

$$x_2^{(2)} = -\frac{1}{3}(x_3^{(2)} + x_4^{(2)}) + 2 = \frac{10}{9}\alpha$$

olarak hesaplanır. Adım büyüklüğüne göre ifade edilen tasarım yöneyinin, amaç fonksiyonunda yerine konulması ve fonksiyondaki azalmayı doruklaştıracak adım büyüklüğünün tespiti;

$$\frac{df(\mathbf{x}^{(2)})}{d\alpha} = 0$$

ifadesinden hareketle;

$$\frac{df(\mathbf{x}^{(2)})}{d\alpha} = \frac{592}{81}\alpha - \frac{68}{9} = 0$$

$$\alpha^* = 1.034$$

şeklinde yapılabilir. Buna göre yeni nokta;

$$\mathbf{x}^{(2)} = \begin{bmatrix} \frac{14}{9}\alpha \\ \frac{10}{9}\alpha \\ 2 - \frac{2}{3}\alpha \\ 4 - \frac{8}{3}\alpha \end{bmatrix} = \begin{bmatrix} 1.608 \\ 1.149 \\ 1.311 \\ 1.243 \end{bmatrix}$$

ifadesi ile elde edilmiş olur. Elde edilen noktanın, mevcut koşulları sağlayıp sağlamadığı adım büyüklüğünün doğruluğu açısından önemlidir. Bu nedenle varılan noktanın tüm kısıtları sağlayıp sağlamadığı kontrol edilir;

$$g_1(\mathbf{x}^{(2)}) = -x_1^{(2)} + 2x_2^{(2)} - 2 + x_3^{(2)} = -1.608 + 2 \times 1.149 - 2 + 1.311 = 0$$

$$g_2(\mathbf{x}^{(2)}) = x_1^{(2)} + x_2^{(2)} - 4 + x_4^{(2)} = 1.608 + 1.149 - 4 + 1.243 = 0$$

$$x_3^{(2)} \geq 0$$

$$x_4^{(2)} \geq 0$$

Görüldüğü üzere varılan nokta, tüm kısıtları sağlamaktadır. Yeni noktanın dikkate alınması ile yeni azaltılmış gradyan yöneyi;



$$\mathbf{G}_R = \nabla_z f(\mathbf{x}^{(2)}) - \left[ \left[ \nabla_y \mathbf{H}(\mathbf{x}^{(2)}) \right]^{-1} \nabla_z \mathbf{H}(\mathbf{x}^{(2)}) \right]^T \nabla_y f(\mathbf{x}^{(2)})$$

tanımı gereği;

$$\nabla_z f(\mathbf{x}) = \begin{bmatrix} \frac{\partial f(\mathbf{x})}{\partial x_3} \\ \frac{\partial f(\mathbf{x})}{\partial x_4} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \nabla_y f(\mathbf{x}) = \begin{bmatrix} \frac{\partial f(\mathbf{x})}{\partial x_1} \\ \frac{\partial f(\mathbf{x})}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 2x_1 - 2 \\ 2x_2 - 4 \end{bmatrix} \quad \nabla_y f(\mathbf{x}^{(2)}) = \begin{bmatrix} 1.216 \\ -1.714 \end{bmatrix}$$

$$\nabla_z \mathbf{H}(\mathbf{x}) = \begin{bmatrix} \frac{\partial g_1(\mathbf{x})}{\partial x_3} & \frac{\partial g_1(\mathbf{x})}{\partial x_4} \\ \frac{\partial g_2(\mathbf{x})}{\partial x_3} & \frac{\partial g_2(\mathbf{x})}{\partial x_4} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \nabla_y \mathbf{H}(\mathbf{x}) = \begin{bmatrix} \frac{\partial g_1(\mathbf{x})}{\partial x_1} & \frac{\partial g_1(\mathbf{x})}{\partial x_2} \\ \frac{\partial g_2(\mathbf{x})}{\partial x_1} & \frac{\partial g_2(\mathbf{x})}{\partial x_2} \end{bmatrix} = \begin{bmatrix} -1 & 2 \\ 1 & 1 \end{bmatrix}$$

ifadelerinden hareketle;

$$\mathbf{G}_R = \begin{bmatrix} 0 \\ 0 \end{bmatrix} - \left[ \begin{bmatrix} -1 & 2 \\ 1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right]^T \begin{bmatrix} 1.216 \\ -1.714 \end{bmatrix} = \begin{bmatrix} 0.9767 \\ -0.2393 \end{bmatrix}$$

olarak elde edilir. Tarama sürecinin yakınsama kontrolü yapıldığında;

$$\|\mathbf{G}_R\| = 1.0056 \geq \epsilon = 0.01$$

değeri nedeniyle bir sonraki adıma geçilir. Buna göre belirlenen yön doğrultusunda bağımsız değişkenler;

$$\mathbf{d}^{(2)} = -\mathbf{G}_R = - \begin{bmatrix} 0.9767 \\ -0.2393 \end{bmatrix}$$

$$\mathbf{x}^{(3)} = \mathbf{x}^{(2)} + \alpha \mathbf{d}^{(2)} \Rightarrow \begin{bmatrix} x_3^{(3)} \\ x_4^{(3)} \end{bmatrix} = \begin{bmatrix} 1.311 \\ 1.243 \end{bmatrix} - \alpha \begin{bmatrix} 0.9767 \\ -0.2393 \end{bmatrix} = \begin{bmatrix} 1.311 - 0.9767\alpha \\ 1.243 + 0.2393\alpha \end{bmatrix}$$

şeklinde elde edilip, bağımlı değişkenler;

$$x_1^{(3)} = \frac{1}{3}(x_3^{(3)} - 2x_4^{(3)}) + 2 = 1.608 - 0.485\alpha$$

$$x_2^{(3)} = -\frac{1}{3}(x_3^{(3)} + x_4^{(3)}) + 2 = 1.148 + 0.246\alpha$$





olarak hesaplanır. Adım büyüklüğüne göre ifade edilen tasarım yöneyinin, amaç fonksiyonunda yerine konulması ve fonksiyondaki azalmayı doruklaştıracak adım büyüklüğünün tespiti;

$$\frac{df(\mathbf{x}^{(3)})}{d\alpha} = 0$$

ifadesinden hareketle;

$$\frac{df(\mathbf{x}^{(3)})}{d\alpha} = 0.5915\alpha - 1.0089 = 0$$

$$\alpha^* = 1.7058$$

şeklinde yapılabilir. Yeni nokta;

$$\mathbf{x}^{(3)} = \begin{bmatrix} 1.608 - 0.485\alpha \\ 1.148 + 0.246\alpha \\ 1.311 - 0.9767\alpha \\ 1.243 + 0.2393\alpha \end{bmatrix} = \begin{bmatrix} 0.7806 \\ 1.5676 \\ -0.3550 \\ 1.6512 \end{bmatrix}$$

ifadesi ile elde edilmiş olur. Ancak daha önce de ifade edildiği üzere elde edilen noktanın mevcut koşulları sağlayıp sağlamadığı, adım büyüklüğünün doğruluğu açısından önemlidir. Bu nedenle varılan noktanın, tüm kısıtları sağlayıp sağlamadığı kontrol edilir;

$$g_1(\mathbf{x}^{(3)}) = -x_1^{(3)} + 2x_2^{(3)} - 2 + x_3^{(3)} = -0.7806 + 2 \times 1.5676 - 2 - 0.3550 = -0.0004 \cong 0$$

$$g_2(\mathbf{x}^{(3)}) = x_1^{(3)} + x_2^{(3)} - 4 + x_4^{(3)} = 0.7806 + 1.5676 - 4 + 1.6512 = -0.0006 \cong 0$$

$$x_3^{(3)} < 0$$

$$x_4^{(3)} \geq 0$$

Görüldüğü üzere varılan nokta tüm kısıtları sağlamamaktadır. Özellikle üçüncü kısıt ihlal edilmektedir. Dolayısıyla adım büyüklüğünün, bu kısıt dikkate alınarak tekrar hesap edilmesi gerekir. Bu meydana üçüncü kısıttan hareketle adım büyüklüğü tekrar hesaplanabilir;

$$x_3^{(3)} = 1.311 - 0.9767\alpha = 0$$

$$\alpha^* = 1.342$$

Bu adım büyüklüğüne göre yeni nokta;

$$\mathbf{x}^{(3)} = \begin{bmatrix} 1.608 - 0.485\alpha \\ 1.148 + 0.246\alpha \\ 1.311 - 0.9767\alpha \\ 1.243 + 0.2393\alpha \end{bmatrix} = \begin{bmatrix} 0.9571 \\ 1.4781 \\ 0 \\ 1.5641 \end{bmatrix}$$

olarak hesaplanmış olur.





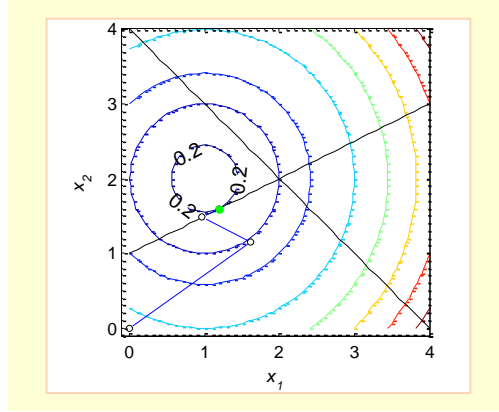
Varılan yeni noktanın, tüm kısıtları sağlayıp sağlamadığı kontrol edilir;

$$g_1(\mathbf{x}^{(3)}) = -x_1^{(3)} + 2x_2^{(3)} - 2 + x_3^{(3)} = -0.9571 + 2 \times 1.4781 - 2 - 0 = -0.0009 \cong 0$$

$$g_2(\mathbf{x}^{(3)}) = x_1^{(3)} + x_2^{(3)} - 4 + x_4^{(3)} = 0.9571 + 1.4781 - 4 + 1.5641 = -0.0007 \cong 0$$

$$x_3^{(3)}, x_4^{(3)} \geq 0$$

Yapılan adım büyüklüğü değişikliği sayesinde tüm koşulların sağlandığı gözlemlenmektedir. Benzer işlemler takip edilerek  $\mathbf{x}^{*,T} = [1.2 \ 1.6 \ 0 \ 1.2]$ ,  $f^* = 0.2$  şeklindeki optimal noktaya ve amaç fonksiyonu değerine ulaşılabilir<sup>(66)</sup>. Şekil 2.85 dahilinde, amaç fonksiyonu eşdeğer çizgileri üzerinde taranan noktalar ve ulaşılan optimal nokta gösterilmektedir.



Şekil 2.85 Amaç fonksiyonu yüzeyi ve taranan noktalar, (yeşil nokta optimal noktadır).

## (5) Ardışık ikilenik programlama yöntemi

Ardışık ikilenik programlama yönteminin kökeni 1960'lı yıllara gitmektedir. R. B. Wilson'ın "Yakınsak programlama için simpleksel yöntem"<sup>(67)</sup> adlı doktora çalışması<sup>(67)</sup> ile gündeme gelen söz konusu yöntem<sup>(68)</sup>, 1970'li yıllarda oldukça popüler ve yaygın olarak kullanılan bir metot haline gelmiştir. Yöntem, bazı literatürde izdüşümsel Lagrange yöntemi<sup>ii</sup> olarak da

<sup>i</sup> A simplicial method for convex programming

<sup>ii</sup> Projected Lagrangian method



adlandırılır<sup>(69)</sup>. Günümüzde ardışık ikilenik programlama yöntemi, doğrusal olmayan kısıtlı eniyileme sorunlarının çözümünde gradyan esaslı yöntemler arasında neredeyse en etkin yöntem olarak görülmekte olup yöntemin daha da geliştirilmesi adına aktif olarak bilimsel çalışmalara devam edilmektedir<sup>(70)</sup>. Bununla beraber yöntemin bazı çekimser kalınan yönleri de söz konusudur. Bu çekincelerden ilki, gerek amaç ve gerekse de kısıt fonksiyonlarının birinci ve ikinci mertebeden türevlenebilir olmaları gereği, diğeri ise hesaplamalarda ihtiyaç duyulan her iki mertebeden türevlerin eldesinin zaman zaman zor ve değişken sayısının fazla olması durumunda da hesaplama yükü açısından ağır olmasıdır<sup>(71)</sup>. Ayrıca hesaplamalarda ihtiyaç duyulan ikincil mertebeden türev verisinin pozitif tanımlı olması da diğer önem arz eden bir husustur.

Ardışık ikilenik programlama yönteminde temel husus, soruna ikinci dereceden bir polinomial yaklaşımda bulunarak sorunun ikilenik programlama sorununa indirgenmesi ve kurgulanan bu alt sorunun çözümünü kullanarak ana problemin çözümü adına bir öngöründe bulunulmasıdır. Varılan her yeni nokta, alt sorun niteliğindeki ikilenik programlama sorununun yeniden kurgulanması için tekrar kullanılır ve ardışık işlemlere devam edilerek istenen hassasiyetteki çözüme ulaşmak hedeflenir. Yöntemin etkinliğini sağlayan ana unsur, alt problem niteliğindeki ikilenik programlama sorununun çözümünün etkin ve nispeten kolay bir şekilde elde edilebilmesidir. Hatırlanacağı üzere daha önce ikilenik programlama sorunu ele alınmış ve Lagrange çarpanları yöntemi ile çözüme nasıl gidileceği ifade edilmişti.

Ardışık ikilenik programlama yönteminin temel işlem adımları aşağıdaki gibi sıralanabilir;

- ▶ 1. Başlangıç noktası temel alınarak amaç fonksiyonu ile kısıtların dikkate alındığı Lagrange fonksiyonunun tanımlanması,
- ▶ 2. Tanımlanan Lagrange fonksiyonunun Taylor serisi açılımını (ikinci mertebeden) temel alarak ikilenik yaklaşımın yapılması,
- ▶ 3. Kısıt denklemlerine yine Taylor serisi (birinci mertebeden) açılımını kullanarak doğrusal yaklaşımın yapılması,
- ▶ 4. Tanımlanan yaklaşımların dikkate alınması ile ikilenik programlama sorununun kurgulanması ve çözümünün elde edilerek ana soruna ait gidilecek doğrultu yönünün elde edilmesi,
- ▶ 5. Belirlenen doğrultuda atılacak adım büyüklüğünün tespit edilmesi,



- 6. Yeni çözüm noktasının eldesi ve yakınsama kontrolünün yapılması, yakınsamanın sağlanamaması durumunda 1inci işlem adımına gidilmesi ve ardışık işlemlere devam edilmesi.

İfade edilen ilk dört maddenin dayandığı temel ilkelerin anlaşılması, yöntemin izahı açısından önem arz etmektedir. Hatırlanacağı üzere en genel haliyle doğrusal olmayan kısıtlı eniyileme sorunu;

$$\begin{aligned} & \min f(\mathbf{x}) \\ \text{Kısıt} \quad & h_i(\mathbf{x}) = 0, \\ & g_j(\mathbf{x}) \leq 0 \end{aligned}$$

şeklinde dikkate alınırsa Lagrange fonksiyonu;

$$L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f(\mathbf{x}) + \sum \lambda_i h_i(\mathbf{x}) + \sum \mu_j g_j(\mathbf{x})$$

biçiminde ifade edilmektedir. Ayrıca Lagrange fonksiyonunun minimumu, öngörülen koşullar altında amaç fonksiyonunun azami değeri anlamına gelmektedir. Öte yandan Newton yöntemine göre tarama sürecinde, bir sonraki varılacak nokta;

$$\mathbf{X} = \mathbf{X}_k - \frac{\nabla L_k}{\nabla^2 L_k}$$

ifadesi baz alınarak elde edilebilmektedir. Bu ifadenin çıkış noktası, fonksiyonun ikinci mertebeden Taylor serisine açılmasıdır. Burada  $\mathbf{X}_k = [\mathbf{x}_k \quad \boldsymbol{\lambda}_k \quad \boldsymbol{\mu}_k]^T$  şeklindeki Lagrange tasarım yöneyidir. Eldeki bu ifade yeniden düzenlenirse;

$$\begin{bmatrix} \mathbf{x} \\ \boldsymbol{\lambda} \\ \boldsymbol{\mu} \end{bmatrix} - \begin{bmatrix} \mathbf{x}_k \\ \boldsymbol{\lambda}_k \\ \boldsymbol{\mu}_k \end{bmatrix} = -\frac{\nabla L_k}{\nabla^2 L_k} \rightarrow \nabla^2 L_k \begin{bmatrix} \mathbf{x} - \mathbf{x}_k \\ \boldsymbol{\lambda} - \boldsymbol{\lambda}_k \\ \boldsymbol{\mu} - \boldsymbol{\mu}_k \end{bmatrix} = -\nabla L_k \quad (2.209)$$

eşitliği yazılabilir. Öte yandan birincil ve ikincil türev değerleri daha açık bir halde yazılırsa;

$$\nabla L_k = \begin{bmatrix} (\partial L / \partial \mathbf{x})_k \\ (\partial L / \partial \boldsymbol{\lambda})_k \\ (\partial L / \partial \boldsymbol{\mu})_k \end{bmatrix} \quad (2.210)$$



$$\nabla^2 L_k = \begin{bmatrix} (\partial^2 L / \partial \mathbf{x}^2)_k & (\partial^2 L / \partial \mathbf{x} \partial \boldsymbol{\lambda})_k & (\partial^2 L / \partial \mathbf{x} \partial \boldsymbol{\mu})_k \\ (\partial^2 L / \partial \boldsymbol{\lambda} \partial \mathbf{x})_k & (\partial^2 L / \partial \boldsymbol{\lambda}^2)_k & (\partial^2 L / \partial \boldsymbol{\lambda} \partial \boldsymbol{\mu})_k \\ (\partial^2 L / \partial \boldsymbol{\mu} \partial \mathbf{x})_k & (\partial^2 L / \partial \boldsymbol{\mu} \partial \boldsymbol{\lambda})_k & (\partial^2 L / \partial \boldsymbol{\mu}^2)_k \end{bmatrix} \quad (2.211)$$

ve Lagrange fonksiyonu dikkate alınarak elde edilen değerler yerine koyulursa;

$$\nabla L_k = \begin{bmatrix} (\nabla_x f + \lambda \nabla_x \mathbf{h} + \mu \nabla_x \mathbf{g})_k \\ (\mathbf{h})_k \\ (\mathbf{g})_k \end{bmatrix} \quad (2.212)$$

$$\nabla^2 L_k = \begin{bmatrix} (\nabla_{xx}^2 f + \lambda \nabla_{xx}^2 \mathbf{h} + \mu \nabla_{xx}^2 \mathbf{g})_k & (\nabla_x \mathbf{h})_k & (\nabla_x \mathbf{g})_k \\ (\nabla_x \mathbf{h})_k & 0 & 0 \\ (\nabla_x \mathbf{g})_k & 0 & 0 \end{bmatrix} \quad (2.213)$$

eşitliklerine ulaşılır. Bu ifadeler Newton yönteminde yerine koyulursa;

$$\begin{bmatrix} (\nabla_{xx}^2 f + \lambda \nabla_{xx}^2 \mathbf{h} + \mu \nabla_{xx}^2 \mathbf{g})_k & (\nabla_x \mathbf{h})_k & (\nabla_x \mathbf{g})_k \\ (\nabla_x \mathbf{h})_k & 0 & 0 \\ (\nabla_x \mathbf{g})_k & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} - \mathbf{x}_k \\ \boldsymbol{\lambda} - \boldsymbol{\lambda}_k \\ \boldsymbol{\mu} - \boldsymbol{\mu}_k \end{bmatrix} = - \begin{bmatrix} (\nabla_x f + \lambda \nabla_x \mathbf{h} + \mu \nabla_x \mathbf{g})_k \\ (\mathbf{h})_k \\ (\mathbf{g})_k \end{bmatrix} \quad (2.214)$$

denklem grubu elde edilmiş olur. Matris çarpımları sonrasında elde edilen üç denklem, her iki taraftaki aynı terimlerin sadeleştirilmesi sonrasında;

$$\begin{bmatrix} (\nabla_{xx}^2 f + \lambda \nabla_{xx}^2 \mathbf{h} + \mu \nabla_{xx}^2 \mathbf{g})_k & (\nabla_x \mathbf{h})_k & (\nabla_x \mathbf{g})_k \\ (\nabla_x \mathbf{h})_k & 0 & 0 \\ (\nabla_x \mathbf{g})_k & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} - \mathbf{x}_k \\ \boldsymbol{\lambda}_k \\ \boldsymbol{\mu}_k \end{bmatrix} = - \begin{bmatrix} (\nabla_x f)_k \\ (\mathbf{h})_k \\ (\mathbf{g})_k \end{bmatrix} \quad (2.215)$$

veya açık hali ile,

$$(\nabla_{xx}^2 f + \lambda \nabla_{xx}^2 \mathbf{h} + \mu \nabla_{xx}^2 \mathbf{g})_k (\mathbf{x} - \mathbf{x}_k) + \boldsymbol{\lambda}_k (\nabla_x \mathbf{h})_k + \boldsymbol{\mu}_k (\nabla_x \mathbf{g})_k + (\nabla_x f)_k = 0 \quad (2.216)$$

$$(\nabla_x \mathbf{h})_k (\mathbf{x} - \mathbf{x}_k) + (\mathbf{h})_k = 0 \quad (2.217)$$

$$(\nabla_x \mathbf{g})_k (\mathbf{x} - \mathbf{x}_k) + (\mathbf{g})_k = 0 \quad (2.218)$$



şeklinde belirlenmiş olur. Diğer taraftan bu denklemler bir ikilenik programlama sorununun çözümü ile aynı karakterdedir. Şöyle ki; amaç fonksiyonu ikinci dereceden bir polinomiyal, kısıt denklemleri ise birinci dereceden doğrusal denklemlere sahip aşağıdaki gibi bir ikilenik eniyileme sorununu ele alalım;

$$\begin{aligned} \min Q(x) &= (\nabla_x f)_k^T \Delta x + \frac{1}{2} \Delta x^T (\nabla_{xx}^2 L)_k \Delta x && (2.219) \\ \text{Kısıt} & && \\ &(\mathbf{h})_k + (\nabla_x \mathbf{h})_k \Delta x = 0 && \\ &(\mathbf{g})_k + (\nabla_x \mathbf{g})_k \Delta x = 0 && \end{aligned}$$

Burada  $\Delta \mathbf{x} = \mathbf{x} - \mathbf{x}_k$  şeklinde tanımlanmaktadır. Görüldüğü üzere bahse konu sorunda  $k$  alt indisli ifadeler belli bir noktadaki sayısal değerleri göstermekte, dolayısıyla amaç fonksiyonunu ikinci dereceden, kısıtların ise birinci dereceden doğrusal birer polinom olmalarını sağlamaktadır. İkilenik programlama niteliğindeki bu sorunun çözümü sadedinde Lagrange çarpanları yöntemi dikkate alınır;

$$L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = Q(\mathbf{x}) + \boldsymbol{\lambda}^T \mathbf{h}(\mathbf{x}) + \boldsymbol{\mu}^T \mathbf{g}(\mathbf{x}) \quad (2.220)$$

veya açık hali ile;

$$\begin{aligned} L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) &= (\nabla_x f)_k^T \Delta x + \frac{1}{2} \Delta x^T (\nabla_{xx}^2 L)_k \Delta x + \boldsymbol{\lambda}((\mathbf{h})_k + (\nabla_x \mathbf{h})_k \Delta x) + \boldsymbol{\mu}((\mathbf{g})_k \\ &+ (\nabla_x \mathbf{g})_k \Delta x) \end{aligned} \quad (2.221)$$

şeklindeki Lagrange fonksiyonu yazılabilir. Optimal noktada KKT şartlarının  $(\nabla L = 0)$  dikkate alınması halinde sorunun çözümü;

$$(\nabla_x f)_k + (\nabla_{xx}^2 L)_k \Delta x + \boldsymbol{\lambda}(\nabla_x \mathbf{h})_k + \boldsymbol{\mu}(\nabla_x \mathbf{g})_k = 0 \quad (2.222)$$

$$(\mathbf{h})_k + (\nabla_x \mathbf{h})_k \Delta x = 0 \quad (2.223)$$

$$(\mathbf{g})_k + (\nabla_x \mathbf{g})_k \Delta x = 0 \quad (2.224)$$

problemine indirgenmiş olur. Görüldüğü üzere çözüm bekleyen bu üç denklem, Newton yöntemi ve Lagrange çarpanları yaklaşımı ile elde edilen ana sorunun çözümü ile aynı özelliğe sahiptir. Dolayısıyla her seferinde alt problem niteliğindeki ikilenik programlama sorunu çözülebilir ve ardışık tekrarlı işlemlerle optimal çözüme ulaşılabilir. Alt problemin çözümü



sonucu elde edilen yöney, ana problem için gidilecek yön hükmündedir. Ana sorunda bir sonraki adım, basit haliyle;

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \Delta \mathbf{x} \quad (2.225)$$

ifadesi kullanılarak elde edilebilir. İkilenik programlama probleminin kurgulanmasında ihtiyaç duyulan ikinci mertebeden türev ifadesi  $((\nabla_{\mathbf{xx}}^2 L)_k$  veya diğer bir ifade ile  $H$ ), genellikle bir tahminle başlatılır ve BFGS yöntemi ile güncellenir. Güncellemede;

$$H_{k+1} = H_k + \frac{q_k q_k^T}{q_k (\mathbf{x}_{k+1} - \mathbf{x}_k)} - \frac{H_k (\mathbf{x}_{k+1} - \mathbf{x}_k) (\mathbf{x}_{k+1} - \mathbf{x}_k)^T H_k^T}{(\mathbf{x}_{k+1} - \mathbf{x}_k)^T H_k (\mathbf{x}_{k+1} - \mathbf{x}_k)} \quad (2.226)$$

ifadesi dikkate alınabilir. Burada;

$$q_k = [\nabla f(\mathbf{x}_{k+1}) + \sum \lambda_i \nabla g_i(\mathbf{x}_{k+1})] - [\nabla f(\mathbf{x}_k) + \sum \lambda_i \nabla g_i(\mathbf{x}_k)] \quad (2.227)$$

şeklinde tanımlanmaktadır. Eşitsizlik kısıtlarının, ilave yapay değişkenler kullanılarak eşitlik kısıtına dönüştürüldüğü kabul edilmektedir.

İkilenik programlama sorununun çözümü sonucu belirlenen doğrultuda, doğrudan adım atmak yerine optimal adım büyüklüğü ile adım atılması istendiğinde, atılacak adım büyüklüğünün tespit edilmesi için ilave bir fayda fonksiyonunun tanımlanması gerekir. Her ne kadar Wilson'a ait orijinal algoritmada adım büyüklüğü dikkate alınmadan, sadece yön ile bir sonraki nokta tayini yapılırsa da genellikle aşağıdaki gibi bir fayda fonksiyonu tanımı ve çözümü yapılır;

$$\Phi(\mathbf{x}^{(k+1)}) = f(\mathbf{x}^{(k+1)}) + \mathbf{r}g(\mathbf{x}^{(k+1)}) \quad (2.228)$$

Burada  $\mathbf{x}^{(k+1)}$  ifadesi, adım büyüklüğünü de içermekte olup  $\mathbf{r}$  yöneyi ceza katsayılarını içeren bir sabit sayı dizisidir. Adım büyüklüğü, kısıtların sağlandığı ve amaç fonksiyonunda azalma meydana getiren bir değerde olmalıdır. Ardışık ikilenik programlama sorununun işlem adımlarının daha iyi anlaşılması adına müteakip sorunu dikkate alabiliriz;

$$\begin{aligned} \text{Kısıt} \quad \min f(\mathbf{x}) &= -x_1^2 + (x_2 - 2)^2 \\ h(\mathbf{x}) &= 4x_1^2 + x_2^2 - 1 = 0 \end{aligned}$$

<sup>i</sup> Merit function



İfade edilen sorunu, Wilson'ın ardışık ikilenik programlama algoritması ile çözmek adına yapılacak ilk işlem, Lagrange fonksiyonunun tanımlanmasıdır;

$$L(\mathbf{x}, \lambda) = -x_1^2 + (x_2 - 2)^2 + \lambda(4x_1^2 + x_2^2 - 1)$$

Akabinde eldeki amaç, kısıt ve Lagrange fonksiyonları dikkate alınarak gradyan ve ikincil türev matrisi türetilebilir;

$$\nabla f(\mathbf{x}) = \begin{bmatrix} -2x_1 \\ 2x_2 - 4 \end{bmatrix}, \quad \nabla h(\mathbf{x}) = \begin{bmatrix} 8x_1 \\ 2x_2 \end{bmatrix}, \quad \nabla_{xx}^2 L(\mathbf{x}, \lambda) = \begin{bmatrix} 8\lambda - 2 & 0 \\ 0 & 2\lambda + 2 \end{bmatrix}$$

Sorunun çözümünde  $\mathbf{x}^{(0),T} = [2 \ 4]$  başlangıç noktası ve  $\lambda^{(0)} = 0.5$  Lagrange çarpanı olarak seçilip, söz konusu gradyan ve Hessian matrisi değerleri;

$$\nabla f(\mathbf{x}^{(0)}) = \begin{bmatrix} -4 \\ 4 \end{bmatrix}, \quad \nabla h(\mathbf{x}^{(0)}) = \begin{bmatrix} 16 \\ 8 \end{bmatrix}, \quad \nabla_{xx}^2 L(\mathbf{x}^{(0)}, \lambda^{(0)}) = \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}, \quad h(\mathbf{x}^{(0)}) = 31$$

şeklinde elde edilebilir. İfade edilen tanımlar dahilinde alt sorun niteliğindeki ikilenik programlama problemi kurgulanırsa;

$$\begin{aligned} \text{Bağıl} \quad \min Q(\mathbf{x}) &= [-4 \ 4] \Delta \mathbf{x} + \frac{1}{2} \Delta \mathbf{x}^T \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix} \Delta \mathbf{x} \\ &31 + [16 \ 8] \Delta \mathbf{x} = 0 \end{aligned}$$

veya cebirsel formda;

$$\begin{aligned} \text{Bağıl} \quad \min Q(\mathbf{x}) &= -4\Delta x_1 + 4\Delta x_2 + \Delta x_1^2 + \frac{3}{2} \Delta x_2^2 \\ &16\Delta x_1 + 8\Delta x_2 + 31 = 0 \end{aligned}$$

şeklinde elde edilmiş olur. İfade edilen bu sorunu ikilenik programlama çözüm algoritmalarından biri ile çözmek mümkündür. Uygun bir çözüm algoritması uygulaması<sup>i</sup> sonrasında sorunun çözüm yöneyi  $\Delta \mathbf{x}^{(0),T} = [-0.8036 \ -2.2679]$  ve Lagrange çarpanı  $\lambda^{(1)} = 0.3504$  şeklinde hesaplanabilir. Buradaki Lagrange çarpanı değeri, aynı zamanda ana problemin eşitlik kısıtı çarpan katsayısı olarak kullanılmaktadır. Alt problemin çözüm yöneyi dikkate alınarak ana sorunun yeni noktası;

<sup>i</sup> Örneğin Matlab fonksiyonu: “quadprog”





$$\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \Delta \mathbf{x}^{(0)}$$

ifadesinden hareketle;

$$\mathbf{x}^{(1)} = \begin{bmatrix} 2 \\ 4 \end{bmatrix} + \begin{bmatrix} -0.8036 \\ -2.2679 \end{bmatrix} = \begin{bmatrix} 1.1964 \\ 1.7321 \end{bmatrix}$$

olarak elde edilir. Görüldüğü üzere belirlenen doğrultu haricinde ilave bir adım büyüklüğü hesaplamasına gidilmemiş olup orijinal ardışık ikilenik programlama algoritmasına sadık kalmıştır. Yeni noktanın tespiti sonrasında ikinci adıma geçilir ve yeni nokta temel alınarak gradyan ve Hessian matrisi değerleri tekrar hesaplanır;

$$\begin{aligned} \nabla f(\mathbf{x}^{(1)}) &= \begin{bmatrix} -2.3929 \\ -0.5358 \end{bmatrix} & \nabla h(\mathbf{x}^{(1)}) &= \begin{bmatrix} 9.5712 \\ 3.4642 \end{bmatrix} \\ \nabla_{xx}^2 L(\mathbf{x}^{(1)}, \lambda^{(1)}) &= \begin{bmatrix} 0.8032 & 0 \\ 0 & 2.7008 \end{bmatrix} & h(\mathbf{x}^{(1)}) &= 7.7257 \end{aligned}$$

Müteakiben alt sorun niteliğindeki ikilenik programlama problemi kurgulanırsa;

$$\begin{aligned} \text{Kısıt} \quad \min Q(\mathbf{x}) &= [-2.3929 \quad -0.5358] \Delta \mathbf{x} + \frac{1}{2} \Delta \mathbf{x}^T \begin{bmatrix} 0.8032 & 0 \\ 0 & 2.7008 \end{bmatrix} \Delta \mathbf{x} \\ &7.7257 + [9.5712 \quad 3.4642] \Delta \mathbf{x} = 0 \end{aligned}$$

sorunu tanımlanmış olur. Uygun bir çözüm algoritması uygulaması sonrasında sorunun çözüm yöneyi  $\Delta \mathbf{x}^{(1),T} = [-0.7343 \quad -0.2013]$  ve Lagrange çarpanı  $\lambda^{(2)} = 0.3116$  olarak hesaplanabilir. Alt problemin çözüm yöneyi dikkate alınarak ana sorunun yeni noktası;

$$\mathbf{x}^{(2)} = \mathbf{x}^{(1)} + \Delta \mathbf{x}^{(1)}$$

ifadesinden hareketle;

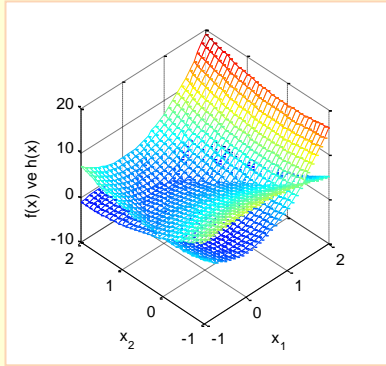
$$\mathbf{x}^{(2)} = \begin{bmatrix} 1.1964 \\ 1.7321 \end{bmatrix} + \begin{bmatrix} -0.7343 \\ -0.2013 \end{bmatrix} = \begin{bmatrix} 1.4621 \\ 1.5308 \end{bmatrix}$$

olarak elde edilir. Sonraki adımlarda benzer işlemler tekrar edilerek müteakip çizelge değerlerini<sup>(72)</sup> ve taranan noktaların amaç ile kısıt fonksiyonu üzerindeki yerlerini (Şekil 2.86) elde etmek mümkündür.

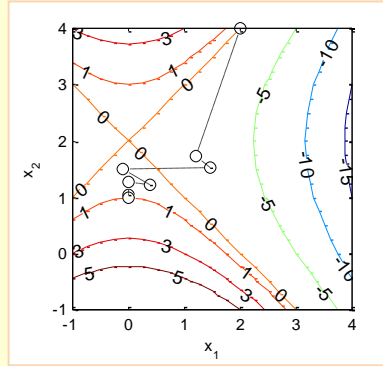




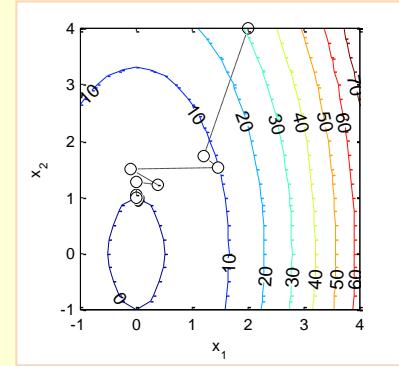
$x$	$\lambda$	$f$
[2.0000 4.0000]	0.5	0
[1.1964 1.7321]	0.3504	-1.3596
[1.4621 1.5308]	0.3116	-1.9176
[-0.113 1.5070]	0.327	0.2303
[0.3770 1.2150]	0.584	0.4741
[-0.005 1.2590]	0.588	0.5491
[0.0040 1.0270]	0.881	0.94671
[-0.001 1.0000]	0.996	1
[-0.0005 1.000]	1	1
[0.0000 1.0000]	1	1



(a) Amaç ile kısıt fonksiyonu yüzeyleri.



(b) Taranan noktalar ile amaç fonksiyonu eşdeğer çizgileri,  $f^* = 1$ .



(c) Taranan noktalar ile kısıt fonksiyonu eşdeğer çizgileri,  $g(x^*) = 0$ .

Şekil 2.86 Amaç ile kısıt fonksiyonu yüzeyi, eşdeğer çizgileri ve taranan noktalar.



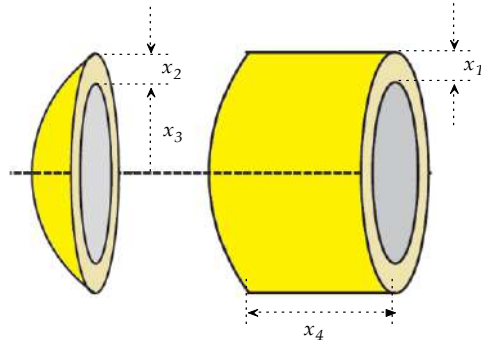
Şekil 2.87'de görüldüğü üzere 2000 psi basınca maruz kalan bir basınçlandırılmış hava tankının, belirli koşullar altında ebatlarının belirlenmesi istenmektedir. Kabin azami hacmi 750 ft<sup>3</sup>tür. Kap silindirik orta kısım ile kaynakla entegre edilmiş iki adet yarı küresel kabuk yapıdan müteşekkildir. Tasarımdaki temel gaye, öngörülen basınca dayanaklı, en az maliyete sahip yapının elde edilmesidir. Maliyeti etkileyen unsurlar; malzeme tutarı, şekil verme ve kaynak işçiliğidir. Tasarım parametreleri sırasıyla;

- $x_1$ : silindirin et kalınlığı,
- $x_2$ : yarı küresel kabin et kalınlığı,
- $x_3$ : yarı kürenin yarıçapı,
- $x_4$ : silindirin uzunluğudur.



**MATLAB uygulaması**





Şekil 2.87 Basıncılı kap tasarımı ve tasarım parametreleri.

Boyutsal parametreler “inch” birimi ile ölçümlendirilmektedir. Maliyetin en aza indirilmek istendiği eniyileme sorununu, uygun yapısal kısıtları da dikkate alarak aşağıdaki gibi tanımlamak mümkündür<sup>(73)</sup>;

$$\min f(x) = 0.6224 x_1 x_3 x_4 + 1.7781 x_2 x_3^2 + 3.1661 x_1^2 x_4 + 19.84 x_1^2 x_3$$

Kısıt

$$\begin{aligned} -\pi x_3^2 x_4 - \frac{4}{3} \pi x_3^2 + 1,296,000 \leq 0 & \quad -x_1 + 0.0193 x_3 \leq 0 & \quad 1 \times 0.0625 \leq x_1, x_2 \leq 99 \times 0.0625 \\ & \quad -x_2 + 0.0095 x_3 \leq 0 & \quad 10 \leq x_3, x_4 \leq 200 \\ & \quad x_4 - 240 \leq 0 \end{aligned}$$

```
clc; clear;
x0 = [0.7 0.3 30 190];

options=optimset('Algorithm','sqp');

[x,fval,[],[],[],grad,hessian]=fmincon(@myfun,x0,[-1 0 0.0193 0;0 -1 0.0095 0;0 0 0 1],[0;0;240],[],[],[0.0625;0.0625;10;10],[99*0.0625;99*0.0625;200;200],@mycon,
options)

function f = myfun(x)
f = 0.6224*x(1)*x(3)*x(4)+1.7781*x(2)*x(3)^2+3.1661*x(1)^2*x(4)+19.84*x(1)^2*x(3);
end

function [c,ceq] = mycon(x)
c=-pi*x(3)^2*x(4)-4/3*pi*x(3)^2+1296000;
ceq = [];
end
```





```
x =  
0.8736    0.4300    45.2658    200.0000  
  
fval =  
7.6581e+03  
  
grad =  
1.0e+03 *  
8.3102  
3.6433  
0.1931  
0.0270  
  
hessian =  
1.0e+04 *  
1.7515    1.2646   -0.0240   -0.0008  
1.2646    0.9154   -0.0170   -0.0010  
-0.0240   -0.0170    0.0004   -0.0000  
-0.0008   -0.0010   -0.0000    0.0001
```



## (6) Ceza fonksiyonu yöntemleri

Ceza fonksiyonu yöntemleri, eşitlik ve/veya eşitsizlik kısıtlarının ilave bazı katsayılar yardımı ile amaç fonksiyonuna eklenmesi, dolayısıyla yeniden tanımlanmış ve bir manada genişletilmiş amaç fonksiyonununun kısıtsız eniyileme sorununa dönüştürülerek çözülmesi düşüncesine dayanır<sup>(74)</sup>. Tabiatıyla farklı katsayı değerleri, farklı genişletilmiş amaç fonksiyonu tanımları ve kısıtsız optimizasyon süreci nihayetinde farklı çözüm noktaları anlamına gelir. Katsayı değerlerinin ardışık olarak değiştirilmesi neticesinde orijinal soruna ait en uygun noktanın bulunması hedeflenir. Bu meyanda ceza fonksiyonu yöntemleri ardışık kısıtsız eniyileme yöntemleri<sup>i</sup> olarak da anılırlar. Diğer taraftan kısıt tanımlamalarını içine alan genişletilmiş amaç fonksiyonu tanımlamaları, gradyan bilgisine ihtiyaç duyulmayan sezgisel yöntemler için de uygun bir çözüm zemini sağlamaktadır. Ceza fonksiyonu yöntemlerinin geçmişi 1950’li yıllara dayanır. Frisch’e ait “Yakınsak programlama için logaritmik potansiyel yöntem<sup>ii</sup>” isimli çalışma, ceza fonksiyonu yöntemlerinin ilk örneği olarak görülmektedir. Bununla beraber gerek kuramsal ve

<sup>i</sup> Sequential unconstrained minimization techniques (SUMT)

<sup>ii</sup> “The logarithmic potential method of convex programming”



gerekse de sayısal olarak yöntemi literatüre kazandıranlar, 1968 tarihli çalışmaları ile Fiacco ve McCormick'tir<sup>(75)</sup>.

Ceza fonksiyonu yöntemleri, kısıt ve başlangıç noktasının niteliğine göre değişik sınıflandırmalara tabi tutulabilir. Örneğin başlangıç noktasının, olası çözüm uzayında olması halinde dikkate alınan ceza fonksiyonu yöntemleri içsel nokta yöntemleri<sup>i</sup> olarak adlandırılır. Bu yöntemlerin diğer bir adı engel bazlı yöntemlerdir. Buna mukabil başlangıç noktasının olası çözüm uzayı dışında olması halinde yöntemlerin genel adı dışsal nokta yöntemleri<sup>ii</sup> olarak tanımlanır.

### (a) İçsel nokta yöntemleri

İçsel nokta bazlı ceza fonksiyonu yöntemlerinde, olası çözüm uzayındaki bir başlangıç noktasından hareket edilerek tarama yapılır ve her seferinde sınırları koşullarla belirli çözüm uzayı içerisinde kalınarak en uygun noktaya ulaşılması hedeflenir. Sadece eşitsizlik kısıtı içeren optimizasyon sorununun;

$$\begin{aligned} & \min f(\mathbf{x}) \\ \text{Kısıt} & \\ & g_j(\mathbf{x}) \leq 0, \quad j = 1, 2, \dots, m \end{aligned}$$

şeklinde tanımlandığı bir kısıtlı eniyileme probleminde genişletilmiş amaç fonksiyonu<sup>iii</sup>;

$$\min F(\mathbf{x}, r_k) = f(\mathbf{x}) + r_k \sum_{j=1}^m G(g_j(\mathbf{x})) \quad (2.229)$$

biçiminde ifade edilebilir. Burada  $r_i$  ardışık olarak değeri değiştirilen ceza parametresi,  $G(g_j(\mathbf{x}))$  ise kısıtları içeren bir fonksiyonlar fonksiyonudur. Genellikle söz konusu fonksiyonlar fonksiyonu;

$$G(g_j(\mathbf{x})) = -\frac{1}{g_j(\mathbf{x})} \quad (2.230)$$

veya

<sup>i</sup> “Interior point methods” veya “Barrier methods”

<sup>ii</sup> Exterior point methods

<sup>iii</sup> Augmented objective function



$$G(g_j(x)) = \log(-g_j(x)) \quad (2.231)$$

olarak dikkate alınır. Tabiatıyla çözümün, olası çözüm uzayı içerisinde kalmasını sağlayacak farklı fonksiyonlar fonksiyonu tanımlamaları yapmak mümkündür. Ardışık işlemlerin yapıldığı tarama süreçlerinde  $r_k$  değeri her seferinde belli bir sistematige göre, örneğin  $r_{k+1} = r_k/2$  ifadesine göre azaltılır. Optimizasyon sorununun eşitlik ve eşitsizlik kısıtı içeren;

$$\begin{aligned} & \min f(x) \\ \text{Kısıt} & \\ & h_i(x) = 0, \quad i = 1, 2, \dots, n \\ & g_j(x) \leq 0, \quad j = 1, 2, \dots, m \end{aligned}$$

şeklinde tanımlandığı bir kısıtlı eniyileme probleminde genişletilmiş amaç fonksiyonu;

$$\min F(x, r_k) = f(x) + r_k \sum_{j=1}^m G(g_j(x)) + R(r_k) \sum_{i=1}^n h(x)_i^2 \quad (2.232)$$

şeklinde tanımlanabilir. Burada eşitlik kısıtını içeren terimin katsayısı niteliğindeki ifadeyi;

$$R(r_k) = \frac{1}{\sqrt{r_k}} \quad (2.233)$$

denklemleri ile somutlaştırmak mümkündür. Gerek ilk ve gerekse de daha kapsamlı ikinci sorunun çözümünde herhangi bir kısıtsız optimizasyon yöntemi dikkate alınabilir.

İçsel nokta bazlı ceza fonksiyonu yönteminin daha iyi anlaşılması amacıyla örnek bir sorunu ele alabiliriz. Buna göre;

$$\begin{aligned} & \min f(x) = x^2 - 5x + 6 \\ \text{Kısıt} & \\ & g_1(x) = 4 - x \leq 0 \\ & g_2(x) = x - 5 \leq 0 \end{aligned}$$

sorununun çözmek için öncelikle genişletilmiş amaç fonksiyonunu,  $r_1 = 1$  ceza parametresini dikkate alarak aşağıdaki gibi tanımlamak mümkündür;

$$\min F(x, r_0) = x^2 - 5x + 6 - \left( \frac{1}{4-x} + \frac{1}{x-5} \right)$$

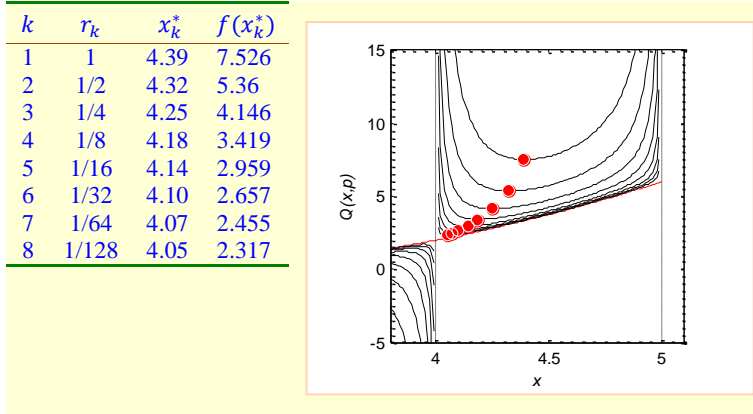




Böyle bir amaç fonksiyonunun minimum değeri, kısıtsız optimizasyon tekniklerinden biri ile tarandığında elde edilecek optimal nokta değeri  $x_1^* = 4.39$  olarak kolayca belirlenebilir. Benzer şekilde ceza parametresinin  $r_2 = 1/2$  değerine düşürülmesi halinde;

$$\min F(x, r_1) = x^2 - 5x + 6 - \frac{1}{2} \left( \frac{1}{4-x} + \frac{1}{x-5} \right)$$

biçiminde yazılabilen genişletilmiş amaç fonksiyonu dikkate alınarak  $x_2^* = 4.32$  optimal nokta değerine ulaşılabilir. Müteakip çizelgede farklı ceza parametre değerleri ve elde edilen genişletilmiş amaç fonksiyonuna göre ulaşılan optimal nokta değerleri, Şekil 2.88'de ise genişletilmiş amaç fonksiyonları görülmektedir;



**Şekil 2.88** Ceza parametre değerine göre şekillenen genişletilmiş amaç fonksiyonu çizimleri, ulaşılan optimal noktalar ve olası çözüm bölgesi.

Şekilden de görüleceği üzere analitik çözümü  $x^* = 4, f(x^*) = 2$  olan sorunda ceza parametresinin her adımda küçültülmesi ile kısıtlar çerçevesinde optimal noktaya bir adım daha yaklaşılmakta ve istenilen hassasiyette çözüme ulaşılabilir. Bununla beraber kısıtsız optimizasyon sürecine başlamak için gerekli olan başlangıç noktasının, olası çözüm uzayında olmasına dikkat edilmelidir. Aksi halde genişletilmiş amaç fonksiyonu, tarama sürecini çözüm bölgesinden uzaklaştırabilir.





## (b) Dışsal nokta yöntemleri

Dışsal nokta bazlı ceza fonksiyonu yöntemlerinde, olası çözüm uzayı dışındaki bir başlangıç noktasından hareket edilerek tarama yapılır ve her seferinde sınırları koşullarla belirli çözüm uzayı dışında kalınarak en uygun noktaya ulaşılması hedeflenir. Sadece eşitsizlik kısıtı içeren optimizasyon sorununun;

$$\begin{aligned} & \min f(\mathbf{x}) \\ \text{Kısıt} \quad & g_j(\mathbf{x}) \leq 0, \quad j = 1, 2, \dots, m \end{aligned}$$

şeklinde tanımlandığı bir kısıtlı eniyileme probleminde, bir önceki yönteme benzer şekilde genişletilmiş amaç fonksiyonu;

$$\min F(\mathbf{x}, r_k) = f(\mathbf{x}) + r_k \sum_{j=1}^m G(g_j(\mathbf{x})) \quad (2.234)$$

biçiminde ifade edilebilir. Burada  $r_i$  ardışık olarak değeri değiştirilen ceza parametresi ve  $G(g_j(\mathbf{x}))$  kısıtları içeren bir fonksiyonlar fonksiyonudur. Genellikle söz konusu fonksiyonlar fonksiyonu;

$$G(g_j(\mathbf{x})) = \langle 0, g_j(\mathbf{x}) \rangle \quad (2.235)$$

veya

$$G(g_j(\mathbf{x})) = \{\langle 0, g_j(\mathbf{x}) \rangle\}^q, \quad q \geq 2 \quad (2.236)$$

olarak dikkate alınır. Tabiatıyla burada eşitsizlik kısıtının ihlal edilip edilmediği önemlidir. Zira  $\langle 0, g_j(\mathbf{x}) \rangle$  ifadesi;

$$G \begin{cases} \text{Eğer } & g_j(\mathbf{x}) > 0 & g_j(\mathbf{x}) \\ \text{Eğer } & g_j(\mathbf{x}) \leq 0 & 0 \end{cases} \quad (2.237)$$

şeklinde dikkate alınır. Ardışık işlemlerin yapıldığı tarama süreçlerinde  $r_k$  değeri, her seferinde belli bir sistematığe göre, örneğin  $r_{k+1} = 2r_k$  ifadesine göre arttırılır.

Optimizasyon sorununun eşitlik ve eşitsizlik kısıtı içeren;





$$\begin{aligned} & \min f(\mathbf{x}) \\ \text{Kısıt} \quad & h_i(\mathbf{x}) = 0, \quad i = 1, 2, \dots, n \\ & g_j(\mathbf{x}) \leq 0, \quad j = 1, 2, \dots, m \end{aligned}$$

formunda ifade edildiği bir kısıtlı eniyileme probleminde genişletilmiş amaç fonksiyonu;

$$\min F(\mathbf{x}, r_k) = f(\mathbf{x}) + r_k \sum_{j=1}^m G(g_j(\mathbf{x})) + r_k \sum_{i=1}^n h(\mathbf{x})_i^2 \quad (2.238)$$

şeklinde tanımlanabilir. Eşitsizlik kısıtını içeren  $G$  fonksiyonlar fonksiyonu bir önceki tanımlar ile benzer şekilde ifade edilebilir. Dışsal nokta bazlı ceza fonksiyonu yönteminin daha iyi anlaşılması amacıyla bir önceki yöntemde çözülen örnek sorun tekrar dikkate alınabilir;

$$\begin{aligned} & \min f(x) = x^2 - 5x + 6 \\ \text{Kısıt} \quad & g_1(x) = 4 - x \leq 0 \\ & g_2(x) = x - 5 \leq 0 \end{aligned}$$

Sorunun çözümü sadedinde yapılacak ilk işlem  $r_1 = 1$  ceza parametresini dikkate alarak genişletilmiş amaç fonksiyonunun tanımlanmasıdır;

$$\min F(x, r_k) = x^2 - 5x + 6 + \{< 0, g_j(x) >\}^2$$

Bu ifadedeki kısıt etkisi başlangıç noktasına göre değerlendirilir. Tarama sürecinde başlangıç noktası olarak olası çözüm uzayı dışındaki  $x_0 = 1$  noktası dikkate alındığında;

$$\begin{aligned} g_1(1) &= 4 - 1 = 3 > 0 \\ g_2(1) &= 1 - 5 = -4 \leq 0 \end{aligned}$$

olduğu görülür. Dolayısıyla ilk kısıtın ihlal edildiği, ikinci kısıtın ise sağlandığı gözlemlenmektedir. Bu nedenle genişletilmiş amaç fonksiyonu;

$$\min F(x, r_k) = x^2 - 5x + 6 + (4 - x)^2$$

şeklinde elde edilebilir. Bu fonksiyonun uygun bir kısıtsız eniyileme algoritması ile çözülmesi halinde optimal nokta değeri  $x_1^* = 3.3$  olarak kolayca elde edilebilir. Sonraki aşamada





$r_2 = 2 \times r_1 = 2$  ceza parametresi ve ulaşılan nokta baz alınarak kısıt ihlali hususu değerlendirildiğinde;

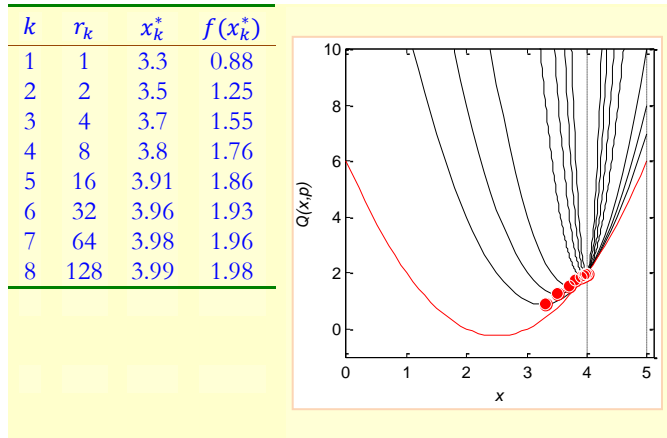
$$g_1(3.3) = 4 - 3.3 = 0.7 > 0$$

$$g_2(3.3) = 3.3 - 5 = -1.7 \leq 0$$

olduğu görülür. Dolayısıyla ilk adımda olduğu gibi bu adımda da ilk kısıtın ihlal edildiği, ikinci kısıtın ise sağlandığı gözlemlenmektedir. Bu nedenle genişletilmiş amaç fonksiyonu ceza parametresi hariç ilk adımla aynıdır;

$$\min F(x, r_k) = x^2 - 5x + 6 + 2(4 - x)^2$$

Bu fonksiyonun uygun bir kısıtsız eniyileme algoritması ile çözülmesi halinde optimal nokta değeri  $x_2^* = 3.5$  olarak elde edilebilir. Müteakip çizelge dahilinde farklı ceza parametre değerleri ve elde edilen genişletilmiş amaç fonksiyonuna göre ulaşılan optimal nokta değerleri, Şekil 2.89'da ise genişletilmiş amaç fonksiyonları görülmektedir;



**Şekil 2.89** Ceza parametre değerine göre şekillenen genişletilmiş amaç fonksiyonu çizimleri, ulaşılan optimal noktalar ve olası çözüm bölgesi.

Şekilden de görüleceği üzere ceza parametresinin her adımda büyütülmesi ile olası çözüm uzayı dışında optimal noktaya bir adım daha yaklaşılmakta ve istenilen hassasiyette çözüme





ulaşılabilir. Genellikle dışsal nokta yöntemlerinin uygulaması daha kolay olup bu yöntemleri kullanarak daha hassas çözümlere daha az adımda ulaşmak mümkün olabilmektedir<sup>(76)</sup>.



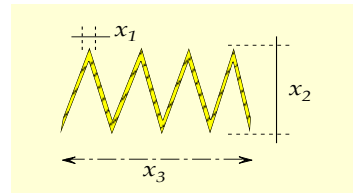
**MATLAB uygulaması**

Çeki ve bası kuvvetlerine maruz kalan esnek bir yayın, belirli koşullar altında en az ağırlık değeri ile tasarlanması istenmektedir. Bu meyanda dikkate alınan üç tasarım parametresi söz konusu olup bu parametreler sırasıyla;

- $x_1$ ; yay telinin ortalama çapı,
- $x_2$ ; sargı halindeki yayın çapı,
- $x_3$ ; yay sargı sayısıdır.

Şekil 2.89'da tasarım parametreleri görülmektedir. Tasarım kapsamında dikkate alınabilecek amaç ve kısıt fonksiyonları aşağıdaki gibi tanımlanabilir;

$$\begin{aligned} \text{Kısıt} \quad \min f(x) &= (x_3 + 2)x_1^2 x_2 \\ \frac{4x_2^2 - x_1 x_2}{12566(x_2 x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} - 1 &\leq 0 & 1 - \frac{x_2^3 x_3}{71785x_1^4} &\leq 0 & 0.05 \leq x_1 \leq 2.00 \\ & & & & 0.25 \leq x_2 \leq 1.30 \\ & & 1 - \frac{140.45x_1}{x_2^2 x_3} &\leq 0 & 2.00 \leq x_3 \leq 15.0 \\ & & \frac{x_1 + x_2}{1.5} - 1 &\leq 0 & \end{aligned}$$



**Şekil 2.89** Yay tasarımı ve tasarım parametreleri.



```
clc; clear;
x0 = [0.05 1.3 15];

options=optimset('Algorithm','interior-point');
[x,fval]=fmincon(@myfun,x0,[],[],[],[],[0.05;0.25;2],[2;1.3;15],@mycon,options)

function f = myfun(x)
    f = (x(3)+2)*x(1)^2*x(2);
end

function [c,ceq] = mycon(x)
    c(1) = 1 - x(2)^3*x(3)/71785/x(1)^4;
    c(2) = (4*x(2)^2-x(1)*x(2))/(12566*(x(2)*x(1)^3)-x(1)^4) + 1/(5108*x(1)^2)-1;
    c(3) = 1 - 140.45*x(1)/(x(2)^2*x(3));
    c(4) = (x(1)+x(2))/1.5-1;
    ceq = [];
end

x = 0.0500    0.3745    8.5457

fval = 0.0099
```





## İrfan Damlası



**D**

ıllar önce hastaneye çok ağır hasta bir kız getirdiler... Tek yaşam şansı beş yaşındaki kardeşinden acil kan nakli yapılmasıydı. Küçük oğlan aynı hastalıktan mucizevî şekilde kurtulmuş ve kanında o hastalığın mikroplarını yok eden bağışıklık oluşmuştu.

Doktor durumu küçük kardeşe anlattı, ablasına kan verip veremeyeceğini sordu. Masum gözleri ile kardeş önce derin bir nefes aldı ve şöyle dedi;

“Eğer kurtulacaksa, veririm kanımı...”

Kan nakledilirken ablasının gözlerinin içine bakıyor ve gülümsüyordu.

Kızın yanaklarına yeniden renk gelmeye başlamıştı, ama küçük çocuğun yüzü de giderek soluyordu, ardından gülümsemesi de yok oldu. Titreyen bir sesle doktora sordu:

“Hemen mi öleceğim..?”

Küçük çocuk doktoru yanlış anlamış, ablasına vücudundaki tüm kanı verip öleceğini sanmıştı.





- [1] Alıntı: <[http://img11.deviantart.net/747f/i/2008/316/e/a/a\\_portrait\\_of\\_determinism\\_by\\_relaxed\\_e\\_r.jpg](http://img11.deviantart.net/747f/i/2008/316/e/a/a_portrait_of_determinism_by_relaxed_e_r.jpg)> [12 Haziran 2015].
- [2] Alıntı: <[http://www.city-journal.org/html/7\\_3\\_urbanities-isaac.html](http://www.city-journal.org/html/7_3_urbanities-isaac.html)> [15 Haziran 2015].
- [3] W. Punurai, Optimization by a Graphical Solution, Alıntı: <<http://www.egmu.net/civil/wonsiri/egce491/Lecture11.pdf>> [15 Haziran 2015].
- [4] Alıntı: <<http://mgiprocess.com/user/cimage/Linde-gas.jpg>> [15 Haziran 2015].
- [5] Alıntı: <<http://i.forbesimg.com/images/forbes/2000/1113/microwave.jpg>> [16 Haziran 2015].
- [6] Optimization Modeling with Spreadsheets, Second Edition. Kenneth R. Baker # 2011 John Wiley & Sons, Inc. Published 2011 by John Wiley & Sons, Inc., sf. 377-381. Alıntı: <<http://onlinelibrary.wiley.com/doi/10.1002/9780470949108.app2/pdf>> [16 Haziran 2015].
- [7] Nonlinear programming solution techniques, Module D. Alıntı: <[http://wps.prenhall.com/wps/media/objects/1159/1186960/cdrom\\_modules/module\\_d.pdf](http://wps.prenhall.com/wps/media/objects/1159/1186960/cdrom_modules/module_d.pdf)> [16 Haziran 2015].
- [8] Alıntı: <<http://hss-prod.hss.aol.com/hss/storage/midas/4ec7f03268fd63fcd082de9c7ea63339/201571198/463763436.jpg>> [16 Haziran 2015].
- [9] Alıntı: <<http://www.iip.ist.i.kyoto-u.ac.jp/member/cuturi/Teaching/ORF522/lec1v2.pdf>> [17 Haziran 2015].
- [10] Alıntı: <<http://roseanalytics.com/wp-content/uploads/2013/09/SimplexLattice-300x217.png>> [17 Haziran 2015].
- [11] Alıntı: <[http://www.jstor.org/stable/2627082?seq=1#page\\_scan\\_tab\\_contents](http://www.jstor.org/stable/2627082?seq=1#page_scan_tab_contents)> [17 Haziran 2015].
- [12] Origins of the Simplex Method, G. B. Dantzig. Alıntı: <<http://www.dtic.mil/dtic/tr/fulltext/u2/a182708.pdf>> [17 Haziran 2015].
- [13] Alıntı: <<http://roseanalytics.com/wp-content/uploads/2013/09/MultidemProbs-300x217.png>> [17 Haziran 2015].
- [14] Tributes to George Dantzig and Leonid Khachiyan, M. Todd. Alıntı: <<http://stanford.edu/group/SOL/GBD/Dantzig-Khachiyan.pdf>> [17 Haziran 2015].
- [15] Alıntı: <<http://static-content.springer.com/lookinside/art%3A10.1007%2F02579150/000.png>> [17 Haziran 2015].
- [16] A New Polynomial-time Algorithm for Linear Programming, N. Karmarkar, Combinatorica 4 (4) (1984) 373-395. Alıntı: <<http://retis.sssup.it/~bini/teaching/optim2010/karmarkar.pdf>> [17 Haziran 2015].
- [17] Alıntı: <[http://csclab.murraystate.edu/bob.pilgrim/445/munkres\\_files/image7\\_2.gif](http://csclab.murraystate.edu/bob.pilgrim/445/munkres_files/image7_2.gif)> [23 Haziran 2015].
- [18] The Hungarian method for the assignment problem, H. W. Kuhn. Alıntı: <<http://tom.host.cs.st-andrews.ac.uk/CS3052-CC/Practicals/Kuhn.pdf>> [24 Haziran 2015].
- [19] Hungarian algorithm. Alıntı: <[https://en.wikipedia.org/wiki/Hungarian\\_algorithm](https://en.wikipedia.org/wiki/Hungarian_algorithm)> [24 Haziran 2015].



- [20] Optimization methods in management science/operation research, Alıntı: < ocw.mit.edu/courses/sloan-school-of-management> [07 Eylül 2015].
- [21] Linear programming, Alıntı: < college.cengage.com/mathematics/larson/elementary\_linear/4e/shared/downloads/c09s3.pdf> [24 Haziran 2015].
- [22] The Big-M method, Alıntı: < www-home.uwo.ca/~heinicke/courses/ 236\_03/bigM.pdf> [08 Eylül 2015].
- [23] Optimization methods: linear programming, D. N. Kumar, Alıntı: < www.slideshare.net/shamjithkeyem/numerical-analysis-m3-I6slides> [11 Eylül 2015].
- [24] Karmarkar's method, W. Kang, Alıntı: <wit.kaist.ac.kr/Linear%20algebra/ch8.1-8.3\_Linear%20Programming%20and%20Game%20Theory.pdf> [11 Eylül 2015].
- [25] Paper review—Interior point methods for LP, Y. Hu and C. Yu, Alıntı: <https://openeclass.teimes.gr/modules/documents/file.php/CIED189/Eng614-03-Interior.ppt>[15 Eylül 2015].
- [26] S.S. Rao, Engineering optimization: theory and practice, 3rd Edition, John Wiley and Sons, Inc., 1996, sf. 247.
- [27] Jacobi's bound for a diffiety defined by a quasi-regular system, F. Ollivier ve B. Sadik, Alıntı: <arxiv.org/pdf/math/0701838.pdf>[28 Eylül 2015].
- [28] The assignment problem and the Hungarian method, Alıntı: <www.math.harvard.edu/archive/20\_spring\_05/handouts/assignment\_overheads.pdf>[29 Eylül 2015].
- [29] Hungarian algorithm, V. Movahedi, Alıntı: <elderlab.yorku.ca/~vida/ ppt/hungarian/ppt> [01 Ekim 2015].
- [30] Duality theory in LP, C. Jyothimon, Alıntı: <www.slideshare.net /jyothimonc/duality-in-linear-programming> [05 Ekim 2015].
- [31] Convex optimization, Alıntı: <www.convexoptimization.com/dattorro/ duality.html> [05 Ekim 2015].
- [32] Linear programming: Chapter 5 Duality, R. J. Vanderbei, Alıntı: www.princeton.edu/~rvdb/542/lectures/ lec5.pdf, [05 Ekim 2015].
- [33] Lagrange, Alıntı< wiki.ubc.ca/Lagrange> [19 Ekim 2015].
- [34] Karush-Kuhn-Tucker conditions, G. Gordon ve R. Tibshirani, https Alıntı<://www.cs.cmu.edu/~ggordon/ 10725-F12/slides/16-kkt.pdf > [21 Ekim 2015].
- [35] Lagrangian duality: Basics, Encyclopedia of optimization, C. A. Floudas ve P. M. Pardalos, 2nci Baskı, 2009. Alıntı< https://books.google.com.tr> [27 Ekim 2015].
- [36] Jack Kiefer (statistician), Alıntı<https://en.wikipedia.org/wiki/ Jack\_Kiefer\_(statistician)> [03 Kasım 2015].
- [37] Sequential minimax search for a maximum, J. Kiefer, Journal: Proc. Amer. Math. Soc. 4 (1953), 502-506.
- [38] Alıntı< https://1millionmonkeystyping.files.wordpress.com/2013/03/ sun.jpg> [12 Kasım 2015].
- [39] J. Kirk, sunflower, Matlab exchange central, Alıntı<www.mathworks. com/matlapcentral/fileexchange> [13 Kasım 2015].
- [40] E.K.P. Chong ve S.H. Zak, An introduction to optimization, 2nci Baskı, , John Wiley and Sons, Inc., 2001, sf. 99.



- [41] Pingala, Alıntı<<https://en.wikipedia.org/wiki/Pingala>> [16 Kasım 2015].
- [42] Fibonacci, Alıntı<<https://en.wikipedia.org/wiki/Fibonacci>> [16 Kasım 2015].
- [43] Alıntı< <http://www.sciencedump.com/content/how-are-fibonacci-numbers-expressed-nature>> [16 Kasım 2015].
- [44] Alıntı<<http://conservefewell.org/humanitys-relationship-with-the-natural-world/>>[16 Ksm 2015].
- [45] Hayley Rhind papyrus, Alıntı<<http://9044c4.medialib.edu.glogster.com>> [27 Kasım 2015].
- [46] J. M. Papakonstantinou ve R. A. Tapia, Origin and Evolution of the Secant Method in One Dimension, The American Mathematical Monthly, Vol. 120, No. 6 (June–July 2013), pp. 500-518.
- [47] Powell, M. J. D. (1964). "An efficient method for finding the minimum of a function of several variables without calculating derivatives". Computer Journal 7 (2): 155–162.
- [48] Unconstrained optimization, Alıntı< [web.mae.ufl.edu/nkim/eas6939/ UnconstrainedOpt.pdf](http://web.mae.ufl.edu/nkim/eas6939/UnconstrainedOpt.pdf)> [11 Aralık 2015].
- [49] Beam Deflection formulae, Alıntı< [ruina.mae.cornell.edu/Courses/ME4735-2012/Rand4770 Vibrations/BeamFormulas.pdf](http://ruina.mae.cornell.edu/Courses/ME4735-2012/Rand4770Vibrations/BeamFormulas.pdf)> [14 Aralık 2015].
- [50] M.A. Cauchy, Méthode générale pour la résolution des systèmes d'équations simultanées, C.R. Acad. Sci. Paris, 25 :536-538, 1847. Alıntı : < [www.cs.xu.edu/math/Sources/Cauchy/Orbits/1847%20CR%20536\(383\).pdf](http://www.cs.xu.edu/math/Sources/Cauchy/Orbits/1847%20CR%20536(383).pdf)> [21 Aralık 2015].
- [51] Two degree of freedom systems, Alıntı< [web.itu.edu.tr/~gundes/ 2dof.pdf](http://web.itu.edu.tr/~gundes/2dof.pdf)> [24 Aralık 2015].
- [52] Y.H. Dai ve Y. Yuan, A nonlinear conjugate gradient method with a strong global convergence property, SIAM J. Optim., Vol. 10, sf. 177-182, 1999.
- [53] Approximate methods: Rayleigh Ritz method, Alıntı<[sameradeeb.srv.ualberta.ca](http://sameradeeb.srv.ualberta.ca)> [29 Ocak 2015].
- [54] Davidon-Fletcher-Powell formula, Wikipedia, Alıntı<[https://en.wikipedia.org/wiki/Davidon-Fletcher-Powell\\_formula](https://en.wikipedia.org/wiki/Davidon-Fletcher-Powell_formula)> [02 Şubat 2016].
- [55] Broyden-Fletcher-Goldfarb-Shanno algorithm, Wikipedia, Alıntı<[https://en.wikipedia.org/wiki/Broyden-Fletcher-Goldfarb-Shanno\\_algorithm](https://en.wikipedia.org/wiki/Broyden-Fletcher-Goldfarb-Shanno_algorithm)> [02 Şubat 2016].
- [56] Quasi-Newton Method, Wikipedia, Alıntı<[https://en.wikipedia.org/wiki/ Quasi-Newton-method](https://en.wikipedia.org/wiki/Quasi-Newton_method)> [02 Şubat 2016].
- [57] V. S. Vassiliadis ve R. Conejeros, Broyden family of methods and the BFGS update, Encyclopedia of Optimization . Editörler : C. A. Floudas ve P. M. Pardalos, 2nci baskı, Springer, 2009, sf. 345-346.
- [58] Multi variable optimization procedures, Successive linear programming, Alıntı< [www.mpri.lsu.edu/textbook/chapter6-b.htm#successive](http://www.mpri.lsu.edu/textbook/chapter6-b.htm#successive)> [19 Şubat 2016].
- [59] M. Avriel ve A. C. Williams, An extension of geometric programming with applications in engineering optimization, Journal of Engineering Mathematics, Vol. 5, No. 3, 1971, sf. 187-194.
- [60] Multi-stage heat exchanger, Alıntı< [image.made-in-china.com](http://image.made-in-china.com)> [18 Şubat 2016].
- [61] PHPSimplex, Optimizing resources with linear programming, Alıntı< [www.phpsimplex.com/en/](http://www.phpsimplex.com/en/)> [22 Şubat 2016].
- [62] X. Chen ve M. M. Kostreva, Methods of feasible directions: A review, Progress in Optimization, Vol. 39, sf. 205-219.
- [63] D. Chakraborty, Nonlinear programming – Constrained optimization techniques, Alıntı<[textofvideo.nptel.iitm.ac.in/111105039/lec32.pdf](http://textofvideo.nptel.iitm.ac.in/111105039/lec32.pdf)> [29 Şubat 2016].





- [64] P. Wolfe, The reduced gradient method, Rand Corporation, 1962. Alıntı <web.stanford.edu> [29 Şubat 2016].
- [65] J. Abadie ve J. Carpentier, Generalization of the Wolfe reduced gradient method to the case of nonlinear constraints, in Optimization. Ed. R. Fletcher, Academic Press., 1969. Alıntı <www.mpri.lsu.edu> [02 Mart 2016].
- [66] S.S. Rao, Engineering optimization: theory and practice, 3rd Edition, John Wiley and Sons, Inc., 1996, sf. 258.
- [67] R. B. Wilson, A simplicial method for convex programming, Ph D thesis, Harvard University, 1963.
- [68] M. J. Goldsmith, Sequential quadratic programming methods based on indefinite hessian approximations, Ph D thesis, Stanford University, 1999.
- [69] Constrained optimization, Alıntı<www2.mae.ufl.edu/nkim/eas6939/ConstrainedOpt.pdf> [18 Mart 2016].
- [70] P. E. Gill ve E. Wong, Sequential quadratic programming methods, Alıntı<www.ccom.ucsd.edu/~peg/papers/sqpReview.pdf> [18 Mart 2016].
- [71] B. Goodman, Sequential quadratic programming, Alıntı<https://optimization.mccormick.northwestern.edu/index.php/Sequential\_quadratic\_programming > [18 Mart 2016].
- [72] SQP and PDIP algorithms for nonlinear programming, Alıntı< www.doc.ic.ac.uk/~br/berc/CODsqpipnotes.pdf> [23 Mart 2016].
- [73] H. Garg, Solving structural engineering design optimization problems using an artificial bee colony algorithm, Journal of Industrial and Management Optimization, Volume 10, Number 3, July 2014, pp. 777–794.
- [74] C-R Lee, Numerical optimization, Alıntı< www.cs.nthu.edu.tw/~cherung/teaching/2011cs5321/handout9.pdf> [28 Mart 2016].
- [75] D. Shanno, who invented the interior-point method?, Documenta Mathematica Extra volume ISMP (2012) 55-64, Alıntı< www.math. uiuc.edu/documenta/vol-ism/20\_shanno-david.pdf> [28 Mart 2016].
- [76] Algorithms for constrained optimization, Alıntı <https://www.me. utexas.edu/~jensen/ORMM/suplements/units/nlp\_methods/const\_opt.pdf> [31 Mart 2016].





## SEZGİSEL YÖNTEMLER

“Zekâ veya kuvvet değil, gerçek gücümüzü ortaya çıkaran sürekli çabadır...”

Winston Churchill <sup>(2)</sup>

	Sayfa
3.1 Giriş	396
3.2 Nelder-Mead yöntemi	399
3.3 Tabu algoritmaları	408
3.4 Markov Zinciri	423
3.5 Rastlantısal tarama yöntemi	426
3.6 Benzetimli tavlama yöntemi	437
3.7 Karınca kolonisi algoritması	451
3.8 Yapay arı sürüsü algoritması	475
3.9 Parçacık sürü algoritması	489
3.10 Genetik algoritma	508
İrfan Damlası	544
Kaynakça	545

(1)



# 3





### 3.1 Giriş



Sezgisellik kavramının geçmişi oldukça eskilere dayanır. İngilizce literatürdeki karşılığı olan “heuristic” kelimesinin Latin edebiyatındaki kökeni “heuristicus”, eski Yunan dilindeki kökeni ise “heuristicus” olup çıkış kaynağı itibari ile sözlük anlamı keşfetmektir. Çağdaş edebiyatta bu kavram, “tecrübeye dayalı öğrenme ve gelişme”, “tecrübe ve deneme-yanılma üzerine kurulu sorun çözme ya da öğrenme”, “başarıyı artırmak için geri besleme temelli kendi kendine öğrenme yöntemlerini kullanma” anlamlarına gelmektedir<sup>(3)</sup>. Tanımlardan da anlaşılacağı üzere sezgisellik kavramı, pek çok bilimsel disiplinde dikkate alınmakta olup özellikle yapay zekâ, psikoloji, hukuk, bilgisayar bilimleri ve hatta tıp alanında alt dallara sahiptir. Yapay zekâ literatüründeki ilk kullanımına, 1945 yılında G. Polya (1887-1985) tarafından yayımlanan “**Nasil çözülür: matematiksel modellemede yeni bir bakış**”<sup>(4-5)</sup> başlıklı kitapta rastlandığı söylenebilir<sup>(4-5)</sup>. Sezgisel yöntemleri, yapay zekâ disiplininin bir alt dalı olarak nitelendirmek mümkündür. Özellikle 1970’li yılların başlarında karmaşıklık kuramının geliştirilmesi ile beraber literatürde sıkça rastlanan bazı sorunların (gezici-satıcı sorunu, rota planlaması vb.) tam anlamıyla NP-Zor sorunlar olduğu, kabul edilebilir bir zaman dilimi içerisinde analitik veya daha da tercih edilebilir olan en uygun çözüme ulaşmanın mümkün olmayabileceği kanaati yaygınlaşmış, bu nedenle daha kısa zaman dilimi içerisinde yaklaşık da olsa makul çözümler üreten yöntemlere başvurulmaya başlanmıştır. Bu meyanda o yıllardan günümüze neredeyse yüzlerce sezgisel yöntem geliştirilmiştir. Öyle ki zaman içerisinde sezgisellik kavramı üst tanımlamalara tabi tutulmuş ve başlangıçta sezgisel yöntem olarak adlandırılan yöntemler zaman içerisinde üst-sezgisel<sup>(iii)</sup> ve daha da üst yapılanmalar içeren öte-sezgisel<sup>(iv)</sup> yöntemlere evrilmiştir.

Sezgisel yöntemler tanımlarından da anlaşılacağı üzere büyük oranda tecrübe üzerine dayalı yöntemlerdir. En temel sezgisel yöntem “deneme-yanılma” yöntemi olup elde edilen sonuca göre yeni arayışa yönelir. Sezgisel yöntemin sorunu çözmede başarılı olabilmesi, büyük oranda üzerinde çalışılan sorunun iyi anlaşılmasına ve elde edilen tecrübenin azami derecede değerlendirilmesine bağlıdır. Bu anlamda sezgisel yöntem, sorun odaklı çözüm yöntemi olarak adlandırılabilir<sup>(6)</sup>. Böyle bir bağın olumlu yönü, soruna kısa sürede uygun bir çözümün bulunmasıdır. Ancak diğer taraftan sorun odaklı çözüm yöntemi, yerel bir çözümde takılıp

<sup>i</sup> Heuristic

<sup>ii</sup> “How to solve it: a new aspect of mathematical model”

<sup>iii</sup> Meta-heuristics

<sup>iv</sup> Hyper-heuristics

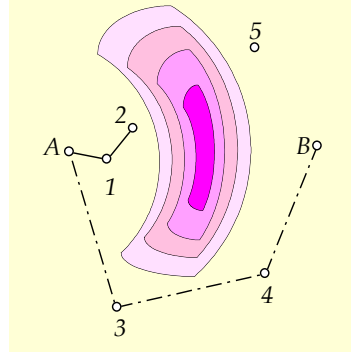


kalmaya neden olabilmektedir. Bu nedenle sezgisel yöntemlerin tarama özelliklerini daha da geliştiren ve onu sorun odaklı yöntem olmaktan ziyade genel bir yönteme dönüştüren yaklaşımlara yönelinmiştir. Üst-sezgisel yöntemler olarak adlandırılan söz konusu yöntemler, bazen temel sezgisel yöntemleri kullanmakla beraber genellikle kendilerine has yöntemlere başvururlar. Bir anlamda daha üst seviye işlem adımlarını içeren üst-sezgisel yöntemlerin, yerel çözüme saplanıp kalmadan genel çözüme ulaşmayı sağlayan, bununla beraber yerel çözümleri de gayet etkin bir şekilde çözümde kullanan yaklaşımları söz konusudur. Öte yandan üst-sezgisel yöntemler üzerinde yapılan çalışmalar ve elde edilen tecrübeler sezgisel yöntemler kadar olmasa da bazı üst-sezgisel yöntemlerin de bazı sorun türlerinde oldukça başarılı, bazı sorunlarda ise düşük verimli olduklarını göstermektedir. Öyle ki aynı sorunun farklı çözüm aşamalarında farklı üst-sezgisel yöntemler değişik performanslar sergileyebilmektedir. Örneğin gezici satıcı sorununun çözümünün başlangıç aşamalarında yaygın bir üst-sezgisel yöntem olan genetik algoritma, elde edilen çözümler üzerinden diğer bir üst-sezgisel yöntem olan benzetimli tavlama ve son aşamada ise bambaşka bir üst-sezgisel yöntem olan karınca kolonisi algoritmasının kullanılmasının en uygun sonucu verebildiği gözlemlenmiştir. Dolayısıyla belirli bir sorunun çözüm aşamalarında hangi üst-sezgisel yöntemlerin hangi sırayla dikkate alınabileceği hususu öte-sezgisel yöntemlerin geliştirilmesine vesile olmuştur<sup>(7)</sup>. Öte-sezgisel yöntemler, sorunun doğasına ve çözüm aşamasına göre daha basit nitelikteki sezgisel veya üst-sezgisel yöntemlerin seçilmesini ve çözüm süreci ile bütünleşmesini sağlayan, bir anlamda sezgisel yöntemlerin sezgiselidir. Üst sezgisel yöntemlerin, çözüm yeteneklerine göre kapsayabildiği çözüm uzayını taradığı, öte-sezgisel yöntemlerin ise yöntemle bağlı çözüm uzayları arasında çözümü aradığı söylenebilir. Üst-sezgisel yöntemlerin geçmiş 1990'lı yıllara dayanmakta olup günümüzde de oldukça aktif bir araştırma alanı niteliğindedir<sup>(8-9)</sup>.

Sezgisel yöntemlerin genelinde rastsallık önemli bir etmendir. Öyle ki aynı sorunun her seferinde aynı yöntemle çözümünde farklı varış noktalarına ulaşmak mümkün olabilmektedir. Böyle bir çeşitliliğin nedeni, sezgisel yöntemin her aşamasında rastsal sayıların kullanımınıdır. Ancak tüm sezgisel yöntemlerin rassal doğasının olduğunu söylemek hatalı olabilir. Zira bazı sezgisel yöntemler doğası gereği deterministiktir. Üst-sezgisel yöntemlerin sınıflandırılması sadedinde değişik ölçütler dikkate alınabilmekte ve bu sayede farklı tasniflere gidilebilmektedir. Söz konusu tasnif ölçütlerinden birisi de yöntemin belirlemci karakteridir. Örneğin Nelder-Mead veya belli bir bakış açısı ile tabu tarama yöntemleri deterministik yöntemler olup tarama süreçlerinde rassal sayı kullanımı ilkesel olarak söz konusu değildir. Buna mukabil yaygın üst-sezgisel yöntemlerden genetik algoritmalar (GA), parçacık sürü algoritmaları (PSO), benzetimli tavlama (SA), karınca



kolonisi (ACO) ve yapay arı sürüsü (ABC)<sup>i</sup> gibi algoritmaların tamamı rassal sayılara bağlı olarak tarama yapan algoritmalardır.



**Şekil 3.1** Sezgisel ve üst-sezgisel çözüm yaklaşımları: sürekli çizgi ile gösterilen sezgisel yaklaşımda  $A$  noktasından  $B$  noktasına ulaşmak için en uygun çözüm 1 ve akabinde 2 numaralı konumlara hareket etmektir. Ancak bu durumda engeli aşmak mümkün olmayıp yerel çözümde saplanıp kalma söz konusudur. Üst-sezgisel yöntemler daha genel tarama süreçlerine sahip olup bir sonraki adımda 3 ve 4 numaralı konumlara hareketi öngörebilirler, (kesikli çizgi ile gösterilen çözüm).

Gerek çözüm arayışının başlangıcında ve gerekse de sonraki adımlarda elde edilen çözümlerin tekil veya çoğul olmasına göre de sınıflandırmaya gidilebilmektedir. Eniyileme algoritmasının her seferinde tek bir çözüm üretmesi ve bu çözüm üzerinde iyileştirilmelere gidilmesi durumunda yöntem, tekil çözüm bazlı üst sezgisel algoritma olarak adlandırılır. Buna mukabil ardışık işlemler esnasında bir grup çözüm kümesi elde ediliyor ve bu küme üyeleri her seferinde yenilenerek en uygun çözüme ulaşılması hedefleniyorsa bu takdirde yöntem, nüfus bazlı<sup>ii</sup> üst sezgisel algoritma olarak isimlendirilir. Tekil çözüm bazlı yöntemler arasında rastlantısal tarama, benzetimli tavlama, tabu tarama ve Nelder-Mead yöntemi sayılabilir. Nüfus temelli üst-sezgisel yöntemler arasında GA, PSO, ACO ve ABC yaygın olarak bilinen yöntemlerdir.

Sınıflandırma ile ilgili literatürde ilave birkaç ölçüte daha rastlanabilmektedir. Örneğin yöntemin esin kaynağının doğa olması<sup>iii</sup> veya algoritmanın doğrudan matematiksel temellere dayandırılarak

<sup>i</sup> Kısaltmalarda İngilizce kelimelerin baş harfleri dikkate alınmıştır.

<sup>ii</sup> Population based meta-heuristic

<sup>iii</sup> Nature-inspired meta-heuristics



türetilmesi, tarama sürecinin yerel veya genel çözüm uzayını kapsamaması gibi özellikler ilave tasnif nedenleri arasında sayılabilir. Şunu da ifade edelim ki üst-sezgisel yöntemlerin büyük bir çoğunluğu doğadaki fiziksel, biyolojik veya kimyasal süreçlerden esinlenilerek geliştirilmektedir. Aynı bir yöntemler grubu olarak dikkate alınmasa da bazı hallerde yeni bir tür çözüm algoritması olarak sunulan ve esasen birkaç üst-sezgisel yöntemin veya üst-sezgisel yöntemle bir diğer yapay zekâ algoritmasının ve hatta gradyan esaslı yöntemin beraber kullanılabilirdiği yöntemlere de rastlanılabilmektedir. Söz konusu yöntemler grubu melez-sezgisel yöntemler olup oldukça etkin çözümler sunabilmektedirler<sup>(10)</sup>.

Sezgisel yöntemlerin kuramsal alt yapıları deterministik yöntemlere göre genellikle zayıftır. Uygulanan sezgisel yöntemin neden başarılı olduğu ya da gidilen yolun sonunda gerçekten de en uygun noktaya varılıp varılamayacağı sorularını cevaplamak bazen mümkün olsa da çoğunlukla zor ya da üzerinde durulmayan bir konudur. Bununla beraber literatürdeki sezgisel yöntemlerin kuramsal altyapılarının geliştirilmesi adına pek çok çalışma sürdürülmektedir. Örneğin tekil ama rassal çözüm bazlı yöntemlerden biri olan benzetimli tavlama yönteminin yakınsama özelliği Markov zinciri yaklaşımı ile modellenilebilmektedir<sup>(11)</sup>. Benzer yaklaşımlar, GA ve PSO için de geçerlidir<sup>(12-13)</sup>. Elde edilen kuramsal sonuçlara göre söz konusu üst-sezgisel yöntemlerin en uygun çözüme ulaşımı kesin olmayıp, genel çözüme belli bir olasılıkla erişim söz konusudur. Tabiatıyla bu sonuç, sezgisel yöntemlerin genel çözüme erişemeyeceği anlamına da gelmemektedir. Büyük bir olasılıkla üst-sezgisel yöntemler genel çözüme oldukça yakın çözümler sunabilmekte ve uygulamalarda sıkça kullanılabilirler. Diğer taraftan aynı mülhaza deterministik yöntemler için geçerli değildir. Dolayısıyla bazı sezgisel yöntemlerin yakınsama özellikleri kesindir.

### 3.2 “Nelder-Mead” yöntemi

Doğrudan tarama bazlı yöntemlerin birisi olan “Nelder-Mead” veya diğer adı ile “simplex” yöntemi 1962 tarihli, “Optimizasyon ve evrimsel aramalarda “simplex” tasarımlarının ardı ardına kullanımı” başlıklı makaleye dayanır<sup>(14)</sup>. Yöntemin daha sonra yaygınlaşmasına ve geliştirilmesine vesile olan çalışma ise 1965 tarihli, “Fonksiyon minimizasyonu için “simplex” yöntemi” başlıklı makaledir<sup>(15)</sup>. “Nelder-Mead” yöntemi, esas itibarıyla basit bir yaklaşımla temellendirilmiş olup “simplex” geometrisi çerçevesinde tanımlanan olası noktalardaki amaç fonksiyonu değerine göre yeni noktaların tanımlanması ve bu sayede optimal noktanın taranması

<sup>i</sup> “Sequential application of simplex designs in optimization and evolutionary operations”

<sup>ii</sup> “A simplex method for function minimization”



amaçlanmaktadır. Yöntem, bu haliyle sezgisel bir yöntem olup deneme-yanılma özelliği göstermesi nedeniyle kuramsal altyapısı nispeten zayıftır<sup>(16)</sup>.

Daha önceki bölümlerde “simplex” geometrisi üzerinde kısa da olsa durulmuş ve bazı tanımlamalar yapılmıştı. Hatırlanacağı üzere  $n$  boyutlu bir ortamda  $n + 1$  nokta ile tanımlanan geometriye “ $n$ -simplex” denmekteydi. Tanımlanan bu geometrinin özelliği, kapalı bir yüzeyin eldesi için gerekli asgari nokta sayısını belirtmesiydi. Optimal nokta taramasında yüzey tanımlama önemli bir husus olup amaç fonksiyonunun azalma veya artma trendi içinde olup olmadığı “simplex” geometrisi sayesinde asgari nokta sayısı ile öngörülebilmektedir. “Nelder-Mead” yönteminin uygulanabilmesi için her adımda  $n + 1$  noktanın belirlenmesi gerekir. Dolayısıyla bir başlangıç noktası yerine  $n + 1$  nokta ile tarama işlemine başlanmalıdır. Bununla beraber  $x_0$  gibi tek bir başlangıç noktasının var olması halinde müteakip denklemlerin kullanılması ile birbirine eşit mesafedeki diğer simplex noktalarının hesaplanması mümkündür;

$$x_i = x_0 + p e_i + \sum_{\substack{j=1 \\ j \neq i}}^n q e_j \quad i = 1, 2, \dots, n \quad (3.1)$$

$$p = \frac{a}{n\sqrt{2}} (\sqrt{n+1} + n - 1) \quad (3.2)$$

$$q = \frac{a}{n\sqrt{2}} (\sqrt{n+1} - 1) \quad (3.3)$$

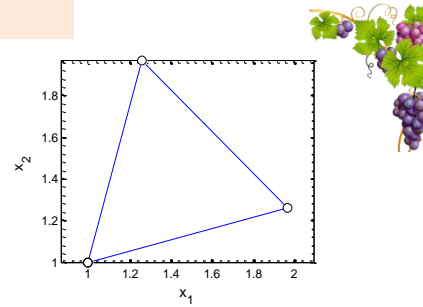
Burada  $e_i$  her bir eksendeki birim yöneyi ve  $a$  noktalar arası mesafeyi ifade etmektedir. Diğer taraftan başlangıç noktası dikkate alınarak daha basit yaklaşımlarla da simplex noktaları oluşturulabilir. Örneğin başlangıç noktası köşe nokta olmak üzere her bir ana eksen dâhilinde  $a$  uzaklığa sahip nokta tespiti de yapılabilir.

### 2B ortamında “Simplex” üretimi

MATLAB uygulaması

```

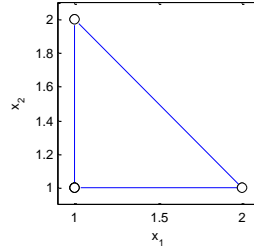
clc; clear; n0 = [1;1];
n=size(n0,1); a=1;
p = a/n/sqrt(2) * (sqrt(n+1)+n-1);
q = a/n/sqrt(2) * (sqrt(n+1)-1);
n1 = n0 + p * [1;0] + q * [0;1];
n2 = n0 + p * [0;1] + q * [1;0];
data = [n0';n1';n2';n0'];
plot(data(:,1),data(:,2));
axis equal; hold on;
plot(data(:,1),data(:,2),'ko');
    
```





```

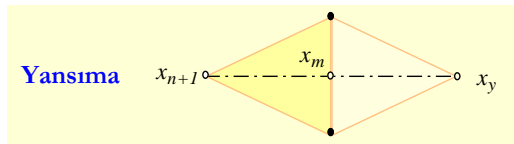
clc; clear; n = 2; x(1,:) = ones(1,n);
                    a = 1; e = eye(2);
for i = 2:n+1
x(i,:) = x(1,:) + a * e(:,i-1);
end
                    data = [x;x(1,:)];
                    plot(data(:,1),data(:,2),'ko')
                    hold on;
                    xlim([0.9 2.1]); ylim([0.9 2.1])
                    plot(data(:,1),data(:,2)); axis square
    
```



“Nelder-Mead” yönteminin başlangıçtaki “simplex” noktalarının tespiti sonrasında takip edilecek işlem adımları aşağıdaki gibi sıralanabilir;

- ▶ 1. Her bir simplex noktasının amaç fonksiyon değeri hesaplanarak minimizasyon amacı çerçevesinde asgariden azamiye olacak şekilde sıralama yapılır  $\{f(\mathbf{x}_i) < f(\mathbf{x}_j)\}$  ve asgari amaç fonksiyon değerine sahip nokta  $\mathbf{x}_L$ , azami amaç fonksiyon değerine sahip nokta  $\mathbf{x}_H$  olarak kaydedilir.
- ▶ 2. Fonksiyon değeri açısından en kötü durumdaki  $\mathbf{x}_H$  noktası hariç diğer noktalar dikkate alınarak bu noktaların merkezi niteliğindeki orta nokta elde edilir;

$$\mathbf{x}_m = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i \quad (3.4)$$



- ▶ 3. Elde edilen  $\mathbf{x}_m$  orta noktası dikkate alınarak en kötü nokta konumundaki  $\mathbf{x}_H$  noktasının belli bir uzaklıktaki yansıma noktası tespit edilir;

$$\mathbf{x}_y = \mathbf{x}_m + \alpha(\mathbf{x}_m - \mathbf{x}_H) \quad (3.5)$$

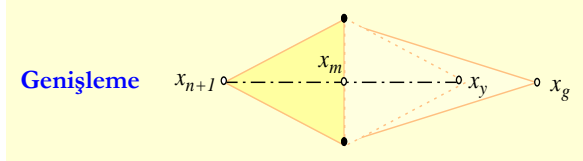
$\mathbf{x}_y$  yansıma noktasının orta noktadan olan uzaklığını yansıma katsayısı niteliğindeki  $\alpha$  değeri belirler. Genellikle  $\alpha = 1$  olarak alınır. Yansıma işlemi neticesinde elde edilen noktanın amaç







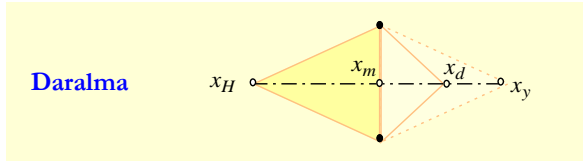
fonksiyon değeri, en iyi konumdaki nokta ve en kötü konumdaki ikinci nokta amaç fonksiyon değerleri ile karşılaştırılır. Eğer  $f(\mathbf{x}_L) < f(\mathbf{x}_y) < f(\mathbf{x}_k)$  sıralaması söz konusu ise yeni bir “simplex” geometrisi teşkil edilir. Bu amaçla en kötü nokta, yansıma noktası ile yer değiştirilir:  $\mathbf{x}_H = \mathbf{x}_y$  ve 1inci adıma tekrar gidilir.



- 4. Yansıma noktasının amaç fonksiyon değeri mevcut noktalara göre daha iyi bir sonuç veriyorsa,  $f(\mathbf{x}_y) < f(\mathbf{x}_L)$ ; bu takdirde yansıma noktasının bulunduğu hatta azalma trendi olduğu düşünülebilir. Dolayısıyla bu hat dâhilinde yeni bir genişleme noktası hesaplanır;

$$\mathbf{x}_g = \mathbf{x}_y + \beta(\mathbf{x}_y - \mathbf{x}_m) \quad (3.6)$$

$\mathbf{x}_g$  genişleme noktasının büyüklüğünü, genişleme katsayısı niteliğindeki  $\beta$  değeri belirler ve genellikle değeri 2 olarak dikkate alınır. Genişleme noktasının, amaç fonksiyon değerinin yansıma noktasına göre daha iyi bir sonuç vermesi halinde  $\{f(\mathbf{x}_g) < f(\mathbf{x}_y)\}$ , en kötü nokta olan  $\mathbf{x}_H$  noktası yeni elde edilen  $\mathbf{x}_g$  noktası ile yer değiştirilir ve bu sayede yeni “simplex” nokta grubu teşkil edilerek başlangıçtaki ilk adıma tekrar gidilir. Aksi halde bir sonraki adıma geçilir.



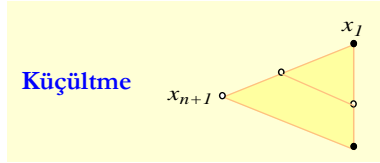
- 5. Yansıma noktasının amaç fonksiyon değerinin, en kötü ikinci nokta niteliğindeki  $\mathbf{x}_k$  noktasının amaç fonksiyon değerine göre daha kötü olması halinde  $\{f(\mathbf{x}_y) \geq f(\mathbf{x}_k)\}$ , daralma işlemine gidilir ve yeni bir daralma noktası tespit edilir;

$$\mathbf{x}_d = \mathbf{x}_m + \gamma(\mathbf{x}_H - \mathbf{x}_m) \quad (3.7)$$





$x_d$  daralma noktasının büyüklüğünü, daralma katsayısı niteliğindeki  $\gamma$  değeri belirler. Daralma katsayısı değeri genellikle 0.5 olarak alınır. Daralma sonrası elde edilen noktanın en kötü nokta olan  $x_H$  noktasından daha iyi bir amaç fonksiyon değeri vermesi halinde  $\{f(x_d) < f(x_H)\}$ , en kötü nokta olan  $x_H$  noktası, yeni elde edilen  $x_d$  noktası ile yer değiştirilir ve bu sayede yeni “simplex” nokta grubu teşkil edilerek başlangıçtaki ilk adıma tekrar gidilir. Aksi halde bir sonraki adıma geçilir.



- 6. Bu adımda başlangıçtaki “simplex” noktaları yerine yeni bir başlangıç yapılır. Bu işlem yapılırken en iyi nokta konumundaki  $x_L$  noktası muhafaza edilir, ancak diğer noktalar;

$$x_i = x_L + \delta(x_i - x_L), i = 2, 3, \dots, n + 1 \quad (3.8)$$

ifadesine göre küçülmeye tabi tutularak yeni “simplex” geometrisi oluşturulur. Denklem 3.8’de geçen  $\delta$  ifadesi, küçültme katsayısı olup genellikle 0.5 değeri dikkate alınır. Yeni bir başlangıç ile beraber ilk adıma tekrar gidilir<sup>(17)</sup>.

İşlem adımları arasındaki ilk geometrik değişikliği içeren yansıma işleminin temel dayanağı, azami noktanın tersi yönündeki istikametın azalma eğiliminde olacağı beklentisidir. Bu beklentinin karşılanması halinde, genişleme işlemi ile aynı doğrultu biraz daha büyük bir adımla test edilmiş olur. İstenen değişimin sağlanamaması durumunda daraltma yapılarak gidilen doğrultu üzerinden daha kısa adımlarla tarama tercih edilir. Küçültme ise aynı bölgede daha küçük bir yerel tarama ile yeni bir başlangıcın kapısını aralar. Tarama dâhilinde yapılan tekrarlı işlemlerin sonlandırılması için dikkate alınan ölçüt, genellikle “simplex” noktalarındaki amaç fonksiyonu değerinin birbirlerine yakın olup olmadığını gösteren standart sapma değeridir;

$$\left\{ \sum_{i=1}^{n+1} \frac{1}{n+1} [f(x_i) - f(x_m)]^2 \right\}^{1/2} \leq \epsilon \quad (3.9)$$

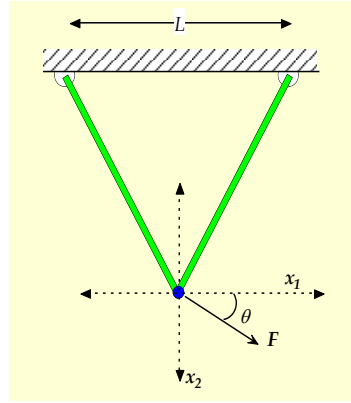
Burada  $\epsilon$  değeri standart sapma için dikkate alınan hassasiyet değeridir. Yakınsamada dikkate alınabilecek diğer bir ölçüt de amaç fonksiyonu varyans değeri olup;



$$\sum_{i=1}^{n+1} [f(x_i) - \bar{f}]^2 \leq \epsilon \quad (3.10)$$

$$\bar{f} = \frac{1}{n+1} \sum_{i=1}^{n+1} f(x_i) \quad (3.11)$$

denklemleri dikkate alınarak hesaplanabilir. “Nelder-Mead” yönteminin daha iyi anlaşılabilmesi için daha önce de dikkate alınan ve Şekil 3.2’deki gibi bir sorunu belirtilen işlem adımlarını takip ederek çözebiliriz;



Şekil 3.2 İki-kiriş düzeneği ve uç noktada oluşan yer değiştirme mesafelerinin tespiti.

Buna göre kenarları ve mafsıl aralığı eşit mesafede olan iki-kirişli bir düzeneği dikkate alalım. Böyle bir yapının potansiyel enerji denklemini hatırlanacağı üzere;

$$f(x) = \frac{EA}{4L} x_1^2 + \frac{3EA}{4L} x_2^2 - Fx_1 \cos \theta - Fx_2 \sin \theta$$

şeklinde yazabiliriz<sup>(18)</sup>. Burada  $E$  elastisite modülü,  $A$  kirişe ait kesit alan değeri,  $L$  mesafe değeri,  $F$  uç noktaya  $\theta$  açısı ile etki eden kuvvet,  $x_1$  ve  $x_2$  ise uç noktanın yer değiştirme miktarlarıydı. Uygulanan kuvvet nedeniyle kirişlerin birleşim noktasında meydana gelen yer değiştirme değerlerini, potansiyel enerji denkleminin minimizasyonu yoluyla elde etmek mümkündür. Denklemdaki katsayı değerlerini;  $E = 200 \times 10^9 Pa$ ,  $A = 10^{-5} m^2$ ,  $L = 5 m$ ,  $F = 10000 N$  ve  $\theta = 30^\circ$  olarak dikkate aldığımızda aynı denklemi;



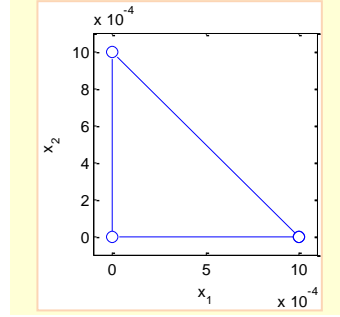


$$f(\mathbf{x}) = 10^5 x_1^2 + 3 \times 10^5 x_2^2 - 8660 x_1 - 5000 x_2 \text{ Nm}$$

şeklinde ifade edebiliriz. Çözüm dâhilinde başlangıç noktası olarak  $\mathbf{x}_1^T = [0 \ 0]$ , birbirine eşit mesafedeki diğer “simplex” noktalarının hesaplanması için kullanılacak mesafe değerini  $a = 10^{-3}m$  ve yakınsama ölçütü olarak da  $\epsilon = 0.1$  değerini dikkate alalım. Buna göre her bir ekseninde eşit mesafe yaklaşımı dikkate alınarak diğer noktalar;

$$\mathbf{x}_2 = \begin{bmatrix} 1 \times 10^{-3} \\ 0 \\ 0 \end{bmatrix}$$

$$\mathbf{x}_3 = \begin{bmatrix} 0 \\ 1 \times 10^{-3} \\ 0 \end{bmatrix}$$



olarak belirlenebilir. Eldeki üç noktaya ait amaç fonksiyon değerleri hesaplandığında;

$$f(\mathbf{x}_1) = 0$$

$$f(\mathbf{x}_2) = -8.56$$

$$f(\mathbf{x}_3) = -4.7$$

değerleri elde edilir. Bu değerler dikkate alınarak küçükten büyüğe doğru bir sıralama ve ona uygun alt indis düzenlemesi yapılırsa;

$$f(\mathbf{x}_2) < f(\mathbf{x}_3) < f(\mathbf{x}_1)$$

$$\mathbf{x}_L = \begin{bmatrix} 1 \times 10^{-3} \\ 0 \\ 0 \end{bmatrix}$$

$$\mathbf{x}_k = \begin{bmatrix} 0 \\ 1 \times 10^{-3} \\ 0 \end{bmatrix}$$

$$\mathbf{x}_H = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

yazılabilir. Fonksiyon değeri açısından en kötü durumdaki  $\mathbf{x}_H$  noktası hariç diğer noktalar dikkate alınarak, bu noktaların merkezi niteliğindeki orta nokta elde edilir;

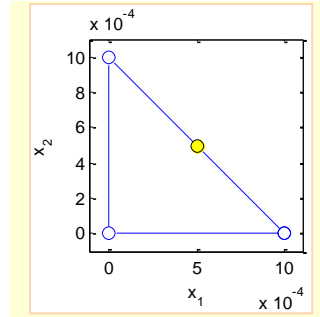




$$\mathbf{x}_m = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i$$

$$\mathbf{x}_m = \frac{1}{2} \left\{ \begin{bmatrix} 1 \times 10^{-3} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \times 10^{-3} \end{bmatrix} \right\}$$

$$\mathbf{x}_m = \begin{bmatrix} 0.5 \times 10^{-3} \\ 0.5 \times 10^{-3} \end{bmatrix}$$



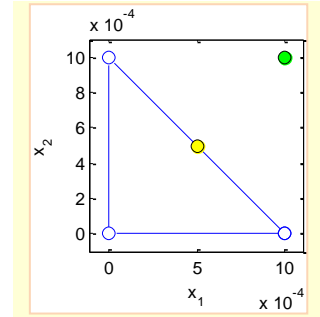
Elde edilen  $\mathbf{x}_m$  orta noktası dikkate alınarak en kötü nokta konumundaki  $\mathbf{x}_H$  noktasının belli bir uzaklıktaki yansıma noktası ve bu noktanın amaç fonksiyon değeri tespit edilir;

$$\mathbf{x}_y = \mathbf{x}_m + \alpha(\mathbf{x}_m - \mathbf{x}_H)$$

$$\mathbf{x}_y = \begin{bmatrix} 0.5 \times 10^{-3} \\ 0.5 \times 10^{-3} \end{bmatrix} + 1 \times \left[ \begin{bmatrix} 0.5 \times 10^{-3} \\ 0.5 \times 10^{-3} \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right]$$

$$\mathbf{x}_y = \begin{bmatrix} 0.001 \\ 0.001 \end{bmatrix}$$

$$f(\mathbf{x}_y) = -13.26$$



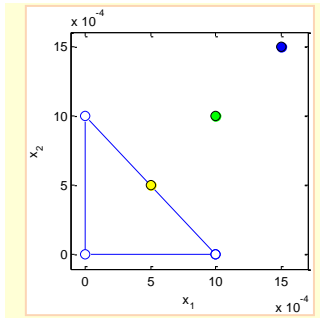
Yansıma noktasına ait amaç fonksiyon değerinin mevcut noktalara göre daha iyi bir sonuç vermesi nedeniyle,  $f(\mathbf{x}_y) = -13.26 < f(\mathbf{x}_L) = -8.56$ ; aynı hat dâhilinde yeni bir genişleme noktası ve amaç fonksiyon değeri hesaplanır;

$$\mathbf{x}_g = \mathbf{x}_y + \beta(\mathbf{x}_y - \mathbf{x}_m)$$

$$\mathbf{x}_g = \begin{bmatrix} 0.001 \\ 0.001 \end{bmatrix} + 2 \times \left[ \begin{bmatrix} 0.001 \\ 0.001 \end{bmatrix} - \begin{bmatrix} 0.5 \times 10^{-3} \\ 0.5 \times 10^{-3} \end{bmatrix} \right]$$

$$\mathbf{x}_g = \begin{bmatrix} 0.0015 \\ 0.0015 \end{bmatrix}$$

$$f(\mathbf{x}_g) = -19.59$$



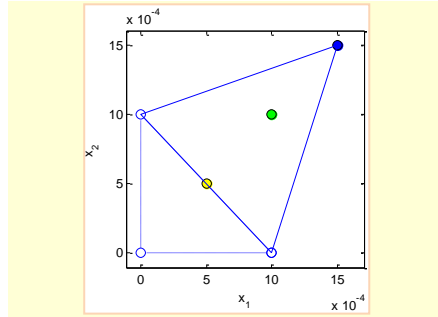


Genişleme noktasının amaç fonksiyon değerinin, yansıma noktasına göre daha iyi bir sonuç vermesi nedeniyle,  $f(\mathbf{x}_g) = -19.59 < f(\mathbf{x}_y) = -13.26$ ; en kötü nokta olan  $\mathbf{x}_H$  noktası yeni elde edilen  $\mathbf{x}_g$  noktası ile değiştirilir ve bu sayede yeni “simplex” nokta grubu teşkil edilir;

$$\mathbf{x}_1 = \begin{bmatrix} 0.0015 \\ 0.0015 \end{bmatrix} \quad f(\mathbf{x}_1) = -19.59$$

$$\mathbf{x}_2 = \begin{bmatrix} 1 \times 10^{-3} \\ 0 \end{bmatrix} \quad f(\mathbf{x}_2) = -8.56$$

$$\mathbf{x}_3 = \begin{bmatrix} 0 \\ 1 \times 10^{-3} \end{bmatrix} \quad f(\mathbf{x}_3) = -4.7$$



Eldeki üç noktaya ait amaç fonksiyon değerleri tekrar dikkate alınıp yeni sıralama ve ona uygun alt indis düzenlemesi yapılırsa;

$$f(\mathbf{x}_1) < f(\mathbf{x}_2) < f(\mathbf{x}_3)$$

$$\begin{aligned} \mathbf{x}_L &= \begin{bmatrix} 0.0015 \\ 0.0015 \end{bmatrix} \\ \mathbf{x}_k &= \begin{bmatrix} 1 \times 10^{-3} \\ 0 \end{bmatrix} \\ \mathbf{x}_H &= \begin{bmatrix} 0 \\ 1 \times 10^{-3} \end{bmatrix} \end{aligned}$$

yazılabilir. Müteakiben eldeki “simplex” noktalarının amaç fonksiyon değerlerinin birbirlerine yakın olup olmadığını gösteren varyans değeri belirlenir;

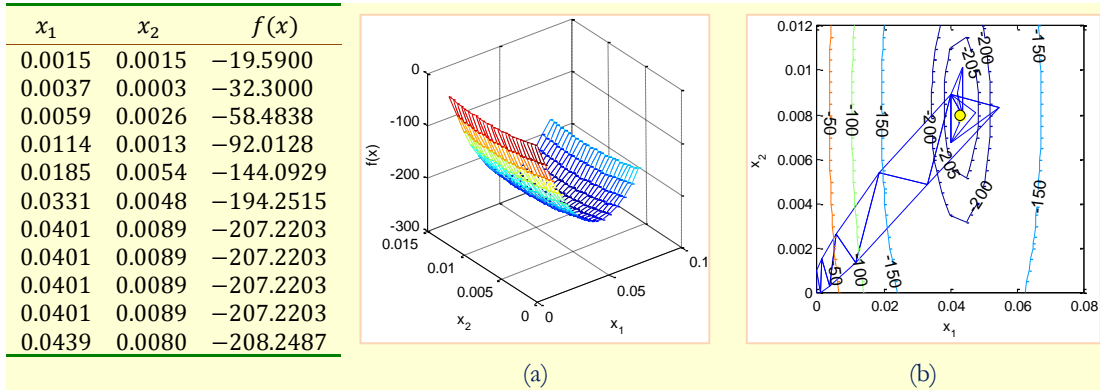
$$\bar{f} = \frac{1}{n+1} \sum_{i=1}^{n+1} f(\mathbf{x}_i) = \frac{1}{3}(-19.59 - 8.56 - 4.7) = -10.95$$

$$\sum_{i=1}^{n+1} [f(\mathbf{x}_i) - \bar{f}]^2 = 119.42 > \epsilon = 0.1$$

Görüldüğü üzere yakınsama ölçütü mevcut noktalar ile karşılanmamakta olup daha önce ifade edilen işlem adımları takip edilerek yakınsama ölçütüne göre tekrarlı işlemlere devam edilir.



Şekil 3.3'teki çizelgede müteakip işlem adımlarında elde edilen  $\mathbf{x}_L$  ve bu noktalara ait amaç fonksiyon değerleri görülmektedir. Ayrıca tarama esnasında dikkate alınan “simplex” geometrileri ile ulaşılan optimal nokta da Şekil 3.3(b) üzerinde görülmektedir.



Şekil 3.3 İki-kiriş düzeneğine ait (a) amaç fonksiyonu yüzeyi ve (b) eniyileme süreci.

Buna göre söz konusu iki-kiriş düzeneğinin uç noktasında meydana gelen yer değiştirme değerleri  $\mathbf{x} = [0.0439 \ 0.008] m$  olup minimum potansiyel enerji değeri  $208.2487 Nm$  olarak elde edilmiştir. Bulunan değerler, birinci mertebeden türev alınarak elde edilebilecek analitik sonuçlarla örtüşmektedir.

Genel olarak “Nelder-Mead” yönteminin, boyut sayısının artması durumunda verimliliği azalmakta, hatta bazı problemlerde optimal noktaya yakınsama gerçekleşmeyebilmektedir<sup>(19)</sup>. Bununla beraber herhangi bir gradyan bilgisine veya tek değişkenli tarama sürecine ihtiyacı olmaması nedeniyle değişken sayısının az olduğu sorunlarda etkin bir yöntemdir.

### 3.3 Tabu algoritması

Tabu kelimesi, Polinezya dil ailesine ait bir kelime olup kimi ilkel kabilelerde kutsal kabul edilen, korkuyla karışık saygı duyulan, aksi yapıldığında zararı dokunulacağı düşünülen nesne veya davranış anlamına gelmektedir<sup>(20)</sup>. Söz konusu kavramın eniyileme sorunlarının çözümünde kullanması yönünde atılan ilk adımlar F. W. Glover’e (1937-) aittir. 1977 yılında yayımladığı

<sup>i</sup> “Tabu” veya “taboo”



“Tamsayı programlamada vekil kısıtların kullanılması ile sezgisel yöntemler” adlı makalesi ile Glover’in, tabu tarama algoritmasına ait pek çok temel yaklaşımın temellerini ifade ettiği söylenebilir<sup>(21)</sup>. Bu çalışmanın getirdiği en ciddi katkı, sözde kısıt manasında tekrarlı işlemlerden kaçınmayı sağlayan kısa vadeli hafızanın kullanımı ve bu sayede çözüm arayışında yerel çözümlere takılmaktan kurtulma fırsatı veren mekanizmaların geliştirilmesidir. Yaklaşımların yaygınlaşarak genel bir çözüm algoritması olarak kabul görmesi ise yine Glover’in ve aynı zaman diliminde yayımlanan P. Hansen’in 1986 tarihli yayınları sayesinde<sup>(22)</sup>.

Tabu tarama algoritmasının kullanımı, genellikle sıralama sorunlarının çözümünde yaygındır<sup>(23)</sup>. Bununla beraber gerçek sayıların söz konusu olduğu sorunlarda da kendisinden yararlanılabilir. Karakteri itibari ile yöntem tekrarlı bir yöntem olup istenilen seviyedeki çözüme ulaşıncaya kadar aynı işlem adımları ardışık olarak yinelenir. En uygun nokta taramasında başvuru temel yaklaşım deneme-yanılma ile yerel çözümün aranmasıdır. Ancak yerel çözümün aranması esnasında daha önceden uğranılan noktaların tekrar ziyaret edilmemesi adına tabu taraması bazı önemli araçları kullanır. Bunlardan ilki, daha önceki tarama noktalarının hafızada tutularak aynı noktanın tekrar ziyaret edilmesinden kaçınılması, diğeri ise eldeki çözümün, kötü olması halinde bile kabul görmesi ve etrafında yerel taramaya devam edilmesidir.

Hafıza kavramı, tabu tarama yönteminde önemli bir yere sahip olup genellikle iki tür hafızadan bahsedilebilir; kısa dönem hafıza ve sıklığa dayalı hafıza. İkinci tür hafıza bazen orta ve uzun dönem hafıza olarak da bilinir. Kısa dönem hafıza, en temel hafıza türü olup yakın geçmişte yapılan nokta ziyaretlerinin bir süre için tabu olarak kayıt altına alınması ve tekrarından kaçınılması amaçlanmaktadır. Kayıt altına alınanların oluşturduğu listeye tabu listesi adı verilir. Orta ve uzun dönem hafızanın oluşturulmasında başvuru sıklık ölçütü her bir hareketin toplam tekrar sayısına ya da en yüksek veya ortalama tekrar sayılarına bakılarak düzenlenebilir<sup>(24)</sup>.

Tabu listesinin başlangıçtan itibaren kaydedilen tüm hareketleri hafızada tutması genellikle tercih edilmez. Zira böyle bir uygulama hafızada çok yer kaplayabilir veya tarama seçeneklerinin çok daralmasına ve yerel çözümde saplanıp kalmaya sebebiyet verebilir. Dolayısıyla sabit uzunlukta ve sürekli güncellenen bir tabu listesinin dikkate alınması yaygındır. Liste uzunluğu bazı uygulamalarda değişkenlik göstermekte olup rastsal olarak uzatılıp kısaltılabilir. Tarama esnasında aynı anda birden fazla tabu listesinin oluşturulması ve kullanılması da söz konusu olabilir. Diğer taraftan tabu listesindeki bir çözümün taramada tekrar dikkate alınması adına tabu yıkma<sup>ii</sup>

<sup>i</sup> “Heuristics for integer programming using surrogate constraints”

<sup>ii</sup> Aspiration criteria

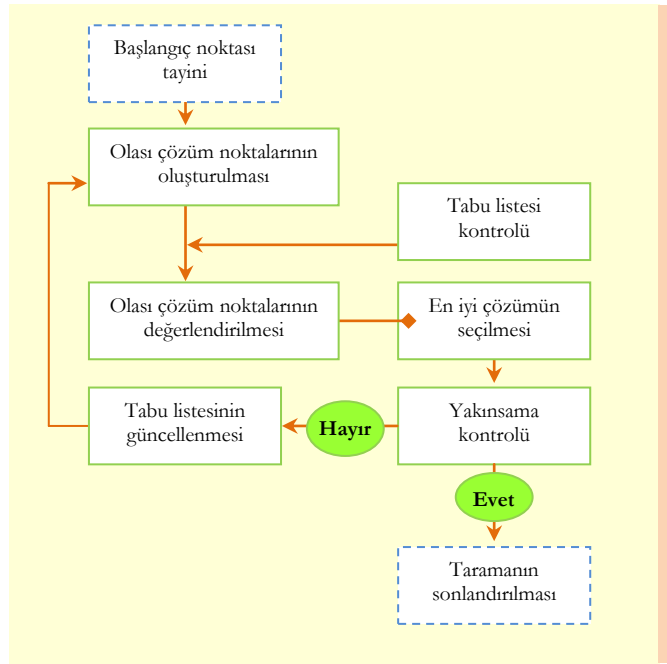






ölçütleri de geliştirilebilir. En yaygın tabu yıkma ölçütü, içinde bulunulan adımdaki çözüme göre daha iyi bir sonuç veren tabu hareketinin yapılmasına müsaade edilmesidir. Diğer bir basit yöntem ise tabu listesinin uzunluğuna göre en eski tabunun listeden çıkarılması ve yerine yeni tabuların eklenmesidir. Bu haliyle tabu yıkma işlemi zamana bağlı hale gelmiş olur.

Tabu taramada üzerinde bulunulan  $\mathbf{x}$  noktasından  $N(\mathbf{x}) \subset \mathbf{X}$  yakın komşuluğu çerçevesinde başka bir  $\mathbf{x}' \in N(\mathbf{x})$  noktasına yerel tarama yapılarak ulaşmak mümkündür. Ancak yakın komşuluğun daraltılması söz konusu olup  $N^*(\mathbf{x})$  yerel çözüm uzayının belirlenmesinde tabu listesi etkilidir<sup>(25)</sup>. Tabu tarama yönteminin genel akış şeması aşağıdaki şekilde görülmektedir;



**Akış Diyagramı 3.1** Tabu tarama akış diyagramı.

Buna göre sorunun çözümünde öncelikle başlangıç noktasının tespit edilmesi gerekir. Seçilen başlangıç noktasının olası çözüm uzayı içinde olması şart değildir. Noktanın tespiti sonrasında olası çözüm noktaları teşkil edilir. Söz konusu teşkilde genellikle yerel tarama ön plandadır. Elde



edilen çözüm noktaları kısa süreli hafızaya alınarak her bir noktanın istenen çözüm özelliklerini ne kadar sağladıkları, amaç fonksiyon değerlerine bakılarak değerlendirilir. Eldeki çözüm noktaları arasından en iyi çözüm seçilerek yakınsama kontrolüne tabi tutulur. İstenen hassasiyette bir yakınsamanın söz konusu olması halinde tarama işlemi sonlandırılır. Aksi halde tabu listesi güncellenerek o işlem adımına ait eniyi çözüm üzerinden yeniden olası çözüm noktaları teşkil edilir. Bahsedilen işlem adımları, istenen yakınsama sağlanıncaya kadar tekrar edilir. Yakınsama ölçütleri arasında azami iterasyon sayısı, azami hesaplama süresi, belirli sayıdaki tekrarlı işlemler sonrasında daha iyi bir sonuca ulaşamama gibi ölçütler dikkate alınabilir.

Tabu algoritmasının daha iyi anlaşılabilmesi için aşağıdaki gibi bir sorunu belirtilen işlem adımlarını takip ederek çözebiliriz. Buna göre ticari bir işletmenin dört yeni makine tezgâhı satın aldığı, satın alınan tezgâhların yerleştirilebileceği dört konumun bulunduğunu düşünelim. Her bir makinenin öngörülen konuma aktarımı için bir gider söz konusu olup aşağıdaki çizelgede bahse konu giderler görülmektedir;

Makine/Konum	1	2	3	4
A	180	10	96	146
B	138	28	166	172
C	114	186	4	158
D	14	154	150	46

Yerleşim dâhilinde güdülen temel gaye, giderin minimize edilmesi olup bu amaç çerçevesinde uygun sıralamanın belirlenmesi gerekmektedir. İfade edilen sorunun tabu algoritması ile çözümünü sadedinde yapılacak ilk işlem, belirli bir öngörü ile veya rastsal olarak bir başlangıç çözümünün oluşturulmasıdır. Ayrıca tarama işleminin başlangıcında azami iterasyon sayısının ve tabu listesinin, çözüm noktası veya iterasyon sayısına göre uzunluğunun belirlenmesi gerekir. Üzerinde çalışılan sorun için azami tekrar sayısının 10, tabu listesi uzunluğunun ise en fazla 10 çözüm noktası olacak şekilde düzenlenmesi ölçütleri dikkate alınabilir. Tabu listesine ilave eklemeler olduğunda zamansal olarak daha önce listeye giren çözüm noktalarının listeden çıkarılması öngörülebilir. Çözüm için gerekli başlangıç noktası olarak ABCD sıralaması dikkate alınabilir. Belirtilen çözüme ait gider toplamı;

$$ABCD \rightarrow 180 + 28 + 4 + 46 = 258$$

şeklinde hesaplanabilir.



Bir sonraki adımda eldeki çözümden hareketle yerel çözümler üretilir. Bu meyanda akla gelen yerel çözümler  $A$  ile  $B$ 'nin,  $B$  ile  $C$ 'nin ve son olarak  $C$  ile  $D$ 'nin yer değiştirilerek yerleştirilmesidir;

$$\begin{aligned} BACD &\rightarrow 138 + 10 + 4 + 46 = 198 \\ ACBD &\rightarrow 180 + 186 + 166 + 46 = 428 \\ ABDC &\rightarrow 180 + 28 + 150 + 158 = 516 \end{aligned}$$

Oluşturulan yerel olası çözümler içerisinde en iyi çözümün  $BACD$  sıralaması olduğu görülmektedir. Diğer taraftan başlangıç çözümü ile beraber diğer çözümler denendiği için oluşturulan ilk tabu listesine dâhil edilir;

$$T = \{ABCD, BACD, ACBD, ABDC\}$$

İkinci iterasyonda eldeki en iyi çözümden hareketle yeni yerel çözümler oluşturulur. Örneğin  $BACD$  sıralamasında  $B$  ile  $A$ 'nın,  $A$  ile  $C$ 'nin ve son olarak  $C$  ile  $D$ 'nin yer değiştirilmesi mümkündür;

$$\begin{aligned} ABCD &\rightarrow \text{tabu listesinde} \\ BCAD &\rightarrow 138 + 186 + 96 + 46 = 466 \\ BADC &\rightarrow 138 + 10 + 150 + 158 = 456 \end{aligned}$$

Görüldüğü üzere ilk yerel çözüm daha önceden denendiği ve tabu listesinde bulunduğu için dikkate alınmamıştır. Eldeki çözümler arasında en iyi çözüm,  $BADC$  olup bu çözümden hareketle yeni yerel çözümler oluşturulur. Bu arada tabu listesi de güncellenebilir;

$$T = \{ABCD, BACD, ACBD, ABDC, BCAD, BADC\}$$

Üçüncü iterasyon dâhilinde  $BADC$  sıralamasında  $B$  ile  $A$ 'nın,  $A$  ile  $D$ 'nin ve son olarak  $D$  ile  $C$ 'nin yer değiştirilmesi mümkündür;

$$\begin{aligned} ABDC &\rightarrow \text{tabu listesinde} \\ BDAC &\rightarrow 138 + 154 + 96 + 158 = 546 \\ BACD &\rightarrow \text{tabu listesinde} \end{aligned}$$

Benzer şekilde ilk yerel çözüm ile üçüncü yerel çözüm daha önceden denendiği ve tabu listesinde bulunduğu için dikkate alınmamıştır. Eldeki çözümler arasında en iyi çözüm  $BDAC$  olup bu



çözümünden hareketle yeni yerel çözümler oluşturulur. Eldeki çözüm ise güncel tabu listesine eklenir;

$$T = \left\{ \begin{array}{l} ABCD, BACD, ACBD, ABDC \\ BCAD, BADC \\ BDAC \end{array} \right\}$$

Dördüncü iterasyon dâhilinde  $BDAC$  sıralamasında  $B$  ile  $D$ 'nin,  $D$  ile  $A$ 'nın ve son olarak  $A$  ile  $C$ 'nin yer değiştirilmesi mümkündür;

$$\begin{array}{l} DBAC \rightarrow 14 + 28 + 96 + 158 = 296 \\ BADC \rightarrow \text{tabu listesinde} \\ BDCA \rightarrow \text{tabu listesinde} \end{array}$$

Eldeki çözümler arasında en iyi çözüm,  $DBAC$  olup bu çözümünden hareketle yeni yerel çözümler oluşturulur. Eldeki çözüm ise güncel tabu listesine eklenir;

$$T = \left\{ \begin{array}{l} ABCD, BACD, ACBD, ABDC \\ BCAD, BADC \\ BDAC \\ DBAC \end{array} \right\}$$

Beşinci iterasyon dâhilinde  $DBAC$  sıralamasında  $D$  ile  $B$ 'nin,  $B$  ile  $A$ 'nın ve son olarak  $A$  ile  $C$ 'nin yer değiştirilmesi mümkündür;

$$\begin{array}{l} BDAC \rightarrow \text{tabu listesinde} \\ DABC \rightarrow 14 + 10 + 166 + 158 = 348 \\ DBCA \rightarrow 14 + 28 + 4 + 146 = 192 \end{array}$$

Eldeki çözümler arasında en iyi çözüm  $DBCA$  olup bu çözümünden hareketle yeni yerel çözümler oluşturulur. Eldeki çözüm ise güncel tabu listesine eklenir;

$$T = \left\{ \begin{array}{l} ABCD, BACD, ACBD, ABDC \\ BCAD, BADC \\ BDAC \\ DBAC \\ DABC, DBCA \end{array} \right\}$$



Altıncı iterasyon dâhilinde *BDCA* sıralamasında *D* ile *B*'nin, *B* ile *C*'nin ve son olarak *C* ile *A*'nın yer değiştirilmesi mümkündür;

$$\begin{aligned} BDCA &\rightarrow 138 + 154 + 4 + 146 = 442 \\ DCBA &\rightarrow 14 + 186 + 166 + 146 = 512 \\ DBAC &\rightarrow \text{tabu listesinde} \end{aligned}$$

Eldeki çözümler arasında en iyi çözüm *BDCA* olup bu çözümden hareketle yeni yerel çözümler oluşturulur. Ziyaret edilen çözümler ise güncel tabu listesine eklenir. Ancak bu iterasyonda azami tabu listesine erişilmiş olup listedeki ilk iki çözüm listeden çıkarılıp, yerine son iterasyondaki iki çözüm eklenir;

$$T = \left\{ \begin{array}{l} ACBD, ABDC \\ BCAD, BADC \\ BDAC \\ DBAC \\ DABC, DBCA \\ BDCA, DCBA \end{array} \right\}$$

Mütekip tekrarlı işlemler benzer şekilde yapıldığında görülecektir ki 10 iterasyon sonunda elde edilen en iyi çözüm, *BDCA* sıralaması olup bu çözümde toplam gider 192'dir<sup>(24)</sup>;

Makine/Konum	1	2	3	4
A	180	10	96	146
B	138	28	166	172
C	114	186	4	158
D	14	154	150	46

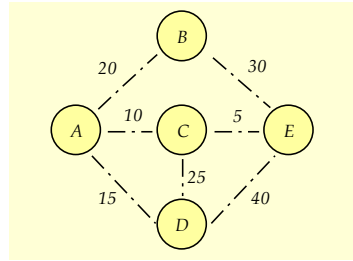
Çözüme ulaşma sürecinde toplamda 10 çözüm noktası taranmış olup beşinci iterasyonda optimal çözüme ulaşılmıştır.

Tabu algoritması, her ne kadar deterministik bir karakterde görünse de bazen çözüm noktası taramasında rassal çözümler de üretilebilir. Örneğin yukarıda çözülen sorunun yerel çözüm oluşturma aşamasında daha iyi bir noktaya varılmaması durumunda yerel çözüm yerine daha radikal bir yaklaşımla rassal bir şekilde çözüm üretilerek yerel taramaya gidilebilir. İfade edilen yaklaşımın daha iyi anlaşılması için 2nci iterasyon tekrar ele alınabilir. Hatırlanacağı üzere bu iterasyon sonunda elde edilen en iyi çözüm, *BACD* sıralaması olup bu sıralama baz alınarak üretilen yerel çözümler *ABCD, BCAD, BADC* şeklindeydi. Üretilen bu çözümlerden ilki tabu



listesinde olduğu için dikkate alınmamıştı. Diğer iki çözüm ise 466 ve 456 gider değerleri ile hâlihazırda bulunan en iyi çözüme göre (198) daha kötü çözümlerdi. Bu aşamada çözüm arayışına *BADC* ile devam etmek yerine eldeki en iyi çözüm üzerinde (*BACD*) rastgele bir değişikliğe gidilebilir. Örneğin yerel çözümler yerine rastgele olarak *B* ile *D*'nin yeri değiştirilerek yeni bir çözüm üretilebilir (*DACB*) ve bu çözüm üzerinden yerel çözüm arayışı yapılabilir.

Diğer taraftan tabu algoritmasını kısıtlı sıralama sorunlarında da kullanmak mümkündür. Bu bağlamda yapılacak ilave işlem, kısıtların sağlanmaması durumunda amaç fonksiyonuna ilave terimler (ceza maliyeti) eklenmesidir. Algoritmanın kısıtlı eniyileme sorununda uygulanması sadedinde müteakip sorunu çözmek mümkündür. Literatürde asgari maliyetle kapsama ağacı<sup>1</sup> olarak bilinen sorunda  $n$  adet noktanın birbirleri ile toplamda en kısa mesafe ile bağlanması hedeflenir. Bu meyanda  $A, B, C, D$  ve  $E$  gibi beş noktanın birbirleri ile bağlanma maliyetleri aşağıdaki gibi verilmiş olsun;



Şekil 3.4 Örnek niteliğindeki asgari maliyetle kapsama ağacı sorunu.

Eniyileme sorunundaki amaç, tüm konumların kapsam içine alınmasıdır. Ancak kapsama bağlantıları oluşturulurken iki kısıtın sağlanması gerekmekte olup bu kısıtlardan ilki *AD* bağlantısının sadece *DE* bağlantısı mevcutken mümkün olabileceği, aksi halde ilave 100 puanlık bir cezanın söz konusu olduğu şeklindedir. İkinci kısıt ise bağlantı ağacı oluşumunda *AD, CD* ve *AB* bağlantılarından sadece birinin aynı anda olabileceği, aksi halde iki bağlantının bulunması durumunda ilave 100 puanlık bir cezanın, üç bağlantının da bağlantı ağacında olması durumunda ilave 200 puanlık cezanın söz konusu olacağı şeklindedir. Buna göre amaç fonksiyonu değerini;

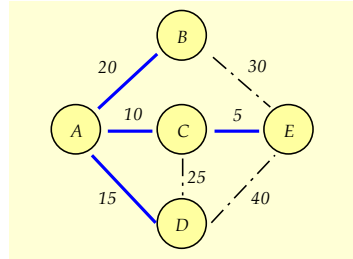
$$f = \sum_{k=1}^{n-1} p_{ij} + \sum_{k=1}^m g_k$$

<sup>1</sup> Minimum spanning tree





ifadesi ile belirlemek mümkündür. Burada  $n$  nokta sayısını,  $p_{ij}$  bağlantı kurulan herhangi  $i$  ve  $j$  noktaları arasındaki bağlantı maliyetini,  $m$  kısıt sayısını,  $g_k$  ise koşulun sağlanamaması durumunda etki edecek ceza maliyetini göstermektedir. Tabu algoritması ile sorunun çözümü sürecinde yapılacak ilk işlem, daha önce de ifade edildiği üzere bir başlangıç çözümünün öngörülmesidir. Ayrıca üzerinde çalışılan sorun için azami tekrar sayısının 10, tabu listesi uzunluğunun ise en fazla 20 çözüm noktası olacak şekilde düzenlenmesi ölçütleri dikkate alınabilir. Buna göre iterasyon sırası olarak 0 indisi ile gösterilebilen başlangıç çözümü olarak;

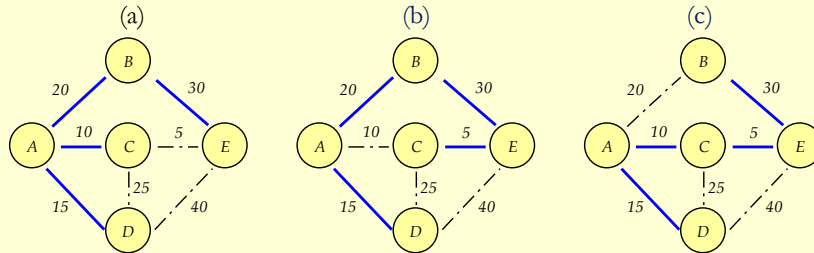


$$f = \sum_{k=1}^4 p_{ij} + \sum_{k=1}^2 g_k$$

$$f = [20 + 10 + 15 + 5] + [100 + 100] = 250$$

seçeneği dikkate alınabilir ve akabinde bu seçenek üzerinden yerel olası çözümler üretilebilir. Önerilen çözüme göre bağlantı ağında dikkate alınmayan  $BE$ ,  $DE$  ve  $CD$  hatları ekleme olasılıkları içerisinde iken her bir ekleme olasılığına karşılık farklı hatlar devreden çıkarılabilir. Müteakip çizelge ve şekillerde olası yerel çözümler ve maliyet değerleri görülmektedir;

İnci iterasyon



$BE$  ekle  $CE$  çıkar

$$f = [20 + 10 + 15 + 30] + [100 + 100] = 275$$

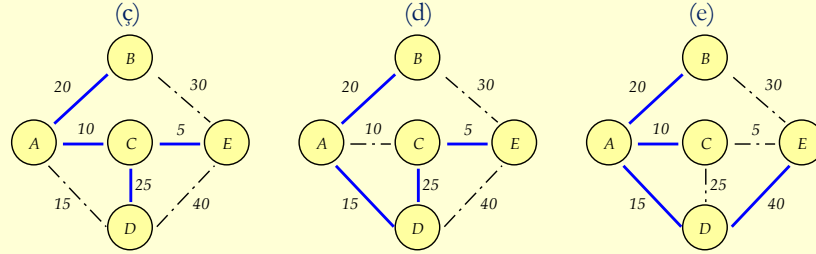
$BE$  ekle  $AC$  çıkar

$$f = [20 + 15 + 5 + 30] + [100 + 100] = 270$$

$BE$  ekle  $AB$  çıkar

$$f = [10 + 15 + 5 + 30] + [100] = 160$$

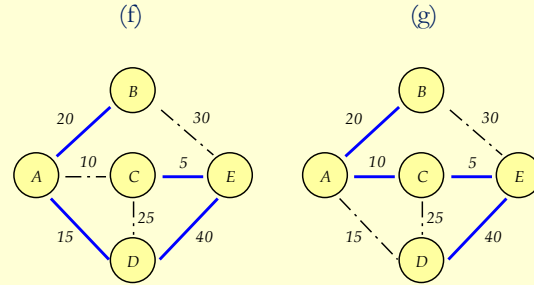




$$CD \text{ ekle } AD \text{ çıkar} \quad \left| \quad f = [20 + 10 + 5 + 25] + [100] = 160$$

$$CD \text{ ekle } AC \text{ çıkar} \quad \left| \quad f = [20 + 15 + 25 + 5] + [100 + 200] = 365$$

$$DE \text{ ekle } CE \text{ çıkar} \quad \left| \quad f = [20 + 10 + 15 + 40] + [100] = 185$$



$$DE \text{ ekle } AC \text{ çıkar} \quad \left| \quad DE \text{ ekle } AD \text{ çıkar} \quad \left| \quad f = [20 + 15 + 5 + 40] + [100] = 180 \quad \left| \quad f = [20 + 10 + 5 + 40] + [0] = 75$$

Oluşturulan yerel olası çözümler içerisinde en iyi çözümün 75 puan maliyetle 1inci iterasyon  $g$  şıkkı: 1( $g$ ) kapsama ağacı olduğu görülmektedir. Diğer taraftan başlangıç çözümü ile beraber diğer olası çözümler oluşturulacak ilk tabu listesine dâhil edilir;

$$T = \{0, 1(a), 1(b), 1(c), 1(d), 1(e), 1(f), 1(g)\}$$

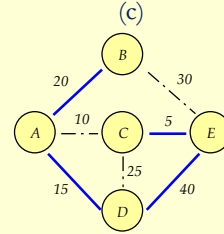
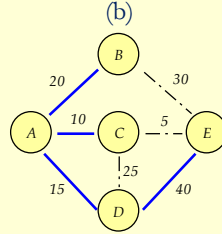
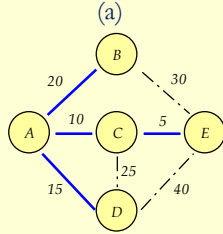
İkinci iterasyonda eldeki en iyi çözümden hareketle yeni yerel çözümler oluşturulur. Önerilen çözüme göre bağlantı ağında dikkate alınmayan  $BE$ ,  $AD$  ve  $CD$  hatları, ekleme olasılıkları içerisinde iken her bir ekleme olasılığına karşılık farklı hatlar devreden çıkarılabilir. Müteakip çizelgede olası yerel çözümler ve maliyet değerleri görülmektedir;







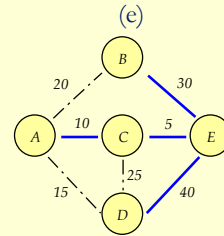
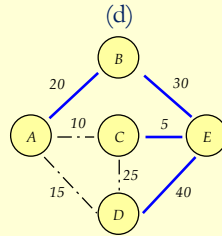
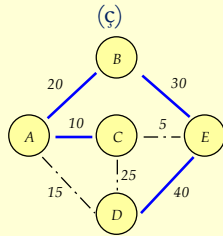
2nci iterasyon



*AD* ekle *DE* çıkar  
Tabu listesinde mevcut: 0

*AD* ekle *CE* çıkar  
Tabu listesinde mevcut: 1(e)

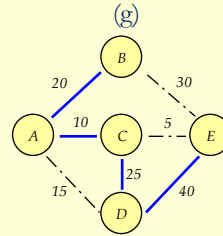
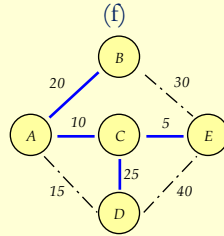
*AD* ekle *AC* çıkar  
Tabu listesinde mevcut: 1(f)



*BE* ekle *CE* çıkar  
 $f = [20 + 10 + 30 + 40] + [0] = 100$

*BE* ekle *AC* çıkar  
 $f = [20 + 30 + 5 + 40] + [0] = 95$

*BE* ekle *AB* çıkar  
 $f = [30 + 10 + 5 + 40] + [0] = 85$



*CD* ekle *DE* çıkar  
Tabu listesinde mevcut: 1(ç)

*CD* ekle *CE* çıkar  
 $f = [20 + 10 + 25 + 40] + [100] = 195$

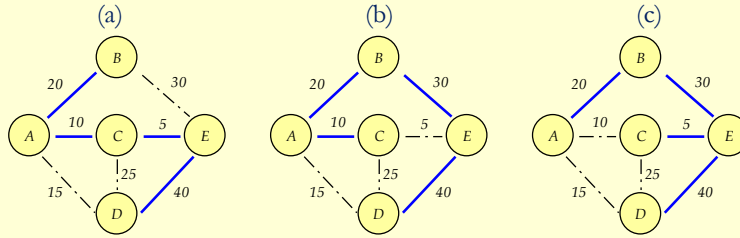


Oluşturulan yerel olası çözümler içerisinde en iyi çözümün 85 puan maliyetle 2(e) kapsama ağacı olduğu görülmektedir. Diğer taraftan başlangıç çözümü ile beraber listede bulunmayan diğer olası çözümler, oluşturulacak ilk tabu listesine dâhil edilir;

$$T = \left\{ \begin{array}{l} 0, \\ 1(a), 1(b), 1(c), 1(d), 1(e), 1(f), 1(g) \\ 2(\zeta), 2(d), 2(e), 2(g) \end{array} \right\}$$

Üçüncü iterasyonda eldeki en iyi çözümden hareketle yeni yerel çözümler oluşturulur. Önerilen çözüme göre bağlantı ağında dikkate alınmayan  $AB, AD$  ve  $CD$  hatları, ekleme olasılıkları içerisinde iken her bir ekleme olasılığına karşılık farklı hatlar devreden çıkarılabilir. Müteakip çizelgede yeni iterasyona ait olası yerel çözümler ve maliyet değerleri görülmektedir;

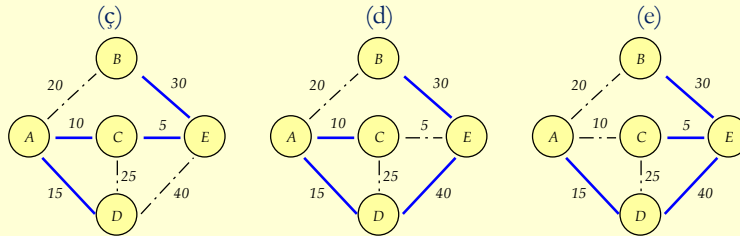
3üncü iterasyon



$AB$  ekle  $BE$  çıkar  
Tabu listesinde mevcut: 1(g)

$AB$  ekle  $CE$  çıkar  
Tabu listesinde mevcut: 2( $\zeta$ )

$AB$  ekle  $AC$  çıkar  
Tabu listesinde mevcut: 2(d)

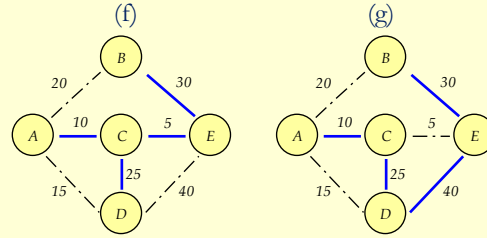


$AD$  ekle  $DE$  çıkar  
Tabu listesinde mevcut: 1(g)

$AD$  ekle  $CE$  çıkar  
 $f = [10 + 15 + 30 + 40] + [0] = 95$

$AD$  ekle  $AC$  çıkar  
 $f = [15 + 40 + 5 + 30] + [0] = 90$





$$f = [10 + 25 + 5 + 30] + [0] = 70 \quad \left| \quad \begin{array}{l} CD \text{ ekle } DE \text{ çıkar} \\ CD \text{ ekle } CE \text{ çıkar} \end{array} \right. \quad f = [10 + 25 + 40 + 30] + [0] = 105$$

Oluşturulan yerel olası çözümler içerisinde en iyi çözümün 70 puan maliyetle 3(f) kapsama ağacı olduğu görülmektedir. Diğer taraftan başlangıç çözümü ile beraber diğer olası çözümler oluşturulacak ilk tabu listesine dâhil edilir;

$$T = \left\{ \begin{array}{l} 0 \\ 1(a), 1(b), 1(c), 1(d), 1(e), 1(f), 1(g) \\ 2(c), 2(d), 2(e), 2(g) \\ 3(d), 3(e), 3(f), 3(g) \end{array} \right\}$$

Dördüncü iterasyonda daha önce yapılan işlemlere benzer şekilde eldeki en iyi çözümden hareketle yeni yerel çözümler oluşturulur. Önerilen çözüme göre bağlantı ağında dikkate alınmayan  $AB, AD$  ve  $ED$  hatları, ekleme olasılıkları içerisinde iken her bir ekleme olasılığına karşılık farklı hatlar devreden çıkarılabilir. Müteakip tekrarlı işlemler benzer şekilde yapıldığında görülecektir ki 10 iterasyon sonunda elde edilen en iyi çözüm, 3(f) kapsama ağacı olup bu çözümde toplam maliyet 70'tir<sup>(22)</sup>.

Tabu algoritmasının yaygın olarak sıralama sorunlarında kullanılması yanında uygun kısıtların dikkate alınması kaydıyla gerçek sayıların söz konusu olduğu sorunlara da uygulanması mümkündür. Böyle bir yaklaşımda genellikle birbiri ile ilişkili iki ayrı tabu listesinin oluşturulması ve karşılaştırmaların kademeli olarak gerçekleştirilmesi yapılabilir. Örneğin üretilen çözümlerin amaç fonksiyon değerlerini içeren ilk tabu listesi ve aynı çözümlere ait tasarım yöneylerini içeren ikinci bir tabu listesi oluşturulabilir. Diğer taraftan yerel çözüm üretiminde belirli bir sistematik dâhilinde rassal sayılar kullanılabilir. Üretilen yerel çözümün tabu listesinde olup olmadığı öncelikle ilk tabu listesi dâhilindeki amaç fonksiyonu değerleri ile karşılaştırılır. Genellikle amaç fonksiyonu değerinin tabu listesindeki amaç fonksiyonu değerlerine göre belli bir aralıkta olup olmadığı kıyas edilir;





$$f(\mathbf{x}_y) \in [f(\mathbf{x}_i) - \%B_1 \cdot f(\mathbf{x}_i), f(\mathbf{x}_i) + \%B_1 \cdot f(\mathbf{x}_i)] \quad (3.12)$$

Burada  $\%B_1$  ifadesi, amaç fonksiyonu deęerleri arasındaki yakınlığı ifade eden ilk bant genişlięi yüzdesini göstermektedir. Yerel çözümün belirtilen bant aralıęında olması durumunda ikinci bir kıyas işlemi uygulanır. Dikkate alınan eşleşmeye ait tasarım yöneyleri arasında;

$$\mathbf{x}_y \in [\mathbf{x}_i - \%B_2 \cdot \mathbf{x}_i, \mathbf{x}_i + \%B_2 \cdot \mathbf{x}_i] \quad (3.13)$$

ilişkinin söz konusu olması durumunda üretilen yerel çözümün tabu listesinde olduęu düşünülerek başka çözümlere yönelinir. Burada  $\%B_2$  ifadesi tasarım yöneyi deęerleri arasındaki yakınlığı ifade eden ikinci bant genişlięi yüzdesini göstermektedir<sup>(25)</sup>.





## “Gezici Satıcı Sorunu”

Dünya üzerindeki 30 şehre ait gezi programının yapılması kapsamında tipik bir gezici satıcı sorununu tabu algoritmasını kullanarak çözmek mümkündür. Üç boyutlu uzay ortamında şehirler, şehirlerarası mesafeler ve her bir şehre ait konum verisi müteakip satırlarda verilmiştir<sup>(20)</sup>. Mesafe değerleri 100 mil ile ölçeklenmiştir.

Şehirler; Azores, Baghdad, Berlin, Bombay, Buenos Aires, Cairo, Capetown, Chicago, Guam, Honolulu, Istanbul, Juneau, London, Manila, Melbourne, Mexico City, Montreal, Moscow, New Orleans, New York, Panama City, Paris, Rio de Janeiro, Rome, San Francisco, Santiago, Seattle, Shanghai, Sydney, Tokyo şeklindedir.



Mesafe matrisi;

```
[ 0 39 22 59 54 33 57 32 89 73 29 46 16 83 120 45 24 32 36 25 38 16 43 21 50 57 46 72 121 73
39 0 20 20 81 8 49 64 63 84 10 61 25 49 81 81 58 16 72 60 78 24 69 18 75 88 68 44 83 52
22 20 0 39 74 18 60 44 71 73 11 46 6 61 99 61 37 10 51 40 59 5 62 7 57 78 51 51 100 56
59 20 39 0 93 27 51 81 48 80 30 69 45 32 61 97 75 31 89 78 97 44 83 38 84 100 77 31 63 42
54 81 74 93 0 73 43 56 104 76 76 77 69 111 72 46 56 84 49 53 33 69 12 69 64 7 69 122 73 114
33 8 18 27 73 0 45 61 71 88 8 63 22 57 87 77 54 18 68 56 71 20 61 13 75 80 68 52 90 60
57 49 60 51 43 45 0 85 88 115 52 103 60 75 64 85 79 63 83 78 70 58 38 52 103 49 102 81 69 92
32 64 44 81 56 61 85 0 74 43 55 23 40 81 97 17 8 50 8 7 23 41 53 48 19 53 17 70 92 63
89 63 71 48 104 71 88 74 0 38 69 51 75 16 35 75 77 61 77 80 90 76 116 76 58 98 57 19 33 16
73 84 73 80 76 88 115 43 38 0 81 28 72 53 55 38 49 70 42 50 53 75 83 80 24 69 27 49 51 39
29 10 11 30 76 8 52 55 69 81 0 55 16 57 91 71 48 11 62 50 68 14 64 9 67 81 61 49 93 56
46 61 46 69 77 63 103 23 51 28 55 0 44 59 81 32 26 46 29 29 45 47 76 53 15 73 9 49 77 40
16 25 6 45 69 22 60 40 75 72 16 44 0 67 105 56 33 16 46 35 53 2 57 9 54 72 48 57 106 60
83 49 61 32 111 57 75 81 16 53 57 59 67 0 39 88 82 51 87 85 103 67 113 65 70 109 67 12 39 19
120 81 99 61 72 87 64 97 35 55 91 81 105 39 0 84 104 90 93 104 90 104 82 99 79 70 82 50 4 51
45 81 61 97 46 77 85 17 75 38 71 32 56 88 84 0 23 67 9 21 15 57 48 64 19 41 23 80 81 70
24 58 37 75 56 54 79 8 77 49 48 26 33 82 104 23 0 44 14 3 25 34 51 41 25 54 23 70 100 65
32 16 10 32 84 18 63 50 61 70 11 46 16 51 90 67 44 0 58 47 67 16 72 15 59 88 52 42 90 47
36 72 51 89 49 68 83 8 77 42 62 29 46 87 93 9 14 58 0 12 16 48 48 55 19 45 21 77 89 69
25 60 40 78 53 56 78 7 80 50 50 29 35 85 104 21 3 47 12 0 22 36 48 43 26 51 24 73 100 68
38 78 59 97 33 71 70 23 90 53 68 45 53 103 90 15 25 67 16 22 0 54 33 59 33 31 37 93 88 84
16 24 5 44 69 20 58 41 76 75 14 47 2 67 104 57 34 16 48 36 54 0 57 7 56 72 50 57 105 61
43 69 62 83 12 61 38 53 116 83 64 76 57 113 82 48 51 72 48 48 33 57 0 57 66 18 69 113 84 115
21 18 7 38 69 13 52 48 76 80 9 53 9 65 99 64 41 15 55 43 59 7 57 0 63 74 57 57 101 61
50 75 57 84 64 75 103 19 58 24 67 15 54 70 79 19 25 59 19 26 33 56 66 63 0 59 7 61 74 52
57 88 78 100 7 80 49 53 98 69 81 73 72 109 70 41 54 88 45 51 31 72 18 74 59 0 64 117 71 107
46 68 51 77 69 68 102 17 57 27 61 9 48 67 82 23 23 52 21 24 37 50 69 57 7 64 0 57 77 48
72 44 51 31 122 52 81 70 19 49 49 49 57 12 50 80 70 42 77 73 93 57 113 57 61 117 57 0 49 11
121 83 100 63 73 90 69 92 33 51 93 77 106 39 4 81 100 90 89 100 88 105 84 101 74 71 77 49 0 48
73 52 56 42 114 60 92 63 16 39 56 40 60 19 51 70 65 47 69 68 84 61 115 61 52 107 48 11 48 0]
```



MATLAB uygulaması

Konum verileri;

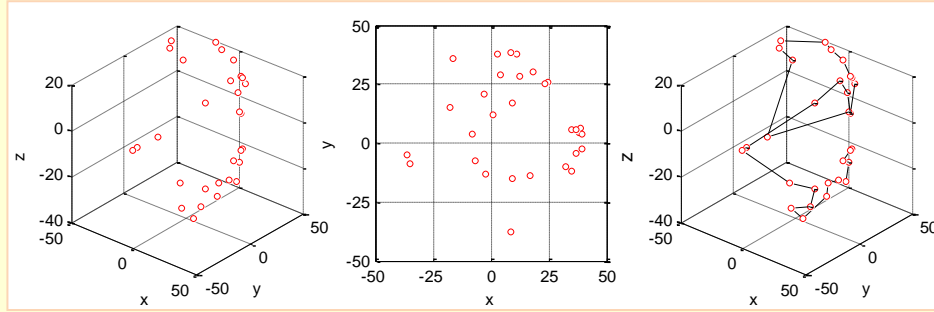
```
[36.902328 5.865524 13.083650
32.268564 -9.772170 -20.751207
38.430958 4.616171 -8.313876
17.085076 -13.717743 -32.974108
-2.378420 -12.912119 37.349544
35.229322 -11.952753 -13.542328
8.734037 -37.925268 7.263521
18.213856 30.067160 18.209597
-17.367236 14.994152 -32.263332
-16.055342 36.092639 -2.629740
36.893950 -4.489910 -13.639841
11.354847 37.879510 -1.891633
39.040952 6.060413 -2.538418
-7.949413 3.900184 -38.587090
-35.113496 -9.111273 -15.855447
3.920608 28.751273 26.932531
24.411612 26.044301 17.121206
34.976862 5.514005 -17.708273
12.745841 28.464310 24.386363
23.195509 25.060627 20.032510
9.107489 16.970196 34.589510
39.281107 4.053225 -2.816748
9.510174 -15.065098 35.354879
39.103806 -2.333589 -5.728422
3.162014 37.917558 10.937482
-6.567441 -7.311222 38.350787
8.555034 38.095727 6.549431
0.938855 11.778574 -37.785604
-36.020211 -4.791656 -15.714724
-3.040275 20.917682 -33.474997]
```



Komut<sup>(27)</sup> ve sonucu (yazılım için kaynağa başvurunuz).

```
best_tour =
13  22  24  3  18  11  6  2  4  14  28  30  9  15
29  10  25  27  12  17  20  8  19  16  21  26  5  23
7  1
```

Müteakip şekillerde sırasıyla şehirlerin 3 boyutlu ortamda konumları, konumların üstten görünüşü ve tabu taraması neticesinde elde edilen çözüme ait şehir planı görülmektedir;



### 3.4 Markov Zinciri

Özellikle rastsal karaktere sahip üst-sezgisel yöntemlerin kuramsal modellemelerinde kullanılan Markov zinciri yaklaşımının irdelenmesinin, sezgisel yöntemlerin doğasını anlama bağlamında faydalı olacağı değerlendirilmektedir. Bu nedenle müteakip paragraflarda Markov zinciri ele alınmaktadır.

Literatürde Markov zinciri<sup>i</sup> olarak bilinen ve esasen stokastik süreçleri inceleyen yaklaşımın temeli, 1900'lü yıllarda A. Markov'un rassal süreçlerle ilgili çalışmalarına dayanır. Markov zinciri, sistemin bir halden başka bir hale<sup>ii</sup> geçişini irdelleyen süreçtir. Ancak bahse konu geçişte bazı belirli özellikler söz konusudur. Örneğin bir halden bir sonraki hale geçişte etkin olan durum sadece içinde bulunulan durumdur. Dolayısıyla seri halinde devam eden bir süreçte durumlar

<sup>i</sup> Markov chains

<sup>ii</sup> State

arası etkileşim ardışık iki hal için geçerlidir. Bir sonraki halin özelliklerini geçmişteki haller değil sadece mevcut hal belirler. Bu anlamda Markov zinciri hafızasızdır<sup>i</sup>. Sistemin halleri arasındaki değişimler, geçiş<sup>ii</sup> süreci olarak adlandırılır. Süreci belirleyen temel kavramlar; başlangıç hali, geçiş olasılık matrisi ve durum uzayıdır<sup>iii</sup>. Durum uzayı, geçiş sonrası ortaya çıkacak tüm olası halleri kapsar. Geçiş sürecinin temel özelliği rassal bir karaktere sahip olmasıdır. Bu nedenle içinde bulunulan halden bir sonraki hali, belirlenimci bir bakış açısıyla öngörmek mümkün değildir<sup>(28)</sup>.

Durum uzayının  $S = \{s_1, s_1, \dots, s_i, \dots, s_n, \}$  şeklinde tanımlandığı bir Markov zincirinde, içinde bulunulan  $s_i$  durumundan bir sonraki  $s_j$  haline geçiş ihtimalini belirleyen olasılık değeri  $p_{ij}$  olarak ifade edilebilir. Örneğin  $s_1$  halinden  $s_5$  haline geçiş ihtimali  $p_{15}$  şeklinde tanımlanır. Diğer taraftan içinde bulunulan halin aynen tekrarlanma olasılığı da söz konusu olup bu durumda geçiş olasılığı  $p_{ii}$  şeklindedir. Olasılık değerlerinin dayanağının tecrübe ve geçmişe dönük eldeki veriler olduğu söylenebilir. Durum uzayı belirli sayıda hali kapsayabilir, ancak sürecin sonu söz konusu değildir. Zira süreç, belli ihtimaller çerçevesinde sürekli değişim gösterebilir. Markov zincirinin daha iyi anlaşılabilmesi için aşağıdaki gibi bir örnek dikkate alınabilir. Buna göre hava durumunun değişkenlik gösterdiği bir şehirde üç farklı hava durumu söz konusu olsun: yağmurlu ( $Y$ ), güneşli ( $G$ ) ve karlı ( $K$ ) hava. Belirtilen hava durumlarını dikkate alarak üç elemanlı durum uzayını  $S = \{Y, G, K\}$  biçiminde ifade edebiliriz. Diğer taraftan yağmurlu bir gün söz konusu iken ertesi günün yine yağmurlu olma ihtimalinin  $1/2$ , güneşli olma ihtimalinin  $1/4$ , karlı olma ihtimalinin ise  $1/4$  olduğunu düşünelim. Benzer olasılıkları bugünün güneşli olması veya karlı olması halinde de ifade edebiliriz. Eldeki verilere göre günler arası geçişte hava durumunun olasılıklarını içeren  $P$  geçiş olasılık matrisini;

$S$	$Y$	$G$	$K$
$Y$	$1/2$	$1/4$	$1/4$
$G$	$1/2$	$0$	$1/2$
$K$	$1/4$	$1/4$	$1/2$

$P = \begin{bmatrix} 1/2 & 1/4 & 1/4 \\ 1/2 & 0 & 1/2 \\ 1/4 & 1/4 & 1/2 \end{bmatrix}$

olarak somutlaştırmak mümkündür. Görüldüğü üzere burada yatay ve dikey başlıklar hava durumlarını (durum uzayını) göstermektedir. Örneğin matrise göre bugünün güneşli ( $G, i = 2$ ) olması halinde yarının tekrar güneşli ( $G, i = 2$ ) olma ihtimali  $p_{22} = 0$  olarak görünmektedir. Benzer çıkarsamalar diğer günler ve ertesi günleri için de geçerlidir. Bugünün hava durumuna göre  $t$  gün

<sup>i</sup> Memorylessness

<sup>ii</sup> Transition

<sup>iii</sup> An initial state, transition probability matrix, and a state space



sonrasının hava durumunun gerçekleşme olasılığı ise  $p_{ij}^{(t)}$  terimiyle ifade edilmektedir. Örneğin bugün yağmurlu ( $i = 1$ ) olduğunu düşünelim. İki gün sonra havanın karlı ( $j = 3$ ) olma ihtimalini  $p_{13}^{(2)}$  ifadesi ile simgeselleştirebiliriz. Bu olasılığın gerçekleşmesi için farklı öngörüler söz konusu olabilir;

- Hava bugün yağmurlu iken yarın tekrar yağmurlu ve iki gün sonrasında da karlı olabilir. Yarının yağmurlu olma ihtimali  $p_{11} = 1/2$ 'dir. Yağmurlu yarın sonrasında karlı hava durumunun olma ihtimali  $p_{13} = 1/4$ 'tür. Her iki ihtimalin gerçekleşme olasılığı ise;

$$p_{11}p_{13} = \frac{1}{2} \times \frac{1}{4} = \frac{1}{8} \text{ dir.}$$

- Hava bugün yağmurlu iken yarın güneşli ve iki gün sonrasında karlı olabilir. Yarının güneşli olma ihtimali  $p_{12} = 1/4$ 'tür. Güneşli yarın sonrasında karlı hava durumunun olma ihtimali  $p_{23} = 1/2$ 'dir. Her iki ihtimalin gerçekleşme olasılığı ise;

$$p_{12}p_{23} = \frac{1}{4} \times \frac{1}{2} = \frac{1}{8} \text{ dir.}$$

- Havanın bugün yağmurlu iken yarın karlı ve iki gün sonrasında karlı olma ihtimali söz konusudur. Yarının karlı olma ihtimali  $p_{13} = 1/4$ 'tür. Karlı yarın sonrasında karlı hava durumunun olma ihtimali  $p_{33} = 1/2$ 'dir. Her iki ihtimalin gerçekleşme olasılığı ise;

$$p_{13}p_{33} = \frac{1}{4} \times \frac{1}{2} = \frac{1}{8} \text{ dir.}$$

Durum uzayı çerçevesinde ifade edilen bu hallerin beraberce dikkate alınması halinde iki gün sonra havanın karlı olma ihtimalini  $p_{13}^{(2)} = p_{11}p_{13} + p_{12}p_{23} + p_{13}p_{33} = 3/8$  şeklinde elde etmek mümkündür. Olasılık hesaplamasının genelleştirilmesi çerçevesinde bu ifadeyi;

$$p_{ij}^{(2)} = \sum_{k=1}^n p_{ik}p_{kj}, \quad n = 3 \quad (3.14)$$

denklemi ile simgeselleştirebiliriz. Görüldüğü üzere bir sonraki günün hava durumunun ne olacağı sadece o güne ve geçiş olasılıklarına bağlı olup iki gün veya daha öncesi geçmişine bağlı değildir. İki gün sonrasına ait geçiş olasılık matrisini, yukarıda ifade edilen benzer olasılıkları dikkate alarak;



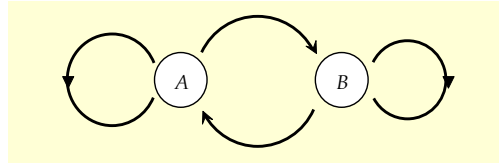


S	Y	G	K	
Y	$P^{(2)} =$	0.438	0.188	0.375
G		0.375	0	0.375
K		0.375	0.188	0.438

biçiminde elde etmek mümkündür<sup>(29)</sup>. Öte yandan Markov zincirini, az sayıdaki durum uzayı söz konusu olduğunda şekilsel olarak görselleştirmek de mümkündür. Örneğin  $A$  ve  $B$  gibi iki durumun söz konusu olduğu bir durum uzayında geçiş olasılık matrisi;

S	A	B	
A	$P =$	$p_{11}$	$p_{12}$
B		$p_{21}$	$p_{22}$

şeklinde yazılabilir. Aynı zincir, müteakip şekilde görüldüğü üzere görsel hale de getirilebilir. Buna göre, sistem  $A$  halinden tekrar  $A$  haline veya  $B$  haline geçebilir. Aynı durum diğer başlangıç durumu için de geçerlidir: sistem  $B$  halinden tekrar  $B$  haline veya  $A$  haline geçebilir.



Şekil 3.5 İki halli bir durum uzayı için şekilsel gösterim<sup>(30)</sup>.

### 3.5 Rastlantısal tarama yöntemi

K. Pearson'ın 1905 yılında yayımladığı “Rastsal yürüyüş sorunu” adlı makalesi ile literatüre kazandırılan rastlantısal tarama<sup>ii</sup> kavramı, ardı ardına atılan rastgele adımlara dayalı bir tarama sürecidir<sup>(31)</sup>. Bünyesinde barındırdığı rassal hareketler nedeniyle süreç ve sürecin geleceğine dair yaklaşımlar rassal sayılara ve dolayısıyla olasılık dağılımlarına bağlıdır. Geçmiş dönemde rastsal tarama sorunu basit bir örnekle, bir sarhoşun bardan çıkıp evine gitmesi veya bir kazaya kurban olup uçurumdan aşağıya yuvarlanması mizansenî ile örneklendirilirdi. Düşünce yoksunluğu nedeniyle hareketlerinin rassal olduğu düşünülen sarhoşun, bulunduğu  $x$  konumundayken evine varma olasılığını simgesel olarak;

<sup>i</sup> “The problem of the random walk”

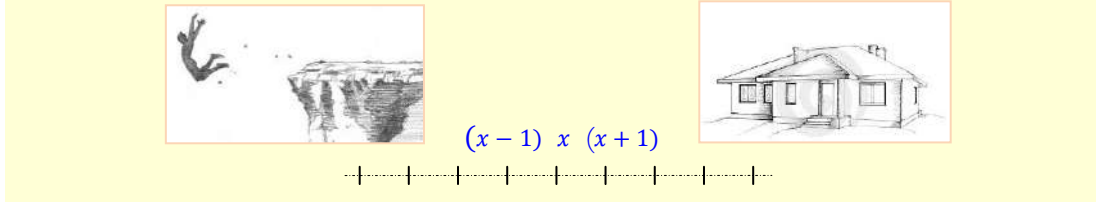
<sup>ii</sup> Random walk





$P_x$

şeklinde ifade etmek mümkündür. Bu tanımlamaya göre konum olarak evde bulunulduğunda  $P_e = 1$  ve uçurumda bulunulduğunda ise  $P_u = 0$  yazılabilir.



Şekil 3.6 Bardan çıkan sarhoşun<sup>1</sup> geleceğine dair öngörü.

Diğer taraftan basitleştirme adına bir boyutlu bir hareket ortamında olduğu düşünülen sarhoşun, eve doğru atacağı adım ile uçuruma doğru atacağı adımın olasılığının eşit olduğu düşünülebilir. Dolayısıyla  $x$  konumuna göre sarhoşun evine varma olasılığı sonraki konumlara dayalı olarak;

$$P_x = \frac{1}{2}P_{x-1} + \frac{1}{2}P_{x+1} \quad (3.15)$$

ifadesi ile betimlenebilir. Bu denklem, konum yanında zaman kavramını da içerecek şekilde yeniden yazılabilir. Buna göre  $(n+1)$ 'inci adımda  $x$  pozisyonuna ait olasılık,  $n$ 'inci adımda bulunan  $x-1$  veya  $x+1$  konum olasılıklarına göre tekrar yazılırsa;

$$P_{x,n+1} = \frac{1}{2}P_{x-1,n} + \frac{1}{2}P_{x+1,n} \quad (3.16)$$

denkleme ulaşılabilir<sup>(32)</sup>. Elde edilen denklemin  $n$ 'inci adım ve  $x$  konumu olasılığını da içerecek şekilde güdülenmesi çerçevesinde eşitliğin her iki tarafından  $P_{x,n}$  terimi çıkarılıp yeniden düzenleme yapılırsa;

$$P_{x,n+1} - P_{x,n} = \frac{1}{2}[P_{x-1,n} - 2P_{x,n} + P_{x+1,n}] \quad (3.17)$$

eşitliğine varılır. Nispeten basit ve belirli bir mantık çerçevesinde elde edilen bu ifade, 1800'lü yıllarda R. Brown (1773-1858) tarafından ortaya atılan ve uzun süre çözülemeyen bir sorunun

<sup>1</sup> Drunkard's walk

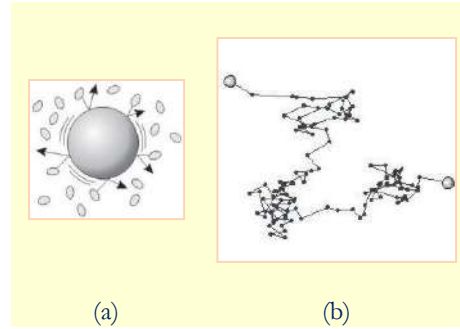




çözümü niteliğindedir. Rastlantısal tarama yaklaşımının bazı kaynaklarda Brown tarzı hareket<sup>i</sup> olarak da adlandırılmasına neden olan söz konusu sorun, basit bir gözleme dayanır. 1826 yılında içine polen benzeri çok küçük parçacıkların atıldığı su damlasındaki hareketleri mikroskopla gözlemleyen Brown, söz konusu parçacıkların rastgele hareket ettiklerini ifade etmiş ve nedenini açıklayamamıştı. 19uncu yüzyıl boyunca üzerinde konuşulan ama açıklanamayan söz konusu hareket, 20nci yüzyıl başlarında, 1905 yılında ünlü fizikçi Einstein tarafından çözümlenebildi. Ona göre Brown tarzı hareketin nedeni, parçacıkların uzay-zaman ortamında rastgele hareket etmeleriydi. Söz konusu rastlantısal hareketin nedeni ise akışkan ortamında rastgele hareket eden akışkan moleküllerin çarpışmalar nedeniyle polenlere ilettiği momentum ve nihayetinde kuvvet etkileşimleriydi. Mekaniğin temel prensipleri çerçevesinde Einstein'ın öne sürdüğü konum bazlı hareket denklemi;

$$\frac{\partial P}{\partial t} = D \frac{\partial^2 P}{\partial x^2}$$

şeklindeydi. Burada  $D$  terimi, difüzyon katsayısı olup belli kabuller çerçevesinde  $\frac{1}{2}$  olarak dikkate alınabilmekteydi. Dikkat edileceği üzere Einstein'ın zamana ve mekâna bağlı olarak ifade edilen analitik denklemin sayısal açılımı, daha önce elde edilen ve rastgele hareket eden sarhoş adam için geçerli olan denklemle büyük benzerlik arz etmektedir<sup>(33)</sup>.



**Şekil 3.7** Brown tarzı harekete dair görsel açıklama; (a) rastlantısal olarak moleküller ve parçacık arasındaki etkileşim, (b) parçacığın izlediği rastlantısal yol<sup>(34)</sup>.

<sup>i</sup> Brownian motion





## “Brown tarzı hareket”



Komut<sup>(35)</sup>

**MATLAB uygulaması**

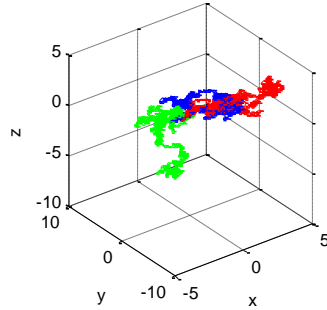


```
clear; clc;
h= input('Adim buyuklugu: '); N= input('Adim sayisi: '); np=input('Parcacik
sayisi: ');
% parçacıkların çıkış merkezi
x = zeros(1,np); y = zeros(1,np); z = zeros(1,np);
for j=1:np
    for i=1:N
        x(i+1,j)=x(i,j)+h*randn();
        y(i+1,j)=y(i,j)+h*randn();
        z(i+1,j)=z(i,j)+h*randn();
    end
end
cmap = hsv(np); % renk düzenlemesi
for k=1:np
    plot3(x(:,k),y(:,k),z(:,k),'Color',cmap(k,:));
    hold on;
end
grid on;
```



Sonuç

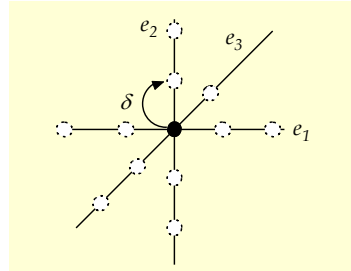
Adim buyuklugu: 0.1  
Adim sayisi: 1000  
Parcacik sayisi: 3



Diğer taraftan rastlantısal yürüyüşte bir sonraki adım, tamamen rassal olmakla beraber son hareket itibari ile konum, önceki hareketlerin toplamı niteliğindedir. Bu nedenle  $n$ 'inci adımdaki konumu;

$$p_n = \sum_{i=1}^n x_i \quad (3.18)$$

ifadesi ile elde etmek mümkündür. Burada  $x_i$  terimi, birbirinden bağımsız rassal adımları temsil etmektedir<sup>(36)</sup>. Atılan adımların adım büyüklüğü ve yönü itibari ile tamamen rassal olması mümkündür. Bununla beraber adım büyüklüğü bazen sabit, bazen de 1 boyutlu arama yöntemleri ile elde edilen, belirli ama optimal bir değer olarak dikkate alınabilir. Diğer taraftan yöntemde adım doğrultusunun rassal olarak seçilmesi kesindir. Yönü rastsallığa dayanan bir seyahatte rassal sayıların eldesi genellikle olasılık dağılım fonksiyonları ile karakterize edilir. Bu nedenle rastlantısal tarama, dikkate alınan dağılım fonksiyonuna göre “Cauchy uçuşu”, “Rayleigh uçuşu” ve benzeri isimlerle de adlandırılır. Dağılım fonksiyonları, belirli bir tercihle kullanılabilceği gibi farklı yönlerde farklı dağılım fonksiyonlarının kullanılması da mümkün olabilir. Benzer şekilde farklı yönlerde farklı olasılık değerleri de dikkate alınabilir. Bununla beraber tüm yönlerde eşit olasılığın söz konusu olduğu bir tarama, simetrik rastlantısal tarama olarak adlandırılır.



Şekil 3.8 Üç boyutlu örgü taramasında olası yerel konumlar.

Rastlantısal tarama yönteminde değişik yaklaşımlar söz konusu olabilir. Bununla beraber genellikle üç tip tarama yönteminden bahsetmek mümkündür. Bu yöntemler; örgü taraması, küresel tarama ve atış taramasıdır. Örgü taraması, örgü yapısı içerisindeki her hangi bir noktadan

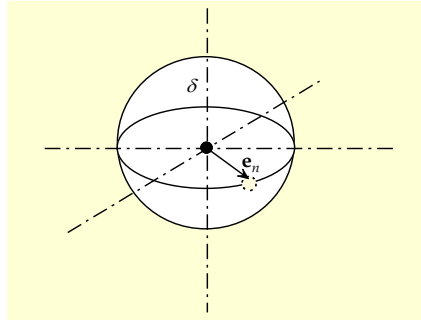
<sup>i</sup> Cauchy flight





başka bir örgü noktasına yönelimi ifade eder. Buna göre  $e_i$  örgü yapısı içerisinde eksenel birim yöney olmak üzere, üzerinde bulunulan  $x$  noktasından hareketle bir sonraki nokta  $x \pm \delta e_i$  şeklinde elde edilebilir. Burada  $\delta$  değeri, sabit bir değer olup örgü noktaları arasındaki eksenel mesafedir. Tabiatıyla  $e_i$  birim yöneyinin belirlenmesi rassal bir süreçtir. Şekil 3.8'e göre 3 boyutlu bir örgü uzayında olası 6 konum söz konusu olup simetrik rastlantısal taramada her bir konumun seçilmesi olasılığı  $1/6$  olarak ifade edilebilir.

Küresel tarama yaklaşımında, üzerinde bulunulan  $x$  noktasından hareketle bir sonraki nokta  $x \pm \delta e_n$  şeklinde elde edilebilir. Şekil 3.9'dan da görüleceği üzere burada  $\delta$  değeri, sabit bir mesafe niteliğindeki küre yarıçapını,  $e_n$  ise herhangi bir yöndeki birim yöneyi temsil etmektedir. İfade edilen tanıma göre bir sonraki nokta,  $\delta$  yarıçapına sahip bir kürenin üzerindeki herhangi bir nokta olarak seçilebilmektedir. Önceki tarama yöntemine benzer şekilde  $e_n$  birim yöneyinin belirlenmesi rassal bir süreçtir.

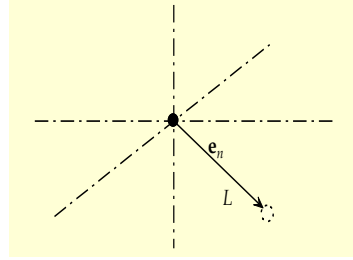


Şekil 3.9 Üç boyutlu küresel taramada olası yerel konum.

Son yaklaşım niteliğindeki atış tarama yaklaşımında, üzerinde bulunulan  $x$  noktasından hareketle bir sonraki nokta  $x \pm L e_n$  şeklinde elde edilebilir. Şekil 3.10'dan da görüleceği üzere burada  $L$  değeri herhangi bir mesafe değeri olup  $e_n$  ise herhangi bir yöndeki birim yöneyi temsil etmektedir. Diğer tarama yöntemine benzer şekilde  $e_n$  birim yöneyinin belirlenmesi rassal bir süreçtir<sup>(37)</sup>.

Rastlantısal tarama sürecinde karşılaşılabilecek diğer bir husus, taramaya dair kısıt seçenekleridir. Taramada dikkate alınan yaklaşıma göre örgü noktası veya küre yarıçapı mesafelerine dair bazı kısıtlamalara gidilebilir<sup>(38)</sup>. Ancak yöntemin doğasının rassal bir süreç olduğu unutulmamalıdır.





Şekil 3.10 Üç boyutlu atış taramasında olası yerel konum.

Rastlantısal tarama yönteminin işlem adımlarını aşağıdaki gibi sıralamak mümkündür;

- ▶ 1. Öncelikle sürece ait  $\mathbf{x}^{(0)}$  başlangıç noktası ve bu noktaya ait  $f(\mathbf{x}^{(0)})$  amaç fonksiyonu değeri tespit edilir. Söz konusu belirlemede, rastgele seçim yapılabileceği gibi tecrübeye dayalı bir öngöründe de bulunulabilir. Akabinde yakınsama ölçütü belirlenir. Daha önce gradyan esaslı yöntemlerde ifade edilen yakınsama ölçütlerine ilave olarak azami tekrar sayısı veya asgari adım büyüklüğü ( $\epsilon$ ) gibi farklı yakınsama ölçütleri dikkate alınabilir.
- ▶ 2. Sonraki adımı belirlemek üzere rassal olarak gidilecek yön tayin edilir. Yön tayininde;

$$\mathbf{d} = \frac{1}{(r_1^2 + r_2^2 + \dots + r_n^2)^{1/2}} \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{bmatrix} \quad (3.19)$$

$$-1 < r_i < 1 \quad (3.20)$$

$$(r_1^2 + r_2^2 + \dots + r_n^2)^{1/2} \leq 1 \quad (3.21)$$

ifadeleri dikkate alınır. Görüldüğü üzere yön tayininde küresel tarama yaklaşımı öngörülmektedir.

- ▶ 3. Gidilecek yönün belirlenmesi sonrasında  $\alpha$  adım büyüklüğünün tespiti gerekir. Adım büyüklüğü için başlangıçta sabit bir değer dikkate alınabilir. Bununla beraber daha etkin bir yöntem olan tek boyutlu tarama seçeneği düşünülebilir. Buna göre;

$$\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} + \alpha \mathbf{d}$$

tanımlamasına ilave olarak;



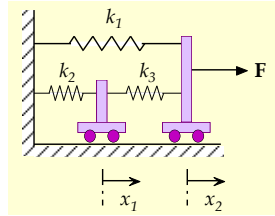


$$\frac{\partial f(\mathbf{x}^{(i+1)})}{\partial \alpha} = 0$$

eşitliği ile optimal adım büyüklüğünün tespiti mümkündür. Hatırlanacağı üzere diğer bir yaklaşım, sayısal nitelikteki altın kesit yöntemi olabilir.

- ▶ 4. Müteakiben sonraki nokta ve bu noktaya ait amaç fonksiyon değeri hesaplanır. Hesaplama sonrasında yakınsama kontrolü ve devamında da karşılaştırmaya gidilir. Elde edilen yeni noktanın önceki noktaya göre daha iyi sonuç vermesi halinde  $(f(\mathbf{x}^{(i+1)}) < f(\mathbf{x}^{(i)}))$  yeni nokta kabul görmüş olur. Aksi halde yeni yön arayışına gidilir ki bu da madde 2'ye tekrar dönüş ve sürecin tekrarı anlamına gelir.
- ▶ 5. Bu arada optimal adım büyüklüğü yerine sabit adım büyüklüğü tercihi yapılırsa, belirli bir tekrar işlemi sonrasında adım büyüklüğü  $\alpha_{t+1} = \alpha_t/2$  şeklinde küçültülebilir ve aynı işlemler grubu küçültülmüş yeni adım büyüklüğüne göre tekrar edilir,  $\alpha_{t+1} \leq \epsilon$  olması halinde ise tarama işlemi sonlandırılabilir.

Rastlantısal tarama yönteminin daha iyi anlaşılması için örnek bir eniyileme sorununu, belirtilen işlem adımlarını takip ederek çözebiliriz. Buna göre, daha önce de üzerinde durduğumuz üç yaylı bir kütle-yay düzeneğini düşünelim.



Şekil 3.11 Üç yay ve iki hareketli kütle-yay düzeneği.

Tanımlı kütle-yay düzeneğinin üzerine  $F$  kuvveti uygulandığında yer değiştirme değerlerini gösteren  $\mathbf{x}$  vektörünü, potansiyel enerjinin minimizasyonu süreci ile elde etmek mümkündür. Buna göre sisteme ait potansiyel enerji değerini, yaylardaki gerilme enerjisi ile yapılan iş arasındaki farka dayandırabiliriz;





$$f(\mathbf{x}) = \frac{1}{2}k_2x_1^2 + \frac{1}{2}k_3(x_2 - x_1)^2 + \frac{1}{2}k_1x_2^2 - Fx_2$$

Burada  $k_i$  yay katsayılarını,  $x_i$  yer değiştirme mesafelerini göstermektedir. Sorunun somutlaştırılması adına yay katsayı değerlerini;  $k_1 = 300 \text{ N/cm}$ ,  $k_2 = 200 \text{ N/cm}$ ,  $k_3 = 400 \text{ N/cm}$ , uygulanan kuvveti ise  $F = 2000 \text{ N}$  olarak dikkate alalım. Verilen değerler enerji denkleminde dikkate alınıp uygun düzenlemeler yapılırsa;

$$f(\mathbf{x}) = 100x_1^2 + 200(x_2 - x_1)^2 + 150x_2^2 - 2000x_2$$

denklemi elde edilir. Bu denklemi minimize eden  $(x_1, x_2)$  değerlerinin rastlantısal tarama yöntemi ile bulunabilmesi için öncelikle sürece ait başlangıç noktasını  $\mathbf{x}^{(0)} = [0 \ 0]^T$ , azami tekrar sayısını  $N = 100$ , başlangıç ve asgari adım büyüklüklerini;  $\alpha = 1$ ,  $\epsilon = 0.01$  şeklinde tanımlayabiliriz. Başlangıç noktasına ait amaç fonksiyonu değeri  $f(\mathbf{x}^{(0)}) = 0$  olarak tespit edilebilir. Sonraki adımı belirlemek üzere rassal olarak gidilecek yön tayin edilir. Buna göre  $-1 < r_i < 1$  ve  $(r_1^2 + r_2^2)^{1/2} \leq 1$  şartlarının geçerli olduğu uygun bir rassal sayı üretici kullanılarak;

$$\mathbf{r} = \begin{bmatrix} 0.2440 \\ -0.6309 \end{bmatrix}, \quad \mathbf{d} = \begin{bmatrix} 0.3581 \\ -0.9337 \end{bmatrix}$$

doğrultusu elde edilebilir. Gidilecek yönün belirlenmesi sonrasında  $\alpha$  adım büyüklüğünün tespiti gerekir. Bu uygulamada sabit adım büyüklüğü değeri dikkate alınabilir. Dolayısıyla bir sonraki adım ve bu adıma yönelik amaç fonksiyon değeri;

$$\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \alpha \mathbf{d}$$

$$\mathbf{x}^{(1)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} + 1 \begin{bmatrix} 0.3581 \\ -0.9337 \end{bmatrix} = \begin{bmatrix} 0.3581 \\ -0.9337 \end{bmatrix}$$

$$f(\mathbf{x}^{(1)}) = 2344.7$$

olarak hesaplanabilir. Hesaplama sonrasında yakınsama kontrolü dâhilinde  $\alpha = 1 > \epsilon = 0.01$  olması nedeni ile bir sonraki adıma geçilir ve devamında da karşılaştırmaya gidilir. Elde edilen yeni noktanın önceki noktaya göre daha iyi sonuç vermemesi nedeniyle;  $(f(\mathbf{x}^{(1)}) = 2344.7 > f(\mathbf{x}^{(0)}) = 0$ , yeni nokta kabul edilmez. Dolayısıyla bu durum madde 2'ye tekrar dönüş ve sürecin tekrarı anlamına gelir. Madde 2 kapsamında  $-1 < r_i < 1$  ve  $(r_1^2 + r_2^2)^{1/2} \leq 1$  şartlarının geçerli olduğu;

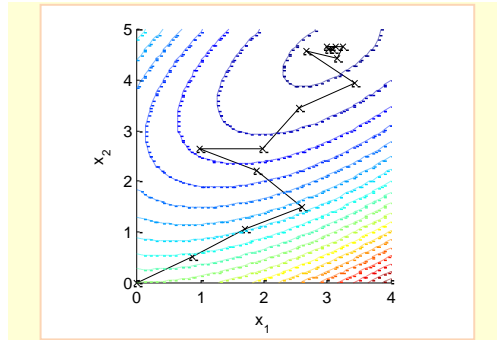


$$\mathbf{r} = \begin{bmatrix} 0.7785 \\ 0.4440 \end{bmatrix}, \quad \mathbf{d} = \begin{bmatrix} 0.86878 \\ 0.4954 \end{bmatrix}$$

yeni doğrultu elde edilebilir. Bir sonraki adım ve bu adıma yönelik amaç fonksiyon değeri;

$$\begin{aligned} \mathbf{x}^{(1)} &= \mathbf{x}^{(0)} + \alpha \mathbf{d} \\ \mathbf{x}^{(1)} &= \begin{bmatrix} 0 \\ 0 \end{bmatrix} + 1 \begin{bmatrix} 0.86878 \\ 0.4954 \end{bmatrix} = \begin{bmatrix} 0.86878 \\ 0.4954 \end{bmatrix} \\ f(\mathbf{x}^{(1)}) &= -850.61 \end{aligned}$$

olarak hesaplanabilir. Elde edilen yeni noktanın önceki noktaya göre daha iyi sonuç vermesi nedeniyle;  $f(\mathbf{x}^{(1)}) = -850.61 < f(\mathbf{x}^{(0)}) = 0$ , yeni nokta olarak elde edilen konum kabul edilebilir. Benzer işlemler iterasyon sayısının azami tekrar sayısı  $N = 100$  değerine ulaşması halinde sonlandırılır ve adım büyüklüğü olarak  $\alpha_1 = \alpha_0/2 = 1/2 = 0.5$  değeri dikkate alınarak tekrar ilave 100 adım atılır. Bu süreç,  $\alpha_{t+1} \leq \epsilon$  şartı sağlanana kadar devam eder. Nitekim ilk 100 adım sonunda en uygun nokta  $[2.6718 \ 4.5831]$  ve amaç fonksiyon değeri  $-4571$  olarak, ikinci 100 adım sonunda en uygun nokta  $[3.1497 \ 4.4358]$  ve amaç fonksiyon değeri  $-4597$  olarak elde edilebilir. Bu uygulamada 7nci tur sonunda  $\alpha_7 = 0.0078 \leq \epsilon = 0.01$  şartı sağlanmış ve neticede  $[3.0820 \ 4.6198]$  optimal noktası ile yaklaşık olarak  $-4615.4$  amaç fonksiyon değerine ulaşılmıştır. Diğer taraftan sorunun analitik çözümünün ise  $[3.077 \ 4.615]$  konumu ve  $-4615.4$  amaç fonksiyon değeri şeklinde olduğunu belirtelim. Şekil 3.12'de söz konusu taramaya ait optimal nokta süreci görülmektedir. Rastlantısal tarama yönteminin tamamen rassal sayılara bağlı olması nedeniyle her çözüm sürecinde farklı doğrultu ve konumlara ulaşılır. Dolayısıyla, aynı işlemlerin tekrar edilmesi ile aynı sonuçlara ulaşılması mümkün görünmemektedir. Bununla beraber birbirine yakın optimal nokta değerlerine ulaşılması, yöntemin yakınsama özelliğini göstermektedir.



Şekil 3.12 Rastlantısal tarama neticesinde daha iyi konum olarak belirlenen noktalar.



## “Kütle-yay düzeneği”

MATLAB uygulaması

```
clc; clear;

N = 100; % azami tekrar sayısı
s = 1; % başlangıç adım büyüklüğü
eps = 0.01; % adım toleransı
x = [0;0]; % başlangıç noktası
k1 = 300; k2 = 200; k3 = 400; P = 2000; % düzenek değerleri
f = 0.5*k2*x(1)^2 + 0.5*k3*(x(2)-x(1))^2+0.5*k1*x(2)^2-P*x(2); % amaç fonksiyonu
xmin(:,1) = x; fmin = f; k = 1;

for t =1:10;
    for i=1:N;
        for j=1:10
            r = -1 + 2* rand(2,1); R = norm(r,'fro');
            if R<=1
                break;
            end;
            end
            u = 1/R*r; xg =xmin(:,k) + s * u;
            fg = 0.5*k2*xg(1,1)^2 + 0.5*k3*(xg(2,1)-xg(1,1))^2+...
                0.5*k1*xg(2,1)^2-P*xg(2,1);
            if fg<fmin
                k = k + 1; fmin = fg; xmin(:,k) = xg;
            end;
        end
        s = s/2;
        if s<eps
            break
        end
    end
end

xmin(:,k)
fmin
```



Sonuç

```
ans =
    3.0811
    4.6182

fmin =
-4.6154e+03
```





### 3.6 Benzetimli tavlama yöntemi

Benzetimli tavlama<sup>i</sup> yöntemi, literatüre S. Kirkpatrick ve arkadaşları tarafından kazandırılan ve esinlenme kaynağı olarak metal bilimindeki ısı işlemleri dikkate alan bir yöntemdir. 1983 tarihli “Benzetimli tavlama ile optimizasyon<sup>ii</sup>” makalesi ile gündeme gelen yöntemin dikkate aldığı önemli bir diğer yaklaşım ise 1953 tarihli Metropolis-Hastings algoritmasıdır<sup>(39)</sup>. N. Metropolis ve arkadaşlarının yayımladığı “Hızlı hesaplama makineleri ile hal denklemlerinin hesaplanması” başlıklı makalede ana gaye, rastsal olarak gezinen moleküllerden hareketle sisteme ait basınç ve sıcaklık gibi hal özelliklerinin örnekleme yaklaşımı ile hesaplanmasıydı. Bir anlamda Monte Carlo benzetimi yaklaşımını içeren yöntem, termodinamik sistem dâhilindeki tüm moleküllerin dikkate alınması yerine ( $cm^3$  dâhilinde  $10^{23}$  mertebesinde atom sayısı) rastgele seçilen belirli sayıdaki molekülün dikkate alınmasını öngörmektedir<sup>(40)</sup>. Basit bir yaklaşımla iki boyutlu ortamda  $N$  tane molekülün, belirli bir örgü ağı içerisinde dizildiği kabul edilen yöntemde, her bir molekülün bir sonraki konumu;

$$x \rightarrow x + \alpha \varepsilon_1 \quad (3.22)$$

$$y \rightarrow y + \alpha \varepsilon_2 \quad (3.23)$$

denklemleri ile belirlenmektedir. Burada  $\alpha$ , molekülün azami yer değiştirme mesafesini,  $\varepsilon_1$  ve  $\varepsilon_2$  terimleri ise değerleri  $-1$  ve  $+1$  arasında değişebilen rastsal sayıları ifade etmektedir. Rastsal hareket sonrası molekül,  $2\alpha$  kenar uzunluğuna sahip bir karenin içerisinde herhangi bir konumda olabilmektedir. Hareket sonrası enerji değişimi hesaplanmakta ve  $\Delta E < 0, (E_2 < E_1)$  durumunun geçerli olması halinde öngörülen yeni konum kabul edilmektedir. Zira böyle bir durum, enerji seviyesinin daha düşük olması ve sistemde kararlılığın artması anlamına gelmektedir. Diğer taraftan  $\Delta E > 0, (E_2 > E_1)$  durumunda, moleküler hareketin reddedilmesi yerine ona bir şans verilmektedir. Buna göre 0 ile 1 arasında ilave olarak üretilen rastsal  $\varepsilon_3$  sayısının;

$$\varepsilon_3 < e^{-\frac{\Delta E}{kT}} \quad (3.24)$$

şartını sağlaması halinde, daha düşük bir enerji seviyesine geçilirse dahi molekülün bir sonraki konumu kabul edilmektedir. Burada  $k$  terimi Boltzmann sabitini,  $T$  sıcaklığı temsil etmektedir. Koşulun sağlanmaması halinde yeni konum reddedilmekte, aynı adım farklı rastsal sayılarla



**Şekil 3.13** Isıl işlem uygulamaları, metalin kristal yapısını daha düzenli hale getirmektedir.

<sup>i</sup> Simulated annealing

<sup>ii</sup> “Optimization by simulated annealing”



tekrarlanmaktadır. Konum değişikliklerinin kabul edildiği yeni atomik konumlandırma dikkate alınarak sistemin hal özelliklerinden herhangi biri ( $p$ ), ortalama alınarak elde edilebilmektedir;

$$\bar{p} = \frac{1}{N} \sum_{i=1}^N p_i \quad (3.25)$$

Metropolis yaklaşımının en önemlisi getirisi, sürecin tamamen rassal bir karakterde olması yerine belirli bir olasılık çerçevesinde şekillendirilmesi ve bunun da sıcaklık ve enerji seviyesi gibi fiziksel bazı parametrelere bağlı hale getirilmesidir. Diğer taraftan hareketin kabulü meyanında kabul veya ret gibi kesin çizgilerin de belirli olasılıklar çerçevesinde yumuşatıldığı ve iyi bir sonuç doğurmasa da bazı hareketlere fırsat verildiği görülmektedir<sup>(41)</sup>.

İstatistiksel mekanik ile ilgili söz konusu çalışma, yıllar sonra Kirkpatrick ve arkadaşlarına ilham kaynağı olmuş ve belirli bir sıcaklıktaki ısı dengeye sahip sistemlerin davranış özelliklerinin çok değişkenli eniyileme süreçlerinde kullanılabileceği görülmüştür. Metalürjik çalışmalara göre metal malzeme dâhilinde düzenli ve yapısal hataları azalmış kristal yapısının oluşabilmesi için uygun bir tavlama sürecine ihtiyaç duyulmaktadır. Tavlama sürecinde malzeme ısı işleme tabi tutularak önce erime noktasına kadar ısıtılır, akabinde yavaşça soğutulularak kristal yapının kontrollü olarak gelişmesi sağlanır. Bu sayede mikro seviyede atomların kristal yapıda uygun konumları alması ve hatalı konumlardan kurtulmaları sağlanmış olur<sup>(42)</sup>. İfade edilen tavlama sürecinin eniyileme sürecine uyarlanması ise aşağıdaki işlem adımlarına göre yapılabilir<sup>(43)</sup>;

- 1. Başlangıç noktasının belirlenmesi:

$$\mathbf{x} = \mathbf{x}^{(0)}$$

- 2. Tavlama işlemi için başlangıç sıcaklığının belirlenmesi:

$$T_i = T^{(0)}$$

- 3. Tavlama süresinin belirlenmesi:

$$N_i$$

- 4. Tavlama süresi boyunca aşağıdaki tekrarlı işlemlerin yapılması;  $j = 1 \rightarrow N_i$ :

- a. Eldeki çözüm noktasından hareketle yerel komşulukta rassal olarak yeni çözüm noktasının elde edilmesi:

$$\mathbf{x}' \in N(\mathbf{x})$$





- b. Eldeki çözüm ile üretilen yeni çözüm noktasına ait amaç fonksiyon değerlerinin karşılaştırılması:

$$\Delta f = f(x') - f(x)$$

- (1) Yakınsama kontrolünün yapılması, istenen yakınsama ölçütünün karşılanmaması halinde bir sonraki adıma geçilmesi.
- (2)  $\Delta f \leq 0$  olması durumunda yeni çözüm noktasının eldeki çözüm olarak kabul edilmesi:

$$x = x'$$

- (3) Aksi halde  $P_a$  kabul olasılık değerinin hesaplanması ve üretilen  $r$  rassal sayısı ile karşılaştırma yapılması:

$$P_a = e^{-\frac{\Delta f}{T_i}}$$
$$r = rand(0,1)$$

- (4)  $r < P_a$  olması durumunda yeni çözüm noktasının eldeki çözüm olarak kabul edilmesi:

$$x = x'$$

- (5) Aksi halde 4.a işlem adımına dönülmesi.

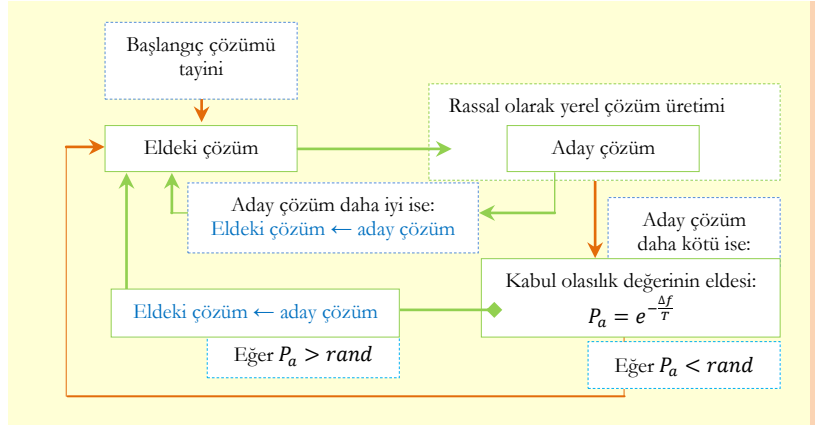
- 5. Tavlama süresi ile sıcaklığının yeniden belirlenmesi ve 4.üncü madde işlemlerinin tekrar edilmesi:

$$N_{i+1} = f(N_i)$$
$$T_{i+1} = f(T_i)$$

Benzetimli tavlama işlem adımları Akış diyagramı 3.2 dâhilinde resmedilmektedir.

Tanımlanan işlem adımlarından da görüldüğü üzere benzetimli tavlama yönteminde tavlama işlemi, çözüm arayışında kullanılan süreç olup bu süreçte başlangıç sıcaklığı, o sıcaklıktaki tekrarlı işlem sayısı ve akabinde sıcaklığın yeniden belirlenmesi çerçevesinde soğuma şekli önemli etmenlerdir. Bu etmenlerin değerleri, üzerinde çalışılan soruna göre belirlenir. Bununla beraber literatürde sıkça kullanılan bazı yaklaşımların burada ifade edilmesinde fayda mülhaza edilmektedir.





**Akış diyagramı 3.2** Benzetimli tavlama akış diyagramı.

Örneğin tavlama sürecindeki belirli sıcaklık değerinde üretilecek yeni çözüm sayısını ve dolayısıyla o sıcaklıktaki taramanın büyüklüğünü belirleyen tavlama süresini;

$$N_i = c; \quad (3.26)$$

şeklinde sabitleyerek,

$$N_{i+1} = N_i + c \quad (3.27)$$

biçiminde aritmetik olarak arttırarak,

$$N_{i+1} = \frac{N_i}{c}, \quad c < 1 \quad (3.28)$$

şeklinde geometrik olarak büyütürük,

$$N_{i+1} = \frac{c}{\log T_i} \quad (3.29)$$

biçiminde sıcaklığın düşürülmesine paralel olarak logaritmik ölçekte arttırarak,

$$N_{i+1} = N_i^c, \quad c < 1 \quad (3.30)$$



biçiminde üssel yaklaşımla yükselterek dikkate almak mümkündür. Tavlama sürecinde sıcaklığın soğuma evresinde ise benzer yaklaşımlar sergilenebilir. Örneğin tavlama sıcaklığını;

$$T_{i+1} = T_i - c \quad (3.31)$$

şeklinde aritmetik olarak azaltmak,

$$\begin{aligned} T_{i+1} &= T_i c \\ c &< 1 \\ c &\sim 1 \end{aligned} \quad (3.32)$$

biçiminde geometrik olarak küçültmek,

$$T_{i+1} = \frac{c}{1+i} \quad (3.33)$$

şeklinde iterasyon sayısına bağlı olarak azaltmak,

$$T_{i+1} = \frac{c}{\log(1+i)} \quad (3.34)$$

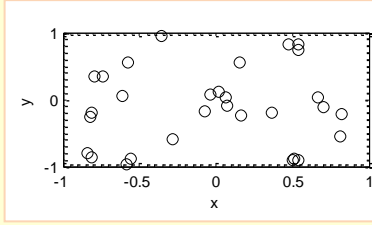
denklemini dikkate alarak logaritmik ölçekte azaltmak mümkündür<sup>(44)</sup>.

Tavlama sıcaklığının azaltımının etkisi, yeni üretilen ancak daha iyi bir sonuç vermeyen çözüm noktalarının değerlendirilmesinde görülebilir. Zira kabul olasılık değerinin hesaplanmasında sıcaklık doğrudan etki etmekte olup sıcaklığın azalması ile beraber daha kötü tasarım noktalarının çözüm olarak kabul görme şansı da azalmaktadır.

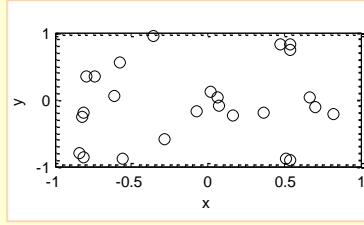
Şekil 3.14'te rassal olarak üretilen 30 noktanın,  $\Delta f = 10$  ve 3 farklı sıcaklık değeri baz alınarak rassal sayı karşılaştırması neticesinde kabul görmüş noktaları görülmektedir. Buna göre sıcaklığın yüksek olduğu ilk durumda kabul olasılık değeri 0.9048 olmakta ve bu değere göre kabul edilen nokta sayısı 25 olarak görülmektedir. Sıcaklığın onda bir oranında düşürülmesi neticesinde kabul olasılık değeri 0.3679 değerine düşmekte ve kabul edilen nokta sayısı 13 olmaktadır. Son örnekte ise sıcaklık mevcut değere göre yarı yarıya düşürülmekte, buna mukabil kabul olasılık değeri 1.1353 olmakta ve kabul edilen nokta sayısı da 4'e inmektedir.



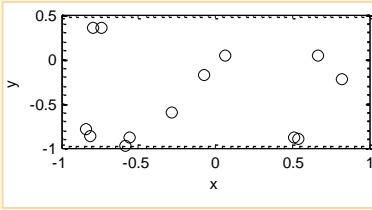




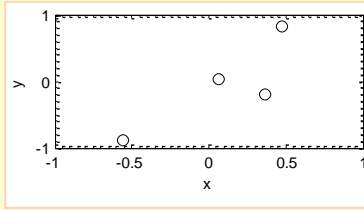
(a) Rassal olarak üretilen 30 nokta



(b)  $T = 100$ ,  $P_a = 0.9048$  değerlerine göre rassal olarak kabul gören 25 nokta



(c)  $T = 10$ ,  $P_a = 0.3679$  değerlerine göre rassal olarak kabul gören 13 nokta



(ç)  $T = 5$ ,  $P_a = 0.1353$  değerlerine göre rassal olarak kabul gören 4 nokta

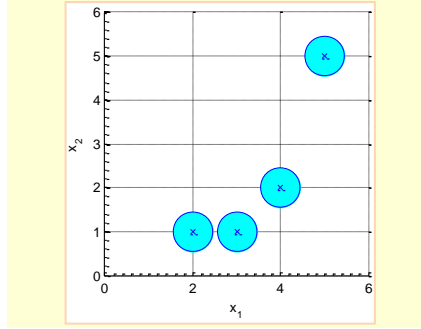
**Şekil 3.14** Kabul olasılık değerine sıcaklığın etkisi ve kabul edilen nokta dağılımı.

Benzetimli tavlama yönteminde yakınsama kontrolü adına değişik ölçütleri dikkate almak mümkündür. Genellikle tavlama sıcaklığının belli bir değerin altına inmesi, belli sayıda yapılan iterasyona rağmen daha iyi bir nokta elde edilememesi, azami iterasyon sayısına ulaşılması veya üretilen çözüm noktası sayısına göre elde edilen daha iyi nokta sayısının oranının belirlenen bir orana göre düşük olması gibi ölçütler yakınsamada dikkate alınabilmektedir. Benzetimli tavlama yönteminin daha iyi anlaşılması için Şekil 3.15'teki gibi bir eniyileme sorununu, belirtilen işlem adımlarını takip ederek çözebiliriz. Buna göre iki boyutlu bir örgü ağı içerisinde konumlandırılması istenen  $n = 4$  adet atomu dikkate alalım. Söz konusu dört atomun her birinin konumlanabileceği olası konum indisleri  $i = 1, 2, \dots, 5$  ve  $j = 1, 2, \dots, 5$  arası değişebilmekte olup buna göre toplam konum sayısı  $p = 25$ 'tir. Dizilişte ana gaye, konuma göre elde edilebilecek enerji seviyesinin minimize edilmesidir. Diziliş yapılandırmasına göre herhangi bir atoma ait enerji seviyesi potansiyel ve kinetik enerji değerlerinin toplamı şeklinde;



$$E_i = mgx_{i,2} + \sum_{i \neq j}^n \sqrt{(x_{i,1} - x_{j,1})^2 + (x_{i,2} - x_{j,2})^2}$$

ifadesine göre hesaplanabilmektedir.



Şekil 3.15 Örnek atom dizilişi.

Denklemden  $m$  kütle değerini,  $g$  yerçekimi ivmesini,  $(x_{i,1}, x_{i,2})$  ise  $i$  atomuna ait koordinat verilerini göstermektedir. Diziliş konfigürasyonuna göre sahip olunan enerji seviyesi;

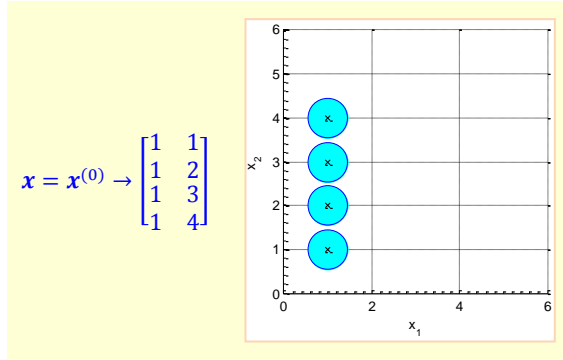
$$f = \sum_{i=1}^n E_i$$

denkleminde göre atomların sahip olduğu enerjilerin toplamı olarak ifade edilmektedir. Dört atomun 25 konuma göre beraberce oluşturabileceği olası diziliş konfigürasyonu sayısı ise;

$$\frac{p!}{(p-n)!} = \frac{25!}{(25-4)!} = 6,375,600$$

değerindedir. Her ne kadar aynı konuma birden fazla atomun yerleştirilemeyeceği koşulu dikkate alınsa da, görüldüğü üzere diziliş konfigürasyonu olasılıklarının her birinin denenerken minimum enerji bazlı konfigürasyona ulaşılması yöntemi pratik olarak oldukça zahmetli bir işittir. Sorunun benzetimli tavlama yöntemi ile çözümü sadedinde yapılacak ilk işlem, başlangıç dizilişinin belirlenmesidir. Genellikle rassal olarak belirlenen başlangıç çözümü burada belirli bir sistematığe göre müteakip şekildeki gibi belirlenebilir;





Bu ifadeye satırlar her bir atoma ait koordinat verilerini içermektedir. Başlangıç noktası tayini sonrası başlangıç sıcaklığı  $T_1 = 100$  ve tavlama süresi  $N = 10$  şeklinde öngörülebilir. Diğer taraftan azami yakınsama ölçütü olarak ardı ardına beş tekrarlı işlem sonucunda amaç fonksiyonunda herhangi bir değişim olmaması dikkate alınabilir. Tavlama süresi boyunca müteakip tekrarlı işlemlerin icrası gerçekleştirilir;  $j = 1 \rightarrow 10$ . Bu meyanda ilk işlem, eldeki çözüm noktasından hareketle yerel komşulukta rassal olarak yeni çözüm noktasının elde edilmesidir;  $\mathbf{x}' \in N(\mathbf{x})$ . Örneğin rastgele herhangi bir atom seçilerek yine rastgele konum belirlenebilir;

$\text{randint}^i(1,1,[1\ 4]) \rightarrow 2$   
 $\text{randint}(1,2,[1\ 6]) \rightarrow [2\ 1]$

$$\mathbf{x}' = \begin{bmatrix} 1 & 1 \\ 2 & 1 \\ 1 & 3 \\ 1 & 4 \end{bmatrix}$$

Müteakiben eldeki çözüm ile üretilen yeni çözüm noktasına ait amaç fonksiyon değerlerinin karşılaştırılması işlemine geçilir;

$$\begin{aligned} f(\mathbf{x}) &= 1 \times 10 \times 1 + \sqrt{(1-1)^2 + (1-2)^2} + \sqrt{(1-1)^2 + (1-3)^2} + \sqrt{(1-1)^2 + (1-4)^2} \\ &+ 1 \times 10 \times 2 + \sqrt{(1-1)^2 + (2-1)^2} + \sqrt{(1-1)^2 + (2-3)^2} + \sqrt{(1-1)^2 + (2-4)^2} \\ &+ 1 \times 10 \times 3 + \sqrt{(1-1)^2 + (3-1)^2} + \sqrt{(1-1)^2 + (3-2)^2} + \sqrt{(1-1)^2 + (3-4)^2} \\ &+ 1 \times 10 \times 4 + \sqrt{(1-1)^2 + (4-1)^2} + \sqrt{(1-1)^2 + (4-2)^2} + \sqrt{(1-1)^2 + (4-3)^2} = 120 \end{aligned}$$

<sup>i</sup> Matlab dâhilinde rastsal olarak tam sayı üreten komut: “randint”.



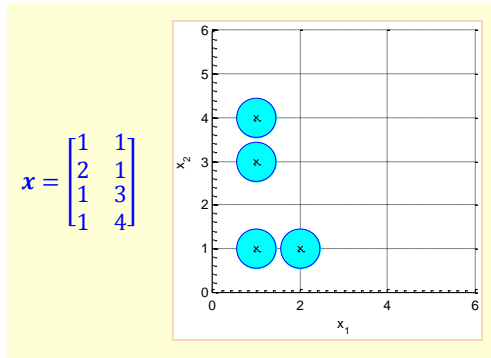


$$\begin{aligned}
 f(\mathbf{x}') &= 1 \times 10 \times 1 + \sqrt{(1-2)^2 + (1-1)^2} + \sqrt{(1-1)^2 + (1-3)^2} + \sqrt{(1-1)^2 + (1-4)^2} \\
 &+ 1 \times 10 \times 1 + \sqrt{(2-1)^2 + (1-1)^2} + \sqrt{(2-1)^2 + (1-3)^2} + \sqrt{(2-1)^2 + (1-4)^2} \\
 &+ 1 \times 10 \times 3 + \sqrt{(1-1)^2 + (3-1)^2} + \sqrt{(1-2)^2 + (3-1)^2} + \sqrt{(1-1)^2 + (3-4)^2} \\
 &+ 1 \times 10 \times 4 + \sqrt{(1-1)^2 + (4-1)^2} + \sqrt{(1-2)^2 + (4-1)^2} + \sqrt{(1-1)^2 + (4-3)^2} = 114.8
 \end{aligned}$$

$$\Delta f = f(\mathbf{x}') - f(\mathbf{x})$$

$$\Delta f = 114.8 - 120 = -5.2$$

Yapılan karşılaştırma neticesinde  $\Delta f \leq 0$  olması nedeniyle yeni çözüm noktası, eldeki çözüm olarak kabul edilir;



Bir sonraki aşamada, eldeki yeni çözüm noktasından hareketle yerel komşulukta rassal olarak yeni çözüm noktası elde edilir;

$$\begin{aligned}
 \text{randint}(1,1,[1\ 4]) &\rightarrow 4 \\
 \text{randint}(1,2,[1\ 6]) &\rightarrow [5\ 6]
 \end{aligned}$$

$$\mathbf{x}' = \begin{bmatrix} 1 & 1 \\ 2 & 1 \\ 1 & 3 \\ 5 & 6 \end{bmatrix}$$

Müteakiben eldeki çözüm ile üretilen yeni çözüm noktasına ait amaç fonksiyon değerlerinin karşılaştırılması işlemine geçilir;



$$f(\mathbf{x}) = 114.8$$

$$\begin{aligned} f(\mathbf{x}') &= 1 \times 10 \times 1 + \sqrt{(1-2)^2 + (1-1)^2} + \sqrt{(1-1)^2 + (1-3)^2} + \sqrt{(1-5)^2 + (1-6)^2} \\ &+ 1 \times 10 \times 1 + \sqrt{(2-1)^2 + (1-1)^2} + \sqrt{(2-1)^2 + (1-3)^2} + \sqrt{(2-5)^2 + (1-6)^2} \\ &+ 1 \times 10 \times 3 + \sqrt{(1-1)^2 + (3-1)^2} + \sqrt{(1-2)^2 + (3-1)^2} + \sqrt{(1-5)^2 + (3-6)^2} \\ &+ 1 \times 10 \times 6 + \sqrt{(5-1)^2 + (6-1)^2} + \sqrt{(5-2)^2 + (6-1)^2} + \sqrt{(5-1)^2 + (6-3)^2} = 154.9 \end{aligned}$$

$$\Delta f = 154.9 - 114.8 = 40.1$$

Yapılan karşılaştırma neticesinde  $\Delta f > 0$  olması nedeniyle  $P_a$  kabul olasılık değerinin hesaplanması ve üretilen  $r$  rassal sayısı ile karşılaştırma yapılması işlemlerine geçilir;

$$P_a = e^{-\frac{\Delta f}{T_1}} = e^{-\frac{40.1}{100}} = 0.6696$$

$$r = \text{rand}(0,1) = 0.6557$$

Buna göre  $r < P_a$  olması nedeniyle yeni çözüm noktası, eldeki çözüm noktası olarak kabul edilir ve yeni çözüm noktasından hareketle yerel komşulukta rassal olarak yeni çözüm noktası elde edilir;

$\text{randint}(1,1,[1\ 4]) \rightarrow 3$   
 $\text{randint}(1,2,[1\ 6]) \rightarrow [3\ 6]$

$$\mathbf{x}' = \begin{bmatrix} 1 & 1 \\ 2 & 1 \\ 3 & 6 \\ 5 & 6 \end{bmatrix}$$

Eldeki çözüm ile üretilen yeni çözüm noktasına ait amaç fonksiyon değerlerinin karşılaştırılması işlemine geçilir;

$$f(\mathbf{x}) = 154.9$$

$$\begin{aligned} f(\mathbf{x}') &= 1 \times 10 \times 1 + \sqrt{(1-2)^2 + (1-1)^2} + \sqrt{(1-3)^2 + (1-6)^2} + \sqrt{(1-5)^2 + (1-6)^2} \\ &+ 1 \times 10 \times 1 + \sqrt{(2-1)^2 + (1-1)^2} + \sqrt{(2-3)^2 + (1-6)^2} + \sqrt{(2-5)^2 + (1-6)^2} \\ &+ 1 \times 10 \times 6 + \sqrt{(3-1)^2 + (6-1)^2} + \sqrt{(3-2)^2 + (6-1)^2} + \sqrt{(3-5)^2 + (6-6)^2} \\ &+ 1 \times 10 \times 6 + \sqrt{(5-1)^2 + (6-1)^2} + \sqrt{(5-2)^2 + (6-1)^2} + \sqrt{(5-3)^2 + (6-6)^2} = 191.4 \end{aligned}$$

$$\Delta f = 191.4 - 154.9 = 36.5$$





Yapılan karşılaştırma neticesinde, bir önceki adımda olduğu üzere  $\Delta f > 0$  olması nedeniyle  $P_a$  kabul olasılık değerinin hesaplanması ve üretilen  $r$  rassal sayısı ile karşılaştırma yapılması işlemlerine geçilir;

$$P_a = e^{-\frac{\Delta f}{T_1}} = e^{-\frac{36.5}{100}} = 0.6942$$

$$r = rand(0,1) = 0.7244$$

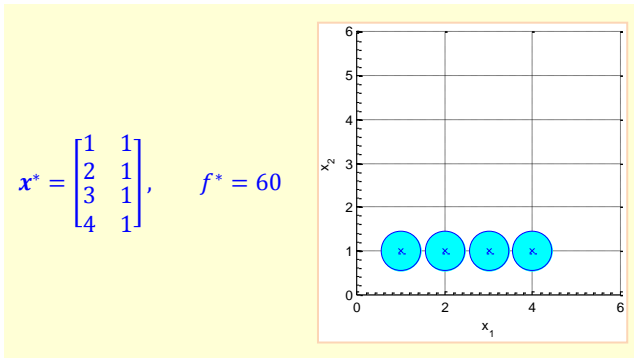
Rastlantı eseri  $r > P_a$  olması nedeniyle üretilen ve maalesef daha kötü bir sonuç doğuran aday çözüm kabul edilmez ve eldeki çözüm noktasından hareketle yerel komşulukta rassal olarak yeni çözüm noktası elde edilir. Bu işlemlere aynı sıcaklık değeri ile onuncu tekrara kadar devam edilir. Tavlama süresinin tamamlanmasını müteakip yeni tavlama sıcaklığı belirlenir. Bu uygulamada  $T_{i+1} = T_i c$ ;  $c = 0.95$  yaklaşımı dikkate alınmaktadır. Dolayısıyla yeni tavlama sıcaklığı;

$$T_2 = 0.95T_1 = 95$$

olarak elde edilebilir. Aynı işlem adımında tespit edilecek diğer bir husus ise tavlama süresi olup bu uygulamada tavlama süresi sabit olarak dikkate alınmaktadır;

$$N_i = c, \quad c = 10$$

Yeni tavlama sıcaklığının baz alındığı tavlama işlemleri benzer adımlarla yakınsama ölçütü sağlanana kadar devam ettirilir. Aşağıdaki şekilde elde edilen optimal çözüm ve amaç fonksiyonu değeri görülmektedir<sup>(45)</sup>;



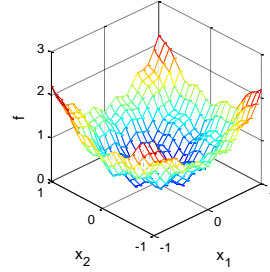


### Sinüzoidal yüzey

$$f(x) = 0.2 + x_1^2 + x_2^2 - 0.1 \cos 5\pi x_1 - 0.1 \cos 4\pi x_2$$



**MATLAB uygulaması**



Komut<sup>(46)</sup>

```
clc; clear;
% Başlangıç noktası
x_start = [0.8, -0.5];
%% Eşdeğer çizgilerinin eldesi
i1 = -1.0:0.01:1.0; i2 = -1.0:0.01:1.0; [x1m, x2m] = meshgrid(i1, i2);
fm = 0.2 + x1m.^2 + x2m.^2 - 0.1*cos(5.0*pi*x1m) - 0.1*cos(4.0*pi*x2m);
fig = figure(1); [C,h] = contour(x1m,x2m,fm); clabel(C,h,'Labelspacing',250);
title('Benzetimli Tavlama'); xlabel('x1'); ylabel('x2'); hold on;
% Tavlama süreci tekrarı
n = 50;
% Tavlama süresi
m = 50;
% kabul edilen çözüm sayısı
na = 0.0;
% Başlangıç noktasının kabul edilmesinde dikkate alınacak olasılık değeri
p1 = 0.7;
% Son noktanın kabul edilmesinde dikkate alınacak olasılık değeri
p50 = 0.001;
% Başlangıç sıcaklığı
t1 = -1.0/log(p1);
% Son sıcaklık
t50 = -1.0/log(p50);
% Her tavlama süreci sonunda sıcaklıktaki azalma oranı
frac = (t50/t1)^(1.0/(n-1.0));
x = zeros(n+1,2); x(1,:) = x_start; xi = x_start; na = na + 1.0;
% Eldeki en iyi sonuç
xc = x(1,:); fc = f(xi); fs = zeros(n+1,1); fs(1,:) = fc;
% Eldeki sıcaklık
t = t1;
```



```

% Ortalama enerji deęeri
DeltaE_avg = 0.0;
for i=1:n
    disp(['Cycle: ',num2str(i),' with Temperature: ',num2str(t)])
    for j=1:m
        % yeni nokta tayini
        xi(1) = xc(1) + rand() - 0.5; xi(2) = xc(2) + rand() - 0.5;
        % limit deęer aralıęı
        xi(1) = max(min(xi(1),1.0),-1.0); xi(2) = max(min(xi(2),1.0),-1.0);
        DeltaE = abs(f(xi)-fc);
        if (f(xi)>fc)
            if (i==1 && j==1)
                DeltaE_avg = DeltaE;
            end
            p = exp(-DeltaE/(DeltaE_avg * t));
            if (rand()<p)
                accept = true;
            else
                accept = false;
            end
        else
            accept = true;
        end
        if (accept==true)
            xc(1) = xi(1); xc(2) = xi(2);
            fc = f(xc); na = na + 1.0;
            DeltaE_avg = (DeltaE_avg * (na-1.0) + DeltaE) / na;
        end
    end
    x(i+1,1) = xc(1); x(i+1,2) = xc(2); fs(i+1) = fc;
    % Tavlama sıcaklıęının dűşürűlmesi
    t = frac * t;
end
% Sonuęların yazdırılması iřlemleri
disp([Eniyi cozum: ',num2str(xc)]); disp(['Eniyi amac fonksiyonu: ',num2str(fc)]);
plot(x(:,1),x(:,2),'ro'); saveas(fig,'contour','png')
fig = figure(2); subplot(2,1,1); plot(fs,'r.-'); legend('amac'); subplot(2,1,2);
hold on; plot(x(:,1),'b.-'); plot(x(:,2),'g.-'); legend('x_1','x_2')

%% Amac fonksiyonu

function obj = f(x)
x1 = x(1); x2 = x(2);
obj = 0.2 + x1^2 + x2^2 - 0.1*cos(5.0*pi*x1) - 0.1*cos(4.0*pi*x2);
end

```

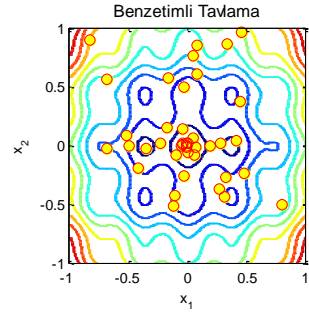




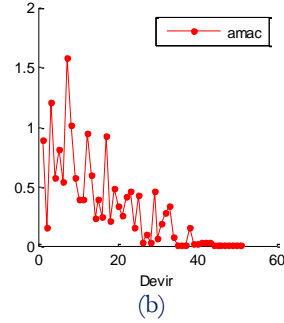
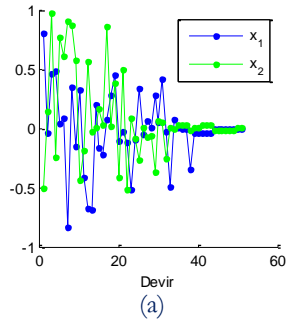


## Sonuç

Eniyi cozum: -0.0073923 0.0025357  
Eniyi amac fonksiyonu: 0.00078526



Şekil 3.16 Amaç fonksiyonu eşdeğer çizgileri ve tarama esnasında uğranılan konumlar.



Şekil 3.17 (a)Tavlama süreçlerinde tasarım yöneyi değişikliği ve (b) amaç fonksiyonu değişimi.



### 3.7 Karınca kolonisi algoritması

Karınca kolonisi algoritması yöntemi, literatüre M. Dorigo (1961- ) tarafından kazandırılan ve esinlenme kaynağı olarak doğadaki karıncaları dikkate alan bir yöntemdir<sup>(47)</sup>. 1992 tarihli “Optimizasyon, öğrenme ve doğal algoritmalar<sup>ii</sup>” başlıklı doktora tezi ile gündeme gelen yaklaşımın kökenini 1950’li yıllara kadar götürmek mümkündür. Özellikle hayvan davranışlarını inceleyen bilim insanları, geçmiş dönemde karıncalar üzerinde çalışmalar yapmakta, zekâ sahibi olmayan bu canlıların besin ile yuva arasındaki en kısa mesafeyi nasıl bulduklarını anlamaya çalışmaktaydılar. O yıllardaki biyoloji literatürüne bakıldığında koloniler halinde yaşayan canlıların ve özellikle de termitlerin davranışlarını inceleyen makaleler görülmektedir. Örneğin 1959 tarihli P. P. Grasse imzalı bir makalede, termitlerin davranışları incelenmekte ve sosyal varlık olarak yaşayan termit kolonilerinin kendi kendilerine organize olabildikleri ifade edilmekteydi. Grasse’ye göre söz konusu organizasyon sürecinin temelinde, dolaylı iletişim çeşitlerinden biri olan işaret bazlı hareket<sup>iii</sup> kavramı yatmaktaydı. Buna göre termitler doğrudan temas yanında birbirlerini görmeden, sadece çevrelerine bıraktıkları işaretlerle de haberleşebilmekte ve organize olabilmekteydiler<sup>(48)</sup>. Benzer şekilde 1980’li yıllarda termitlere benzer yaşam sürdüren karıncaların davranışlarını inceleyen çok sayıda makaleye rastlanmaktadır. Özellikle Denebourg ve meslektaşlarının karıncaların olasılık bazlı davranışları üzerine yazdığı makalelerin Dorigo ve meslektaşlarınınca dikkate alındığı görülmektedir<sup>(49)</sup>.

Karıncalar, günümüzden yaklaşık 110 veya 130 milyon yıl öncesi zaman diliminde yeryüzünde görülmeye başlanan ve sosyal böcek türleri arasındaki “Formicidae” ailesine üye böcek türlerindedir. Genellikle koloniler halinde yaşayan bu canlıların koloni büyüklüğü birkaç düzine karıncadan oluşabileceği gibi milyonlarca bireyden de oluşabilir. Karınca kolonileri adeta bir süper organizma niteliğinde olup çok sayıda birey barındırmasına rağmen tekil bir yapı gibi hareket edebilmekte ve koordinasyon içinde çalışabilmektedirler. Karınca toplumlarında iş bölümü, bireyler arasında iletişim ve bu sayede de karmaşık sorunları çözme yetenekleri söz konusudur. Bireyler arası iletişim dokunma, ses veya feromen<sup>iv</sup> denilen kimyasal sinyaller vasıtası ile gerçekleşir. Daha önce işaret-bazlı-hareket kavramı ile ifade edilen iletişim yönteminin temelinde karıncaların hareket ettikleri ortamlara bıraktıkları feromenler önemli rol oynar. Gruplar halinde rassal olarak besin arayan karıncalar, herhangi bir yiyecek karşlaştıklarında,



Şekil 3.18 Doğada karınca.

<sup>i</sup> Ant colony optimization

<sup>ii</sup> “Optimization, learning, and natural algorithms”

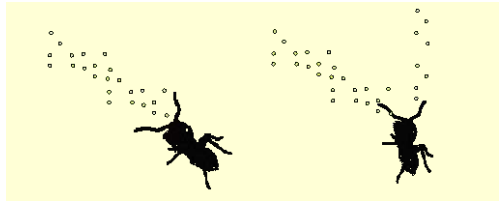
<sup>iii</sup> “Stigmergy” kelimesi Yunanca “stigma” ve “ergon” kelimelerinden türetilmiştir. Terimi oluşturan ilk sözcüğünün anlamı işaret, ikinci sözcüğün anlamı ise iş veya harekettir.

<sup>iv</sup> Pheromone



yiyeceğin miktarı ve türüne göre dönüş yolunda çevrelerine feromon bırakırlar. Feromon izlerinin olduğu bölgeden geçen diğer karıncalar, aldıkları kimyasal sinyallerden etkilenerek güzergâhlarını değiştirirler ve feromon izlerini izleyerek besine ulaşabilirler. Zaman içerisinde yiyecekleri yuvaya taşıyan karıncalar sayesinde besin ile yuva arasında birikmiş feromon izleri ile belirlenmiş bir hat oluşur. Diğer taraftan besinin tükenmesi ile beraber dönüş yolunda feromon bırakma işlemi sonlandırılır. İkame edilmeyen feromon izleri geçen sürede buharlaşarak kaybolur ve karıncalar rastsal olarak başka güzergâhları deneyerek besin aramaya devam ederler. Feromon izleri üzerinde herhangi bir engel çıktığında karıncaların başka yolları denedikleri ve besine ulaştıkları takdirde yeniden dönüş yolunda feromon bıraktıkları gözlenmektedir. Böyle bir davranış, şartların sürekli değiştiği dinamik süreçlerde de söz konusu yaklaşımın başarılı olduğunu göstermektedir<sup>(50)</sup>.

Karıncaların güzergâhları üzerinde birden fazla feromon izinin söz konusu olması durumunda, genellikle feromon miktarı daha fazla olan izi takip ettikleri görülmektedir. Dolayısıyla feromon miktarının fazla olduğu güzergâhın karınca tarafından seçilme olasılığının arttığı söylenebilir<sup>(51)</sup>.



**Şekil 3.19** Karıncaların feromon miktarına göre yol tercihi yaptıkları gözlemlenmektedir. Buna göre iki farklı feromon miktarının söz konusu olduğu güzergâh seçiminde karınca, büyük olasılıkla soldaki güzergâhı tercih edecektir.

Biyoloji literatüründeki karınca davranışı üzerine yapılan çalışmalardan hareketle optimizasyon sorunlarının çözümü adına geliştirilen ilk yöntem, yine Dorigo ve meslektaşlarının 1996 yılında yayımladığı “**Karıncı sistemi: koordinasyon içersindeki bireylerden oluşan koloni ile eniyileme**” isimli makale ile gündeme gelen karınca sistemi yaklaşımıdır<sup>(49)</sup>. Karınca sistemi yaklaşımında öncelikle gerçek ve sanal karınca bireyleri arasındaki farklar ifade edilmekte, ardından işlem adımları belirtilerek örnek sorunlardaki uygulamalar sergilenmektedir. Karınca sistemi yaklaşımı ile çözülen ilk sorun, sıralama problemlerinden biri olan gezici satıcı sorunu olup algoritmanın



**Şekil 3.20** Kolonide karınca cinsleri.

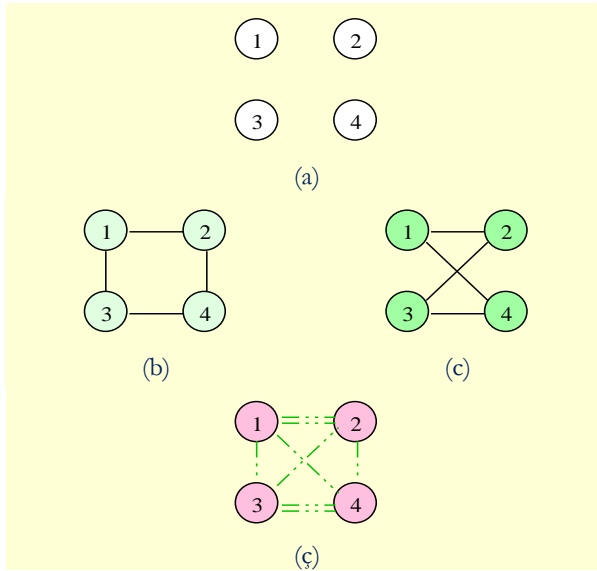
<sup>i</sup> “Ant system: optimization by a colony of operating agents”





performansı, tabu tarama ve benzetimli tavlama yaklaşımları ile karşılaştırılmakta, makale değerlendirmesinde karınca sistemi algoritmasının daha iyi sonuçlar verdiği beyan edilmektedir.

Karınca koloni algoritması, esasen nüfus bazlı bir eniyileme algoritması olup tarama sürecinde birden fazla tasarım yöneyi dikkate alınmaktadır. Karınca olarak adlandırılan ve koloni niteliğindeki nüfusu oluşturan bireyler, her tarama süreci sonunda rassal olarak takip ettikleri güzergâhlara feromen bırakmakta ve bir sonraki tarama sürecinde, bırakılan feromen miktarlarına göre yine rassal olarak yeni güzergâhlar belirlemektedirler. Bırakılan feromen miktarının tespiti önemli bir husus olup bireylerin tarama sürecine ait veri birikimini gösteren en önemli unsurdur. Dolayısıyla kolektif olarak yapılan tarama sürecinde, amaç fonksiyonu yüzeyine ait elde edilen verilerin sonraki tarama süreçlerine girdi bilgisi sağlaması, feromen güncelleme işlemleri ile gerçekleşir. Karıncalar tarafından seçilen güzergâh ve bu güzergâhlarda bıraktıkları feromen miktarının etkisini müteakip şekilde verilen örnek üzerinden izah etmek mümkündür;



**Şekil 3.21** Gezici satıcı sorununda (a) şehir konumları, (b) 1inci karıncanın takip ettiği güzergah, (c) 2nci karıncanın takip ettiği güzergah, (d) İki karıncanın takip ettiği güzergahlar neticesinde bırakılan feromen izleri<sup>(52)</sup>.



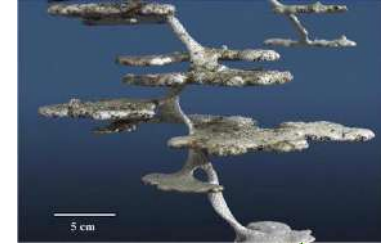
Buna göre simetrik dağılıma sahip dört şehrin dolaşıldığını ve tur sonunda yine başlangıç şehrine dönülmesi gerektiğini, kolonideki iki karıncadan ilkinin (b) güzergâhını, 2nci karıncanın ise (c) güzergâhını takip ederek turu tamamladığını düşünelim. Her iki karıncanın ziyaretler esnasında uğradığı güzergâh bileşenlerine bıraktığı feromen miktarları dikkate alındığında gözlemlenebilecek feromen iz hatları aynı şeklin (ç) şikkında görülmektedir. Şekilden de gözlemlenebileceği üzere şehir 1–2 ve 3–4 arası güzergâh bileşenlerinde daha fazla feromen birikmesi söz konusu olup yeni tur belirlemelerinde bu bileşenlerin seçilme olasılığı daha fazla olabilir.

Koloni büyüklüğünün eniyileme sürecine etkisi önemlidir. Genellikle kolonideki birey sayısının artması, daha geniş bir alanın taranmasına ve dolayısıyla da çözümde iyileşmeye vesile olur. Ancak böyle bir artış beraberinde sayısal işlem sayısını da artırır. Sıralama sorunları ile ilgili yapılan çözüm denemelerinde elde edilen tecrübeye göre karınca sayısının, sıralamada dikkate alınan obje sayısına, örneğin gezici satıcı sorununda şehir sayısına, eşit seçilmesi uygun görünmektedir<sup>(53)</sup>.

Karıncı kolonisi yönteminin en uygun çözüme yakınsama süresi ile ilgili kesin bir yargıdan bahsetmek mümkün olmasa da yakınsamanın eninde sonunda gerçekleşeceğine dair genel bir kanı mevcuttur. Diğer taraftan çıkış kaynağı itibari ile karınca kolonisi yöntemi, tasarım yöneyi olarak ayrık karaktere sahip sorunların çözümünde yaygın olarak kullanılabilir. Bununla beraber tasarım yöneylerinin gerçek sayılardan oluştuğu sorunlarda da yöntemin uygulanması mümkündür. Takdir edileceği üzere özellikle gezici satıcı problemi gibi sıralama sorunlarında, çözüm uzayı şehir sayısına bağlı olarak çok büyük de olsa sonlu bir değerdedir<sup>i</sup>. Ancak gerçek sayıların olduğu sorunlarda çözüm uzayı sayısal manada sonsuzdur. Bununla beraber çözüm uzayından belirli sayıda örnekleme yapılarak tarama sürecinin yönlendirilmesi mümkündür.

Karıncı kolonisi algoritmasının işlem adımlarını birkaç basamakta sıralamak mümkündür. Buna göre algoritmanın başlangıcında tarama sürecini belirleyen azami nesil sayısı, koloninin büyüklüğünü gösteren  $n$  karınca sayısı, feromen etkisinin kontrol edilebilmesini sağlayan ve kullanıcı tarafından belirlenen  $\alpha$  parametresi, ayrık noktalar arası mesafeyi ve onun etkisinin yansıtmak için kullanılan kullanıcı bazlı  $\beta$  parametresi, feromenin buharlaşması özelliğini benzeştirmek için seçilen  $\rho$  sayısı belirlenir. Daha sonra her bir karınca için belli bir şehir veya rassal bir şehir seçimi yapılarak sonraki tekrarlı işlemlere başlanır. Her bir iterasyonda tüm

<sup>i</sup> Örneğin  $m = 10$  şehrin gezilmesi söz konusu olduğunda  $(m - 1)! \rightarrow 9! \rightarrow 362,880$  çözüm seçeneğinin varlığından bahsedilebilir.



**Şekil 3.22** Karınca yuvasının sıvı alüminyum ile alınmış kalıp görünümü.





karıncaların, gezilmesi öngörülen tüm şehirlere bir defa uğraması hedeflenir. Bu maksatla karıncaların sırasıyla bir tam turu tamamlayan güzergâhları belirlenir. Herhangi bir karıncanın bulunduğu konumdan bir sonraki konuma geçişi esnasında hangi güzergâhı tercih edeceği rassal olarak tespit edilir. Ancak bu tespit tüm güzergâhlar belli bir olasılıkla dikkate alınır. Sonraki konuma ait olasılık değerleri;

$$p_k(i, j) = \begin{cases} \frac{\tau^\alpha(i, j)\eta^\beta(i, j)}{\sum_{u \in J_k^{(i)}} \tau^\alpha(i, u)\eta^\beta(i, u)} & j \in J_k^{(i)} \\ 0 & j \notin J_k^{(i)} \end{cases} \quad (3.35)$$

ifadesine göre hesaplanır. Burada;  $p$  olasılık değerini,  $k$  karınca numarasını,  $i$  bulunulan konumu,  $j$  seçilmesi muhtemel bir sonraki konumu,  $J_k^{(i)}$   $k$  karıncası için  $i$  konumundan sonra gidilmesi olası noktalar kümesini,  $\tau(i, j)$   $i$  ve  $j$  noktaları arasındaki feromen değerini göstermektedir. Görünürlük<sup>i</sup> değeri olarak ifade edilen  $\eta(i, j)$ ,  $i$  ve  $j$  noktaları arasındaki mesafenin tersi değerini tanımlamakta olup;

$$\eta(i, j) = \frac{1}{d(i, j)} \quad (3.36)$$

şeklindeki denklemlerle temsil edilmektedir. Burada  $d(i, j)$ ,  $i$  ve  $j$  noktaları arasındaki mesafeyi ifade etmektedir. Görüldüğü üzere  $J_k^{(i)}$  ifadesi,  $k$  karıncasının ziyaret ettiği noktaların dışında kalan ve bir manada hafıza görevi gören bir küme olup aynı şehrin tekrar ziyaret edilmesini önleyici bir rol oynar.

Olasılıklar toplamı;

$$\sum_{j=1}^{|J_k|} p_k(i, j) = 1 \quad (3.37)$$

ifadesi çerçevesinde 1 değerine eşit olup bir sonraki nokta, rassal sayılara bağlı olarak bulunabilir. Tahmin edilebileceği üzere olasılık değerinin büyük olduğu  $i - j$  güzergâhının seçilme şansı daha fazladır. Ardı ardına yapılan konum tayinleri sonrasında  $k$  karıncasının bütün şehirleri ziyareti gerçekleşmiş olur. Benzer işlem adımları diğer tüm karıncalar için tekrarlanır ve bu sayede ilk

<sup>i</sup> Visibility



iterasyon tamamlanmış olur. Akabinde her bir karıncaya ait amaç fonksiyon değerleri hesaplanır. Hesaplama genellikle her bir karıncaya ait ziyaret güzergâhının mesafe değerleri toplanır;

$$L^k(t+1) = \sum_{q=1}^{m-1} d(s^{(k)}(q), s^{(k)}(q+1)) \quad (3.38)$$

Denklemden geçen  $s^{(k)}$  terimi,  $k$  karıncasına ait çözüm kümesini ve  $m$  toplam şehir sayısını ifade etmektedir. Eldeki çözümlerin değerlendirilmesi ve yakınsama kontrolü sonrasında tarama işlemine devam edilir veya işlemler sonlandırılır.

Yakınsamada genellikle azami iterasyon sayısına ulaşıp ulaşılmadığı, amaç fonksiyon değerinin istenilen değerden daha düşük olup olmadığı, ardı ardına yapılan iterasyonlarda amaç fonksiyonunda kayda değer bir değişikliğin gerçekleşip gerçekleşmediği gibi farklı ölçütler dikkate alınabilir.

Yakınsama kriterinin sağlanmaması durumunda yapılacak ilk işlem feromon güncellemesidir. Güncelleme işlemlerinde;

$$\tau^{(t+1)}(i, j) = (1 - \rho)\tau^{(t)}(i, j) + \sum_{k=1}^n \Delta_k \tau^{(t+1)}(i, j) \quad (3.39)$$

ifadesi dikkate alınır. Burada  $t$  iterasyon sayısını temsil etmektedir. Denklemden de görüleceği üzere  $i - j$  aralığındaki feromonun buharlaşması süreci  $\rho$  katsayısı ile kontrol edilebilmektedir.

Diğer taraftan aynı aralığa ait feromon birikimi ise;

$$\Delta_k \tau^{(t+1)}(i, j) = \begin{cases} \frac{Q}{L^k(t+1)} & k \text{ karıncası } i - j \text{ yolunu kullanmış ise} \\ 0 & \text{kullanmamış ise} \end{cases} \quad (3.40)$$

eşitliğine göre hesaplanır. Denklemden  $Q$  sabit bir sayı olup genellikle 1 olarak dikkate alınır. Buna göre turun kısaltığı  $i - j$  aralığı arasına bırakılan feromon miktarını artırıcı rol oynar. Feromon güncelleme işlemlerinden sonra her bir karınca için yeni feromon miktarları ve olasılık değerlerine göre yeni güzergâh çözümleri üretilir. Tarama sürecine yakınsama ölçütü sağlanana kadar benzer işlemlere devam edilir.



Feromen/Algoritma	Karınca-yoğun <sup>i</sup>	Karınca-miktar <sup>ii</sup>	Karınca-sistemi <sup>iii</sup>	Maks-Min Karınca	Mertebe bazlı Karınca
Güncelleme zamanı	Her adım sonrasında	Her adım sonrasında	Tam tur sonrasında	Tam tur sonrasında	Tam tur sonrasında
Artış miktarı	$\Delta\tau(i, j) = Q_1$	$\Delta\tau(i, j) = Q_2/d(i, j)$	$\Delta\tau(i, j) = Q_3/L_k$	$\Delta\tau(i, j) = Q_4/L_{eni}$	$\Delta\tau(i, j) = Q_4/L_{r-eni}$
Güncellenen karınca	Her karınca	Her karınca	Her karınca	En iyi karınca	En iyi r karınca

**Çizelge 3.1** Literatürde sıkça karşılaşılan bazı karınca koloni algoritmaları.

Diğer taraftan karınca koloni algoritmaları feromenin ne zaman, ne kadar ve nasıl güncelleneceği sorularına verilebilecek farklı cevaplar dikkate alınarak çeşitlendirilebilir. Örneğin feromen güncellemesi, karıncanın attığı her bir adımı müteakiben veya karıncanın tam bir turu tamamlaması sonrasında güncellenebilir. Güncelleme işleminin her bir karıncayı kapsamaması söz konusu olabileceği gibi belirli bir karıncayı, örneğin en iyi çözümü bulan karıncayı, ya da belirli sayıdaki karıncaları, örneğin en iyi  $r$  sayıdaki çözümü veren karıncaları kapsayabilir. Güncellemede belirli bir feromen değeri veya konumlar arası mesafe ile tur mesafelerinin şekillendirdiği feromen değerleri dikkate alınabilir. Çizelge 3.1'de literatürde rastlanan ve bahse konu ölçütlere göre tanımlanan bazı karınca koloni algoritmaları görülmektedir<sup>(54)</sup>;

Literatürde karşılaşılabilecek diğer bir feromen bazlı yaklaşım, pozitif veya negatif feromen tanımlamasıdır. Genel uygulamaların aksine negatif feromen uygulamasında, tarama yapan karıncaların istenmeyen güzergâhlardan uzaklaştırılması hedeflenmektedir<sup>(55)</sup>. Karınca koloni algoritmalarının sayısal hesaplamalarda dikkate alınabilecek diğer bir özelliği, paralel hesaplama yeteneğidir. Nüfuz bazlı yöntemler grubunda olması nedeniyle koloninin her bir karıncasına ait tur hesaplamalarını ayrı bir işlemci ile aynı anda hesaplamak mümkündür. Bu sayede hesaplama süresinde ciddi kazanımlar elde edilebilir<sup>(56)</sup>. Karınca koloni algoritmasının daha iyi anlaşılabilmesi için müteakip örnek bir sorunu karınca-sistemi algoritmasını kullanarak çözmek mümkündür. Buna göre klasik bir gezici satıcı sorunu kapsamında 1nci şehir başlangıç olmak üzere mevcut beş şehri dolaşmak isteyen satıcının, tur sonunda tekrar 1inci şehre dönmesi istenmektedir. Şehirlerarası mesafeyi gösteren mesafe matrisi;

$$d = \begin{bmatrix} 0 & 10 & 12 & 11 & 14 \\ 10 & 0 & 13 & 15 & 8 \\ 12 & 13 & 0 & 9 & 14 \\ 11 & 15 & 9 & 0 & 16 \\ 14 & 8 & 14 & 16 & 0 \end{bmatrix}$$

<sup>i</sup> Ant-density algorithm

<sup>ii</sup> Ant-quantity algorithm

<sup>iii</sup> Ant-cycle algorithm





şeklinde verilmektedir. Verilen matriste satır ve sütun numaraları şehir numarası anlamına gelmekte olup örneğin  $d(2,3)$  ifadesi 2nci ve 3üncü şehirlerarası mesafenin 13 km olduğunu göstermektedir. Tanımlı mesafe matrisi kullanılarak görünürlük matrisi de kolayca yazılabilir;

$$\eta(i,j) = \frac{1}{d(i,j)} \quad i \neq j$$

$$\eta(i,j) = 0 \quad i = j$$

$$\eta = \begin{bmatrix} 0.0000 & 0.1000 & 0.0833 & 0.0909 & 0.0714 \\ 0.1000 & 0.0000 & 0.0769 & 0.0667 & 0.1250 \\ 0.0833 & 0.0769 & 0.0000 & 0.1111 & 0.0714 \\ 0.0909 & 0.0667 & 0.1111 & 0.0000 & 0.0625 \\ 0.0714 & 0.1250 & 0.0714 & 0.0625 & 0.0000 \end{bmatrix}$$

Çözüm dâhilinde kullanılacak karınca sayısı  $n = 3$ , algoritma parametre değerleri;  $Q = 1, \alpha = 1, \beta = 2, \rho = 0.5$ , azami iterasyon sayısı  $T = 100$ , güzergahlar arası başlangıç feromen değerleri  $\tau^{(0)}(i,j) = 1$  olarak belirlenebilir. Parametre değerlerinin belirlenmesi sonrasında ilk iterasyona,  $t = 1 \rightarrow T$  işlemine başlanır. İterasyon dâhilinde yapılacak ilk işlem, 1inci karınca için güzergâh belirleme işlemidir. Hatırlanacağı üzere karıncanın geziye başlangıç noktası, 1inci noktayı. Dolayısıyla 1inci karıncanın bulunduğu konum olan 1inci şehirden bir sonraki olası noktalar kümesi;

$$J_1^{(1)} = \{2,3,4,5\}$$

olarak ifade edilebilir. Sonraki noktanın belirlenmesi için öncelikle her bir olası güzergâha ait olasılık değerinin hesaplanması gerekir. Örneğin 1inci karıncanın, bulunduğu 1inci şehirden olası 2nci şehri seçme olasılığı;

$$p_1(1,2) = \frac{\tau^1(1,2)\eta^2(1,2)}{\sum_{u \in J_1^{(1)}} \tau^1(1,u)\eta^2(1,u)}$$

ifadesine göre elde edilebilir. Tabiatıyla denklemin paydasının belirlenebilmesi için her bir olası şehrin dikkate alındığı bireysel kısımların hesaplanması gerekir;

$$\begin{aligned} \tau^1(1,2)\eta^2(1,2) &= 1 \times (0.1000)^2 = 0.0100 \\ \tau^1(1,3)\eta^2(1,3) &= 1 \times (0.0833)^2 = 0.0069 \\ \tau^1(1,4)\eta^2(1,4) &= 1 \times (0.0909)^2 = 0.0083 \\ \tau^1(1,5)\eta^2(1,5) &= 1 \times (0.0714)^2 = 0.0051 \\ \sum_{u \in J_1^{(1)}} \tau^1(1,u)\eta^2(1,u) &= 0.0100 + 0.0069 + 0.0083 + 0.0051 = 0.0303 \end{aligned}$$

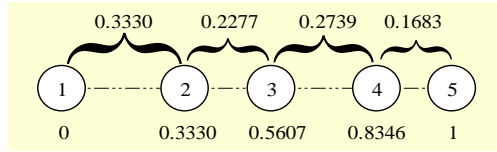


Dolayısıyla her bir olası şehre ait olasılık değerleri;

$$p_1(1,2) = \frac{0.0100}{0.0303} = 0.3300, \quad p_1(1,3) = \frac{0.0069}{0.0303} = 0.2277$$

$$p_1(1,4) = \frac{0.0083}{0.0303} = 0.2739, \quad p_1(1,5) = \frac{0.0051}{0.0303} = 0.1683$$

şeklinde elde edilebilir. Söz konusu dört güzergâhın hangisinin seçileceği ise rassal olarak belirlenen bir husustur. Mevcut olasılık değerlerinin bir tabla üzerinde dikkate alınması halinde;



sıralaması yapılabilir. Bu aşamada uygun bir rassal sayı üretici kullanılarak 0 ile 1 arasında rastgele bir sayı üretilir; örneğin  $rand \rightarrow 0.7296$ . Üretilen bu sayı görüldüğü üzere 0.5607 ile 0.8346 arasına düşmekte olup buna göre bir sonraki şehir 4üncü şehir olarak belirlenebilir. Sonraki şehrin belirlenmesi için aynı adımlar tekrar edilir. Dolayısıyla 1inci karıncanın hali hazırda bulunduğu konum olan 4üncü şehirden bir sonraki olası şehirler kümesi;

$$J_1^{(4)} = \{2,3,5\}$$

şeklinde ifade edilebilir. Olası güzergâh için olasılık değerinin hesaplanması dâhilinde 1inci karıncanın bulunduğu 4üncü şehirden olası 2nci şehri seçme olasılığı;

$$p_1(4,2) = \frac{\tau^1(4,2)\eta^2(4,2)}{\sum_{u \in J_1^{(4)}} \tau^1(4,u)\eta^2(4,u)}$$

ifadesine göre belirlenebilir. Denklemdaki bireysel kısımların hesaplanması çerçevesinde;

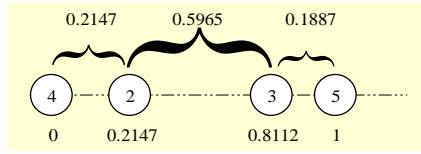
$$\begin{aligned} \tau^1(4,2)\eta^2(4,2) &= 1 \times (0.0667)^2 = 0.0044 \\ \tau^1(4,3)\eta^2(4,3) &= 1 \times (0.1111)^2 = 0.0123 \\ \tau^1(4,5)\eta^2(4,5) &= 1 \times (0.0625)^2 = 0.0039 \\ \sum_{u \in J_1^{(4)}} \tau^1(4,u)\eta^2(4,u) &= 0.0044 + 0.0123 + 0.0039 = 0.0207 \end{aligned}$$



değerlerine ulaşılır. Dolayısıyla her bir olası şehre ait olasılık değerleri;

$$p_1(4,2) = \frac{0.0044}{0.0207} = 0.2147, \quad p_1(4,3) = \frac{0.0123}{0.0207} = 0.5965, \quad p_1(4,5) = \frac{0.0039}{0.0207} = 0.1887$$

olarak elde edilebilir. Mevcut olasılık değerlerinin bir tabela üzerinde gösterilmesi halinde;



sıralaması gözlemlenebilir. Rassal sayı üretici kullanılarak rastgele bir sayı üretilir; örneğin  $rand \rightarrow 0.5744$ . Üretilen bu sayı görüldüğü üzere 0.2147 ile 0.8112 arasında düşmekte olup buna göre bir sonraki şehir 3üncü şehir olarak belirlenebilir.

1inci karıncanın bulunduğu son konum olan 3üncü şehirden sonraki olası şehirler kümesi;

$$J_1^{(3)} = \{2,5\}$$

şeklinde ifade edilebilir. Olası güzergâh için olasılık değerinin hesaplanması dâhilinde 1nci karıncanın bulunduğu 3üncü şehirden olası 2nci şehri seçme olasılığı;

$$p_1(3,2) = \frac{\tau^1(3,2)\eta^2(3,2)}{\sum_{u \in J_1^{(3)}} \tau^1(3,u)\eta^2(3,u)}$$

ifadesine göre belirlenebilir. Denklemdaki bireysel kısımların hesaplanması çerçevesinde;

$$\begin{aligned} \tau^1(3,2)\eta^2(3,2) &= 1 \times (0.0769)^2 = 0.0059 \\ \tau^1(3,5)\eta^2(3,5) &= 1 \times (0.0714)^2 = 0.00509 \\ \sum_{u \in J_1^{(3)}} \tau^1(3,u)\eta^2(3,u) &= 0.0059 + 0.00509 = 0.01099 \end{aligned}$$

değerlerine ulaşılır. Dolayısıyla her bir olası şehre ait olasılık değerleri;



$$p_1(3,2) = \frac{0.0059}{0.01099} = 0.5368, \quad p_1(3,5) = \frac{0.00509}{0.01099} = 0.4632$$

olarak elde edilebilir. Mevcut olasılık değerlerinin dikkate alınması ve rassal sayı üretici kullanılarak 0 ile 1 arasında rastgele bir sayı üretilmesi neticesinde; örneğin  $rand \rightarrow 0.6032$ , bu sayının 0 ile 0.5368 arasına düştüğü görülür. Buna göre bir sonraki şehir 2inci şehir olarak belirlenebilir. Tur dâhilinde son şehir doğal olarak 5inci şehirdir. İlk karıncanın tamamladığı tur;

$$\text{1inci karınca: } 1 - 4 - 3 - 2 - 5 - 1$$

şeklinde neticelenmiş olup bu turun oluşumunda dikkate alınan işlem adımları aynen diğer 2 ve 3üncü karıncalar için de gerçekleştirilebilir. Eğer benzer işlemler gerçekleştirilir ve olası değerler değerlendirilirse;

$$\begin{aligned} \text{2nci karınca: } & 1 - 4 - 2 - 5 - 3 - 1 \\ \text{3üncü karınca: } & 1 - 4 - 5 - 2 - 3 - 1 \end{aligned}$$

çözümlerine ulaşılabilir. 1inci iterasyon neticesinde elde edilen çözümler değerlendirilerek yakınsama kontrolü yapılabilir. Azami iterasyon sayısının 100 olması nedeniyle bir sonraki iterasyona geçilir. Bu aşamada yapılacak ilk işlem feromen miktarlarının güncellenmesidir;

$$\tau^{(1)}(i,j) = (1 - \rho)\tau^{(0)}(i,j) + \sum_{k=1}^3 \Delta_k \tau^{(1)}(i,j)$$

Güncellemede yapılacak ilk işlem adımı, feromen buharlaşmasının dikkate alınmasıdır;

$$(1 - \rho)\tau^{(0)}(i,j) \rightarrow (1 - 0.5)\tau^{(0)}(i,j) \rightarrow \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0.5 & 0.5 \end{bmatrix}$$

Daha sonra her bir karıncanın takip ettiği güzergâhlar dikkate alınarak ilave feromen izleri eklenir. İlk karıncanın güzergâhının maliyet değeri;

$$\begin{aligned} L^1 &= d(1,4) + d(4,3) + d(3,5) + d(5,2) + d(2,1) \\ L^1 &= 11 + 9 + 14 + 8 + 10 = 52 \end{aligned}$$



olup geçtiği güzergâhlara bıraktığı ilave feromen miktarı;

$$\Delta_1 \tau(i, j) = \begin{cases} \frac{1}{L^1} = \frac{1}{52} = 0.0192 & \text{1 karıncası } i - j \text{ yolunu kullanmış ise} \\ 0 & \text{kullanmamış ise} \end{cases}$$

şeklindedir. Dolayısıyla ilk karıncanın bıraktığı izlerden dolayı feromen matrisi;

$$\begin{bmatrix} 0.5000 & 0.5000 & 0.5000 & 0.5192 & 0.5000 \\ 0.5192 & 0.5000 & 0.5000 & 0.5000 & 0.5000 \\ 0.5000 & 0.5000 & 0.5000 & 0.5000 & 0.5192 \\ 0.5000 & 0.5000 & 0.5192 & 0.5000 & 0.5000 \\ 0.5000 & 0.5192 & 0.5000 & 0.5000 & 0.5000 \end{bmatrix}$$

halini alır. İkinci karıncanın güzergâhının maliyet değeri;

$$\begin{aligned} L^2 &= d(1,4) + d(4,2) + d(2,5) + d(5,3) + d(3,1) \\ L^2 &= 11 + 15 + 8 + 14 + 12 = 60 \end{aligned}$$

olup geçtiği güzergâhlara bıraktığı ilave feromen miktarı;

$$\Delta_2 \tau(i, j) = \begin{cases} \frac{1}{L^2} = \frac{1}{60} = 0.0166 & \text{2 karıncası } i - j \text{ yolunu kullanmış ise} \\ 0 & \text{kullanmamış ise} \end{cases}$$

şeklindedir. Buna göre ikinci karıncanın bıraktığı izlerden dolayı feromen matrisi;

$$\begin{bmatrix} 0.5000 & 0.5000 & 0.5000 & 0.5358 & 0.5000 \\ 0.5192 & 0.5000 & 0.5000 & 0.5000 & 0.5166 \\ 0.5166 & 0.5000 & 0.5000 & 0.5000 & 0.5192 \\ 0.5000 & 0.5166 & 0.5192 & 0.5000 & 0.5000 \\ 0.5000 & 0.5192 & 0.5166 & 0.5000 & 0.5000 \end{bmatrix}$$

şeklinde elde edilir. Son karıncanın güzergâhının maliyet değeri;

$$\begin{aligned} L^3 &= d(1,4) + d(4,5) + d(5,2) + d(2,3) + d(3,1) \\ L^3 &= 11 + 16 + 8 + 13 + 12 = 60 \end{aligned}$$

olup geçtiği güzergâhlara bıraktığı ilave feromen miktarı;



$$\Delta_3 \tau(i, j) = \begin{cases} \frac{1}{L^3} = \frac{1}{60} = 0.0166 & 3 \text{ karıncası } i - j \text{ yolunu kullanmış ise} \\ 0 & \text{kullanmamış ise} \end{cases}$$

şeklindedir. Buna göre son karıncanın bıraktığı izlerden dolayı feromen matrisi;

$$\tau^{(1)} = \begin{bmatrix} 0.5000 & 0.5000 & 0.5000 & 0.5524 & 0.5000 \\ 0.5192 & 0.5000 & 0.5166 & 0.5000 & 0.5166 \\ 0.5332 & 0.5000 & 0.5000 & 0.5000 & 0.5192 \\ 0.5000 & 0.5166 & 0.5192 & 0.5000 & 0.5166 \\ 0.5000 & 0.5358 & 0.5166 & 0.5000 & 0.5000 \end{bmatrix}$$

şeklindeki son halini alır. Feromen güncellemesi sonrasında, daha önceki iterasyonda her bir karınca için yapılan tüm işlemler aynen tekrar edilir. Buna göre yine 1inci karınca için, başlangıç şehri 1inci şehir olmak üzere, bir sonraki şehre ait olasılık değerleri hesaplanabilir. 1inci karıncanın bulunduğu konum olan 1inci şehirden bir sonraki olası noktalar kümesi;

$$J_1^{(1)} = \{2,3,4,5\}$$

şeklinde olup olasılık hesabında dikkate alınan bireysel kısımlar;

$$\begin{aligned} \tau^1(1,2)\eta^2(1,2) &= 0.5000 \times (0.1000)^2 = 0.00500 \\ \tau^1(1,3)\eta^2(1,3) &= 0.5000 \times (0.0833)^2 = 0.00347 \\ \tau^1(1,4)\eta^2(1,4) &= 0.5524 \times (0.0909)^2 = 0.00456 \\ \tau^1(1,5)\eta^2(1,5) &= 0.5000 \times (0.0714)^2 = 0.00255 \\ \sum_{u \in J_1^{(1)}} \tau^1(1,u)\eta^2(1,u) &= 0.00500 + 0.00347 + 0.00456 + 0.00255 = 0.01558 \end{aligned}$$

olarak elde edilebilir. Dolayısıyla her bir olası şehre ait olasılık değerleri;

$$\begin{aligned} p_1(1,2) &= \frac{0.00500}{0.01558} = 0.3209, & p_1(1,3) &= \frac{0.00347}{0.01558} = 0.2226 \\ p_1(1,4) &= \frac{0.00456}{0.01558} = 0.2929, & p_1(1,5) &= \frac{0.00255}{0.01558} = 0.1635 \end{aligned}$$

şeklindedir. Mevcut olasılık değerlerinin bir tabela üzerinde dikkate alınması ve rassal sayı üretici ile rastgele aralık tayini yapılması halinde, örneğin  $rand \rightarrow 0.1347$ , üretilen bu sayının 0 ile 0.3209 arasına düşmekte olduğu, dolayısıyla bir sonraki şehrin 2inci şehir olduğu belirlenebilir. Sonraki



adımlar, yukarıdaki yapılan işlemlerin tekrarı niteliğinde olup azami iterasyon sayısınca yapılan işlemler neticesinde optimal şehir turu ve amaç fonksiyon değerinin;

*optimal karınca turu: 1 – 2 – 5 – 3 – 4 – 1*  
*f\* = 52*

olduğu görülebilir<sup>(57)</sup>.

2B ortamında 5 şehir için seyahat planlaması.

**MATLAB uygulaması**

```

clc; clear
% mesafe matrisi
d = [0,10,12,11,14;10,0,13,15,8;12,13,0,9,14;11,15,9,0,16;14,8,14,16,0];
iter = 1000; % azami iterasyon sayısı
m = 10; % karınca sayısı
n = length(d); % şehir sayısı
e = 0.4; % buharlaşma parametresi
alpha = 1; % algoritma parametresi
beta = 1; % algoritma parametresi
for i=1:n % görünürlük matrisi oluşumu
    for j=1:n
        if d(i,j)==0
            h(i,j)=0;
        else
            h(i,j)=1/d(i,j);
        end
    end
end
thoinit=.1*ones(n); % başlangıç feromen değerleri
for i=1:iter
    for i=1:m
        % karıncaların ilk yerleşimi
        rute(i,1)=1 ; % linci şehirden başlama
    end
    % karınca turları
    for i=1:m
        mh = h ;
        for j=1:n-1
            c = rute(i,j) ;
            mh(:,c) = 0 ;
            tho = (thoinit(c,:).^beta).*(mh(c,:).^alpha); % feromen güncelleme
            s = (sum(tho)) ;
            p = (1/s).*tho ; % güzergah olasılığı
            r = rand ; % rassal sayı üretimi
            s = 0 ;
            for k = 1:n

```





```
s = s + p(k); % rulet tablası
if r<=s
    rute(i,j+1) = k ;
    break
end
end
end
end
end
rute_c = horzcat(rute,rute(:,1)) ; % Tur başlangıcına dönüş
% amaç fonksiyonu hesaplamaları
for i = 1:m
    s = 0;
    for j=1:n
        s = s+d(rute_c(i,j),rute_c(i,j+1));
    end
    f(i)=s;
end
F = f;
[minf,idk]=min(f);
% feromen buharlaşması
thoinit = (1-e)*thoinit;
% feromen güncelleme
for i = 1:m
    for j=1:n
        dt = 1/(f(i)-minf) ; % ortak maliyetin devreden çıkarılması
        thoinit(rute_c(i,j),rute_c(i,j+1))=thoinit(rute_c(i,j),rute_c(i,j+1))+dt ;
    end
end
[mincost(i),number] = min(F); % asgari maliyet
besttour(i,:) = rute_c(number,:); % en iyi tur
end
besttour = besttour(end,:)'
mincost = mincost(end)
```



### Sonuç

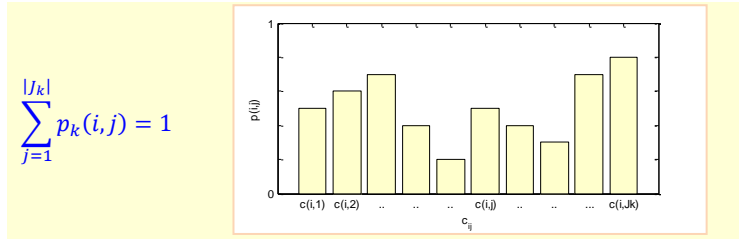
```
besttour = 1    2    5    3    4    1
mincost = 52
```







Karınca koloni optimizasyon algoritmasının, ayrık tasarım parametreleri yanında gerçek sayıları içeren mühendislik sorunlarının çözümünde de kullanılması mümkündür. Bu amaç çerçevesinde, algoritmanın doğası değişmeden bazı yaklaşım değişikliklerine gitmek gerekir. Ayrık karaktere sahip değişkenlerden süreklilik arz eden tasarım değişkenlerine geçişte temel düşünce, ayrık nitelikteki olasılık dağılım fonksiyonundan süreklilik arz eden olasılık fonksiyon dağılımlarına geçiştir. Hatırlanacağı üzere sıralama sorunlarında bulunan  $c(i, j)$  konumunu takiben bir sonraki olası konumun seçiminde, olası konumlara ait  $p(i, j)$  olasılık değerleri dikkate alınmaktaydı. Olası konum sayısı, olasılık değeri sayısını belirleyen unsurdur. Sonlu sayıda konumun söz konusu olduğu problemlerde olasılık değerleri de belirli bir sayıda ve toplamları 1 olacak şekilde dağılmaktaydı;



Şekil 3.23 Ayrık formda olasılık dağılım fonksiyonu.

İlkesel olarak süreklilik arz eden olasılık dağılım fonksiyonu;

$$\int_{-\infty}^{+\infty} p(x) dx = 1, \quad p(x) \geq 0 \quad (3.41)$$

koşullarını sağlayan herhangi bir fonksiyon olabilir. Genellikle dağılım fonksiyonu olarak normal dağılımı fonksiyonunu temsil eden Gauss fonksiyonu tercih edilir. Söz konusu fonksiyon;

$$g(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (3.42)$$

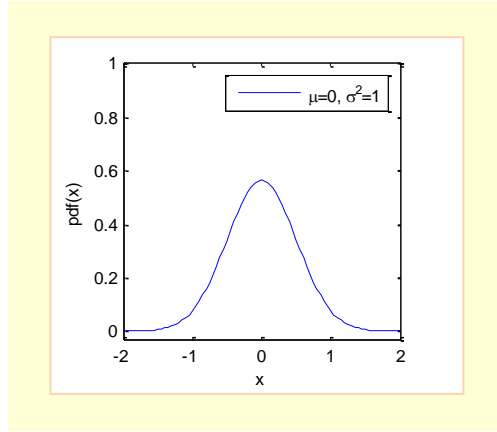
denklemleri ile tanımlanmakta olup burada  $\mu$  ortalama değeri,  $\sigma^2$  ise varyans değerini temsil etmektedir.

<sup>i</sup> Probability distribution function: PDF



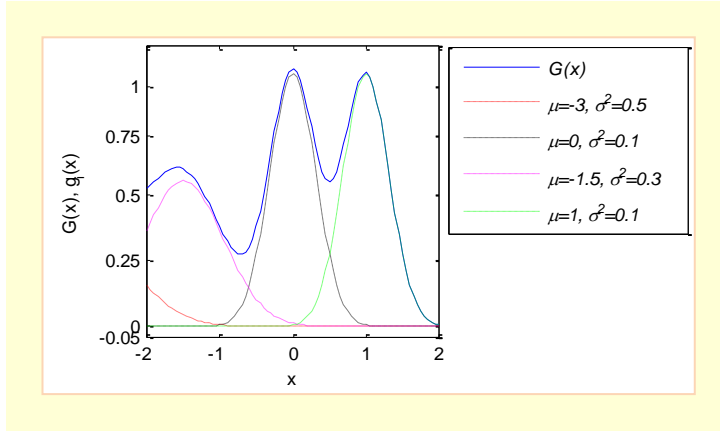


Gauss fonksiyonu, ortalama değeri merkez olmak üzere tek bir tepe değerine sahiptir.



Şekil 3.24 Örnek Gauss dağılım fonksiyonu.

Bu nedenle tek başına bir olasılık değeri ifade etmesine rağmen sonsuz seçeneğin söz konusu olduğu, ya da belirli bir örnek uzayının söz konusu olduğu durumlarda kullanımı uygun olmaz.



Şekil 3.25 Örnek Gauss ve Gauss fonksiyonlar fonksiyonu dağılımları.



Bununla beraber birden fazla ama farklı istatistikî değerlere sahip Gauss fonksiyonlar fonksiyonunun<sup>i</sup> kullanımı, söz konusu ihtiyaca cevap verebilir;

$$G^i(x) = \sum_{j=1}^k w_j g_j^i(x), i = 1, \dots, n \quad (3.43)$$

$$g_j^i(x) = \frac{1}{\sigma_j^i \sqrt{2\pi}} e^{-\frac{(x-\mu_j^i)^2}{2\sigma_j^{i2}}} \quad (3.44)$$

Burada  $n$  terimi tasarım parametresinin boyutunu ve  $k$  terimi örnekleme yaklaşımı dâhilinde dikkate alınan çözüm sayısını göstermektedir. Diğer taraftan fonksiyonlar fonksiyonu dâhilindeki  $w_j$   $j$ 'ninci çözüme ait ağırlık katsayısını,  $\mu_j^i$  terimi ortalama değer niteliğinde  $i$ 'ninci boyuta ait  $j$ 'ninci çözümün değerini,  $\sigma_j^i$  simgesi ise standart sapma niteliğinde  $i$ 'ninci boyuta ait  $j$ 'ninci çözümün standart sapma değerini temsil etmektedir. Ayrık sorunlarda bir sonraki noktanın hangi konumlar arasından seçilebileceği belirli olduğundan, sadece olası söz konusu konumlara dair olasılık hesabı yapılması yeterlidir. Ancak süreklilik arz eden sorunlarda olası çözüm noktası sonsuzdur. Bu nedenle muhtemel çözüm noktası öngörüsü pek mümkün değildir. Dolayısıyla sonsuz sayıdaki çözüm noktası yerine örnekleme yaklaşımı ile belirli sayıdaki çözümlerden oluşan bir çözüm arşivi kaydı tutularak olası çözüm noktaları kümesi oluşturulur. Söz konusu çözüm arşivi kümesi dâhilindeki çözümler her iterasyonda güncellenerek bir manada feromen birikimi ve güncellemesi işlemleri gerçekleştirilmiş olur. Çözüm arşivi, genellikle en iyi çözümü veren tasarım yöneyinden başlayarak sırasıyla en kötü çözüme doğru sıralanır. Tabiatıyla çözüm kümesine ait ağırlık değerleri, en yüksekte en düşüğe doğru sıralanmış olur. Müteakip çizelgede çözüm arşivi ve her bir çözüme ait yöney elemanları, amaç fonksiyonu ve ağırlık katsayısı sıralamaları görülmektedir;

$$f(s_1) \leq f(s_2) \leq \dots \leq f(s_k)$$

$$w_1 \geq w_2 \geq \dots \geq w_k$$

$s_1$	$s_1^1$	$s_1^2$	$s_1^3$	...	$s_1^n$	$f(s_1)$	$w_1$
$s_2$	$s_2^1$	$s_2^2$	$s_2^3$	...	$s_2^n$	$f(s_2)$	$w_2$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	...	$\vdots$	$\vdots$	$\vdots$
$s_k$	$s_k^1$	$s_k^2$	$s_k^3$	...	$s_k^n$	$f(s_k)$	$w_k$
	$G^1$	$G^2$	$G^3$	...	$G^n$		

<sup>i</sup> Gaussian Kernel pdf



Görüldüğü üzere Gauss fonksiyonlar fonksiyonu her bir tasarım yöneyi boyutuna göre, tüm arşiv çözümlerinin o boyuttaki yöney elemanı değerlerinin dikkate alınması ile oluşturulmaktadır. Oluşumda dikkate alınan diğer bir husus, çözüm arşivindeki çözümlerin etkisinin ağırlık katsayısı ile belirlenmiş olmasıdır. Fonksiyonlar fonksiyonunda geçen ortalama değerleri daha önce de tanımlandığı üzere;

$$\mu^i = \{s_1^i, s_2^i, \dots, s_k^i\} \quad (3.45)$$

şeklinde ifade edilebilir. Aynı fonksiyondaki standart sapma değerleri ise;

$$\sigma_j^i = \gamma \sum_{a=1}^k \frac{|s_a^i - s_j^i|}{k-1}, \quad 0 < \gamma < 1 \quad (3.46)$$

ifadesine göre belirlenir. Burada  $\sigma_j^i$   $i$ 'nci çözümün  $j$ 'nci yöney elemanının standart sapma değerini göstermektedir. Standart sapma katsayısı olarak adlandırılan  $\gamma$  sabit bir katsayı olup tüm boyutlar için aynı değer olarak dikkate alınmaktadır. Söz konusu katsayı bir anlamda ayrık yaklaşımdaki buharlaşma katsayısı görevini görmektedir. Tanımdan da gözlemleneceği üzere standart sapma değerinin belirlenmesinde arşiv çözümlerinin o boyuttaki dağılımları ve arşiv çözüm kümesinin büyüklüğü değerlendirilmektedir.

Diğer taraftan ağırlık katsayılarının belirlenmesi, doğrudan doğruya amaç fonksiyon değeri sıralamasına göre yapılabilmektedir;

$$w_j = \frac{1}{qk\sqrt{2\pi}} e^{-\frac{(j-1)^2}{2q^2k^2}}, \quad 0 < q < 1 \quad (3.47)$$

Burada  $j$  ifadesi sıralamadaki sıra numarasını, seçim etmen katsayısı olarak adlandırılan  $q$  terimi sıralamanın etkisini kontrol eden sabit bir sayıyı ve  $k$  arşiv çözüm kümesindeki çözüm sayısını göstermektedir. Bu tanımlamaya göre iterasyonlar dâhilinde arşiv çözüm kümesindeki o sıraya ait çözüm değışı de ağırlık katsayısı değışmemektedir. Sıralama dâhilindeki çözümlerden hangisinin tercih edileceği sorusu, olasılık değerine göre cevaplanabilir. Olasılık değeri, ağırlık katsayılarına göre aşağıdaki gibi belirlenebilir<sup>(58)</sup>;

$$p_j = \frac{w_j}{\sum_{a=1}^k w_a} \quad (3.48)$$



Karınca koloni algoritmasının gerçek sayıları içeren sorunların çözümünde takip edilen işlem adımlarını müteakip maddeler halinde sıralamak mümkündür;

- ▶ 1. Öncelikle azami iterasyon sayısı  $T$ , arşiv çözüm kümesi büyüklüğü  $n_p$ , çözüm uzayının örneklendirmesi çerçevesinde dikkate alınacak örnekleme küme büyüklüğü  $n_s$ , seçim etmen katsayısı  $q$ , standart sapma katsayısı  $\gamma$  ve her bir tasarım değişkenine ait alt ve üst sınır değerleri;  $(x_j^l, x_j^u, j = 1, \dots, n)$  belirlenir.
- ▶ 2. Parametre değerlerinin tespiti sonrasında başlangıçta kullanılacak arşiv çözümlerini içeren kümenin üretilmesi gerekir. Optimizasyon süreci ilerledikçe güncellenecek olan arşiv yada diğer bir ifade ile envanter çözüm kümesinin başlangıç topluluğu, genellikle rassal olarak üretilir. Üretimde tasarım değişkenlerine ait alt ve üst sınır değerler dikkate alınır;

$$N_p^{(0)}(x_j) = x_j^l + (x_j^u - x_j^l)rand \quad (3.49)$$

Başlangıç envanter çözüm kümesinin eldesini müteakip her bir çözüme ait amaç fonksiyon değeri hesaplanır ve elde edilen değerlere göre küçükten büyüğe doğru küme sıralaması yeniden düzenlenir;

$$f[N_p^{(0)}(x^i)], \quad i = 1, \dots, n_p$$

$$N_p^{(0)} \leftarrow N_p^{(0)}(\text{sıra})$$

Buna göre envanterdeki çözümlerden ilki, minimum amaç fonksiyonu değerini veren çözümdür. Sonuncu çözüm yöneyi ise en kötü çözümdür;

$$N_p^{(0)}(x^1) \rightarrow \text{En iyi}$$

$$N_p^{(0)}(x^{n_p}) \rightarrow \text{En kötü}$$

Bu arada ilerleyen adımlarda kullanılmak üzere eldeki en iyi çözüm yöneyi ve amaç fonksiyon değeri kayıt altına alınır;

$$x^{en\ iyi} \leftarrow x^1$$

$$f^{en\ iyi} \leftarrow f^1$$

- ▶ 3. Sonraki işlem adımında envanter çözümlerine ait  $w_j, j = 1, \dots, n_p$  ağırlık değerleri hesaplanır. Ağırlık yöneyi hatırlanacağı üzere doğrudan envanter çözüm kümesinin büyüklüğüne ( $n_p$ ) ve seçim etmen katsayısına ( $q$ ) bağlı olup optimizasyon süreci boyunca değişmemektedir. Akabinde,



ağırlık değeri yöneyine bağlı olarak envanterdeki çözüm kümesi yöneylerinin tercih edilme olasılıklarını içeren  $p_j$  seçilme olasılık yöneyi elde edilir. Dikkat edileceği üzere tıpkı ağırlık yöneyinde olduğu üzere, ağırlık yöneyi değişmedikçe optimizasyon süreci boyunca seçilme olasılık yöneyi de değişmeden kullanılabilir.

- ▶ 4. Müteakiben tekrarlı işlemleri içeren eniyileme sürecine geçilir. Azami iterasyon sayısınca tekrar edilecek bu işlemlerden ilki, ortalama ve standart sapma yöneylerinin tespitidir. Ortalama yöneyleri ( $\mu^i$ ) envanter çözüm kümesi yöneyleri ile aynıdır. Bu adımda sadece standart sapma yöneyleri ( $\sigma_j^i$ ), envanter çözüm kümesi yöney ve yöneye ait değişken değerleri dikkate alınarak hesaplanır.
- ▶ 5. Ortalama ve standart sapma değerlerinin belirlenmesi sonrasında yeni çözümlerin üretilmesi gerekir. Bu meyanda başvurulan yöntem, örnekleme çözüm uzayının üretilmesidir. Üretimde etkin olan husus, envanterdeki çözümlerin olasılık değerine göre rastgele seçilmesi ve seçilen yöneye ait değışkende rassal olarak değışikliğe gidilmesidir;

$$N_s(x^{(a)}) \leftarrow \mu_j^r + \sigma_j^r \cdot \text{rand}, a = 1, \dots, n_s \quad (3.50)$$

Burada  $\sigma_j^r$  ifadesi, örneğin rulet tablası yaklaşımına göre rastgele seçilen  $r$ 'ninci envanter çözümüne ait  $j$ 'ninci boyut değerinin standart sapmasını,  $\mu_j^r$  aynı çözüme ait ortalama değeri veya diğer bir ifade ile çözüme ait değışken değerini göstermektedir. Görüldüğü üzere ortalama ve standart sapma değerleri esas itibari ile örnekleme çözüm uzayının üretilmesinde dikkate alınan değerler olup üretimde seçilme olasılığı yüksek olan bireylerin şansı daha fazladır. Sonraki adımda örnekleme çözüm uzayının amaç fonksiyon değerleri belirlenir. Bilahare envanter çözüm kümesi ile örnekleme çözüm kümesi birleştirilerek genel çözüm kümesi oluşturulur. Oluşturulan genel çözüm kümesi, amaç fonksiyon değerlerine göre yeniden küçükten büyüğe olacak şekilde sıralamaya tabi tutularak ilk  $n_p$  adedi güncel envanter çözüm kümesi olarak dikkate alınır;

$$\begin{aligned} N &\leftarrow [N_p \ N_s] \\ N &\leftarrow N(\text{sıra}) \\ N_p &\leftarrow N(1:n_p) \end{aligned}$$

Bir anlamda feromen güncelleme işlemlerinin yapıldığı adım sonrasında, yeni envanter çözümleri üzerinde 4üncü adım ve sonrası tekrarlı işlemlere devam edilir.



$$f(x) = \sum_{j=1}^n x_j^2$$

Komut<sup>(59)</sup>

MATLAB uygulaması



```
clc; clear; close all;
% Problem tanımı
CostFunction = @(x) Sphere(x); % amaç fonksiyonu
nVar=10; % değişken sayısı
VarSize=[1 nVar];
VarMin=-10; % değişken alt sınırı
VarMax= 10; % değişken üst sınırı
% Algoritma parametreleri
MaxIt = 100 ; % Azami iterasyon sayısı
nPop = 10 ; % envanter çözüm kümesi büyüklüğü
nSample = 40 ; % örnekleme çözüm kümesi büyüklüğü
q = 0.5 ; % seçim etmen katsayısı
gama = 1 ; % standart sapma katsayısı
% Başlangıç işlemleri
empty_individual.Position = [] ;
empty_individual.Cost = [] ;
% envanter çözüm kümesinin oluşturulması
Pop = repmat(empty_individual,nPop,1) ;
for i = 1:nPop
    pop(i).Position = unifrnd(VarMin,VarMax,VarSize) ;
    pop(i).Cost = CostFunction(pop(i).Position) ;
end
% sıralama işlemi
 [~, SortOrder] = sort([pop.Cost]) ;
pop = pop(SortOrder) ;
% en iyi çözümün kaydedilmesi
BestSol = pop(1) ;
% en iyi amaç fonksiyon değeri
BestCost=zeros(MaxIt,1);
% ağırlık katsayılarının hesaplanması
w=1/(sqrt(2*pi)*q*nPop)*exp(-0.5*((1:nPop)-1)/(q*nPop)).^2);
% seçilme olasılık değerlerinin hesaplanması
p=w/sum(w);
% İterasyon işlemleri başlangıcı
for it=1:MaxIt
    % ortalama değerler
    s=zeros(nPop,nVar);
    for l=1:nPop
        s(l,:)=pop(l).Position;
    end
    % Standart sapma değerleri
    sigma=zeros(nPop,nVar);
    for l=1:nPop
```





```
D=0;
for r=1:nPop
    D=D+abs(s(l,:)-s(r,:));
end
sigma(l,:)=gama*D/(nPop-1);
end
% Örneklem çözüm uzayının eldesi
newpop=repmat(empty_individual,nSample,1);
for t=1:nSample
    newpop(t).Position=zeros(VarSize);
    for i=1:nVar
        l=RouletteWheelSelection(p);
        newpop(t).Position(i)=s(l,i)+sigma(l,i)*randn;
    end
    % amaç fonksiyon değeri hesabı
    newpop(t).Cost=CostFunction(newpop(t).Position);
end
% genel çözüm kümesi teşkili
pop = [pop; newpop];
% genel sıralama
 [~, SortOrder] = sort([pop.Cost]);
Pop = pop(SortOrder);
% envanter çözüm kümesinin güncellenmesi
pop = pop(1:nPop);
% en iyi çözüm güncellemesi
BestSol = pop(1);
% en iyi amaç fonksiyonu güncellemesi
BestCost(it) = BestSol.Cost;
end

% sonuçlar
BestSol.Position
BestSol.Cost

figure;
semilogy(BestCost,'LineWidth',2);
xlabel('Iterasyon'); ylabel('En iyi amaç fonksiyon degeri'); grid on;
% amaç fonksiyonu
function z = Sphere(x)
    z = sum(x.^2);
end
% rulet tablası
function j=RouletteWheelSelection(P)
    r = rand;
    C = cumsum(P) ;
    j = find(r<=C,1,'first');
end
```



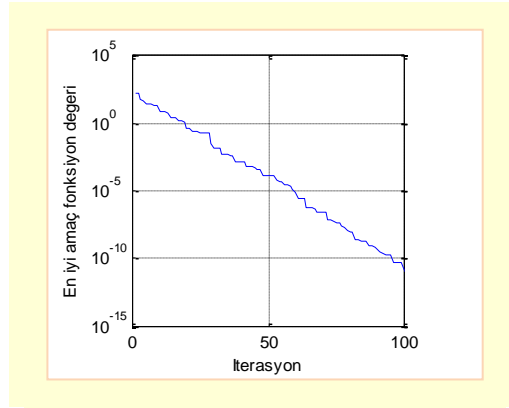




## Sonuç

```
ans =  
1.0e-05 *  
0.0205 0.1196 0.2131 0.1647 -0.1224 -0.0539 -0.0957 0.1455  
-0.0467 0.0346
```

```
ans =  
1.3886e-11
```



Amaç fonksiyon değerinin iterasyonlar sürecinde değişimi.



### 3.8 Yapay arı sürüsü algoritması

Yapay arı sürüsü (YAS) algoritmaları, tıpkı karınca koloni algoritmalarında olduğu üzere esin kaynağı olarak doğada sürü halinde yaşayan böcekleri, özelde bal arılarını ve onların davranışlarını temel alan yöntemler grubudur. Arılar, günümüzden yaklaşık 100 milyon yıl öncesi zaman diliminde yeryüzünde görülmeye başlanan ve sosyal böcek türleri arasındaki “Apoidea” ailesine üye böcek türlerindedir. O dönemde söz konusu arıların böcek avcısı olarak yaşamlarını sürdürdükleri düşünülmektedir.

Günümüzde arılara ait bilinen 20,000’den fazla alt tür bulunmaktadır. Genellikle koloniler halinde yaşayan bu canlıların koloni büyüklüğü, mevsimlere ve mevsim şartlarına göre değişmekle beraber kış aylarında 20,000 civarı, yaz aylarında ise 60,000 civarı bireyden oluşabilir. Arıların ebatları da türlerine göre değişiklik göstermekle beraber yaşayan arılar 2 mm ile devasa boyutlardaki 39 mm uzunluğa kadar erişebilmektedirler. Karıncalarla karşılaştırıldığında üçüncü boyutu da kullanabilen arılar, kanatlarını saniyede 200’den fazla çırpabilmektedirler<sup>(60)</sup>.

Kovanlarında koloni halinde yaşayan arıların üç cinsinden bahsedilebilir; kraliçe arı, erkek ve işçi arılar<sup>i</sup>. Arı kolonisinde bir tane kraliçe arı, birkaç yüz tane mertebesinde erkek arı ve binler mertebesinde işçi arı bulunur. İşçi arılar, cinsi itibarı ile dişi arı olmakla beraber üreme yetilerinden mahrumdurlar. Görevleri oldukça geniş bir yelpazeye yayılı işçi arılarının<sup>iii</sup> ana gayesi, bulunan besin kaynaklarından yuvaya yiyecek taşımaktır. Bununla beraber besin kaynağının bulunması ve yuvaya taşınması sadedinde girift bir görev dağılımından bahsedilebilir. Yuva etrafındaki besin kaynaklarının bulunması görevi, kâşif arılar<sup>iv</sup> olarak adlandırılan işçi arılara ait bir görevdir. Buna karşılık yiyeceği yuvaya getiren işçi arıları gözlemleyerek göreve çıkmaya karar veren arılara ise gözcü arı<sup>v</sup> denmektedir. İşçi arının besin durumuna göre gözcü arı olması veya gözcü iken kâşif arı olması mümkündür. İşçi arılar, besin kaynağına tekrar gitmeden önce yuvadaki arıların arasında dolaşır ve bir takım hareketler sergileyerek gözcü niteliğindeki diğer arılarla bilgi paylaşımında bulunurlar. Söz konusu bilgi paylaşımı, besinin konumu ve kalitesi ile ilgili verileri içerebilir<sup>(61)</sup>. Yüksek kalitede veya bol miktarda besin olması durumunda, kâşif arı görevindeki işçi arının sergilediği sallama dansı<sup>vi</sup> daha uzun sürelidir. Dolayısıyla gözcü arıların

<sup>i</sup> Artificial bee colony (ABC)

<sup>ii</sup> Queen bee, drones, worker bees

<sup>iii</sup> “Worker” veya “forager bees”

<sup>iv</sup> Scout bees

<sup>v</sup> Onlooker bees

<sup>vi</sup> Waggle dance



Şekil 3.26 Bal arısı cinsleri: sırası ile erkek, kraliçe ve işçi arı.



ikna olmaları ve belirtilen besin kaynağını tercih etme olasılıkları daha yüksektir. Şekil 3.27(a)'da besin kaynağı hakkında bilgi veren kâşif arının sergilediği sallama dansı ve onu gözlemleyen gözcü arılar, şeklin diğer seçeneğinde (b) ise sergilenen dansın anlamı görülmektedir. Buna göre dansın doğrultusu, güneşe göre besin kaynağının yönünü temsil etmektedir.



**Şekil 3.27** (a) Kâşif arının sergilediği sallama dansı, (b) bu dansın güneşe göre besin kaynağının yönünü gösteren anlamı.

Arıların davranışları üzerine yapılan gözlemler oldukça eskilere, 1800'lü yıllara kadar gider. Ancak gözlemlenen davranışlara dayalı olarak hesaplamalı yöntemlerin geliştirilmesi 1990'lı yıllara rastlar. Arılar pek çok davranış şekli sergilemekte olup söz konusu davranış şekilleri arasında iletişim aracı olarak danslar, yiyecek arama yöntemleri, kraliçe arı ve görev paylaşimleri, kolektif karar verme yetileri, yuva yeri seçimi, çiftleşme ve seyrüsefer yöntemleri sayılabilir. 1993 yılında A. Walker ile meslektaşları tarafından kaleme alınan ve robot hareketlerinin arılardaki seyrüsefer davranışlarını dikkate alarak modellenmesi içerikli çalışma, literatürdeki ilk sistematik çalışma olarak düşünülebilir. Sonraki dönemde yapılan çalışmalar arasında çokça dikkate alınan ve üzerinde durulan yöntem ise bir Türk bilim adamı olan D. Karaboğa'nın geliştirdiği yapay arı kolonisi algoritmasıdır<sup>(64)</sup>. Arıların yiyecek arama yöntemlerinden esinlenilerek geliştirilen yöntem 2005 yılında "sayısal optimizasyonda bal arısı sürüsüne dayalı düşünce" isimli teknik raporla literatüre tanıtılmış ve sonraki yıllarda giderek yaygınlaşmıştır. 2009 yılında yayımlanan başka bir çalışmada<sup>(65)</sup> standart tanımları çerçevesinde yapay arı sürüsü algoritması ile genetik algoritma, parçacık sürü algoritması ve benzeri diğer sezgisel yöntemlerle karşılaştırma yapılmış ve daha az kontrol parametresine sahip yapay arı sürüsü algoritmasının daha başarılı olduğu beyan edilmiştir. Literatüre girdiği dönem itibari ile gerçek sayılara dayalı kısıtsız optimizasyon sorunlarının çözümü için teklif edilen YAS algoritması zaman içerisinde kısıtlı optimizasyon problemlerine<sup>(66)</sup>, ayırık nitelikli sorunlara da uygulanmıştır. Ayrıca nüfus bazlı yöntem olması hasebiyle YAS yöntemi, paralel hesaplama imkânları kullanılarak daha hızlı sonuçlar veren



seçeneklerle de desteklenmektedir. Yapay arı sürüsü algoritmasının işlem adımları çerçevesinde dört ana başlığın varlığından bahsedilebilir. Söz konusu işlem başlıkları; başlangıçta rastgele bir şekilde besin kaynaklarının belirlenmesi, işçi arılarının belirlenen besin kaynakları ve civarına gönderilmesi, görevden dönen arıların yaptıkları danslara göre gözlemlenmesi ve gözcü arılarının yiyecek kaynağının niteliğine bağlı olarak rassal şekilde göreve gönderilmesi, yiyeceği tüketilen besin kaynaklarının terk edilmesi ve kâşif arıların yeni besin kaynakları arayışına başlaması şeklinde sıralanabilir. Böyle bir yaklaşımın eniyileme sorununa indirgenmesi sadedinde besin kaynağının konumu, soruna çözüm olarak önerilen tasarım yöneyini, besin kaynağının niteliği ise amaç fonksiyonunun değeri temsil etmektedir. İşçi ile gözcü arıların besin kaynağına yönelik değerlendirmelerini yerel çözüm arayışları, kâşif arıların yeni besin kaynaklarına yönelmelerini ise genel çözüm arayışı olarak düşünmek mümkündür.

Yapay arı sürüsü algoritmasının işlem adımlarını aşağıdaki gibi sıralamak mümkündür;

- ▶ 1. Öncelikle soruna ait tasarım parametre sınır değerleri  $(x_j^l, x_j^u; j = 1, 2, \dots, n)$ , azami iterasyon sayısı  $(T)$ , sürüdeki işçi arı sayısı  $(n_p)$ , aynı sürüdeki gözcü arı sayısı  $(n_g)$ , besin kaynağının terk edilmesi amacıyla dikkate alınan azami deneme sayısı  $(L)$  gibi yonteme ait sabit değerler belirlenir. Uğranılan besin kaynağının daha iyi çözümler vaat etmediği zamanlarda dikkate alınan azami deneme sayısının belirlenebilmesi için;

$$L = \alpha \cdot n \cdot n_p \quad (3.51)$$

ifadesi dikkate alınabilir. Burada  $\alpha$  sabit sayı olup kullanıcının takdirine göre belirlenir.

- ▶ 2. Daha sonra başlangıç sürüsünün üretilmesi safhasına geçilir. Başlangıç nüfusu genellikle tasarım parametre sınır değerleri ile rassal sayıların dikkate alındığı;

$$x_j^i = x_j^l + (x_j^u - x_j^l) \cdot \text{rand}(0,1), \quad i=1,2,\dots,n_p, \quad j=1,2,\dots,n \quad (3.52)$$

denkleminde göre belirlenir. Başlangıç kolonisi aynı zamanda başlangıçta dikkate alınan olası besin kaynaklarının konumunu da göstermekte olup müteakiben belirlenen konumların besin değerlerinin, ya da diğer bir ifade ile amaç fonksiyon değerlerinin tespit edilmesi işlemleri yapılır;

$$f^i = f(x^i)$$





- 3. Başlangıç nüfusu ile amaç fonksiyonu değerlerinin hesaplanması sonrasında, yakınsama odaklı tekrarlı işlemlere geçilir. Söz konusu tekrarlı işlemlerden ilki, eldeki besin konumlarının rassal olarak derlenmesi ve daha iyi bir sonuç elde edilmesi halinde mevcut konumun yeni konum ile güncellenmesidir. Konum derleme işlemi;

$$x_j^{y,i} = x_j^i + \Phi_j^i (x_j^i - x_j^k), \quad i=1,2,\dots,n_p, \quad j=1,2,\dots,n, \quad k \neq i$$

$$k \in \text{rand}\{1,2,\dots,n_p\}, \quad k \neq i$$

ifadesine göre gerçekleştirilir. Burada  $\Phi_j^i$  değeri;

$$\Phi_j^i = \beta \cdot \text{rand}(-1,1)$$

eşitliği ile belirlenen rassal bir katsayıdır. Eşitlikte geçen  $\beta$ , ivmelenme katsayısı olarak adlandırılan ve kullanıcı tarafından belirlenen sabit bir sayıdır. Derleme işleminde geçen  $x_j^k$  ifadesi, rastgele şekilde belirlenen  $k$ 'ninci konuma ait  $j$ 'ninci değişken değerini göstermektedir. Buna göre sırası ile her konum rassal olarak belirlenen diğer bir konum ile derlenmekte ve yeni bir olası konum,  $x^{y,i}$  elde edilmektedir. Elde edilen yeni olası konumun;

$$f(x^{y,i}) \leq f(x^i)$$

şartını sağlaması halinde;

$$x^i \leftarrow x^{y,i}$$

güncellemesine gidilir. Aksi halde mevcut konum korunur. Bu aşamada yapılan diğer bir işlem, güncel konumların uygunluk değerleri ile seçilme olasılık değerlerinin hesaplanmasıdır. Her bir konuma ait uygunluk değeri, ortalama ve bireye ait amaç fonksiyon değerine göre belirlenebilir;

$$\bar{f} = \frac{1}{n_p} \sum_{i=1}^{n_p} f^i$$

$$F^i = e^{-\frac{f^i}{\bar{f}}}$$

Burada  $\bar{f}$ , o nesle ait ortalama amaç fonksiyon değerini ve  $F^i$  her bireye ait uygunluk değerini temsil etmektedir. Literatürde farklı uygunluk değeri hesaplama tanımları mevcut olup örnek olma meyanında,



Şekil 3.28 Bal arısı kovanı.



$$F^i = \begin{cases} \frac{1}{1+f^i} & f^i \geq 0 \\ \frac{1}{1+|f^i|} & f^i < 0 \end{cases} \quad (3.59)$$

yaklaşımı dikkate alınabilir. Uygunluk değerlerine bağlı olarak seçilme olasılık değerleri;

$$p^i = \frac{F^i}{\sum_{i=1}^{n_p} F^i} \quad (3.60)$$

şeklinde elde edilir. Seçilme olasılık değerleri, gözcü arıların güncel besin kaynaklarını tercih etmesinde önemli bir rol oynar. Zira seçilme olasılığı yüksek olan besin kaynağının, daha çok gözcü tarafından ziyaret edilmesi mümkündür.

- 4. Tekrarlı işlemlerin müteakip basamağında gözcü arıların konum seçimi işlemleri gerçekleştirilir. Bir önceki adımda sıra ile ama rassal olarak yapılan güncelleme işlemleri, bu adımda seçilme olasılığına göre rassal olarak yapılır. Dolayısıyla seçilme olasılığı yüksek olan konumların, rassal çerçevede birden fazla seçilmesi ve rassal olarak belirlenen bir başka konum ile güncellenmesi mümkündür. Güncellenecek konum genellikle rulet tablası yaklaşımı ile belirlenir;

$$\begin{aligned} x_j^{y,i} &= x_j^i + \Phi_j^i(x_j^i - x_j^k) \\ i &\leftarrow \text{rulet}(P) \\ j &= 1, 2, \dots, n \\ i &\in \{1, 2, \dots, n_p\} \\ k &\in \text{rand}\{1, 2, \dots, n_p\}, \quad i \neq k \end{aligned} \quad (3.61)$$

Burada  $x^i$  rulet tablasına göre belirlenen konumu,  $x^k$  rastgele belirlenen konumu göstermektedir. Elde edilen yeni konumun;

$$f(x^{y,i}) \leq f(x^i) \quad (3.62)$$

şartını sağlaması halinde;

$$x^i \leftarrow x^{y,i} \quad (3.63)$$

ataması yapılarak ikinci defa konum güncellemesi gerçekleştirilmiş olur.



- 5. Tekrarlı işlemlerin üçüncü fazında kâşif arıların görevlendirilmesi adımına geçilir. Belirli sayıda ziyaret yapılmış olmasına rağmen amaç fonksiyonunda iyileşme sağlanamayan konumlar, tıpkı başlangıç nüfusunun oluşturulması faslında olduğu gibi yeniden tanımlanır;

$$x_j^p = x_j^l + (x_j^u - x_j^l) \cdot \text{rand}(0,1) \quad (3.64)$$

$$j = 1, 2, \dots, n$$

$$\theta(p) \geq L$$

Burada  $\theta(p)$ ,  $p$  konumuna ait başarısız geçen ziyaret ya da diğer bir ifade ile deneme sayısını göstermektedir.

- 6. Tekrarlı işlemlerin son adımında yakınsama kontrolü yapılır. Yakınsamada azami iterasyon sayısı veya amaç fonksiyonu değeri bazlı ölçütler dikkate alınabilir.

Yapay arı sürüsü algoritmasının daha iyi anlaşılması için örnek bir sorunu bahse konu yöntemle çözmek mümkündür. Buna göre;

$$\min_x f(x) = \sum_{i=1}^5 x_i^2$$

şeklinde ifade edilen beş değişkenli küre fonksiyonunun YAS yöntemi ile minimizasyonu çerçevesinde; öncelikle tasarım yöneyi değişkenlerine ait alt ve üst sınır değerlerini  $[-10 \ 10]$ , azami iterasyon sayısını  $T = 100$ , sürüdeki işçi arı sayısını  $n_p = 10$ , aynı sürüdeki gözcü arı sayısı  $n_g = 10$ , besin kaynağının terk edilmesi amacıyla dikkate alınan azami deneme sayısını;

$$L = \alpha \cdot n \cdot n_p = 0.6 \times 10 \times 10 = 60$$

ivmelenme katsayısını  $\beta = 1$  şeklinde belirleyebiliriz. Akabinde başlangıç sürüsünün üretilmesi safhasına geçilir. Başlangıç nüfusu genellikle tasarım parametre sınır değerleri ile rassal sayıların dikkate alındığı;

$$x_j^i = -10 + (10 - (-10)) \cdot \text{rand}(0,1), \quad \begin{matrix} i=1,2,\dots,10 \\ j=1,2,\dots,5 \end{matrix}$$

denklemine göre belirlenebilir. Buna göre rassal sayılara bağlı olarak başlangıç sürüsünü;



$$\begin{aligned}
 x^1 &= [ -4.4795 & 3.5941 & 3.1020 & -6.7478 & -7.6200 ] \\
 x^2 &= [ -0.0327 & 9.1949 & -3.1923 & 1.7054 & -5.5238 ] \\
 x^3 &= [ 5.0253 & -4.8981 & 0.1191 & 3.9815 & 7.8181 ] \\
 x^4 &= [ 9.1858 & 0.9443 & -7.2275 & -7.0141 & -4.8498 ] \\
 x^5 &= [ 6.8143 & -4.9144 & 6.2857 & -5.1295 & 8.5853 ] \\
 x^6 &= [ -3.0003 & -6.0681 & -4.9783 & 2.3209 & -0.5342 ] \\
 x^7 &= [ -2.9668 & 6.6166 & 1.7053 & 0.9945 & 8.3439 ] \\
 x^8 &= [ -4.2832 & 5.1440 & 5.0746 & -2.3911 & 1.3564 ] \\
 x^9 &= [ -8.4829 & -8.9210 & 0.6160 & 5.5833 & 8.6802 ] \\
 x^{10} &= [ -7.4019 & 1.3765 & -0.6122 & -9.7620 & -3.2575 ]
 \end{aligned}$$

şeklinde ifade etmek mümkündür. Müteakiben belirlenen konumların amaç fonksiyon değerlerinin tespit edilmesi işlemleri yapılır;

$$f = f(x^i)$$

$$f = [146.2027 \quad 128.1577 \quad 126.2344 \quad 210.2269 \quad 210.1149 \quad 76.2794 \quad 126.0982 \quad 78.1153 \quad 258.4433 \quad 162.9647]$$

Buna göre en iyi konum, 6ncı konum olup amaç fonksiyon değeri 76.2794 şeklindedir. Başlangıç nüfusu ile amaç fonksiyonu değerlerinin hesaplanması sonrasında, besin konumlarının rassal olarak derlenmesi ve daha iyi bir sonuç elde edilmesi halinde mevcut konumun yeni konum ile güncellenmesi işlemlerine geçilir. Konum derleme işlemi çerçevesinde ilk konuma ait rassal sayı değerleri;

$$\Phi_j^i = \beta \cdot \text{rand}(-1,1) = 1 \times \text{rand}(-1,1)$$

$$\Phi^1 = [0.5886 \quad -0.3776 \quad 0.0571 \quad -0.6687 \quad 0.2040]$$

şeklinde elde edilip, ilk konumun derleneceği diğer konum da rassal yaklaşımla;

$$k \in \text{rand}\{2, \dots, 10\}$$

$$k = 3$$

olarak belirlenebilir. Akabinde ilk konuma ait derleme işlemi;

$$x_j^{y,1} = x_j^1 + \Phi_j^1(x_j^1 - x_j^3), \quad j = 1, 2, \dots, n$$

ifadesine göre gerçekleştirilir;

$$x^{y,1} = [-10.0738 \quad 0.3877 \quad 3.2722 \quad 0.4269 \quad -10.7689]$$





Konum derleme sonrasında yeni konuma ait amaç fonksiyon değeri belirlenir;

$$f(x^{y,1}) = 228.4886$$

Elde edilen bu değerin derleme öncesi konuma ait amaç fonksiyon değeri ile karşılaştırılması neticesinde;

$$f(x^{y,1}) = 228.4886 > f(x^1) = 146.2027$$

olduğu görülebilir. Buna göre derleme daha iyi bir sonuç vermemiş olup eski konum aynen muhafaza edilir. Bu arada ilk konuma ait başarısız geçen ziyaret sayısı da;

$$\theta(1) = 1$$

olarak kayıt altına alınmış olur. Benzer işlemler sırası ile diğer konumlar için de gerçekleştirilirse güncel konumlar yine rassal sayılara bağlı olarak;

$$\begin{array}{l} \mathbf{x}^1 = [ -4.4795 \quad 3.5941 \quad 3.1020 \quad -6.7478 \quad -7.6200 ] \\ \mathbf{x}^2 = [ -0.0327 \quad 9.1949 \quad -3.1923 \quad 1.7054 \quad -5.5238 ] \\ \mathbf{x}^3 = [ 5.0253 \quad -4.8981 \quad 0.1191 \quad 3.9815 \quad 7.8181 ] \\ \mathbf{x}^4 = [ 9.1858 \quad 0.9443 \quad -7.2275 \quad -7.0141 \quad -4.8498 ] \\ \mathbf{x}^5 = [ 6.8143 \quad -4.9144 \quad 6.2857 \quad -5.1295 \quad 8.5853 ] \\ \mathbf{x}^6 = [ -3.0003 \quad -6.0681 \quad -4.9783 \quad 2.3209 \quad -0.5342 ] \\ \mathbf{x}^7 = [ -2.9668 \quad 6.6166 \quad 1.7053 \quad 0.9945 \quad 8.3439 ] \\ \mathbf{x}^8 = [ -4.2832 \quad 5.1440 \quad 5.0746 \quad -2.3911 \quad 1.3564 ] \\ \mathbf{x}^9 = [ -8.4829 \quad -8.9210 \quad 0.6160 \quad 5.5833 \quad 8.6802 ] \\ \mathbf{x}^{10} = [ -7.4019 \quad 1.3765 \quad -0.6122 \quad -9.7620 \quad -3.2575 ] \end{array}$$

şeklinde yazılabilir. Maalesef yapılan ziyaretlerin hiç birinde, rastlantı eseri başarılı bir derleme gerçekleştirilememiştir. Bu arada konumlara göre başarısız geçen ziyaret sayıları da;

$$\theta = [1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1]$$

olarak kayıt altına alınmış olur. Sonraki aşamada güncel konumların uygunluk değerleri ile seçilme olasılık değerleri hesaplanır. Buna göre amaç fonksiyonu değerleri ortalaması;

$$\bar{f} = \frac{1}{10} \sum_{i=1}^{10} f^i = 152.2837$$



olarak elde edilip, uygunluk değerleri de;

$$F^i = e^{-\frac{f^i}{152.2837}}, \quad i = 1, 2, \dots, 10$$

$$F = [0.3829 \quad 0.4310 \quad 0.4365 \quad 0.2515 \quad 0.2516 \quad 0.6060 \quad 0.4369 \quad 0.5987 \quad 0.1832 \quad 0.3430]$$

şeklinde bulunabilir. Bu değerler ışığında seçilme olasılık büyüklükleri;

$$p^i = \frac{F^i}{\sum_{i=1}^{10} F^i}$$

$$P = [0.0976 \quad 0.1099 \quad 0.1113 \quad 0.0641 \quad 0.0642 \quad 0.1545 \quad 0.1114 \quad 0.1527 \quad 0.0467 \quad 0.0875]$$

biçiminde elde edilir. Bu dağılıma göre seçilme olasılığı en fazla olan besin kaynağı, 6ncı konumdur. Tekrarlı işlemlerin müteakip basamağında gözcü arıların konum seçimi işlemleri gerçekleştirilir. Bu aşamada önce rulet tablası yardımı ile olası besin kaynağı seçilir;

$$8 \leftarrow \text{rulet}(P)$$

Daha sonra seçili konum ile derlenecek ikinci konum rassal olarak belirlenir;

$$k \in \text{rand}\{1, 2, \dots, 10\}, \quad k \neq 8$$

$$k = 4$$

Konum güncelleme işlemi çerçevesinde 8inci konuma ait rassal ayı değerleri;

$$\Phi_j^8 = \beta \cdot \text{rand}(-1, 1) = 1 \times \text{rand}(-1, 1)$$

$$\Phi^8 = [-0.5166 \quad -0.1922 \quad -0.8071 \quad -0.7361 \quad 0.8841]$$

şeklinde elde edilip,

$$x_j^{y,8} = x_j^8 + \Phi_j^8(x_j^8 - x_j^4)$$

denklemine göre yeni konum;

$$x^{y,8} = [2.6751 \quad 4.3369 \quad -4.8543 \quad -5.7939 \quad 6.8434]$$

olarak belirlenmiş olur. Konum belirleme sonrasında yeni konuma ait amaç fonksiyon değeri belirlenir;



$$f(x^{y,8}) = 129.9309$$

Bu değerin, derleme öncesi konuma ait amaç fonksiyon değeri ile karşılaştırılması neticesinde;

$$f(x^{y,8}) = 129.9309 > f(x^8) = 78.1153$$

olduğu görülebilir. Buna göre gözcü arının seçimi daha iyi bir sonuç vermemiş olup eski konum aynen muhafaza edilir. Bu arada ilk konuma ait başarısız geçen ziyaret sayısı da;

$$\theta(8) = 2$$

olarak kayıt altına alınmış olur. Benzer işlemler sırası ile diğer konumlar için de gerçekleştirilirse güncel konumlar;

$x^1 =$	[	-4.4795	3.5941	3.1020	-6.7478	-7.6200	]
$x^2 =$	[	-0.0327	9.1949	-3.1923	1.7054	-5.5238	]
$x^3 =$	[	2.3491	-2.4380	0.7383	6.7941	4.2110	]
$x^4 =$	[	9.1858	0.9443	-7.2275	-7.0141	-4.8498	]
$x^5 =$	[	6.8143	-4.9144	6.2857	-5.1295	8.5853	]
$x^6 =$	[	-1.8207	-5.1805	-4.0156	2.7448	-3.5802	]
$x^7 =$	[	-2.9668	6.6166	1.7053	0.9945	8.3439	]
$x^8 =$	[	-4.2832	5.1440	5.0746	-2.3911	1.3564	]
$x^9 =$	[	-8.4829	-8.9210	0.6160	5.5833	8.6802	]
$x^{10} =$	[	-7.4019	1.3765	-0.6122	-9.7620	-3.2575	]

şeklinde yazılabilir. Elde edilen konumların değerlendirilmesi neticesinde, yapılan ziyaretlerin ikisinde iyileşme sağlanarak güncellemeye gidilmiş, geri kalan ziyaretlerde ise başarısız olunmuştur. Buna göre başarısız geçen ziyaret sayıları;

$$\theta = [2 \ 2 \ 2 \ 1 \ 1 \ 1 \ 1 \ 1 \ 4 \ 1 \ 2]$$

olarak kayıt altına alınmış olur. Bu arada güncel sürüye ait amaç fonksiyon değerleri;

$$f = [146.2027 \ 128.1577 \ 75.8998 \ 210.2269 \ 210.1149 \ 66.6291 \ 126.0982 \ 78.1153 \ 258.4433 \ 162.9647]$$

şeklinde sıralanabilir. İlk iterasyon neticesinde elde edilen en iyi konum, 6ncı konum olup en iyi amaç fonksiyon değeri de 66.6291 şeklindedir. Gözcü arı ziyaretleri sonrasında kâşif arıların görevlendirilmesi adımına geçilir. Buna göre;





$$\theta(i) < L = 60, i = 1, 2, \dots, 10$$

olması nedeniyle herhangi bir kaşif arı görevlendirilmesi yapılmaz. Tekrarlı işlemlerin son adımında yakınsama kontrolü yapılır. Bulunulan aşamada henüz azami iterasyon sayısına ulaşılmaması nedeniyle tekrar başa dönülüp sırasıyla tüm konumlar birbirleri ile derleme işlemlerine tabi tutulur ve benzer işlemlere devam edilir.

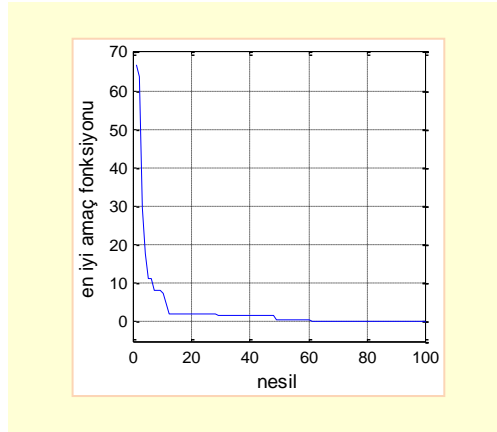
Şekil 3.29'da iterasyon adımları ve elde edilen en iyi amaç fonksiyon değerleri görülmektedir. Buna göre 100'üncü iterasyon sonrasında elde edilen en iyi amaç fonksiyon değeri;

$$f^* = 0.3053$$

olup bu değeri sağlayan tasarım yöneyi;

$$\mathbf{x}^* = [0.0622 \quad 0.0594 \quad -0.0845 \quad -0.3635 \quad 0.3983]$$

şeklindedir.



Şekil 3.29 Yakınsama sürecinde amaç fonksiyonu değişimi.



$$\min_x f(x) = \sum_{j=1}^n x_j^2$$

Komut (67)

MATLAB uygulaması



```
clc; clear; close all;
%% Problem tanımı
CostFunction = @(x) Sphere(x); % amaç fonksiyonu
nVar = 5; % değişken sayısı
VarSize=[1 nVar];
VarMin=-10; % alt sınır değeri
VarMax= 10; % üst sınır değeri
%% yöntem parametreleri
MaxIt=1000; % azami iterasyon sayısı
nPop=10; % sürü büyüklüğü
nOnlooker=nPop; % gözcü sayısı
L=round(0.6*nVar*nPop); % deneme sınır değeri
a=1; % ivmelenme katsayısı
%% Başlangıç
empty_bee.Position=[];
empty_bee.Cost=[];
pop=repmat(empty_bee,nPop,1);
BestSol.Cost=inf;
BestCost=zeros(MaxIt,1);
% deneme sınır değeri kaydı
C = zeros(nPop,1);
% başlangıç sürüsünün eldesi
for i=1:nPop
    pop(i).Position=unifrnd(VarMin,VarMax,VarSize);
    pop(i).Cost=CostFunction(pop(i).Position);
    if pop(i).Cost<=BestSol.Cost
        BestSol=pop(i);
    end
end
%% YAS tekrarlı işlemleri
for it=1:MaxIt
    for i=1:nPop
        K=[1:i-1 i+1:nPop];
        k=K(randi([1 numel(K)]));
        phi=a*unifrnd(-1,+1,VarSize);
        newbee.Position=pop(i).Position+phi.*(pop(i).Position-pop(k).Position);
        newbee.Cost=CostFunction(newbee.Position);
        if newbee.Cost<=pop(i).Cost
            pop(i)=newbee;
        else
            C(i)=C(i)+1;
        end
    end
end
```





```
% uygunluk deęeri ve seilme olasılıęı hesabı
F=zeros(nPop,1);
MeanCost = mean([pop.Cost]);
for i=1:nPop
    F(i) = exp(-pop(i).Cost/MeanCost); % Convert Cost to Fitness
end
P=F/sum(F);
% Gözcü arı işlemleri
for m=1:nOnlooker
    i=RouletteWheelSelection(P);
    K=[1:i-1 i+1:nPop];
    k=K(randi([1 numel(K)]));
    phi=a*unifrnd(-1,+1,VarSize);
    newbee.Position=pop(i).Position+phi.*(pop(i).Position-pop(k).Position);
    newbee.Cost=CostFunction(newbee.Position);
    if newbee.Cost<=pop(i).Cost
        pop(i)=newbee;
    else
        C(i)=C(i)+1;
    end
end
% kâşif arı işlemleri
for i=1:nPop
    if C(i)>=L
        pop(i).Position=unifrnd(VarMin,VarMax,VarSize);
        pop(i).Cost=CostFunction(pop(i).Position);
        C(i)=0;
    end
end
% en iyi konum kaydı
for i=1:nPop
    if pop(i).Cost<=BestSol.Cost
        BestSol=pop(i);
    end
end
BestCost(it)=BestSol.Cost;
End

%% sonuç
BestCost(it)
BestSol.Position

figure;
plot(BestCost,'LineWidth',2);
semilogy(BestCost,'LineWidth',2);
xlabel('Iteration');
ylabel('Best Cost');
grid on; axis square
% amaç fonksiyonu
function z = Sphere(x)
```



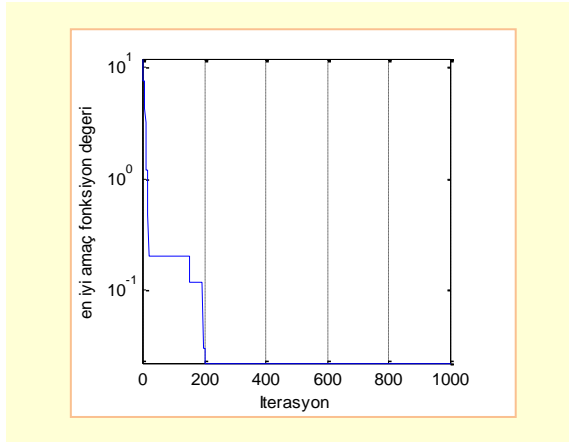


```
z = sum(x.^2);  
end  
  
% rulet tablası  
function j=RouletteWheelSelection(P)  
    r = rand ;  
    C = cumsum(P) ;  
    j = find(r<=C,1,'first');  
end
```



Sonuç

```
ans =  
    0.0220  
  
ans =  
-0.0076    0.0255   -0.0669   -0.1234    0.0398
```



Amaç fonksiyon değerinin iterasyonlar sürecinde değişimi.



### 3.9 Parçacık sürü algoritması

Parçacık sürü optimizasyon (PSO) algoritması, 1995 yılında J. Kennedy ve R. Eberhart tarafından uluslararası konferansta sunulan “Parçacık Sürü optimizasyonu” başlıklı bildiri ile literatüre kazandırılan bir algoritmadır<sup>(68)</sup>. Söz konusu bildiride yöntemin ana adımları ifade edilmekte ve ayrıca yapay sinir ağlarında ağırlık katsayılarının tespiti ile taraması zor bir test fonksiyonunda optimal noktanın bulunması uygulamaları sunulmaktadır. Yaklaşımın temelinde, tıpkı daha önceki yöntemlerde de olduğu üzere, doğadaki canlıların davranışlarından esinlenerek çözüm üretme düşüncesi yatmaktadır. Bu meyanda dikkate alınan davranış şekilleri, sosyal biyoloji biliminin temelini oluşturan canlılardaki sosyal davranışların irdelenmesi ve elde edilen bulgulardan yapılan çıkarsamalarıdır. 1975 yılında E. O. Wilson tarafından yayımlanan “Sosyobioloji: yeni sentez”<sup>ii</sup> isimli kitap çalışması esasen PSO yönteminin geliştirilmesinde etkili olan belki de en önemli kaynak niteliğindedir. Sonraki yıllarda C. W. Reynolds (1987) ve F. Heppner ile U. Grenander (1990) ikilisinin, özellikle kuşların birlikte gerçekleştirdikleri uçuşları incelediklerini ve bazı simülasyonları gerçekleştirdiklerini görmekteyiz. Kuş sürüsü<sup>iii</sup> olarak tanımlanan kavram, grup halinde hareket eden kuşların birbirine benzer hızlarda, yaklaşık aynı doğrultuda ve nispeten yakın mesafede uçmalarını ifade etmektedir. Doğada kuşların birkaç bireyden veya bazen onbinlercesinden oluşan sürüler halinde uçtukları veya besin aradığı bilinmektedir<sup>(70)</sup>. Gözlemlere dayalı olarak yapılan davranış modellemelerinde özellikle Reynolds tarafından geliştirilen “Kuşlar”<sup>iv</sup> isimli benzeşiminin, günümüzde de hala yaygın olarak kullanıldığını söylemek mümkündür<sup>(71)</sup>. Reynolds’a göre sürü halinde hareket eden kuşlarda üç temel davranış şekli gözlemlenmekte olup bunlar; belirli bir uzay ortamında kalabalıklaşmayı önleyen ayrılma davranışı, yakın bölgedeki diğer kuşların ortalama uçuş yönüne göre doğrultu belirlemeyi ifade eden hizalama hareketi ve yine yakın civardaki sürü üyeleri ile benzer mesafede uçuşu sağlayan bağlılık davranışdır<sup>v</sup>. Davranış şekilleri haricinde önem arz eden diğer bir tanımlaması ise yakın komşuluk kavramıdır. Reynolds yakın komşuluğu, belirli bir yarıçapa sahip daire içersinde kalan kuşlar veya belirli sayıdaki en yakın kuş grubu olarak tanımlamaktadır.

Reynolds’ın gözlemlerine ilave olarak F. Heppner ile U. Grenander ikilisinin eklediği davranış şekli ise rassal bozuntu kavramıdır. Bu sayede benzeşim bazlı çalışmalarda sürü hareketine yerel de olsa rassal davranışlar eklenebilmektedir.

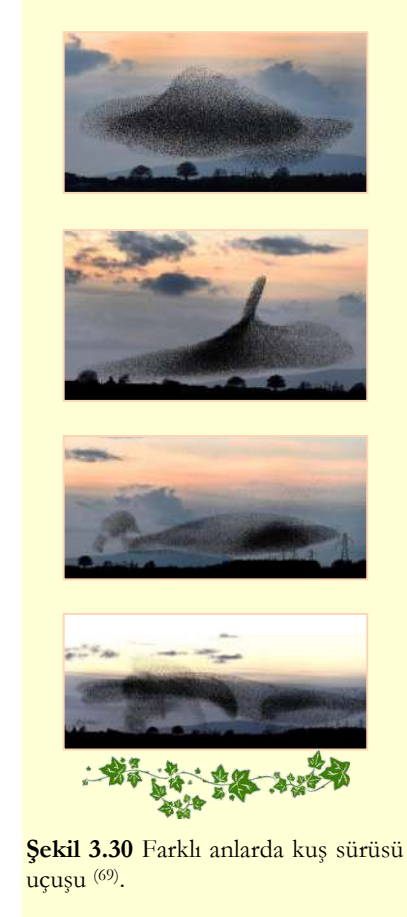
i “Particle swarm optimization”

ii “Sociobiology: the new synthesis”

iii Bird flocking

iv “Boids”; sokak İngilizcesinde kuşlar için kullanılan bir argo kelime.

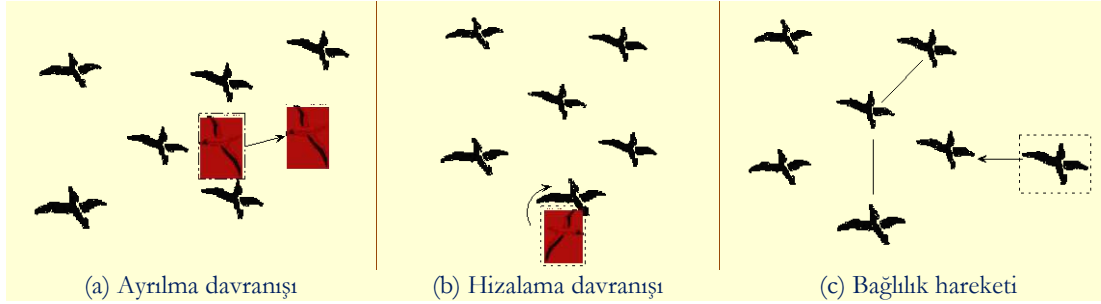
v Separation, alignment, cohesion



Şekil 3.30 Farklı anlarda kuş sürüsü uçuşu <sup>(69)</sup>.







Şekil 3.31 Reynolds'ın gözlemlerine göre sürüde temel kuş hareketleri<sup>(72)</sup>.

Diğer taraftan kuş sürüsü haricinde üzerinde gözlem yapılan diğer bir canlı grubu balıklardır. Balıklar avcılardan korunmak, su içerisinde oluşan sürüklemeyi azaltmak veya daha etkin bir şekilde yiyecek aramak amacıyla sürü davranışları sergilemektedirler. Sürü halinde dolaşan balıklarda genellikle iki tür gruplaşmadan bahsedilebilir. Bunlardan ilki, sığılaşma<sup>i</sup> olarak tabir edilen ve aralarında belirli bir koordine olmayan, ancak bir şekilde grup olarak bulunan sürüye denmektedir. Diğer gruplaşma, belirli bir yöne doğru seyir ve koordinasyon içeren sürü<sup>ii</sup> hareketidir<sup>(73)</sup>. Kuşların sürüsel davranışları her ne kadar PSO yaklaşımına etki etse de algoritmanın temel çıkış kaynağı sosyobiolog Wilson'ın kitabında belirttiği;

- ▶ “En azından kuramsal olarak, yiyecek arama sürecinde balık sürüsündeki bireyler sürüdeki diğer bireylerin keşiflerinden ve geçmişteki tecrübelerinden faydalanabilmektedirler...”

ifadeleridir<sup>(74)</sup>. Buna göre sürü içerisindeki balıkların yiyecek arama sürecinde verdikleri kararlar, kendi kişisel tecrübelerine ve ayrıca sürüdeki diğer balıkların yaptıkları keşiflere bağlı olarak şekillenmekte, bu sayede bireylerde kişisel hafıza şekillenmekte ve sürü içerisinde bireysel tecrübeye dayalı bilgi paylaşımı gerçekleşmektedir. Parçacık sürü optimizasyonu algoritmasında sürü olarak adlandırılan olası çözüm kümesi  $P = \{p_1, p_2, \dots, p_s\}$  belirli sayıda  $p_i$  parçacıklarını içermektedir. Sürüdeki her bir parçacığın konumu, optimizasyon sorununun olası çözümü olarak



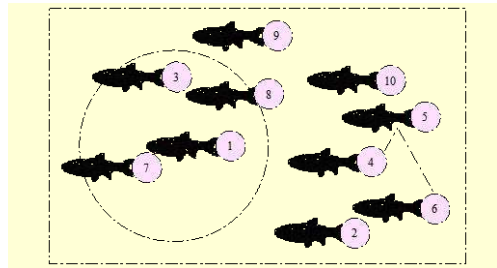
Şekil 3.32 Balıklarda sığılaşma ve sürüleşme davranışları.

<sup>i</sup> Fish shoaling

<sup>ii</sup> Fish schooling

dikkate alınır ve her bir parçacık, optimum çözümü bulma adına daha iyi konumları ferdi olarak tarar. Tarama sürecindeki herhangi bir  $t$  adımında her  $p_i$  parçacığı, belirli bir  $x_i$  konumuna ve  $v_i$  hızına sahiptir. Ayrıca her  $p_i$  parçacığının hafızasında, o adıma kadar elde ettiği en iyi amaç fonksiyon değerini veren konum bilgisi,  $p_{i,b}$  yer alır. Diğer taraftan parçacığın  $N_i \subseteq P$  yakın komşuluğundaki diğer bireylerin sahip olduğu  $p_{j,b}$  verileri de karar sürecinde dikkate alınan diğer bir bilgidir. Dolayısıyla parçacıklar arasında bilgi alışverişi söz konusudur.

Herhangi bir parçacığın sürüdeki hangi parçacıklarla bilgi alış verişi yapacağını belirleyen yakın komşuluğu, genellikle şematik olarak simgelenir;  $G = \{V, E\}$ . Burada  $V$  ifadesi parçacığı,  $E$  ifadesi söz konusu parçacığın ilişkide olduğu, ya da diğer bir ifade ile yakın komşuluğunda olduğu düşünülen diğer parçacıkları temsil eder. Bahse konu şemaya genellikle sürünün topolojik görünümü adı verilir<sup>(75)</sup>. Yakın komşuluk kavramını üç farklı bağlamda değerlendirmek mümkündür. Bunlardan ilki konumsal, ikincisi sosyal ve sonuncusu da genel komşuluk<sup>i</sup> tanımlamasıdır. Konumsal komşuluk tanımlamasında parçacığa belirli bir mesafe ve altındaki uzaklıklarda yer alan parçacıklar dikkate alınır. Dolayısıyla iki boyutlu ortamda merkezde  $p_i$  parçacığı olmak üzere belirli bir yarıçapa sahip daire içersinde kalan  $p_j$  parçacıkları  $p_i$  parçacığının yakın komşuluğunu temsil etmektedir. Sosyal komşuluk kavramı, konumsal bir anlamdan ziyade parçacıklar arasında kurulan ilişkiye göre yakın komşuluğun tanımlanması düşüncesine dayanır. Genel komşuluk kavramı ise yakın komşuluk yerine sürüdeki tüm parçacıkların dikkate alındığı genel nüfusu dikkate alır. Şekil 3.33'ten de görüleceği üzere sürü halindeki parçacıkların konumsal komşuluğu dikkate alındığında 1 numaralı parçacığın yakınında 3,7 ve 8 numaralı bireyler yer almaktadır. Sosyal komşuluk dikkate alındığında, 5 numaralı parçacığın komşuluğunda 4 ve 6 numaralı bireylerin yer aldığı gözlemlenebilir. Genel komşulukta ise her bireyin komşuluğunda diğer tüm bireyler yer alır.

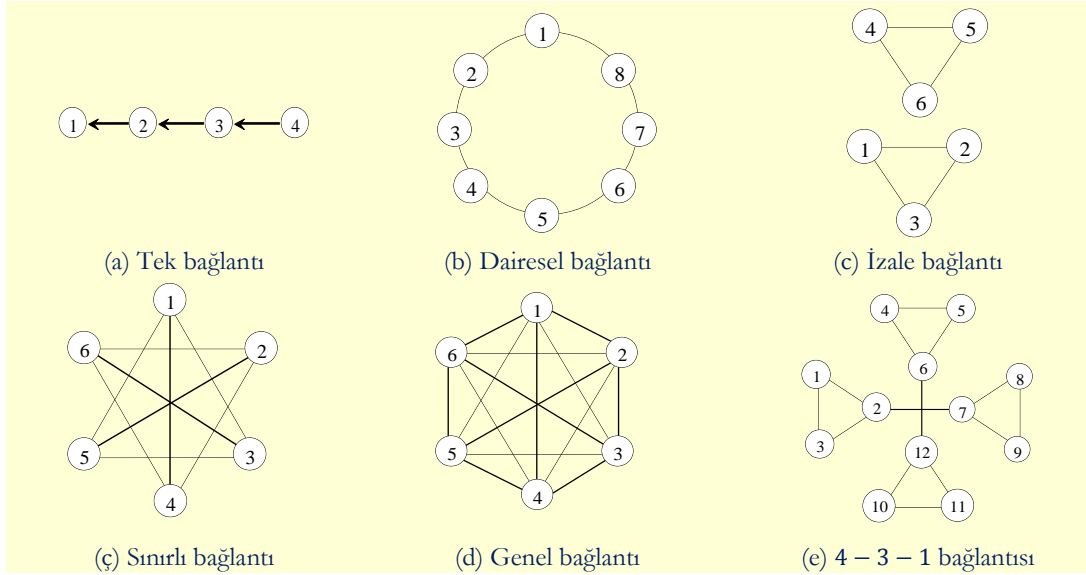


Şekil 3.33 Yakın komşuluk tanımlamaları.

<sup>i</sup> Geographical, social, global



Sosyal komşuluk kavramı, farklı sosyal ilişkilerin tanımlanması ile oldukça zengin bir topolojik çeşitliliğe yayılabilir. Bu meyanda bahsedilebilecek topolojik düzenlemeler tek bağlantı, örnekte de dikkate alınan dairesel bağlantı, izale bağlantı, sınırlı bağlantı,  $n - m - k$  bağlantı ve genel bağlantı düzenlemeleridir. Müteakip şekillerde belirtilen bağlantı topolojileri görülmektedir.



Şekil 3.34 Parçacıklar arasında tanımlanabilen sosyal topoloji örnekleri.

Buna göre tek bağlantı ilişkisinde her parçacık önündeki veya arkasındaki, ancak sadece tek yöndeki diğer parçacıkla ilişkilendirilebilir. Örneğin Şekil 3.34(a) seçeneğinde 2 numaralı parçacık bir önceki 1 numaralı parçacık ile ilişkili görünmektedir. Dairesel bağlantı topolojisinde her parçacık sadece önceki ve sonraki parçacık ile ilişkilendirilebilmektedir. Örneğin aynı şeklin (b) seçeneğinde belirtilen topolojiye göre 5 numaralı parçacık 4 ve 6 ile ilişkilendirilir. İzale bağlantı topolojisinde belirli parçacık grupları tanımlanarak bu parçacıklar arasında ilişkiler belirlenebilir. Örneğin (c) seçeneğinde iki ayrı grup tanımlanmakta olup ilk gruptaki 4 numaralı parçacık sadece 5 ve 6 numaralı parçacıklar ile ilişkilidir. Sınırlı bağlantı ya da literatürdeki diğer adı ile von Neumann topolojisinde belirli parçacıkların diğer belirli parçacıklarla ilişkilendirilmesi mümkündür. Örneğin (ç) seçeneğinde sürüde 6 parçacık mevcut olmasına rağmen her parçacık sadece diğer üç parçacıkla ilişkilendirilmektedir. Mesela 1 numaralı parçacık 3, 4 ve 5 numaralı



parçacıklarla ilişkilendirilmiştir. Genel bağlantı topolojisinde her parçacık sürüdeki diğer tüm parçacıklarla ilişki içersindedir. Diğer topolojilere göre karmaşık sayılabilecek  $n - m - k$  bağlantısında  $n$  alt grup sayısını,  $m$  alt grup dâhilindeki bağlantı sayısını,  $k$  ilişki bağlamında birbirine zıt konumdaki alt gruplar arası bağlantı sayısını tanımlar. Örneğin (e) seçeneğinde  $4 - 3 - 1$  topolojisi tanımlanmış olup sürüde 4 alt grup, her alt grupta üç bağlantı ve zıt alt gruplar arasında da 1 bağlantı tanımlanmaktadır <sup>(76-77)</sup>. Sosyal komşuluk tanımlamasının en önemli getirisi, konumsal olarak belirli bir bölgedeki yoğunlaşmayı engelleme şansı tanınması ve bu sayede yerel optimum noktalara takılma riskini azaltmasıdır. Zira sosyal ilişki bağlamında yan yana görünen parçacıkların konumları, birbirinden tamamen farklı olabilir.

Parçacık sürü algoritmasının temelini, parçacığa ait hızın güncellenmesi işlemi oluşturur. Güncellenen hıza göre parçacığın yeni konumu belirlenir. Basitleştirilmiş bir boyutlu hali ile herhangi bir parçacığın herhangi bir adımdaki hızını ve konumunu;

$$v^{t+1} = av^t + b_1r_1[p_1 - x^t] + b_2r_2[p_2 - x^t] \quad (3.65)$$

$$x^{t+1} = cx^t + dv^{t+1} \quad (3.66)$$

ifadeleri ile tanımlamak mümkündür. Burada bir sonraki hıza ait denklemdeki ilk terim, parçacığın hali hazırdaki hızına ait kısmı, ikinci terim kendi geçmişine ait  $p_1$  tecrübesinin etkisini, son terim ise yakın komşuluğundaki en iyi parçacığa ait  $p_2$  tecrübesinin yansımasını temsil etmektedir. Denklemlerde geçen  $a$ ,  $b_1$ ,  $b_2$ ,  $c$  ve  $d$  sayıları, söz konusu etkilerin ağırlığını ifade eden katsayılardır. İlave olarak dikkate alınan  $r_1$  ve  $r_2$  katsayıları, değerleri  $[0, 1]$  arasında değişebilen rassal sayıları ifade etmektedir. Hız ve konuma ait denklemlerdeki katsayıların gerçek manada etkisinin değerlendirilmesi adına bazı güdüleme işlemleri yapmak mümkündür. Öncelikle sorunun rassal karakteri, rassal katsayılara sabit birer değer atanarak determinist bir forma dönüştürülebilir;

$$r_1 = r_2 = \frac{1}{2} \quad (3.67)$$

Ayrıca  $b_i$  ve  $p_i$  gibi ifadeler de;

$$b = \frac{b_1 + b_2}{2} \quad (3.68)$$

$$p = \frac{b_1}{b_1 + b_2}p_1 + \frac{b_2}{b_1 + b_2}p_2 \quad (3.69)$$

şeklinde güdülenebilir.



Bu tanımlamalar ana denklemlerdeki (Denklem 3.65 ve 3.66) ilk eşitlikte yerine konulduğunda;

$$x^{t+1} = cx^t + d[av^t + b(p - x^t)] \quad (3.70)$$

elde edilebilir. Ana denklemlerdeki ikinci eşitlik kullanılarak;

$$v^t = x^t - cx^{t-1} \quad (3.71)$$

ifadesi elde edilip bir önceki eşitlikte yerine konulursa;

$$x^{t+1} = cx^t + d[a(x^t - cx^{t-1}) + b(p - x^t)] \quad (3.72)$$

ve bazı sadeleştirme ile düzenlemeler yapılırsa;

$$x^{t+1} + (bd - c - a)x^t + acx^{t-1} = bdp \quad (3.73)$$

sonucuna ulaşılabilir. Görüldüğü üzere eşitliğin sağ tarafındaki ifadede  $bd$  çarpımının önemi söz konusu olup bireysel anlamda  $b$  veya  $d$ 'nin anlamı yoktur. Bu nedenle uygulamalarda  $d = 1$  tercihinin dikkate alınması sonucu değiştirmemektedir. Diğer taraftan tekrarlı işlemlerin sonucunda;

$$\lim_{t \rightarrow \infty} x^t = p \quad (3.74)$$

optimal değerine ulaşıldığı düşünüldüğünde bir önceki denklemin;

$$p + (b - c - a)p + acp = bdp \quad (3.75)$$

şeklinde tekrar yazılması mümkündür. Bu ifadede uygun düzenlemelere gidildiğinde;

$$\begin{aligned} p + bp - pc - pa + acp &= bdp \\ p(1 - c - a + ac) &= 0 \\ 1 - c - a(1 - c) &= 0 \\ (1 - c)(1 - a) &= 0 \end{aligned} \quad (3.76)$$

sonucuna ulaşılabilir. Görüldüğü üzere  $c = 1$  veya  $a = 1$  seçimi sonuç üzerinde herhangi bir değişikliğe neden olmayacaktır. Dolayısıyla  $c = 1$  tercihi yapılabilir. Hülasa, yapılan basitleştirme ve uygun güdüleme işlemleri neticesinde güncelleme denklemlerindeki konum ifadesini;



$$x^{t+1} = x^t + v^{t+1} \quad (3.77)$$

şeklinde kullanmak mümkündür<sup>(78)</sup>. Belirtilen güncelleme denklemleri dâhilindeki farklı kısım ve katsayıların mevcudiyeti nedeniyle değişik PSO seçenekleri düşünülebilir. Literatürdeki en basit PSO algoritmasına ait güncelleme denklemleri;

$$v_{i,j}^{t+1} = v_{i,j}^t + b_1 r_1 [p_{b,i,j}^t - x_{i,j}^t] + b_2 r_2 [p_{g,j}^t - x_{i,j}^t] \quad (3.78)$$

$$x_{i,j}^{t+1} = x_{i,j}^t + v_{i,j}^{t+1} \quad (3.79)$$

şeklinde. Burada  $p_{b,i,j}^t$  ifadesi  $t$  adımında  $i$ 'nci parçacığa ait bireysel en iyi konum yönünün  $j$ 'nci boyuttaki değerini,  $p_{g,j}^t$  yine  $t$  adımında  $i$ 'nci bireyin yakın komşuluğundaki parçacıklara ait bireysel en iyi konum yönünün  $j$ 'nci boyuttaki değerini temsil etmektedir.

Algoritmanın tarama sürecinde olası çözüm uzayından uzaklaşmaması adına, bazı uygulamalarda hıza sınırlama getirilebilmektedir. Sınırlama sadece üst kısıt niteliğinde olmayıp durağanlığı önleyici nitelikte alt kısıt uygulaması da yapılabilir. Buna meyanda hız ifadesi;

$$v_{i,j}^{t+1} = \begin{cases} v_{i,j}^{t+1} & v_{i,j}^{t+1} < v_{maks,j} \\ v_{maks,j} & v_{i,j}^{t+1} > v_{maks,j} \end{cases} \quad (3.80)$$

$$v_{i,j}^{t+1} = \begin{cases} v_{i,j}^{t+1} & v_{i,j}^{t+1} > v_{min,j} \\ v_{min,j} & v_{i,j}^{t+1} < v_{min,j} \end{cases} \quad (3.81)$$

ifadelerine göre belirli bir aralıkta tutulabilir. Burada  $v_{min,j}$  ve  $v_{maks,j}$  ifadeleri,  $o$  boyuttaki hız alt ve üst sınır değerlerini göstermektedir. Hız sınırlaması yerine dikkate alınabilecek diğer bir yaklaşım, eylemsizlik katsayısı uygulamasıdır. Bahse konu katsayı, literatürde eylemsizlik ağırlık<sup>ii</sup> katsayısı olarak ifade edilmekte olup hız güncellemesi;

$$v_{i,j}^{t+1} = w v_{i,j}^t + b_1 r_1 [p_{b,i,j}^t - x_{i,j}^t] + b_2 r_2 [p_{g,j}^t - x_{i,j}^t] \quad (3.82)$$

eşitliğine göre yapılır.

<sup>i</sup> Velocity clamping

<sup>ii</sup> Inertia weight



Burada  $w$  simgesi eylemsizlik katsayısını göstermektedir. Söz konusu katsayı  $0 < w < 1$  aralığında sabit bir değer alınabileceği gibi tekrar sayısına bağlı olarak değiştirilebilir de. Değişimde;

$$w^t = w_{maks} - \frac{w_{maks} - w_{min}}{T} t \quad (3.83)$$

ifadesi dikkate alınabilir. Burada eylemsizlik katsayısı başlangıçta  $w_{maks}$  değerinden başlatılmakta ve  $T$  azami iterasyon sayısına ulaşıldığında  $w_{min}$  değerine indirgenmektedir. Hız sınırlamasına ihtiyaç duyulmadan yapılacak tarama yöntemlerinden bir diğeri, sınırlama etmeninin<sup>1</sup> dikkate alınmasıdır. Buna göre hız güncellemesi;

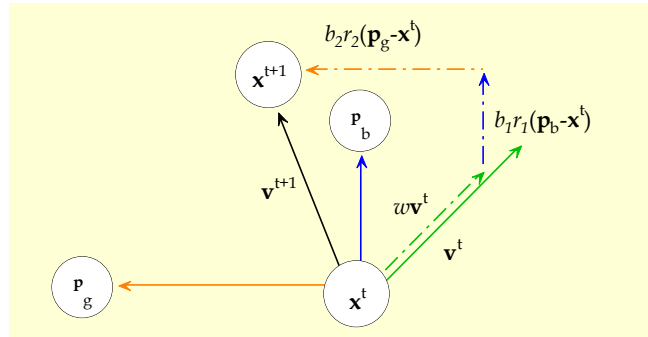
$$v_{i,j}^{t+1} = \kappa \{v_{i,j}^t + b_1 r_1 [p_{b,i,j}^t - x_{i,j}^t] + b_2 r_2 [p_{g,j}^t - x_{i,j}^t]\} \quad (3.84)$$

eşitliğine göre yapılır. Burada  $\kappa$  simgesi sınırlama etmeni olup;

$$\kappa = \frac{2}{|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|} \quad (3.85)$$

$\varphi = b_1 + b_2$   
 $\varphi \geq 4$

kıstaslarına göre belirlenir. Yaklaşımına göre, uygun seçilen sınırlama etmeni neticesinde algoritmanın optimum bir konuma yakınsaması kesindir<sup>(79)</sup>.



Şekil 3.35 PSO yaklaşımında parçacığın bir sonraki konumunun yöneysel olarak gösterimi.

<sup>1</sup> Constriction factor



Hız ve konum güncelleme işlemi, yöneysel bağlamda Şekil 3.35'te görülmektedir. Buna göre bir sonraki konum, parçacığın cari hızı ile bireysel ve sürüsel yönlendirmelerin bir eseridir. Cari hızın etkisi, kullanıcı tarafından belirlenmekle beraber bireysel ve sürüsel yönlendirmeler belli oranda rastsal karaktere haiz olup etkileri sınırlı kalmaktadır. Bununla beraber ivmelenme katsayıları olarak da adlandırılan  $b_1$  ve  $b_2$  katsayılarının uygun büyüklükte seçilmesi halinde o yöndeki konumun etkisi kendisinden büyük olabilir. Denklemlerden de görüleceği üzere  $b_1$  katsayısı, bireyin bilişsel<sup>i</sup> hafızasının bir sonraki adıma yansımaları kontrol etmektedir. Bahse konu katsayının değer olarak büyük alınması, geçmiş dönemdeki tecrübenin etkisinin artırılması anlamına gelir. Öte yandan  $b_2$  katsayısı, parçacığın yakın komşuluğundaki sürüsel tecrübeye atfen belirlenen bir katsayıdır. Söz konusu katsayının değer olarak yüksek mertebede seçilmesi, bireyin kendi tecrübesinden ziyade sürü tecrübesine daha fazla değer verdiği anlamına gelir. Sürü içerisindeki  $i$ 'nci bireyin  $t$ 'nci adıma kadar kişisel geçmişinde elde ettiği en iyi amaç fonksiyon değerini veren konum niteliğindeki  $p_{b,i}^t$  yöneyi;

$$p_{i,b}^t = \begin{cases} p_{i,b}^{t-1} & f(x_i^t) > p_{b,i}^{t-1} \\ x_i^t & f(x_i^t) \leq p_{b,i}^{t-1} \end{cases} \quad (3.86)$$

tanımlamasına göre belirlenir. Diğer taraftan parçacığın yakın komşuluğu dâhilindeki en iyi bilişsel konum yöneyi;

$$p_g^t = \min\{p_{i,b}^t\}, i \in N_i \quad (3.87)$$

ifadesinden de anlaşılacağı üzere yakın komşuluğundaki bireylerin, içinde bulunulan adım dahil geçmişlerindeki en iyi konumunu göstermektedir.

Parçacık sürü algoritması, temelde basit bir yaklaşım olup genellikle gerçek sayıların söz konusu olduğu sorunlarda dikkate alınan bir yöntemdir. Bununla beraber ayrık değişkenlere sahip sorunların çözümünde de kullanılabilir. Performans açısından diğer sezgisel yöntemlerle karşılaştırıldığında PSO yaklaşımının daha basit ve uygulanması daha kolay bir yaklaşım olduğu söylenebilir. Ancak genel bir ilke olarak, yöntemlerin sorun çözmedeki verimliliklerinin soruna bağlı olduğu unutulmamalıdır.

PSO algoritmasının işlem adımlarını müteakip maddelerdeki gibi sıralamak mümkündür;

---

<sup>i</sup> Cognitive part





- 1. Yöntemin başlangıcında olası çözüm kümesi niteliğindeki başlangıç sürüsü ve hızları teşkil edilir. Genellikle başlangıç sürüsü bireyleri olası çözüm uzayı sınırları içerisinde rassal olarak üretilir;

$$x_{i,j}^{(0)} = x_j^l + (x_j^u - x_j^l)rand(0,1), \quad \begin{matrix} i=1,2,\dots,n_s \\ j=1,2,\dots,n \end{matrix}$$

Burada  $[x_j^l, x_j^u]$  ifadeleri  $j$ 'nci boyuta ait alt ve üst sınır değerlerini,  $n$  değişken sayısını ve  $n_s$  sürü büyüklüğünü göstermektedir. Benzer yaklaşım, başlangıç sürüsüne ait hız değerlerinin belirlenmesi için de sergilenebilir;

$$v_{i,j}^{(0)} = [x_j^l + (x_j^u - x_j^l)rand(0,1)]/\Delta t, \quad \begin{matrix} i=1,2,\dots,n_s \\ j=1,2,\dots,n \end{matrix}$$

Burada  $\Delta t$  değeri, konuma göre hız değerlerinde belirli bir ölçeklendirme imkânı sunan ve kullanıcı tarafından belirlenen bir değerdir. Ayrıca yönteme ait azami iterasyon sayısı  $T$ , ivmelenme katsayıları  $b_1$  ve  $b_2$  ile eylemsizlik katsayısı  $w$  belirlenir.

- 2. Sürünün belirlenmesi akabinde sürüdeki parçacıklara ait amaç fonksiyon değerleri elde edilir,  $f(x_i^t)$ ,  $t = 0$ . Belirlenen değerler çerçevesinde yakınsama kontrolü yapılabilir. Yakınsama kontrolü meyanda değişik ölçütlerden bahsedilebilir. Genellikle iterasyon sayısının azami iterasyon sayısına ulaşip ulaşmadığı dikkate alınabilir. Bununla beraber yönteme has bazı yakınsama ölçütleri tanımlamak da mümkündür. Bu tanımlamalardan ilki, sürüye ait ortalama hız değerinin belirli bir sınır değerinin altına düşmesi halidir;

$$\bar{v} \leq \epsilon_1 \quad (3.88)$$

$$\bar{v}^t = \frac{1}{n_s} \sum_{i=1}^{n_s} v_i^t, \quad i = 1, 2, \dots, n_s \quad (3.89)$$

Diğer bir tanımlama, sürünün erişebildiği en iyi konuma göre sürü üyelerinin dağılımını dikkate alır. Bu meyanda sürüye ait normalize edilmiş bir sürü yarıçapı belirlenebilir. Belirlenen söz konusu sürü yarıçapının değerinin belirli bir sınır değerden düşük olması halinde yakınsama gerçekleşmiş kabul edilebilir;

$$R_{norm} = \frac{R_{maks}}{R_{maks,0}} \leq \epsilon_2 \quad (3.90)$$

$$R_{maks} = maks\{\|x_i^t - p_g^t\|\}, \quad i = 1, 2, \dots, n_s \quad (3.91)$$





Burada  $R_{norm}$  normalize edilmiş sürü yarıçapını,  $R_{maks}$  sürüye ait  $\mathbf{p}_g$  nirengi noktası olmak üzere sürüdeki en uzak parçacığı gösteren azami yarıçapı ifade etmektedir.  $R_{maks,0}$  aynı kavramın başlangıç nüfusundaki değerini temsil etmektedir. Gerek sürüye ait ortalama hız ve gerekse de sürü yarıçapı kavramları esasen sürünün konumsal olarak belirli bir bölgeye yığılıp yığılmadığını gösteren imgelerdir.

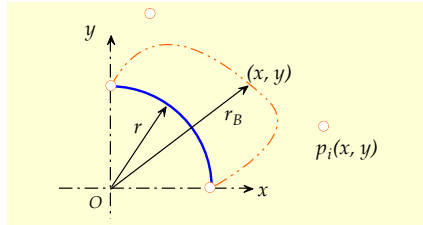
- ▶ 3. Yakınsama kontrolü neticesinde arzu edilen koşulun sağlanması halinde tarama süreci sonlandırılır. Aksi halde tarama sürecine devam edilir. Bu meyanda önce  $\mathbf{p}_b$  ve  $\mathbf{p}_g$  yöneyleri güncellenir.
- ▶ 4. Akabinde parçacıkların hız ve konumları güncellenir. Güncellemede genellikle daha önce de ifade edilen;

$$v_{i,j}^{t+1} = wv_{i,j}^t + b_1r_1 [p_{b,i,j}^t - x_{i,j}^t] + b_2r_2 [p_{g,j}^t - x_{i,j}^t]_{j=1,2,\dots,n}^{i=1,2,\dots,n_s}$$

$$x_{i,j}^{t+1} = x_{i,j}^t + v_{i,j}^{t+1}$$

denklemleri dikkate alınır. Yeni konumların belirlenmesini müteakiben tekrarlı işlem adımları dâhilinde 2nci adımla aynı işlemlere devam edilir.

Parçacık sürü optimizasyon algoritmasının daha iyi anlaşılması için örnek bir eğri uydurma sorununu belirtilen işlem adımlarını takip ederek çözmek mümkündür.



Şekil 3.36 Çembere ait yay parçası ve örnek bir Bézier eğrisi.

Buna göre bir çemberin 90°'lik yay parçasını, Bézier kontrol noktalarını kullanarak elde etmek isteyelim. Böyle bir problemde çembere ait yay parçası noktaları belirli olup aynı yay eğrisini veren Bézier kontrol noktaları, belirlenmesi gereken tasarım değişkenleridir. Bézier eğrileri, belirli



sayıdaki kontrol noktaları ile şekil verilebilen eğriler olup;  $p_i$  kontrol noktası olmak üzere eğriye ait koordinat verisi;

$$P(t) = \sum_{i=0}^m \binom{m}{i} (1-t)^{m-i} t^i p_i \quad (3.92)$$

ifadesine göre belirlenir. Burada  $m$  kontrol nokta sayısını ve  $t$  parametrik değeri göstermektedir,  $0 \leq t \leq 1$ . Şekil 3.36'da çember yayı ve örnek kübik Bézier eğrisi görülmektedir. Toplamda dört kontrol noktası öngörülmeyle beraber iki noktanın sabit olması nedeniyle diğer iki noktanın belirlenmesi yeterlidir. Çemberin yarıçapının  $r = 2br$  olarak dikkate alınması halinde ilk ve son nokta koordinatları sırasıyla  $(2,0)$ ,  $(0,2)$  şeklinde ifade edilebilir. İlk nokta  $p_0(x_0, y_0)$  ve birinci ve ikinci noktalar da  $p_1(x_1, y_1)$  ve  $p_2(x_2, y_2)$  olmak üzere tasarım yöneyi;

$$z = \begin{bmatrix} x_1 \\ x_2 \\ y_1 \\ y_2 \end{bmatrix}$$

şeklinde yazılabilir. Dört noktanın söz konusu olduğu kübik Bézier eğrisi açık halde;

$$P(t) = (1-t)^3 p_0 + 3(1-t)^2 t p_1 + 3(1-t) t^2 p_2 + t^3 p_3 \quad (3.93)$$

denklemleri ile ifade edilebilir. Bu ifade, her bir koordinat için düzenlendiğinde;

$$x(t) = 2(1-t)^3 + 3(1-t)^2 t x_1 + 3(1-t) t^2 x_2 \quad (3.94)$$

$$y(t) = 3(1-t)^2 t y_1 + 3(1-t) t^2 y_2 + 2t^3 \quad (3.95)$$

$$0 \leq t \leq 1$$

eşitlikleri elde edilir. Koordinatları belirlenen Bézier eğrisine ait noktanın yöneysel şiddeti;

$$r_B = \sqrt{x^2 + y^2} \quad (3.96)$$

denklemleri ile tanımlanabilir. Bu tanım kullanılarak Bézier eğrisinin daire yayı ile farkı elde edilebilir ve eniyileme sürecinin gayesi olarak söz konusu farkın minimizasyonu düşünülebilir;

$$f = \sum_{j=1}^k (2 - r_{B,j})^2 \quad (3.97)$$

burada  $k$  ifadesi eğrinin nokta sayısını göstermektedir.



Dairesel yay parçasını temsil edebilecek uygun kontrol noktalarının PSO yöntemi ile bulunması sürecinde yapılacak ilk işlem, yaklaşıma ait parametrelerin ve akabinde de başlangıç nüfusunun belirlenmesidir. Bu meyanda azami iterasyon sayısı  $T = 100$ , sürü büyüklüğü  $n_s = 10$ , ivmelenme katsayıları  $b_1 = 1.5$  ve  $b_2 = 1.5$  ve eylemsizlik katsayısı  $w = 0.5$  olarak belirlenebilir. Başlangıç sürüsü bireyleri, olası çözüm uzayı sınırları içerisinde rastsal olarak üretilir. Üretim dahilinde,  $[0, 4]$  aralığı ve rastsal sayılar dikkate alınarak aşağıdaki başlangıç sürüsünü elde etmek mümkündür;

$$x_{i,j}^{(0)} = 4 \cdot \text{rand}(0,1)_{\substack{i=1,2,\dots,10 \\ j=1,2,\dots,4}}$$

$$\begin{aligned} x_1^{(0)} &= [1.6561 \quad 2.6582 \quad 3.6456 \quad 2.5029], & x_2^{(0)} &= [1.8415 \quad 3.9134 \quad 0.7205 \quad 3.6495] \\ x_3^{(0)} &= [1.2458 \quad 0.0722 \quad 1.1240 \quad 3.4940], & x_4^{(0)} &= [0.5889 \quad 1.9577 \quad 0.6364 \quad 2.7528] \\ x_5^{(0)} &= [1.0959 \quad 0.0913 \quad 0.3590 \quad 3.5673], & x_6^{(0)} &= [0.3670 \quad 3.9052 \quad 3.8703 \quad 1.4068] \\ x_7^{(0)} &= [0.9390 \quad 1.6511 \quad 3.5208 \quad 0.9969], & x_8^{(0)} &= [1.3019 \quad 2.6986 \quad 1.7612 \quad 0.0785] \\ x_9^{(0)} &= [2.6862 \quad 0.2706 \quad 2.0155 \quad 3.6763], & x_{10}^{(0)} &= [3.0008 \quad 0.4159 \quad 0.3505 \quad 1.6153] \end{aligned}$$

Benzer şekilde sürüye ait hız değerleri de;

$$v_{i,j}^{(0)} = 4 \cdot \text{rand}(0,1) / \Delta t_{\substack{i=1,2,\dots,10 \\ j=1,2,\dots,4}}$$

$$\Delta t = 4$$

$$\begin{aligned} v_1^{(0)} &= [0.5626 \quad 0.7020 \quad 0.2403 \quad 0.8258], & v_2^{(0)} &= [0.0774 \quad 0.4367 \quad 0.8691 \quad 0.6831] \\ v_3^{(0)} &= [0.0349 \quad 0.6098 \quad 0.2528 \quad 0.4429], & v_4^{(0)} &= [0.4459 \quad 0.5798 \quad 0.0977 \quad 0.5146] \\ v_5^{(0)} &= [0.1053 \quad 0.5774 \quad 0.7016 \quad 0.5846], & v_6^{(0)} &= [0.7312 \quad 0.3380 \quad 0.7869 \quad 0.0356] \\ v_7^{(0)} &= [0.5087 \quad 0.9878 \quad 0.2208 \quad 0.1042], & v_8^{(0)} &= [0.8829 \quad 0.8630 \quad 0.9622 \quad 0.7619] \\ v_9^{(0)} &= [0.0792 \quad 0.0337 \quad 0.8478 \quad 0.8854], & v_{10}^{(0)} &= [0.9371 \quad 0.6077 \quad 0.6989 \quad 0.0204] \end{aligned}$$

biçiminde belirlenebilir. Sürünün belirlenmesi akabinde sürüdeki parçacıklara ait amaç fonksiyon değerleri, ifade edilen minimizasyon hedefi çerçevesinde elde edilebilir;

$$f = \sum_{j=1}^{66} (2 - r_{B,j})^2$$

$$f^{(0)} = [39.868 \quad 41.557 \quad 5.737 \quad 5.259 \quad 7.582 \quad 22.313 \quad 2.429 \quad 1.657 \quad 21.178 \quad 2.826]$$

Eğri üzerinde 66 nokta dikkate alınarak amaç fonksiyon değeri belirlenmiştir. Bu aşamada yakınsama ölçütü olarak azami tekrar sayısı temel alınabilir. Henüz başlangıç aşamasında bulunduğu için bir sonraki işleme geçilebilir. Bu meyanda önce  $p_b$  ve  $p_g$  yöneyleri güncellenir.



Bireysel en iyi konumlar, daha başlangıç aşamasında bulunduğu için hali hazırdaki mevcut konumlar ile aynı olup;

$$\begin{aligned}
 \mathbf{p}_{b,1}^{(0)} &= [1.6561 \ 2.6582 \ 3.6456 \ 2.5029], & \mathbf{p}_{b,2}^{(0)} &= [1.8415 \ 3.9134 \ 0.7205 \ 3.6495] \\
 \mathbf{p}_{b,3}^{(0)} &= [1.2458 \ 0.0722 \ 1.1240 \ 3.4940], & \mathbf{p}_{b,4}^{(0)} &= [0.5889 \ 1.9577 \ 0.6364 \ 2.7528] \\
 \mathbf{p}_{b,5}^{(0)} &= [1.0959 \ 0.0913 \ 0.3590 \ 3.5673], & \mathbf{p}_{b,6}^{(0)} &= [0.3670 \ 3.9052 \ 3.8703 \ 1.4068] \\
 \mathbf{p}_{b,7}^{(0)} &= [0.9390 \ 1.6511 \ 3.5208 \ 0.9969], & \mathbf{p}_{b,8}^{(0)} &= [1.3019 \ 2.6986 \ 1.7612 \ 0.0785] \\
 \mathbf{p}_{b,9}^{(0)} &= [2.6862 \ 0.2706 \ 2.0155 \ 3.6763], & \mathbf{p}_{b,10}^{(0)} &= [3.0008 \ 0.4159 \ 0.3505 \ 1.6153]
 \end{aligned}$$

şeklinde yazılabilir. Yakın komşuluk topolojisi olarak genellikle genel bağlantı topolojisi tercih edilebilir. Dolayısıyla mevcut bireysel en iyi konumlar arasındaki 8 numaralı bireye ait konum, sürüye ait en iyi konum ve amaç fonksiyonu değeri olarak kayda geçirilebilir;

$$\begin{aligned}
 \mathbf{p}_g^{(0)} &= \min \{f\{\mathbf{p}_{i,b}^{(0)}\}\}, \quad i = 1, 2, \dots, 10 \\
 \mathbf{p}_g^{(0)} &= [1.3019 \ 2.6986 \ 1.7612 \ 0.0785] \\
 f_b^{(0)} &= 1.6578
 \end{aligned}$$

Daha sonra parçacıkların hız ve konumları rassal sayılara bağlı olarak güncellenir;

$$\begin{aligned}
 v_{i,j}^{(1)} &= 0.5v_{i,j}^{(0)} + 1.5r_1 [p_{b,i,j}^{(0)} - x_{i,j}^{(0)}] + 1.5r_2 [p_{g,j}^{(0)} - x_{i,j}^{(0)}]_{j=1,2,\dots,4}^{i=1,2,\dots,10} \\
 x_{i,j}^{(1)} &= x_{i,j}^{(0)} + v_{i,j}^{(1)} \\
 r_1, r_2 &\in \text{rand}(0,1)
 \end{aligned}$$

$$\begin{aligned}
 \mathbf{v}_1^{(1)} &= [ \ 0.2133 \ 0.3574 \ -0.5496 \ -0.0001 \ ] \\
 \mathbf{v}_2^{(1)} &= [ -0.0984 \ 0.0254 \ 0.7377 \ 0.0313 \ ] \\
 \mathbf{v}_3^{(1)} &= [ \ 0.0190 \ 0.3208 \ 0.1542 \ -0.1613 \ ] \\
 \mathbf{v}_4^{(1)} &= [ \ 0.2965 \ 0.4828 \ 0.4676 \ 0.2291 \ ] \\
 \mathbf{v}_5^{(1)} &= [ \ 0.0773 \ 0.6354 \ 0.6646 \ -0.7513 \ ] \\
 \mathbf{v}_6^{(1)} &= [ \ 0.6482 \ 0.1582 \ 0.0254 \ -0.1909 \ ] \\
 \mathbf{v}_7^{(1)} &= [ \ 0.3438 \ 0.8050 \ 0.0647 \ -0.1200 \ ] \\
 \mathbf{v}_8^{(1)} &= [ \ 0.4414 \ 0.4315 \ 0.4811 \ 0.3810 \ ] \\
 \mathbf{v}_9^{(1)} &= [ -0.2154 \ 0.6930 \ 0.3916 \ -0.8403 \ ] \\
 \mathbf{v}_{10}^{(1)} &= [ \ 0.2647 \ 0.6879 \ 0.6314 \ -0.2132 \ ]
 \end{aligned}$$



$$\begin{aligned} \mathbf{x}_1^{(1)} &= [ 1.8693 & 3.0156 & 3.0960 & 2.5028 ] \\ \mathbf{x}_2^{(1)} &= [ 1.7431 & 3.9388 & 1.4582 & 3.6808 ] \\ \mathbf{x}_3^{(1)} &= [ 1.2649 & 0.3930 & 1.2782 & 3.3327 ] \\ \mathbf{x}_4^{(1)} &= [ 0.8854 & 2.4405 & 1.1040 & 2.9819 ] \\ \mathbf{x}_5^{(1)} &= [ 1.1732 & 0.7267 & 1.0236 & 2.8160 ] \\ \mathbf{x}_6^{(1)} &= [ 1.0152 & 4.0633 & 3.8957 & 1.2159 ] \\ \mathbf{x}_7^{(1)} &= [ 1.2828 & 2.4561 & 3.5855 & 0.8768 ] \\ \mathbf{x}_8^{(1)} &= [ 1.7433 & 3.1301 & 2.2423 & 0.4595 ] \\ \mathbf{x}_9^{(1)} &= [ 2.4708 & 0.9636 & 2.4071 & 2.8360 ] \\ \mathbf{x}_{10}^{(1)} &= [ 3.2654 & 1.1038 & 0.9818 & 1.4021 ] \end{aligned}$$

Yeni konumların belirlenmesini müteakiben sürüye ait amaç fonksiyon değerleri elde edilir;

$$\mathbf{f}^{(1)} = [38.521 \ 52.436 \ 4.896 \ 7.417 \ 2.726 \ 31.223 \ 7.633 \ 4.686 \ 15.845 \ 4.746]$$

Yakınsama kontrolü dâhilinde iterasyon sayısının 1 olması nedeniyle bir sonraki işleme geçilebilir. Bu meyanda  $\mathbf{p}_b$  ve  $\mathbf{p}_g$  yöneyleri güncellenir. Güncellemede bireysel en iyi yöneylere ait amaç fonksiyon değerleri ile güncel sürüye ait amaç fonksiyon değerleri karşılaştırması dikkate alınır;

$f(\mathbf{p}_{b,i}^{(0)})$		$f^{(1)}$	$\mathbf{p}_{b,i}^{(1)}, i = 1, 2, \dots, 10$			
39.8685	>	38.5214	1.8693	3.0156	3.0960	2.5028
41.5573	<	52.4367	1.8415	3.9134	0.7205	3.6495
5.7370	>	4.8968	1.2649	0.3930	1.2782	3.3327
5.2590	<	7.4175	0.5889	1.9577	0.6364	2.7528
7.5821	>	2.7264	1.1732	0.7267	1.0236	2.8160
22.3132	<	31.2230	0.3670	3.9052	3.8703	1.4068
2.4293	<	7.6339	0.9390	1.6511	3.5208	0.9969
1.6578	<	4.6867	1.3019	2.6986	1.7612	0.0785
21.1780	>	15.8453	2.4708	0.9636	2.4071	2.8360
2.8269	<	4.7460	3.0008	0.4159	0.3505	1.6153

$$\mathbf{p}_g^{(1)} = \min \left\{ f \left\{ \mathbf{p}_{i,b}^{(1)} \right\} \right\}$$

$$i = 1, 2, \dots, 10$$

$$\mathbf{p}_g^{(1)} = [1.3019 \ 2.6986 \ 1.7612 \ 0.0785], f_b^{(0)} = 1.6578$$

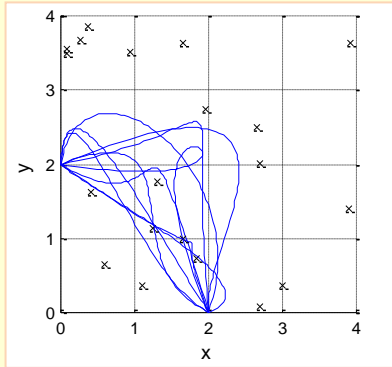




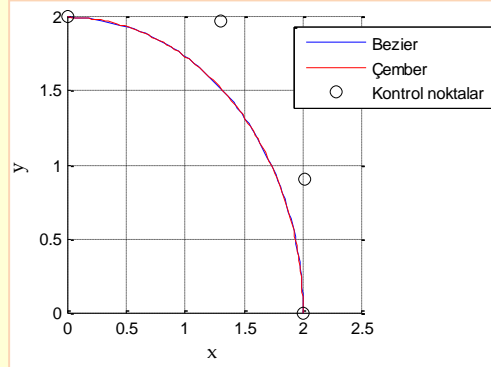
Daha sonra parçacıkların hız ve konumları,  $\mathbf{p}_b$  ve  $\mathbf{p}_g$  ile rassal sayılar kullanılarak güncellenir;

$$\begin{array}{l}
 v_1^{(1)} = [ \quad 0.0565 \quad 0.0834 \quad -0.6037 \quad -0.0760 \quad ] \\
 v_2^{(1)} = [ \quad -0.1718 \quad -0.0767 \quad 0.4528 \quad -0.3833 \quad ] \\
 v_3^{(1)} = [ \quad 0.0167 \quad 0.9174 \quad 0.1183 \quad -0.3314 \quad ] \\
 v_4^{(1)} = [ \quad 0.2183 \quad 0.2671 \quad 0.2373 \quad -0.8964 \quad ] \\
 v_5^{(1)} = [ \quad 0.0536 \quad 0.4908 \quad 0.4453 \quad -1.0306 \quad ] \\
 v_6^{(1)} = [ \quad 0.2495 \quad -0.4043 \quad -0.3606 \quad -0.0517 \quad ] \\
 v_7^{(1)} = [ \quad 0.0607 \quad 0.3147 \quad -0.0909 \quad -0.2243 \quad ] \\
 v_8^{(1)} = [ \quad 0.0175 \quad 0.0547 \quad 0.0436 \quad 0.0892 \quad ] \\
 v_9^{(1)} = [ \quad -0.1283 \quad 0.8188 \quad 0.0130 \quad -1.3535 \quad ] \\
 v_{10}^{(1)} = [ \quad -0.4695 \quad 0.6964 \quad 0.4735 \quad -0.1965 \quad ] \\
 x_1^{(1)} = [ \quad 2.0955 \quad 3.3493 \quad 0.6813 \quad 2.1990 \quad ] \\
 x_2^{(1)} = [ \quad 1.0560 \quad 3.6318 \quad 3.2694 \quad 2.1475 \quad ] \\
 x_3^{(1)} = [ \quad 1.3316 \quad 4.0625 \quad 1.7513 \quad 2.0071 \quad ] \\
 x_4^{(1)} = [ \quad 1.7588 \quad 3.5087 \quad 2.0533 \quad -0.6038 \quad ] \\
 x_5^{(1)} = [ \quad 1.3877 \quad 2.6899 \quad 2.8048 \quad -1.3064 \quad ] \\
 x_6^{(1)} = [ \quad 2.0134 \quad 2.4462 \quad 2.4532 \quad 1.0091 \quad ] \\
 x_7^{(1)} = [ \quad 1.5257 \quad 3.7148 \quad 3.2217 \quad -0.0205 \quad ] \\
 x_8^{(1)} = [ \quad 1.8134 \quad 3.3488 \quad 2.4165 \quad 0.8164 \quad ] \\
 x_9^{(1)} = [ \quad 1.9577 \quad 4.2388 \quad 2.4592 \quad -2.5780 \quad ] \\
 x_{10}^{(1)} = [ \quad 1.3875 \quad 3.8896 \quad 2.8758 \quad 0.6159 \quad ]
 \end{array}$$

Yeni konumların belirlenmesi sonrasında daha önceki işlem adımları aynen tekrar edilerek azami tekrar sayısına kadar benzer işlemlere devam edilir. Müteakip şekillerde başlangıç sürüsü ve nihayetinde ulaşılan en uygun çözüm görülmektedir;



(a) Başlangıç sürüsü



(b) Optimal Bézier kontrol noktaları, eğrisi ve çember yayı

**Şekil 3.37** (a) Başlangıç sürüsü ve kontrol noktaları, (b) en uygun Bézier kontrol noktaları, eğrisi ve çember yayı.



Buna göre yapılan tarama neticesinde en iyi amaç fonksiyon değeri  $f_b^{(100)} = 8.5167e - 05$  ve tasarım yöneyi  $x^* = [2.0141 \ 1.2968 \ 0.9081 \ 1.9774]$  olarak elde edilebilir.

Çember yayı için en uygun Bézier kontrol noktalarının tespiti.

**MATLAB uygulaması**




```
clear; clc
s = 10 ; % sürü büyüklüğü
bird_setp = 100 ; % azami tekrar sayısı
n = 4 ; % sorunun boyutu
c1 = 1.5 ; % pso parametresi, parçacık hafızasının etkisi
c2 = 1.5 ; % pso parametresi, sürü etkisi
w = .5 ; % pso parametresi, eylemsizlik etkisi
dt = 4 ; % hız için süre
border_1 = 0 ; % alt sınır
border_2 = 4 ; % üst sınır
fitness = 0*ones(s,bird_setp) ;
rand('state',sum(100*clock))
R1 = rand(s,n) ;
R2 = rand(s,n) ;
% başlangıç sürüsü
current_position = border_1 + rand(s,n) * (border_2-border_1) ;
velocity = (border_1 + rand(s,n) * (border_2-border_1))/dt ;
local_best_position = current_position;
% Bézier matrisi
for i=1:1:n
    for j=i:1:n
        m(i,j)=(-1)^(j-i)*factorial(n-1)/factorial(j-1)/factorial(n-j)*factorial(j-1)/factorial(j-i)/factorial(i-1);
    end
end
% amaç fonksiyonu değerlendirmeleri
for i = 1:s
    z = current_position(i,:);
    Z_x = [2 z(1) z(2) 0] ;
    Z_y = [0 z(3) z(4) 2] ;
    t=0;
    for tt=0:1/65:1
        t=t+1;
        for j=1:1:n
            T(j)=tt^(j-1);
        end
        mt=m*T';
        x(t) = Z_x*mt;
        y(t) = Z_y*mt;
    end
    r_t = sqrt(y.^2+x.^2);
```





```

    current_fitness(i) = sum((2-r_t).^2) ;
    plot(x,y); hold on
    plot(Z_x(2),Z_y(2),'kx')
    plot(Z_x(3),Z_y(3),'kx')
end
local_best_fitness      = current_fitness ;
[global_best_fitness,g] = min(local_best_fitness)      ;
for i = 1:s
    globl_best_position(i,:) = local_best_position(g,:) ;
end
% hız güncellemesi
velocity = w *velocity + c1*(R1.*(local_best_position-current_position))/dt +
c2*(R2.*(globl_best_position-current_position)/dt) ;
% konum güncellemesi
current_position = current_position + velocity ;
%% iterasyonlar
iter = 0 ;
while ( iter < bird_setp )
iter = iter + 1;
for i = 1:s
    z = current_position(i,:);
    Z_x = [2 z(1) z(2) 0] ;
    Z_y = [0 z(3) z(4) 2] ;
    t=0;
    for tt=0:1/65:1
        t=t+1;
        for j=1:1:n
            T(j)=tt^(j-1);
        end
        mt=m*T';
        x(t) = Z_x*mt;
        y(t) = Z_y*mt;
    end
    r_t = sqrt(y.^2+x.^2);
    current_fitness(i) = sum((2-r_t).^2) ;
    if rem(iter,20)==0
        plot(x,y); hold on
        plot(Z_x(2),Z_y(2),'kx')
        plot(Z_x(3),Z_y(3),'kx')
    end
end
for i = 1 : s
    if current_fitness(i)      < local_best_fitness(i)
        local_best_fitness(i) = current_fitness(i)      ;
        local_best_position(i,:) = current_position(i,:) ;
    end
end
[current_global_best_fitness,g] = min(local_best_fitness) ;
if current_global_best_fitness < global_best_fitness
    global_best_fitness = current_global_best_fitness ;

```



```
for i=1:s
    globl_best_position(i,:) = local_best_position(g,:) ;
end
end
R1 = rand(s,n) ;
R2 = rand(s,n) ;
velocity = w *velocity + c1*(R1.*(local_best_position-current_position)/dt) +
c2*(R2.*(globl_best_position-current_position)/dt) ;
current_position = current_position + velocity*dt ;
end
[Jbest_min,I] = min(current_fitness) % asgari amaç fonksiyon değeri
current_position(I,:) % en iyi çözüm
```



### Sonuç

Jbest\_min =

8.5167e-05

I =

10

ans =

2.0141 1.2968 0.9081 1.9774



### 3.10 Genetik algoritma

Genetik algoritma (GA), doğadaki değişim süreçlerinden esinlenilerek geliştirilen sezgisel bir tarama yöntemidir. Geneli itibari ile gürbüz ve esnek bir yaklaşım olup gerek ayrık ve gerekse de sürekli karaktere sahip pek çok sorunun çözümünde kullanılabilir<sup>(80)</sup>. Genetik algoritmaların içinde bulunduğu genel çözüm yöntemleri grubu, evrimsel algoritmaları olarak adlandırılmakta olup esasen evrimsel algoritmalar da yapay zekâ biliminin bir alt dalı niteliğindedir<sup>(81)</sup>. Kuramsal bazda evrimsel algoritmaların kökeninin 1950'li yıllara dayandığı söylenebilir. Özellikle A. M. Turing'in (1912-1954) 1950 tarihine ait “**Hesaplama makinesi ve zekâ<sup>ii</sup>**” adlı çalışması, evrimsel algoritmaların temellerini oluşturan pek çok kavramı içinde barındırmaktadır.

Turing'in öncelikle dikkat çektiği husus, tasarlanması düşünülen makinenin nasıl çalışabileceği hususuydu. Bu noktada yaptığı ilk yaklaşım, makinenin önceden yapması gereken işleri bilmesi, dolayısıyla herhangi bir iş verildiğinde hafızasındaki talimatlara göre işlemi icra etmesiydi. Bu nedenle böyle bir makine ancak kendisine söylenen veya belirtilen işlemi, bildiği talimatlar çerçevesinde yapabilir, sürpriz niteliğinde herhangi bir davranış sergileyemezdi. Olgun bir zekânın bir manada özelliğini yansıtan söz konusu makine yaklaşımı yerine Turing, çocuk zekâsının örnek alınmasını tavsiye etmekteydi. Böylelikle sorun iki yöne ayrılmakta, çocuk zekâsı ve eğitim süreci olarak tanımlanmaktaydı. Eğitim süreci kapsamında ilk denemelerin başarılı olamayacağını, ancak deneme ve yanılma süreci ile daha iyi sonuçlara ulaşılabileceğini, bu yaklaşımın evrimsel süreçlerle doğrudan bağlantılı olduğunu belirtmekteydi. Çözümün başlangıcını miras, çözümü bulma adına yapılan değişiklikleri mutasyon ve elde edilen sonucu değerlendirmeyi de doğal seçim olarak adlandırmıştı. Turing'in dikkat çeken diğer bir düşüncesi, rassal öğeye sahip dijital bilgisayar fikriydi. Özellikle rassallığın, sorunun çözümü adına yapılan tarama işlemlerinde faydalı olacağını öngörmekteydi. A. Turing'in başka bir öngörüsü de sadece evrimsel sürece dayalı olarak yapılacak tarama işleminin yavaş olacağı, dolayısıyla zekâya dayalı bir yetenek ile sürecin hızlandırılabilceği düşüncesiydi<sup>(82)</sup>.

Aynı dönemin ortalarına doğru, 1954 yılında N. A. Barricelli (1912-1993) adlı diğer bir matematikçinin, ilkel de olsa doğrudan bilgisayarlarda evrimsel hesaplama yöntemlerini denediğini görmekteyiz. Ancak daha somut bir yöntem olarak öne sürülen algoritmaların çıkışı 1960'lı yıllara rastlar. 1960 yılında I. Rechenberg ve H. J. Bremermann tarafından dillendirilen evrimsel hesaplama yöntemleri, daha sonra değişik dallara ayrılarak çeşitlenmiştir<sup>(83)</sup>.

<sup>i</sup> Evolutionary algorithms

<sup>ii</sup> “Computing machinery and intelligence”





Bu meyanda;

- L. J. Fogel tarafından 1962 yılında tanıtılan evrimsel programlama<sup>i</sup>,
- Yine I. Rechenberg tarafından 1965 yılında gündeme getirilen evrim stratejileri<sup>ii</sup>,
- Müteakiben 1970 yılında J. H. Holland tarafından geliştirilen genetik algoritmalar<sup>iii</sup>,
- 1992 yılında G. R. Koza tarafından geniş kapsamlı olarak ele alınan genetik programlama<sup>iv</sup>,
- Son olarak 1996 yılında R. Storm tarafından gündeme getirilen diferansiyel evrim algoritması<sup>v</sup>

evrimsel hesaplama yöntemleri arasında yaygın olarak kullanılan yöntemlerdir<sup>(84)</sup>. Bununla beraber ifade edilen evrimsel algoritmalar arasında en tanınan yöntemler grubunun genetik algoritmalar olduğu söylenebilir<sup>(85)</sup>.



**Şekil 3.38** İnsana ait kromozom görüntüsü.

Genetik algoritmaların temeli, doğal seçim ve bu meyanda çevreye uyum kavramlarına dayanır. Yaşam döngüsü içinde canlıların, yaşadıkları çevreye uyum sağlamalarına bağlı olarak hayatta kalabildikleri düşünülmektedir. Çevreye uyum süreci, canlıların fiziksel özelliklerini belirleyen gen

---

<sup>i</sup> Evolutionary programming

<sup>ii</sup> Evolution strategies

<sup>iii</sup> Genetic algorithms

<sup>iv</sup> Genetic programming

<sup>v</sup> Differential evolution

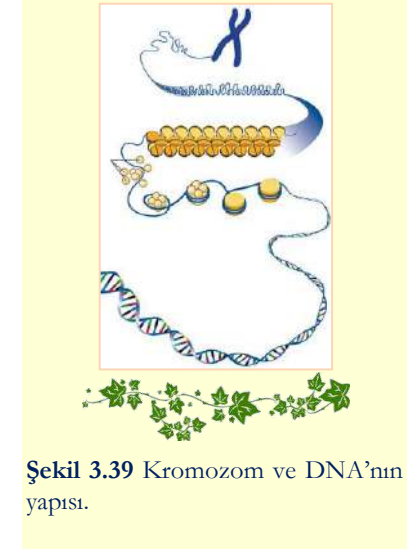


yapılarındaki deęişimlerle yönetilir. Genetik bilimi, biyolojinin bir dalı olup esas itibariyle canlılardaki kalıtım, gen ve genlerdeki deęişimi inceleyen bir bilim dalıdır<sup>(86)</sup>. Kelimenin kökeni eski Yunan dilindeki “genetikos” sözcüğü olup çıkış kaynağı itibari ile sözlük anlamı köken<sup>ii</sup>dir. Bilindiğı üzere yeryüzündeki her canlı mikro boyuttaki milyarlarca hücreden oluşur. Her hücre bünyesinde çekirdek ve her çekirdek de içinde kromozom denen DNA<sup>iii</sup> sarmalları içerir. DNA zinciri nükleotid denilen dizinlerden oluşur. Temelde dört farklı nükleotid yapısı mevcut olup bunlar; adenin, sitozin, guanin ve timin olarak tanımlanır. Baz olarak da adlandırılan bu yapılar esasen azot, hidrojen, oksijen ve karbon atomlarından kurulu moleküler yapılardır. Canlılara göre deęişse de bir kromozomda  $10^7$  ila  $10^{10}$  arası baz çifti bulunur. Bir manada DNA sarmalının yatay satırları olan baz çiftleri, genel olarak gruplanabilir ve gen yapısını oluştururlar. Genlerin baz çiftleri sayısı ise  $10^3$  ila  $10^6$  arasında deęişebilir<sup>(87)</sup>.

Canlılara ait fiziksel özellikler, sahip olunan genler sayesinde nesilden nesile kalıtsal olarak aktarılır. Örneğin insanın göz rengini kontrol eden gen, belirli bir kromozomun belirli bir bölgesinde yer alır. Aynı kişisel özelliğı farklı gen dizinleri kontrol edebilir. Bu tür gen seçeneklerine allel adı verilir. Örneğin mavi göz rengi yerine yeşil göz rengi, farklı DNA dizinleri içeren ama aynı özelliğı kontrol eden allellerdir. Genin DNA dizisine genotip<sup>iv</sup>, bu dizinin kişisel özellikteki karşılığına fenotip<sup>v</sup> adı verilir. Genler bir veya birden fazla fiziksel özelliğı kontrol edebilirler. Şekil 3.39’de bir kromozoma ve DNA sarmalına ait yapı, müteakip çizimde ise sanal bir DNA yapısı görülmektedir.



Şekil 3.40 Sanal DNA görüntüsü.



Şekil 3.39 Kromozom ve DNA’nın yapısı.

<sup>i</sup> Genetics

<sup>ii</sup> Origin

<sup>iii</sup> Deoxyribonucleic acid (DNA)

<sup>iv</sup> Genotype

<sup>v</sup> Phenotype





Canlılar arasındaki eşleşme ve üreme yoluyla eşlerin kendilerine ait DNA zincirleri, kopyalanma ve rassal olarak tekrar birleşme süreçleri geçirirler. DNA dizinindeki değişiklikler, doğal olarak sonraki nesillere aktarılmış olur. Genler, çevreye uyum bağlamında canlının pek çok karakteristik özelliğini değişik şekillerde etkilerler. Tüm bu etkileşim ortak bir olguya, “uyum” veya adaptasyonla şekillenir. Benzer şekilde canlılardaki değişiklikler, çevresel belirsizlik ve değişikliklerin bir göstergesi niteliğindedir<sup>(88)</sup>. Bünyesinde adaptasyon içeren sistemler, büyük oranda yapının her kademesine etki eden “operatörlerin” karışımı ile karakterize edilirler. Operatörlerin dâhil olduğu kavram, “adaptasyon planı” olarak adlandırılmakta olup söz konusu plan hangi canlıların hangi özelliklerinden dolayı yaşamaya devam edeceğini, hangi canlıların ise hangi eksikliklerinden dolayı yeryüzünden silineceğini belirler. Değişik ortamlarda aynı yapı değişik uyum özellikleri gösterir, ancak sadece adaptasyon planı yapının çevreye daha iyi uyum sağlayıp sağlamadığına karar verir<sup>(89)</sup>. Genel anlamda adaptasyon planı üç temel etmeni içerir. Bu etmenler; kromozomların genetik temsili, onların uyum yeterliliğini tespit eden uygunluk fonksiyonu tanımlaması ve son olarak genetik kombinezonu değiştiren adaptasyon operatörleridir. Genetik algoritmalarda dikkate alınan temel adaptasyon operatörleri çaprazlama ve mutasyondur.

Kromozomların genetik temsili hususu doğrudan doğruya adaptasyonun dili ile ilgilidir. Gelecek nesillerin çevreye uyum sadedinde daha iyi bir performans gösterebilmeleri, çevreden elde edilen geri besleme bilgisinin genlere uygun bir şekilde yansıtılmasına, dolayısıyla genler ile çevre arasında iyi bir iletişim dilinin kurulmasına bağlıdır. İyi bir iletişimin kurulabilmesi için seçilen dilin bazı önemli özellikleri göstermesi gerekir; bu özelliklerden ilki, dili oluşturan unsurların gerek genlerin temsili ve gerekse de genler üzerinde işlem yapan operatörlerin ifade edilmesi için uygun olmasıdır. İkincisi, dil dâhilindeki unsurların ifade ettiği mana, unsurun bulunduğu yerden bağımsız olmalıdır. Üçüncüsü, kullanılan dil nesiller boyu süre giden süreçte etkinliğini muhafaza etmelidir<sup>(90)</sup>. Son yıllarda değişik dil tanımlama (kodlama) yöntemleri geliştirilmiş ve genetik algoritma dâhilinde uygulanmıştır.

Değişik problemlerde değişik kodlama yöntemleri kullanmak algoritmanın etkinliğini arttırabilir. Kullanılan dilleri dört kategoriye ayırmak mümkündür. Bunlar; ikili kodlama (0 ve 1), reel sayı kodlaması, tam sayı kodlaması ve genel veri yapı kodlamasıdır. Optimizasyon problemlerinde reel sayı kodlaması en çok tercih edilen yöntemdir. Reel kodlama, gerek genotip ve gerekse de fenotip arasındaki ilişkinin gerçek problemle örtüşmesini sağlar. Aynı örtüşme, genetik operatörler ile gerçek çözüm uzayı arasında da geçerlidir. Doğadaki kromozom veya birey, bireye ait genler ve çevreye uyum kavramları genetik algoritmalarda tasarım yöneyi, tasarım değişkenleri





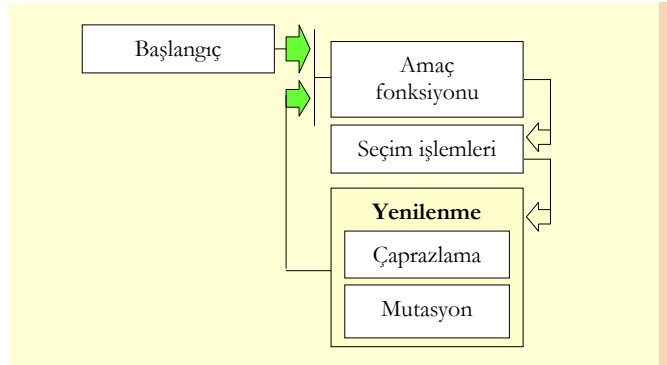
ve amaç fonksiyonu değeri olarak dikkate alınır. Buna göre reel kodlama dâhilinde herhangi bir kromozom veya diğer bir ifade ile popülasyondaki birey;

$$x_{i,j}, i=1,2,\dots,N_p, j=1,2,\dots,n$$

şeklinde ifade edilebilir. Burada  $x_i$  nüfus içerisindeki  $i$ 'nci bireyi temsil eden kromozomu,  $x_{i,j}$  o kromozoma ait  $j$ 'nci geni,  $n$  değişken niteliğindeki gen sayısını ve  $N_p$  nüfus büyüklüğünü göstermektedir. Genetik algoritmanın çözüm arama sürecindeki temel işlem adımları aşağıdaki gibi sıralanabilir;

- ▶ 1. Optimizasyon sürecine ait temel parametre ve tanımlamaların yapılması,
- ▶ 2. Başlangıç nüfusunun oluşturulması,
- ▶ 3. Amaç fonksiyonu değerlerinin hesaplanması,
- ▶ 4. Amaç fonksiyon değerlerine bağlı olarak uygunluk değerlerinin elde edilmesi ve eş seçimi yapılması,
- ▶ 5. Çaprazlama ve mutasyon işlemlerinin icra edilmesi,
- ▶ 6. Daha önce ifade edilen üçüncü maddeye gidilerek yakınsama ölçütleri sağlanıncaya kadar tekrarlı işlemlere devam edilmesi.

Algoritma sürecinin akış şeması müteakip diyagramda görülmektedir;



**Diyagram 3.3** Genetik almıtmaya ait akış şeması.



Genetik algoritma kapsamında optimizasyon sürecine ait temel parametre ve tanımlar arasında müteakip hususlar belirtilebilir;

- Amaç fonksiyonunun tanımlanması,
- Yakınsama ölçütünün tespit edilmesi,
- Nüfus büyüklüğünün belirlenmesi,
- Amaç fonksiyon değerinin ölçeklendirilmesi için yöntem seçimi,
- Seçim mekanizmasının belirlenmesi,
- Çaprazlama oranı ve yönteminin tanımlanması,
- Mutasyon oranı ve yönteminin belirlenmesi,
- Elitizm uygulamasına dair yaklaşımın tanımlanması.

Görüldüğü üzere daha başlangıç aşamasında yönteme dair pek çok tanımlama ve değer belirleme söz konusudur. Yapılan tanımlamalar sonrasında optimizasyon sürecinin başlayabilmesi için başlangıç nüfusuna ihtiyaç duyulur. Tıpkı diğer nüfus bazlı yöntemlerde olduğu üzere, başlangıç nüfusunu değişik şekillerde üretmek mümkündür. İlk yöntem, popülasyondaki her bireyin rastsal sayı operatörü vasıtasıyla üretilmesidir. Bu yöntem, çözüm hakkında hiçbir fikrin olmadığı optimizasyon problemleri için tercih edilir;

$$x_{i,j}^{(0)} = x_j^l + (x_j^u - x_j^l)rand(0,1), \begin{matrix} i=1,2,\dots,N_p \\ j=1,2,\dots,n \end{matrix}$$

Burada  $[x_j^l, x_j^u]$  ifadeleri  $j$ 'ninci gene ait alt ve üst sınır değerlerini ifade etmektedir. İkinci yöntem, problem hakkında tecrübeye sahip olunmasını gerektirir. Olası çözümler hakkında tecrübesi olan bir tasarımcı, başlangıç nüfusunu en uygun çözüme yakın olacak şekilde üretebilir. Bu durumda optimizasyon süreci, optimal çözüme yakın bir noktadan başlayacaktır. Bu ise yakınsama için gerekli süreyi oldukça kısaltabilir.

Değerlendirmeye tabi nüfusun belirlenmesi akabinde amaç fonksiyonu değerleri hesaplanabilir. Popülasyondaki bireylerin amaç fonksiyonunu ya da diğer bir ifade ile uygunluk değerlerinin belirlenmesini takiben genellikle yapılan ilk işlem, nüfus dâhilindeki en iyi birey ya da bireylerin belirlenmesi ve elit bireyler olarak kayıt altına alınmasıdır.

### 3.10.1 Elitizm

Genetik algoritma dahilinde elitizm kavramı, yakınsama süreci açısından önem arz eden bir husus olup optimizasyon sürecinde ulaşılan en iyi çözümün nesilden nesile aktarılmasını, bu





sayede uygunluk değerinin azalmamasını sağlar. Elit birey sayısı  $N_e$ , toplam nüfus büyüklüğüne göre belirlenmekle beraber genellikle nüfusun %10'u mertebesinde birey elit birey olarak muhafaza edilebilir. Elit bireyler, amaç fonksiyonu sıralamasında en iyi durumdaki bireylerden seçilir. Her nesilde elit birey güncellemesi müteakip usul çerçevesinde yapılır;

$$f^{el}(t) = \underset{f_i(t)}{\operatorname{argmin}} f(x_i(t)) \quad (3.98)$$

$$x^{el}(t) = \begin{cases} x^{el}(t-1) & f^{el}(t) > f^{el}(t-1) \\ x^{el}(t) & f^{el}(t) \leq f^{el}(t-1) \end{cases} \quad (3.99)$$

Burada  $t$  nesil sayısını,  $f^{el}(t)$  elit bireye ait amaç fonksiyon değerini ve  $x^{el}(t)$  o nesildeki elit bireye ait tasarım yöneyini göstermektedir.

### 3.10.2 Seçim

Genetik algoritmanın altında yatan genel prensiplerden birisi de doğal seçimdir. Doğal seçim, genetik algoritma içinde adeta süreci tahrik eden unsur görünümündedir. Günümüze kadar pek çok seçim yöntemi teklif edilmiş, denenmiş ve verimlilikleri karşılaştırılmıştır. Literatürdeki en basit eş seçimi yaklaşımı, amaç fonksiyon değeri sıralamasına göre sıra ile çift seçimidir. Örneğin eşleşmede erkek olarak belirlenebilecek bireyler sıralamada tek sayılı konumlar (1,3,5,...), dişi olarak dikkate alınabilecek bireyler sıralamada çift sayılı konumlar (2,4,6,...) olabilir. Böyle bir yaklaşımda amaç fonksiyonu kötü olan bireylerin gelecek nesle gen aktarma şansı yoktur. Buna mukabil tamamen rassal olarak eş belirleme yöntemi diğer basit bir yaklaşımdır. Bu yaklaşımda eşleşecek erkek ( $x_e$ ) ve dişi ( $x_d$ ) tamamen rassal olarak belirlenebilir;

$$x_e = \operatorname{randint}[1 \ N_s] \quad (3.100)$$

$$x_d = \operatorname{randint}[1 \ N_s] \quad (3.101)$$

İfadelerde geçen  $N_s$  terimi, çaprazlama için muhafaza edilecek birey sayısını ifade etmektedir. Eş belirleme sürecinde eklenmesi gereken diğer bir husus, aynı bireyin hem erkek ve hem de dişi olamayacağıdır. Bu bireyler aynı zamanda çaprazlama sonrası nesle aynen aktarılırlar. Çaprazlamada dikkate alınacak birey sayısı,

$$N_s = N_p(1 - P_c) \quad (3.102)$$

ifadesine göre tespit edilir. Burada  $P_c$  çaprazlama oranını göstermektedir. Tabiatıyla çaprazlama ile elde edilecek birey sayısı da;



$$N_c = P_c N_p \quad (3.103)$$

eşitliğine göre belirlenebilir. Basit yaklaşımlardan ziyade olasılık üzerine kurulu ve yaygın olarak kullanılan diğer eş seçim yöntemleri; sıralama ağırlığı, amaç değeri ağırlığı, uygunluk değeri ve turnuva<sup>i</sup> seçimi yöntemleridir.

Sıralama yaklaşımında dikkate alınan temel unsur, bireylerin amaç fonksiyonu değerine göre yapılan sıralamadaki konumudur. Optimizasyon sürecinin amacına göre yapılan küçükten büyüğe veya büyükten küçüğe sıralamasında bulunulan konum dikkate alınarak her bir bireye ait seçilme olasılık değerleri;

$$P_m = \frac{N_s - m + 1}{\sum_{n=1}^{N_s} m} \quad (3.104)$$

denkleme göre belirlenebilir. Burada  $m$  değeri, amaç fonksiyonu değer sıralamasındaki konumu ifade etmektedir. Yapılan tanıma göre aynı konumdaki bireyin seçilme olasılığı her zaman aynı olup nesilden nesile değişim göstermez. Yaklaşımın daha iyi anlaşılması için basit bir örnekle somutlaştırmaya gidilebilir. Örneğin nüfus büyüklüğü  $N_p = 10$  ve çaprazlama oranı  $P_c = 0.5$  olan bir tarama sürecinde çaprazlama için dikkate alınabilecek birey sayısı;

$$N_s = N_p(1 - P_c) = 10(1 - 0.5) = 5$$

olarak kolayca hesaplanabilir. Buna mukabil amaç fonksiyonu değeri sıralamasındaki ilk 5 bireye ait sıralama ağırlık değerleri;

$$P_m = \frac{N_s - m + 1}{\sum_{n=1}^{N_s} m} = \frac{5 - m + 1}{1 + 2 + 3 + 4 + 5} = \frac{6 - m}{15}$$

ifadesinden hareketle;

$m$	$P_m$	$\sum_{i=1}^m P_m$
1	0.333	0.333
2	0.266	0.599
3	0.200	0.799
4	0.133	0.932
5	0.066	1

şeklinde elde edilebilir.

<sup>i</sup> Rank weighting, cost weighting, fitness ratio, and tournament selection methods



Sıralama ağırlık değeri tespiti sonrasında eş seçimi, rassal sayı üretimi ve bu sayının denk geldiği olasılık aralığı (bir anlamda çark-ı felek) baz alınarak yapılır. Örneğin  $rand = 0.4766$  sayısı 0.333 ile 0.599 sayı aralığına denk gelmekte olup buna göre  $x_e$  olarak 2 numaralı birey seçimi yapılabilir. Akabinde yapılacak diğer rassal sayı üretimi dâhilinde, örneğin  $rand = 0.7985$  ve bu sayının denk geldiği 0.599 ile 0.799 sayı aralığı dikkate alınarak,  $x_d$  için 3 numaralı birey seçimi yapılabilir. Bu işlemler çaprazlama için gerekli birey sayısına ulaşıncaya kadar tekrar edilir. Tahmin edileceği üzere sıralamada önde bulunan bireylerin çaprazlamada rol alma şansı daha fazladır.

Amaç değeri ağırlığı yönteminde, sıralamadan ziyade amaç değerlerinin büyüklüğü önem arz eder. Bu meyanda seçilme olasılık değerleri;

$$P_m = \left| \frac{F_m}{\sum_{n=1}^{N_s} F_m} \right| \quad (3.105)$$

$$F_m = f_m - f_{N_s+1} \quad (3.106)$$

denklemlerine göre hesaplanabilir. Burada  $m$  indisi sıralamadaki konumu ve  $f_m$  o konumdaki bireye ait amaç fonksiyon değerini temsil etmektedir. Minimizasyon amaçlı bir eniyileme sürecinde  $F_m$  değerinin her zaman negatif çıkması nedeniyle, olasılık değeri hesaplamasında mutlak değer dikkate alınmaktadır. Amaç değeri ağırlığı yönteminin somutlaştırılması çerçevesinde müteakip örnek dikkate alınabilir. Buna göre nüfus büyüklüğü ve çaprazlama oranı bir önceki örnekle aynı olan bir süreçte, ilk 6 bireye ait amaç fonksiyon değerlerinin;

$m$	1	2	3	4	5	$N_s + 1 \rightarrow 6$
$f_m$	205	356	433	712	966	1020

şeklinde sıralanması halinde bireylere ait seçilme olasılık değerleri;

$m$	$F_m$	$P_m$	$\sum_{i=1}^m P_m$
1	$205 - 1020 = -815$	0.335	0.335
2	$356 - 1020 = -664$	0.273	0.608
3	$433 - 1020 = -587$	0.241	0.849
4	$712 - 1020 = -308$	0.126	0.975
5	$966 - 1020 = -54$	0.022	1
	$\sum_{n=1}^5 F_m = -2428$		



olarak elde edilebilir. Amaç fonksiyon değerine bağlı olarak tespit edilen seçilme olasılık değerleri sonrasında eş seçimi, rassal sayı üretimi ve bu sayının denk geldiği olasılık aralığı temel alınarak yapılır. Örneğin  $rand = 0.015$  sayısı 0 ile 0.335 sayı aralığına denk gelmekte olup buna göre  $x_e$  olarak 1 numaralı birey seçimi yapılabilir. Akabinde yapılacak diğer rassal sayı üretimi sonucunda, örneğin  $rand = 0.541$  ve bu sayının denk geldiği 0.335 ile 0.608 sayı aralığı dikkate alınarak,  $x_d$  için 2 numaralı birey seçimi yapılabilir. Bu işlemler çaprazlama için gerekli birey sayısına ulaşıncaya kadar tekrar edilir. Görüldüğü üzere amaç fonksiyon değerinin değişimine bağlı olarak her nesilde konuma göre seçilme olasılığı değişebilmekte ve göreceli olarak sahip olunan amaç fonksiyonu değer farkı, çaprazlamada rol alma şansını belirlemektedir.

Uygunluk değeri yönteminde, rastlantı ile belirlenimci yaklaşımların karma bir uygulaması dikkate alınır. Bu meyanda öncelikle tüm bireylerin, amaç fonksiyon değerine göre küçükten büyüğe sıralaması gerçekleştirilir. Daha sonra her bir bireye ait uygunluk değeri;

$$U_i = \text{yuvarla} \left\{ \frac{1/f_i}{\sum_{j=1}^{N_p} 1/f_j} N_p \right\} \quad (3.107)$$

ifadesine göre hesaplanır. Çaprazlama dâhilinde belirlenmesi gereken eşlerden ilki, uygunluk değeri en yüksek olan bireyden başlanmak üzere sırayla seçilir. Eşlerden diğeri, nüfus bünyesindeki öteki tüm bireyler arasından rassal olarak seçilir;

$$x_e = [rand \times N_p] \quad (3.108)$$

Eş olarak sırayla seçilen bireylerin uygunluk değeri, her çaprazlama işlemi sonrasında azaltılarak ( $U_i = U_i - 1$ ) sonraki en iyi bireyin çaprazlamaya tabi tutulması sağlanmış olur. Uygunluk değeri yönteminin somutlaştırılması çerçevesinde müteakip örnek dikkate alınabilir. Buna göre nüfus büyüklüğü  $N_p = 6$  olan bir süreçte bireylere ait sıralanmış amaç fonksiyon değerlerinin;

$m$	1	2	3	4	5	6
$f_m$	205	356	433	712	966	1020

şeklinde olması halinde ilk bireye ait uygunluk değeri;



$$\sum_{j=1}^6 1/f_j = 1/(205 + 356 + 433 + 712 + 966 + 1020) = 0.0134$$

$$U_1 = \text{yuvarla} \left\{ \frac{1/f_1}{\sum_{j=1}^6 1/f_j} 6 \right\} = \text{yuvarla} \left\{ \frac{1/205}{0.0134} 6 \right\} = 2$$

olarak hesaplanabilir. Dolayısıyla amaç fonksiyon değeri açısından en iyi olan bireyin, iki defa çaprazlama hakkına sahip olduğu söylenebilir. Popülasyon dâhilindeki diğer bireylere ait uygunluk değerleri aşağıdaki gibi elde edilebilir;

<i>m</i>	1	2	3	4	5	6
<i>U<sub>i</sub></i>	2	1	1	1	0	0

Çaprazlama dâhilindeki diğer eş, ifade edildiği üzere rassal olarak belirlenir. Örneğin ilk bireyin ilk çaprazlama işleminde eş olarak;

$$\text{rand} = 0.776, \quad \mathbf{x}_e = [\text{rand} \times 6] = [4.6596] = 5$$

değerinden hareketle 5inci birey seçilebilir. Çaprazlama sonrasında ilk bireye ait uygunluk değeri  $U_1 = U_1 - 1 = 2 - 1 = 1$  şeklinde güncellenir.

Eş seçiminde kullanılabilen ve nispeten basit bir yaklaşım olan turnuva yönteminde, rastgele seçilen iki veya üç birey arasından amaç fonksiyonu en iyi olan birey eş olarak seçilerek çaprazlama sürecine geçilir. Müteakip örnekte eşleşme dâhilindeki erkek bireyin seçimi görülmektedir;

$$\begin{aligned} \mathbf{x}_{e1} &= \text{randint}[1 \quad N_p] \\ \mathbf{x}_{e2} &= \text{randint}[1 \quad N_p] \rightarrow \mathbf{x}_e = \min[f(\mathbf{x}_{e1}) \quad f(\mathbf{x}_{e2}) \quad f(\mathbf{x}_{e3})] \\ \mathbf{x}_{e3} &= \text{randint}[1 \quad N_p] \end{aligned} \quad (3.109)$$

Benzer işlemler diğer dişi ve erkek eşler için tekrarlanır. Turnuva yönteminin en önemli yönü, tüm popülasyonu içeren herhangi bir amaç fonksiyonu sıralamasına ihtiyaç duyulmaması ve ayrıca doğadaki eş seçimi uygulaması ile büyük benzerlik göstermesidir. Eşleşecek bireylerin belirlenmesini müteakip yenilenme<sup>i</sup> safhasına geçilir.

<sup>i</sup> Reproduction phase



### 3.10.3 Çarpazlama

Genetik algoritma kapsamında çözüm uzayının taranmamış bölgelerinin değerlendirilmesi, yenilenme safhasında dikkate alınan çarpazlama ve akabinde mutasyon<sup>i</sup> operatörleri ile gerçekleştirilir. Biyolojik sistemlerde çarpazlama, kromozom dâhilindeki genlerin yeniden düzenlenmesini ve bu sayede ebeveynlerin özelliklerinin yeniden düzenlenerek yeni nesillere geçirilmesini sağlar. Bazen bu düzenleme, daha iyi bireylerin doğmasına vesile olabilir, bazen de nesil kötüye gidebilir<sup>(91)</sup>. Değişim, genin yerinden ziyade kendisinde oluşur. Çarpazlamanın, gen havuzunda doğrudan etkisi söz konusu olup en önemli yansıması yeni bireylerin, önceki nesildeki bireylerden hiçbiriyile aynı olmamasıdır. Diğer taraftan eşleşme kapsamında çarpazlamaya tabi tutulan iki bireye karşılık süreç sonunda yine iki bireyin elde edilmesi esastır. Literatürde değişik çarpazlama yöntemleri söz konusu olup bu meydana yeknesak, tek-nokta, iki-nokta, ara-nokta, sezgisel, aritmetik ve BLX- $\alpha$ <sup>ii</sup> yöntemlerinden bahsedilecektir. Yöntemlerin açıklanması öncesi erkek eşin ( $e$ ), dişi eşin ( $d$ ) ve elde edilecek yeni neslin ( $c$ ) üst indisi ile temsil edildiğini, bu meydana tasarım yöneyleri ve amaç fonksiyonu değerlerinin;

$$x_j^e, x_j^d, x_j^c, \quad j = 1, 2, \dots, n$$

$$f^e, f^d$$

şeklinde belirtildiğini ifade edelim.

#### a. Yeknesak çarpazlama

Yeknesak çarpazlama yöntemi dâhilinde yeni nesil;

$$\begin{array}{ll} \text{Eğer} & r < 0.5 \quad x_j^c = x_j^d \quad j=1,2,\dots,n \\ \text{Aksi halde} & r \geq 0.5 \quad x_j^c = x_j^e \end{array} \quad (3.110)$$

ifadelerine göre belirlenir. Burada  $r$  değeri [0 1] aralığında değişebilen rassal sayı olup yeni neslin genlerinin muhtemelen yarısı anneden, diğer yarısı ise babadan gelir.

#### b. Tek-nokta çarpazlama

Tek-nokta çarpazlama yöntemi dâhilinde yeni nesil;

<sup>i</sup> Crossover and mutation

<sup>ii</sup> Uniform, single point, two-point, intermediate, heuristic, arithmetic, and blend crossover as BLX- $\alpha$



$$x_j^c |^{j=1,2,\dots,n} = [x_1^d, x_2^d, \dots, x_p^d, x_{p+1}^e, x_{p+2}^e, \dots, x_n^e] \quad (3.111)$$

ifadesine göre belirlenir. Burada  $p$  değeri,  $[1, n]$  aralığında değişebilen tam sayı niteliğindeki rassal sayıdır. Buna göre yeni neslin genlerinin bir kısmı babadan, geri kalan kısmı anneden alınmakta, gen ayırım bölgesi rassal olarak belirlenmektedir.

### c. İki-nokta çaprazlama

İki-nokta çaprazlama yöntemi dâhilinde yeni nesil;

$$x_j^c |^{j=1,2,\dots,n} = [x_1^d, x_2^d, \dots, x_p^d, x_{p+1}^e, x_{p+2}^e, \dots, x_q^e, x_{q+1}^d, x_{q+2}^d, \dots, x_n^d] \quad (3.112)$$

ifadesine göre belirlenir. Burada  $p$  ve  $q$  değerleri,  $[1, n]$  aralığında değişebilen ve birbirinden farklı tam sayı niteliğindeki rastsal sayılardır. Buna göre yeni neslin genlerinin bir kısmı babadan, geri kalan kısmı anneden alınmakta, gen ayırımı bir nokta yerine iki noktadan yapılmakta ve ayırım bölgeleri rastsal olarak belirlenmektedir.

### ç. Ara-nokta çaprazlama

Ara-nokta çaprazlama yöntemi dâhilinde yeni nesil;

$$x_j^c |^{j=1,2,\dots,n} = x_j^d + \alpha(x_j^e - x_j^d) \quad (3.113)$$

$$\alpha = uR$$

ifadesine göre belirlenir. Burada  $u$  değeri,  $[0, 1]$  aralığında değişebilen rassal sayıyı,  $R$  kullanıcı tarafından belirlenen ve etkileşim bandını belirleyen sabit bir değeri temsil etmektedir.

### d. Sezgisel çaprazlama

Sezgisel çaprazlama yöntemi dâhilinde yeni nesil;

$$\begin{aligned} \text{Eğer } f^d < f^e \quad x_j^c &= x_j^e + R(x_j^d - x_j^e) \\ \text{Eğer } f^e < f^d \quad x_j^c &= x_j^d + R(x_j^e - x_j^d) \end{aligned} \quad j=1,2,\dots,n \quad (3.114)$$

ifadesine göre belirlenir.



Burada  $R$  kullanıcı tarafından belirlenen ve bir önceki yaklaşıma benzer şekilde etkileşim bandını belirleyen sabit bir değeri temsil etmektedir.  $R$  değeri genellikle 1.2 ve üzeri mertebelerde alınabilir.

#### e. Aritmetik çaprazlama

Aritmetik çaprazlama yöntemi dâhilinde yeni nesil;

$$x_j^e |^{j=1,2,\dots,n} = \lambda x_j^d + (1 - \lambda)x_j^e \quad (3.115)$$

ifadesine göre belirlenir. Burada  $\lambda$  değeri, [0 1] aralığında değişebilen rassal sayıyı temsil etmektedir.

#### f. BLX- $\alpha$ çaprazlama

BLX- $\alpha$  çaprazlama yöntemi dâhilinde yeni nesil;

$$x_j^{c1} |^{j=1,2,\dots,n} = \alpha x_j^d + (1 - \alpha)x_j^e \quad (3.116)$$

$$x_j^{c2} |^{j=1,2,\dots,n} = (1 - \alpha)x_j^d + \alpha x_j^e \quad (3.117)$$

$$\alpha = \gamma u - 0.5 \quad (3.118)$$

ifadesine göre belirlenir. Burada  $u$  değeri, [0 1] aralığında değişebilen rassal sayıyı,  $\gamma$  kullanıcı tarafından belirlenen sabit değeri temsil etmektedir. Görüldüğü üzere  $\gamma$  değeri, etkileşim bandını belirleyen değer niteliğinde olup atanan  $\gamma$  değerine göre gen değerleri arasında veya dışında nesil eldesi mümkün görünmektedir.

### 3.10.4 Mutasyon

Mutasyon, çaprazlamadan sonra yaygın olarak kullanılan bir diğer genetik operatördür. Genetikte mutasyon, rastgele seçilen bir kromozomun rastgele seçilen bir genine uygulanır ve bu değişiklik yapının aynı nesilde farklılaşmasını sağlar. Ancak sayısal uygulamalarda mutasyon, belirli kromozomların belirli genlerini kapsayabileceği gibi belirli kromozomların tüm genlerini ve hatta tüm kromozomların tüm genlerini de kapsayabilir. Diğer taraftan doğal seçim, mutasyonların neden olduğu değişiklikleri sadece kabul veya reddetme işlevi değil aynı zamanda gen havuzunda nesiller boyu sürediden değişikliklerin çevreye uyumunu da kontrol eder<sup>(92)</sup>.





Genellikle mutasyona uğrıtılacak gen sayısı  $G_m$ , mutasyon oranı  $R_m$  ile belirlenir;

$$G_m = nR_m(N_p - N_e) \quad (3.119)$$

Denklemden nüfus büyüklüğünün  $N_e$  birey eksiltilmesi, mutasyonun tüm bireyler yerine en iyi birey ya da bireyler hariç, diğer bireyleri kapsamayı düşüncesinden kaynaklanır. Mutasyona uğrıtılacak gen sayısının belirlenmesi sürecini müteakip örnekte görebiliriz. Buna göre; tasarım yöneyi değişken sayısının  $n = 4$ , nüfus büyüklüğünün  $N_p = 10$ , elit birey sayısının  $N_e = 1$  ve mutasyon oranının da  $R_m = 0.1$  olduğu bir genetik algoritma sürecinde mutasyona uğrıtılacak gen sayısı;

$$G_m = nR_m(N_p - 1) = 4 \times 0.1 \times (10 - 1) \cong 4$$

şeklinde belirlenebilir. Uygulamadan da görüldüğü üzere sayısal olarak elde edilen değerlerde tam sayıya yuvarlama işlemi yapılır. Pratikte mutasyon oranı düşük seviyelerde tutulur. Ancak bazı yöntemlerde çok yüksek oranda mutasyonlar da uygulanabilmekte ve başarılı sonuçlar elde edilebilmektedir. Değişik çaprazlama yöntemleri olduğu gibi değişik mutasyon seçenekleri de söz konusudur. Burada yaygın olarak kullanılan Gauss, yeknesak, adaptif olası ve titreşimli mutasyon<sup>i</sup> yöntemleri ifade edilecektir. Mutasyon yöntemlerinin açıklanması öncesi mutasyona uğrıtılacak kromozomun  $p$ , elde edilecek yeni neslin  $c$  üst indisi ile temsil edildiğini, bu meyanda tasarım yöneylerinin;

$$x_j^p, x_j^c \quad j = 1, 2, \dots, n$$

şeklinde belirtildiğini, ayrıca içinde bulunulan neslin  $g$ , azami nesil sayısının ise  $G$  ile temsil edildiğini ifade edelim.

#### a. Gauss mutasyonu

Gauss mutasyon yöntemi dâhilinde yeni birey;

$$\beta_2 = \beta_1 - \gamma \beta_1 \frac{g}{G} \quad (3.120)$$

$$\beta_3 = \beta_2(x_j^u - x_j^l) \quad (3.121)$$

$$x_j^c \big|_{j=1,2,\dots,n} = x_j^p + \beta_3 u \quad (3.122)$$

ifadelerine göre belirlenir.

<sup>i</sup> Gaussian, uniform, adaptive feasible, and vibrational mutations



Burada  $\beta_1$  kullanıcının tanımladığı ve ölçeklendirme amacı ile kullanılan sabit bir sayıyı,  $\gamma$  yine kullanıcının tanımlayabildiği, daraltma amacı için kullanılabilen başka bir sabit sayıyı ve  $u$  terimi, değeri  $[0, 1]$  aralığında değişebilen rassal sayıyı temsil etmektedir. Görüldüğü üzere mutasyon işlemi, belirli bir bireyin tüm genlerine uygulanmakta olup nesiller geçtikçe etkisi azalmaktadır.

### b. Yeknesak mutasyon

Yeknesak mutasyon yöntemi dâhilinde yeni birey;

$$\Delta x_j = x_j^U - x_j^L \quad (3.123)$$

$$I = nR_m \quad (3.124)$$

$$x_i^C \mid^{i=1,2,\dots,I} = x_i^L + u\Delta x_i \quad (3.125)$$

ifadelerine göre belirlenir. Burada  $R_m$  mutasyon oranını ve  $u$  terimi, değeri  $[0, 1]$  aralığında değişebilen rassal sayıyı temsil etmektedir. Görüldüğü üzere mutasyon işlemi belirli sayıdaki ( $I$ ) genlere uygulanmakta, etkilenen gen sayısını doğrudan doğruya mutasyon oranı belirlemektedir.

### c. Adaptif olası mutasyon

Adaptif olası mutasyon yöntemi dâhilinde yeni birey;

$$x_j^C \mid^{i=1,2,\dots,I} = x_j^P + \alpha\beta d_j \quad (3.126)$$

ifadesine göre belirlenir. Burada  $\alpha$  adım büyüklüğünü,  $\beta$  ölçeklendirme etmenini ve  $d_j$  gidilecek yöne ait doğrultu bileşeninin temsil etmektedir. Yöntem her ne kadar mutasyon adı altında tanımlansa da, esas itibarı ile deterministik bir karaktere sahip olup gerek adım büyüklüğü ve gerekse de gidilecek doğrultu yönü genellikle tek boyutlu tarama ve gradyan esaslı yöntemlerle tespit edilir.

### ç. Titreşimli mutasyon

Şimdiye kadar tanımlanan mutasyon yöntemleri her nesilde uygulanan yöntemlerdir. Ancak titreşimli mutasyon yöntemi her nesilden ziyade belirli devirlerde nesle uygulanan bir mutasyon yöntemidir. İlave olarak mutasyon işlemleri tüm kromozomların tüm genlerine uygulanır.





Titreşimli mutasyon yöntemi dâhilinde yeni birey;

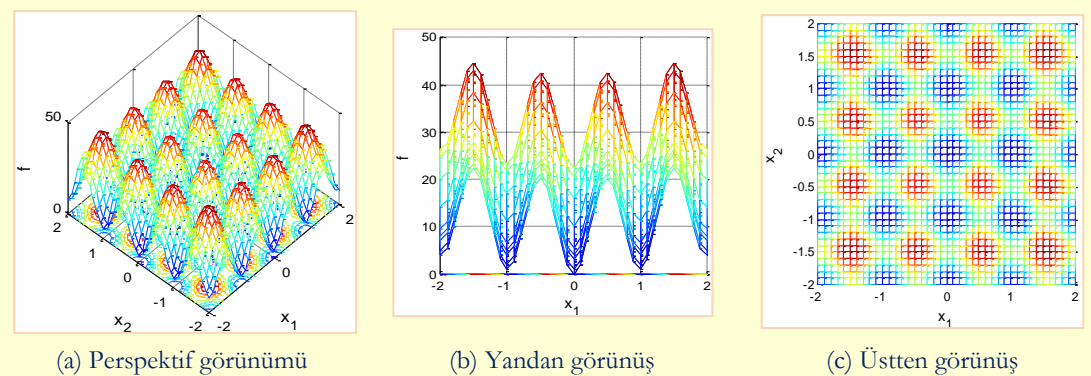
$$x_{i,j} \Big|_{\substack{i=1,2,\dots,N_p-N_e \\ j=1,2,\dots,n}} = \begin{cases} x_{i,j}[1 + w\beta(0.5 - u)] & t \bmod f_r = 0 \\ x_{i,j} & t \bmod f_r \neq 0 \end{cases} \quad (3.127)$$

ifadesine göre belirlenir. Burada  $w$  kullanıcının belirlediği ağırlık katsayısını,  $\beta$  ölçeklendirme etmenini ve  $u$  terimi, değeri  $[0, 1]$  aralığında değişebilen rassal sayıyı göstermekte,  $f_r$  simgesi de mutasyon uygulama frekansını temsil etmektedir.



Mutasyonların uygulanması sonrasında yenilenme sürecinin tamamlandığı söylenebilir. Akabinde amaç fonksiyonu değerlerinin hesaplanması ve yakınsama ölçütleri sağlanıncaya kadar benzer işlemlerin tekrar edilmesi şeklinde sürece devam edilir. Genetik algoritma yönteminin daha iyi anlaşılabilmesi için örnek bir sorunu belirtilen işlem adımlarını takip ederek çözmek mümkündür. Buna göre  $n = 2$  boyutlu bir test fonksiyonunun minimizasyonunu düşünelim;

$$\min_x f(x) = 10n + \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i)]$$



Şekil 3.41 Rastrigin fonksiyonunun farklı açılardan görüntüleri.

Şekil 3.41'den de görüleceği üzere test fonksiyonu, oldukça çok sayıda yerel tepe ve çukur noktaları olan bir fonksiyon olup fonksiyonun minimum noktasının GA yöntemi ile bulunması çerçevesinde yapılacak ilk işlem, sürece ait kontrol parametrelerinin belirlenmesidir. Bu



bağlamda; nüfus büyüklüğü  $N_p = 10$ , azami iterasyon sayısı  $T = 100$ , elit birey sayısı  $N_e = 1$ , seçim yöntemi uygunluk değeri, çaprazlama yöntemi BLX- $\alpha$  ve mutasyon yöntemi olarak da titreşimli mutasyon yaklaşımı dikkate alınabilir. Yakınsama ölçütünün tanımlanması meyanında azami tekrar sayısı dikkate alınabileceği gibi amaç fonksiyon değeri için bir ölçüt de belirlenebilir;  $f^{el} \leq 10^{-9}$ . Ayrıca başlangıç nüfusunun üretilmesi çerçevesinde değişkenlerin alt ve üst sınır değerleri,  $[-2 \ 2]$  aralığı olarak tespit edilebilir. Eniyileme süreci parametrelerinin belirlenmesi sonrasında ilk popülasyon niteliğindeki başlangıç nüfusu;

$$x_{i,j}^{(0)} = -2 + (2 - (-2)).rand(0,1)_{j=1,2,\dots,2}^{i=1,2,\dots,10}$$

ifadesi ve üretilen rassal sayılar çerçevesinde;

rand		$x_i^{(0),T}$	
0.4283	0.9788	-0.2870	1.9152
0.7470	0.1244	0.9880	-1.5023
0.7515	0.7678	1.0060	1.0712
0.2627	0.2830	-0.9490	-0.8681
0.6980	0.5455	0.7922	0.1820
0.6828	0.6036	0.7311	0.4142
0.9976	0.4451	1.9905	-0.2196
0.3396	0.2122	-0.6417	-1.1513
0.3449	0.1893	-0.6204	-1.2429
0.4861	0.9270	-0.0555	1.7081

şeklinde elde edilebilir. Oluşturulan nüfus dâhilindeki bireylerin amaç fonksiyon değerleri ve sıralaması ile uygunluk değerleri aşağıdaki gibi hesaplanabilir;

$f_i$	Sıra	$U_i$
3.1511	3	3
5.4062	4	2
12.1291	7	1
13.8989	5	1
16.1263	10	1
17.4402	1	1
22.2169	8	0
23.2605	2	0
28.7540	9	0
30.4743	6	0



Burada örnek olması açısından ilk bireye ait uygunluk değerinin;

$$U_1 = \text{yuvarla} \left\{ \frac{1/f_1}{\sum_{j=1}^{10} 1/f_j} 10 \right\} = \text{yuvarla} \left\{ \frac{1/3.1511}{0.9317} 10 \right\} = 3$$

yaklaşımı ile elde edildiği kolayca gösterilebilir. Dolayısıyla amaç fonksiyon değeri açısından en iyi olan birey, 3üncü birey olup söz konusu bireyin 3 defa çaprazlamada dikkate alınması mümkündür. Bu arada çaprazlama öncesi elit birey ve bu bireye ait amaç fonksiyon değeri kayıt altına alınarak bir sonraki nesle aktarılacak üzere muhafaza edilir;

$$f^{el}(1) = 3.1511, \quad \mathbf{x}^{el,T}(1) = [1.0060 \quad 1.0712]$$

Elitizm uygulaması sonrasında çaprazlama işlemlerine geçilir. Buna göre çaprazlamada ilk eş olarak sıralamadaki ilk birey dikkate alınır. Buna mukabil diğer eş rassal olarak seçilir;

$$\begin{aligned} \mathbf{x}_e &= \mathbf{x}_3 \rightarrow [1.0060 \quad 1.0712] \\ \mathbf{x}_d &= \mathbf{x}_{[rand \times 10]} \\ rand &\rightarrow 0.3867, \quad \mathbf{x}_d = \mathbf{x}_4 \rightarrow [-0.9490 \quad -0.8681] \end{aligned}$$

Eş seçimi sonrasında BLX- $\alpha$  çaprazlama yöntemine göre çaprazlamaya geçilir. Bu meydana etkileşim bandını belirleyen  $\gamma = 2$  değeri dikkate alınarak aşağıdaki işlemler gerçekleştirilir;

$$\begin{aligned} \alpha &= 2rand - 0.5 \rightarrow 0.6817 \\ x_1^{c1} &= 0.6817x_{d,1} + (1 - 0.6817)x_{e,1} = 0.6817 \times (-0.9490) + 0.3183 \times (1.0060) = -0.3267 \\ x_1^{c2} &= 0.6817x_{e,1} + (1 - 0.6817)x_{d,1} = 0.6817 \times (1.0060) + 0.3183 \times (-0.9490) = 0.3837 \end{aligned}$$

$$\begin{aligned} \alpha &= 2rand - 0.5 \rightarrow 0.4902 \\ x_2^{c1} &= 0.4902x_{d,2} + (1 - 0.4902)x_{e,2} = 0.4902 \times (-0.8681) + 0.5098 \times (1.0712) = 0.1206 \\ x_2^{c2} &= 0.4902x_{e,2} + (1 - 0.4902)x_{d,2} = 0.4902 \times (1.0712) + 0.5098 \times (-0.8681) = 0.0825 \end{aligned}$$

Buna göre iki ebeveynin çaprazlanması neticesinde iki yeni yöney elde edilmiş olup söz konusu yöneyler;

$$\begin{aligned} \mathbf{x}^{c1,T} &= [-0.3267 \quad 0.1206] \\ \mathbf{x}^{c2,T} &= [0.3837 \quad 0.0825] \end{aligned}$$

şeklinde dir.



İlk çaprazlama sonrasında ilk uygunluk değeri güncellenerek ( $U_1 = U_1 - 1 = 3 - 1 = 2$ ), ikinci çaprazlamaya geçilir. Görüldüğü üzere ilk sıradaki bireyin iki ilave çaprazlama şansı devam etmektedir. Bir önceki BLX- $\alpha$  çaprazlama yöntem adımları aynen tekrar edilerek işlemlere devam edilirse;

$x_i^{(1),T}$	
-0.3267	0.1206
0.3837	0.0825
-0.9195	-1.3827
1.2838	1.3026
-0.5481	1.1606
1.4986	1.6187
0.1607	-0.2212
-1.1653	1.0613
-0.8199	-0.7166
1.0060	1.0712

şeklindeki yeni nesle ulaşılmış olur. Çaprazlama işlemleri ile dokuz yeni birey elde edilmektedir. Zira mutasyon işlemleri sonrasında elit birey nesle eklenmekte ve bu sayede toplam nüfus sayısı sabit kalmaktadır. Çaprazlama süreci sonrasında mutasyon işlemlerine geçilir. Seçilen mutasyon yönteminin titreşimli mutasyon yaklaşımı olması nedeniyle operatör;

$$\begin{aligned} w &= 2 \\ \beta &= 1 \\ f_r &= 5 \end{aligned}$$

$$x_{i,j} \Big|_{j=1,2}^{i=1,2,\dots,9} = \begin{cases} x_{i,j}[1 + 2(0.5 - u)] & t \bmod 5 = 0 \\ x_{i,j} & t \bmod 5 \neq 0 \end{cases}$$

şeklinde tanımlanabilir. Buna göre  $t = 1$  olması nedeniyle herhangi bir mutasyon işlemi yapılmadan bir sonraki nesil işlemlerine geçilebilir. Bu meyanda yapılacak işlem, amaç fonksiyon değerlerinin hesaplanması ve akabinde seçim işlemlerine devam edilmesidir. Benzer işlemler yapılarak 5inci nesle gelindiğinde aşağıdaki elit ve bireylerin eldesi olasıdır;

$$f^{el}(5) = 1.1365$$

$$x^{el,T}(1) = [-0.0027 \quad 1.0215]$$



$x_i^{(5),T}$	
0.0271	1.0214
0.0167	1.0216
-0.7376	1.0122
-0.3319	1.0167
-1.0824	0.7875
0.0552	0.9920
0.4861	1.0080
-1.5685	1.0209
-0.2489	1.0277
-0.0027	1.0215

Bu adımda  $t \bmod 5 = 0$  olması nedeniyle mutasyon işlemleri uygulanır. Mutasyon, tüm bireylerin tüm genlerine uygulanmakta olup sadece elit birey süreçten hariç tutulur. Örneğin ilk bireye mutasyon işlemi aşağıdaki gibi uygulanabilir;

$$u = 0.4704$$

$$x_{1,1}^c = x_{1,1}[1 + 2(0.5 - 0.4704)] = 0.0271 \times 1.0592 = 0.0287$$

$$u = 0.1874$$

$$x_{1,2}^c = x_{1,2}[1 + 2(0.5 - 0.1874)] = 1.0214 \times 1.6252 = 1.6601$$

Benzer mutasyonlar rassal sayılara bağlı olarak diğer bireylere uygulandığında;

$x_i^{(5),T}$	
0.0287	1.6601
0.0236	1.6453
-0.0770	1.6000
-0.6013	1.5013
0.0234	1.4979
0.4064	0.4590
-1.2823	0.6142
-0.4538	1.1860
-1.5078	0.0702
-0.0027	1.0215

nesli elde edilebilir. Müteakiben yapılacak işlem, amaç fonksiyon değerlerinin hesaplanması ve sonrasında seçim işlemlerine devam edilmesidir. Şekil 3.42 dahilinde nesiller boyunca elit bireylere ait amaç fonksiyonu değişimi ve süreç nihayetinde ulaşılan optimal konum görülmektedir.



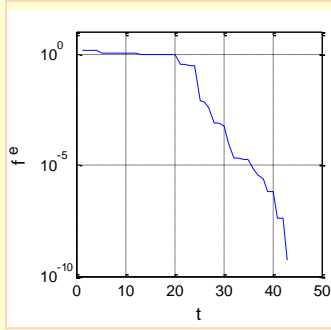


Sonuçlardan da gözlemlenebileceği üzere eniyileme süreci, amaç fonksiyon değerinin  $10^{-10}$  mertebesine ulaşması nedeniyle 43üncü nesilde sonlandırılmıştır. Buna göre optimal nokta koordinatları ve ulaşılan amaç fonksiyon değeri;

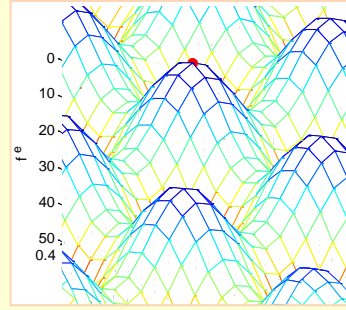
$$\mathbf{x}^* = 1 \times 10^{-5} [0.1548 \quad 0.0685]$$

$$f^* = 5.6838 \times 10^{-10}$$

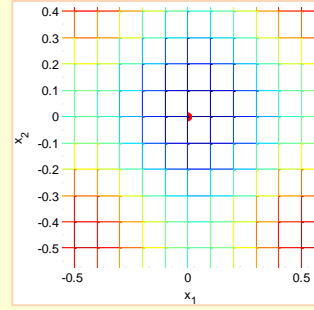
şeklinde elde edilmiş olup özellikle titreşimli mutasyon adımlarının eniyileme sürecinde etkin rol aldığı görülmektedir.



(a) Amaç fonksiyonu değışimi.



(b) Amaç fonksiyonu yüzeyi dahilinde optimal noktanın (kırmızı nokta) konumu (ters perspektif ve üstten görünüş).



**Şekil 3.42** Rastrigin fonksiyonuna ait (a) minimizasyon süreci ve (b) amaç fonksiyonu üzerindeki en uygun nokta görünüşleri.



Elimizde NACA 0012 kanat profilinin olduğunu, bu profilin belirli uçuş şartları altında optimize edilmek istendiğini düşünelim. Aerodinamik optimizasyon sürecinde, aerodinamik şekli kontrol eden ve tasarım parametreleri olarak ifade edilen tasarım yöneyinin tanımlanması gerekir.

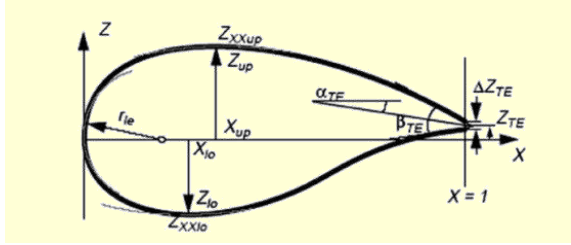


Örnek uygulama





Literatürde kanat profillerinin tanımlanması çerçevesinde değişik yöntemler söz konusu olup üzerinde çalışılan sorunda kanat profilinin Parsec yöntemi ile tanımlandığını düşünelim. Parsec yöntemi, kanat profil eğrisinin tanımlanması sadedinde Bezier eğrisinden farklı olarak fiziksel anlam içeren tasarım değişkenlerini içerir. Müteakip şekilde Parsec yöntemine göre bir kanat profilinin tasarım değişkenleri görülmektedir;



Şekil 3.43 Parsec tasarım değişkeni tanımlamaları<sup>(93)</sup>.

Buna göre herhangi bir kanat profili, toplamda 11 tasarım değişkeni ile temsil edilebilir. Bu değişkenler; hücum kenarı eğrilik yarıçapı ( $r$ ), üst yüzey eğrisine ait tepe noktanın koordinatları ile eğriliği ( $x_{up}, z_{up}, z_{xxup}$ ), alt yüzeye ait tepe noktanın koordinatları ile eğriliği ( $x_{lo}, z_{lo}, z_{xxlo}$ ), firar kenarının ordinatı ( $z_{TE}$ ), kalınlığı ( $\Delta z_{TE}$ ) ile firar kenarı alt ve üst yüzey bitiş doğrultularının açıl değerleri ( $\alpha_{TE}, \beta_{TE}$ ) şeklinde ifade edilebilir. Söz konusu tasarım değişkenleri, profil eğrilerinin elde edilmesinde doğrudan değil, dolaylı olarak kullanılırlar. Zira üst veya alt yüzeye ait herhangi bir profil eğrisi, polinomiyal bir yaklaşımla;

$$z = \sum_{i=1}^6 a_i x^{i-\frac{1}{2}} \quad (3.128)$$

şeklindeki denklemlerle tanımlanır. Burada  $a_i$  katsayıları, Parsec parametreleri ile dolaylı olarak ilişkilendirilen katsayılardır. Örneğin profil geometrisi üst yüzey eğrisi için;

$$r = \frac{1}{2} a_1^2 \quad (3.129)$$

$$z_{TE} = z(1) = \sum_{i=1}^6 a_i \quad (3.130)$$

$$z_{UP} = z(x_{UP}) = \sum_{i=1}^6 a_i x_{UP}^{i-\frac{1}{2}} \quad (3.131)$$



$$\tan(\alpha_{TE}) = z'(1) = \sum_{i=1}^6 \frac{(2n-1)}{2} a_i \quad (3.132)$$

$$z'(x_{UP}) = \sum_{i=1}^6 \frac{(2n-1)}{2} a_i x_{UP}^{n-\frac{3}{2}} \quad (3.133)$$

$$z_{XXUP} = z''(x_{UP}) = \sum_{i=1}^6 \frac{(2n-1)(2n-3)}{2} a_i x_{UP}^{n-\frac{5}{2}} \quad (3.134)$$

denklemleri yazılabilir. Bu tanımlamalardan hareketle profil üst eğrisi katsayılarını;

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ x_{UP}^{1/2} & x_{UP}^{3/2} & x_{UP}^{5/2} & x_{UP}^{7/2} & x_{UP}^{9/2} & x_{UP}^{11/2} \\ 1/2 & 3/2 & 5/2 & 7/2 & 9/2 & 11/2 \\ \frac{1}{2}x_{UP}^{-1/2} & \frac{3}{2}x_{UP}^{1/2} & \frac{5}{2}x_{UP}^{3/2} & \frac{7}{2}x_{UP}^{5/2} & \frac{9}{2}x_{UP}^{7/2} & \frac{11}{2}x_{UP}^{9/2} \\ \frac{1}{4}x_{UP}^{-3/2} & \frac{3}{4}x_{UP}^{-1/2} & \frac{15}{4}x_{UP}^{1/2} & \frac{35}{4}x_{UP}^{3/2} & \frac{63}{4}x_{UP}^{5/2} & \frac{99}{4}x_{UP}^{7/2} \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{bmatrix} = \begin{bmatrix} z_{TE} \\ z_{UP} \\ \tan(\alpha_{TE}) \\ \Delta z_{TE} \\ z_{XXUP} \\ \sqrt{2r} \end{bmatrix} \quad (3.135)$$

eşitliğinin çözümü ile elde etmek mümkündür. Benzer yaklaşım alt yüzey için de geçerlidir;

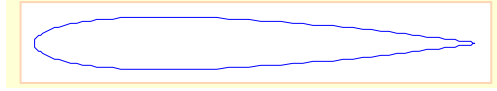
$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ x_{LO}^{1/2} & x_{LO}^{3/2} & x_{LO}^{5/2} & x_{LO}^{7/2} & x_{LO}^{9/2} & x_{LO}^{11/2} \\ 1/2 & 3/2 & 5/2 & 7/2 & 9/2 & 11/2 \\ \frac{1}{2}x_{LO}^{-1/2} & \frac{3}{2}x_{LO}^{1/2} & \frac{5}{2}x_{LO}^{3/2} & \frac{7}{2}x_{LO}^{5/2} & \frac{9}{2}x_{LO}^{7/2} & \frac{11}{2}x_{LO}^{9/2} \\ \frac{1}{4}x_{LO}^{-3/2} & \frac{3}{4}x_{LO}^{-1/2} & \frac{15}{4}x_{LO}^{1/2} & \frac{35}{4}x_{LO}^{3/2} & \frac{63}{4}x_{LO}^{5/2} & \frac{99}{4}x_{LO}^{7/2} \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_7 \\ a_8 \\ a_9 \\ a_{10} \\ a_{11} \\ a_{12} \end{bmatrix} = \begin{bmatrix} z_{TE} \\ z_{LO} \\ \tan(\beta_{TE}) \\ \Delta z_{TE} \\ z_{XXLO} \\ -\sqrt{2r} \end{bmatrix} \quad (3.136)$$

Kanat profilini oluşturan alt ve üst yüzey eğrilerinin, hücum kenarında aynı eğrilik yarıçapına sahip olması nedeniyle polinomiyal denklem katsayılarından  $a_1$  ila  $a_7$  terimleri aynı değeri ifade etmektedir. Sonuç olarak kanat profilini tanımlayan tasarım yöneyi değişkenlerini;

$$\mathbf{Z}^T = [z_{TE} \ z_{UP} \ \alpha_{TE} \ \Delta z_{TE} \ z_{XXUP} \ r \ x_{UP} \ z_{LO} \ \beta_{TE} \ z_{XXLO} \ x_{LO}] \quad (3.137)$$

biçiminde ifade etmek mümkündür. Üzerinde durulan optimizasyon sorununun ana gayesi, NACA 0012 kanat profilini temsil eden Parsec parametrelerinin bulunması şeklinde tanımlanabilir.





Şekil 3.44 NACA 0012 kanat profili.

NACA 0012 kanat profili, simetrik bir profil olup alt ve üst yüzey noktaları yatay eksene göre simetriklik arz etmektedir. Ayrıca firar kenarı kalınlığı kuramsal olarak 0 şeklinde ifade edilebilir. Müteakip çizelgede profile ait referans noktalar listelenmiştir<sup>(94)</sup>. Buna göre referans nokta tanımlaması, firar kenarı üst yüzeyinden başlamakta, hücum kenarına doğru devam etmekte ve hücum kenarından alt yüzey boyunca firar kenarına doğru sonlanmaktadır.

NACA 0012 profil noktaları															
Üst yüzey						Alt yüzey									
x	y	x	y	x	y	x	y	x	y	x	y	x	y	x	y
1.0000	0	0.8578	0.0196	0.5121	0.0522	0.1595	0.0544	0	0	0.1422	-0.0526	0.4879	-0.0537	0.8578	-0.0196
0.9994	0.0013	0.8405	0.0216	0.4879	0.0537	0.1422	0.0526	0.0006	-0.0043	0.1595	-0.0544	0.5121	-0.0522	0.8743	-0.0176
0.9977	0.0016	0.8224	0.0237	0.4638	0.0551	0.1257	0.0507	0.0023	-0.0084	0.1776	-0.0559	0.5362	-0.0505	0.8898	-0.0157
0.9948	0.0020	0.8036	0.0258	0.4397	0.0563	0.1102	0.0485	0.0052	-0.0125	0.1964	-0.0572	0.5603	-0.0488	0.9045	-0.0139
0.9907	0.0026	0.7840	0.0280	0.4158	0.0574	0.0955	0.0460	0.0093	-0.0165	0.2160	-0.0582	0.5842	-0.0469	0.9182	-0.0122
0.9855	0.0033	0.7638	0.0302	0.3921	0.0583	0.0818	0.0434	0.0145	-0.0203	0.2362	-0.0590	0.6079	-0.0450	0.9310	-0.0105
0.9791	0.0042	0.7430	0.0323	0.3686	0.0590	0.0690	0.0406	0.0209	-0.0241	0.2570	-0.0596	0.6314	-0.0430	0.9427	-0.0090
0.9717	0.0052	0.7216	0.0345	0.3455	0.0596	0.0573	0.0376	0.0283	-0.0277	0.2784	-0.0599	0.6545	-0.0409	0.9534	-0.0076
0.9631	0.0063	0.6997	0.0367	0.3227	0.0599	0.0466	0.0345	0.0369	-0.0312	0.3003	-0.0600	0.6773	-0.0388	0.9631	-0.0063
0.9534	0.0076	0.6773	0.0388	0.3003	0.0600	0.0369	0.0312	0.0466	-0.0345	0.3227	-0.0599	0.6997	-0.0367	0.9717	-0.0052
0.9427	0.0090	0.6545	0.0409	0.2784	0.0599	0.0283	0.0277	0.0573	-0.0376	0.3455	-0.0596	0.7216	-0.0345	0.9791	-0.0042
0.9310	0.0105	0.6314	0.0430	0.2570	0.0596	0.0209	0.0241	0.0690	-0.0406	0.3686	-0.0590	0.7430	-0.0323	0.9855	-0.0033
0.9182	0.0122	0.6079	0.0450	0.2362	0.0590	0.0145	0.0203	0.0818	-0.0434	0.3921	-0.0583	0.7638	-0.0302	0.9907	-0.0026
0.9045	0.0139	0.5842	0.0469	0.2160	0.0582	0.0093	0.0165	0.0955	-0.0460	0.4158	-0.0574	0.7840	-0.0280	0.9948	-0.0020
0.8898	0.0157	0.5603	0.0488	0.1964	0.0572	0.0052	0.0125	0.1102	-0.0485	0.4397	-0.0563	0.8036	-0.0258	0.9977	-0.0016
0.8743	0.0176	0.5362	0.0505	0.1776	0.0559	0.0023	0.0084	0.1257	-0.0507	0.4638	-0.0551	0.8224	-0.0237	0.9994	-0.0013
						0.0006	0.0043					0.8405	-0.0216	1.0000	0

Parsec parametrelerine bağlı eğri uydurma sorunu olan söz konusu problem için amaç fonksiyonunu;

$$\max_{\mathbf{z}} \frac{1}{f(\mathbf{z})} \quad (3.138)$$

$$f(\mathbf{z}) = \sum_{j=1}^m (z_{r,j} - z_j)^2 \quad (3.139)$$

şeklinde tanımlamak mümkündür.



Diğer taraftan ortak parametreler barındırorsa da, üst yüzey eğrisi ile alt yüzey eğrilerinin birbirinden bağımsız olması nedeniyle sorun iki ayrı alt eniyileme süreci olarak ele alınabilir. Dolayısıyla üst ve alt yüzey eğrileri için iki ayrı amaç fonksiyonu;

$$\max_z \frac{1}{f(z^u)}, \quad f^u(z) = \sum_{j=1}^m (z_{r,j}^u - z_j^u)^2 \quad (3.140)$$

$$\max_z \frac{1}{f(z^a)}, \quad f^a(z) = \sum_{j=1}^m (z_{r,j}^a - z_j^a)^2 \quad (3.141)$$

biçiminde tanımlanabilir. Burada  $m$  nokta sayısını,  $z_{r,j}^u$  üst yüzeye ait  $j$ 'nci referans nokta değerini,  $z_j^u$  üst yüzeye ait bireyin  $j$ 'nci nokta değerini,  $z_{r,j}^a$  alt yüzeye ait  $j$ 'nci referans nokta değerini,  $z_j^a$  alt yüzeye ait bireyin  $j$ 'nci nokta değerini temsil etmektedir. Benzer yaklaşım tasarım yöney tanımlamaları için de geçerlidir;

$$\mathbf{Z}^{u,T} = [z_{TE} \quad z_{UP} \quad \alpha_{TE} \quad \Delta z_{TE} \quad z_{XXUP} \quad r \quad x_{UP}] \quad (3.142)$$

$$\mathbf{Z}^{a,T} = [z_{TE} \quad z_{LO} \quad \beta_{TE} \quad \Delta z_{TE} \quad z_{XXLO} \quad r \quad x_{LO}] \quad (3.143)$$

Eğri uydurma probleminin GA yöntemi ile çözülmesi çerçevesinde yapılacak ilk işlem, sürece ait kontrol parametrelerinin belirlenmesidir. Bu meyanda nüfus büyüklüğü  $N_p = 10$ , azami iterasyon sayısı  $T = 100$ , elit birey sayısı  $N_e = 1$ , seçim yöntemi uygunluk değeri, çaprazlama yöntemi BLX- $\alpha$  ve mutasyon yöntemi olarak da titreşimli mutasyon yaklaşımı dikkate alınabilir. Yakınsama ölçütünün tanımlanması meyanda azami tekrar sayısı dikkate alınabilir. Diğer taraftan başlangıç nüfusunun üretilmesi çerçevesinde müteakip ifadelere göre hareket edilebilir;

$$z_{TE} = 0 \quad (3.144)$$

$$z_{UP} = 0.08 + rand \times 0.1 \quad (3.145)$$

$$\alpha_{TE} = -3 - rand \times 10 \quad (3.146)$$

$$\Delta z_{TE} = 0 \quad (3.147)$$

$$z_{XXUP} = -0.3 \times rand \quad (3.148)$$

$$r = 0.002 + rand \times 0.028 \quad (3.149)$$

$$x_{UP} = 0.3 + rand \times 0.4 \quad (3.150)$$

$$z_{LO} = -0.08 - rand \times 0.1 \quad (3.151)$$

$$\beta_{TE} = 3 + rand \times 10 \quad (3.152)$$

$$z_{XXLO} = 0.3 \times rand \quad (3.153)$$

$$x_{LO} = 0.3 + rand \times 0.4 \quad (3.154)$$

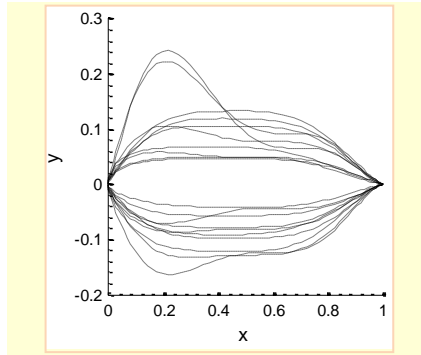
Söz konusu ifadeler, simetriye ve tecrübeye dayalı olup belli aralıklarda rastsallık içermektedir. Buna göre olası başlangıç nüfusu;





$Z_i^{u,T}$						$Z_i^{a,T}$							
0	0.0499	-3.8511	0	-0.1446	0.0108	0.4686	0	-0.1282	8.3226	0	0.0366	0.0108	0.5237
0	0.0458	-8.8401	0	-0.0369	0.0090	0.4793	0	-0.0778	6.3305	0	0.1149	0.0090	0.5027
0	0.1056	-12.1409	0	-0.2392	0.0086	0.3591	0	-0.0437	9.8889	0	0.2990	0.0086	0.6617
0	0.0679	-5.5807	0	-0.1352	0.0290	0.3887	0	-0.1245	8.4725	0	0.1725	0.0290	0.5891
0	0.0771	-11.2256	0	-0.1357	0.0171	0.6049	0	-0.0921	10.1578	0	0.2559	0.0171	0.4210
0	0.0918	-11.0040	0	-0.2451	0.0244	0.6646	0	-0.1224	6.7913	0	0.2761	0.0244	0.4863
0	0.1328	-8.2981	0	-0.2531	0.0045	0.4887	0	-0.0979	11.9619	0	0.1430	0.0045	0.5052
0	0.1192	-12.0594	0	-0.1623	0.0034	0.4210	0	-0.0427	5.6903	0	0.1811	0.0034	0.5109
0	0.0490	-7.3732	0	-0.1493	0.0183	0.5721	0	-0.0813	7.6341	0	0.1138	0.0183	0.5384
0	0.0656	-4.6610	0	-0.1707	0.0061	0.6911	0	-0.0567	10.0110	0	0.0831	0.0061	0.5119

şeklinde elde edilebilir. Genotipi ifade edilen söz konusu bireylerin fenotipi, müteakip şekilde görülmektedir.



Şekil 3.45 Başlangıç nüfusu görünümü.

Oluşturulan nüfus dâhilindeki bireylerin amaç fonksiyon değerleri ve sıralaması ile uygunluk değerleri;

$1/f_i^u$	Sıra	$U_i^u$	$1/f_i^a$	Sıra	$U_i^a$
554.3251	9	4	292.4735	3	4
473.4315	1	3	271.5745	10	3
274.2964	2	2	109.8430	8	1
171.9128	4	1	50.9983	2	1
18.9429	5	0	29.9462	9	0
12.2608	3	0	26.0049	5	0
8.4114	8	0	16.7518	7	0
5.8856	7	0	7.3670	6	0
2.4354	6	0	5.3868	1	0
2.2782	10	0	3.6588	4	0



şeklinde hesaplanabilir Burada örnek olması açısından, üst yüzey popülasyonundaki ilk bireye ait uygunluk değerinin;

$$U_1^u = \text{round} \left\{ \frac{1/f_1^u}{\sum_{j=1}^{10} 1/f_j^u} 10 \right\} = \text{round} \left\{ \frac{554.3251}{1524.2} 10 \right\} = 4$$

yaklaşımı ile elde edildiği kolayca gösterilebilir. Amaç fonksiyon değeri açısından en iyi olan birey, 9uncu birey olup söz konusu bireyin 4 defa çaprazlamada dikkate alınması mümkündür. Bu arada çaprazlama öncesi elit bireyler ve bu bireylere ait amaç fonksiyon değerleri kayıt altına alınarak bir sonraki nesle aynen aktarılmak üzere muhafaza edilir;

$$\mathbf{z}^{el,u,T}(1) = [0 \quad 0.0490 \quad -7.3732 \quad 0 \quad -0.1493 \quad 0.0183 \quad 0.5721],$$

$$\frac{1}{f^{el,u}(1)} = 554.3251$$

$$\mathbf{z}^{el,a,T}(1) = [0 \quad -0.0437 \quad 9.8889 \quad 0 \quad 0.2990 \quad 0.0086 \quad 0.6617],$$

$$\frac{1}{f^{el,a}(1)} = 292.4735$$

Elitizm uygulaması sonrasında çaprazlama işlemlerine geçilir. Buna göre çaprazlamada ilk eş olarak sıralamadaki ilk birey dikkate alınır. Buna mukabil diğer eş, rassal olarak seçilir. Örneğin üst yüzey popülasyonundaki 9uncu birey dikkate alınarak;

$$\mathbf{z}_e^u = \mathbf{z}_9^u \rightarrow [0 \quad 0.0490 \quad -7.3732 \quad 0 \quad -0.1493 \quad 0.0183 \quad 0.5721]$$

$$\mathbf{z}_d^u = \mathbf{z}_{[rand \times 10]}^u$$

$$rand \rightarrow 0.3867$$

$$\mathbf{z}_d^u = \mathbf{z}_4^u \rightarrow [0 \quad 0.0679 \quad -5.5807 \quad 0 \quad -0.1352 \quad 0.0290 \quad 0.3887]$$

eşleşmesi yapılabilir. Eş seçimi sonrasında BLX- $\alpha$  çaprazlama yöntemine göre çaprazlamaya geçilir. Bu meydana etkileşim bandını belirleyen  $\gamma = 2.6$  değeri dikkate alınarak örnek mahiyetteki yeni nesle ait ikinci genin değeri için müteakip işlemler gerçekleştirilir;

$$\alpha = 2.6 \times rand - 0.5 \rightarrow 0.1162$$

$$x_2^{c1} = 0.1162z_{e,2}^u + (1 - 0.1162)z_{d,2}^u = 0.1162 \times (0.0679) + 0.8838 \times (0.0490) = 0.0512$$

$$z_2^{c2} = 0.1162z_{d,2}^u + (1 - 0.1162)z_{e,2}^u = 0.1162 \times (0.0490) + 0.8838 \times (0.0679) = 0.0657$$

Benzer işlemler her iki yüzeye ait diğer gen ve bireyler için aynen gerçekleştirilerek olası yeni nesil elde edilebilir;



$\mathbf{z}_i^{u,1,T}$						$\mathbf{z}_i^{a,1,T}$							
0	0.0512	-3.7769	0	-0.1271	0.0154	0.6401	0	-0.1455	9.5340	0	0.0769	0.0127	0.4466
0	0.0657	-9.1770	0	-0.1575	0.0319	0.3207	0	-0.0264	8.6776	0	0.2587	0.0067	0.7388
0	0.0371	-11.070	0	-0.1657	0.0283	0.7184	0	-0.1074	10.1710	0	0.1226	0.0101	0.4376
0	0.1037	-7.3063	0	-0.2287	0.0144	0.5183	0	-0.0645	8.0405	0	0.2129	0.0093	0.7478
0	0.0645	-11.251	0	-0.2100	0.0165	0.5792	0	-0.0514	9.8417	0	0.2652	0.0004	0.7085
0	0.1774	-8.3081	0	-0.1665	0.0094	0.4716	0	-0.1111	8.6555	0	0.1880	0.0452	0.6911
0	-0.0092	-11.124	0	-0.1451	0.0123	0.5215	0	-0.0571	9.7059	0	0.2835	0.0213	0.5598
0	0.0438	-13.181	0	-0.1137	0.0086	0.4904	0	-0.0970	7.6193	0	0.1007	0.0052	0.5071
0	0.0519	0.4902	0	-0.0678	0.0112	0.4574	0	-0.0409	10.0258	0	0.0962	0.0191	0.5432

Bu arada her çaprazlama sonrasında uygunluk değeri güncelleme işlemleri unutulmamalıdır. Örneğin ilk çaprazlama neticesinde ilk sıradaki uygunluk değeri güncellenerek ( $U_1^u = U_1^u - 1 = 4 - 1 = 3$ ), ikinci çaprazlamaya geçilir. Çaprazlama süreci sonrasında mutasyon işlemleri uygulanır. Seçilen mutasyon yönteminin titreşimli mutasyon yaklaşımı olması nedeniyle operatör;

$$x_{i,j} \Big|_{j=1,2}^{i=1,2,\dots,9} = \begin{cases} x_{i,j}[1 + 2(0.5 - u)] & t \bmod 4 = 0 \\ x_{i,j} & t \bmod 4 \neq 0 \end{cases}$$

şeklinde tanımlanabilir. Buna göre  $t = 1$  olması nedeniyle herhangi bir mutasyon işlemi yapılmadan bir sonraki nesil işlemlerine geçilebilir. Bu bağlamda yapılacak işlem, amaç fonksiyon değerlerinin hesaplanması ve akabinde seçim işlemlerine devam edilmesidir. Benzer işlemler yapılarak 4üncü nesle gelindiğinde aşağıdaki elit ve bireylerin eldesi olasıdır;

$$\mathbf{z}^{el,u,T}(4) = [0 \quad 0.0519 \quad 0.4902 \quad 0 \quad -0.0678 \quad 0.0124 \quad 0.4574] \quad \frac{1}{f^{el,u}(4)} = 819.09$$

$$\mathbf{z}^{el,a,T}(4) = [0 \quad -0.0586 \quad 9.7437 \quad 0 \quad 0.3414 \quad 0.0144 \quad 0.4232] \quad \frac{1}{f^{el,a}(4)} = 929.87,$$

$\mathbf{z}_i^{u,4,T}$						$\mathbf{z}_i^{a,4,T}$							
0	0.1544	-0.0132	0	-0.0670	0.0136	0.4935	0	0.0014	9.9860	0	0.3283	0.0073	0.3833
0	0.0193	0.4284	0	-0.0682	0.0126	0.4424	0	-0.0864	9.7079	0	0.4451	0.0143	0.6113
0	0.0561	0.4449	0	-0.1483	0.0115	0.3827	0	-0.0580	11.0079	0	0.3430	0.0105	0.4538
0	0.0583	2.5066	0	-0.0378	0.0124	0.4650	0	-0.0548	9.2244	0	0.3426	0.0155	0.5129
0	0.0519	0.4902	0	-0.0678	0.0124	0.4574	0	-0.0586	9.7437	0	0.3414	0.0144	0.4232
0	0.0523	-1.1133	0	0.0910	0.0126	0.1633	0	-0.0557	9.6936	0	0.3417	0.0113	0.3519
0	0.0533	-4.8965	0	-0.0935	0.0115	0.5726	0	-0.0603	5.2198	0	0.3741	0.0140	0.4404
0	0.0637	5.5803	0	-0.1229	0.0107	0.2780	0	-0.0766	11.9349	0	0.3695	0.0142	0.5506
0	0.0635	2.5211	0	-0.0707	0.0112	0.3739	0	-0.0586	9.7437	0	0.3414	0.0144	0.4232
0	0.0511	2.2777	0	-0.1077	0.0124	0.3882	0	-0.0586	9.7437	0	0.3414	0.0144	0.4232



Bu adımda  $t \bmod 4 = 0$  olması nedeniyle mutasyon işlemleri uygulanır. Mutasyon, tüm bireylerin tüm genlerine uygulanmakta olup sadece elit birey (burada 5inci birey) süreçten hariç tutulur. Örneğin üst yüzeye ait ilk bireyin ilk genine mutasyon işlemi;

$$u = 0.5045$$

$$x_{1,1}^c = x_{1,1}[1 + 2(0.5 - 0.5045)] = 0.1544 \times 0.991 = 0.1530$$

şeklinde uygulanabilir Benzer mutasyonlar diğer bireylere uygulandığında;

$Z_i^{u,5,T}$						$Z_i^{a,5,T}$							
0	0.1530	-0.0131	0	-0.0681	0.0136	0.5004	0	0.0014	10.0634	0	0.3316	0.0075	0.3815
0	0.0190	0.4175	0	-0.0685	0.0128	0.4510	0	-0.0857	9.7927	0	0.4417	0.0143	0.6169
0	0.0551	0.4358	0	-0.1468	0.0116	0.3761	0	-0.0577	11.0190	0	0.3381	0.0104	0.4565
0	0.0588	2.5210	0	-0.0367	0.0126	0.4666	0	-0.0559	9.1595	0	0.3354	0.0158	0.5100
0	0.0519	0.4902	0	-0.0678	0.0124	0.4574	0	-0.0586	9.7437	0	0.3414	0.0144	0.4232
0	0.0528	-1.1301	0	0.0933	0.0124	0.1616	0	-0.0562	9.6212	0	0.3435	0.0111	0.3461
0	0.0523	-5.0309	0	-0.0927	0.0115	0.5602	0	-0.0601	5.3231	0	0.3785	0.0140	0.4445
0	0.0649	5.5576	0	-0.1208	0.0104	0.2829	0	-0.0749	12.0889	0	0.3715	0.0143	0.5408
0	0.0648	2.4891	0	-0.0712	0.0109	0.3806	0	-0.0585	9.6399	0	0.3370	0.0143	0.4229
0	0.0500	2.2175	0	-0.1093	0.0121	0.3919	0	-0.0583	9.8849	0	0.3420	0.0144	0.4322

olması nesli elde edilmiş olur. Müteakip şekilde nesiller boyunca elit bireylere ait amaç fonksiyonu değişimi ve süreç nihayetinde ulaşılan optimal konum görülmektedir. Buna göre optimal nokta koordinatları ve ulaşılan amaç fonksiyon değerleri;

$$z^{*,u,T}(100) = [0 \quad 0.06 \quad -9.6933 \quad 0 \quad -0.4510 \quad 0.0147 \quad 0.3018]$$

$$1/f^{el,u}(100) = 1.2924 \times 10^5$$

$$z^{*,a,T}(100) = [0 \quad -0.06 \quad 9.3927 \quad 0 \quad 0.4411 \quad 0.0149 \quad 0.3039]$$

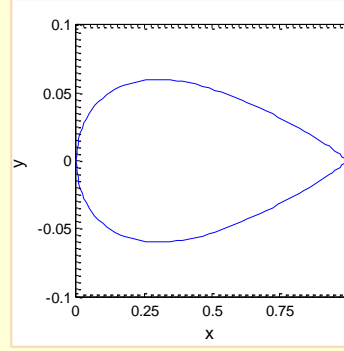
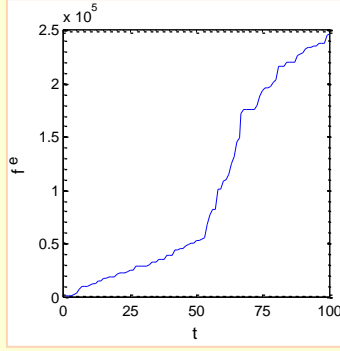
$$1/f^{*,a}(100) = 1.1808 \times 10^5$$

şeklinde elde edilmiş olup özellikle titreşimli mutasyon adımlarının, 50nci nesil sonrasında eniyileme sürecinde etkin rol aldığı görülmektedir.

Diğer taraftan profilin simetrik olması nedeniyle, daha iyi sonuç veren eğri tanımlamasının her iki yüzey için de dikkate alınması mümkün olup burada daha genel olan yaklaşımlar önceliklendirilmiştir.







(a) Amaç fonksiyonu değişimi (alt ve üst yüzey amaç fonksiyonları toplamı). (b) Eniyileme süreci sonrasında elde edilen kanat profil geometrisi.

Şekil 3.46 Kanat profili eğri uydurma sürecine ait (a) eniyileme süreci ve (b) geometrik sonucu.

2B ortamında Parsec yaklaşımı ile profil geometrisi üretimi.

MATLAB uygulaması

```

clc; clear; rand('state',sum(100*clock));
pn = 20 ; % nüfus büyüklüğü
mg = 100 ; % azami iterasyon sayısı
fq = 10 ; % mutasyon frekansı
w = 1; beta = 1 ;
alfa= 0.8 ; % çaprazlama katsayısı
% referans eğri
[x y] = textread('n0012.dat') ;
refxlp = x(66:131)';refzlp = y(66:131)' ;
for i=1:66
    refxup(i) = x(67-i);
    refzup(i) = y(67-i);
end
xup = refxup; xlo = refxlp;
% başlangıç nüfusu
r = 0.002+rand(1,pn)*0.028;
Zte = zeros(1,pn);
alphaTE = -3-rand(1,pn)*10;
betaTE = 3+rand(1,pn)*10;

```



```

deltaZte = zeros(1,pn);
Xup = 0.3+rand(1,pn)*0.4;
Zup = 0.04+rand(1,pn)*0.1;
Zxxup = -0.3*rand(1,pn);
Xlo = 0.3+rand(1,pn)*0.4;
Zlo = -0.04-rand(1,pn)*0.1;
Zxxlo = 0.3*rand(1,pn);

% Eniyileme süreci başlangıcı
for g=1:1:mg
    rand('state',sum(100*clock));
    % eğri noktaları
    for i=1:pn
        UC(1,:) = ones(1,6);
        UC(2,:) = Xup(i).^(0.5:1:5.5);
        UC(3,:) = (1:2:11)./2;
        UC(4,:) = Xup(i).^(-0.5:1:4.5).*(0.5:1:5.5);
        UC(5,:) = Xup(i).^(-1.5:1:3.5).*[-0.25 0.75 3.75 8.75 15.75 24.75];
        UC(6,:) = zeros(1,6);UC(6,1) = 1;
        LC(1,:) = ones(1,6);
        LC(2,:) = Xlo(i).^(0.5:1:5.5);
        LC(3,:) = (1:2:11)./2;
        LC(4,:) = Xlo(i).^(-0.5:1:4.5).*(0.5:1:5.5);
        LC(5,:) = Xlo(i).^(-1.5:1:3.5).*[-0.25 0.75 3.75 8.75 15.75 24.75];
        LC(6,:) = zeros(1,6);LC(6,1) = 1;
        UV=[Zte(i) Zup(i) tand(alphaTE(i)) deltaZte(i) Zxxup(i) sqrt(2*r(i))]' ;
        LV=[Zte(i) Zlo(i) tand(betaTE(i)) deltaZte(i) Zxxlo(i) -sqrt(2*r(i))]' ;
        UPC(i,:) = (inv(UC)*UV)';
        LPC(i,:) = (inv(LC)*LV)';
        clear('UC','LC','UV','LV')
    end
    for i=1:pn
        zup(i,:) = (UPC(i,1)*xup.^0.5+UPC(i,2)*xup.^1.5+UPC(i,3)*...
            xup.^2.5+UPC(i,4)*xup.^3.5+UPC(i,5)*xup.^4.5+UPC(i,6)*xup.^5.5);
        zlo(i,:) = (LPC(i,1)*xlo.^0.5+LPC(i,2)*xlo.^1.5+LPC(i,3)*...
            xlo.^2.5+LPC(i,4)*xlo.^3.5+LPC(i,5)*xlo.^4.5+LPC(i,6)*xlo.^5.5);
    end
    % Amaç fonksiyonu hesaplaması
    ffu = (zup - repmat(refzup,pn,1)).^2;
    ffl = (zlo - repmat(refzlp,pn,1)).^2;
    Ffu=sum(ffu,2);
    Ffl=sum(ffl,2);
    % sıralama işlemleri
    orderu=1:1:pn;
    orderl=orderu;
    for i=1:1:pn-1
        for j=i:1:pn
            if Ffu(i)>Ffu(j)
                d1=Ffu(i);d2=orderu(i);
                Ffu(i)=Ffu(j);orderu(i)=orderu(j);
            end
        end
    end

```





```
Ffu(j)=d1;orderu(j)=d2;
end
if Ffl(i)>Ffl(j)
    d1=Ffl(i);d2=orderl(i);
    Ffl(i)=Ffl(j);orderl(i)=orderl(j);
    Ffl(j)=d1;orderl(j)=d2;
end
end
end
Ffu=1./Ffu;
Ffl=1./Ffl;
if g==1
    FuO = sum(Ffu)/pn;
    FLO = sum(Ffl)/pn;
End
% uygunluk deęeri hesabı
totalffu=sum(Ffu);
totalffl=sum(Ffl);
fru=round((Ffu/totalffu)*pn);
frl=round((Ffl/totalffl)*pn);
% elitizm uygulaması
elitffu(g)=Ffu(1);
elitffl(g)=Ffl(1);
elitUP(g,:) = [Zte(orderu(1)) Zup(orderu(1)) alphaTE(orderu(1))...
    deltaZte(orderu(1)) Zxxup(orderu(1)) r(orderu(1))...
    Xup(orderu(1))];
elitLO(g,:) = [Zte(orderl(1)) Zlo(orderl(1)) betaTE(orderl(1))...
    deltaZte(orderl(1)) Zxxlo(orderl(1)) r(orderl(1))...
    Xlo(orderl(1))];
elitzup(g,:)=zup(orderu(1),:);
elitzlo(g,:)=zlo(orderl(1),:);
if g>1
    if elitffu(g)<elitffu(g-1)
        elitffu(g)=elitffu(g-1);
        elitUP(g,:)=elitUP(g-1,:);
        elitzup(g,:)=elitzup(g-1,:);
    end
    if elitffl(g)<elitffl(g-1)
        elitffl(g)=elitffl(g-1);
        elitLO(g,:)=elitLO(g-1,:);
        elitzlo(g,:)=elitzlo(g-1,:);
    end
end
end
% sonuçların görselleştirilmesi
gen=1:1:g;
figure(1)
cla;
plot(xup,elitzup(g,:), 'b',xlo,elitzlo(g,:), 'b')
figure(2)
```





```
if g>1
    plot([g-1 g],[elitffu(g-1)+elitffl(g-1) elitffu(g)+elitffl(g)]);
    xlim([0 mg])
    hold on
    figure(2);title(['elit fitness value = ',num2str(elitffu(g)+elitffl(g))]);
end
% Çaprazlama işlemleri
PU = [Zte' Zup' alphaTE' deltaZte' Zxxup' r' Xup'];
PL = [Zte' Zlo' betaTE' deltaZte' Zxxlo' r' Xlo'];
s=1;
for i=1:1:pn
    if fru(i)>0
        for k=1:1:fru(i)
            if s>=pn
                break
            end
            fru(i)=fru(i)-1;
            wife=ceil(rand*pn);
            if wife==orderu(i)
                wife=ceil(rand*pn);
            end
            for j=1:7
                gama=(1+2*alfa)*rand-0.5;
                nPU(s,j)=(1-gama)*PU(orderu(i),j)+gama*PU(wife,j);
                nPU(s+1,j)=gama*PU(orderu(i),j)+(1-gama)*PU(wife,j);
                if j==6
                    if nPU(s,j)<0
                        u=rand;
                        nPU(s,j)=PU(orderu(i),j)*u+(1-u)*PU(wife,j);
                    end
                    if nPU(s+1,j)<0
                        u=rand;
                        nPU(s+1,j)=PU(orderu(i),j)*u+(1-u)*PU(wife,j);
                    end
                end
            end
            end
            s=s+2;
        end
    end
    if s>=pn
        break
    end
end
s=1;
for i=1:1:pn
    if frl(i)>0
        for k=1:1:frl(i)
            if s>=pn
                break
            end
        end
    end
end
```



```

frl(i)=frl(i)-1;
wife=ceil(rand*pn);
if wife==orderl(i)
    wife=ceil(rand*pn);
end
for j=1:7
    gama=(1+2*alfa)*rand-0.5;
    nPL(s,j)=(1-gama)*PL(orderl(i),j)+gama*PL(wife,j);
    nPL(s+1,j)=gama*PL(orderl(i),j)+(1-gama)*PL(wife,j);

    if j==6
        if nPL(s,j)<0
            u=rand;
            nPL(s,j)=PL(orderl(i),j)*u+(1-u)*PL(wife,j);
        end
        if nPL(s+1,j)<0
            u=rand;
            nPL(s+1,j)=PL(orderl(i),j)*u+(1-u)*PL(wife,j);;
        end
    end
end

    end
    s=s+2;
end
end
if s>=pn
    break
end

end
PU = nPU;
PL = nPL;
% mutasyon işlemleri
if rem(g,fq)==0
    for i=1:7
        for j=1:1:pn
            PU(j,i)=PU(j,i)*(1+w*beta*(0.5-rand));
            PL(j,i)=PL(j,i)*(1+w*beta)*(0.5-rand);
        end
    end
end

Zte = (PU(:,1))'; Zup = (PU(:,2))'; alphaTE = (PU(:,3))'; deltaZte =
(PU(:,4))'; Zxxup = (PU(:,5))';
r = (PU(:,6))'; Xup = (PU(:,7))'; Zlo = (PL(:,2))'; betaTE = (PL(:,3))'; Zxxlo =
(PL(:,5))';
Xlo = (PL(:,7))';
% elit birey yerleşimi
Zte(5) = elitUP(g,1); Zup(5) = elitUP(g,2); alphaTE(5) = elitUP(g,3); ...
deltaZte(5) = elitUP(g,4); Zxxup(5) = elitUP(g,5); r(5) = elitUP(g,6); ...

```

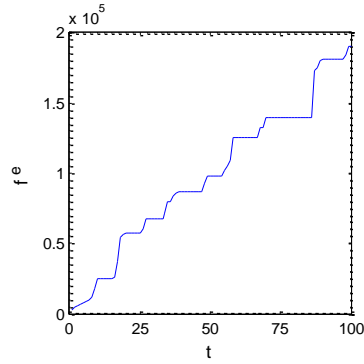




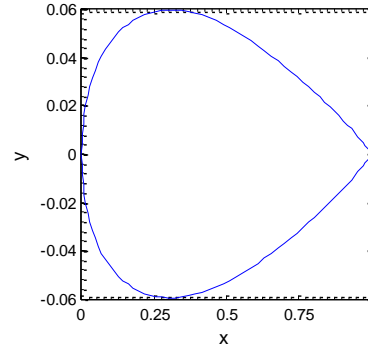
```
Xup(5) = elitUP(g,7);  
Zte(5) = elitLO(g,1);Zlo(5) = elitLO(g,2); betaTE(5) = elitLO(g,3);...  
deltaZte(5) = elitLO(g,4);Zxxlo(5) = elitLO(g,5);r(5) = elitLO(g,6);...  
Xlo(5) = elitLO(g,7);  
end
```



### Sonuç



(a)



(b)

Şekil 3.47 (a) Kanat profili eğri uydurma süreci, (b) eniyileme süreci geometrik sonucu.



## İrfan Damlası



ünlerden birgün köylerden birinde adamın birinin eşiği kuyunun birine düşmüş. Niye düşer, nasıl düşer diye sormayın, eşek bu, düşmüş işte... Belki kör bir kuyuydu, ağzı tahtayla kapatılmıştı belki, üzerine de toprak dökülmüştü. Zamanla tahta çürüdü, zayıfladı, toprakta biten otları yemek isteyen eşiğin ağırlığını çekemedi ve kırıldı...

Kuyuya düşen hayvancık saatlerce acı içinde kıvrandı, bağırdı kendince. Sesini duyan sahibi gelip baktı ki vaziyet kötü. Zavallı karakaçanı kuyunun dibinde melul mahsun bakınıyor, üstelik yaralanmış. Karşılaştığı bu durumda kendini eşiği kadar zavallı hisseden adamcağız köylüleri yardıma çağırdı. Ne yaparsak, ne etsek nasıl çıkarsak soruları havada kaldı. Sonunda karar verildi ki kurtarmak için çabalamaya değmez. Bu umutsuz durumun tek çaresi kuyuyu toprakla örtmek...

Köylüler ellerine aldıkları kürekle etraftan kuyunun içine toprak attılar. Zavallı hayvan üzerine gelen toprakları her seferinde silkinerek dibe döktü. Ayaklarının altına aldığı toprak sayesinde her adımda biraz daha yükseldi, yükseldi, yükseldi... Sonunda yukarıya kadar çıkmış oldu. Köylüler ağzı açık baka kaldı.

Hayat bazen bizim de üzerimizde kötü etkiler bırakır. Toz toprakla örtmeye çalışanlar çok olur. Bunlarla baş etmenin tek yolu yakınıp sızlanmak değil, düşünüp silkinmek ve kurtulmak, aydınlığa adım atmaktır... Kör kuyuda olsak bile.





- [1] Alıntı <<http://curioustester.blogspot.com.tr/2015/07/heuristic-evaluation-whats-that.html>> [01 Nisan 2016].
- [2] Alıntı <<https://www.pinterest.com/pin/357895501614554639/>> [01 Nisan 2016].
- [3] Heuristic, Merriam-Webster Dictionary, Alıntı < [www.merriam-webster.com/dictionary/heuristic](http://www.merriam-webster.com/dictionary/heuristic)> [11 Nisan 2016].
- [4] G. Polya, How to solve it: a new aspect of mathematical method, 2<sup>nd</sup> Edition, Princeton University, 1945, Alıntı < [https://notendur.hi.is/hei2/teaching/Polya\\_HowToSolveIt.pdf](https://notendur.hi.is/hei2/teaching/Polya_HowToSolveIt.pdf)> [11 Nisan 2016].
- [5] M. H. J. Romanycia and F. J. Pelletier, What is heuristic?, Comput. Intell. Vol. 1, 47-58 (1985). Alıntı < [www.sfu.ca/~jeffpell/papers/RomanyciaPelletierHeuristics85.pdf](http://www.sfu.ca/~jeffpell/papers/RomanyciaPelletierHeuristics85.pdf)> [11 Nisan 2016].
- [6] T. S. Costa, What are the differences between heuristics and meta-heuristics?, ResearchGate, Alıntı < [https://www.researchgate.net/post/What\\_are\\_the\\_differences\\_between\\_heuristics\\_and\\_metaheuristics](https://www.researchgate.net/post/What_are_the_differences_between_heuristics_and_metaheuristics)> [11 Nisan 2016].
- [7] W. Herbawi, What is the major difference between heuristic, hyper heuristic and meta heuristics? And their relationship with global optimization methods?, ResearchGate, Alıntı < [https://www.researchgate.net/post/What\\_is\\_the\\_major\\_difference\\_between\\_heuristic\\_hyper\\_heuristic\\_and\\_meta\\_heuristics\\_and\\_their\\_relationship\\_with\\_global\\_optimization\\_methods](https://www.researchgate.net/post/What_is_the_major_difference_between_heuristic_hyper_heuristic_and_meta_heuristics_and_their_relationship_with_global_optimization_methods)> [11 Nisan 2016].
- [8] Hyper-heuristic, Wikipedia: The free encyclopedia, Alıntı < <https://en.wikipedia.org/wiki/Hyper-heuristic>> [11 Nisan 2016].
- [9] J. Denzinger, M. Fuchs, and M. Fuchs, High performance ATP systems by combining several AI methods, Alıntı <[citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.75.9733&rep=rep1&type=pdf](http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.75.9733&rep=rep1&type=pdf)> [11 Nisan 2016].
- [10] Metaheuristic, Wikipedia: The free encyclopedia, Alıntı< [https://en.wikipedia.org/wiki/Metaheuristic#Hybridization\\_and\\_memetic\\_algorithms](https://en.wikipedia.org/wiki/Metaheuristic#Hybridization_and_memetic_algorithms)> [11 Nisan 2016].
- [11] El-Ghazali Talbi, Metaheuristics: from design to implementation, John Wiley & Sons Inc., 2009, sf. 128-129.
- [12] A. E. Nix and M. D. Vose, Modeling genetic algorithms with Markov chains, Annals of mathematics and artificial intelligence 5 (1992) 79-88.
- [13] F. Pan, X-L Li, Q. Zhou, W-X Li and Q. Gao, Analysis of standard particle swarm optimization algorithm based on Markov chain, Acta Automatica Sinica, Vol. 39, No:4, 2013.
- [14] W. Spendley, G.R. Hext ve F.R. Himsforth, “Sequential application of simplex designs in optimization and evolutionary operation”, Technometrics 4 (1962), sf. 441-461.
- [15] J.A. Nelder ve R. Mead, “A simplex method for function minimization”, Computer Journal 7: 308-313, 1965.
- [16] J.J. Tomick, On convergence of the Nelder-Mead simplex algorithm for unconstrained stochastic optimization. Alıntı: < [www.dtic.mil/dtic/tr/fulltext/u2/a289453.pdf](http://www.dtic.mil/dtic/tr/fulltext/u2/a289453.pdf)> [17 Aralık 2015].





- [17] Nelder-Mead method, Wikipedia, the free encyclopedia. Alıntı: < [https://en.wikipedia.org/wiki/Nelder-Mead\\_method](https://en.wikipedia.org/wiki/Nelder-Mead_method)> [17 Aralık 2015].
- [18] J.S. Arora, Introduction to optimum design, 3<sup>rd</sup> Edition, Elsevier Inc., 2012, sf. 474.
- [19] R.M. Lewis, V. Torczon ve M.W. Trosset, Direct search methods: then and now, Journal of Computational and Applied Mathematics, Vol. 124, Issue 1-2, December 2000, sf. 191-207.
- [20] Tabu, The Free Dictionary, Alıntı < [www.thefreedictionary.com/tabu](http://www.thefreedictionary.com/tabu)> [22 Nisan 2016].
- [21] F. Glover, Heuristics for integer programming using surrogate constraints, Decision Sciences 8, 156-166. Alıntı < [onlinelibrary.wiley.com](http://onlinelibrary.wiley.com)> [22 Nisan 2016].
- [22] L. Li, H. Liu ve R. Lu, Tabu search, Alıntı < [https://www.mii.lt/zilinskas/uploads/Heuristic\\_Algorithms/Lectures/Lect6/TabuSearch3.ppt](https://www.mii.lt/zilinskas/uploads/Heuristic_Algorithms/Lectures/Lect6/TabuSearch3.ppt)> [23 Mart 2017].
- [23] Tabu search, Wikipedia: The free encyclopedia, Alıntı < [https://en.wikipedia.org/wiki/Tabu\\_search](https://en.wikipedia.org/wiki/Tabu_search)> [22 Nisan 2016].
- [24] İ. Çayiroğlu, İleri algoritma analizi – tabu arama algoritması, Alıntı < [www.ibrahimcayiroglu.com](http://www.ibrahimcayiroglu.com)> [22 Nisan 2016].
- [25] F. Glover, Tabu search, Alıntı < [www.uv.es/rmarti/paper/docs/ts2.pdf](http://www.uv.es/rmarti/paper/docs/ts2.pdf)> [22 Nisan 2016].
- [26] CITIES City Distance Datasets: h30\_main.txt, Alıntı < <https://people.sc.fsu.edu/~jburkardt/datasets/cities/cities.html>> [28 Nisan 2016].
- [27] S. Jayaswal, A comparative study of tabu search and simulated annealing for traveling salesman problem, Alıntı < [www.eng.uwaterloo.ca/~sjayaswa/projects/MSCI703\\_project.pdf](http://www.eng.uwaterloo.ca/~sjayaswa/projects/MSCI703_project.pdf)> [28 Nisan 2016].
- [28] Markov chain, Wikipedia: The free encyclopedia, Alıntı < [https://en.wikipedia.org/wiki/Markov\\_chain](https://en.wikipedia.org/wiki/Markov_chain)> [12 Nisan 2016].
- [29] Markov Chains, Alıntı <[https://www.dartmouth.edu/~chance/teaching\\_aids/books\\_articles/probability\\_book/Chapter11.pdf](https://www.dartmouth.edu/~chance/teaching_aids/books_articles/probability_book/Chapter11.pdf)>[12 Nisan 2016].
- [30] V. Powell, Markov Chains: explained visually, Alıntı < [setosa.io/ev/markov-chains/](http://setosa.io/ev/markov-chains/)> [12 Nisan 2016].
- [31] Random walk, Wikipedia: The free encyclopedia, Alıntı < [https://en.wikipedia.org/wiki/Random\\_walk](https://en.wikipedia.org/wiki/Random_walk)> [02 Mayıs 2016].
- [32] N. T. Baranasuriya, G. D. de Silva, R. Singhal, K. Yadati ve Z. Zhou, Random walks and Markov chains, Alıntı < [www.comp.nus.edu.sg/~rahul/allfiles/cs6234-13-random.pptx](http://www.comp.nus.edu.sg/~rahul/allfiles/cs6234-13-random.pptx)> [02 Mayıs 2016].
- [33] S. Arora, Random walks, Markov chains, and how to analysis them, Alıntı <<https://www.cs.princeton.edu/courses/archive/fall13/cos521/lecnotes/lec12.pdf>> [03 Mayıs 2016].
- [34] Alıntı<<http://lcp.elis.ugent.be/sites/lcp.elis.ugent.be/files/brownianmotion.jpg>> [03 Mayıs 2016].
- [35] N. Gupta, Brownian motion, File exchange: Matlab Central, Alıntı< [http://www.mathworks.com/matlabcentral/fileexchange/32067-brownian-motion/content/Brownian\\_motion.m](http://www.mathworks.com/matlabcentral/fileexchange/32067-brownian-motion/content/Brownian_motion.m)> [04 Mayıs 2016].
- [36] S. E. Alm, Simple random walk, Alıntı<[www2.math.uu.se/~sea/kurser/stokprocmn1/slumpvandring\\_eng.pdf](http://www2.math.uu.se/~sea/kurser/stokprocmn1/slumpvandring_eng.pdf)> [03 Mayıs 2016].



- [37] S. Vempala, Geometric random walks: A survey, Combinatorial and Computational Geometry, MSRI Publications, Vol. 52, pp. 573-612, 2005.
- [38] Random walk method, Alıntı <[https://people.rit.edu/pnveme/EMEM820n/Mod6\\_Numerical/Mod6\\_content/Mod6\\_sec2\\_randon\\_walk.html](https://people.rit.edu/pnveme/EMEM820n/Mod6_Numerical/Mod6_content/Mod6_sec2_randon_walk.html)> [04 Mayıs 2016].
- [39] Simulated annealing, Wikipedia: The free encyclopedia, Alıntı < [https://en.wikipedia.org/wiki/Simulated\\_annealing](https://en.wikipedia.org/wiki/Simulated_annealing)> [09 Mayıs 2016].
- [40] Equations of State Calculations by Fast Computing Machines, Wikipedia: The free encyclopedia, Alıntı < [https://en.wikipedia.org/wiki/Equations\\_of\\_State\\_Calculations\\_by\\_Fast\\_Computing\\_Machines](https://en.wikipedia.org/wiki/Equations_of_State_Calculations_by_Fast_Computing_Machines)> [09 Mayıs 2016].
- [41] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller ve E. Teller, Equation of state calculations by fast computing machines, J. Chem. Phys. 21, 1087 (1953), Alıntı < [bayes.wustl.edu/Manual/EquationOfState.pdf](http://bayes.wustl.edu/Manual/EquationOfState.pdf)> [09 Mayıs 2016].
- [42] S. Kirkpatrick, C. D. Gelatt ve M. P. Vecchi, Optimization by simulated annealing, Science, Vol. 220, No 4598, 1983.
- [43] P. Eles, Simulated annealing, Alıntı < [www.ida.liu.se/~zebpe83/heuristic/lectures/SA\\_lecture.pdf](http://www.ida.liu.se/~zebpe83/heuristic/lectures/SA_lecture.pdf)> [13 Mayıs 2016].
- [44] M. A. Akçayol, Zeki optimizasyon teknikleri, Alıntı < [w3.gazi.edu.tr/~akcayol/files/ZOL3Tavlama.pdf](http://w3.gazi.edu.tr/~akcayol/files/ZOL3Tavlama.pdf)> [13 Mayıs 2016].
- [45] O. de Weck, Simulated annealing: a basic introduction, Alıntı < [ocw.mit.edu/courses/engineering-systems-division.pdf](http://ocw.mit.edu/courses/engineering-systems-division.pdf)> [16 Mayıs 2016].
- [46] Simulated annealing tutorial, Alıntı < [apmonitor.com/me575/index.php/Main/SimulatedAnnealing](http://apmonitor.com/me575/index.php/Main/SimulatedAnnealing)> [17 Mayıs 2016].
- [47] Ant colony optimization algorithms, Wikipedia: The free encyclopedia, Alıntı < [https://en.wikipedia.org/wiki/Ant\\_colony\\_optimization\\_algorithms](https://en.wikipedia.org/wiki/Ant_colony_optimization_algorithms)> [02 Haziran 2016].
- [48] Stigmergy, Wikipedia: The free encyclopedia, Alıntı < <https://en.wikipedia.org/wiki/Stigmergy>> [02 Haziran 2016].
- [49] M. Dorigo, V. Maniezzo ve A. Coloni, "Ant system: optimization by a colony of cooperating agents," IEEE Transactions on systems, man, and cybernetics-Part B: Cybernetics, Vol. 26, No. 1, pp. 29-41, 1996.
- [50] Ant, Wikipedia: The free encyclopedia, Alıntı < <https://en.wikipedia.org/wiki/Ant#Communication>> [02 Haziran 2016].
- [51] A. Shekhavant, P. Poddar ve D. Boswal, Ant colony optimization algorithms: introduction and beyond, Alıntı < [mat.uab.cat/~alseda/MasterOpt/ACO\\_Intro.pdf](http://mat.uab.cat/~alseda/MasterOpt/ACO_Intro.pdf)> [03 Haziran 2016].
- [52] C. Blum, Ant colony optimization, Alıntı < <https://www.lri.fr/~hansen/proceedings/2011/GECCO/companion/p963.pdf>> [03 Haziran 2016].
- [53] İ. Aydın, Karınca koloni algoritması, Alıntı < [web.firat.edu.tr/iaydin/bmu579/bmu\\_579\\_bolum6.pdf](http://web.firat.edu.tr/iaydin/bmu579/bmu_579_bolum6.pdf)> [03 Haziran 2016].



- [54] İ. Çayıroğlu, İleri Algoritma Analizi: Karınca Koloni Algoritması, Alıntı < [www.ibrahimcayiroglu.com/dokumanlar/ilerialgoritmaanalizi/ilerialgoritmaanalizi-6.hafta-karincakoloni.pdf](http://www.ibrahimcayiroglu.com/dokumanlar/ilerialgoritmaanalizi/ilerialgoritmaanalizi-6.hafta-karincakoloni.pdf)> [06 Haziran 2016].
- [55] M. Herrmann, Natural computing: Ant colony optimization, Alıntı < [www.inf.ed.ac.uk/teaching/courses/nat/slides/lecture10\\_ACO.pdf](http://www.inf.ed.ac.uk/teaching/courses/nat/slides/lecture10_ACO.pdf)> [06 Haziran 2016].
- [56] M. Dorigo ve G. D. Caro, Ant algorithms for discrete optimization, Alıntı < [people.idsia.ch/~luca/ij\\_23-alife99.pdf](http://people.idsia.ch/~luca/ij_23-alife99.pdf)> [06 Haziran 2016].
- [57] B. Santosa, Tutorial on ant colony optimization, Alıntı < <https://bsantosa.files.wordpress.com/2015/03/aco-tutorial-english.pdf>> [06 Haziran 2016].
- [58] K. Socha ve M. Dorigo, Ant colony optimization for continuous domains, European Journal of Operational Research 185 (2008) 1155-1173.
- [59] Ant colony optimization for continuous domains (ACOR), by S. M. K. Heris, File exchange: Matlab Central, Alıntı< <http://www.mathworks.com/matlabcentral/fileexchange/52860-ant-colony-optimization-for-continuous-domains--acor->> [08 Haziran 2016].
- [60] Bee, Wikipedia: The free encyclopedia, Alıntı < <https://en.wikipedia.org/wiki/Bee>> [13 Haziran 2016].
- [61] M. Özdemir, Zaman kısıtı altında takım oryantiring problemlerinin yapay arı kolonisi yaklaşımı ile çözümü, Y. Lisans Tezi, İstanbul Üniversitesi Sosyal Bilimler Enstitüsü, İstanbul, 2013.
- [62] Why bees dance for their honey, Alıntı < [http://www.creationtips.com/img/bee\\_dance.jpg](http://www.creationtips.com/img/bee_dance.jpg)> [14 Haziran 2016].
- [63] Movement theatre in the beehive, Alıntı < [http://fokusz.info/Image/62.\\_szam/Risza.jpg](http://fokusz.info/Image/62._szam/Risza.jpg)> [14 Haziran 2016].
- [64] D. Karaboğa, An idea based on honey bee swarm for numerical optimization, Technical Report – TR06, Erciyes Üniversitesi, 2005.
- [65] D. Karaboga and B. Akay, A survey: algorithms simulating bee swarm intelligence, Artif Intell Rev (2009) 31:61-85.
- [66] D. Karaboga, B. Gorkemli, C. Ozturk and N. Karaboga, A comprehensive survey: artificial bee colony (ABC) algorithm and applications, Artif Intell Rev (2014) 42:21-57.
- [67] Artificial Bee Colony (ABC) in Matlab, by S. M. K. Heris, File exchange: Matlab Central, Alıntı< <http://www.mathworks.com/matlabcentral/fileexchange/52966-artificial-bee-colony--abc--in-matlab>> [15 Haziran 2016].
- [68] J. Kennedy and R. Eberhart, Particle swarm optimization, Proceedings of IEEE International Conference on Neural Networks, 1995, pp. 1942-1948.
- [69] O. Humphreys, The coordinated beauty of thousands of birds moving in sync, Alıntı < <http://gencept.com/the-coordinated-beauty-of-thousands-of-birds-moving-in-sync>> [24 Haziran 2016].
- [70] Flock (birds), Wikipedia: The free encyclopedia, Alıntı < [https://en.wikipedia.org/wiki/Flock\\_\(birds\)](https://en.wikipedia.org/wiki/Flock_(birds))> [24 Haziran 2016].
- [71] C-O Erneholm, Simulation of the flocking behavior of birds with the boids algorithm, Bachelor's thesis in computer science, Royal Institute of Technology, Stockholm-Sweden, 2011.



- [72] A. Pinto, H. Alves, I. Domingues, L. Rocha, S. Cruz, The particle swarm optimization algorithm, Alıntı < <https://www.google.com.tr>> [24 Haziran 2016].
- [73] Shoaling and schooling, Wikipedia: The free encyclopedia, Alıntı < [https://en.wikipedia.org/wiki/Shoaling\\_and\\_schooling](https://en.wikipedia.org/wiki/Shoaling_and_schooling)> [24 Haziran 2016].
- [74] J-B Leem, W. Jeon, C-H Yun, and S-H Lee, Quantitative analysis of fish schooling behavior with different numbers of medaka (*oryzias latipes*) and goldfish (*carassius auratus*), Ocean Science Journal, Vol. 47, Issue 4, pp. 445-451, 2012.
- [75] M. Dorigo, M. A. Montes de Oca, and A. Engelbrecht, Particle swarm optimization, Scholarpedia, 3(11):1486, Alıntı < [www.scholarpedia.org/article/Particle\\_swarm\\_optimization](http://www.scholarpedia.org/article/Particle_swarm_optimization)> [24 Haziran 2016].
- [76] Particle swarm optimization, Alıntı < [mnemstudio.org/particle-swarm-introduction.htm](http://mnemstudio.org/particle-swarm-introduction.htm)> [27 Haziran 2016].
- [77] S. Talukder, Mathematical modeling and applications of particle swarm optimization, Master's thesis, No: 2010:8, Blekinge Institute of Technology, Sweden, 2011.
- [78] I. C. Trelea, The particle swarm optimization algorithm: convergence analysis and parameter selection, Information Processing Letters 85 (2003) 317-325.
- [79] D. P. Rini, S. M. Shamsuddin, and S. S. Yuhani, Particle swarm optimization: technique, system, and challenges, International journal of Computer Applications, Vol. 14, No 1, pp. 19-27, 2011.
- [80] J. McCall, Genetic algorithms for modeling and optimization, Journal of Computational and Applied Mathematics, Vol. 184, Issue 1, pp. 205-222.
- [81] Evolutionary computation, Wikipedia: The free encyclopedia, Alıntı < [https://en.wikipedia.org/wiki/Evolutionary\\_computation](https://en.wikipedia.org/wiki/Evolutionary_computation)> [21 Temmuz 2016].
- [82] A. M. Turing, Computing machinery and intelligence, Mind: A Quarterly Review of Psychology and Philosophy, Vol. LIX, No. 236, pp. 433-460, 1950.
- [83] Genetic algorithm, Wikipedia: The free encyclopedia, Alıntı <[https://en.wikipedia.org/wiki/Genetic\\_algorithm](https://en.wikipedia.org/wiki/Genetic_algorithm)> [21 Temmuz 2016].
- [84] R. C. Chakraborty, Fundamentals of Genetic Algorithms: AI Course Lecture 39-40, Alıntı < [www.myreaders.info/09\\_Genetic\\_Algorithms.pdf](http://www.myreaders.info/09_Genetic_Algorithms.pdf)> [21 Temmuz 2016].
- [85] Evolutionary algorithm, Wikipedia: The free encyclopedia, Alıntı <[https://en.wikipedia.org/wiki/Evolutionary\\_algorithm](https://en.wikipedia.org/wiki/Evolutionary_algorithm)> [22 Temmuz 2016].
- [86] Genetics, Merriam-Webster Dictionary, Alıntı < [www.merriam-webster.com/dictionary/genetics](http://www.merriam-webster.com/dictionary/genetics)> [22 Temmuz 2016].
- [87] Gene, Wikipedia: The free encyclopedia, Alıntı <<https://en.wikipedia.org/wiki/Gene>> [22 Temmuz 2016].
- [88] M. Melanie, An introduction to genetic algorithms, A Bradford Book the MIT Press, 5<sup>th</sup> Printing, 1999.
- [89] Levins R., "Evolution in Changing Environments", Princeton University Press, Princeton, 1968, sf. 6-7.
- [90] J. H. Holland, "Adaptation in Natural and Artificial Systems", the University of Michigan Press, 1975, sf. 3-4.



- [91] J. H. Holland, adı geçen eser (age), sf. 143-144.
- [92] R. Cheng, M. Gen, “Genetic Algorithms and Engineering Optimization”, John Wiley and Sons, Inc., 2000, sf. 8.
- [93] N. P. Salunke, J. Ahamad, S.A. Channiwala, Airfoil Parameterization Techniques: A Review, Alıntı < <http://pubs.sciepub.com/ajme/2/4/1/image/fig2.png> > [27 Temmuz 2016].
- [94] E. A. von Doenhoff, H. I. Abbott, “Theory of Wing Sections”, Dover Publications, Inc., 1959, sf. 111-115.





## VEKİL MODEL YÖNTEMLERİ

“ Bazen mutlu ve mesutmuşsun gibi davran, zamanla âdetin saadetin olmaya, sonra da tabiatın olmaya başlar...”

Anonim <sup>(2)</sup>

	Sayfa
4.1 Giriş	552
4.2 Vekil Model yaklaşımı	554
4.3 Regresyon Yöntemleri	561
4.4 Mesafe Ağırlıklı Yaklaşım	579
4.5 Kriging Yöntemi	586
4.6 Destek Yöney Yöntemi	619
4.7 Yapay Sinir Ağları	652
4.8 Bulanık Mantık	673
Tarihten bir Sayfa	706
Kaynakça	707

(1)



4



## 4.1 Giriş



Optimizasyon yöntemleri dâhilinde şimdiye kadar ifade edilen deterministik ve sezgisel yöntemlerin kendilerine göre olumlu ve olumsuz yönleri söz konusudur. Özellikle gradyan esaslı yöntemlerin en ciddi sorunu, yerel çözümlere takılıp kalma ve bazı hallerde de gradyan bilgisinin hesaplanmasında yaşanan zorluklardır. Buna mukabil sezgisel yöntemlerde karşılaşılan genel sorunsal ise çok fazla sayıda amaç fonksiyonu hesaplamalarına ihtiyaç duyulmasıdır. Söz konusu sorun, nüfus bazlı yöntemlerde daha belirgin bir hal alır. Amaç ve bazı hallerde de kısıt fonksiyonlarının çok fazla sayıda değerlendirilmesinin bir sorun olarak görülmesinin altında yatan ana neden, hesaplama için gerekli devasa zaman gereksinimleridir. Mühendislik tasarım süreçlerinin pek çoğunda deneysel ve benzeşim temelli eniyileme süreçleri dikkate alınır. Meselenin daha somut bir şekilde açıklanması meyanında örnek bir mühendislik tasarımı, örneğin aerodinamik tasarım süreci ele alınabilir. Genel manada aerodinamik tasarımın deneysel süreçleri, rüzgâr tüneli deneylerini ve uçuş testlerini kapsar. Simülasyon bazlı eniyileme süreçleri, deneysel çalışmalardan çok daha önce gerçekleştirilir. Havacılık disipliniinde hesaplamalı akışkanlar dinamiği (HAD) olarak adlandırılan simülasyon çalışmaları sayesinde tasarım uzayının pek çok çözüm adayı denenebilir ve aerodinamik tasarımın ilk aşaması böylelikle tamamlanmış olur<sup>(3)</sup>. Ancak simülasyon çalışmalarında dikkate alınması gereken en önemli çekince, hesaplama süresidir. Esasen gerçek mühendislik problemlerini içeren basit simülasyon yazılımları dahi bazen dakikalar, saatler ve hatta günler alabilir<sup>(4)</sup>. Diğer taraftan evrimsel bazlı eniyileme süreçlerinde binlerce aday tasarımın değerlendirilmesi gerekir<sup>(5)</sup>. Dolayısıyla tasarım aşamasına ait simülasyon çalışmalarının bile aylar aldığı ürün geliştirme projeleri söz konusu olabilir. Buna mukabil eniyileme süreçlerinin daha hızlı sonuç vermesi ve etkinliğinin artırılması amacıyla değişik çözüm yöntemlerinden bahsetmek de mümkündür. Bahse konu çözüm yöntemleri arasında paralel hesaplama, melezleme ve seviyelendirme<sup>ii</sup> yaklaşımları sayılabilir.

Paralel hesaplama yöntemleri, hesaplama yükünün parçalara bölünmesi ve bölünen bu işlemlerin ayrı ayrı donanım imkânları ile aynı anda yapılması düşüncesine dayanır. Yaklaşım, büyük oranda bilgi işlem teçhizatı ya da diğer bir ifade ile işlemci özelliklerine dayalı olup çözüm yönteminin paralel hesaplama için uygun olması da dikkate alınması gereken diğer bir husustur. Paralel hesaplama uygulamalarında iki ana yaklaşımdan bahsedilebilir. İlk yaklaşımda ayrı çözüm adaylarına ait amaç fonksiyon değerlerinin ayrı işlemcilerde aynı anda hesaplanması ve bu sayede

<sup>i</sup> Computational fluid dynamics (CFD)

<sup>ii</sup> Multi-processing, hybridization, and multi-level approach





ardı ardına yapıldığında ihtiyaç duyulan hesaplama süresinin kayda değer bir şekilde kısaltılmasıdır. Özellikle GA, PSO ve ABC gibi nüfus bazlı eniyileme algoritmaları, bahse konu paralel hesaplama yaklaşımı için oldukça uygundur. Zira aynı anda nesildeki tüm bireylerin amaç fonksiyonu değerleri birbirine paralel olarak hesaplanabilir ve nesiller boyunca bu işlemlere devam edilebilir. İkinci yaklaşımda, tekil amaç fonksiyonu hesaplaması işyükünün parçalara ayrılması ve ihtiyaç duyulan sürenin belirli oranda azaltılmasıdır. Örneğin HAD sorunsalında çözüm ağının alt bölümlere ayrılması ve ayrı bölgelerde ayrı hesaplama yapılarak sonuçların birleştirilmesi, bahse konu ikinci yaklaşım için iyi bir örnek niteliğindedir. Sorunun çözümü sadedinde, sahip olunan yazılımsal ve donanımsal imkânlarla göre paralel hesaplama yöntemlerinden aynı anda veya ayrı ayrı faydalanılabilir<sup>(6)</sup>.

Mezleme yöntemleri, farklı üstünlükleri olan optimizasyon yöntemlerinin veya yöntem adımlarının, aynı sorunun çözümünde kullanılması düşüncesine dayanır. Genellikle deterministik yöntemler ile sezgisel algoritmaların beraberce kullanılması yaygındır. Örneğin global tarama yeteneğine sahip GA yöntemi ile beraber yerel tarama açısından oldukça verimli olan gradyan esaslı bir yöntemin aynı eniyileme sürecinin aynı adımında kullanılması melez yöntem uygulaması açısından iyi bir örnektir. Başka bir örnek de GA operatörlerinden biri olan mutasyon uygulamalarının PSO yöntemi içerisinde kullanılmasıdır. Melez yöntem uygulamalarını, değişik eniyileme yaklaşımlarının farklı kombinasyonları ile çeşitlendirmek mümkündür<sup>(7)</sup>.

Seviyelendirme yaklaşımı, farklı seviyelerde tarama, tasarım yöneyi tanımlaması ve değerlendirme şeklindeki üç farklı yöntemi barındırır. Farklı seviyelerde tarama düşüncesinden kastedilen, eniyileme sürecinin farklı safhalarında farklı tarama yönteminin kullanılmasıdır. Örneğin optimizasyonun başlangıç aşamalarında GA veya PSO gibi global tarama özelliği yüksek olan bir algoritmanın tercih edilmesi, sürecin sonlarına doğru ise yerel tarama özelliği verimli olan gradyan esaslı bir yöntemin seçilmesi, farklı seviyelerde optimum noktanın taranması anlamına gelmektedir. Bahse konu yaklaşımın melez yöntemden farkı, aynı adımda değil farklı adımlarda farklı yöntem tercihlerinin söz konusu olmasıdır. Farklı seviyelerde tasarım yöneyi tanımlamalarında, süreç içinde tasarım yöneyi değişkenlerinin tanım ve sayılarında değişikliğe gidilebilir. Genellikle eniyileme sürecinin başlangıç aşamalarında az sayıda tasarım yöneyi değişkeni dikkate alınarak hızlı bir ilerleme kaydedilebilir. Ancak sürecin ilerleyen aşamalarında ilave tasarım değişkenleri soruna eklenerek daha hassas ve iyi çözüm seçeneklerine ulaşmak mümkündür. Söz konusu sorunun kısıtlı eniyileme sorunu olması halinde tasarım yöneyi seviyelendirmesinde ilave bir özellik daha dikkate alınabilir: kısıtlar. Buna göre başlangıç

---

<sup>1</sup> Multi-level search, parameterization, and evaluation





aşamalarında bazı kısıtların gözardı edilmesi ve az sayıda kısıtın dikkate alınması, sürecin ilerleyen aşamalarında ise diğer kısıtların da hesaba katılması, sürecin hızlandırılması açısından tercih edilebilir bir yaklaşımdır<sup>(8)</sup>.

Farklı seviyelerde değerlendirmeden kastedilen, amaç fonksiyonunun farklı seviyelerde değerlendirmesidir. Özellikle optimizasyon sürecinin başlangıç aşamalarında amaç fonksiyonu değerinin, yüksek çözünürlüklü<sup>i</sup> hesaplama algoritmalarından ziyade daha düşük çözünürlüklü, ancak hızlı sonuç veren hesaplama yöntemleri ile elde edilmesi, farklı seviyelerde değerlendirme anlamına gelmektedir. Örneğin aerodinamik optimizasyon sorununda, akış alanının panel yaklaşımı ile çözülmesi halinde amaç fonksiyonu değeri hesaplama problemi doğrusal denklem takımı çözümüne indirgenmiş olur. Ancak akış alanının Navier-Stokes denklemleri dikkate alınarak çözülmesi halinde, sorun oldukça karmaşık kısmi diferansiyel denklemlerin çözümüne dayandırılmış olur<sup>(9)</sup>.

Farklı seviyelerde değerlendirmeden kastedilen bir diğer yaygın görüş, amaç fonksiyonunun hesaplanmasında karmaşık ve fiziksel özelliklere dayalı matematiksel modellerin kullanılması yerine daha basit ve kısa sürede sonuç veren vekil modellerin<sup>ii</sup> kullanılmasıdır. Bununla beraber vekil model yaklaşımı, amaç fonksiyonu değeri hesaplaması ile sınırlı kalmayıp bazen kısıt veya gradyan değeri hesaplamalarında da kullanılabilir. Zira bazı sorunlarda fonksiyonlara ait gradyan verilerinin hesaplanması karmaşık ve zor olabilir. Dolayısıyla vekil model yaklaşımı sadece sezgisel yöntemlerde değil gradyan esaslı yöntemlerde de kullanılabilir<sup>(10)</sup>.

## 4.2 Vekil model yaklaşımı

Vekil modeller, deneysel veya sayısal verilere dayalı olarak kurgulanan analitik modellerdir. Literatürde vekil model kavramı yanında yanıt yüzey, yardımcı model, yaklaşık model, meta-model, benzetim modeli<sup>iii</sup> gibi değişik isimlerle de anılmaktadır<sup>(11)</sup>. Analitik model kurgulama sürecinde önemli olan, girdi ve çıktı verilerinin<sup>iv</sup> özellikleridir. Bu nedenle bazı çalışmalarda söz konusu kurgulama, davranışsal modelleme veya kara-kutu modellemesi<sup>v</sup> olarak adlandırılır. Vekil modelin kurgulanması için gerekli olan girdi ve çıktı verileri, deneysel ölçümlerle elde edilen veriler olabileceği gibi simülasyon bazlı yazılımlarla elde edilen veriler de olabilir. Deneysel

<sup>i</sup> High-fidelity solver

<sup>ii</sup> Surrogate model

<sup>iii</sup> Response surface model, auxiliary model, approximation model, meta-model, and emulator model

<sup>iv</sup> Input and output data

<sup>v</sup> Behavioral modeling or black-box modeling



verilerin söz konusu olduğu veri tabanlarında, girdi verisi değişkenleri ile çıktı verileri arasındaki fiziksel ilişkilerin bilinmesine ihtiyaç duyulmayabilir. Zira önemli olan, hangi girdi verilerinin hangi çıktı verileri ile sonuçlandırıldığıdır. Ancak simülasyon bazlı veri tabanlarında, girdi verilerinin karşılığının ancak simülasyon yazılımları ile elde edilmesi mümkün olduğundan girdi ve çıktı arasındaki ilişki mutlaka bilinmelidir. Bununla beraber simülasyon modelinin bilinmesi, vekil model kurgulamasına bir engel değildir. Zira vekil model kurgulamanın dayanağı sadece girdi ve çıktı verileri arasındaki fiziksel ilişkileri bilmemek değil, bilinse dahi fiziksel ilişkilere dayalı simülasyon yazılımlarının sonuçlarının çok uzun sürede elde edilme olasılığıdır. Daha önce de ifade edildiği üzere özellikle gerçek mühendislik problemlerinde hesaplama süreleri oldukça uzun süreçleri kapsayabilmektedir.

Vekil model kurgulama süreci temelde üç aşamadan oluşmaktadır. İlk aşama, girdi ve çıktı verilerinden oluşan veri havuzunun oluşturulması işlemlerini kapsar. Veri havuzu, herhangi bir sebeple elde edilmiş girdi ve çıktı verilerinden oluşabileceği gibi belirli bir sistematik dâhilinde oluşturulan girdi verileri ile bu verilerin karşılığında elde edilecek çıktı verilerinden de oluşabilir. İkinci aşamada, girdi ve çıktı verileri arasında göreceli olarak basit ve kısa sürede sonuç veren bir analitik modelin kurgulanması işlemi gerçekleştirilir. Son aşama ise oluşturulan vekil modelin veri havuzu dışındaki test amaçlı başka veri çiftleri ile denenmesi ve bu sayede model hassasiyetinin belirlenmesidir. Genel manada model hassasiyeti, veri havuzunun büyüklüğüne ve verilerin dağılımına bağlıdır. Eğer elde herhangi bir veri yoksa, doğal olarak veri havuzunun oluşturulması gerekir. Veri havuzunun oluşturulması sürecinde değişik örnekleme yöntemleri dikkate alınabilir. Daha önce örnekleme yöntemlerinden bahsedilmiş ve sistematik, rassal ve karma örnekleme yöntemleri anlatılarak örnekler verilmişti. Bu nedenle burada örnekleme yöntemleri üzerinde durulmayacaktır.

Literatürde var olan değişik vekil model kurgulama yöntemlerinden bahsedilebilir. Yaygın olarak dikkate alınan vekil modeller arasında polinomiyal karaktere sahip regresyon yöntemleri, Kriging yöntemi, destek yöney metodu, yapay sinir ağları ve bulanık mantık<sup>i</sup> yaklaşımı sayılabilir. Bahse konu yöntemler, tekil olarak kullanılabilir gibi birkaçının beraber olarak dikkate alındığı kombinasyonlar da düşünülebilir. Vekil model yöntemlerinin performansı, aynı girdi verilerine karşılık gerçek çıktı verileri ile modelin ürettiği çıktı verileri arasındaki uyuma bağlıdır. Bununla beraber müteakip paragraflarda ifade edilen ölçütler, vekil modellerin performans karşılaştırmalarında kullanılan kıstaslar olarak sıralanabilir;

---

<sup>i</sup> Regression, Kriging, support vector machine, artificial neural networks, and fuzzy logic.



**Hassasiyet:** gerçek çıktı verisi ile vekil modelin ürettiği çıktı verisi arasındaki farkı baz alan hassasiyet değerlendirmesini içerir.

**Gürbüzlük:** vekil modelin, sorunun boyutları ve tipi farklılaştıkça göstermiş olduğu tahminleme yeteneğini dikkate alan bir ölçüttür.

**Etkinlik:** vekil modelin kurgulanması için gerekli süreyi temel alan ölçüttür.

**Kavramsal basitlik:** vekil modelin kurgulanmasının ve uygulanmasının zorluğunu irdeleyen bir kıstastır.

Sıralanan bahse konu ölçütler arasında özellikle hassasiyet ve etkinlik ölçütlerinin yaygın olduğu söylenebilir<sup>(12)</sup>. Diğer taraftan sayısal anlamda  $\mathbf{x}_{t,i}, i = 1, 2, \dots, m$  test girdi verisi olmak üzere aşağıdaki performans verileri dikkate alınabilir;

$$R^2 = 1 - \frac{\sum_{i=1}^m (\mathbf{y}_{t,i} - \hat{\mathbf{y}}_{t,i})^2}{\sum_{i=1}^m (\bar{\mathbf{y}}_{t,i} - \mathbf{y}_{t,i})^2} \quad (4.1)$$

$$OMH = \frac{\sum_{i=1}^m (\hat{\mathbf{y}}_{t,i} - \mathbf{y}_{t,i})^2}{m} \quad (4.2)$$

$$MMH = \max\{|\mathbf{y}_{t,i} - \hat{\mathbf{y}}_{t,i}|\}, \quad i = 1, 2, \dots, m \quad (4.3)$$

Burada  $\mathbf{y}_{t,i}$   $i$ 'ninci test noktasına ait gerçek çıktı değerini,  $\hat{\mathbf{y}}_{t,i}$   $i$ 'ninci test noktasına ait vekil model çıktı değerini,  $\bar{\mathbf{y}}_{t,i}$  test noktaları ortalama değerini,  $OMH$  ortalama mutlak hata değerini ve  $MMH$  maksimum mutlak hata değerini temsil etmektedir<sup>(13)</sup>.

Literatürde karşılaşılan diğer hata değeri değerlendirme parametreleri;

$$\bar{e} = \frac{1}{m} \sum_{i=1}^m e_i, \quad e_i = \left\| \frac{\hat{\mathbf{y}}_{t,i} - \mathbf{y}_{t,i}}{\mathbf{y}_{t,i}} \right\| \quad (4.4)$$

$$\sigma_e = \sqrt{\frac{\sum_{i=1}^m [e_i]^2}{m}} \quad (4.5)$$

şeklinde tanımlanabilir<sup>(14)</sup>.



Burada  $\bar{\epsilon}$  ortalama göreceli hata değerini,  $\sigma_\epsilon$  ortalama hata kök değerini temsil etmektedir. Öte yandan gerçek veri ile vekil model verisi arasında;

$$\begin{aligned} \mathbf{y} &= f(\mathbf{x}) \\ \hat{\mathbf{y}} &= \hat{f}(\mathbf{x}) \end{aligned} \quad (4.6)$$

$$\begin{aligned} f(\mathbf{x}) &= \hat{f}(\mathbf{x}) + \Delta(\mathbf{x}) \\ \Delta(\mathbf{x}) &= \epsilon(\mathbf{x}) + \delta(\mathbf{x}) \end{aligned}$$

şeklinde doğrudan ilişki kurulabilir. Burada  $\Delta(\mathbf{x})$  toplam hata değerini,  $\epsilon(\mathbf{x})$  vekil modelin kurgu model olmasından kaynaklanan hata değerini,  $\delta(\mathbf{x})$  yuvarlama ve sayısal yaklaşım kaynaklı hata değerlerini göstermektedir.

Vekil modellerin eniyileme süreçlerinde kullanımı iki farklı şekilde olabilir. Bunlardan ilki, tamamen vekil modele dayanılarak eniyileme sürecinin icra edilmesi, ikincisi gerçek model ile vekil modelin beraber kullanılması, diğer bir ifade ile vekil modelin asıl optimizasyon sürecine destek unsuru veya ivmelendirici olarak katkı sağlaması şeklindedir. Hatırlanacağı üzere standart optimizasyon sorunu;

$$\begin{aligned} &\min_{\mathbf{x}} f(\mathbf{x}) \\ \text{Kısıt} & \quad g_j(\mathbf{x}), \quad j = 1, 2, \dots, k \\ & \quad \mathbf{x}^U \leq \mathbf{x} \leq \mathbf{x}^L \end{aligned} \quad (4.7)$$

şeklinde ifade edilebilmekteydi. Gerçek amaç ve kısıt fonksiyonları yerine sadece vekil modellerin kullanılması durumunda aynı eniyileme süreci;

$$\begin{aligned} &\min_{\mathbf{x}} \hat{f}(\mathbf{x}) \\ \text{Kısıt} & \quad \hat{g}_j(\mathbf{x}), \quad j = 1, 2, \dots, k \\ & \quad \mathbf{x}^U \leq \mathbf{x} \leq \mathbf{x}^L \end{aligned} \quad (4.8)$$

ifadesi ile tanımlanır. Burada  $\hat{f}(\mathbf{x})$  ve  $\hat{g}_j(\mathbf{x})$  ifadeleri vekil modelleri göstermektedir.

Gerçek amaç ve kısıt fonksiyonları ile beraber vekil modellerin destek unsuru olarak kullanılması durumunda eniyileme süreci;





$$\min_x f(x), \hat{f}(x)$$

(4.9)

Kısıt

$$g_j(x), \hat{g}_j(x), j = 1, 2, \dots, k$$

$$x^U \leq x \leq x^L$$

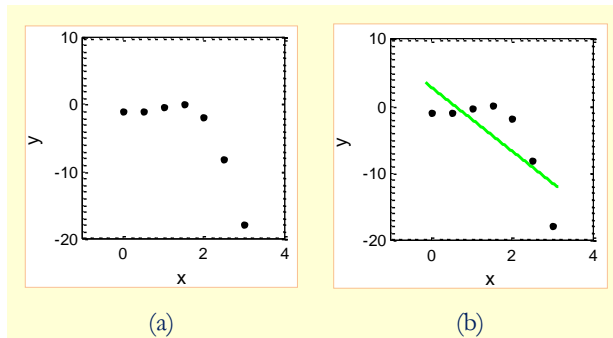
şeklindeki karma ifade ile betimlenebilir. Ancak burada unutulmamalıdır ki enyileme sürecini yöneten asıl unsur gerçek modellerdir.

Vekil model ile gerçek model kavramlarının daha iyi anlaşılabilmesi çerçevesinde müteakip örnek dikkate alınabilir. Buna göre elimizde yedi adet girdi ve çıktı verisinin var olduğunu ve bu verilerin;

$$x = [0 \quad 0.5 \quad 1 \quad 1.5 \quad 2 \quad 2.5 \quad 3]$$

$$y = [-1 \quad -0.9278 \quad -0.3597 \quad 0.02624 \quad -1.929 \quad -8.205 \quad -17.83]$$

şeklinde tanımlandığını düşünelim. Eldeki verilere göre kurgulanabilecek en basit analitik modelin, 1inci dereceden bir polinomial denklem olabileceği düşünüldüğünde Şekil 4.1'deki gibi bir modelleme yapılabilir;



Şekil 4.1 (a)Yedi nokta içeren veri havuzu ve (b) doğrusal vekil model yaklaşımı.

Şekilden de anlaşılacağı üzere doğrusal olmayan bir karaktere sahip veriler için yapılan doğrusal vekil model kurgulama işlemi neticesinde;

$$\hat{y} = -4.758x + 2.819$$

denklemini yazılabilir.

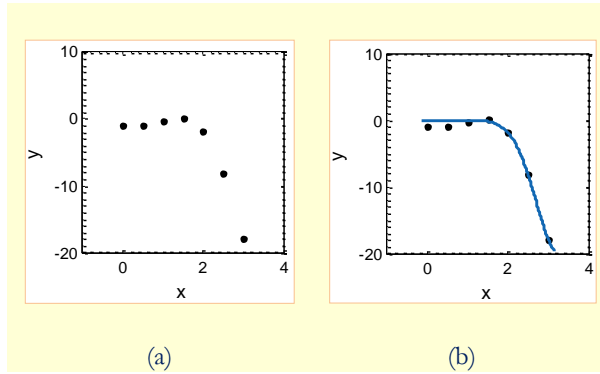




İki boyutlu böyle bir yaklaşım, literatürde eğri uydurma<sup>i</sup> yaklaşımı olarak tanımlanmaktadır. Boyut sayısının artması durumunda söz konusu yaklaşım, yanıt yüzey uydurma<sup>ii</sup> şeklinde isimlendirilir. Kurgulanan basit analitik modelin getirdiği en önemli yarar, veri havuzunda olmayan verilere karşılık belirli bir değer belirlenebilmesi olanağıdır. Örneğin veri havuzunda olmayan herhangi bir  $x = 5$  girdi değerinin karşılığı olarak vekil modelin yardımıyla  $\hat{y} = -20.971$  değeri kolayca elde edilebilir. Bununla beraber doğrusal model verileri ile gerçek verilerin karşılaştırılması yapıldığında arada ciddi farklılıkların olduğu gözlemlenebilir. Örneğin veri havuzundaki  $x = 1.5$  girdi değerinin karşılığı olarak gerçek model değerinin  $y = 0.02624$  olduğu, buna mukabil vekil modele göre yapılan değerlendirmede  $\hat{y} = -4.318$  değerinin belirlenebileceği görülür. Dolayısıyla toplam hata değerinin  $\Delta(\mathbf{x}) = 4.34$  gibi göreceli olarak ciddi bir mertebede olduğu söylenebilir. Gerçek veriler ile vekil model verileri arasındaki hata değerinin azaltılması amacıyla farklı ama daha karmaşık vekil model yaklaşımı yapılabilir. Örneğin veri havuzu kullanılarak Gauss fonksiyonu yaklaşımı yapıldığında;

$$\hat{y} = -20.19 \times e^{-\left(\frac{x-3.296}{0.8399}\right)^2}$$

eğrisini uydurmak mümkündür. Müteakip şekilde veri havuzu ve doğrusal olmayan karaktere sahip vekil model görülmektedir;



Şekil 4.2 (a) Veri havuzu ve (b) doğrusal olmayan vekil model yaklaşımı.

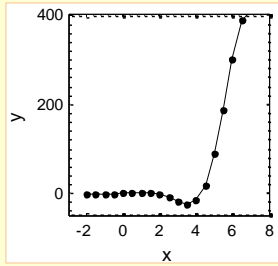
<sup>i</sup> Curve fitting

<sup>ii</sup> Response surface fitting

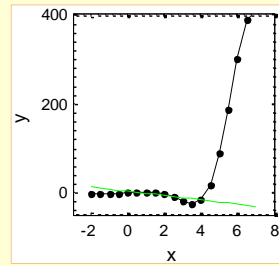




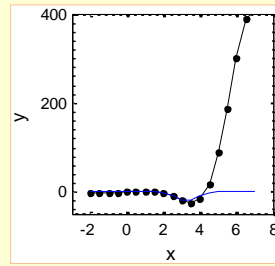
Bir önceki vekil model yaklaşımında olduğu gibi veri havuzunda olmayan herhangi bir girdi değerinin ve mesela  $x = 5$  değerinin karşılığı olarak yeni vekil modelin yardımıyla  $\hat{y} = -0.329$  değeri elde edilebilir. Yine benzer şekilde mevcut veriler ile vekil model veri değerleri arasındaki fark değerleri karşılaştırıldığında, aradaki farkların bir önceki modele göre çok daha az olduğu gözlemlenebilir. Örneğin veri havuzundaki  $x = 1.5$  girdi değerinin karşılığı olarak gerçek model değerinin  $y = 0.02624$  olduğu, buna mukabil vekil modele göre yapılan değerlendirmede ise  $\hat{y} = -0.2083$  değerinin belirlenebileceği görülür. Dolayısıyla toplam hata değerinin  $\Delta(x) = 0.234$  gibi daha önceki farka göre çok daha düşük mertebede olduğu söylenebilir. Hülusa, ikinci modelin daha iyi bir kurgulama olduğu sonucuna ulaşmak mümkündür. Ancak unutulmamalıdır ki ikinci modelin kurgulanması ilkinin göre daha zor ve işlem yükü de daha fazladır. Zira ilk vekil modelin kurgulanmasında sadece doğrusal denklem takımı çözümü yeterli iken ikinci vekil model yaklaşımında, model parametrelerinin eldesi amacıyla ayrı bir parametre eniyileme sürecinin uygulanması gerekir.



(a) Gerçek model değerleri.



(b) Gerçek ve doğrusal vekil model (yeşil) yaklaşımı.



(c) Gerçek ve Gauss vekil model (mavi) yaklaşımı.

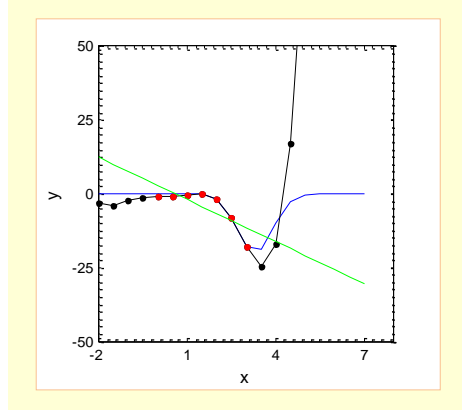
**Şekil 4.3** Veri havuzu ve dış bölgedeki gerçek ve vekil model davranışları.

Vekil model kurgulama sürecinde önem arz eden diğer bir husus, kurgulama esnasında dikkate alınan veri havuzunun dışına çıkıldığında modelin göstereceği davranıştır. Gürbzlük özelliğinin bir göstergesi olarak da değerlendirilebilecek söz konusu davranış çerçevesinde genellikle veri havuzundan çıkışla beraber vekil modellerin hata değerleri artar. Şekil 4.3'de veri havuzu ve dış bölgedeki gerçek ve vekil model davranışları görülmektedir. Örneğin veri havuzu dışındaki  $x = 5$  değerinin karşılığı olarak ilk vekil modelin değeri  $\hat{y} = -20.971$  şeklindeydi. Aynı girdi değerine karşılık ikinci vekil modelin ürettiği değer ise  $\hat{y} = -0.329$  biçimindeydi. Buna karşılık aynı girdi değerinin gerçek çıktı değeri ise  $y = 86.96$  olarak verilmektedir. Buna göre veri havuzu dışı bölgede





her iki vekil modelin de modelleme açısından anlamsız olduğu gözlemlenmektedir. Şekil 4.4'te her iki vekil ve gerçek model değerleri, görüntüsel olarak yaklaştırılmış bir bakış açısıyla görülmektedir. Buna göre kırmızı noktalar veri havuzu verilerini, siyah noktalar gerçek modelin veri havuzu dışı bölgedeki değerlerini, yeşil hat doğrusal vekil modelini ve mavi eğri Gauss vekil modelini göstermektedir.



**Şekil 4.4** Yaklaştırılmış gerçek ve vekil model davranışları, (siyah noktalar: gerçek model noktaları, kırmızı noktalar: veri havuzu, yeşil hat: doğrusal vekil model, mavi eğri: doğrusal olmayan vekil model).

İfade edilen sorunlar nedeniyle, optimizasyon süreçlerinde vekil modellerin kullanılması söz konusu olduğunda genellikle veri havuzu içinde kalınmaya çalışılır. Veri havuzu dışına çıkılması söz konusu olduğunda yapılabilecek en uygun davranışın, veri havuzu yakınlarında kalınması olduğu söylenebilir.

### 4.3 Regresyon yöntemleri

Regresyon yöntemleri<sup>ii</sup>, geçmişi 1800'lü yıllara kadar giden ve üzerinde uzunca bir zamandır çalışılan modelleme yöntemlerinden olup kelime anlamı itibari ile geriye doğru nedenselleme, daha basit ve ilkel şartlara doğru hareket manalarına gelmektedir<sup>(15)</sup>. Regresyon

<sup>i</sup> Gerçek model  $y = 0.5x^3 - 0.4x^2 + x^3 \cos(x) - 1$  şeklinde tanımlanmıştır.

<sup>ii</sup> Regression methods

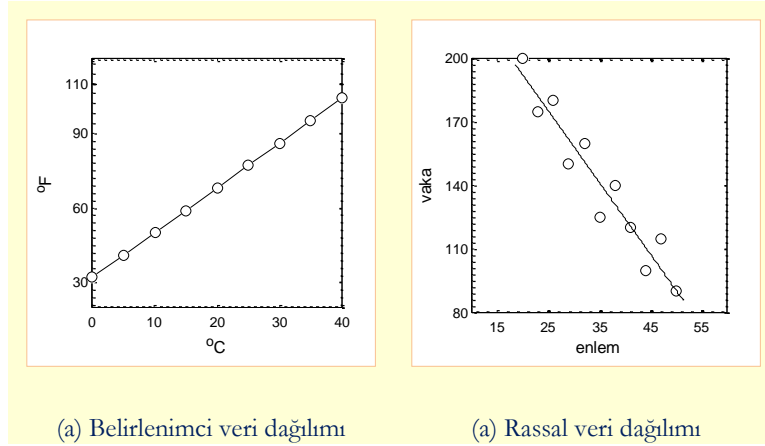






yöntemlerinin ilk formu niteliğindeki en küçük kareler yöntemi, 1805 yılında A. H. Legendre ve 1809 yılında C. F. Gauss tarafından yayımlanan makalelerde görülmektedir. Doğrudan doğruya regresyon terimini literatüre kazandıran bilim insanı ise 1885 tarihinde yayımladığı antropoloji içerikli makalesi ile F. Galton'dır<sup>(16)</sup>.

Regresyon analizi, istatistiksel modelleme disiplini kapsamında değerlendirilmekte olup bağımsız ve bağımsız değişkenlerle bağımlı değişkenler arasındaki ilişkiyi belirlemeye çalışan, daha da ötesi bağımsız değişkenler ile bağımlı olarak düşünülen değişkenler arasındaki bağımlılığı da ölçümlendirebilen bir bilim dalıdır. Geneli itibari ile bağımsız ve bağımlı değişkenler arasındaki ilişkinin, deterministik veya belirli oranda rassallık içeren istatistiksel nitelikte olabileceği söylenebilir. Regresyon yöntemlerinin üzerinde durduğu temel ilişki karakteristiği, rassal nitelikteki bağlantıdır. Bahse konu hususun daha iyi anlaşılabilmesi için müteakip örnekler üzerinde durulabilir.



Şekil 4.5 (a) Deterministik ve (b) rassal karaktere sahip veri dağılımları.

Şekil 4.5'in ilk seçeneğinde görülen dağılım türü, deterministik bir karaktere sahip olup veri havuzu dâhilindeki tüm noktalar, belirlenen doğrusal ilişki ile uyumlu olarak tam manasıyla hat üzerinde konuşlanabilmektedir. Diğer seçenekte, mükemmel bir ilişkiden ziyade bir genel

<sup>1</sup>Least squares method



yönelimi ve yönetime bağlı olarak değişen bir dağılım<sup>ii</sup> karakteristiği görülmektedir. Regresyon yöntemlerinin, deterministik ilişkilerden ziyade istatistiksel bağlantılar üzerinde yoğunlaşan bir yöntemler grubu olduğu söylenebilir<sup>(17)</sup>. Bununla beraber rassal ilişkiler için kurgulanan yöntemler, belirlenimci yapıya sahip ilişkilerin tespiti dâhilinde de rahatlıkla kullanılabilir.

Literatürde çok sayıda regresyon yönteminin varlığından bahsedilebilir. Söz konusu yöntemler bağımsız ve bağımlı değişken sayısına, bağımlı değişken türüne ve ilişki çeşidine göre tasnif edilebilir<sup>(18)</sup>.

Herhangi bir regresyon modeli temelde üç değişken türü içerir<sup>(19)</sup>;

- Bağımsız değişken/ler<sup>iii</sup>;  $\mathbf{x}$ ,
- Bağımlı değişken<sup>iv</sup>/ler;  $\mathbf{y}$ ,
- Bilinmeyen parametreler<sup>v</sup>,  $\boldsymbol{\beta}$

Söz konusu değişken ve parametreler arasındaki ilişki;

$$\mathbf{y} \cong f(\mathbf{x}, \boldsymbol{\beta}) \quad (4.10)$$

şeklinde tanımlanabilir. Değişkenler arasında doğrudan bir tür bağımlılığı söz konusu olmayıp bağımsız değişkenler bir sayıl veya yöney, bağımlı değişkenler de bir sayıl veya yöney türünde olabilir. Ancak bağımsız değişkenlerin veri sayısı ile bilinmeyen parametreler arasında belirli bir bağdan bahsedilebilir. Genellikle somut bir model ilişkisinin kurulabilmesi için veri havuzundaki  $n$  adet  $\mathbf{x}_i, i = 1, 2, \dots, n$  bağımsız değişken verisine karşılık  $\boldsymbol{\beta}_j, j = 1, 2, \dots, m$  olmak üzere  $n \geq m$  olması beklenir.

Literatürdeki en yaygın regresyon yaklaşımı doğrusal regresyondur. Bilinmeyen parametrelerin arasındaki bağlantının doğrusal karakterdeki bir fonksiyonla ilişkilendirildiği söz konusu yaklaşımda değişken sayılarına göre ilave tasnife gidilebilir. Örneğin  $\mathbf{x}$  ve  $\mathbf{y}$  değişkenlerinin sayıl

---

<sup>i</sup> Trend

<sup>ii</sup> Scatter

<sup>iii</sup> Regressor, exogenous variables, explanatory variables, covariates, input variables, predictor variables, or independent variables.

<sup>iv</sup> Regressand, endogenous variable, response variable, measured variable, criterion variable, yield, dependent variable.

<sup>v</sup> Parameter vector, effects, regression coefficients



olduğu bir modelleme yaklaşımı tekil doğrusal regresyonu<sup>i</sup> olarak adlandırılır.  $x$  değişkeninin yöney ve  $y$  değişkeninin sayıl olduğu bir modelleme yaklaşımı, çoğul doğrusal regresyonu<sup>ii</sup> olarak adlandırılır. İlave olarak  $x$  değişkeninin yöney ve  $y$  değişkeninin de yöney olduğu bir modelleme yaklaşımı, çok değişkenli doğrusal regresyonu<sup>iii</sup> olarak adlandırılır. Burada bir hususu hatırlatmakta fayda mülahaza edilmektedir; değişken sayısı ile o değişkene ait gözlem sayısı farklı kavramlardır. Örneğin bir bağımsız ve bir bağımlı değişkenin söz konusu olduğu regresyon modellemesinde  $n$  tane gözlem ve bu gözlemlere dayalı veri havusu olabilir.

#### 4.3.1 Tekil doğrusal regresyon

Elimizde bir bağımsız değişken ile bir bağımlı değişkenin olduğunu, bu ikiliye ait  $n$  adet gözlemin yapıldığını, dolayısıyla  $(x_i, y_i)$ ,  $i = 1, 2, \dots, n$  noktalarının bulunduğunu düşünelim. Tekil doğrusal regresyonun amacı söz konusu girdi verilerine karşılık en yakın çıktı verileri üretebilen doğrusal bir denklemin elde edilmesidir. Bu bağlamda öngörülebilecek denklem;

$$\hat{y} = \beta_0 + \beta_1 x \quad (4.11)$$

şeklinde modellenebilir. Genellikle modeldeki  $\beta_0$  terimi kestirme<sup>iv</sup> ve  $\beta_1$  terimi de eğim<sup>v</sup> olarak adlandırılır. Diğer taraftan daha önce de ifade edildiği üzere modelin öngördüğü çıktı değeri ile gerçek çıktı değeri arasında;

$$y_i = \hat{y}_i + \epsilon_i \quad (4.12)$$

ifadesi dikkate alınabilir. Burada  $\epsilon_i$  teriminin hata değerini<sup>vi</sup> temsil ettiğini de hatırlatalım. Modellemenin amacının, hata değerlerini minimize eden bir denklem kurgulaması olması nedeniyle Denklem 4.13'teki gibi bir sorun tanımlaması yapılabilir;

$$\min_{\beta} q(\beta) = \sum_{i=1}^n \epsilon_i^2 \quad (4.13)$$

<sup>i</sup> Simple linear regression

<sup>ii</sup> Multiple linear regression

<sup>iii</sup> Multivariate linear regression

<sup>iv</sup> Intercept

<sup>v</sup> Slope

<sup>vi</sup> Total error, residual, prediction error



Söz konusu sorun tanımlaması, literatürde daha önce de bahsedilen en küçük kareler yöntemi olarak adlandırılır. Verili bu ifadede hata tanımı dikkate alınarak;

$$\min_{\beta} q(\beta) = \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i)^2 \quad (4.14)$$

yazılabilir. Analitik bir ifade olan bu denklemi minimize etmek amacıyla her bir değişken için birinci mertebeden kısmi türev alma ve elde edilen ifadeleri sıfıra eşitleme işlemleri yapılırsa nihayetinde;

$$\frac{\partial q(\beta)}{\partial \beta_0} = 0, \quad \frac{\partial q(\beta)}{\partial \beta_1} = 0 \quad (4.15)$$

$$\beta_0 = \bar{y} - \beta_1 \bar{x} \quad (4.16)$$

$$\beta_1 = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2} \quad (4.17)$$

veya

$$\beta_1 = \frac{\bar{x}\bar{y} - \bar{x}\bar{y}}{\bar{x}^2 - \bar{x}^2} \quad (4.18)$$

denklemleri elde edilebilir. Burada;

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad (4.19)$$

$$\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i \quad (4.20)$$

$$\bar{x}\bar{y} = \frac{1}{n} \sum_{i=1}^n x_i y_i \quad (4.21)$$

$$\bar{x}^2 = \frac{1}{n} \sum_{i=1}^n x_i^2 \quad (4.22)$$





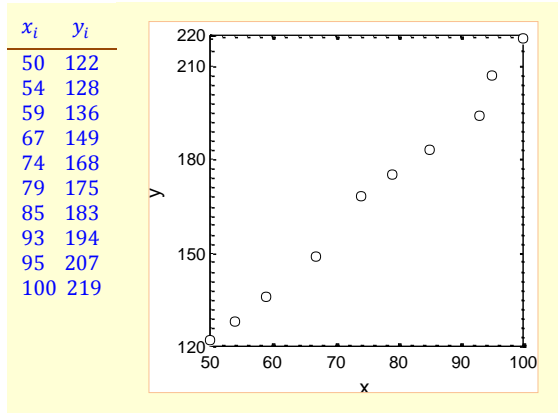
$$\bar{x}^2 = \left\{ \frac{1}{n} \sum_{i=1}^n x_i \right\}^2 \quad (4.23)$$

şeklindeki tanımlar dikkate alınır. Kestirme terimi olmadan yapılacak bir modellemede ise;

$$\hat{y} = \beta_1 x \quad (4.24)$$

$$\beta_1 = \frac{\sum_{i=1}^n x_i y_i}{\sum_{i=1}^n x_i^2} = \frac{\bar{xy}}{\bar{x^2}} \quad (4.25)$$

ifadelerine ulaşmak mümkündür<sup>(20)</sup>.



Şekil 4.6 Örnek veri dağılımı ve veriler<sup>(21)</sup>.

Elde edilen denklemlerin Şekil 4.6 dâhilinde verilen örnek veri dağılımına uygulanması durumunda;

$$\beta_1 = 1.8798$$

$$\beta_0 = 25.9877$$

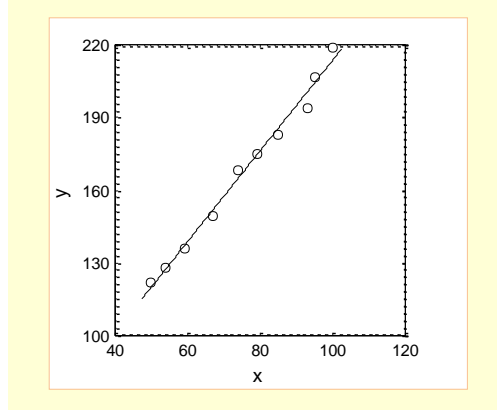
$$\hat{y} = 25.9877 + 1.8798x$$

denklemini elde edilebilir.





Ayrıca gerçek veriler ile veki model doğrusu çizimi müteakip şekilde görülmektedir;



Şekil 4.7 Örnek veri dağılımı ve tekil doğrusal regresyon modeli.

#### 4.3.2 Çoğul doğrusal regresyon

Daha önceki yonteme benzer şekilde elimizde  $p$  bağımsız deęişken ile bir bağımlı deęişkenin olduğunu, bu ikiliye ait  $n$  adet gözlemin yapıldığını, dolayısıyla  $(x_{i,j}, y_i); i = 1, 2, \dots, n; j = 1, 2, \dots, p$  noktalarının bulunduğunu düşünelim. Çoğul doğrusal regresyonun amacı, söz konusu girdi verilerine en yakın çıktı verileri üretebilen doğrusal bir denklemin elde edilmesidir. Buna göre model denklemi;

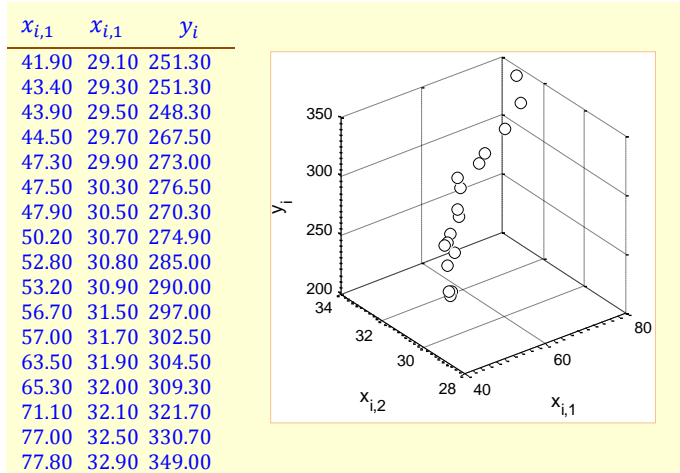
$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p \quad (4.26)$$

şeklinde belirlemek mümkündür. Buna mukabil modelin öngördüğü çıktı deęeri ile gerçek çıktı deęeri arasında da;

$$y_i = \hat{y}_i + \epsilon_i \quad (4.27)$$

ifadesi yazılabilir.





Şekil 4.8 Örnek veri dağılımı ve veriler<sup>(22)</sup>.

Regresyonun amacının toplam hata değerlerini minimize eden bir denklem kurgulaması olması nedeniyle müteakip sorun tanımlaması yapılabilir;

$$\min_{\beta} q(\beta) = \sum_{i=1}^n \epsilon_i^2 \quad (4.28)$$

Tanımlı bu ifade, hata denklemi dikkate alınarak;

$$\min_{\beta} q(\beta) = \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_1 - \beta_2 x_2 - \dots - \beta_p x_p)^2 \quad (4.29)$$

şeklinde tekrar yazılabilir. Analitik bir ifade olan bu denklemi minimize etmek amacıyla her bir değişken için birinci mertebeden kısmi türev alma ve elde edilen ifadeleri sıfıra eşitleme işlemleri yapılırsa nihayetinde;

$$\frac{\partial q(\beta)}{\partial \beta_i} = 0, \quad i = 1, 2, \dots, n \quad (4.30)$$



$$\beta = (X^T X)^{-1} X^T y \quad (4.31)$$

denklemleri elde edilebilir. Burada matris ve vektörler;

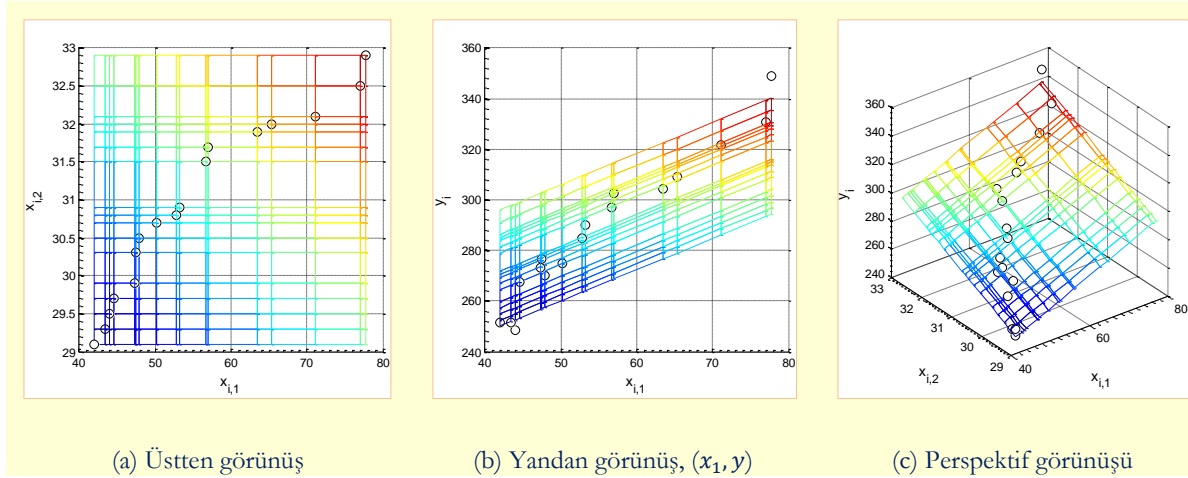
$$\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_n \end{bmatrix}, \quad X = \begin{bmatrix} 1 & x_{1,1} & x_{1,2} & \cdots & x_{1,p} \\ 1 & x_{2,1} & x_{2,2} & \cdots & x_{2,p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n,1} & x_{n,2} & \cdots & x_{n,p} \end{bmatrix}, \quad y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad (4.32)$$

şeklinde tanımlanmaktadır. Elde edilen denklemlerin Şekil 4.8'deki veri dağılımına uygulanması durumunda;

$$\begin{aligned} \beta_0 &= -153.5157 \\ \beta_1 &= 1.2387 \\ \beta_2 &= 12.0824 \end{aligned}$$

$$\hat{y} = -153.5157 + 1.2387x_1 + 12.0824x_2$$

yüzey denklemi elde edilebilir. Ayrıca gerçek veriler ile veki model yüzeyi çizimi müteakip şekillerde görülmektedir;



Şekil 4.9 Örnek veri dağılımı ve çoğul doğrusal regresyon modelinin 3 boyutlu ortamda farklı görüntüleri.





### 4.3.3 Çok değişkenli doğrusal regresyon

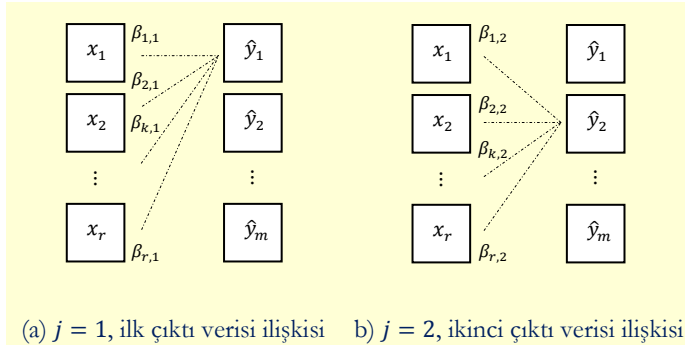
Elimizde birden fazla bağımsız ve bağımlı değişkenin olduğunu, bu gruba ait  $n$  adet gözlemin yapıldığını, dolayısıyla  $(\mathbf{x}_i, \mathbf{y}_i)$ ,  $i = 1, 2, \dots, n$  noktalarının bulunduğunu düşünelim. Ayrıca bağımsız değişkenlerin  $r$  tane, bağımlı değişkenlerin ise  $m$  tane olduğunu öngörelim;

$$\begin{aligned} \mathbf{x}_i &\rightarrow \{x_{i,1}, x_{i,2}, \dots, x_{i,r}\} \\ \mathbf{y}_i &\rightarrow \{y_{i,1}, y_{i,2}, \dots, y_{i,m}\} \end{aligned} \quad \Bigg| \quad i=1,2,\dots,n \quad (4.33)$$

Çok değişkenli doğrusal regresyonun amacı, söz konusu  $\mathbf{x}_i$  girdi verilerine karşılık en yakın  $\mathbf{y}_i$  çıktı verileri üretebilen doğrusal bir denklem takımının elde edilmesidir. Dolayısıyla modellemede doğrusallıktan başka dikkate alınan diğer bir husus da her bir çıktı değeri için ayrı bir modelin kurgulanması konusudur. Buna göre model denklemlerini;

$$\begin{aligned} \hat{y}_1 &= \beta_{01} + \beta_{11}x_1 + \beta_{21}x_2 + \dots + \beta_{r1}x_r \\ \hat{y}_2 &= \beta_{02} + \beta_{12}x_1 + \beta_{22}x_2 + \dots + \beta_{r2}x_r \\ &\vdots \\ \hat{y}_m &= \beta_{0m} + \beta_{1m}x_1 + \beta_{2m}x_2 + \dots + \beta_{rm}x_r \end{aligned} \quad (4.34)$$

şeklinde belirlemek mümkündür.



Şekil 4.10 Örnek veriler arasındaki ilişki ve çok değişkenli doğrusal regresyon modeli.

Buna mukabil modelin öngördüğü çıktı değeri ile gerçek çıktı değeri arasında da;



$$\hat{y}_{i,j} = \sum_{k=1}^r x_{i,k} \beta_{k,j} \quad \left| \begin{array}{l} i=1,2,\dots,n \\ j=1,2,\dots,m \end{array} \right. \quad (4.35)$$

$$y_{i,j} = \hat{y}_{i,j} + \epsilon_{i,j} \quad (4.36)$$

ifadesi yazılabilir. Regresyonun amacının toplam hata değerlerini minimize eden bir denklem kurgulaması olması nedeniyle müteakip sorun tanımlaması yapılabilir;

$$\min_{\beta} q(\beta) = \sum_{i=1}^n \sum_{j=1}^m \epsilon_{i,j}^2 \quad (4.37)$$

Tanımlı bu sorun, hata denklemi dikkate alınarak;

$$\min_{\beta} q(\beta) = \sum_{i=1}^n \sum_{j=1}^m (y_{i,j} - \sum_{k=1}^r x_{i,k} \beta_{k,j})^2 \quad (4.38)$$

şeklinde tekrar yazılabilir. Analitik bir ifade olan bu denklemi minimize etmek amacıyla birinci mertebeden kısmi türev alma ve elde edilen ifadeleri sıfıra eşitleme işlemleri yapılırsa nihayetinde;

$$\beta = (X^T X)^{-1} X^T Y \quad (4.39)$$

denklemi elde edilebilir. Buradaki terimler matris şeklinde olup;

$$\beta = \begin{bmatrix} \beta_{0,1} & \beta_{0,2} & \beta_{0,3} & \cdots & \beta_{0,m} \\ \beta_{1,1} & \beta_{1,2} & \beta_{1,3} & \cdots & \beta_{1,m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \beta_{r,1} & \beta_{r,2} & \beta_{r,3} & \cdots & \beta_{r,m} \end{bmatrix} \quad (4.40)$$

$$X = \begin{bmatrix} 1 & x_{1,1} & x_{1,2} & \cdots & x_{1,r} \\ 1 & x_{2,1} & x_{2,2} & \cdots & x_{2,r} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n,1} & x_{n,2} & \cdots & x_{n,r} \end{bmatrix} \quad (4.41)$$





$$Y = \begin{bmatrix} y_{1,1} & y_{1,2} & y_{1,3} & \cdots & y_{1,m} \\ y_{2,1} & y_{2,2} & y_{2,3} & \cdots & y_{2,m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ y_{n,1} & y_{n,2} & y_{n,3} & \cdots & y_{n,m} \end{bmatrix} \quad (4.42)$$

olarak tanımlanmaktadır.

S/N	Adı	Ekvatoryal Yarıçap $r_1(km)$	Kutupsal Yarıçap $r_2(km)$	Küresel Yarıçap $r_3(km)$	Kütle $m(kg)$	Yoğunluk $\rho(g/cm^3)$
1	Ceres	975	975	909	9.46 x1020	2.09
2	Pallas	570	525	500	2.01 x1020	2.57
3	Juno	320	267	200	2.86 x1019	3.20
4	Vesta	578	560	458	2.59 x1020	3.33
5	Hebe	205	185	170	1.27 x1019	3.77
6	Iris	240	200	200	1.62 x1019	3.21
7	Flora	161	161	161	6.62 x1018	3.04
8	Metis	222	182	130	1.13 x1019	4.12
9	Hygiea	530	407	370	8.67 x1019	2.08
10	Parthenope	153	153	153	6.15 x1018	3.28
11	Egeria	208	208	208	1.59 x1019	3.38
12	Irene	167	153	139	6.94 x1019	3.73
13	Eunomia	357	255	212	3.18 x1019	3.14
14	Psyche	240	185	145	2.27 x1019	6.73
15	Thetis	117	90	90	1.43 x1018	2.89
16	Melpomene	170	155	129	3.00 x1018	1.69
17	Fortuna	225	205	195	8.31 x1018	1.77
18	Massalia	160	145	132	3.34 x1018	2.08

Çizelge 4.1 Örnek veri havuzu<sup>(23)</sup>.

Çok değişkenli doğrusal regresyon yönteminin daha iyi anlaşılabilmesi için Çizelge 4.1'deki gibi bir veri havuzu dikkate alınabilir. Buna göre gökyüzündeki 18 göktaşına ait ebat bilgileri (ekvatoryal ( $r_1$ ), kutupsal ( $r_2$ ) ve küresel ( $r_3$ ) yarıçap değerleri) ile bu göktaşlarına ait kütle ve yoğunluk değerlerinin bilindiğini düşünelim. Söz konusu veriler dâhilinde ebat bilgilerinin girdi niteliğinde, kütle ( $m$ ) ve yoğunluk ( $\rho$ ) verilerinin ise çıktı niteliğinde olduğunu öngörelim. Dolayısıyla;

$$n = 18, \quad r = 3, \quad m = 2$$

$$\mathbf{x}^T = [r_1 \quad r_2 \quad r_3]$$

$$\mathbf{y}^T = [m \quad \rho]$$

yazılabilir.



Modelleme dâhilinde ihtiyaç duyulan matris ifadeleri, verili çizelge değerlerinden hareketle;

$$X = \begin{bmatrix} 1 & 975 & 975 & 909 \\ 1 & 570 & 525 & 500 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 160 & 145 & 132 \end{bmatrix}_{18 \times 4}, \quad Y = \begin{bmatrix} 9.46 \times 10^{20} & 2.09 \\ 2.01 \times 10^{20} & 2.57 \\ \vdots & \vdots \\ 3.34 \times 10^{18} & 2.08 \end{bmatrix}_{18 \times 2}$$

şeklinde ifade edilebilir. Belirlenen matris değerleri daha önce elde edilen bilinmeyenler matrisi denkleminde yerine koyulursa;

$$\beta = (X^T X)^{-1} X^T Y$$

$$\beta = \begin{bmatrix} 1 & 1 & \dots & 1 \\ 975 & 570 & \dots & 160 \\ 975 & 525 & \ddots & 145 \\ 909 & 500 & \dots & 132 \end{bmatrix}^{-1} \begin{bmatrix} 1 & 975 & 975 & 909 \\ 1 & 570 & 525 & 500 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 160 & 145 & 132 \end{bmatrix} \begin{bmatrix} 1 & 1 & \dots & 1 \\ 975 & 570 & \dots & 160 \\ 975 & 525 & \dots & 145 \\ 909 & 500 & \dots & 132 \end{bmatrix} \begin{bmatrix} 9.46 \times 10^{20} & 2.09 \\ 2.01 \times 10^{20} & 2.57 \\ \vdots & \vdots \\ 3.34 \times 10^{18} & 2.08 \end{bmatrix}$$

$$\beta = \begin{bmatrix} 0.2029 & -0.0016 & 0.0014 & -1.6503 \times 10^{-4} \\ -0.0016 & 5.5581 \times 10^{-5} & -7.9726 \times 10^{-5} & 2.6112 \times 10^{-5} \\ 0.0014 & -7.9726 \times 10^{-5} & 2.1099 \times 10^{-4} & -1.4146 \times 10^{-4} \\ -1.6503 \times 10^{-4} & 2.6112 \times 10^{-5} & -1.4146 \times 10^{-4} & 1.2567 \times 10^{-4} \end{bmatrix} \begin{bmatrix} 1.7302 \times 10^{21} & 56.1 \\ 1.2875 \times 10^{24} & 1.6422 \times 10^4 \\ 1.2545 \times 10^{24} & 1.4539 \times 10^4 \\ 1.1513 \times 10^{24} & 1.2890 \times 10^4 \end{bmatrix}$$

$$\beta = \begin{bmatrix} -1.3221 \times 10^{20} & 3.425 \\ -1.1969 \times 10^{18} & -7.4454 \times 10^{-4} \\ 1.6591 \times 10^{18} & 0.0152 \\ 5.5468 \times 10^{17} & -0.0172 \end{bmatrix}$$

sonucuna ulaşılır. Buna göre model denklemlerini;

$$\hat{y}_1 = \beta_{01} + \beta_{11}x_1 + \beta_{21}x_2 + \beta_{31}x_3$$

$$\hat{y}_1 = -1.3221 \times 10^{20} - 1.1969 \times 10^{18}x_1 + 1.6591 \times 10^{18}x_2 + 5.5468 \times 10^{17}x_3$$

$$\hat{y}_2 = \beta_{02} + \beta_{12}x_1 + \beta_{22}x_2 + \beta_{32}x_3$$

$$\hat{y}_2 = 3.425 - 7.4454 \times 10^{-4}x_1 + 0.0152x_2 - 0.0172x_3$$

şeklindeki yüzey denklemleri olarak belirlemek mümkündür.

#### 4.3.4 Polinomiyal doğrusal regresyon

Şimdiye kadar detayları verilen tekil, çoğul ve çok değişkenli doğrusal regresyon yöntemlerinde dikkate alınan temel yaklaşım, birinci dereceden polinomiyal denklem



kurgusuydu. Dolayısıyla elde edilen denklemler de doğru veya düzlem şeklindeki yüzeylerdi. Ancak modellemede 1inci dereceden polinom yerine daha yüksek dereceden polinomial denklemler de kurgulanabilir. Örneğin bir bağımsız ve bir bağımlı değişkenin söz konusu olduğu,  $n$  gözleme dayalı veri havuzu için;

$$\hat{y}_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 \quad | \quad i=1,2,\dots,n \quad (4.43)$$

model yaklaşımı yapılabilir. İfade edilen model her ne kadar 2nci dereceden bir polinomial denklem içerse de model hala doğrusal regresyon modeli olarak adlandırılır. Zira modelin bilinmeyenleri niteliğindeki  $\beta_j$  parametreleri hala 1inci dereceden polinomial karakterdedir. Aynı veri havuzu özellikleri için  $p$ 'ninci dereceden genelleştirilmiş polinomial doğrusal regresyon modelini;

$$\hat{y}_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \dots + \beta_p x_i^p \quad | \quad i=1,2,\dots,n \quad (4.44)$$

şeklinde tanımlamak mümkündür. Veri havuzundaki girdi ve çıktı değerleri arasında;

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^p \\ 1 & x_2 & x_2^2 & \dots & x_2^p \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^p \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix} \quad (4.45)$$

ilişkisi kolayca yazılabilir. Tıpkı daha önceki regresyon yaklaşımlarında olduğu gibi yöntemin amacının, toplam hata değerlerini minimize eden bir denklem kurgulaması olması nedeniyle müteakip sorun tanımlaması yapılabilir;

$$\min_{\beta} q(\beta) = \sum_{i=1}^n \epsilon_i^2 \quad (4.46)$$

Tanımlı bu ifade, hata tanımı dikkate alınarak;

$$\min_{\beta} q(\beta) = \sum_{i=1}^n (y_i - \sum_{k=1}^{p+1} x_{i,k} \beta_k)^2 \quad (4.47)$$

şeklinde tekrar yazılabilir.



Analitik bir ifade olan bu denklemi minimize etmek amacıyla birinci mertebeden kısmi türev alma ve elde edilen ifadeleri sıfıra eşitleme işlemleri yapılırsa nihayetinde;

$$\beta = (X^T X)^{-1} X^T y \quad (4.48)$$

denklemi elde edilebilir. Denklemden geçen  $X$  matrisi tanımı itibarı ile  $x_{i,j} = x_i^{j-1}$  şeklinde de ifade edilebilmekte olup literatürde Vandermonde matrisi olarak bilinir<sup>(24)</sup>.

Polinomial doğrusal regresyon yönteminin daha iyi anlaşılabilmesi için Şekil 4.11'deki gibi bir veri havuzu dikkate alınabilir. Buna göre elimizde ABD'nin Florida eyaletindeki timsahlara ait uzunluk (inch) ve kütle (pound) içerikli 25 adet ölçüm verisi olduğunu düşünelim. Söz konusu veriler için 2nci dereceden polinomial doğrusal regresyon modeli kurgulaması kapsamında;

$$\hat{y}_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 \quad |^{i=1,2,\dots,25}$$

yaklaşımı yazılabilir. Bu modele ait bilinmeyen parametrelerin elde edilmesi amacıyla;

$$\beta = (X^T X)^{-1} X^T y$$

$$X = \begin{bmatrix} 1 & 94 & 8836 \\ 1 & 74 & 5476 \\ \vdots & \vdots & \vdots \\ 1 & 78 & 6084 \end{bmatrix}, \quad y = \begin{bmatrix} 130 \\ 51 \\ \vdots \\ 57 \end{bmatrix}$$

ifadeleri dikkate alınarak;

$$\beta = \begin{bmatrix} 410.4841 \\ -11.3176 \\ 0.0866 \end{bmatrix}$$

sonucuna ulaşılabilir. Elde edilen parametre değerlerinin model denkleme konulması durumunda;

$$\begin{aligned} \beta_0 &= 410.4841 \\ \beta_1 &= -11.3176 \\ \beta_2 &= 0.0866 \end{aligned}$$

$$\hat{y} = 410.4841 - 11.3176x + 0.0866x^2$$

eşitliği elde edilebilir.

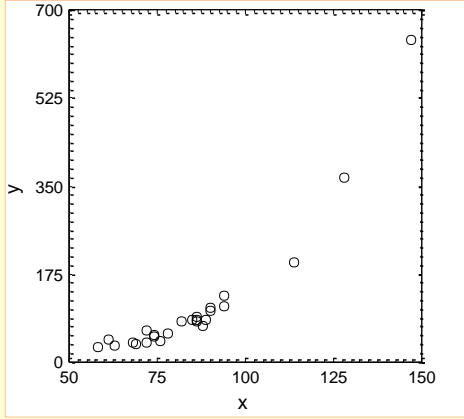




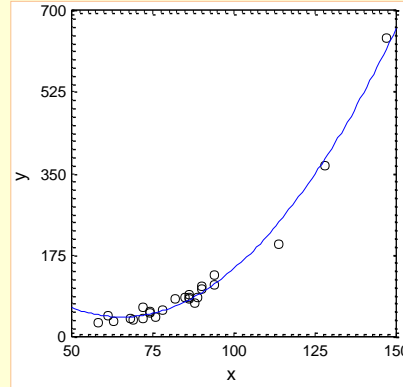
Ayrıca gerçek veriler ile vekil model eğrisi çizimi Şekil 4.12'de görülmektedir;

$$x_i = [94 \ 74 \ 147 \ 58 \ 86 \ 94 \ 63 \ 86 \ 69 \ 72 \ 128 \ 85 \ 82 \ 86 \ 88 \ 72 \ 74 \ 61 \ 90 \ 89 \ 68 \ 76 \ 114 \ 90 \ 78]$$

$$y_i = [130 \ 51 \ 640 \ 28 \ 80 \ 110 \ 33 \ 90 \ 36 \ 38 \ 366 \ 84 \ 80 \ 83 \ 70 \ 61 \ 54 \ 44 \ 106 \ 84 \ 39 \ 42 \ 197 \ 102 \ 57]$$



Şekil 4.11 Örnek veri dağılımı ve veriler<sup>(25)</sup>.



Şekil 4.12 Örnek veri dağılımı ve polinomiyal doğrusal regresyon modeli.

#### 4.3.5 Doğrusal olmayan regresyon

Doğrusal olmayan regresyon modeli yaklaşımında girdi ve çıktı verileri arasındaki ilişki, bilinmeyen niteliğindeki regresyon katsayılarının doğrusal olmayan kombinasyonu ile modellenir. Daha önce genel regresyon modeli için belirtilen;

$$y \cong f(x, \beta) \quad (4.49)$$

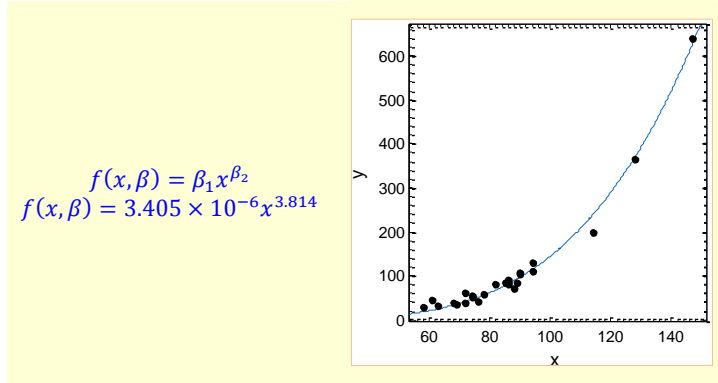
şeklindeki ifade, doğrusal olmayan regresyon modeli için de aynen geçerli olup doğrusal modelleme yaklaşımı arasındaki yegane fark,  $f$  fonksiyonunun  $\beta$  parametrelerine bağlı olarak eğrisellik içermesidir. Literatürde değişik doğrusal olmayan fonksiyon örnekleri mevcut olup bu örnekler arasında üssel, logaritmik, trigonometrik ve Gauss fonksiyonları sayılabilir. Regresyon modeli kapsamında regresyon katsayılarının eldesi için analitik eşitliklerden bahsetmek zordur. Genellikle modelleme için gerekli katsayı değerleri, hata değerlerinin minimizasyonu amaçlı



eniyleme süreçleri neticesinde elde edilir. Örnek olması amacıyla Denklem 4.50 dâhilinde doğrusal olmayan bir regresyon modeli verilmektedir<sup>(26)</sup>;

$$f(x, \beta) = \beta_1 x^{\beta_2} \quad (4.50)$$

Görüldüğü üzere bir bağımsız ve bir bağımlı verinin söz konusu olduğu modellemede dikkate alınan üssel fonksiyon, doğrusal olmayan bir karakterdedir. Bir önceki başlık dâhilinde polinomial doğrusal regresyon modeli ile modellenen veri havuzu, üssel karakterdeki doğrusal olmayan regresyon yaklaşımı ile modellenmiş olup veri değerleri ve uydurulan eğri modeli müteakip şekilde görülmektedir;



Şekil 4.13 Örnek veri dağılımı ve üssel fonksiyon bazlı doğrusal olmayan regresyon modeli.

Literatürde doğrusal ve doğrusal olmayan regresyon modeli yaklaşımlarından farklı olarak değişik isimlerde ve farklı özelliklerde regresyon yaklaşımları da mevcuttur. Bahse konu diğer regresyon yöntemleri arasında lojistik regresyon<sup>i</sup>, adım-adım regresyon<sup>ii</sup>, Ridge regresyon, Lasso regresyon ve elastik-net regresyon<sup>iii</sup> yöntemleri sayılabilir. İfade edilen yöntemler arasında özellikle lojistik regresyon yöntemi yaygın bir yöntem olup ikili temsilin söz konusu olduğu değişkenleri içeren sorunlarda kullanılabilir<sup>(27)</sup>.

<sup>i</sup> Logistic regression

<sup>ii</sup> Stepwise regression

<sup>iii</sup> Elastic regression







2B ortamda polinomiyal doğrusal regresyon model eldesi.

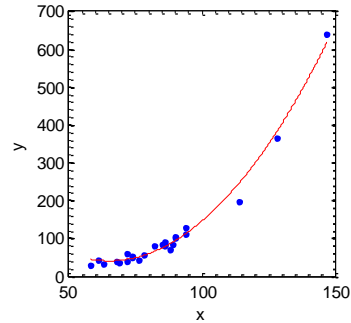


```
% xData ve yData verisi olarak timsah veri havuzu dikkate alınmıştır.  
ft = fitype( 'poly2' ); opts = fitoptions( ft );  
opts.Lower = [-Inf -Inf -Inf]; opts.Upper = [Inf Inf Inf];  
  
% model eldesi  
[fitresult, gof] = fit( xData, yData, ft, opts );  
  
% model ve veri çizimi  
figure( 'Name', 'untitled fit 1' );  
h = plot( fitresult, xData, yData );  
xlabel( 'x' ); ylabel( 'y' );  
axis square
```

**MATLAB uygulaması**



Sonuç



**Şekil 4.14** 2nci dereceden polinomiyal doğrusal regresyon modeli.



#### 4.3.6 Regresyonla ilgili ilave hususlar

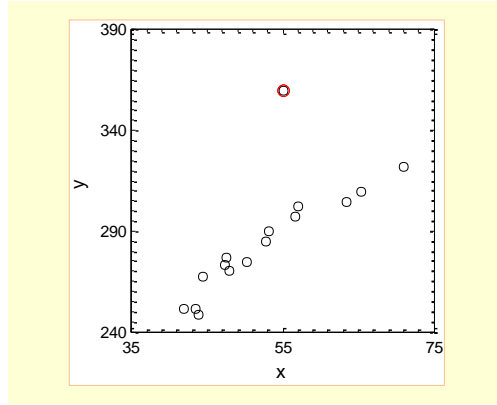
Regresyon modellemesi süreçlerinde veri havuzunun büyüklüğü önem arz eden bir husustur. Genellikle veri havuzunun kapsayıcı ve büyük olması, regresyon modelinin gübüzlüğünü ve hassasiyetini artıran bir unsur olarak görülür. Bağımsız değişken sayısına bağlı





olarak veri havuzunun büyüklüğünün 20:1 oranında olması, genellikle kabul gören bir yaklaşımdır. Örneğin iki bağımsız değişkenin söz konusu olduğu bir regresyon modellemesinde 40 gözlem değerinden müteşekkil bir veri havuzunun olması gereğinden bahsedilebilir. Bununla beraber modellemede asgari gözlem sayısından bahsetmek gerekirse söz konusu oranın 5:1 şeklinde olabileceği söylenebilir.

Diğer taraftan gözlem sayısı yanında gözlem değerlerinin dağılımı da önem arz eden diğer bir husustur. Önceden belirlenmeden rastgele olarak elde edilen gözlem değerlerinde aşırı farklılıkların olması, verinin doğruluğu konusunda bazı şüpheler doğurabilir. Söz konusu aşırı farklı değerler, modellemede aykırı veya sıradışı veri olarak adlandırılmakta olup genellikle veri havuzuna ait ortalama değerden 3 standart sapma değeri üstü veya altı değerler sıradışı değer olarak kabul edilebilir. Regresyon modellemesinde genellikle sıradışı değerler modelleme dışı tutulur<sup>(28)</sup>.



Şekil 4.15 Örnek veri dağılımı ve sıra dışı veri örneği, (kırmızı renkli veri konumu).

#### 4.4 Mesafe ağırlıklı yaklaşım

Mesafe ağırlıklı<sup>i</sup> modelleme yaklaşımı, ara değer kestirim<sup>iii</sup> yöntemlerinden biri olup diğer modelleme yöntemlerine göre basit ve etkili bir yöntemdir. Bununla beraber basit olmasına

<sup>i</sup> Outlier

<sup>ii</sup> Inverse distance weighting

<sup>iii</sup> Interpolation



rağmen geçmişi çok eskilere değil, nisbeten yakın zamanlara, 1960'lı yıllara dayanır. Bilgisayar bazlı grafik çizimlerinin yoğun olarak çalışıldığı 1960'lı yıllarda, D. Shepard'ın yayımladığı “Düzensiz veriler için iki boyutlu ara kestirim fonksiyonu” isimli makale, mesafe ağırlıklı ara değer kestirim yönteminin literatüre kazandırıldığı yayın olarak anılır<sup>(29)</sup>. Yaklaşımın temelinde, veri havuzundaki girdi ve çıktı değerlerinden yararlanılarak herhangi bir başka noktaya ait çıktı değerinin hesaplanması düşüncesi bulunmaktadır. Ancak bu yapılırken;

► “Herşey her şeyle bir şekilde ilişkilidir. Ancak yakın şeyler uzak şeylerden daha çok ilişkilidir.”

W. Tobler (1970)

hususu dikkate alınır<sup>(30)</sup>. Genellikle çıktı değerlerinin değerlendirilmesi, doğrusal bir ilişki çerçevesinde ve mesafeye göre ağırlıklandırılarak yapılır. Buna göre elimizde  $n$  tane farklı konum  $(\mathbf{x}_i)$  ve bu konumlara ait çıktı verisi  $(y_i)$  çiftinin bulunduğunu düşünelim;

$$y_i = y(\mathbf{x}_i), \quad i = 1, 2, \dots, n \quad (4.51)$$

Verili bu değerlerden hareketle mesafe ağırlıklı yaklaşım yönteminde herhangi bir  $\mathbf{x}$  noktasındaki çıktı değeri;

$$\hat{y}(\mathbf{x}) = \begin{cases} \sum_{i=1}^n w_i y(\mathbf{x}_i) & d(\mathbf{x}, \mathbf{x}_i) \neq 0 \\ y(\mathbf{x}_i) & d(\mathbf{x}, \mathbf{x}_i) = 0 \end{cases} \quad (4.52)$$

şeklindeki ifadeye göre belirlenir. Denklemdaki  $w_i$  terimi,  $i$ nci veri noktasına ait ağırlık değeri olup genellikle o noktanın  $\mathbf{x}$  noktasına olan mesafesi ile ilişkilendirilir;

$$w_i = \frac{1}{\sum_{i=1}^n \frac{1}{d(\mathbf{x}, \mathbf{x}_i)^p}} \quad (4.53)$$

Tanımdan da görüldüğü üzere ağırlık katsayıları, noktalar arası mesafe ile ters orantılı olup mesafe arttıkça ağırlık katsayısı değeri azalmaktadır. Ayrıca ağırlık katsayıları arasındaki;

<sup>i</sup> “A two-dimensional interpolation function for irregularly-spaced data.”



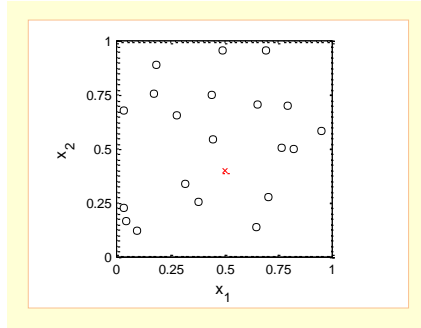
$$\sum_{i=1}^n w_i = 1 \quad (4.54)$$

ilişki koşulu da genellikle dikkate alınır. Diğer taraftan ağırlık katsayısı denkleminde geçen  $d(\mathbf{x}, \mathbf{x}_i)$  terimi, iki nokta arası mesafe değeri olup;

$$d(\mathbf{x}, \mathbf{x}_i) = \sqrt{\sum_{j=1}^m (x_j - x_{i,j})^2} \quad (4.55)$$

şeklindeki Euclidian uzaklığını,  $p$  kullanıcı tarafından belirlenen ve üssel parametre<sup>i</sup> olarak adlandırılan sabit bir sayıyı ve  $m$  konuma ait boyutu göstermektedir.

Mesafe ağırlıklı modelleme yaklaşımı, eldeki veri havuzu dikkate alındığında yerel, hatasız<sup>ii</sup> ve deterministik bir yaklaşım olarak tanımlanabilir. Yöntemin yerel karakterde olmasının sebebi, belirli bir veri havuzunun dikkate alınmasından kaynaklanmaktadır. Veri havuzunun büyük olması durumunda,  $n$  tane verinin tümü yerine belirli bir miktardaki  $q$  adet noktanın dikkate alınması daha yaygındır. Bahse konu  $q$  adet noktanın nasıl seçileceği hususunda değişik yaklaşımlar yapılabilir. Örneğin Şekil 4.16'daki gibi elimizde  $n = 20$  adet verinin olduğunu ve bu verilerden hareketle  $(0.5, 0.4)$  konumuna ait çıktı verisi değerinin hesaplanmasını öngörelim;



Şekil 4.16 Örnek veri dağılımı ve çıktı değeri hesaplanmak istenen  $\mathbf{x}$  konumu (kırmızı renkli).

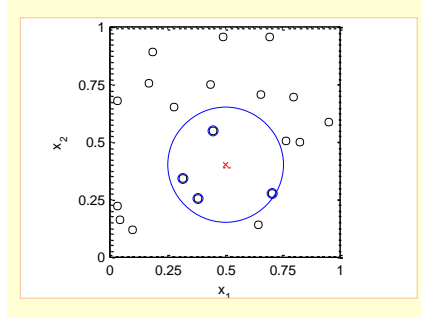
<sup>i</sup> Power parameter

<sup>ii</sup> exact



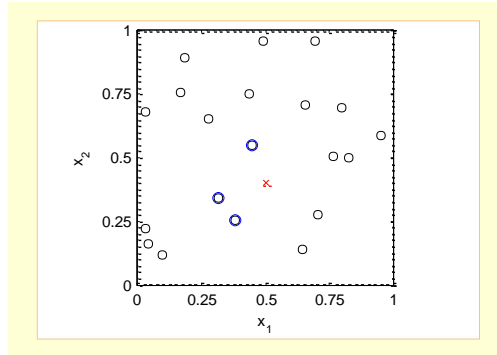


Yaygın olarak dikkate alınan yaklaşımlardan ilki, hesaplanması istenilen  $\mathbf{x}$  konumu merkez olmak üzere belirli bir  $r$  yarıçapına sahip daire dâhilindeki verilerin dikkate alınmasıdır. Müteakip şekilden de gözlemlenebileceği üzere dikkate alınan daire büyüklüğüne göre  $q = 4$  adet yakın çevre bölge noktası hesaplamalarda dikkate alınabilir.



Şekil 4.17 Dairesel bölge tanımlaması ve dikkate alınan veriler.

Bir diğer yaklaşımda bölge tanımlaması yerine Euclidian mesafe değerleri dikkate alınarak en yakın  $q$  adet nokta seçimi yapılabilir. Örneğin en yakın üç noktanın hesaplamalarda dikkate alınması durumunda müteakip konumlar dikkate alınabilir;



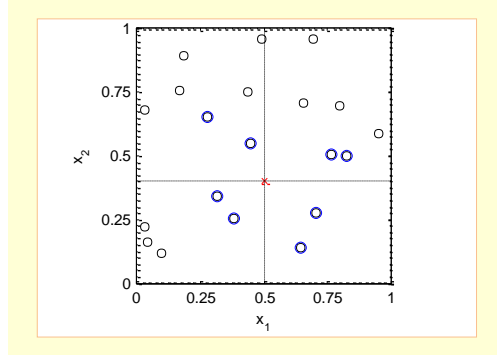
Şekil 4.18 Nokta sayısı tanımlaması ve dikkate alınan veriler (mavi noktalar).

<sup>i</sup> proximity



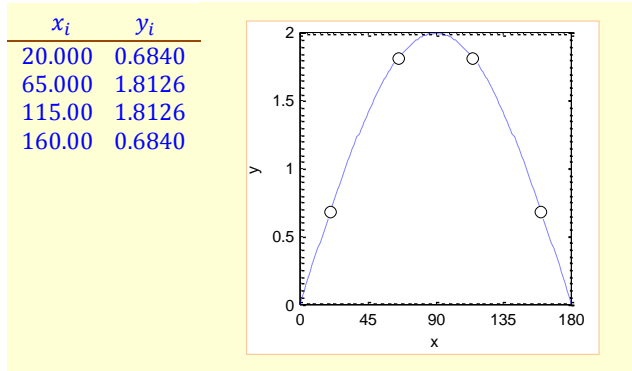


Yerel nokta tanımlamasında bahsedilebilecek son örnek,  $x$  konumu merkez olmak üzere tüm veri uzayının belirli bölgelere bölünmesi ve bölünen bölgelerdeki verilerin belirli kısımlarının ve mesela belirli sayıdaki en yakın noktaların dikkate alınmasıdır. Örneğin Şekil 4.19'daki gibi bir veri uzayı dört bölgeye bölünerek en yakın ikişer nokta tanımlamaları yapılabilir<sup>(31)</sup>. Buna göre dikkate alınan dört bölgedeki nokta sayıları toplamı  $q = 8$  olup bölge sayısı arttıkça örneklem sayısı da artacaktır;



Şekil 4.19 Bölge ve nokta sayısı tanımlaması ile dikkate alınan veriler (mavi noktalar).

Mesafe ağırlıklı modelleme yaklaşımının daha iyi anlaşılabilmesi için basit ve az sayıdaki müteakip verilerin kullanılması ile bir vekil model belirleme sürecine gidilebilir;



Şekil 4.20 Veri havuzu değerleri ve gerçek model eğrisi.



Buna göre  $n = 4$  olup eldeki veri havuzunun gerçek modeli  $y = 2\sin(x)$  şeklindedir. Az sayıda verinin bulunması nedeniyle, verili dört noktanın tümünün mesafe ağırlıklı modellemede kullanılması uygun bir yaklaşım tarzı olacaktır. Bu bağlamda örneğin  $x = 90^\circ$  değeri için yapılan tahminlemede öncelikle söz konusu konum ile veri havuzu konumları arasındaki Euclidian mesafe değerleri hesaplanabilir;

$$\begin{aligned}d(90,20) &= \sqrt{(90 - 20)^2} = 70 \\d(90,65) &= \sqrt{(90 - 65)^2} = 25 \\d(90,115) &= \sqrt{(90 - 115)^2} = 25 \\d(90,160) &= \sqrt{(90 - 160)^2} = 70\end{aligned}$$

Elde edilen bu değerler tahminleme için dikkate alınan ağırlık denkleminde yerine konulduğunda;

$$w_i = \frac{1}{\frac{d(x, x_i)^p}{\sum_{i=1}^n \frac{1}{d(x, x_i)^p}}, \quad p = 2$$

$$\begin{aligned}w_1 &= \frac{\frac{1}{d(x, x_1)^2}}{\sum_{i=1}^4 \frac{1}{d(x, x_i)^2}} = \frac{\frac{1}{70^2}}{\frac{1}{70^2} + \frac{1}{25^2} + \frac{1}{25^2} + \frac{1}{25^2}} = 0.0566, & w_2 &= \frac{\frac{1}{d(x, x_2)^2}}{\sum_{i=1}^4 \frac{1}{d(x, x_i)^2}} = \frac{\frac{1}{25^2}}{\frac{1}{70^2} + \frac{1}{25^2} + \frac{1}{25^2} + \frac{1}{25^2}} = 0.4434 \\w_3 &= \frac{\frac{1}{d(x, x_3)^2}}{\sum_{i=1}^4 \frac{1}{d(x, x_i)^2}} = \frac{\frac{1}{25^2}}{\frac{1}{70^2} + \frac{1}{25^2} + \frac{1}{25^2} + \frac{1}{25^2}} = 0.4434, & w_4 &= \frac{\frac{1}{d(x, x_4)^2}}{\sum_{i=1}^4 \frac{1}{d(x, x_i)^2}} = \frac{\frac{1}{70^2}}{\frac{1}{70^2} + \frac{1}{25^2} + \frac{1}{25^2} + \frac{1}{25^2}} = 0.0566\end{aligned}$$

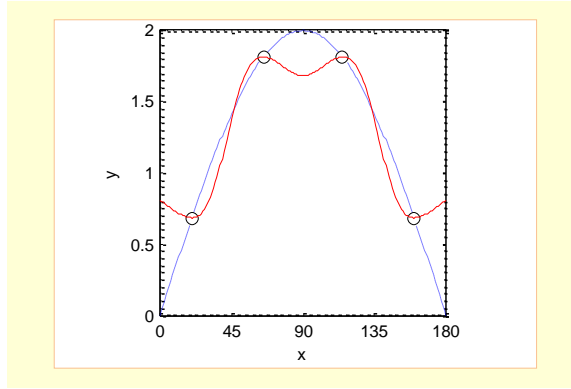
ağırlık değerlerine ulaşılır. Bulunan bu ağırlık değerleri, tahminleme için dikkate alınan doğrusal nitelikteki ana denklemde yerine koyulursa;

$$\hat{y}(x) = \begin{cases} \sum_{i=1}^n w_i y(x_i) & d(x, x_i) \neq 0 \\ y(x_i) & d(x, x_i) = 0 \end{cases}$$

$$\begin{aligned}\hat{y}(90) &= 0.0566 \times 0.6840 + 0.4434 \times 1.8126 + 0.4434 \times 1.8126 + 0.0566 \times 0.6840 \\ \hat{y}(90) &= 1.6849\end{aligned}$$

değerine ulaşılır. Benzer model yaklaşımı  $x = 0, 1, \dots, 180$ ;  $\Delta x = 1^\circ$  değerleri için gerçekleştirildiğinde Şekil 4.21'deki gibi bir karşılaştırma görmek mümkündür;





Şekil 4.21 Veri havuzu, gerçek model (mavi-kesikli) ve vekil model (kırmızı-süreklî) eğrileri.

Şekilden de gözlemlenebileceği üzere ara değer kestirimi bağlamında veri havuzuna ait konumlarda vekil model hatasız olarak sonuç vermekte, ancak özellikle orta bölgelerde tahminleme hatası yüksek değerlerde görülmektedir. Öte yandan problemin doğasına göre  $p$  üssel parametre değeri, bölgesel olarak hata değerlerini azaltabilir ya da artırabilir.

1B ortamda  $p = 1,2,3$  değerleri için mesafe ağırlıklı vekil model eldesi.

MATLAB uygulaması

```

clc; clear
% veri havuzu
xs = [20 65 115 160]; ys = [0.684 1.8126 1.8126 0.684];
for p = 1:3 ; % ussel parametre
    t = 0:180 ; % konumlar
    for i = 1:length(t)
        k = find(t(i)==xs) ;
        if k>0
            y(i) = ys(k);
        else
            for j = 1:length(xs)
                d(j) = abs(t(i)-xs(j)); % mesafe hesabı
            end
            top = 0;
            for j=1:length(ys)
                w(j) = [1/(d(j))^p]/sum(1./d.^p) ; % ağırlık katsayısı
                top = top + w(j)*ys(j);
            end
            y(i) = top;
        end
    end
end

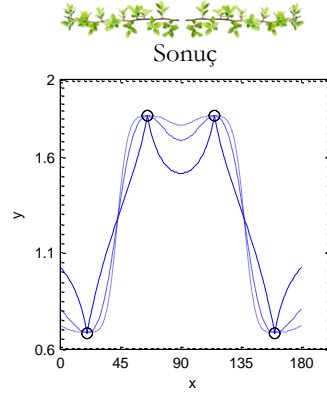
```







```
end  
end  
plot(xs, ys, 'ko'); hold on; plot(t, y, 'b:');  
end
```



Şekil 4.22  $p = 1$  (sürekli),  $p = 2$  (kesikli),  $p = 3$  (noktalı) vekil model eğrileri ve veri havuzu noktaları.



#### 4.5 Kriging yöntemi

Kriging yöntemi, çıkış kaynağı itibari ile yerbilimleri disiplinine ait istatistiksel<sup>1</sup> bir yöntemdir. Bir önceki bölümde üzerinde durulan mesafe ağırlıklı modelleme yaklaşımı ile benzerlikler göstermekle beraber temel bazı farklılıklar da içerir. Diğer yöntemde olduğu gibi Kriging yöntemi de yerel ve veri havuzu noktaları itibari ile hatasız bir ara değer kestirim yöntemidir. Ancak diğerinden farkı, ilkinin deterministik, Kriging yönteminin ise istatistiksel bir yöntem olmasıdır. Yöntemin kökeni Güney Afrikalı bir maden mühendisi olan D. G. Krige'nin çalışmalarına dayanır. Bulunduğu Witwatersrand bölgesindeki altın madeni arama çalışmalarına katılan Krige'nin asıl amacı, az sayıdaki kazı yerlerinin konumları ve o konuma ait altın miktarı verileri ile altın madenin dağılımını tahmin etmek ve bu sayede sağlam bir öngörü ile sondaj çalışmalarını planlamaktır. Krige'ye ait 1951 tarihli **“Witwatersrand bölgesindeki maden**

<sup>1</sup>Geostatistics



çalışmaları ve ilişkili sorunlara ait istatistiksel yaklaşım<sup>i</sup>” isimli yüksek lisans çalışmasının literatüre kazandırdığı asıl katkı, istatistiksel değerlendirmelerin tahminlemede kullanılması yaklaşımıdır<sup>(32)</sup>. Yöntemin kuramsal altyapısı ise Fransız matematikçi G. Matheron’ın “Bölgeselleştirilmiş değişkenler ve tahminleme<sup>ii</sup>” adlı tez çalışmalarına dayanır<sup>(33)</sup>. Bununla beraber Kiging yönteminin temellendirildiği optimal doğrusal öngörüleme tekniklerini 1930’lu yılların sonlarına veya 1940’lı yılların başlarına kadar götürmek de mümkündür. Örneğin H. Wold’un “Durağan zaman serilerinin analizi üzerine çalışma<sup>iii</sup>” veya A. N. Kolmogorov’un “Durağan rassal serilerin ara ve dış değer kestirimi<sup>iv</sup>” adlı çalışmaları bahse konu erken döneme ait literatür arasında sayılabilirler<sup>(34)</sup>.

Kriging yöntemleri, ifade edildiği üzere bilinen örneklem çiftlerinden hareketle belirli bir konuma ait bilinmeyen çıktı değerinin doğrusal bir yaklaşımla öngörülmesi düşüncesi üzerine temellendirilmektedir. Bu bağlamda elimizde  $n$  tane farklı konum ( $\mathbf{x}_i$ ) ve bu konumlara ait çıktı verisi ( $y_i$ ) çiftinin bulunduğunu düşünelim;

$$y_i = y(\mathbf{x}_i), \quad i = 1, 2, \dots, n$$

Verili bu değerlerden hareketle Kriging yönteminde herhangi bir  $\mathbf{x}$  noktasındaki çıktı değeri;

$$\hat{y}(\mathbf{x}) - \mu(\mathbf{x}) = \sum_{i=1}^n w_i(\mathbf{x}) \{y(\mathbf{x}_i) - \mu(\mathbf{x}_i)\} \quad (4.56)$$

ifadesine göre tanımlanır. Burada  $\mu(\mathbf{x})$  durağan seriye ait çıktı verisi bazlı ortalama değeri,  $w_i(\mathbf{x})$  terimi Kriging katsayısını ve  $\mu(\mathbf{x}_i)$  veri havuzundaki noktalara ait ortalama çıktı değerini temsil etmektedir. Kriging katsayısı, her bir veri havuzu çıktı değeri  $y_i$  için ayrı bir değerdir. Ayrıca söz konusu ağırlık katsayısı yöneyi, her  $\mathbf{x}$  konumu için farklılık arz eder<sup>(35)</sup>. Doğrusal nitelikteki bahse konu denklemde geçen terimlerin ve örneklem uzayının özelliğine göre bilinmeyen niteliğindeki ağırlık katsayılarının hesaplanması için farklı yaklaşımlardan bahsedilebilir. Kriging yöntemleri, genellikle doğrusal ve doğrusal olmayan Kriging yöntemleri olarak tasnif edilir. Doğrusal Kriging yaklaşımları arasında sıradan, basit ve tümel Kriging<sup>v</sup> yöntemleri sayılabilir. Doğrusal olmayan

<sup>i</sup> “A statistical approach to some mine valuations and allied problems on the Witwatersrand”

<sup>ii</sup> “Regionalized variables and their estimation: an application of the theory of random functions to natural sciences”

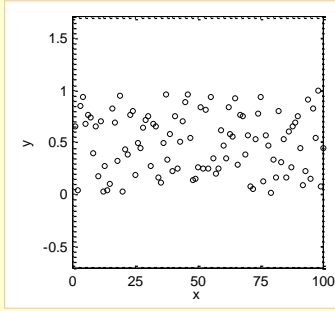
<sup>iii</sup> “A study in the analysis of stationary time series”

<sup>iv</sup> “Interpolation and extrapolation of stationary random sequences”

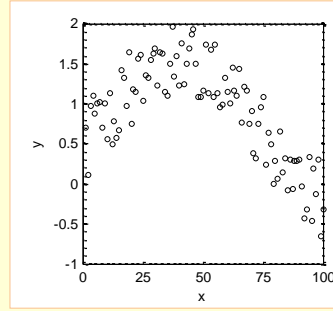
<sup>v</sup> Ordinary, simple, and universal Kriging



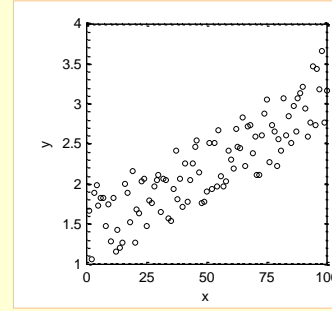
Kriging yöntemleri arasında ayırıcı, gösterge ve lognormal Kriging<sup>i</sup> yöntemlerinden bahsedilebilir. Diğer bir ayırım, çıktı değişkeninin sayısına bağlı olarak yapılan tek veya çok-değişkenli Kriging yöntemleridir. Bu bağlamda çok-değişkenli Kriging yöntemi olarak ko-Kriging yönteminden bahsedilebilir. Literatürde en yaygın olarak kullanılan yöntem, sıradan Kriging yöntemidir<sup>(36)</sup>. Kriging yöntemlerinin açıklanmasına geçmeden önce yöntemlerin dayandırıldığı bazı istatistiksel kavramların hatırlatılmasında fayda mülhaza edilmektedir. Söz konusu istatistikî kavramlar arasında durağanlık, varyans, ortak-varyans ve ilgileşim<sup>ii</sup> sayılabilir.



(a) Durağan seri



(b) Durağan olmayan seri



(c) Durağan olmayan ancak eğilimli seri

Şekil 4.23 Durağanlık kavramına göre veri dağılımları.

**Durağanlık** kavramı, rassal süreçlerin değerlendirilmesinde karşılaşılan bir kavram olup mekânsal veya zamansal serilerin genel özelliğini tanımlar. Buna göre durağan rassal bir süreçte seriye ait istatistikî değerler (örneğin ortalama, varyans vb.) zamana veya mekâna göre değişiklik göstermez. Bu tür veriler durağan seri<sup>iii</sup> olarak adlandırılır. Şekil 4.23(a)'da durağan seri örneği görülmektedir. Buna karşılık Şekil 4.23(b)'de görüldüğü üzere istatistikî verilerin zamana veya mekâna göre değişimi söz konusu olduğunda seri durağan olmayan seri<sup>iv</sup> olarak tanımlanır<sup>(37)</sup>. Diğer taraftan bazı serilerde istatistiksel verilerin değişimi söz konusu olmakla beraber belirli bir

<sup>i</sup> Disjunctive, indicator, and lognormal Kriging

<sup>ii</sup> Stationarity, variance, covariance, and correlation

<sup>iii</sup> Stationary series

<sup>iv</sup> Non-stationary series



eğilim söz konusu olabilir. Bu tür süreçlere durağan olmayan ancak eğilimli süreç<sup>(38)</sup> adı verilir. Nitekim Şekil 4.23(c) seçeneğindeki seride belirli bir artış trendi ve bu eğilim dâhilindeki belirli bir bant aralığında rassal değişimler görülmektedir.

**Varyans** kavramı, aynı değişkene ait farklı ölçüm değerleri arasındaki farkı nitelemek amacıyla tanımlanan bir kavram olup genellikle değişken değeri ile ortalama değer  $\{\mu\}$  arasındaki farka dayandırılır. Ortalama değer, tüm verilerin ortalaması  $\{\mu(x)\}$  olabileceği gibi belirli bir miktardaki örneklem havuzunun ortalama değeri de  $\{\mu(x_i)\}$  olabilir;

$$\sigma^2(x) = \frac{1}{n-1} \sum_{i=1}^n (x_i - \mu(x))^2 \quad (4.57)$$

$$\sigma^2(x_i) = \frac{1}{n-1} \sum_{i=1}^n (x_i - \mu(x_i))^2 \quad (4.58)$$

Bazı tanımlarda paydada  $n$  terimini de görmek mümkündür. Müteakip örnekte iki farklı değişkene ait veri havuzu ortalama ve varyans değerleri görülmektedir;

S/N	$x_i$	$y_i$	$\mu(x_i)$	$\mu(y_i)$
1	20	0.68	$\frac{1}{n} \sum_{i=1}^n x_i$ ↓	$\frac{1}{n} \sum_{i=1}^n y_i$ ↓
2	65	1.81		
3	115	2.12	$\frac{1}{4}(20 + 65 + 115 + 160) = 90$	$\frac{1}{4}(0.68 + 1.81 + 2.12 + 2.48) = 1.7725$
4	160	2.48		

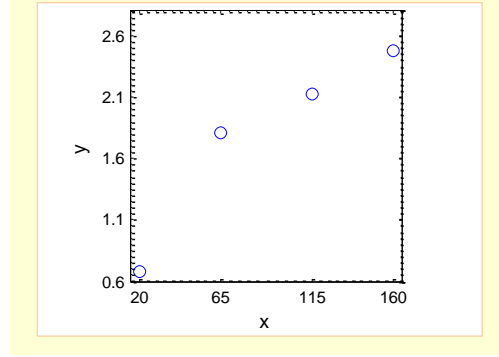
  

$$\sigma^2(x_i) = \frac{1}{3} \left\{ (20 - 90)^2 + (65 - 90)^2 + (115 - 90)^2 + (160 - 90)^2 \right\} = 3683.3$$
  

$$\sigma^2(y_i) = \frac{1}{3} \left\{ (0.68 - 1.7725)^2 + (1.81 - 1.7725)^2 + (2.12 - 1.7725)^2 + (2.48 - 1.7725)^2 \right\} = 0.6054$$

<sup>i</sup>Trend-stationary process





Şekil 4.24 Örnek veri dağılımı.

Varyans değerinin küçük mertebelerde olması, veri dağılımının nisbeten fazla olmadığını, değerlerin belirli bir bölgede yoğunlaştığını gösterir. Aksi durumda, verilerin geniş bir bölgeye yayıldığı ve dağınık bir karaktere sahip olduğu düşünülebilir. Varyans ifadesinin karekökü, literatürde standart sapma olarak tanımlanır;

$$\sigma(x_i) = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \mu(x_i))^2}$$

(4.59)

```
clc; clear
% veri havuzu
xs = [20 65 115 160];
ys = [0.6800 1.8100 2.1200 2.4800];
var(x)
var(y)
```

```
3.6833e+03
0.6054
```



**MATLAB uygulaması**

**Ortak-varyans** ya da diğer adı ile kovaryans kavramı, iki ayrı değişkene ait farklı ölçüm değerlerinin kendi bünyeleri ile aralarındaki değişimi nitelemek amacıyla tanımlanan bir kavramdır. Genellikle  $cov(x, y)$  şeklinde gösterilmekte olup birimi her iki değişkenin birimlerinin çarpımı olarak belirlenir. Ayrıca  $cov(x, x)$  ifadesi esasen  $var(x)$  anlamına gelmektedir. Genel tanımı itibarı ile ortak-varyans denklemi;



$$\text{cov}(x_i, y_i) = E\{(x - E(x))(y - E(y))\} \quad (4.60)$$

şeklinde ifade edilir. Burada  $E\{\dots\}$  terimi, beklenen değer<sup>i</sup> manasında ve genellikle de ortalama değer anlamında kullanılır. Dolayısıyla ortak-varyans denklemi daha açık bir şekilde;

$$\text{cov}(x_i, y_i) = \frac{1}{n-1} \sum_{i=1}^n (x_i - \mu(x_i))(y_i - \mu(y_i)) \quad (4.61)$$

olarak tanımlanabilir. Müteakip çizelgede bir önceki iki farklı değişkene ait veri havuzu ortak-varyans değerinin elde edilişi gözlemlenebilir;

S/N	$x_i$	$y_i$	$x_i - \mu(x_i)$	$y_i - \mu(y_i)$	$(x_i - \mu(x_i))(y_i - \mu(y_i))$	$\text{cov}(x_i, y_i)$
1	20	0.68	20 - 90 = -70	0.68 - 1.7725 = -1.0925	76.4750	$\frac{1}{3}(133.75) = 44.5833$
2	65	1.81	65 - 90 = -25	1.81 - 1.7725 = 0.0375	-0.9375	
3	115	2.12	115 - 90 = 25	2.12 - 1.7725 = 0.3475	8.6875	
4	160	2.48	160 - 90 = 70	2.48 - 1.7725 = 0.7075	49.5250	
$\mu(x_i) \rightarrow 90$ $\mu(y_i) \rightarrow 1.7725$					$\Sigma \rightarrow 133.75$	

Ortak-varyans değerinin pozitif çıkması, değişkenlerin aynı doğrultu dâhilinde benzer davranış (artış ve artış ya da azalış ve azalış) sergilediğini, negatif değer alması ise değişkenlerin birbirlerinin tersine bir davranış (artış ve azalış veya azalış ve artış) sergilediğini gösterir.

```
clc; clear
% veri havuzu
x = [20 65 115 160];
y = [0.6800 1.8100 2.1200 2.4800];
cov(x, y)
```

```
3683.33 44.5833
44.5833 0.60542
```



**MATLAB uygulaması**

**İlgileşim** kavramı, ortak-varyans kavramının bir anlamda boyutsuzlaştırılmış halidir. Her ne kadar ortak-varyans değeri, değişkenlerin değişimi arasındaki ilişkiyi belirginleştirse de birimsel çerçevede çarpım sonucu, fiziksel manada bazen anlam ifade etmeyen birimlerle neticelenebilir. Dolayısıyla boyutsuz bir etkileşim değeri olarak ilgileşim;

<sup>i</sup>Expected value





$$r(x_i, y_i) = \frac{\frac{1}{n-1} \sum_{i=1}^n (x_i - \mu(x_i))(y_i - \mu(y_i))}{\sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \mu(x_i))^2} \sqrt{\frac{1}{n-1} \sum_{i=1}^n (y_i - \mu(y_i))^2}} \quad (4.62)$$

$$r(x_i, y_i) = \frac{cov(x_i, y_i)}{\sigma(x_i)\sigma(y_i)} \quad (4.63)$$

şeklinde tanımlanır. Müteakip çizelgede bir önceki veri havuzuna ait ilgileşim değerinin elde edilişi gözlemlenebilir;

S/N	$x_i$	$y_i$	$cov(x_i, y_i)$	$\sigma(x_i)$	$\sigma(y_i)$	$r(x_i, y_i)$
1	20	0.68				
2	65	1.81				
3	115	2.12	44.5833	60.96	0.7780	$\frac{44.5833}{60.96 \times 0.7780} = 0.94$
4	160	2.48				

İlgileşim değer aralığı  $-1$  ile  $+1$  arası değişebilmekte olup pozitif değerler doğru orantıyı, negatif değerler ters orantıyı gösterir. Değerin sıfır civarında olması, aradaki zayıf ya da olmayan ilişkiyi gösterir.

```
clc; clear
% veri havuzu
x = [20 65 115 160];
y = [0.6800 1.8100 2.1200 2.4800];
corr(x', y')
```

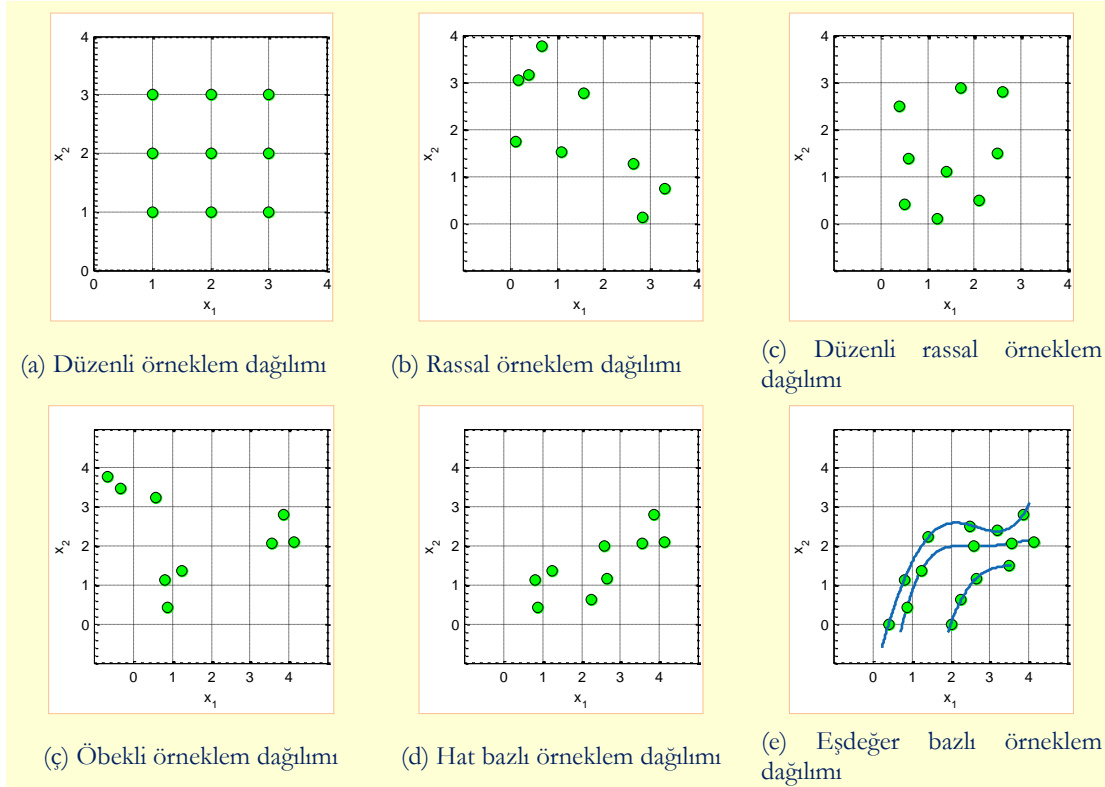
0.9441



MATLAB uygulaması

Diğer taraftan Kriging yönteminde her ne kadar veri havuzu dâhilindeki noktaların dağılımı rassal karakterde olsa da örneklem uzayı noktalarının dağılımı bağlamında bazı ilave düzenlemelere gidilebilir. Aslında Kriging yönteminde rassallıktan kastedilen, örneklem noktalarındaki değerlerin doğrudan ölçülerek kayıt altına alınması ve elde edilen verilerin değerlendirme sürecindeki yaklaşımıdır. Dolayısıyla örneklem uzayının seçiminde belirli bir yaklaşımın sergilenmesinin, rassal karakteri zedeleyici bir etki taşımadığı söylenebilir.





Şekil 4.25 Örnek örneklem uzayı veri dağılımları<sup>(39)</sup>.

Konunun daha iyi anlaşılabilmesi için Şekil 4.25'te sunulan örneklem noktaları dağılımları değerlendirilebilir. Söz konusu seçeneklerden ilki, düzenli örneklem<sup>i</sup> dağılımı olup böyle bir dağılımda örneklem noktaları belirli mesafe aralıkları ile düzenli olarak veri uzayına yayılırlar. Rassal örneklem<sup>ii</sup> dağılımında örneklem noktalarının dağılımı tamamen rassaldır. Düzenli rassal örneklem<sup>iii</sup> dağılımında, belirli bölmeler içerisinde bir rassallık söz konusu olup her bölme bir

<sup>i</sup> Regular sampling

<sup>ii</sup> Random sampling

<sup>iii</sup> Stratified random sampling





rassal konum gelecek şekilde dağılım öngörülebilir. Öbekli örnekleme dağılımı yaklaşımında, örneklem noktaları belirli bölgelerde yoğunlaşacak şekilde dağıtılırlar. Hat bazlı örnekleme dağılımında, örneklem noktaları belirli bir hat ve o hattı merkez alan bant aralığında yayılırlar. Son seçenek niteliğindeki eşdeğer bazlı örnekleme dağılımında ise örneklem noktaları aynı eşdeğere sahip noktalar arasından seçilirler. Örneklem dağılımı, tüm veri havuzunun dağılımında dikkate alınmaktan ziyade yerleştirme çerçevesinde düşünülmelidir. Zira diğer ara değer kestirim yöntemlerinde olduğu üzere Kriging yönteminin temelinde de hedef noktaya yakın yerel verilerden hareketle kestirim öngörüsü yatmaktadır.

#### 4.5.1 Sıradan Kriging yöntemi

Hatırlanacağı üzere en yaygın ve sıkça kullanılan yöntem sıradan Kriging yöntemiymiş. Yöntemin detaylarına geçmeden önce, yapılan bazı ön kabullerin ifade edilmesi gerekir. Sıradan Kriging yöntemine göre eldeki veri havuzuna ait  $y(\mathbf{x}_i)$  değerlerinin normal dağılıma haiz oldukları ve çıktı değerleri arasındaki fark ile diğer istatistikî değerlerin sadece konuma göre değiştiği kabul edilir. Ayrıca durağan seriye ait ortalama çıktı değeri yerine, örneklem veri havuzu çıktı ortalama değeri dikkate alınır;  $\mu(\mathbf{x}) \rightarrow \mu(\mathbf{x}_i)$ . Diğer taraftan genel Kriging tahminleme denklemi tekrar dikkate alınır;

$$\hat{y}(\mathbf{x}) - \mu(\mathbf{x}) = \sum_{i=1}^n w_i(\mathbf{x}) \{y(\mathbf{x}_i) - \mu(\mathbf{x}_i)\} \quad (4.64)$$

ifadesi

$$\hat{y}(\mathbf{x}) = \sum_{i=1}^n w_i(\mathbf{x}) \{y(\mathbf{x}_i) - \mu(\mathbf{x}_i)\} + \mu(\mathbf{x}) \quad (4.65)$$

$$\mu(\mathbf{x}) \rightarrow \mu(\mathbf{x}_i) \quad (4.66)$$

$$\hat{y}(\mathbf{x}) = \sum_{i=1}^n w_i(\mathbf{x}) y(\mathbf{x}_i) + \mu(\mathbf{x}_i) \left(1 - \sum_{i=1}^n w_i(\mathbf{x})\right) \quad (4.67)$$

şeklinde tekrar yazılabilir. Ayrıca herhangi bir  $\mathbf{x}$  noktasına ait Kriging tahmini ile gerçek değer arasındaki fark;

<sup>i</sup> Cluster or nested sampling

<sup>ii</sup> Transect sampling

<sup>iii</sup> Contour sampling



$$\epsilon(\mathbf{x}) = \hat{y}(\mathbf{x}) - y(\mathbf{x}) \quad (4.68)$$

şeklinde ifade edilebilir. Kriging yaklaşımında, bu ifade üzerine dayalı hatanın minimizasyonu temelli eniyileme süreci tanımlanır;

$$\min_{\mathbf{x}} E\{\epsilon(\mathbf{x})^2\} \quad (4.69)$$

Tanımlanan bahse konu kısıtsız optimizasyon problemine ilave olarak sıradan Kriging yönteminin belirgin özelliğini oluşturan diğer bir unsur, ağırlık katsayılarının normalizasyonudur<sup>i</sup>. Buna göre;

$$\sum_{i=1}^n w_i(\mathbf{x}) = 1 \quad (4.70)$$

koşulu dikkate alınırsa eniyileme sorunu;

$$\begin{aligned} & \min_{\mathbf{x}} E\{\epsilon(\mathbf{x})^2\} \\ \text{Kısıt} & \sum_{i=1}^n w_i(\mathbf{x}) = 1 \end{aligned} \quad (4.71)$$

biçimindeki kısıtlı optimizasyon problemine dönüştürülmüş olur. Bazı çalışmalarda aynı sorun farkların varyansının minimizasyonu şeklindeki;

$$\begin{aligned} & \min_{\mathbf{x}} \text{var}(\hat{y}(\mathbf{x}) - y(\mathbf{x})) \\ \text{Kısıt} & E\{\hat{y}(\mathbf{x}) - y(\mathbf{x})\} = 0 \end{aligned} \quad (4.72)$$

formunda da tanımlanabilmekte olup esasen aynı denklemlerin farklı gösterimleridir. Kısıtın, sorunun çözümünde etkisi olduğu gibi tahminleme denkleminde de yansması söz konusudur. Zira koşul kapsamında Kriging tahmini denklemi;

$$\hat{y}(\mathbf{x}) = \sum_{i=1}^n w_i(\mathbf{x})y(x_i) \quad (4.73)$$

<sup>i</sup>Unbiasedness or non-bias condition



şeklindeki halini almış olur. Diğer taraftan tanımlanan söz konusu optimizasyon problemi Lagrange çarpanları yaklaşımı ile çözümlenerek Kriging katsayıları;

$$\mathbf{w} = \Gamma^{-1}\boldsymbol{\gamma} \quad (4.74)$$

$$\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \\ \lambda \end{bmatrix} = \begin{bmatrix} \gamma(x_1, x_1) & \gamma(x_1, x_2) & \cdots & \gamma(x_1, x_n) & 1 \\ \gamma(x_2, x_1) & \gamma(x_2, x_2) & \cdots & \gamma(x_2, x_n) & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \gamma(x_n, x_1) & \gamma(x_n, x_2) & \cdots & \gamma(x_n, x_n) & 1 \\ 1 & 1 & \cdots & 1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} \gamma(x_1, \mathbf{x}) \\ \gamma(x_2, \mathbf{x}) \\ \vdots \\ \gamma(x_n, \mathbf{x}) \\ 1 \end{bmatrix} \quad (4.75)$$

denklemine göre elde edilebilir. Burada  $\lambda$  değeri Lagrange çarpanını,  $\gamma(x_i, x_j)$  ise varyogram<sup>i</sup> değerini temsil etmektedir<sup>(40)</sup>.

Varyogram kavramı, konuma göre çıktı değerlerinin nasıl değiştiğini ve bir manada konumsal bağlılığı gösteren önemli bir fonksiyon olup;

$$2\gamma(x_i, x_j) = \text{var}(y(x_i) - y(x_j)) \quad (4.76)$$

şeklinde tanımlanır. Uygulamada genellikle elde edilen varyogram değerlerinin, konumlar arası mesafeye bağlı olarak grafik şeklinde gösterimi tercih edilir. Hatta G. Matheron'un tanımladığı üzere kuramsal varyogramın;

$$2\gamma(h) = \text{var}(y(\mathbf{x} + \mathbf{h}) - y(\mathbf{x})) \quad (4.77)$$

$$\mathbf{h} = \mathbf{x}_i - \mathbf{x}_j \quad (4.78)$$

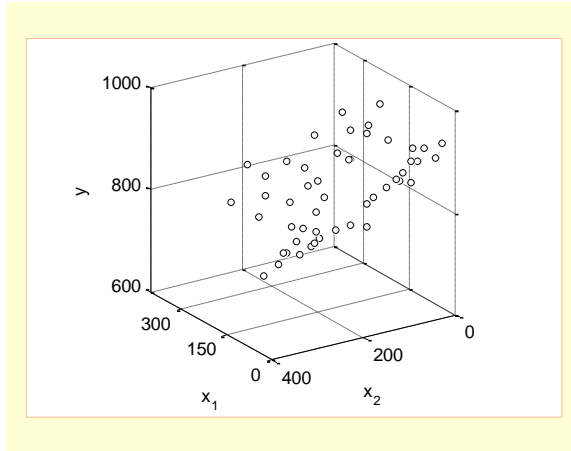
şeklindeki mesafeye bağlı formu ile kullanımı yaygındır. Varyogramın oluşturulma safhalarını; bilinen nokta çiftleri arasındaki Euclidian mesafelerinin hesaplanması, aynı nokta çiftlerine ait benzerlik değerleri bağlamında çıktı değerlerinin farklarının karelerinin alınması ve nokta çiftleri için yatay eksen mesafe olan, dikey eksen ise benzerlik değeri olan grafiğin eldesi şeklinde sıralayabiliriz<sup>(41)</sup>. Bununla beraber özellikle verinin çok olduğu durumlarda genellikle veri bulutu ile karşılaşılır. Bahsedilen veri bulutu hususunun daha iyi anlaşılabilmesi için müteakip konum ve çıktı değeri verilerini<sup>(42)</sup> dikkate alabiliriz;

<sup>i</sup> Variogram, semi-variogram, veya semi-variance



$x_1$	0	55	105	165	270	65	130	155	155	225	250	295	30	100	110	135	160	190	230	300	30	
	7	105	170	210	245	300	0	85	175	300	15	60	90	90	140	210	260	270	295	5	55	55
	90	100	140	190	255	285	270															
$x_2$	305	310	305	310	310	260	255	265	285	280	250	260	210	240	225	225	225	230	210	215		
	160	190	190	175	160	160	170	120	135	115	110	85	90	90	55	55	90	85	50	50	25	
	30	5	35	15	0	40	20	5	150													
$y$	870	793	755	690	800	800	730	728	710	780	804	855	813	762	765	740	765	760	790	820		
	855	812	773	812	827	805	840	890	820	873	875	873	865	841	862	908	855	850	882	910		
	940	915	890	880	870	880	960	890	860	830												

Toplamda elli konum ve bu konuma ait çıktı değerlerini kapsayan veri havuzunun grafiksel görünümü müteakip şekilde görülmektedir;



Şekil 4.26 Örnek örneklem uzayı veri dağılımı.

Buna göre öncelikle verilerin konumları arasındaki mesafelerin hesaplanması gerekir. Örneğin ilk konumla ikinci konum arası mesafe değeri;

$$\begin{aligned} \mathbf{x}_1 &\rightarrow (x_{11}, x_{12}) \rightarrow (0, 305) \\ \mathbf{x}_2 &\rightarrow (x_{21}, x_{22}) \rightarrow (55, 310) \end{aligned}$$

$$\begin{aligned} h_{1,2} &\rightarrow \sqrt{(0 - 55)^2 + (305 - 310)^2} \\ h_{1,2} &= 55.2268 \end{aligned}$$

şeklinde elde edilebilir.

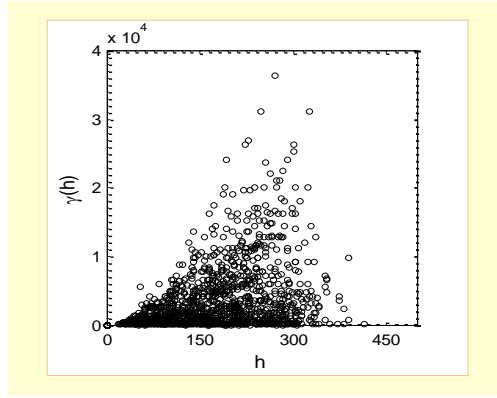




Benzer işlemler diğer konumlar arası mesafeler için de uygulanabilir. İkinci aşamada, konumlara ait çıktı değerlerinin varyansı üzerine dayalı yarı-varyans değerleri hesaplanabilir. Örneğin ilk konumla ikinci konum arasındaki yarı-varyans değeri;

$$\begin{aligned} y_1 &\rightarrow 870 \\ y_2 &\rightarrow 793 \\ \gamma(h_{1,2}) &\rightarrow \gamma(55.2268) = \frac{1}{2}(870 - 793)^2 = 2964.5 \end{aligned}$$

olarak elde edilebilir. Benzer işlemler diğer konumlar arası çıktı değerleri için de uygulanabilir. Konumlar arası mesafe ve bu mesafelere karşılık gelen çıktı değeri değişimlerinin grafiksel gösterimi müteakip şekilde görülmektedir.



Şekil 4.27 Örnek varyogram grafiği.

Şekilden de görüleceği üzere konumlar arası mesafenin değişimine göre çıktı değerinin değişimini gösteren varyogram değerleri, anlamlı bir grafikten ziyade bir veri bulutu hüviyetinde olup böyle bir veri bulutundan manidar çıkarılmalar yapmak zordur. Dolayısıyla grafiğin daha anlamlı hale getirilmesi amacıyla nokta çiftlerinin gruplanması ve grafiğin seyreltilmesi yoluna gidilir. Bu bağlamda varyogram değeri de;

$$\hat{\gamma}(\hat{h}) = \frac{1}{2|N(\hat{h})|} \sum_{i,j \in N(\hat{h})} (y(x_i) - y(x_j))^2 \quad (4.79)$$



$$\hat{h} = \frac{1}{|N(h)|} \sum_{i \in N(h)} h_i \quad (4.80)$$

ifadesine göre hesaplanır. Burada  $\hat{h}$  gruplama dâhilinde birbirine yakın mesafe değerlerinin ortalamasını,  $N(h)$  gruplamada dikkate alınan veri çiftleri setini ve  $\hat{\gamma}(\hat{h})$  terimi ortalaması alınmış deneysel varyogram değerini göstermektedir.

Varyogram grafiği eldesi.



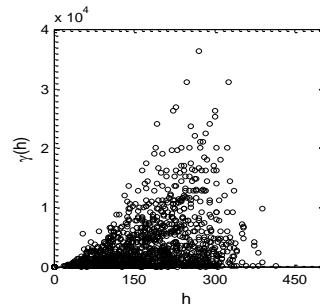
**MATLAB uygulaması**



```
% x1 = data('x1'); x2 = data('x2'); y = data('y')
clc; clear
% mesafe hesabı
for i = 1:length(x1)
    for j = i:length(x2)
        d(i,j) = sqrt((x1(i)-x1(j))^2+(x2(i)-x2(j))^2) ;
    end
end
% çıktı farkı hesabı
for i = 1:length(x1)
    for j = i:length(x2)
        g(i,j) = 0.5*((y(i)-y(j))^2) ;
    end
end
% varyogram çizimi
plot(d,g,'ko','MarkerSize',2)
axis square
```



Sonuç





```

        d(i,j) = sqrt((x1(i)-x1(j))^2+(x2(i)-x2(j))^2) ;
    end
end

% çıktı farkı hesabı
for i = 1:length(x1)
    for j = i:length(x2)
        g(i,j) = 0.5*(y(i)-y(j))^2 ;
    end
end

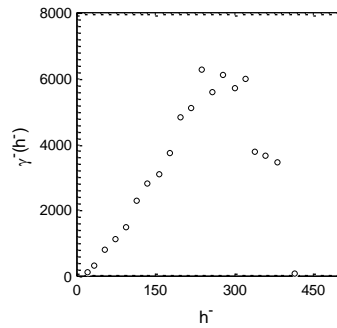
nnd = 20 ; % Aralık sayısı
maxd = max(max(d)) ;
dd = maxd/nnd ;
k = 1 ;

for i = 0:dd: maxd-dd
    [r c] = find(i<d & d<i+dd)
    topd = 0; topg = 0;
    for j = 1:length(r)
        topd = topd + d(r(j),c(j));
        topg = topg + g(r(j),c(j));
    end
    nd(k) = topd/length(r) ;
    ng(k) = topg/length(r) ;
    k = k + 1 ;
end
plot(nd,ng,'ko')
axis square

```

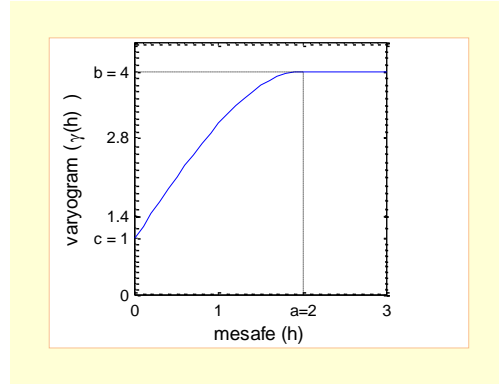


### Sonuç





Varyogram grafiği, noktalar arası mesafeye göre çıktı değerinin nasıl değiştiği hakkında bize önemli bilgiler vermektedir. Ancak bu grafik bilinen noktalardan hareketle elde edilmiş olup ara değerler ile ilgili herhangi bir veri eldesi mümkün değildir. Dolayısıyla herhangi bir  $x$  noktasının söz konusu olduğu durumda tahminleme için gerekli olan  $\gamma(x_i, x), i = 1, 2, \dots, n$  değerlerinin elde edilmesi için analitik bir yaklaşım gerekir. Daha açık bir ifade ile eldeki seyreltilmiş varyogram diyagramı kullanılarak bir manada eğri uydurma yaklaşımı ile parametrik varyogram diyagramının elde edilmesi mümkündür. Söz konusu süreçte genellikle en küçük kareler yaklaşımı bazlı uygun bir eniyileme algoritması kullanılır.



Şekil 4.29 Genel hali ile parametrik varyogram grafiği.

Parametrik varyogramın eldesi sürecinde bazı kavramların hatırlatılmasında fayda mülhaza edilmektedir. Bahse konu kavramlar cevher etkisi, menzil, eşik ve kısmi eşiktir<sup>i</sup>. Şekil 4.29'da bahse konu kavramlara örnek değerler görülmektedir. Buna göre mesafe değerinin sıfır veya sıfıra çok yakın olması halinde bile noktalar arası çıktı değerlerinde farklılıklar olabilir. Normal şartlar altında iki nokta arası mesafenin sıfır veya sıfıra çok yakın olması durumunda noktalar arası çıktı değerlerinde herhangi bir farkın olmaması, dolayısıyla eğrinin merkezden geçmesi beklenir. Ancak bazen birbirine çok yakın iki nokta arasında bile çıktı değeri açısından büyük farklılıklar olabilir. Söz konusu durum Kriging yönteminin çıkış kaynağı olan altın madenlerinde sıkça karşılaşıldığı için böyle bir etkiye cevher etkisi ( $h = 0, \gamma(h) \rightarrow c$ ) adı verilmektedir<sup>(43)</sup>. Menzil kavramı, mesafenin etkisinin çok azaldığı veya kaybedildiği değeri göstermektedir. Örnekte görüldüğü üzere mesafe değerinin ( $h = a, \gamma(h) \rightarrow b$ ) menzil değerine ulaşması halinde varyogram

<sup>i</sup> Nugget effect, range, sill, and partial sill





değerinde değişiklik görülmemektedir. Bu arada menzil mesafesinde varyogramın bir manada yakınsadığı değer, eşik değer olarak adlandırılmakta, eşik değer ile cevher değeri arası fark ise kısmi eşik ( $b - c$ ) olarak tanımlanmaktadır. Uygulamalarda değişik varyogram dağılımları ve bu dağılımlara uygun farklı parametrik varyogram eğri düzenlemeleri mevcuttur. Müteakip Çizelge 4.2'de cevher etkisini gözardı eden ve yaygın olarak kullanılan bazı varyogram eğri düzenlemeleri görülmektedir.

Tanımı & Denklemi	Şekli
<p>Cevher etkisi modeli<sup>i</sup></p> $\gamma(h) = \begin{cases} 0 &  h  = 0 \\ b & h > 0 \end{cases}$	
<p>Sınırlı doğrusal model<sup>ii</sup></p> $\gamma(h) = \begin{cases} b \left( \frac{h}{a} \right) & 0 \leq h \leq a \\ b & h > a \end{cases}$	
<p>İkilenik model<sup>iii</sup></p> $\gamma(h) = \begin{cases} 2 \left( \frac{h}{a} \right) - \left( \frac{h}{a} \right)^2 & 0 \leq h \leq a \\ b & h > a \end{cases}$	

<sup>i</sup> Nugget effect model

<sup>ii</sup> Bounded linear model

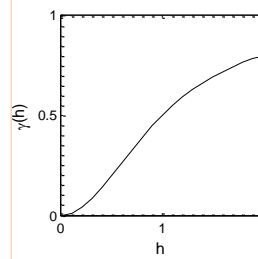
<sup>iii</sup> Quadratic model





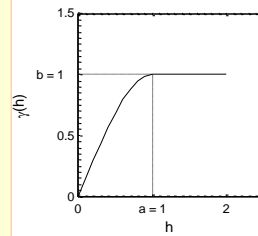
Oransal ikilenik model<sup>i</sup>

$$\gamma(h) = k \frac{\left(\frac{h}{a}\right)^2}{1 + \left(\frac{h}{a}\right)^2} \quad h \geq 0$$



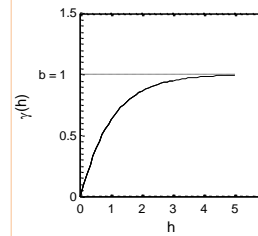
Küresel model<sup>ii</sup>

$$\gamma(h) = \begin{cases} b \left\{ \frac{3}{2} \left(\frac{h}{a}\right) - \frac{1}{2} \left(\frac{h}{a}\right)^3 \right\} & 0 \leq h \leq a \\ b & h > a \end{cases}$$



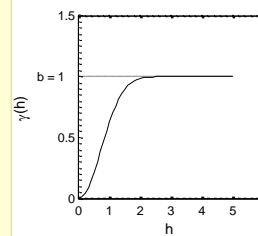
e-üssel model<sup>iii</sup>

$$\gamma(h) = b \left\{ 1 - e^{-\frac{h}{a}} \right\} \quad h \geq 0$$



Gauss modeli<sup>iv</sup>

$$\gamma(h) = b \left\{ 1 - e^{-\left(\frac{h}{a}\right)^2} \right\} \quad h \geq 0$$



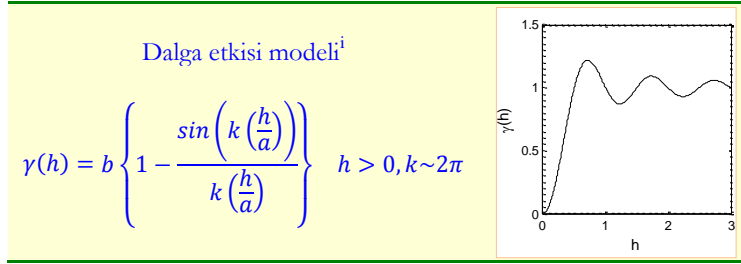
<sup>i</sup> Rational quadratic model

<sup>ii</sup> Spherical model

<sup>iii</sup> Exponential model

<sup>iv</sup> Gaussian model



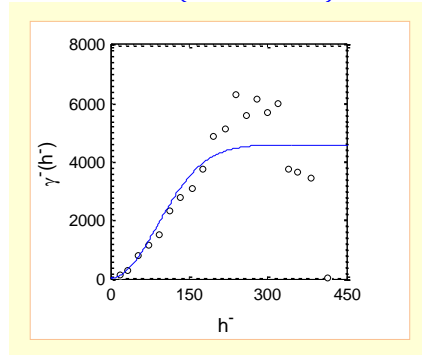


Çizelge 4.2 Parametrik varyogram modelleri

Daha önce elde edilen seyreltilmiş varyogram noktaları için örnek bir eğri uydurma yaklaşımı, müteakip şekildeki gibi yapılabilir. Buna göre varyogram değerlerinin dağılımına en uygun eğrinin Gauss eğrisi modeli olduğu gözlemlenebilir.

Parametrik model

$$\gamma(h) = 4582.6 \left\{ 1 - e^{-\left(\frac{h}{125.7}\right)^2} \right\} \quad h \geq 0$$



Şekil 4.30 Seyreltilmiş ve eğri uydurulmuş varyogram grafiği.

Noktalar arası mesafe ve konumdan ziyade mesafeye bağlı çıktı değeri değişimi içerikli varyogram grafikleri, bazı çalışmalarda her-yönlü varyogram<sup>ii</sup> olarak da isimlendirilir. Bununla

<sup>i</sup> Wave hole effect model

<sup>ii</sup> Omni-directional variogram



beraber mesafe verisine ilave olarak, noktalar arası konum veya daha açık bir ifade ile yön bilgisinin de dikkate alındığı yönsel varyogram grafikleri elde edilebilir. Çıktı değerlerinin tüm yönlerde aynı şekilde değişim gösterdiği verilere yönsemez, yöne göre değişim gösterdiği verilere ise yönser<sup>ii</sup> veri adı verilir<sup>(44)</sup>. Literatürde konuma göre veri değişiminin bir göstergesi olarak varyogram kavramı yanında madogram ve rodogram gibi ilave tanımlara da rastlanabilmektedir. Madogram değeri varyogram değerine benzer şekilde;

$$\hat{\gamma}(h) = \frac{1}{2|N(h)|} \sum_{i,j \in N(h)} |y(x_i) - y(x_j)| \quad (4.81)$$

denkleminde göre hesaplanır. Görüldüğü üzere varyogram ile arasındaki yegâne değişiklik, çıktı değer farkının karesi yerine mutlak değerinin alınmasıdır. Rodogram değeri ise madogram değerinin karekökünün alınması ile elde edilir;

$$\hat{\gamma}(h) = \frac{1}{2|N(h)|} \sum_{i,j \in N(h)} |y(x_i) - y(x_j)|^{1/2} \quad (4.82)$$

Gerek madogram ve gerekse de rodogram hesaplamaları, genellikle varyogram analizi çalışmalarına destek unsuru olarak değerlendirilir. Ampirik varyogram eğrisinin eldesi sonrasında Kriging katsayılarının hesabı işlemine kolayca geçilebilir. Bu aşamada yapılacak işlem, veri havuzu noktaları arasındaki mesafe verileri ile  $x$  noktasına göre mesafe değerlerinin dikkate alınması ve varyogram değerlerinin deneysel varyogram eğrisine göre hesaplanması, sonrasında ise tahminleme yapılmasıdır.



Şekil 4.31 dâhilinde görüldüğü üzere elimizde coğrafi araziye ait örneklem niteliğinde normalize edilmiş sekiz konum ve bu konumlara ait yükselti verisinin olduğunu düşünelim. Eldeki veri havuzundan hareketle bölgenin diğer yükselti verilerini, sıradan Kriging yöntemi ile elde etmeye çalışalım.



Örnek uygulama

<sup>i</sup> Directional variogram

<sup>ii</sup> Yönsemez kavramı “isotropic”, yönser kavramı ise “anisotropic”



Sıradan Kriging yöntemi kapsamında yapılması gereken ilk işlem, konumlar arası mesafe değerlerinin hesaplanmasıdır. Hatırlanacağı üzere her bir noktanın diğer noktalar ile arasındaki mesafe değeri;

$$h_{i,j} = \sqrt{(x_{i,1} - x_{j,1})^2 + (x_{i,2} - x_{j,2})^2} \quad \left| \begin{array}{l} i=1,2,\dots,8 \\ j=i+1,\dots,8 \end{array} \right.$$

$$h_{i,j} = 0, i = j$$

denklemine göre hesaplanabilmekteydi. Buna göre örneğin 1'inci nokta ile 2'nci nokta arası mesafe;

$$h_{1,2} = \sqrt{(1-2)^2 + (3-2)^2}$$

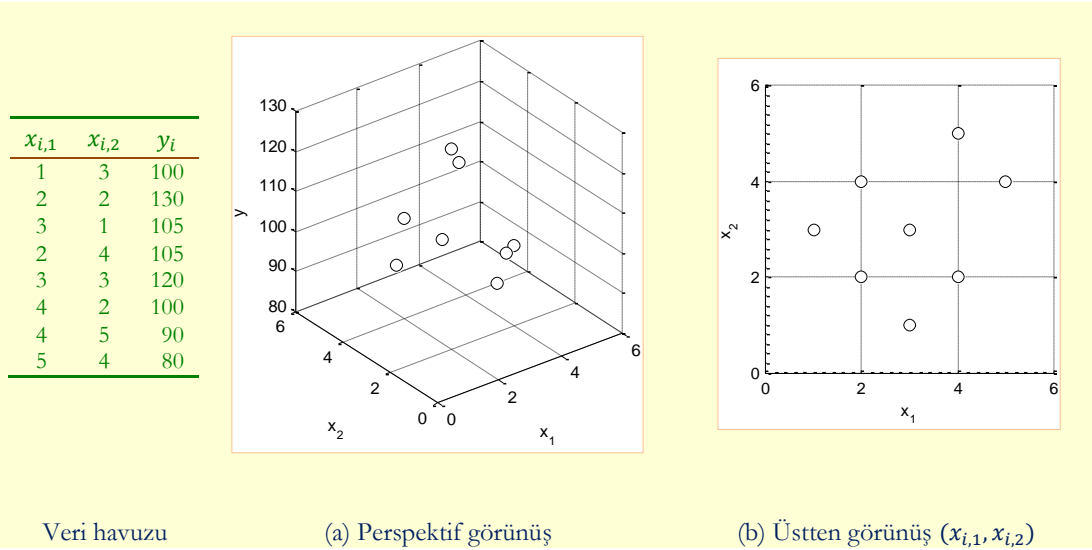
$$h_{1,2} = 1.4142$$

1'inci nokta ile 3'üncü nokta arası mesafe;

$$h_{1,3} = \sqrt{(1-3)^2 + (3-1)^2}$$

$$h_{1,3} = 2.8284$$

olarak elde edilebilir.



Şekil 4.31 Örnek veri havuzu dağılımı ve değerleri.



Benzer işlemler diğer nokta çiftleri için de tekrar edildiğinde müteakip simetrik mesafe matrisini elde etmek mümkündür;

$$H = \begin{bmatrix} 0 & 1.4142 & 2.8284 & 1.4142 & 2.0000 & 3.1623 & 3.6056 & 4.1231 \\ 1.4142 & 0 & 1.4142 & 2.0000 & 1.4142 & 2.0000 & 3.6056 & 3.6056 \\ 2.8284 & 1.4142 & 0 & 3.1623 & 2.0000 & 1.4142 & 4.1231 & 3.6056 \\ 1.4142 & 2.0000 & 3.1623 & 0 & 1.4142 & 2.8284 & 2.2361 & 3.0000 \\ 2.0000 & 1.4142 & 2.0000 & 1.4142 & 0 & 1.4142 & 2.2361 & 2.2361 \\ 3.1623 & 2.0000 & 1.4142 & 2.8284 & 1.4142 & 0 & 3.0000 & 2.2361 \\ 3.6056 & 3.6056 & 4.1231 & 2.2361 & 2.2361 & 3.0000 & 0 & 1.4142 \\ 4.1231 & 3.6056 & 3.6056 & 3.0000 & 2.2361 & 2.2361 & 1.4142 & 0 \end{bmatrix}$$

Mesafe matrisinin eldesi sonrasında varyogram değerlerinin hesaplanması işlemine geçilir. Yine hatırlanacağı üzere herhangi bir mesafe değeri için varyogram değeri;

$$2\gamma(\mathbf{x}_i, \mathbf{x}_j) = \text{var}(y(\mathbf{x}_i) - y(\mathbf{x}_j))$$

$$\gamma_{i,j} = 0, \quad i = j$$

veya daha açık hali ile;

$$\gamma_{i,j} = \frac{1}{2}(y_i - y_j)^2$$

ifadesine göre elde edilmektedir. Bu bağlamda örneğin 1'inci nokta ile 2'nci nokta arası mesafeye karşılık gelen varyogram değeri;

$$\gamma_{1,2} = \frac{1}{2}(100 - 130)^2 = 450$$

1'inci nokta ile 3'üncü nokta arası mesafeye karşılık gelen varyogram değeri;

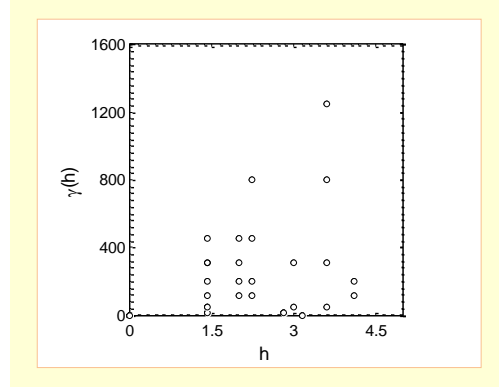
$$\gamma_{1,3} = \frac{1}{2}(100 - 105)^2 = 12.5$$

olarak elde edilebilir. Benzer işlemler diğer nokta çiftleri için tekrar edildiğinde müteakip simetrik varyogram matrisini ve akabinde de mesafe-varyogram değeri grafiğini elde etmek mümkündür. Şekil 4.32'den de gözlemlenebileceği üzere az sayıda veri bilgisi olmasına rağmen elde edilen veri çiftlerinin dağılımı anlamlı bir grafik çizmemektedir.



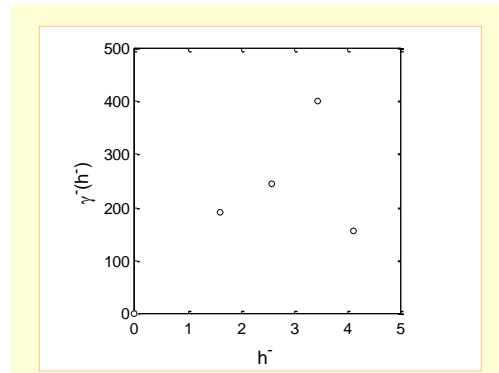


$$\gamma = \begin{bmatrix} 0 & 450 & 12.5 & 12.5 & 200 & 0 & 50 & 200 \\ 450 & 0 & 312.5 & 312.5 & 50 & 450 & 800 & 1250 \\ 12.5 & 312.5 & 0 & 0 & 112.5 & 12.5 & 112.5 & 312.5 \\ 12.5 & 312.5 & 0 & 0 & 112.5 & 12.5 & 112.5 & 312.5 \\ 200 & 50 & 112.5 & 112.5 & 0 & 200 & 450 & 800 \\ 0 & 450 & 12.5 & 12.5 & 200 & 0 & 50 & 200 \\ 50 & 800 & 112.5 & 112.5 & 450 & 50 & 0 & 50 \\ 200 & 1250 & 312.5 & 312.5 & 800 & 200 & 50 & 0 \end{bmatrix}$$



Şekil 4.32 Varyogram grafiği.

Mevcut grafiğin anlamlı hale getirilmesi kapsamında seyreltme işlemlerine geçilebilir. Bu bağlamda mesafe eksenini 1'er birimlik aralıklara bölünebilir ve gerek mesafe için gerekse de varyogram değerleri için ortalama değerler alınabilir. Müteakip şekilde seyreltilmiş varyogram ve akabindeki çizelgede mesafe aralıkları ve bu aralıklara denk gelen değerler ile ortalama değer tespitleri görülmektedir;



Şekil 4.33 Seyreltilmiş varyogram grafiği.



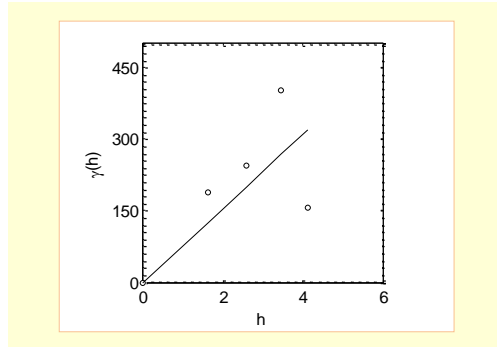




Seyreltilmiş varyogram dağılımının gözlemlenmesi neticesinde doğrusal bir yaklaşımın yapılabileceği, uygun bir eniyileme süreci sonunda (örneğin tekil doğrusal regresyon yöntemi) Şekil 4.34'deki gibi bir denklemin elde edilebileceği görülebilir;

Grup	$h$	$\hat{h}$	$ N(h) $	$\gamma$	$\hat{\gamma}$
$0 < h \leq 1$	–	–	–	–	–
$1 < h \leq 2$	1.41, 1.41, 2, 1.41, 2, 1.41, 2, 2, 1.41, 1.41, 1.41, 1.41	1.607	12	450, 450, 312.5, 312.5, 12.5, 50, 12.5, 200, 200, 50, 112.5, 112.5	189.58
$2 < h \leq 3$	2.83, 2.83, 2.24, 2.24, 3, 3, 2.24, 2.24,	2.575	8	12.5, 12.5, 112.5, 450, 50, 312.5, 800, 200	243.75
$3 < h \leq 4$	3.16, 3.16, 3.6, 3.6, 3.6, 3.6,	3.45	6	0, 0, 50, 800, 1250, 312.5	402.08
$h > 4$	4.12, 4.12	4.12	2	112.5, 200	156.25

Parametrik model  
 $\gamma(h) = 77.7957h \quad h \geq 0$



Şekil 4.34 Seyreltilmiş ve eğri uydurulmuş varyogram grafiği.

DeneySEL varyogramın eldesi neticesinde, Kriging katsayılarının bulunması için gerekli katsayılar matrisinin yeniden hesaplanması mümkündür. Hatırlanacağı üzere ağırlık katsayı değerleri;

$$\mathbf{w} = \Gamma^{-1}\boldsymbol{\gamma}$$

ifadesine göre belirlenmekteydi. Buradaki  $\Gamma$  katsayılar matrisi değerlerini, bilinen mesafe değerlerini deneySEL varyogram denkleminde yerine koyarak kolayca elde edebiliriz. Örneğin  $h_{1,2} = 1.41$  mesafe değeri için deneySEL varyogram değeri;



$$\gamma(1.41) = 77.7957 \times 1.4142 = 110.019$$

veya  $h_{1,2} = 2.83$  mesafe değeri için deneysel varyogram değeri;

$$\gamma(2.83) = 77.7957 \times 2.8284 = 220.03$$

olarak kolayca elde edilebilir. Diğer mesafe değerleri için de benzer işlemler yapıldığında müteakip katsayılar matrisine ulaşılabilir;

$$\Gamma = \begin{bmatrix} 0 & 110.0197 & 220.0395 & 110.0197 & 155.5914 & 246.0116 & 280.4964 & 320.7599 & 1 \\ 110.0197 & 0 & 110.0197 & 155.5914 & 110.0197 & 155.5914 & 280.4964 & 280.4964 & 1 \\ 220.0395 & 110.0197 & 0 & 246.0116 & 155.5914 & 110.0197 & 320.7599 & 280.4964 & 1 \\ 110.0197 & 155.5914 & 246.0116 & 0 & 110.0197 & 220.0395 & 173.9565 & 233.3871 & 1 \\ 155.5914 & 110.0197 & 155.5914 & 110.0197 & 0 & 110.0197 & 173.9565 & 173.9565 & 1 \\ 246.0116 & 155.5914 & 110.0197 & 220.0395 & 110.0197 & 0 & 233.3871 & 173.9565 & 1 \\ 280.4964 & 280.4964 & 320.7599 & 173.9565 & 173.9565 & 233.3871 & 0 & 110.0197 & 1 \\ 320.7599 & 280.4964 & 280.4964 & 233.3871 & 173.9565 & 173.9565 & 110.0197 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

Diğer taraftan çözüm için ihtiyaç duyulan  $\gamma$  yöneyi,  $\mathbf{x}$  konumuna bağlı olup herhangi bir  $\mathbf{x}$  konumu için yapılacak tahmin öncesi belirlenmesi gerekir. Örneğin çıktı değeri bilinmeyen  $\mathbf{x}(3,4)$  konumu için  $\gamma$  yöneyi;

$$\boldsymbol{\gamma} = \begin{bmatrix} \gamma(\mathbf{x}_1, \mathbf{x}) \\ \gamma(\mathbf{x}_2, \mathbf{x}) \\ \vdots \\ \gamma(\mathbf{x}_8, \mathbf{x}) \\ 1 \end{bmatrix}$$

şeklinde olup bu bağlamda öncelikle veri havuzundaki noktalar ile tahminleme yapılması istenen  $\mathbf{x}$  noktası arasındaki mesafelerin, akabinde de deneysel varyogram değerlerinin hesaplanması gerekir. Tahminleme noktası ile veri havuzu noktaları arası mesafe değerleri;

$$h_{i,\mathbf{x}} = \sqrt{(x_{i,1} - x_1)^2 + (x_{i,2} - x_2)^2} \quad i=1,2,\dots,8$$

denklemini çerçevesinde kolayca elde edilebilir. Örneğin ilk veri noktası ile  $\mathbf{x}$  noktası arası mesafe değeri;

$$h_{1,\mathbf{x}} = \sqrt{(1 - 3)^2 + (3 - 4)^2}$$

$$h_{1,\mathbf{x}} = 2.236$$

şeklinde elde edilebilir.





Benzer işlemler diğer veri havuzu noktaları için tekrar edildiğinde;

$$\mathbf{h}^T = [2.236 \quad 2.236 \quad 3 \quad 1 \quad 1 \quad 2.236 \quad 1.414 \quad 2]$$

yönevine ulaşılır. Bu yöneydeki mesafelere ait deneysel varyogram değerleri, doğrusal denklem modeli kullanılarak;

$$\boldsymbol{\gamma}^T = [173.95 \quad 173.95 \quad 233.39 \quad 77.79 \quad 77.79 \quad 173.95 \quad 110.02 \quad 155.59 \quad 1]$$

biçiminde elde edilmiş olur. Hatırlanacağı üzere yöneydeki son değer, Lagrange çarpanına ait çarpan değeridir. Elde edilen katsayılar matrisi ile yeni noktaya ait varyogram yöneyi beraber dikkate alınarak  $\mathbf{x}$  noktasına ait ağırlık katsayıları;

$$\mathbf{w}^T = [-0.0286 \quad -0.0210 \quad -0.0231 \quad 0.3896 \quad 0.3744 \quad 0.0056 \quad 0.2559 \quad 0.0472 \quad -8.0555]$$

veya Lagrange çarpanı hariç tutulursa;

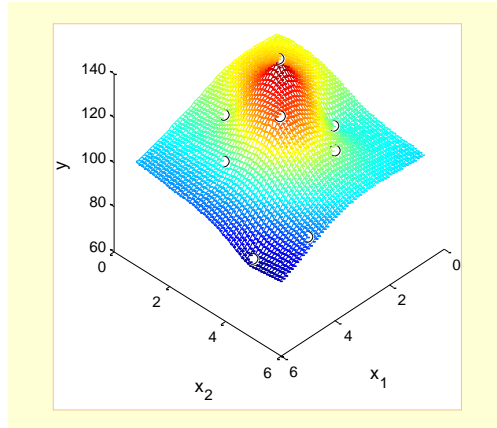
$$\mathbf{w}^T = [-0.0286 \quad -0.0210 \quad -0.0231 \quad 0.3896 \quad 0.3744 \quad 0.0056 \quad 0.2559 \quad 0.0472]$$

şeklinde elde edilebilir. Diğer taraftan  $\mathbf{x}$  noktasına ait belirlenen ağırlık katsayıları yardımıyla aynı noktaya ait tahmini çıktı değeri;

$$y(\mathbf{x}) = \mathbf{w}^T \mathbf{y}$$

$$y(3,4) = 105.1876$$

olarak bulunur.



Şekil 4.35 Kriging yüzeyi ve veri havuzu noktaları.



Mevcut veri havuzuna dayalı olarak belirlenen katsayılar matrisi ve her nokta için tekrar hesaplanan varyogram değerlerinin dikkate alınması sonucu, veri havuzu yakın civarı bölgesinin yüzey modeli Şekil 4.35'deki gibi elde edilebilir.



Kriging yüzeyi eldesi.



**MATLAB uygulaması**



```
clc; clear

% veri havuzu
x11 = [1 2 3 2 3 4 4 5];
x12 = [3 2 1 4 3 2 5 4];
y = [100 130 105 105 120 100 90 80];

% varyogram fonksiyonu \gamma = 77,7957 x h
% katsayılar matrisi
nv = [0,110.019734034309,220.039468068617,110.019734034309,...
155.591400000000,246.011604167161,280.496385360614,...
320.759888318864,1;110.019734034309,0,110.019734034309,...
155.591400000000,110.019734034309,155.591400000000,...
280.496385360614,280.496385360614,1;220.039468068617,...
110.019734034309,0,246.011604167161,155.591400000000,...
110.019734034309,320.759888318864,280.496385360614,1;...
110.019734034309,155.591400000000,246.011604167161,0,...
110.019734034309,220.039468068617,173.956473557180,...
233.387100000000,1;155.591400000000,110.019734034309,...
155.591400000000,110.019734034309,0,110.019734034309,...
173.956473557180,173.956473557180,1;246.011604167161,...
155.591400000000,110.019734034309,220.039468068617,...
110.019734034309,0,233.387100000000,173.956473557180,1;...
280.496385360614,280.496385360614,320.759888318864,...
173.956473557180,173.956473557180,233.387100000000,0,...
110.019734034309,1;320.759888318864,280.496385360614,...
280.496385360614,233.387100000000,173.956473557180,...
173.956473557180,110.019734034309,0,1;1,1,1,1,1,1,1,0];

% tahminleme noktaları
x21 = 0:0.1:5.2;
x22 = 0:0.1:5.2;
[X1 X2] = meshgrid(x21,x22);

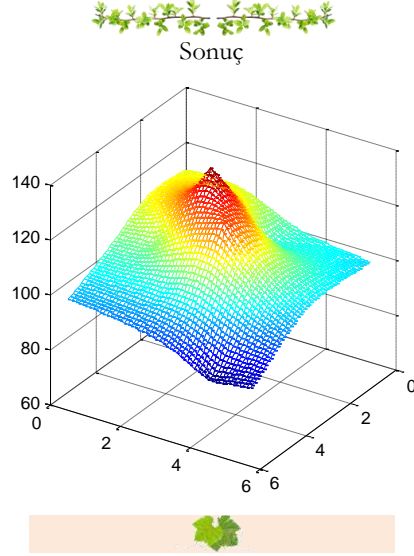
% tahminleme
for i = 1:length(x21)
    for j=1:length(x22)
        dx = sqrt((x11-X1(i,j)).^2+(x12-X2(i,j)).^2);
```



```

nvx = 77.7957*dx; nvx(9) = 1;
nw = inv(nv)*nvx';
w = nw(1:8);
ny(i,j) = y*w;
end
end
mesh(X1,X2,ny); axis square; view(123,28)

```



#### 4.5.2 Basit Kriging yöntemi

Hatırlanacağı üzere sıradan Kriging yönteminde ağırlık katsayılarının normalizasyonu koşulu dikkate alınarak tanımlanan kısıtlı optimizasyon problemi çözümü yapılmakta ve katsayılar için bir denklem elde edilmektedir. Basit Kriging yönteminde ise söz konusu koşul aranmaksızın çözüme gidilir. Ayrıca durağan seriye ait çıktı değeri ortalama değerinin,  $\mu(\mathbf{x})$  bilindiği ve bu değer tüm çözüm uzayı boyunca aynı olduğu kabul edilir. Dolayısıyla basit Kriging tahminleme denklemi;

$$\hat{y}(\mathbf{x}) = \sum_{i=1}^n w_i(\mathbf{x})y(x_i) + \mu(\mathbf{x}) \left(1 - \sum_{i=1}^n w_i(\mathbf{x})\right) \quad (4.83)$$

şeklinde olup ağırlık katsayıları, kısıtsız eniyileme sürecinin çözümü bağlamında;



$$\mathbf{w} = \Gamma^{-1}\boldsymbol{\gamma} \quad (4.84)$$

$$\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} \gamma(x_1, x_1) & \gamma(x_1, x_2) & \cdots & \gamma(x_1, x_n) \\ \gamma(x_2, x_1) & \gamma(x_2, x_2) & \cdots & \gamma(x_2, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ \gamma(x_n, x_1) & \gamma(x_n, x_2) & \cdots & \gamma(x_n, x_n) \end{bmatrix}^{-1} \begin{bmatrix} \gamma(x_1, \mathbf{x}) \\ \gamma(x_2, \mathbf{x}) \\ \vdots \\ \gamma(x_n, \mathbf{x}) \end{bmatrix} \quad (4.85)$$

denkleminde elde edilebilir. Burada  $\gamma(x_i, x_j)$  değeri, tıpkı sıradan Kriging yönteminde olduğu gibi varyogram değerini temsil etmektedir<sup>(45)</sup>. Basit Kriging yöntemi daha basit gibi görünse de uygulamada durağan seriyeye ait ortalama değerin bilinmesi zordur.

### 4.5.3 Tümel Kriging

Tümel Kriging yöntemi, temelde sıradan Kriging yöntemi ile büyük benzerlik arzeder. Sıradan Kriging yöntemi ile yegâne farkı, ilave olarak genel veya yerel eğilim fonksiyonlarının dikkate alınmasıdır. Haliyle üzerinde çalışılan veri dağılımından durağan seri özelliği taşımasını beklemek mümkün değildir. Tümel Kriging yöntemi, özellikle durağan olmayan ancak eğilimli süreçlerin tahminlemede tercih edilir. Tümel Kriging yönteminde dikkate alınan tahminleme denklemini;

$$\hat{y}(\mathbf{x}) = \sum_{i=1}^n w_i(\mathbf{x})y(x_i) + P(\mathbf{x}) \quad (4.86)$$

biçiminde ifade etmek mümkündür. Buradaki ilk terim, sıradan Kriging yöntemindeki ağırlık katsayıları ile aynıdır. İkinci terim ise genel eğilimi temsil eden  $P(\mathbf{x})$  fonksiyonundan müteşekkil doğrusal ilişkiyi temsil etmektedir. Genel eğilim fonksiyonu genellikle 1'inci veya 2'nci dereceden polinomial fonksiyon olarak tanımlanır. Örneğin  $m$  boyutlu bir konum vektörünün söz konusu olduğu durumda 1'inci dereceden doğrusal trend fonksiyonu;

$$P(\mathbf{x}) = \alpha_0 + \sum_{j=1}^m \alpha_j x_j \quad (4.87)$$

şeklinde tanımlanabilir. Daha somut bir çerçevede misal vermek gerekirse, iki boyutlu bir ortamda söz konusu eğilim denklemi;

$$P(\mathbf{x}) = \alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 \quad (4.88)$$

olarak yazılabilir.



Eğilimin ikilenik formda olduğu düşünüldüğünde eğilim fonksiyonu;

$$P(x) = \alpha_0 + \sum_{i=1}^m \alpha_i x_i + \sum_{i=1}^{2m} \sum_{j=m+i}^m \alpha_j x_i^2 + \sum_{\substack{k=1 \\ p>k \\ q=2m+k}}^m \alpha_q x_k x_p \quad (4.89)$$

formunu alır. Örneğin yine iki boyutlu bir ortamda söz konusu ikilenik eğilim denklemi;

$$P(x) = \alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_1^2 + \alpha_4 x_2^2 + \alpha_5 x_1 x_2 \quad (4.90)$$

şeklinde tanımlanabilir. Diğer taraftan gerek ağırlık ve gerekse de eğilim fonksiyonuna ait katsayıların eldesi;

$$\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \\ \lambda \\ \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_m \end{bmatrix} = \begin{bmatrix} \gamma(x_1, x_1) & \gamma(x_1, x_2) & \dots & \gamma(x_1, x_n) & 1 & x_{1,1} & x_{1,1} & \dots & x_{1,m} \\ \gamma(x_2, x_1) & \gamma(x_2, x_2) & \dots & \gamma(x_2, x_n) & 1 & x_{2,1} & x_{2,2} & \dots & x_{2,m} \\ \vdots & \vdots & \dots & \vdots & 1 & \vdots & \vdots & \dots & \vdots \\ \gamma(x_n, x_1) & \gamma(x_n, x_2) & \dots & \gamma(x_n, x_n) & 1 & x_{n,1} & x_{n,2} & \dots & x_{n,m} \\ 1 & 1 & \dots & 1 & 0 & 0 & 0 & \dots & 0 \\ x_{1,1} & x_{1,2} & \dots & x_{1,n} & 0 & 0 & 0 & \dots & 0 \\ x_{2,1} & x_{2,2} & \dots & x_{2,n} & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \vdots & \dots & \vdots \\ x_{m,1} & x_{m,2} & \dots & x_{m,n} & 0 & 0 & 0 & \dots & 0 \end{bmatrix}^{-1} \begin{bmatrix} \gamma(x_1, x) \\ \gamma(x_2, x) \\ \vdots \\ \gamma(x_n, x) \\ 1 \\ x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} \quad (4.91)$$

biçimindeki doğrusal denklem takımının çözümü sonucu elde edilebilir. Burada  $w_i$  ağırlık katsayı değerlerini,  $\lambda$  Lagrange çarpanını,  $\alpha_j$  doğrusal nitelikteki eğilim fonksiyonu katsayılarını,  $\gamma(x_i, x_j)$  veri havuzu noktalarına ait varyogram değerlerini,  $\gamma(x_i, x)$  veri havuzu noktası ile tahminleme yapılacak nokta arasındaki varyogram değerlerini,  $x_{j,i}$  veri havuzu noktalarına ait koordinat verilerini ve  $x_j$  tahminleme yapılacak konuma ait koordinat verilerini temsil etmektedir<sup>(46)</sup>.

#### 4.5.4 Ko-Kriging

Kriging yöntemi bahsinde şu ana kadar ifade edilen yöntemler bir anlamda tekil veya çoğul doğrusal regresyon yöntemleri kapsamında değerlendirilebilir. Ko-Kriging yöntemi ise diğerlerinden farklı olarak birden fazla çıktı değişkeninin söz konusu olduğu çok değişkenli doğrusal regresyon yöntemleri arasındadır. Bazen herhangi bir  $x$  konumuna ait bir çıktı değerinin tahmin edilmesinde, aynı veya farklı konuma ait diğer çıktı değişkenlerinin de etkisi olduğu





düşünülebilir. Böyle durumlarda hedef çıktı değerine<sup>i</sup> ilave olarak diğer ilişkili çıktı verilerinin<sup>ii</sup> de hesaplamalarda dikkate alınması mümkündür. İlişkili çıktı verileri, bazı çalışmalarda ikincil veri<sup>iii</sup> veya yardımcı ilave veri olarak adlandırılır. Ko-Kriging yönteminin uygulanması sadedinde Kriging yöntemi için gerekli koşullar aynen geçerlidir. Ancak önceki yöntemlere ilave olarak daha fazla sayıda varyogram modelleme ve hesaplama işlemleri söz konusudur<sup>(47)</sup>. En genel hali ile  $p$  çıktı değişkeninin söz konusu olduğu bir sorunda hedef çıktı değerinin hesaplanmasında;

$$\hat{y}_1(\mathbf{x}) = \sum_{i=1}^p \sum_{j=1}^q w_{i,j}(\mathbf{x}_j) y_j(\mathbf{x}_j) \quad (4.92)$$

denklemi dikkate alınabilir. Burada  $\hat{y}_1(\mathbf{x})$  herhangi bir  $\mathbf{x}$  noktasındaki tahmini hedef çıktı değerini,  $y_j(\mathbf{x}_j)$  hedef ve diğer ilişkili çıktı değişkenlerinin eldeki veri havuzu noktalarındaki değerlerini,  $w_{i,j}(\mathbf{x}_j)$  hedef ve diğer ilişkili çıktı değişkenlerinin ağırlık katsayı çarpanlarını ifade etmektedir. Daha somut bir çerçevede göstermek amacıyla üç çıktı değişkeninin söz konusu olduğu bir sorunda herhangi bir  $\mathbf{x}$  noktasına ait hedef çıktı değerinin tahminlemesinin;

$$\hat{y}_1(\mathbf{x}) = \sum_{i=1}^a w_{1,i}(\mathbf{x}_i) y_1(\mathbf{x}_i) + \sum_{j=1}^b w_{2,j}(\mathbf{x}_j) y_2(\mathbf{x}_j) + \sum_{k=1}^c w_{3,k}(\mathbf{x}_k) y_3(\mathbf{x}_k) \quad (4.93)$$

şeklinde yazılabileceği öngörülebilir<sup>(48)</sup>. Görüldüğü üzere tahminlemede aynı sayıdaki aynı veri havuzu noktalarının kullanılması zorunluluğu yoktur. Diğer taraftan denklemde geçen ve bilinmeyen niteliğindeki ağırlık katsayıları, hata içerikli ortak-varyansın minimizasyonu yaklaşımı ile elde edilebilir. Ayrıca eniyileme sürecinde dikkate alınan kısıtlar da mevcut normalizasyon kısıtına ilave olarak;

$$\sum_{i=1}^a w_{1,i}(\mathbf{x}_i) = 1 \quad (4.94)$$

$$\sum_{j=1}^b w_{2,j}(\mathbf{x}_j) = 0 \quad (4.95)$$

$$\sum_{k=1}^c w_{3,k}(\mathbf{x}_k) = 0 \quad (4.96)$$

şeklinde dikkate alınır.

<sup>i</sup> predictand

<sup>ii</sup> covariables

<sup>iii</sup> secondary variables





Çıktı değişken sayısının daha fazla olması halinde aynı kısıt yaklaşımı diğer ilişkili değişkenler için geçerlidir. Buna göre bahse konu kısıtlı eniyileme süreci nihayetinde ağırlık katsayılarını;

$$\begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \mathbf{w}_3 \\ \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} = \begin{bmatrix} \Gamma_{11} & \Gamma_{12} & \Gamma_{13} & 1_a & 0 & 0 \\ \Gamma_{21} & \Gamma_{22} & \Gamma_{23} & 0 & 1_b & 0 \\ \Gamma_{31} & \Gamma_{32} & \Gamma_{33} & 0 & 0 & 1_c \\ 1_a^T & 0 & 0 & 0 & 0 & 0 \\ 0 & 1_b^T & 0 & 0 & 0 & 0 \\ 0 & 0 & 1_c^T & 0 & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \\ 1 \\ 0 \\ 0 \end{bmatrix} \quad (4.97)$$

şeklindeki doğrusal denklem takımının çözümü yardımıyla elde etmek mümkündür. Burada  $\Gamma_{ij}$  veri havuzu noktalarını içeren varyogram matrislerini;

$$\Gamma_{11} = \begin{bmatrix} \gamma_{11}(\mathbf{x}_1, \mathbf{x}_1) & \gamma_{11}(\mathbf{x}_1, \mathbf{x}_2) & \cdots & \gamma_{11}(\mathbf{x}_1, \mathbf{x}_a) \\ \gamma_{11}(\mathbf{x}_2, \mathbf{x}_1) & \gamma_{11}(\mathbf{x}_2, \mathbf{x}_2) & \cdots & \gamma_{11}(\mathbf{x}_2, \mathbf{x}_a) \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{11}(\mathbf{x}_a, \mathbf{x}_1) & \gamma_{11}(\mathbf{x}_a, \mathbf{x}_2) & \cdots & \gamma_{11}(\mathbf{x}_a, \mathbf{x}_a) \end{bmatrix}, \quad \Gamma_{12} = \begin{bmatrix} \gamma_{12}(\mathbf{x}_1, \mathbf{x}_1) & \gamma_{12}(\mathbf{x}_1, \mathbf{x}_2) & \cdots & \gamma_{12}(\mathbf{x}_1, \mathbf{x}_b) \\ \gamma_{12}(\mathbf{x}_2, \mathbf{x}_1) & \gamma_{12}(\mathbf{x}_2, \mathbf{x}_2) & \cdots & \gamma_{12}(\mathbf{x}_2, \mathbf{x}_b) \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{12}(\mathbf{x}_a, \mathbf{x}_1) & \gamma_{12}(\mathbf{x}_a, \mathbf{x}_2) & \cdots & \gamma_{12}(\mathbf{x}_a, \mathbf{x}_b) \end{bmatrix} \quad (4.98)$$

$$\Gamma_{13} = \begin{bmatrix} \gamma_{13}(\mathbf{x}_1, \mathbf{x}_1) & \gamma_{13}(\mathbf{x}_1, \mathbf{x}_2) & \cdots & \gamma_{13}(\mathbf{x}_1, \mathbf{x}_c) \\ \gamma_{13}(\mathbf{x}_2, \mathbf{x}_1) & \gamma_{13}(\mathbf{x}_2, \mathbf{x}_2) & \cdots & \gamma_{13}(\mathbf{x}_2, \mathbf{x}_c) \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{13}(\mathbf{x}_a, \mathbf{x}_1) & \gamma_{13}(\mathbf{x}_a, \mathbf{x}_2) & \cdots & \gamma_{13}(\mathbf{x}_a, \mathbf{x}_c) \end{bmatrix}, \quad \Gamma_{22} = \begin{bmatrix} \gamma_{22}(\mathbf{x}_1, \mathbf{x}_1) & \gamma_{22}(\mathbf{x}_1, \mathbf{x}_2) & \cdots & \gamma_{22}(\mathbf{x}_1, \mathbf{x}_b) \\ \gamma_{22}(\mathbf{x}_2, \mathbf{x}_1) & \gamma_{22}(\mathbf{x}_2, \mathbf{x}_2) & \cdots & \gamma_{22}(\mathbf{x}_2, \mathbf{x}_b) \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{22}(\mathbf{x}_b, \mathbf{x}_1) & \gamma_{22}(\mathbf{x}_b, \mathbf{x}_2) & \cdots & \gamma_{22}(\mathbf{x}_b, \mathbf{x}_b) \end{bmatrix} \quad (4.99)$$

$$\Gamma_{23} = \begin{bmatrix} \gamma_{23}(\mathbf{x}_1, \mathbf{x}_1) & \gamma_{23}(\mathbf{x}_1, \mathbf{x}_2) & \cdots & \gamma_{23}(\mathbf{x}_1, \mathbf{x}_c) \\ \gamma_{23}(\mathbf{x}_2, \mathbf{x}_1) & \gamma_{23}(\mathbf{x}_2, \mathbf{x}_2) & \cdots & \gamma_{23}(\mathbf{x}_2, \mathbf{x}_c) \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{23}(\mathbf{x}_b, \mathbf{x}_1) & \gamma_{23}(\mathbf{x}_b, \mathbf{x}_2) & \cdots & \gamma_{23}(\mathbf{x}_b, \mathbf{x}_c) \end{bmatrix}, \quad \Gamma_{33} = \begin{bmatrix} \gamma_{33}(\mathbf{x}_1, \mathbf{x}_1) & \gamma_{33}(\mathbf{x}_1, \mathbf{x}_2) & \cdots & \gamma_{33}(\mathbf{x}_1, \mathbf{x}_c) \\ \gamma_{33}(\mathbf{x}_2, \mathbf{x}_1) & \gamma_{33}(\mathbf{x}_2, \mathbf{x}_2) & \cdots & \gamma_{33}(\mathbf{x}_2, \mathbf{x}_c) \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{33}(\mathbf{x}_c, \mathbf{x}_1) & \gamma_{33}(\mathbf{x}_c, \mathbf{x}_2) & \cdots & \gamma_{33}(\mathbf{x}_c, \mathbf{x}_c) \end{bmatrix} \quad (4.100)$$

$\mathbf{w}_i$  ağırlık katsayı yöneylerini;

$$\mathbf{w}_1 = \begin{bmatrix} w_{1,1} \\ w_{2,1} \\ \vdots \\ w_{a,1} \end{bmatrix}, \mathbf{w}_2 = \begin{bmatrix} w_{1,2} \\ w_{2,2} \\ \vdots \\ w_{b,2} \end{bmatrix}, \mathbf{w}_3 = \begin{bmatrix} w_{1,3} \\ w_{2,3} \\ \vdots \\ w_{c,3} \end{bmatrix} \quad (4.101)$$

$\gamma_i$  veri havuzu noktaları ile  $\mathbf{x}$  tahminleme noktasını içeren;





$$\boldsymbol{y}_1 = \begin{bmatrix} y_{1,1} \\ y_{2,1} \\ \vdots \\ y_{a,1} \end{bmatrix}, \boldsymbol{y}_2 = \begin{bmatrix} y_{1,2} \\ y_{2,2} \\ \vdots \\ y_{b,2} \end{bmatrix}, \boldsymbol{y}_3 = \begin{bmatrix} y_{1,3} \\ y_{2,3} \\ \vdots \\ y_{c,3} \end{bmatrix} \quad (4.102)$$

varyogram yöneylerini,  $\lambda_i$  Lagrange çarpanlarını;  $1_a, 1_b, 1_c$  ise sırasıyla  $a \times 1, b \times 1, c \times 1$  ebatlarında ve tüm elemanları 1 olan sütun şeklindeki yöneyleri temsil etmektedir<sup>(49)</sup>. Hatırlanacağı üzere varyogram değerleri tek bir çıktı söz konusu olduğunda;

$$\gamma_{ii}(h) = \frac{1}{2|N(h)|} \sum_{i,j \in N(h)} (y_i(\boldsymbol{x}_i) - y_i(\boldsymbol{x}_j))^2$$

iki çıktı söz konusu olduğunda ise çapraz varyogram değeri;

$$\gamma_{ij}(h) = \frac{1}{2|N(h)|} \sum_{i,j \in N(h)} (y_i(\boldsymbol{x}_i) - y_i(\boldsymbol{x}_j))(y_j(\boldsymbol{x}_i) - y_j(\boldsymbol{x}_j))$$

ifadelerine göre hesaplanmaktaydı<sup>(50)</sup>.

Gerek Kriging ve gerekse de Ko-Kriging yöntemleri, özellikle belirli bir bölgede yerel olarak dikkate alındığında etkin ve belli derecede hassas sonuçlar verebilmektedir. Ancak bu tür yöntemlerin en ciddi sorunu, her tahminleme noktası için ağırlık katsayılarının yeniden hesaplanması zorunluluğu olup özellikle çok sayıda veri noktasının söz konusu olduğu problemlerde bahse konu zorunluluk ciddi hesaplama yüklerini de beraberinde getirir<sup>(51)</sup>.

#### 4.6 Destek yöney yöntemi

Destek yöney yöntemi<sup>ii</sup>, tıpkı bir önceki yöntemde olduğu üzere istatistiksel bir yöntem olup çıkış kaynağı itibari ile örüntü tanıma<sup>iii</sup> sorunsalına yönelik olarak geliştirilmiştir. 1930'lu yıllarda R. A. Fisher'ın örüntü tanıma amacıyla gündeme getirdiği ilk algoritmaların destek yöney yöntemine dönüşümü, özellikle A. Y. Chervonenkis ve V.N. Vapnik'in çalışmaları ile belli bir olgunluğa ulaşmıştır<sup>(52)</sup>. Günümüzde kullanımı gittikçe yaygınlaşan destek yöney yöntemleri,

<sup>i</sup> Cross variogram

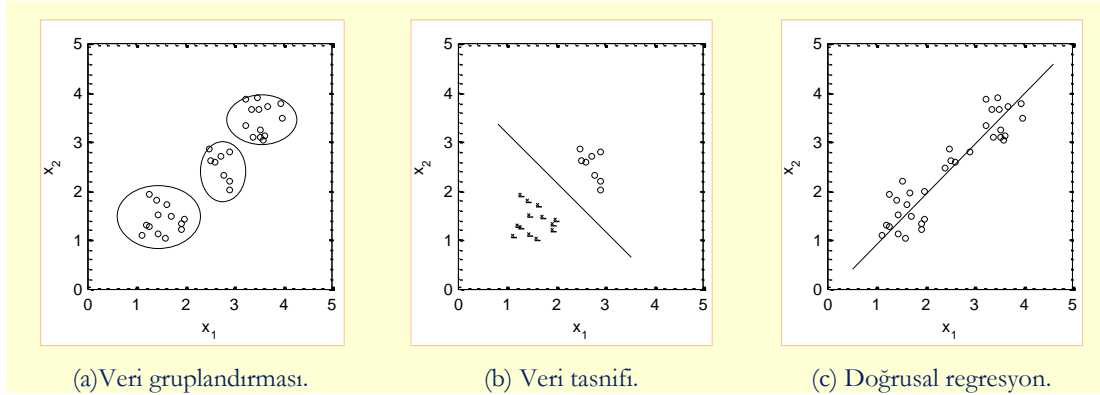
<sup>ii</sup> Support Vector Machines (SVM)

<sup>iii</sup> Pattern recognition





yapay zekâ yaklaşımlarından biri olan makine öğrenmesi kapsamındaki gözetimli öğrenme<sup>ii</sup> algoritmaları adı altında tasnif edilir. Yöntemin öncelikli kullanım amacı sınıflandırma olmakla beraber, vekil model eldesi için de kendisinden yararlanılabilmektedir. Gerek veri tasnifi ve gerekse de vekil model bağlamında eğri uydurma işlemleri, denetimli öğrenme dâhilinde dillendirilmekte, verilerin etiketlenmediği durumlarda yapılan gruplandırma çalışmaları ise gözetimsiz öğrenme<sup>iii</sup> olarak adlandırılmaktadır.



Şekil 4.36 Veri işlemleri.

İfade edilen süreçlerin daha iyi anlaşılması bağlamında Şekil 4.36'daki grafikler gözlemlenebilir. Şekil 4.36(a)'da görüldüğü üzere göre eldeki verinin herhangi bir tanıtım etiketinin olmadığı durumlarda verilerin birbirlerine göre konumları dikkate alınarak gruplandırmaya<sup>iv</sup> gidilebilir. Diğer taraftan verilerin etiketli olduğu durumlarda mevcut veri grubu dışındaki herhangi bir verinin hangi gruba dâhil olacağıнын belirlenmesi hususu, tasnif<sup>v</sup> olarak değerlendirilmektedir. Şekil 4.36(b)'de görüldüğü üzere "x" ve "o" olarak etiketlenen iki veri tipi mevcut olup oluşturulan doğru parçası ile veri havuzu tasnif edilebilmektedir. Veri havuzu ile yapılabilen diğer bir işlem, eğri uydurma niteliğindeki doğrusal regresyon olup Şekil 4.36(c) dâhilinde örnek bir doğrusal regresyon görülmektedir. İfade edildiği üzere destek yöney yöntemleri, veri tasnifi ve

<sup>i</sup> Machine learning

<sup>ii</sup> Supervised learning

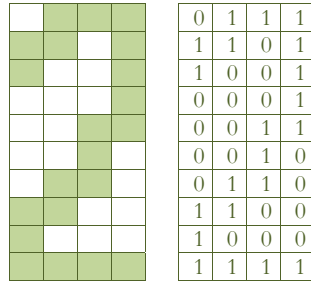
<sup>iii</sup> Unsupervised learning

<sup>iv</sup> Clustering

<sup>v</sup> Classification



doğrusal ya da doğrusal olmayan regresyon gayesi ile kullanılabilen yöntemlerdendir<sup>(53)</sup>. Veri tasnifi işlemlerinin literatürde pek çok uygulamasına rastlanmakta olup sorunun uygun bir örnekle açıklanmasında fayda mülâhaza edilmektedir. Örneğin herhangi bir tarama işlemi sonrasında elde edilen şeklin hangi sayı olduğu sorunsalı, sınıflandırma problemleri meyanında dikkate alınabilir. Müteakip şekilden de görüleceği üzere  $10 \times 4$  tarama grid yapısına sahip bir alanda yapılan tarama işlemi sonrasında elde edilen şekil, 40 elemanlı bir yöney olarak düşünülebilir. Herhangi bir başka tarama neticesinde elde edilen şeklin ne olduğu bağlamında yapılacak karşılaştırmalarda kullanılmak üzere; örnekte kullanılan iki (2) sayısına ait şekle ilave olarak farklı fontlarda yazılan belirli sayıdaki iki (2) sayısına ait diğer yöneyler dikkate alınarak veri havuzunun oluşturulması mümkündür<sup>(54)</sup>.



$$\mathbf{a}^T = [0\ 1\ 1\ 1\ 1\ 1\ 0\ 1\ 1\ 0\ 0\ 1\ 0\ 0\ 0\ 1\ 0\ 0\ 1\ 1\ 0\ 0\ 1\ 0\ 0\ 1\ 1\ 0\ 1\ 1\ 0\ 0\ 1\ 0\ 0\ 0\ 1\ 1\ 1\ 1]$$

Şekil 4.37 Veri tasnifi kapsamında şekil tarama ve sonucunun yöney olarak ifade edilmesi.

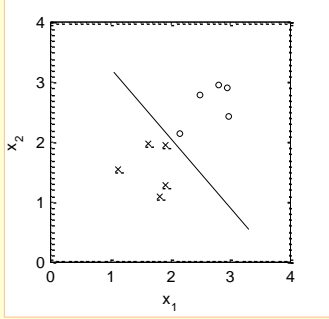
Destek yöney yöntemlerini, üzerinde çalışılan sorunun doğasına göre üç ana grupta tasnif mümkündür. Söz konusu gruplar; doğrusal destek yöney yöntemleri, doğrusal olmayan destek yöney yöntemleri ve çok sınıflı destek yöney yöntemleridir. Doğrusal destek yöney yöntemlerini de kendi bünyesinde iki ayrı alt dala ayırmak mümkündür. Eldeki veri havuzunun, doğrusal bir düzlemle ayrılmasının mümkün olduğu durumlar ile mümkün olmadığı durumlar söz konusu olabilir. Ancak bahse konu ikinci seçeneğin özelliği, uygun bir dönüşümle doğrusal ayrımın mümkün olmasıdır. Şekil 4.38 dâhilinde bahse konu sorunlar şekilsel olarak gözlemlenebilir. Örneğin doğrusal olarak ayrılabilen veri dağılımları (a) ve (b) seçeneklerindedir. Bazı veri dağılımları birbirinden kesin olarak ayrılabilir. Bazı veri havuzları ise belirli esnek sınır tanımlamaları yapılarak doğrusal bir düzlemle birbirinden ayrılabilir. İlkine kesin doğrusal<sup>1</sup>

<sup>1</sup>Hard margin

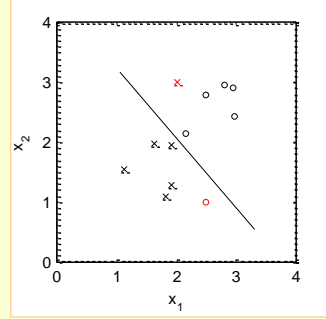




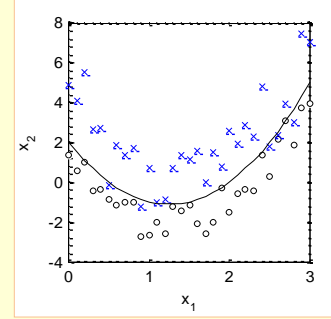
ayrılabilme, diğerine esnek doğrusal<sup>i</sup> ayrılabilme hali adı verilir. İlk seçenek kapsamında görüldüğü üzere  $\times$  ve  $\circ$  işaretleri ile etiketlenen veriler, birbirlerinden konumsal olarak net bir şekilde ayırt edilebilmektedirler. Ancak (b) seçeneğinde, her iki gruba ait birkaç verinin konumsal olarak birbirine geçmiş durumda olması nedeni ile kesin bir ayırım yapmak mümkün değildir. Bu durumda sınır tanımlamalarında bir takım esneklikler gösterilerek tasnife gitmek mümkündür.



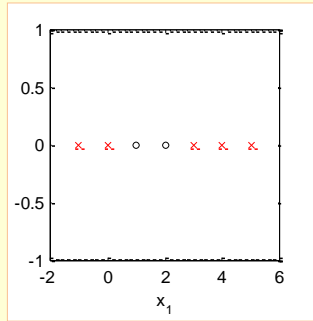
(a) Kesin doğrusal ayrılabilme durumu.



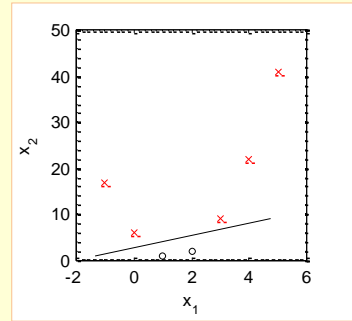
(b) Esnek doğrusal ayrılabilme durumu.



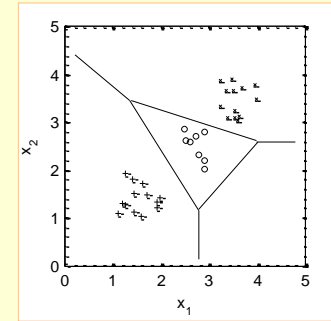
(c) Doğrusal olmayan şekilde ayrılabilme durumu.



(ç) Girdi uzayında doğrusal olarak ayrılamama durumu.



(d) Dönüşüm uzayında doğrusal olarak ayrılabilme durumu.



(e) Çoklu sınıflı ayrılabilme durumu.

Şekil 4.38 Destek yöney yöntemlerinde karşılaşılabilen sorunlar.

<sup>i</sup>Soft margin





Diğer taraftan bazı veri havuzundaki veriler, buldukları ortamda doğrusal olarak ayrılabilirler. Örneğin (ç) seçeneğinde söz konusu durum söz konusudur. Toplam yedi verinin olduğu 1 boyutlu ortamda iki ayrı veri tipi olmakla beraber bu verileri tek bir doğru ile ayırmak mümkün değildir. Bu durumda aynı verilerin, uygun bir fonksiyonla bir üst boyuta dönüşümü yapılarak yeni konumlandırılmada doğrusal ayrımına gidilebilir. Genellikle verilerin bulunduğu ortama girdi uzayı<sup>i</sup> adı verilir. Aynı veri havuzunun uygun bir dönüşümle aktarıldığı yeni ortama ise dönüşüm veya öznelik uzayı<sup>ii</sup> adı verilir. Şekil 4.38(d)'den de gözlemlenebileceği üzere 2 boyutlu ortama taşınan veriler, rahatlıkla doğrusal bir düzlemlenmektedir. Veri havuzu tasnifinde, verilerin doğrusal olarak ayrılmadığı hallerde, Şekil 4.38(c)'de olduğu gibi doğrusal olmayan eğri veya yüzey vasıtası ile bir ayrımına gidilebilir. Tasnif işlemi sürecinde ikiden fazla veri türünün olması halinde, çok sınıflı veri tasnifine gidilir. Örneğin Şekil 4.38(e)'de görüldüğü üzere +, o ve × etiketli üç tür veri söz konusu olup bu verilerin tasnifi amacıyla farklı yaklaşımlar sergilenebilir. Yaygın uygulama, destek yöney yöntemi ile ikili çiftler halinde sınıflandırmaya gidilmesi ve bu sayede ayrım sınırlarının belirlenmesidir<sup>(55)</sup>.

Destek yöney yöntemleri her ne kadar sınıflandırma amacıyla kullanılıyor görünse de benzer yaklaşım, vekil model eldesi için de geçerlidir. Bu nedenle yöntemin sınıflandırma için geçerli olan detaylarına göz atılması sonrasında vekil model eldesi için dikkate alınan hususlara değinilmesinde fayda görülmektedir.

#### 4.6.1 Kesin sınıflandırma

Elimizde  $n$  tane veri olduğunu, her bir verinin iki kümeden birine ait olduğunu düşünelim;

$$(x_1, y_1), (x_2, y_2), \dots, (x_i, y_i), \dots, (x_n, y_n) \quad (4.103)$$

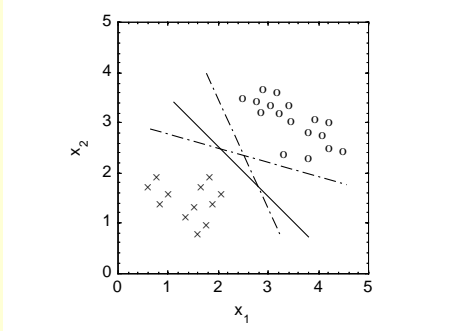
Burada  $x_i$   $i$ 'nci verinin konumunu gösteren yöneyi,  $y_i$  aynı yöneye ait küme aidiyetini belirtmektedir. Genellikle iki kümeli bir veri setinde küme aidiyeti +1 veya -1 olarak değerlendirilir. Şekil 4.39(a)'da örnek veri havuzu ve bu verilerin dağılımı görülmektedir. Buna göre  $y_i = +1$  olarak etiketlenen veri noktaları "o" işaretiyle,  $y_i = -1$  olarak etiketlenen veri noktaları ise "×" işareti ile gösterilmektedir. Destek yöney yöntemi ile hedeflenen ana gaye, eldeki veri havuzundan hareketle veri havuzu dışındaki herhangi bir noktanın hangi kümeye ait olduğunun belirlenmesidir. Bu bağlamda yapılacak ilk yaklaşım, verilerin etiketlerini baz alarak

<sup>i</sup> Input space

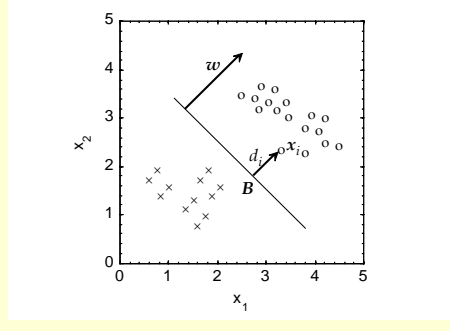
<sup>ii</sup> Feature space



ikiye ayrılmasını sağlayan bir doğrunun eldesidir. Tabiatıyla bahse konu iki kümeyi ayırabilen olası sonsuz sayıda doğru elde edilebilir. Ancak makul olan, mevcut küme noktalarını birbirinden ayırabilen ve bu kümelere ait noktalara en uzak olan doğrunun elde edilmesidir.



(a) İki kümeli veri dağılımı ve olası ayırım hatları



(b) Olası ayırım hattının belirlenmesi

Şekil 4.39 Doğrusal sınıflandırmada temel yaklaşım.

Uzaklık sadedinde dikkate alınan mesafe kavramı, noktaların doğruya olan dikey uzaklığına dayandırılır. Dikey uzaklık kavramı takdir edileceği üzere matematiksel manada izdüşüm veya sayısal çarpım olarak ifade edilebilir. Şekil 4.39(b)'de görüldüğü üzere herhangi bir  $x_i$  noktasının, kümeleri birbirinden ayıran ayırım hattına<sup>1</sup> olan dik uzaklığı  $d_i$  ve bu doğrultudaki yöney de  $w$  olarak ifade edilebilir. Literatürde  $w$  ağırlık yöneyi veya normal yöney olarak tanımlanır. Normal yöneyin özelliği, ayırım hattına dik yönde olmasıdır. Daha sonra da görüleceği üzere normal yöneyin diğer bir özelliği de dik olduğu doğrunun ki burada ayırım hattı olarak tanımlanmaktadır, eğimini belirlemesidir. Elimizdeki  $x_i$  noktasının ayırım hattına olan  $d_i$  dikey mesafesini,  $x_i$  noktası ile bu noktanın ayırım hattının üzerindeki izdüşüm noktası olan  $B$  noktasından yararlanarak elde etmek mümkündür. Şöyle ki; ayırım hattına normal doğrultudaki birim yöneyi;

$$e_n = \frac{w}{\|w\|} \quad (4.104)$$

şeklinde yazabiliriz.

<sup>1</sup>Decision boundary



Bu tanımlamayı dikkate alarak  $\mathbf{B}$  ve  $\mathbf{x}_i$  noktaları arası konumu belirleyen yöneyi;

$$\mathbf{x}_i - d_i \mathbf{e}_n = \mathbf{x}_i - d_i \frac{\mathbf{w}}{\|\mathbf{w}\|} \quad (4.105)$$

şeklinde ifade edebiliriz. Diğer taraftan  $\mathbf{B}$  noktası, ayırım hattı üzerinde bir nokta olup herhangi bir doğrunun, daha genel bir ifade ile düzlemin genel denklemi;

$$\mathbf{w} \cdot \mathbf{x} + b = 0 \quad (4.106)$$

şeklinde yazılabilir. Denklemin daha iyi anlaşılabilmesi bağlamında örnek bir doğru üzerinden mevzu değerlendirilebilir. En basit hali ile iki boyutlu ortamda örnek bir doğru denklemini;

$$x_1 + 2x_2 = 3$$

biçiminde yazabiliriz. Bu ifadeyi;

$$[1 \quad 2]^T \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - 3 = 0$$

formunda da yazmak mümkündür. Buna göre genel denklem terimlerini;

$$\mathbf{w} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad b = -3$$

olarak açmak mümkündür. Dikkat edileceği üzere denklemde geçen  $\mathbf{w}$  terimi doğrunun eğimini belirlemede,  $b$  ifadesi  $x_2$  ekseninin kesim yerini gösteren sabit bir sayıyı temsil etmektedir. Şekil 4.40 dâhilinde  $b$  sabit teriminin doğru şekline etkisi görülmektedir.

Ayırım hattı üzerindeki  $\mathbf{B}$  noktası tekrar dikkate alınıp, bu noktanın düzlem denklemini sağlaması gerektiği göz önüne alınırsa;

$$\mathbf{w} \cdot \left[ \mathbf{x}_i - d_i \frac{\mathbf{w}}{\|\mathbf{w}\|} \right] + b = 0 \quad (4.107)$$

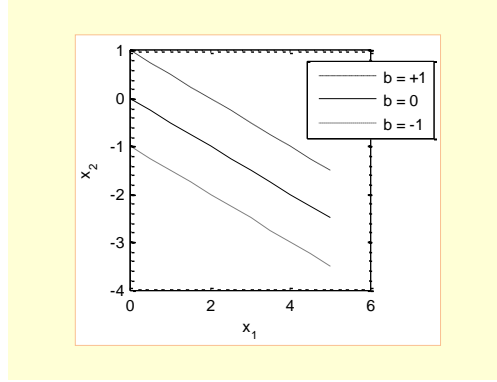
eşitliğine ve bu denklemde mesafe değeri yalnız bırakılırsa;





$$d_i = \frac{1}{\|\mathbf{w}\|} [\mathbf{w} \cdot \mathbf{x}_i + b] \quad (4.108)$$

şeklindeki ifadeye ulaşılabilir.



Şekil 4.40 Düzlem denklemindeki  $b$  sabit teriminin düzleme etkisi.

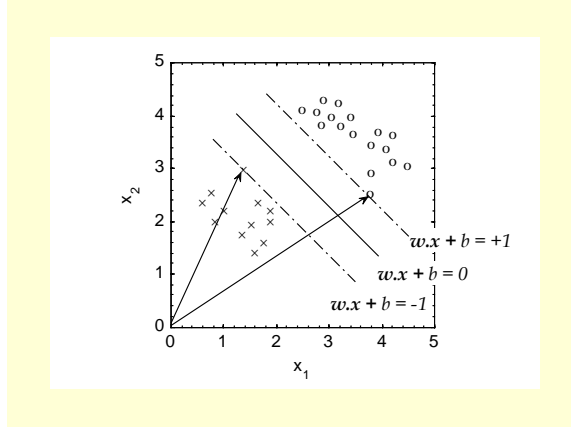
Görüldüğü üzere ayırım hattı ile veri havuzu noktaları arasındaki mesafe büyük oranda normal yöneye ve hatta bu yöneyin şiddetine bağlıdır. Diğer taraftan belirlenmesi istenen ayırım hattının, veri havuzu noktalarına azami derecede uzak olması istenir. Bu nedenle tanımlanabilecek bir eniyileme sürecinde  $\max_{\mathbf{w},b} d_i$  amacı güdülebilir. Bu amaç, mesafe değeri tanımından hareketle  $\min_{\mathbf{w},b} \|\mathbf{w}\|$  şeklinde de ifade edilebilir. Genellikle ikilenik formdaki eniyileme yöntemlerinden faylanmak amacıyla sorunun tanımı;

$$\min_{\mathbf{w},b} \frac{1}{2} \|\mathbf{w}\|^2 \quad (4.109)$$

şeklindeki hali ile dikkate alınır. Öte yandan süreci sınırlayan diğer bir husus, verilerin etiketidir. Veri etiketinin düzlem denkleminde dâhil edilmesi basit bir yaklaşımla, düzlem denkleminin  $+1$  veya  $-1$  değerine eşitlenmesi ile gerçekleştirilir. Bu yaklaşıma göre ayırım hattı denkleminin  $\mathbf{w} \cdot \mathbf{x} + b = 0$  olması yanında iki ilave hat daha tanımlanır. Bunlardan ilki  $+1$  etiketine sahip veri noktalarının, ayırım hattına en yakın olan noktalarını barındıran  $\mathbf{w} \cdot \mathbf{x}_i + b = +1$  doğrusudur. Diğer hat ise  $-1$  etiketine sahip veri noktalarının ayırım hattına en yakın olan noktalarını barındıran  $\mathbf{w} \cdot \mathbf{x}_j + b = -1$  doğrusudur. Tabiatıyla ilave doğruların eğimlerinin aynı ve ayırım hattına paralel



olması öngörülmektedir. Doğrular arasındaki yegâne fark, sabit terim değerinin değişiminden (+1 veya -1) kaynaklanmaktadır. Şekil 4.41’de ayırım hattı ve ilave sınır hatları görülmektedir.



Şekil 4.41 Ayırım hattı ve ilave sınır hatları.

İlave sınır hatlarını belirleyen unsurlar, ayırım hattına en yakın veri noktaları olup esasen destek yöney yönteminin ismi de bu noktalara ait yöneylerden gelmektedir. Tahmin edilebileceği üzere diğer veri noktalarının sınır hatlarına etkisi söz konusu değildir. İfade edilen sınır hatlarından hareketle müteakip kısıtlar yazılabilir;

$$y_i = +1 \text{ olması durumunda; } \mathbf{w} \cdot \mathbf{x}_i + b \geq +1$$

$$y_i = -1 \text{ olması durumunda; } \mathbf{w} \cdot \mathbf{x}_i + b \leq -1$$

Görüldüğü üzere kısıtın eşitlik kısıtı olması halinde veri noktası olarak destek yöneyi, eşitsizlik kısıtı olması halinde ise aynı kümeye ait diğer veri noktaları dikkate alınmaktadır. İki farklı eşitsizlik kısıtı, etiket değeri dikkate alınarak tek bir eşitsizlik kısıtına dönüştürülebilir;

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq +1, \quad i = 1, 2, \dots, n \quad (4.110)$$

İfade edilen kısıt denklemleri ve minimizasyon sorununun beraber dikkate alınması ile müteakip ikilenik programlama problemi tanımlanmış olur;



$$\min \frac{1}{2} \|\mathbf{w}\|^2 \quad (4.111)$$

Kısıt

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 \geq 0, \quad i = 1, 2, \dots, n$$

Daha önce ikilenik programlama sorunlarının çözümüne değinilmiş ve çözüm yöntemlerinden birinin de Lagrange çarpanları yaklaşımı olduğu belirtilmişti. Bu bağlamda söz konusu yaklaşım dikkate alınarak Lagrange fonksiyonu;

$$L = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^n \mu_i (y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1) \quad (4.112)$$

şeklinde ve optimizasyon sorunu;

$$\min L = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^n \mu_i y_i(\mathbf{w} \cdot \mathbf{x}_i + b) + \sum_{i=1}^n \mu_i \quad (4.113)$$

$$\mu_i \geq 0$$

biçiminde yazılabilir. Hatırlanacağı üzere burada  $\mu_i$  Lagrange çarpanıdır. Çözüm kapsamında öncelikle kısmi türevler alınarak eğimin sıfır olduğu noktalar elde edilebilir;

$$\frac{\partial L}{\partial \mathbf{w}} = 0 \quad \mathbf{w} = \sum_{i=1}^n \mu_i y_i \mathbf{x}_i \quad (4.114)$$

$$\frac{\partial L}{\partial b} = 0 \quad \sum_{i=1}^n \mu_i y_i = 0 \quad (4.115)$$

Elde edilen bu ifadelerin ana sorunda dikkate alınması ile asli sorunun eşlenik formuna geçiş sağlanabilir;

$$L_E = \frac{1}{2} \mathbf{w} \cdot \mathbf{w} - \sum_{i=1}^n \mu_i y_i(\mathbf{w} \cdot \mathbf{x}_i + b) + \sum_{i=1}^n \mu_i \quad (4.116)$$

$$L_E = \frac{1}{2} \left\{ \sum_{i=1}^n \mu_i y_i \mathbf{x}_i \cdot \sum_{i=1}^n \mu_i y_i \mathbf{x}_i \right\} - \sum_{i=1}^n \mu_i y_i \mathbf{x}_i \sum_{i=1}^n \mu_i y_i \mathbf{x}_i - b \sum_{i=1}^n \mu_i y_i + \sum_{i=1}^n \mu_i \quad (4.117)$$

Buradan hareketle eniyileme süreci tanımlanabilir;



$$\max L_E = \sum_{i=1}^n \mu_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \mu_i y_i \mu_j y_j \mathbf{x}_i \cdot \mathbf{x}_j \quad (4.118)$$

$$\mu_i \geq 0$$

Tanımlanan bu sorunu, matris formunda yeniden ifade etmek mümkündür;

$$\max L_E = \sum_{i=1}^n \mu_i - \frac{1}{2} \boldsymbol{\mu}^T H \boldsymbol{\mu} \quad (4.119)$$

$$\mu_i \geq 0$$

$$H_{ij} = y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \quad (4.120)$$

Burada  $\boldsymbol{\mu}$  ifadesi, Lagrange çarpanlarını içeren yöneyi ifade etmektedir. Eniyileme sorununun, uygun bir optimizasyon algoritması ile ve mesela ikilenik programlama yöntemi ile çözümü neticesinde Lagrange çarpanları elde edilebilir. Elde edilen çarpanların, normal yöney tanımında dikkate alınması ile de  $\mathbf{w}$  yöneyi belirlenmiş olur. Normal yöneyin tespiti bizi, ayırım hattı denkleminde bir adım daha yaklaştırmış olmaktadır. Ancak denklemdaki  $b$  sabit terimi için ilave işlemlerin yapılması gerekir. Hatırlanacağı üzere destek yöney noktalarının;

$$y_s(\mathbf{w} \cdot \mathbf{x}_s + b) = 1, \quad s \in S$$

şartını sağlaması gerekmektedir. Burada  $S$  kümesi, destek yöney noktalarını içeren seti ifade etmektedir. Diğer taraftan normal yöney için daha önce elde edilen;

$$\mathbf{w} = \sum_{i=1}^n \mu_i y_i \mathbf{x}_i$$

ifadesini, destek yöney noktaları için dikkate aldığımızda;

$$y_s \left( \mathbf{x}_s \sum_i \mu_i y_i \mathbf{x}_i + b \right) = 1, \quad s \in S \quad (4.121)$$

eşitliği elde edilir. Bu eşitliğin her iki tarafının  $y_s$  ile çarpılması neticesinde;

$$y_s^2 \left( \mathbf{x}_s \sum_i \mu_i y_i \mathbf{x}_i + b \right) = y_s \quad (4.122)$$

$$y_s^2 = 1 \quad (4.123)$$



denklemine ve buradan da;

$$b = y_s - x_s \sum_i \mu_i y_i x_i, \quad s \in S \quad (4.124)$$

sonucuna ulaşılır. Dikkat edileceği üzere bu yaklaşımla her destek yöneyi için farklı bir sabit terim değeri elde edilir. Bu nedenle ayırım hattında kullanılmak üzere ortalama bir sabit değer tespiti cihetine gidilir;

$$b = \frac{1}{N_s} \left\{ \sum_{s \in S} \left[ y_s - x_s \sum_i \mu_i y_i x_i \right] \right\}, \quad s \in S \quad (4.125)$$

Burada  $N_s$ , destek yöney sayısını ifade etmektedir. Ayırım hattının belirlenmesi sonrasında herhangi bir  $\mathbf{x}$  noktasının hangi kümeyle ait olduğu;

$$y \leftarrow \text{işaret}\{\mathbf{w} \cdot \mathbf{x} + b\} \quad (4.126)$$

ifadesine göre belirlenir. Buna göre  $y$  değerinin pozitif olması halinde  $\mathbf{x}$  noktasının +1 kümesine, negatif olması halinde ise -1 kümesine ait olduğu öngörülebilir.

#### 4.6.2 Esnek sınıflandırma

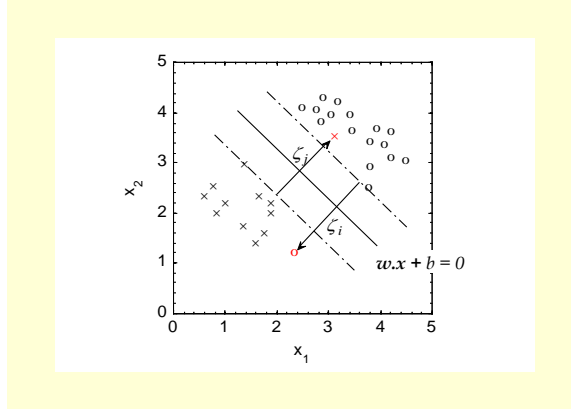
Bir önceki sorunsalda olduğu üzere yine elimizde  $n$  tane veri olduğunu ve her bir verinin de iki kümeden birine ait olduğunu varsayalım;

$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_i, y_i), \dots, (\mathbf{x}_n, y_n)$$

Ancak eldeki veri havuzunun, iki ayrı küme halinde herhangi bir düzlemle kesin bir şekilde birbirinden ayıramayacağını düşünelim. Örneğin Şekil 4.42'den de gözlemlenebileceği üzere, +1 kümesine ait noktalar arasında -1 kümesine ait nokta, benzer şekilde -1 kümesine ait noktalar arasındanda da +1 kümesine ait bir nokta görülmektedir.

Veri dağılımının nispeten karmaşık olduğu söz konusu hallerde sınıflandırma işlemini, destek yöney yönteminin daha önce ifade edilen benzer işlem adımlarını takip ederek yapmak mümkündür. Ancak aykırı noktaların ayırım hattına olan dik uzaklığına bağlı olarak, eniyileme sürecindeki amaç fonksiyonu ve kısıtların yeniden tanımlanması gerekir.





Şekil 4.42 Ayrım hattına aykırı noktaların genel görünümü.

Tabiatıyla hangi noktaların aykırı nokta olduğunu önceden öngörmek mümkün değildir. Bu nedenle tüm veri noktaları baz alınarak amaç fonksiyonu;

$$\min \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \zeta_i \quad (4.127)$$

şeklinde yeniden tanımlanabilir. Buradaki sabit  $C$  katsayısı, sınır esnekliğini belirleyen ve kullanıcı tarafından belirlenen ceza parametresi niteliğindedir. Buna mukabil kısıt denklemleri;

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 + \zeta_i \geq 0, \quad i = 1, 2, \dots, n \quad (4.128)$$

$$\zeta_i \geq 0$$

biçiminde ifade edilir. Tanımlanan optimizasyon sorununun çözümünde, daha önce olduğu gibi yine Lagrange çarpanları yöntemi dikkate alınabilir. Bu bağlamda Lagrange fonksiyonu;

$$L = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \zeta_i - \sum_{i=1}^n \mu_i (y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 + \zeta_i) - \sum_{i=1}^n \alpha_i \zeta_i \quad (4.129)$$

$$\mu_i \geq 0$$

$$\alpha_i \geq 0$$

şeklinde tanımlanabilir. Burada  $\mu_i$  ve  $\alpha_i$  katsayıları, Lagrange çarpanlarını temsil etmektedir. Çözüm kapsamında öncelikle kısmi türevler alınarak eğimin sıfır olduğu noktalar elde edilebilir;



$$\frac{\partial L}{\partial \mathbf{w}} = 0 \quad \mathbf{w} = \sum_{i=1}^n \mu_i \mathbf{y}_i \mathbf{x}_i \quad (4.130)$$

$$\frac{\partial L}{\partial b} = 0 \quad \sum_{i=1}^n \mu_i \mathbf{y}_i = 0 \quad (4.131)$$

$$\frac{\partial L}{\partial \zeta_i} = 0 \quad C = \mu_i + \alpha_i \quad (4.132)$$

Elde edilen bu ifadelerin ana sorunda dikkate alınması ile asli sorunun eşlenik formuna geçiş sağlanabilir;

$$\begin{aligned} \max L_E = & \frac{1}{2} \left\{ \sum_{i=1}^n \mu_i \mathbf{y}_i \mathbf{x}_i \cdot \sum_{i=1}^n \mu_i \mathbf{y}_i \mathbf{x}_i \right\} + C \sum_{i=1}^n \zeta_i - \sum_{i=1}^n \mu_i \mathbf{y}_i \mathbf{x}_i \sum_{i=1}^n \mu_i \mathbf{y}_i \mathbf{x}_i \\ & - b \sum_{i=1}^n \mu_i \mathbf{y}_i + \sum_{i=1}^n \mu_i - \sum_{i=1}^n \mu_i \zeta_i - \sum_{i=1}^n (C - \mu_i) \zeta_i \end{aligned} \quad (4.133)$$

Bu ifadede bazı sadeleştirmelerin ve ara işlemlerin yapılması neticesinde;

$$\begin{aligned} \max L_E = & \sum_{i=1}^n \mu_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \mu_i \mathbf{y}_i \mu_j \mathbf{y}_j \mathbf{x}_i \cdot \mathbf{x}_j \\ & C \geq \mu_i \geq 0 \end{aligned} \quad (4.134)$$

elde edilir. Tanımlanan bu sorunu, matris formunda yeniden ifade etmek mümkündür. Buna göre;

$$H_{ij} = \mathbf{y}_i \mathbf{y}_j \mathbf{x}_i \cdot \mathbf{x}_j \quad (4.135)$$

olmak üzere

$$\begin{aligned} \max L_E = & \sum_{i=1}^n \mu_i - \frac{1}{2} \boldsymbol{\mu}^T H \boldsymbol{\mu} \\ & C \geq \mu_i \geq 0 \end{aligned} \quad (4.136)$$

sonucu yazılabilir. Tıpkı daha önceki sorunsalda olduğu gibi, eniyileme sorununun uygun bir optimizasyon algoritması ile ve mesela ikilenik programlama yöntemi ile çözümü neticesinde  $\mu_i$  Lagrange çarpanları elde edilebilir. Elde edilen çarpanların normal yöney tanımında dikkate



alınması ile  $\mathbf{w}$  yöneyi belirlenmiş olur. Normal yöneyin tespiti sonrasında ise  $b$  sabit teriminin eldesi ve tasnif için dikkate alınan ayırım hattı denkleminde ulaşılmış olur<sup>(56)</sup>.

#### 4.6.3 Dönüşüm temelli sınıflandırma

Daha önce bazı verilerin, buldukları girdi uzayı ortamında doğrusal olarak ayıramayabileceği, bu durumda aynı verilerin, uygun bir fonksiyonla aynı veya bir üst uzaya dönüşümünün sağlanarak doğrusal ayırma gidilebileceği ifade edilmişti. Böyle bir yaklaşımda girdi uzayındaki veri noktası  $\mathbf{x}_i$  yerine çıktı uzayındaki  $\theta(\mathbf{x}_i)$  değeri dikkate alınır. Genellikle  $\theta(\mathbf{x})$  fonksiyonuna eşleştirme<sup>i</sup> veya dönüşüm fonksiyonu adı verilir. Hatırlanacağı üzere kesin veya esnek sınıflandırma işlemlerinde geçen  $H_{ij} = y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j$  matrisi, ikilenik programlama sorununda kilit rol oynamaktaydı. Bu ifadede dikkate alınan ana unsur, veri noktalarının sayısal çarpımıydı. Dolayısıyla, veri yöneylerinin değerinden ziyade sayısal çarpımlarının sonucu çözümü etkilemekteydi. İşte bu düşünceden hareketle, dönüşüm uzayındaki veri konumlarından ziyade dönüşüm uzayı verilerinin sayısal çarpımını baz alan müteakip yaklaşım dikkate alınabilir;

$$H_{ij} = y_i y_j \theta(\mathbf{x}_i) \cdot \theta(\mathbf{x}_j) \quad (4.137)$$

Ancak dönüşüm fonksiyonunun dikkate alınması beraberinde ilave işyükü getirir. Hatta bazı dönüşümlerde dönüşüm uzayındaki karşılıkların bulunması, işyükünden de öte teknik bazı zorlukları da içerebilir. Bu noktada matematiksel bir atlatma<sup>ii</sup> yaklaşımı sergilenebilir. Yaklaşımın çıkış kaynağı, dikey uzaklık veya izdüşüm özelliğidir. Dolayısıyla aynı matris;

$$H_{ij} = y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \quad (4.138)$$

şeklinde yeniden tanımlanabilir. Burada  $K(\mathbf{x}_i, \mathbf{x}_j)$  ifadesi, Kernel fonksiyonu olarak tanımlanmakta olup değişik şekillerde ifade edilebilmekte, ancak mutlaka sayısal çarpım terimleri içermektedir. Bununla beraber literatürde Kernel fonksiyonu tanımlaması yanında başka benzerlik ölçütleri de tanımlanabilmektedir. Konunun daha iyi anlaşılabilmesi amacıyla,  $\mathbf{x} = x_1 \mathbf{i} + x_2 \mathbf{j}$  noktası iki boyutlu girdi uzayındaki yöney olmak üzere, müteakip dönüşüm fonksiyonunu dikkate alabiliriz;

$$\theta(\mathbf{x})^T = [1 \quad \sqrt{2}x_1 \quad \sqrt{2}x_2 \quad x_1^2 \quad x_2^2 \quad \sqrt{2}x_1x_2]$$

<sup>i</sup> Mapping function

<sup>ii</sup> Trick





Görüldüğü üzere dönüşüm uzayı, altı boyutlu bir ortamı ifade etmektedir. Gösterim kolaylığı açısından  $x_i$  ve  $x_j$  gibi iki farklı noktayı,  $\mathbf{x}$  ve  $\mathbf{y}$  noktaları şeklinde simgeselleştirerek girdi uzayında;

$$\begin{aligned}\mathbf{x}^T &= [x_1 \quad x_2] \\ \mathbf{y}^T &= [y_1 \quad y_2]\end{aligned}$$

dönüşüm uzayında ise;

$$\begin{aligned}\theta(\mathbf{x})^T &= [1 \quad \sqrt{2}x_1 \quad \sqrt{2}x_2 \quad x_1^2 \quad x_2^2 \quad \sqrt{2}x_1x_2] \\ \theta(\mathbf{y})^T &= [1 \quad \sqrt{2}y_1 \quad \sqrt{2}y_2 \quad y_1^2 \quad y_2^2 \quad \sqrt{2}y_1y_2]\end{aligned}$$

olarak ifade edebiliriz. Dönüşüm uzayındaki nokta karşılıklarının elde edilebilmesi için her bir yöneye ait altı matematiksel terimin hesaplanması ve akabinde de bu yöneylerin sayısal çarpıma tabi tutulması gerekir;

$$\theta(\mathbf{x})^T \cdot \theta(\mathbf{y}) = [1 \quad \sqrt{2}x_1 \quad \sqrt{2}x_2 \quad x_1^2 \quad x_2^2 \quad \sqrt{2}x_1x_2] \cdot \begin{bmatrix} 1 \\ \sqrt{2}y_1 \\ \sqrt{2}y_2 \\ y_1^2 \\ y_2^2 \\ \sqrt{2}y_1y_2 \end{bmatrix}$$

$$\theta(\mathbf{x})^T \cdot \theta(\mathbf{y}) = 1 + 2x_1y_1 + 2x_2y_2 + x_1^2y_1^2 + x_2^2y_2^2 + 2x_1x_2y_1y_2$$

İfade edilen bu işlemlerin yerine, uygun bir Kernel fonksiyonu tanımlanarak aynı sonuca ulaşmak mümkündür. Örneğin Kernel fonksiyonunu;

$$K(\mathbf{x}, \mathbf{y}) = (1 + \mathbf{x} \cdot \mathbf{y})^2$$

şeklinde tanımlayarak;

$$K(\mathbf{x}, \mathbf{y}) = \left[ 1 + [x_1 \quad x_2] \cdot \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \right]^2 = (1 + x_1y_1 + x_2y_2)^2$$

işlemleri nihayetinde;

$$K(\mathbf{x}, \mathbf{y}) = 1 + 2x_1y_1 + 2x_2y_2 + x_1^2y_1^2 + x_2^2y_2^2 + 2x_1x_2y_1y_2$$

sonucuna ulaşılabilir.



Dönüşüm uzayı noktalarının sayısal çarpımı yerine, girdi uzayı noktalarının sayısal çarpımına dayalı uygun bir Kernel fonksiyonu ile aynı sonuca ulaşılabilmektedir. İlave olarak çok daha az sayıda matematiksel işlemle aynı sonuca ulaşılabildiği de kolayca gözlemlenebilmektedir. Kernel fonksiyonu yaklaşımı ile soruna çözüm bulma şansı yakalanmakla beraber, dönüşüm fonksiyonu ve ona uygun bir Kernel fonksiyonu tanımlamak her zaman mümkün olmayabilir. Genellikle tecrübeye dayalı olarak, deneme ve yanılma ile seçilen Kernel fonksiyonları arasında müteakip tanımlamalar dikkate alınabilir;

Doğrusal Kernel	$K(x_i, x_j) = x_i \cdot x_j$
Polinomial Kernel	$K(x_i, x_j) = (a + x_i \cdot x_j)^b$
Radyal Kernel	$K(x_i, x_j) = e^{-\left\{\frac{\ x_i - x_j\ ^2}{2\sigma^2}\right\}}$
Sigmoid Kernel	$K(x_i, x_j) = \tanh(ax_i \cdot x_j - b)$

Görüldüğü üzere seçilen fonksiyonlar doğrusal olmayan karakterdedirler. Dönüşüm temelli sınıflandırmada dikkate alınan işlem adımları, önceki sorunlarda dikkate alınan işlem adımları ile aynıdır. Kısaca ifade etmek gerekirse;

- Yaklaşımın ilk adımında,  $\theta(x)$  dönüşüm fonksiyonu ile  $K(x_i, x_j)$  Kernel fonksiyonunun seçilmesi gerekir.

- Akabinde  $H$  veya diğer adı ile Kernel matrisinin elemanları hesaplanır;

$$H_{ij} = y_i y_j K(x_i, x_j)$$

- Üçüncü adımda, ayrık kümeye aykırı noktaların etkisini dikkate almak için öngörülen  $C$  ceza parametre değeri seçilir.

- Müteakip adımda Lagrange çarpanlarının elde edilmesi için ikilenik formdaki eniyileme sorunu tanımlanarak çözüme gidilir;



$$\max L_E = \sum_{i=1}^n \mu_i - \frac{1}{2} \boldsymbol{\mu}^T H \boldsymbol{\mu}$$

$$C \geq \mu_i \geq 0$$

- Elde edilen katsayılardan hareketle normal yöney belirlenir;

$$\mathbf{w} = \sum_{i=1}^n \mu_i y_i \boldsymbol{\theta}(\mathbf{x}_i)$$

■ Diğer taraftan ayırım hattı veya düzlem denklemindeki  $b$  sabit teriminin elde edilmesi için öncelikle destek yöney noktalarının belirlenmesi gerekir. Söz konusu belirlemede dikkate alınan ölçüt, Lagrange katsayılarının değeri olup  $C \geq \mu_i > 0$  şartını sağlayan indis numaraları kaydedilerek destek yöney seti elde edilmiş olur. Söz konusu setin belirlenmesi neticesinde;

$$\bar{b} = \frac{1}{N_s} \left\{ \sum_{s \in S} [y_s - \boldsymbol{\theta}(\mathbf{x}_s) \sum_i \mu_i y_i \boldsymbol{\theta}(\mathbf{x}_i)] \right\}, \quad s \in S$$

terimi hesaplanmış olur.

- Nihayetinde herhangi bir  $\mathbf{x}$  noktasının hangi kümeye ait olduğu;

$$y \leftarrow \text{işaret}\{\mathbf{w} \cdot \boldsymbol{\theta}(\mathbf{x}) + b\}$$

ifadesine göre belirlenir.

■ Diğer taraftan bazı uygulamalarda normal yöneyin doğrudan hesaplanmasından ziyade daha pratik bir yaklaşım olarak doğrudan ayırım hattı denklemi hesaplanır. Zira katsayıların elde edilmesi sonrasında normal yöneyin açık hali dikkate alınarak;

$$y \leftarrow \text{işaret}\left\{ \sum_{i=1}^n \mu_i y_i \boldsymbol{\theta}(\mathbf{x}_i) \cdot \boldsymbol{\theta}(\mathbf{x}) + b \right\} \quad \text{veya} \quad y \leftarrow \text{işaret}\left\{ \sum_{i=1}^n \mu_i y_i K(\mathbf{x}_i, \mathbf{x}) + b \right\}$$

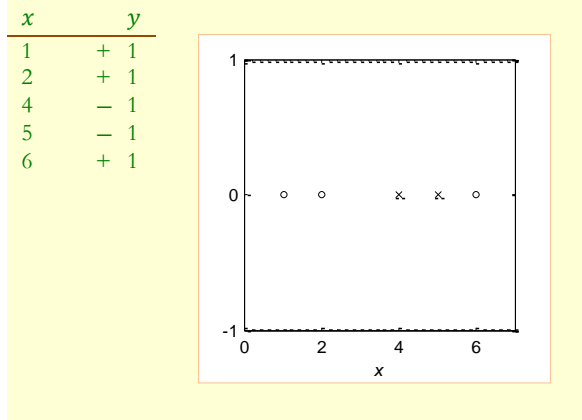
yazılabilir.





### Örnek uygulama

Elimizde beş nokta ve bu noktaların üyesi olduğu küme bilgisi verisini dikkate alalım<sup>(57)</sup>;



Boyutsal olarak bir (1) boyutlu ve veri sayısı olarak da toplamda beş nokta içermesine rağmen veri havuzu, doğrusal olarak tasnif edilemez görünmektedir. Girdi uzayındaki söz konusu sorunun çözümü bağlamında uygun bir dönüşüm fonksiyonunun ( $\theta(\mathbf{x})$ ) bulunduğunu ve bu fonksiyona yönelik olarak da polinomial Kernel fonksiyonunun tanımlandığını düşünelim;

$$K(\mathbf{x}_i, \mathbf{x}_j) = (1 + \mathbf{x}_i \cdot \mathbf{x}_j)^2$$

Ayrıca  $C$  ceza parametre değeri olarak 100 değerini dikkate alalım. Tanımlanan söz konusu fonksiyon ve değer bağlamında eniyileme sorununu;

$$\max L_E = \sum_{i=1}^5 \mu_i - \frac{1}{2} \boldsymbol{\mu}^T H \boldsymbol{\mu}$$

Kısıt

$$100 \geq \mu_i \geq 0$$

$$\sum_{i=1}^5 \mu_i y_i = 0$$

şeklinde ifade etmek mümkündür. Hatırlanacağı üzere burada  $H$  matrisi;



$$H_{ij} = y_i y_j (1 + \mathbf{x}_i \cdot \mathbf{x}_j)^2$$

denklemlerle belirlenmektedir. Amaç fonksiyonunun açık halde yazılması halinde;

$$\max L_E = \mu_1 + \mu_2 + \mu_3 + \mu_4 + \mu_5 - \frac{1}{2} \left( \begin{aligned} &\mu_1 \mu_1 y_1 y_1 (1 + \mathbf{x}_1 \cdot \mathbf{x}_1)^2 + \mu_1 \mu_2 y_1 y_2 (1 + \mathbf{x}_1 \cdot \mathbf{x}_2)^2 + \mu_1 \mu_3 y_1 y_3 (1 + \mathbf{x}_1 \cdot \mathbf{x}_3)^2 + \mu_1 \mu_4 y_1 y_4 (1 + \mathbf{x}_1 \cdot \mathbf{x}_4)^2 + \mu_1 \mu_5 y_1 y_5 (1 + \mathbf{x}_1 \cdot \mathbf{x}_5)^2 + \dots \\ &+ \mu_i \mu_1 y_i y_1 (1 + \mathbf{x}_i \cdot \mathbf{x}_1)^2 + \mu_i \mu_2 y_i y_2 (1 + \mathbf{x}_i \cdot \mathbf{x}_2)^2 + \mu_i \mu_3 y_i y_3 (1 + \mathbf{x}_i \cdot \mathbf{x}_3)^2 + \mu_i \mu_4 y_i y_4 (1 + \mathbf{x}_i \cdot \mathbf{x}_4)^2 + \mu_i \mu_5 y_i y_5 (1 + \mathbf{x}_i \cdot \mathbf{x}_5)^2 + \dots \\ &+ \mu_5 \mu_1 y_5 y_1 (1 + \mathbf{x}_5 \cdot \mathbf{x}_1)^2 + \mu_5 \mu_2 y_5 y_2 (1 + \mathbf{x}_5 \cdot \mathbf{x}_2)^2 + \mu_5 \mu_3 y_5 y_3 (1 + \mathbf{x}_5 \cdot \mathbf{x}_3)^2 + \mu_5 \mu_4 y_5 y_4 (1 + \mathbf{x}_5 \cdot \mathbf{x}_4)^2 + \mu_5 \mu_5 y_5 y_5 (1 + \mathbf{x}_5 \cdot \mathbf{x}_5)^2 \end{aligned} \right)$$

Kısıt

$$\begin{aligned} 100 &\geq \mu_1 \geq 0 \\ 100 &\geq \mu_2 \geq 0 \\ 100 &\geq \mu_3 \geq 0 \\ 100 &\geq \mu_4 \geq 0 \\ 100 &\geq \mu_5 \geq 0 \\ \mu_1 + \mu_2 - \mu_3 - \mu_4 + \mu_5 &= 0 \end{aligned}$$

denklemi ve kısıtları kolayca yazılabilir. Eldeki verilerin eniyileme denkleminde yerine konulması halinde amaç fonksiyonu;

$$\begin{aligned} \max L_E &= \mu_1 + \mu_2 + \mu_3 + \mu_4 + \mu_5 \\ &- \frac{1}{2} \{ 4\mu_1^2 + 25\mu_2^2 + 289\mu_3^2 + 676\mu_4^2 + 1369\mu_5^2 + 18\mu_1\mu_2 - 50\mu_1\mu_3 - 72\mu_1\mu_4 \\ &+ 98\mu_1\mu_5 - 162\mu_2\mu_3 - 242\mu_2\mu_4 + 338\mu_2\mu_5 + 882\mu_3\mu_4 - 1250\mu_3\mu_5 - 1922\mu_4\mu_5 \} \end{aligned}$$

halini alır. İfade edilen amaç fonksiyonu ve kısıtların beraberce dikkate alınması ve uygun bir eniyileme ve mesela ikilenik programlama yöntemi kullanılması sonrasında çözüm olarak;

$$\boldsymbol{\mu}^T = [0 \quad 2.5 \quad 0 \quad 7.333 \quad 4.833]$$

yöneyle elde edilebilir. Buna göre destek yöneylerini  $\mathbf{x}_2$ ,  $\mathbf{x}_4$  ve  $\mathbf{x}_5$  olarak ifade etmek mümkündür. Zira bu yöneylere ait Lagrange katsayı değerleri sıfırdan farklıdır. Hatırlanacağı üzere ayrım hattını belirleyen unsurlar, veri havuzundaki destek yöneyleriydi, veri havuzundaki diğer noktaların ayrım hattına herhangi bir etkisi söz konusu değildi. Katsayı değerlerinin eldesi sonrasında ayrım hattına ait fonksiyonu<sup>i</sup>;

$$y = \sum_{i=1}^5 \mu_i y_i (1 + \mathbf{x}_i \cdot \mathbf{x})^2 + b$$

ifadesine göre;

<sup>i</sup>Bazı yayınlarda “discriminant function” olarak tanımlanır.





$$y = 2.5 \times (+1) \times (1 + 2x)^2 + 7.333 \times (-1) \times (1 + 5x)^2 + 4.833 \times (+1) \times (1 + 6x)^2$$

$$y = 0.6667x^2 - 5.333x + b$$

biçiminde elde edebiliriz. Denklemde geçen  $b$  değerini, destek yöney ve bu yöneylere ait küme etiketi değerlerinden hareketle bulmak mümkündür. Örneğin  $x_2$  destek yöneyinden hareketle;

$$+1 = 0.6667(2)^2 - 5.333 \times 2 + b$$

$$b = 9$$

değerine kolayca ulaşılabilir. Benzer yaklaşım diğer destek yöney noktaları için de geçerlidir. Hülasa, ayırım hattı denklemi;

$$y = 0.6667x^2 - 5.333x + 9$$

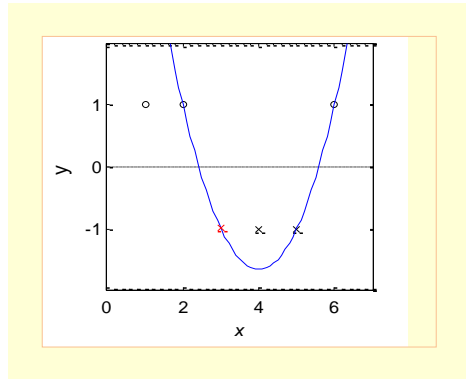
olup herhangi bir  $x$  noktasının ve mesela  $x = 3$  noktasının hangi kümeye ait olduğu hususu;

$$y = \text{işaret}(0.6667 \times 3^2 - 5.333 \times 3 + 9)$$

$$y = \text{işaret}(-0.9996)$$

$$y \leftarrow (-)$$

şeklinde belirlenebilir. Müteakip grafikte ayırım hattına ait denklem fonksiyonu ve örnek noktanın konumu görülmektedir;



Şekil 4.43 Ayırım hattı ve örnek nokta (kırmızı renkli  $\times$  işareti) görünümü.



1B ortamda tasnif modeli eldesi.

**MATLAB uygulaması**

```
% x ve y verisi olarak örnek uygulamadaki veri havuzu dikkate alınmıştır.  
clc; clear  
options = optimset('Algorithm','sqp');  
  
m0 = [0;0;0;0;0]; % başlangıç noktası  
Aeq = [1 1 -1 -1 1]; % eşitlik kısıtı katsayıları  
beq = 0 ; % eşitlik kısıtı sabiti  
lb = [0;0;0;0;0]; % alt sınır  
up = [100;100;100;100;100]; % üst sınır  
  
[m,fval] = fmincon(@objfun,m0,[],[],Aeq, beq, lb, up)  
  
% doruklaştırma amaçlı fonksiyon tanımı  
function f = objfun(m)  
f = -(m(1)+m(2)+m(3)+m(4)+m(5)-0.5*(4*m(1)^2+25*m(2)^2+289*m(3)^2+676*m(4)^2.0 +  
1369*m(5)^2+18*m(1)*m(2)-50*m(1)*m(3)-72*m(1)*m(4)+98*m(1)*m(5)-162*m(2)*m(3)-  
242*m(2)*m(4)+338*m(2)*m(5)+882*m(3)*m(4)-1250*m(3)*m(5)-1922*m(4)*m(5)) ;
```



Sonuç

```
m =  
0.0000  
2.5000  
0.0000  
7.3334  
4.8334  
  
fval =  
-7.3333
```





### Çiçeklerin sınıflandırılması



MATLAB uygulaması



Ebat verileri (xdata)		Küme verileri (group)	Ebat verileri (xdata)		Küme verileri (group)
<i>h</i>	<i>w</i>		<i>h</i>	<i>w</i>	
4.7000	1.4000	'versicolor'	5.1000	1.9000	'virginica'
4.0000	1.3000	'versicolor'	5.8000	2.2000	'virginica'
4.7000	1.6000	'versicolor'	6.3000	1.8000	'virginica'
3.9000	1.4000	'versicolor'	5.1000	2.0000	'virginica'
4.0000	1.0000	'versicolor'	5.0000	2.0000	'virginica'
4.4000	1.4000	'versicolor'	5.5000	1.8000	'virginica'
4.5000	1.5000	'versicolor'	5.0000	1.5000	'virginica'
4.0000	1.3000	'versicolor'	6.7000	2.0000	'virginica'
4.3000	1.3000	'versicolor'	6.0000	1.8000	'virginica'
5.0000	1.7000	'versicolor'	5.6000	2.1000	'virginica'
3.8000	1.1000	'versicolor'	6.4000	2.0000	'virginica'
5.1000	1.6000	'versicolor'	5.6000	1.4000	'virginica'
4.7000	1.5000	'versicolor'	5.5000	1.8000	'virginica'
4.0000	1.3000	'versicolor'	5.6000	2.4000	'virginica'
4.0000	1.2000	'versicolor'	5.9000	2.3000	'virginica'
4.2000	1.2000	'versicolor'	5.0000	1.9000	'virginica'
3.0000	1.1000	'versicolor'	5.1000	1.8000	'virginica'

Çizelge 4.3 Bitkilere ait ebat ve tür tanımlama verileri.



Şekil 4.44 Çiçek türleri.

Botanikte zambak veya süsen<sup>i</sup> çiçeği olarak bilinen çiçeğin değişik türleri söz konusu olup türler arası ayırım, genellikle taç yaprağının<sup>ii</sup> genişlik ve yükseklik mesafelerine göre yapılmaktadır. Zambak çiçeği,

<sup>i</sup> Iris

<sup>ii</sup> Petal



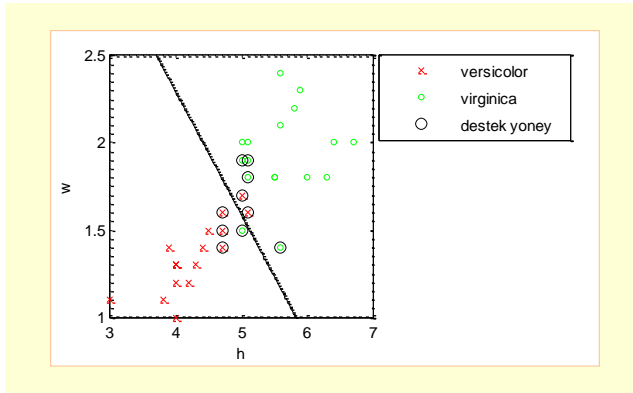




istatistiksel çalışmalarda tarihi öneme sahip bir örnek olup 1936 yılında ünlü biyolog R. A. Fisher tarafından kaleme alınan “Sınıflandırma sorunlarında çoklu ölçümlerin kullanımı”<sup>58)</sup> başlıklı makalede de dikkate alınan veri setlerinden biriydi. Çizelge 4.3’te “iris versicolor” ve “iris virginica” bitki türlerine ait yükseklik ( $h$ ) ve genişlik ( $w$ ) mesafeleri verilmiştir. Buna göre destek yöney yöntemi kullanılarak doğrusal bir tasnife gitmek mümkündür;

### Komut

```
>> svmStruct = svmtrain(xdata,group,'showplot',true);
```



Örnek noktanın tasnifi:  $x^T = [5 \ 2]$  noktası ve tasnif denklemleri değerleri;

```
>> species = svmclassify(svmStruct,[5 2])
```

```
species =  
'virginica'
```

```
>> svmStruct.Alpha'
```

```
ans =  
0.2909 1.000 1.000 1.000 1.000 -0.2909 -1.000 -1.000 -1.000 -1.000
```

```
>> svmStruct.Bias
```

```
ans =  
-0.0101
```



<sup>i</sup>“The use of multiple measurements in taxonomic problems.”



#### 4.6.4 Destek yöney regresyon

Destek yöney yaklaşımı bağlamında doğrusal veya doğrusal olmayan sınıflandırma yöntemleri, uygun tanımlamalarla doğrusal veya doğrusal olmayan eğri uydurma veya diğer ifadesi ile vekil model amaçlı yöntemlere de dönüştürülebilir. Nitekim sınıflandırma yerine model geliştirme maksatlı destek yöney yöntemi seçeneği, 1996 yılında yine V. N. Vapnik ve arkadaşları tarafından gündeme getirilmiştir<sup>(62)</sup>. Destek yöney regresyon<sup>i</sup> yöntemi olarak adlandırılan yaklaşıma göre elimizde  $n$  tane veri olduğunu düşünelim;

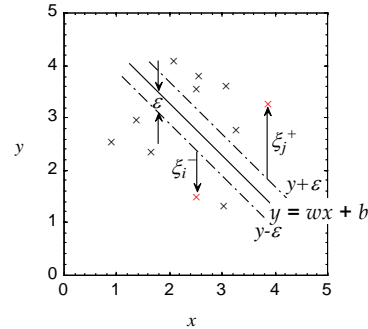
$$(x_1, t_1), (x_2, t_2), \dots, (x_i, t_i), \dots, (x_n, t_n) \quad (4.139)$$

$$x_i \in R^D, \quad t_i \in R$$

Burada  $x_i$   $i$ 'nci verinin konumunu gösteren  $D$  boyutlu yöneyi,  $t_i$  aynı yöneye ait gerçek sayı niteliğindeki çıktı değerini belirtmektedir. Model kurgulamadaki ana gaye;

$$w \cdot x + b = y \quad (4.140)$$

denkleminin elde edilmesidir. Ancak tahmin edilebileceği üzere veri havuzundaki tüm noktaları üzerinde barındırabilecek bir doğrunun eldesi pek mümkün değildir. Bu nedenle doğru hattına olan mesafeyi asgari seviyeye indirebilecek bir eniyileme süreci ile optimal hattın eldesi hedeflenir. Bu bağlamda iki farklı mesafe tanımı yapılır. Bunlardan ilki, kullanıcı tarafından belirlenen tolerans ( $\epsilon$ ), ikincisi ise tolerans dışı mesafedir ( $\xi$ ).

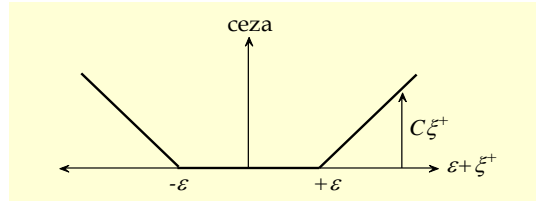


Şekil 4.45 Doğrusal regresyon hattı ve bu hatta göre tanımlanan tolerans ( $\epsilon$ ) ile tolerans dışı mesafe ( $\xi$ ) değerleri.

<sup>i</sup>Support vector regression (SVR)



Şekil 4.45'ten de görülebileceği üzere tolerans mesafesi,  $y$  doğrusu orta hat olmak üzere  $\pm\varepsilon$  mesafesidir. Veri havuzundaki herhangi bir noktanın söz konusu  $\pm\varepsilon$  bant aralığında olması halinde eniyileme sürecine yönelik herhangi bir ceza değeri dikkate alınmaz. Ancak bahse konu bant aralığı dışındaki veri noktaları, bant aralığı dışı mesafesine göre belli bir ceza değerine maruz bırakılır. Söz konusu mesafe, tolerans dışı mesafe olarak tanımlanır. Regresyon hattının bir tarafındaki veri noktasının bant aralığı sınırına olan mesafesi  $\xi^+$  olarak tanımlanırken, diğer tarafındaki veri noktasının bant aralığı sınırına olan mesafesi  $\xi^-$  olarak ifade edilir. Müteakip şekilde tolerans bant aralığı ve ona bağlı olarak amaç fonksiyonuna eklenen ceza değeri değişimi görülmektedir;



Şekil 4.46 Tolerans ve ceza değeri değişimi.

İfade edilen gaye çerçevesinde doğrusal model eldesi için tanımlanan eniyileme sorununa ait amaç fonksiyonunu;

$$\min \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n (\xi_i^+ + \xi_i^-) \quad (4.141)$$

$$\begin{aligned} \xi_i^+ &\geq 0 \\ \xi_i^- &\geq 0 \end{aligned}$$

şeklinde tanımlanabiliriz. Burada  $C$  katsayısı, kullanıcı tarafından belirlenen sabit ceza parametresini göstermektedir. Buna mukabil kısıt denklemleri, hedef fonksiyon değeri ile model fonksiyon değeri arasındaki farka dayalı olarak tarif edilebilir. Şöyle ki;  $t_i$  hedef noktasının, doğrusal modelin yukarı tarafında olması halinde;

$$t_i \leq y_i + \varepsilon + \xi_i^+ \quad (4.142)$$

$t_i$  hedef noktasının, doğrusal modelin aşağı tarafında kalması halinde;

$$t_i \geq y_i - \varepsilon - \xi_i^- \quad (4.143)$$



eşitsizlikleri geçerlidir. İfade edilen amaç fonksiyonu ile kısıtların beraberce dikkate alınması halinde eniyileme sorununu;

$$\min \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n (\xi_i^+ + \xi_i^-) \quad (4.144)$$

Kısıt

$$\begin{aligned} t_i &\leq y_i + \varepsilon + \xi_i^+ \\ t_i &\geq y_i - \varepsilon - \xi_i^- \\ \xi_i^+ &\geq 0 \\ \xi_i^- &\geq 0 \end{aligned}$$

şeklinde tarif etmek mümkündür. Tanımlanan optimizasyon sorununun çözümünde, daha önce olduğu gibi yine Lagrange çarpanları yöntemi dikkate alınabilir. Bu bağlamda Lagrange fonksiyonu;

$$\begin{aligned} L &= \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n (\xi_i^+ + \xi_i^-) - \sum_{i=1}^n \mu_i^+ (\varepsilon + \xi_i^+ + y_i - t_i) \\ &\quad - \sum_{i=1}^n \mu_i^- (\varepsilon + \xi_i^+ - y_i + t_i) - \sum_{i=1}^n \lambda_i^+ \xi_i^+ - \sum_{i=1}^n \lambda_i^- \xi_i^- \\ \mu_i^+, \mu_i^-, \lambda_i^+, \lambda_i^- &\geq 0 \end{aligned} \quad (4.145)$$

şeklinde tanımlanabilir. Burada  $\mu_i^+, \mu_i^-, \lambda_i^+, \lambda_i^-$  katsayıları Lagrange çarpanlarını ifade etmektedir. Ayrıca;

$$\mathbf{w} \cdot \mathbf{x}_i + b = y_i \quad (4.146)$$

denklemini de bahse konu denklem setine eklenebilir. Çözüm kapsamında öncelikle kısmi türevler alınarak eğimin sıfır olduğu noktalar elde edilebilir;

$$\frac{\partial L}{\partial \mathbf{w}} = 0 \quad \mathbf{w} = \sum_{i=1}^n (\mu_i^+ - \mu_i^-) \mathbf{x}_i \quad (4.147)$$

$$\frac{\partial L}{\partial b} = 0 \quad \sum_{i=1}^n (\mu_i^+ - \mu_i^-) = 0 \quad (4.148)$$

$$\frac{\partial L}{\partial \xi_i^+} = 0 \quad C = \mu_i^+ + \lambda_i^+ \quad (4.149)$$



$$\frac{\partial L}{\partial \xi_i^-} = 0 \quad C = \mu_i^- + \lambda_i^- \quad (4.150)$$

Elde edilen bu ifadelerin ana sorunda dikkate alınması ile asli sorunun eşlenik formuna geçiş sağlanabilir;

$$\begin{aligned} \max L_E &= \sum_{i=1}^n (\mu_i^+ - \mu_i^-) t_i - \varepsilon \sum_{i=1}^n (\mu_i^+ - \mu_i^-) - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n (\mu_i^+ - \mu_i^-) (\mu_j^+ - \mu_j^-) \mathbf{x}_i \cdot \mathbf{x}_j & (4.151) \\ \text{Kısıt} & \\ & \sum_{i=1}^n (\mu_i^+ - \mu_i^-) = 0 \\ & 0 \leq \mu_i^+ \leq C \\ & 0 \leq \mu_i^- \leq C \end{aligned}$$

Eşlenik sorunun, uygun bir optimizasyon yöntemi ile ve mesela ikilenik programlama yöntemi ile çözülmesi halinde  $\mu_i^+$  ve  $\mu_i^-$  katsayıları, akabinde ağırlık yöneyi kolayca elde edilebilir. Dolayısıyla herhangi bir  $\mathbf{x}$  noktasına ait  $y$  değeri;

$$y = \sum_{i=1}^n (\mu_i^+ - \mu_i^-) \mathbf{x}_i \cdot \mathbf{x} + b \quad (4.152)$$

denkleminde tahmin edilebilir. Denkleminde geçen  $b$  sabiti;

$$b = t_s - \varepsilon - \sum_{i=1}^n (\mu_i^+ - \mu_i^-) \mathbf{x}_i \cdot \mathbf{x}_s \quad (4.153)$$

eşitliği ile elde edilebilir. Ancak çok sayıda destek yöneyinin varlığı nedeniyle, aynı sabit değer;

$$\bar{b} = \frac{1}{N_s} \left\{ t_s - \varepsilon - \sum_{i=1}^n (\mu_i^+ - \mu_i^-) \mathbf{x}_i \cdot \mathbf{x}_s \right\} \quad (4.154)$$

ortalama yaklaşımı ile belirlenebilir.

#### 4.6.5 Doğrusal olmayan destek yöney regresyon

Daha önce dönüşüm temelli sınıflandırma bahsinde dönüşüm fonksiyonu ve ona uygun Kernel fonksiyonundan bahsedilmişti. Doğrusal olmayan eğri uydurma amacıyla kullanılan destek yöney yaklaşımında da benzer işlem adımlarını takip etmek mümkündür. Buradaki temel



çıkış noktası, tıpkı daha öncesinde olduğu üzere  $\mathbf{x}_i \cdot \mathbf{x}_j$  çarpımı yerine  $K(\mathbf{x}_i, \mathbf{x}_j)$  fonksiyonunun dikkate alınmasıdır. Kısaca ifade etmek gerekirse;

- Yaklaşımın ilk adımında  $\theta(\mathbf{x})$  dönüşüm fonksiyonu ile  $K(\mathbf{x}_i, \mathbf{x}_j)$  Kernel fonksiyonunun seçilmesi gerekir.

- İkinci adımda  $C$  ceza parametre değeri ile  $\varepsilon$  tolerans mesafe değeri seçilir. Müteakiben Lagrange çarpanlarının elde edilmesi için ikilenik formdaki eniyileme sorunu tanımlanarak çözüme gidilir;

$$\max L_E = \sum_{i=1}^n (\mu_i^+ - \mu_i^-) t_i - \varepsilon \sum_{i=1}^n (\mu_i^+ - \mu_i^-) - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n (\mu_i^+ - \mu_i^-) (\mu_j^+ - \mu_j^-) K(\mathbf{x}_i, \mathbf{x}_j) \quad (4.155)$$

Kısıt

$$\sum_{i=1}^n (\mu_i^+ - \mu_i^-) = 0$$

$$0 \leq \mu_i^+ \leq C$$

$$0 \leq \mu_i^- \leq C$$

- Elde edilen katsayılardan hareketle normal yöney belirlenir;

$$\mathbf{w} = \sum_{i=1}^n (\mu_i^+ - \mu_i^-) \theta(\mathbf{x}_i) \quad (4.156)$$

- Diğer taraftan ayırım hattı veya düzlem denklemindeki  $b$  sabit teriminin elde edilmesi için öncelikle destek yöney noktalarının belirlenmesi gerekir. Söz konusu belirlemede dikkate alınan ölçüt, Lagrange katsayılarının değeri olup  $C \geq \mu_i > 0$  ve  $\xi_i = 0$  şartlarını sağlayan indis numaraları kaydedilerek destek yöney seti elde edilmiş olur. Söz konusu setin belirlenmesi neticesinde;

$$\bar{b} = \frac{1}{N_s} \left\{ t_s - \varepsilon - \sum_{i=1}^n (\mu_i^+ - \mu_i^-) K(\mathbf{x}_i, \mathbf{x}_j) \right\} \quad (4.157)$$

terimi hesaplanmış olur.

- Nihayetinde herhangi bir  $\mathbf{x}$  noktasının tahmini çıktı değeri;

$$y = \sum_{i=1}^n (\mu_i^+ - \mu_i^-) K(\mathbf{x}_i, \mathbf{x}) + b \quad (4.158)$$

ifadesine göre belirlenir.





Literatürde destek yöney yöntemlerinin vekil model eldesi amacıyla kullanımına sıkça rastlanmakta olup bazı çalışmalarda diğer vekil model yöntemlerine göre (Kriging, yanıt yüzey yöntemleri ve regresyon gibi) destek yöney yaklaşımının daha hassas ve gürbüz sonuçlar verdiği belirtilmektedir<sup>(63)</sup>. Ancak şunu da ifade edelim ki ilke olarak algoritma etkinliği, sorunun doğasına göre değişebilir.



2B ortamda vekil model eldesi.<sup>(64)</sup>

MATLAB uygulaması

X				Y			
-10.0000	-4.8000	0.4000	5.6000	-0.0544	-0.2075	0.9735	-0.1127
-9.6000	-4.4000	0.8000	6.0000	-0.0182	-0.2163	0.8967	-0.0466
-9.2000	-4.0000	1.2000	6.4000	0.0242	-0.1892	0.7767	0.0182
-8.8000	-3.6000	1.6000	6.8000	0.0665	-0.1229	0.6247	0.0727
-8.4000	-3.2000	2.0000	7.2000	0.1017	-0.0182	0.4546	0.1102
-8.0000	-2.8000	2.4000	7.6000	0.1237	0.1196	0.2814	0.1274
-7.6000	-2.4000	2.8000	8.0000	0.1274	0.2814	0.1196	0.1237
-7.2000	-2.0000	3.2000	8.4000	0.1102	0.4546	-0.0182	0.1017
-6.8000	-1.6000	3.6000	8.8000	0.0727	0.6247	-0.1229	0.0665
-6.4000	-1.2000	4.0000	9.2000	0.0182	0.7767	-0.1892	0.0242
-6.0000	-0.8000	4.4000	9.6000	-0.0466	0.8967	-0.2163	-0.0182
-5.6000	-0.4000	4.8000	10.0000	-0.1127	0.9735	-0.2075	-0.0544
-5.2000	0	5.2000		-0.1699	1.0000	-0.1699	



```
function svrobj = svr_trainer_short(X,Y, c, epsilon, lambda)

% Örnek uygulama

% svrobj = svr_trainer(X,Y,400,0.000000025,'gaussian',0.5);
% y = svrobj.predict(x_test);

% Seçili Kernel: Gaussian

kernel_function = @(x,y) exp(-lambda*norm(x.feature-y.feature,2)^2);
ntrain = size(xdata,1);
for i=1:ntrain
    for j=1:ntrain
        xi(i,j).feature = xdata(i,:);
        xj(i,j).feature = xdata(j,:);
    end
end

% Kernel matrisi oluşumu

M = arrayfun(kernel_function,xi,xj);
M = M + 1/c*eye(ntrain);

% SVM olusumu

options = optimset('Algorithm','interior-point-convex');
H = 0.5*[M zeros(ntrain,3*ntrain); zeros(3*ntrain,4*ntrain)];
figure; imagesc(M); title('Kernel matrisi gorunumu..')
```



```

K(x_i,x_j'); xlabel('veri #'); ylabel('veri #');
lb = [-c*ones(ntrain,1); zeros(ntrain,1); zeros(2*ntrain,1)];
ub = [ c*ones(ntrain,1); 2*c*ones(ntrain,1); c*ones(2*ntrain,1)];
f = [ -ydata; epsilon*ones(ntrain,1); zeros(ntrain,1); zeros(ntrain,1)];
z = quadprog(H,f,[],[],[],[],lb,ub,[],options);
alpha = z(1:ntrain);
figure; stem(alpha); title('SVR Gorunumu');
xlabel('veri #'); ylabel('normal yoney (alpha_i+ - alpha_i-)');

% b sabiti hesabı

for m=1:ntrain
    bmat(m) = ydata(m);
    for n = 1:ntrain
        bmat(m) = bmat(m) - alpha(n)*M(m,n);
    end
    bmat(m) = bmat(m) - epsilon - alpha(m)/c;
end
b = mean(bmat);

% sonuçların kaydı

svrobj.alpha = alpha;
svrobj.b = b;
svrobj.kernel = kernel_function;
svrobj.train_data = xdata;
svrobj.predict = @(x) cellfun(@(u) svr_eval(u),num2cell(x,2));

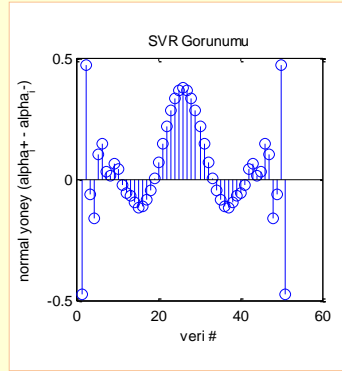
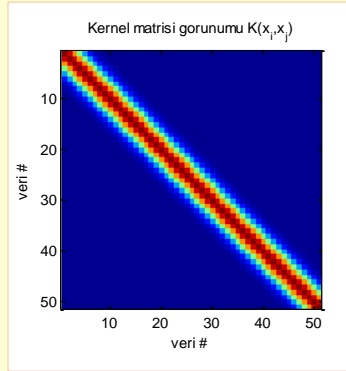
% vekil model değeri

function f = svr_eval(x)
f = 0;
n_predict = size(x,1);
for i=1:n_predict
    sx(i).feature = x(i,:);
end
n_train = size(xdata,1);
for i=1:n_train
    sy(i).feature = xdata(i,:);
end
for i=1:n_train
    f = f + svrobj.alpha(i)*kernel_function(sx(1),sy(i));
end
f = f + b;
f = f/2;
end
end

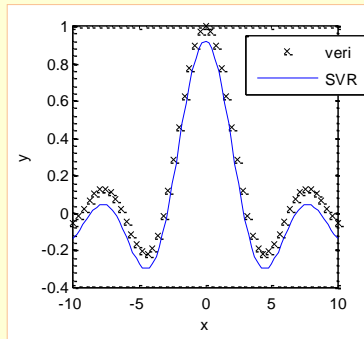
```



```
>> svrobj = svr_trainer_short(X,Y,400,0.000000025,0.5);
```

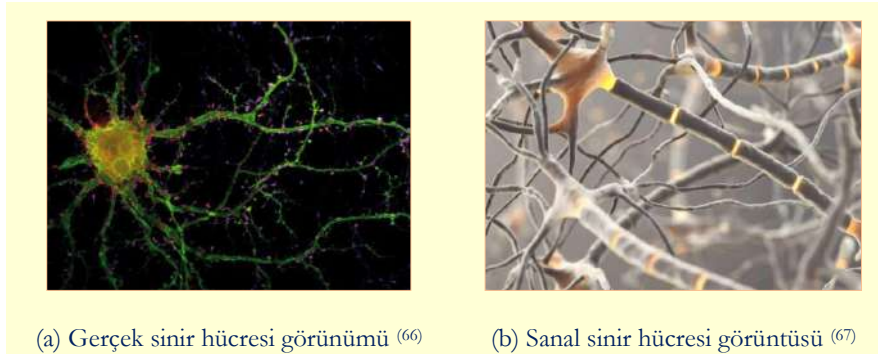


```
>> plot(X,Y,'kx')  
>> hold on  
>> x_test = -10:0.25:10;  
>> x_test = x_test';  
>> y = svrobj.predict(x_test);  
>> plot(x_test,y)
```



#### 4.7 Yapay sinir ağları

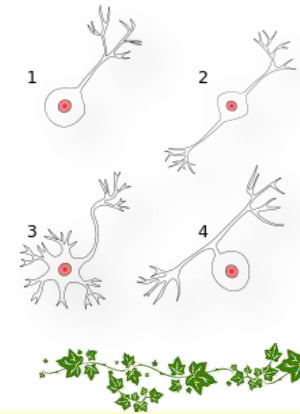
Yapay sinir ağları<sup>i</sup> veya literatürdeki diğer bir tanımlama ile bağlı sistemler<sup>ii</sup>, insan beyninin sorunları çözme yeteneğinden esinlenilerek geliştirilen bir yaklaşımdır. Ancak esinlenme, algoritmik temellerden ziyade biyolojik olgulara dayanır. Beyin ve sinir bilimlerindeki yeni gelişmelere bağlı olarak yapay sinir ağlarında da yeni yaklaşımlar sergilenebilmektedir<sup>(65)</sup>. Bilindiği kadarıyla insan zekâsının kaynağı, beyin ve beyin dâhilindeki elektrokimyasal etkileşimlerdir. Beyin bünyesinde temel hücre türü sinir<sup>iii</sup> hücresi olup müteakip şekilde gerçek ve sanal sinir hücresi görülmektedir;



Şekil 4.47 Sinir hücresi gerçek ve benzeşim görünümleri.

Sinir hücresi, elektriksel olarak uyarılabilen bir hücre tipi olup algıladığı sinyalleri işleme ve neticesinde de iletme yeteneğine sahiptir. Sinir hücreleri arasındaki sinyal iletişimleri, sinapsis<sup>iv</sup> denilen hücreler arası bağlarla gerçekleşir. Söz konusu bağlar sayesinde sinir hücreleri, geniş bağlatılı ağ yapılarını oluştururlar.

Canlılara ve canlılardaki organlara bağlı olarak değişik sinir hücresi yapılarından bahsedilebilir. Bununla beraber Şekil 4.48'den de görülebileceği üzere tek, çift, çok kutuplu ve sanki-tek kutuplu



Şekil 4.48 Sinir hücresi türleri; 1) tek, 2) çift, 3) çok, 4) sanki-tek kutuplu sinir hücresi<sup>(68)</sup>.

<sup>i</sup> Artificial neural networks (ANN)

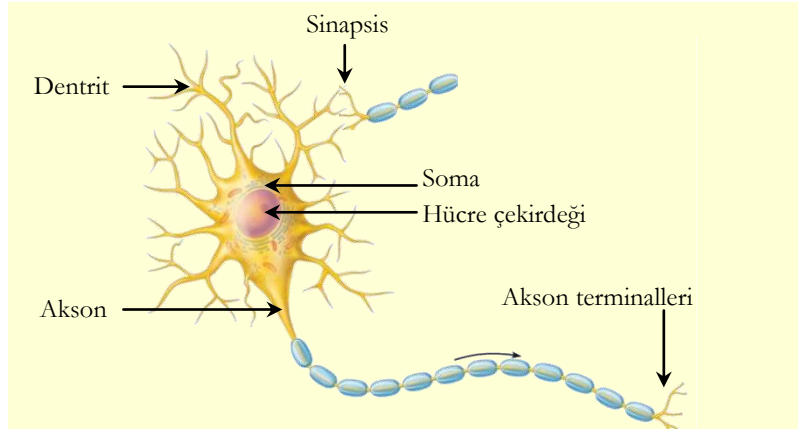
<sup>ii</sup> Connectionist systems

<sup>iii</sup> Neuron

<sup>iv</sup> Synaptic terminals



sinir hücresi türlerinden bahsedilebilir. Şekil 4.49'da tek kutuplu bir sinir hücresine ait temel yapı öğeleri görülmektedir.



Şekil 4.49 Sinir hücresi öğeleri <sup>(69)</sup>.

Buna göre bir sinir hücresinde; hücrenin ana yapısını oluşturan soma<sup>i</sup> ve hücre çekirdeği<sup>ii</sup>, akson<sup>iii</sup> ve dendritler<sup>iv</sup> bulunur. Genellikle soma yekpare bir yapıda, akson ve dendritler ise dallanma şeklinde yapılıdır. Somadan pek çok dendrit dallanabilir, ancak sadece bir akson sürgünü bulunur. Akson ve dendritler diğer sinir hücreleri ile teması sağlarlar. Bahse konu temas noktaları sinapsis<sup>v</sup> olarak adlandırılır. Temas bağlantıları akson-akson, dendrit-dendrit ve akson-dendrit şeklinde olabilir<sup>(68)</sup>.

Bilinen evrendeki en karmaşık yapı olan insan beyninde ortalama  $10^{11}$  civarı sinir hücresi ve her bir sinir hücresinin de ortalama  $7 \times 10^3$  civarı sinapsis bağlantısı bulunmaktadır. Bu bağlamda yaklaşık bağ sayısının  $10^{15}$  civarında olduğu söylenebilir. Günümüzde sinirler arası sinyal üretim ve iletiminin nasıl gerçekleştiği hususu büyük oranda açığa kavuşturulmuş olsa da söz konusu devasa ağ yapısının inanılmaz boyuttaki bilgiyi nasıl değerlendirdiği ve muhakeme sonrasında da

<sup>i</sup> Soma or cell body

<sup>ii</sup> Cell nucleus

<sup>iii</sup> Axon

<sup>iv</sup> Dendrite

<sup>v</sup> Synaptic terminals



karar mekanizmasına dönüştürüldüğü hususu hala gizemini korumaktadır. Ancak üzerinde fikir birliği edilen husus, “iletişim yolu ile kontrol” ilkesinin varlığıdır<sup>(70)</sup>.

Yapay sinir ağları, gerçek sinir ağlarının birebir aynısı olmamakla beraber yapısal bağlamda onlardan esinlenilerek oluşturulmuş organizasyonel yapılardır. Ağ yapısının temelini oluşturan yapay sinir hücresi, bir anlamda matematiksel bir fonksiyon olup bir veya birden fazla girdi verisini alarak (gerçek sinir hücresinde dentritler ile benzeşen yapı) toplama işlemlerine tabi tutar ve akabinde de çıktı verisi üretir, (gerçek sinir hücresinde akson yapısı ile benzeşen oluşum)<sup>(71)</sup>. Yapay sinir hücreleri, kendi aralarında iletişim ağlarına sahip olup iletişimde verinin ağırlıklandırılması mümkündür. Söz konusu ağırlıklandırma işlemi, ağırlık katsayılarının dikkate alınmasıyla sağlanır. Müteakip eşleşme çizelgesinde gerçek sinir hücresi yapısı ile yapay sinir hücresi arasındaki benzeşim görülmektedir;

Gerçek sinir hücresi	Yapay sinir birimi
Dentrit	Yapay sinir hücresine girdi verilerini sağlayan girdi yöneyi hüviyetindedir. Alınan girdi değerlerini ağırlıklandırma özelliğine sahiptir.
Soma	Toparlama işlemi yapan yapı niteliğindedir.
Akson	Yapay sinir hücresinin diğer sinir hücreleri ile iletişimini sağlayan yapı niteliğinde olup ağ özelliğini sağlayan unsurdur.

Yapay sinir ağları, her ne kadar girdi verileri ışığında çıktı verisi üreten sistemler olsa da esas itibari ile karar mekanizmasından ziyade bir eşleştirme sistemi olarak tarif edilebilirler. Dolayısıyla sistem  $n$  boyutlu  $x$  girdi yöneyine karşılık  $m$  boyutlu  $y$  çıktı yöneyi üretecek şekilde tasarlanan  $f: x \rightarrow y$  eşleştirme fonksiyonu niteliğindedir.

Literatürdeki ilk yapay sinir ağı hücresi tasarımı, 1943 yılında W. McCulloch ve W. Pitts tarafından yayımlanan “Sinir aktiviteleri kaynaklı düşünceye dayalı mantıksal hesaplama”<sup>(72)</sup> makalesine dayanır<sup>(72)</sup>. 1940’lı yıllardan günümüze dalgalı bir seyir izlese de yapay sinir ağlarına yönelik literatür gittikçe genişlemekte ve her geçen gün daha karmaşık ve farklı algoritmik yöntemler gündeme gelmektedir. Bununla beraber yapay sinir ağlarının, gerçek sinir ağlarını gerek algoritmik bağlamda ve gerekse de fiziksel anlamda birebir örnek aldığı söylenemez. Hatta günümüzdeki gelişmiş yapay sinir ağlarının milyonlarca ağ elemanı ve bağlarından oluşabildiği

<sup>i</sup> Control through communication

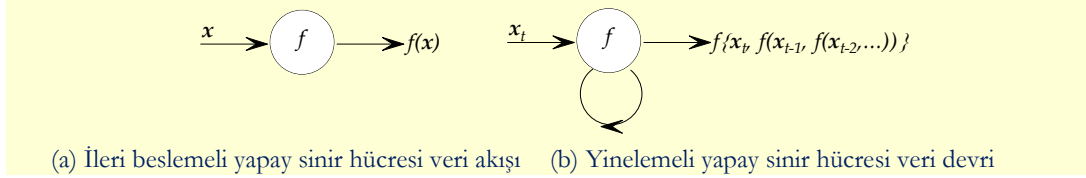
<sup>ii</sup> “A logical calculus of the ideas immanent in nervous activity.”





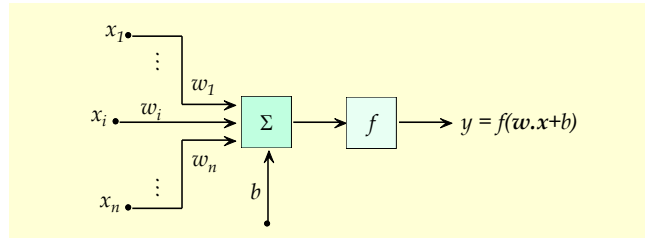
gözlemlense de henüz insan zekâsının çok gerisinde ve belki de bir elma kurdunun hesaplama kapasitesine yakın olduğu söylenebilir<sup>(65)</sup>.

Tipik bir yapay sinir ağının temelde üç özellik ile tanımlanabildiği söylenebilir. Söz konusu nitelikler; yapay sinir hücresinin türü, hücreler arası ilişki kalıbı ve ağırlıklandırma işlemlerinde dikkate alınan ağırlıkların belirlenmesi sürecidir. Hücreler arası örüntünün şekillenmesinde veri akışının önemi büyüktür. Genellikle veri akışı tek ve belirli bir yönde gerçekleşir. Bu tür örüntü yapılarına ileri beslemeli yapay sinir ağları adı verilir. Diğer taraftan verinin aynı hücrede birkaç defa işleme tabi tutulduğu, diğer bir ifade ile çıkış sonrası tekrar girişe yönlendirildiği devirli örüntü yapıları da söz konusu olup bu tür örüntü yapısına sahip ağlara yinelemeli<sup>ii</sup> yapay sinir ağları adı verilir. Müteakip şekillerde bahse konu örüntü yapıları görülmektedir;



Şekil 4.50 Veri akış türleri.

Yapay sinir hücresinin türü bağlamında kastedilen, aktivasyon fonksiyonunun çeşididir. Şekil 4.51'de örnek bir yapay sinir hücresi ve hücre dâhilinde gerçekleştirilen işlemler görülmektedir;



Şekil 4.51 Örnek bir yapay sinir hücresi.

<sup>i</sup> Feed-forwards neural Networks (NN)

<sup>ii</sup> Recursive NN





Şekilden de gözlemlenebileceği üzere öncelikle  $x_i$  girdi verileri,  $w_i$  ağırlık katsayıları ile çarpılmakta ve akabinde de toparlama işlemine tabi tutulmaktadır. Elde edilen değer, eşik değer olarak ifade edilen ilave  $b$  parametresi ile toplanarak hücre için girdi verisine dönüştürülmektedir. Bu veri, hücreye ait  $f$  aktivasyon fonksiyonunda<sup>i</sup> girdi argümanı olarak kullanılır ve sonrasında da fonksiyonel işleme tabi tutularak  $y$  simgesi ile temsil edilen hücre çıktı değeri elde edilmiş olur. Tanımlanan hücre içi işlem adımları, herhangi bir ağın söz konusu olduğu çoklu yapılarda da aynen geçerlidir. Ağ yapılanmasında yapay sinir hücreleri, bir anlamda hesaplama birimleri<sup>ii</sup> olarak adlandırılırlar. Herhangi bir ağ yapısını  $(I, N, O, E)$  alt kümelerinden oluşan bir büyük küme şeklinde düşünebiliriz. Burada;  $I$  girdi verilerini içeren kümeyi,  $N$  hesaplama birimleri niteliğindeki sinir hücrelerini içeren seti,  $O$  çıktı verilerini içeren kümeyi,  $E$  ağırlıklandırılmış ve aynı zamanda yönlendirilmiş bağlantı kenarlarından oluşan seti içermektedir.  $E$  setinin tanımlanmasında diğer setlerin etkisi söz konusu olup set üyeleri  $(u, v, w)$  olarak ifade edildiğinde;

$$\begin{aligned} u &\in I \cup N \\ v &\in N \cup O \\ w &\in E \end{aligned} \quad (4.159)$$

tanımlamaları kolayca yapılabilir. Görüldüğü üzere burada  $u$  ve  $v$  kenar bağlantılarını,  $w$  ise kenar üzerinde geçen değerlerin ağırlık katsayısı çarpanını temsil etmektedir.

Ağ yapısı tanımlamalarında girdi ve çıktı verilerinin olduğu kümeler, hesaplama birimleri niteliğinde olmayıp, sadece verisel nitelik taşırlar. Dolayısıyla bu kümelerin hesaplama algoritmasına doğrudan bir etkisinden bahsedilemez. Ancak sinir hücrelerinin gerek sayısı, gerek örüntü yapısı ve gerekse de aktivasyon fonksiyonu türü, hesaplama işlemlerinde etkin unsurlardır. Hesaplama birimlerini içeren  $N$  seti bazı hallerde  $N_1, N_2, \dots, N_L$  gibi alt set gruplarına bölünebilir. Bu durumda alt setler saklı katmanlar<sup>iii</sup> olarak adlandırılırlar. Aynı terimlerin kullanılması bağlamında girdi verilerinin bulunduğu kümeye girdi katmanı, çıktı verilerinin bulunduğu kümeye de çıktı katmanı<sup>iv</sup> adı verilebilir. Şekil 4.52'de çok katmanlı bir ağ yapısı görülmektedir. Buna göre yapay sinir ağları, ilkesel bağlamda girdi-çıkıtı ikililerinden ve bu verilerin toplamlarından oluşan, esas itibarı ile basit matematiksel fonksiyonlardan oluşmaktadır.

<sup>i</sup> Activation function

<sup>ii</sup> Computing units

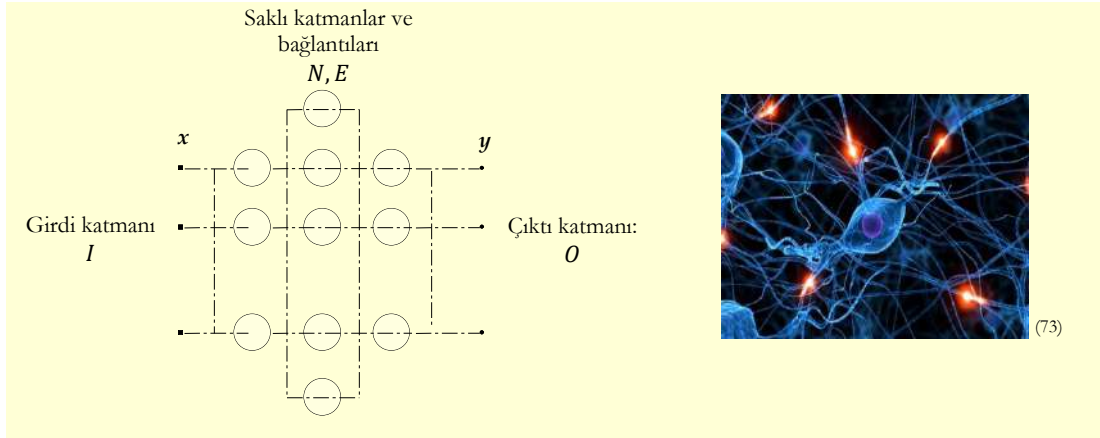
<sup>iii</sup> Hidden layers

<sup>iv</sup> Input and output layers





Hatta yapay sinir ağını, herhangi bir  $n$  argümanlı fonksiyonun tek argümanlı fonksiyonlar ile modellenmesi şeklinde tanımlamak da mümkündür.

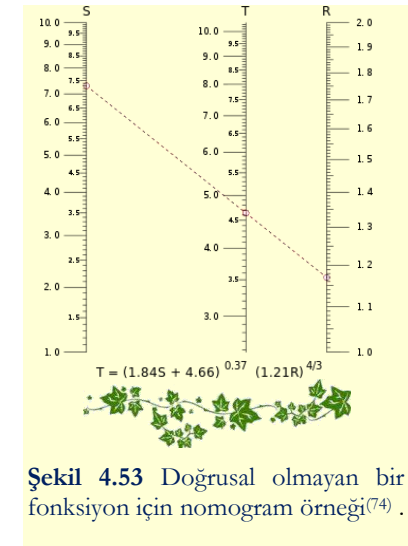


Şekil 4.52 Örnek bir yapay sinir ağı ile sanal sinir hücreleri arası iletişim.

Geçmiş dönemde çok argümanlı fonksiyonların tek argümanlı fonksiyonlarla modellenmesi hususu, matematiksel olarak çözülemeyen sorunlardan sayılmaktaydı. Hatta 1900'lü yıllarda ünlü matematikçi D. Hilbert (1862-1943) tarafından dillendirilen ve o dönemde çözülemeyen ünlü 23 matematiksel problemten birisi de bu hususu içermekteydi. Hilbert'in ifadesi ile "7nci dereceden bir denklemin köklerini nomogram<sup>ii</sup> kullanarak çözmek muhtemelen mümkün değildi". Bu ifade daha net bir şekilde "bahse konu problemin iki argümanlı fonksiyonların sonlu sayıdaki kompozisyonu ile temsil edilemeyeceği" anlamına gelmekteydi. Bu iddia, yapay sinir ağları açısından ciddi sorun doğurmaktaydı, zira bu ifadeye göre herhangi bir  $n$  argümanlı fonksiyonun sonlu sayıdaki yapay sinir hücreleri ile modellenmesi mümkün değildi. Bahse konu modellemenin daha iyi anlaşılabilmesi için daha düşük mertebeden, örneğin ikinci dereceden bir denklemin çözümünün modellenmesi ele alınabilir. Hatırlanacağı üzere  $ax^2 + bx + c$  şeklindeki herhangi bir 2nci dereceden denklemin kökleri;

<sup>i</sup> Argument kavramı, fonksiyonda girdi değişkenlerini temsil etmektedir. Örneğin  $f(x) = 2x$  ifadesinde fonksiyonun değerini değiştirebilen tek değişken  $x$  argümandır.

<sup>ii</sup> "Nomograph", iki argümanlı bir fonksiyonun sonucunun uygun bir ölçekte çizilmiş şekiller kullanılarak elde edilebileceği grafiklerdir.



Şekil 4.53 Doğrusal olmayan bir fonksiyon için nomogram örneği<sup>(74)</sup>.

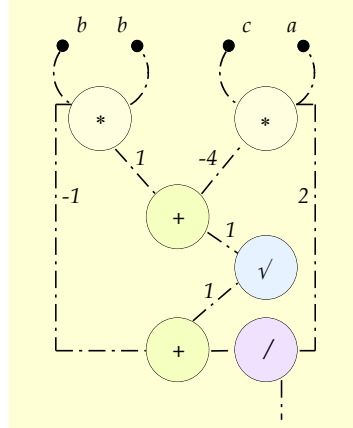






$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

ifadesine göre elde edilebilmektedir. Bu ifadenin ağsal bir yapılanmadaki gösterimi müteakip şekilde yapılabilir;



**Şekil 4.54** Cebrik işlemlere dayalı 2nci dereceden denklemin analitik çözümü için dikkate alınabilecek ağ.

Benzer cebrik işlemler dikkate alınarak 3üncü ve hatta 4üncü dereceden denklemlerin analitik çözümüne de ulaşmak mümkündür. Ancak daha yüksek dereceden denklemlerin modellenmesi 1950'li yıllara kadar mümkün görünmemektedir. Bununla beraber 1957 yılında A. N. Kolomogorov (1903-1987) tarafından söz konusu iddianın doğru olmadığı gösterildi. Nisbeten basit bir örnekle açıklanabilen yaklaşımına göre herhangi bir  $n$  argümanlı fonksiyon, sonlu sayıdaki tek argümanlı fonksiyonların toplanması işlemleri ile modellenebilirdi. Örneğin modellemede zorluk çıkarabilecek işlemlerden birisi olan çarpma işlemini, ifade edilen yaklaşımla;

$$xy = e^{\ln x + \ln y}$$

şeklinde göstermek mümkündür. Yaklaşımın genelleştirilmiş hali ise;

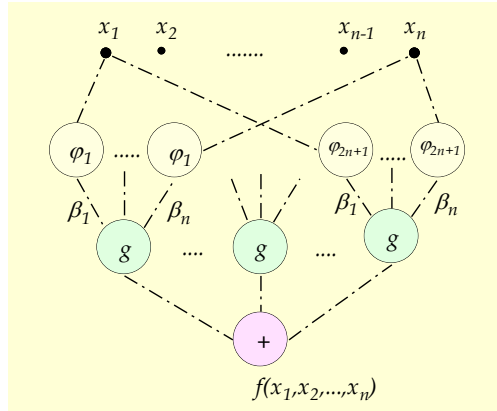
$$f(x) = \sum_{q=1}^{2n+1} g \left\{ \sum_{p=1}^n \beta_p \varphi_q(x_p) \right\}$$

denklemini ile tanımlanabilmektedir.





Burada  $\beta_p$  sabit sayıları ve  $\varphi_q(x_p)$  tek argümanlı fonksiyonları göstermekteydi. Kolomogorov'un ifade ettiği açıklamalar silsilesi, yapay sinir ağlarının kuramsal tabanı açısından büyük önem arz eder. Zira buna göre  $n$  argümanlı herhangi bir fonksiyon, tek argümanlı fonksiyonlardan oluşan sonlu sayıdaki bir ağ yapısı ile ve sadece toplama işlemlerine dayalı olarak modellenebilir. Şekil 4.55'te söz konusu yaklaşımın ağsal görünümü resmedilmektedir.



Şekil 4.55 Çok argümanlı fonksiyon modellemesinde dikkate alınan ağ yapısı<sup>(70)</sup>.

Yapay sinir ağlarında matematiksel işlemlerin toplama operasyonu ile özdeşleştirilmesi, genel kabul gören bir husustur. Ancak hücre girişinde verilerin toplanmasından ziyade toparlanması önemlidir. Toparlama fonksiyonu, bir yapay sinir hücresine ağırlık katsayıları ile çarpılarak gelen girdileri toparlayarak o hücrenin net girdisini hesaplayan bir fonksiyondur. Müteakip çizelgede toplama işlemi yanında kullanılması muhtemel diğer toparlama işlemleri görülmektedir<sup>(75)</sup>;

İşlem	Fonksiyon
Toplama	$net = \sum_{i=1}^n w_i x_i$
Çarpma	$net = \prod_{i=1}^n w_i x_i$
Azami	$net = maks \{w_i x_i\}, \quad i = 1, \dots, n$
Asgari	$net = min \{w_i x_i\}, \quad i = 1, \dots, n$





Toparlama sürecinde toplama işleminin dikkate alınması, ağır doğrusal olmayan sorunlarda yetersiz kalmasına neden olabilir. Diğer taraftan toplarlama yanında burada öne çıkan diđer bir husus, aktivasyon fonksiyonunun varlıđıdır. Yapay sinir ađlarının bir özelliđi olan “dođrusal olmayan modelleme yeteneđi”, uygun seçilen dođrusal olmayan aktivasyon fonksiyonlarından kaynaklanır. Uygun seçimde dikkate alınan özellik, fonksiyonun türevinin kolay hesaplanabilir olmasıdır. Müteakip çizelgede yaygın olarak kullanılan bazı aktivasyon fonksiyonları görölmektedir<sup>(70)</sup>;

İsmi	Tanımı	Şekli
Adım	$y = \begin{cases} 1 & net \geq b \\ 0 & net < b \end{cases}$	
Sigmoid	$y = \frac{1}{1 + e^{-net}}$	
Dođrusal	$y = \sum_{i=1}^n w_i x_i + b$	
Simetrik sigmoid	$y = \frac{1 - e^{-net}}{1 + e^{-net}}$	

**Çizelge 4.4** Aktivasyon fonksiyonu tanımlamaları.

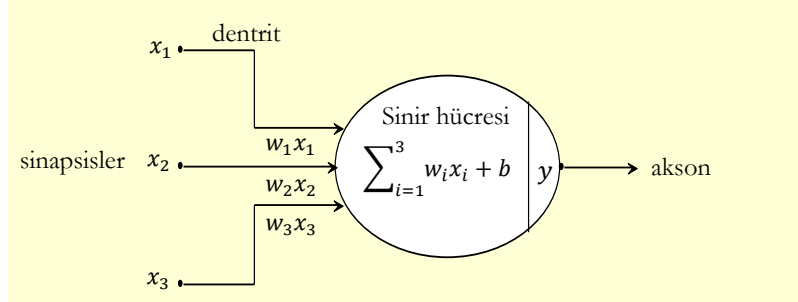
Denklemlerde geçen  $b$  terimi, eşik<sup>i</sup> deđerini temsil etmektedir. Yapay sinir hücresi ve hücreye ait aktivasyon fonksiyonunun daha iyi anlaşılması için müteakip örnek dikkate alınabilir.

<sup>i</sup> Bias





Girdi verisi	Ağırlık katsayısı
$x_1 = 0.5$	$w_1 = -0.2$
$x_2 = 0.6$	$w_2 = 0.6$
$x_3 = 0.2$	$w_3 = 0.2$



Şekil 4.56 Doğal ve yapay sinir hücresi kavramlarının örtüşürülmesi.

Buna göre hücreye ait girdi verileri ve ağırlık katsayılarının verili çizelgedeki gibi olduğunu öngörelim. Ayrıca toplama fonksiyonu olarak toplama, aktivasyon fonksiyonu olarak sigmoid fonksiyonunun ve eşik değer olarak da  $b = 0.1$  değerinin dikkate alındığını düşünelim. İfade edilen girdi ve ağırlık katsayıları ile toplama fonksiyonu baz alınarak hücre içi net giriş değeri;

$$net = \sum_{i=1}^3 w_i x_i + b \quad (4.160)$$

$$net = -0.2 \times 0.5 + 0.6 \times 0.6 + 0.2 \times 0.2 + 0.1 = 0.4$$

$$net = 0.4$$

şeklinde elde edilebilir. Net girdi değerinin aktivasyon fonksiyonunda yerine konulması halinde;

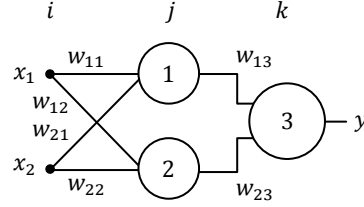
$$f(net) = \frac{1}{1 + e^{-net}} = \frac{1}{1 + e^{-0.4}} = 0.5744$$

sonucuna ulaşılır. Birden fazla yapay sinir hücresinin yer aldığı çok katmanlı bir ağ örneğinde ise müteakip veriler dikkate alınabilir;





Girdi verisi	Ağırlık katsayısı
$x_1 = 1$	$w_{11} = 2$
$x_2 = 0$	$w_{12} = 1$
	$w_{21} = -3$
	$w_{22} = 4$
	$w_{13} = 2$
	$w_{23} = -1$



Görüldüğü üzere yapay sinir ağı girdi, iki saklı ve çıktı katmanından oluşan bir ağ olup ağ kapsamında toplama fonksiyonu olarak toplama, aktivasyon fonksiyonu olarak adım fonksiyonunun ve eşik değeri olarak da  $b = 0$  değerinin dikkate alındığını düşünelim. Buna göre  $j = 1$  ve  $2$  numaralı hücrelerin net giriş değerleri;

$$net_1 = \sum_{i=1}^2 w_{ij}x_i + b = w_{11}x_1 + w_{21}x_2 = 2 \times 1 + (-3) \times 0 = 2$$

$$net_2 = \sum_{i=1}^2 w_{ij}x_i + b = w_{12}x_1 + w_{22}x_2 = 1 \times 1 + 4 \times 0 = 1$$

şeklinde elde edilebilir. Her bir sinir hücresindeki aktivasyon fonksiyonları dikkate alındığında;

$$f(net) = \begin{cases} 1 & net \geq 0 \\ 0 & net < 0 \end{cases}$$

$$\begin{aligned} f(net_1) &= f(2) = 1 \\ f(net_2) &= f(1) = 1 \end{aligned}$$

ara çıktı sonuçlarına ulaşılır. Saklı katmanların ikincisine gelindiğinde ise net girdi değeri;

$$net_3 = \sum_{j=1}^2 w_{jk}f(net_j) + b = w_{13}f(net_1) + w_{23}f(net_2) = 2 \times 1 + (-1) \times 1 = 1$$

olarak bulunur. Son katmandaki aktivasyon fonksiyonu, girdi verisine uygulandığında;

$$f(net_3) = f(1) = 1$$

son çıktı değeri elde edilir.



Yapay sinir ağlarının ağ yapısı dâhilindeki ağırlık katsayılarının ne olacağı hususu, öğrenme safhası olarak tanımlanan süreç ile belirlenir. Öğrenme sürecinde, modellenmesi düşünülen fonksiyona ait gerçek girdi ve çıktı değerleri baz alınır. Ana gaye, eldeki veri havuzuna ait girdi ve çıktı değerlerinin mümkün mertebe aynen elde edilebilmesini sağlayan yapay sinir ağı yapısı ve katsayılarının belirlenmesidir. Bu bağlamda dikkate alınan temel ölçüt, ağ çıktısı ile gerçek çıktı verisinin arasındaki farkın minimize edilmesidir. Genellikle minimizasyonda;

$$mse = \frac{1}{n} \sum_{k=1}^n e_k^2$$

ifadesi temel alınır. Burada  $mse^i$  ortalama karesel hatayı,  $e_k$ ;

$$e_k = (t_k - y_k) \quad (4.161)$$

şeklindeki  $t_k$  gerçek ve  $y_k$  ağ çıktısı farkını temsil etmektedir. Ortalama karesel hatanın, ağ çıktısı baz alınarak ağırlık katsayılarına bağlı ifadesini elde etmek mümkündür. Hatırlanacağı üzere ağ çıktı değeri, toplamaya dayalı toplama fonksiyonu dikkate alınarak;

$$y_k = f \left\{ \sum_{j=1}^m w_{jk} f \left\{ \sum_{i=1}^p w_{ij} x_i + b_j \right\} + b_k \right\} \quad (4.162)$$

denkleminde ifade edilebilmektedir. Bu ifadenin hata denkleminde yerine konulması ile ortalama hata karesini ve onun bağlı olduğu ağırlık katsayılarını, uygun kısmi türevlerin alınması ile elde etmek mümkündür. Ancak ağırlık katsayısının çokluğu ve birbirine bağlı kısmi türevlerin işlem sayıları dikkate alındığında bahse konu yaklaşımın pek kolay olmadığı söylenebilir. Bu nedenle ağırlık katsayı hesaplama sürecinde “hatanın geriye yayılması” olarak adlandırılan bir yaklaşım sergilenir<sup>(76)</sup>. Bahse konu yaklaşım 1974 yılında P.J. Werbos’un yayımladığı doktora tezi<sup>iii</sup> ile gündeme gelmiş<sup>(77)</sup> ve daha sonra yaygın olarak kullanılmaya başlanmıştır<sup>(78)</sup>. Hatanın geriye yayılması işlemlerini irdelemek için Şekil 4.57 ve tanımlı notasyonlar dikkate alınabilir. Buna göre genel hata fonksiyonunu daha basit haliyle;

$$E = \frac{1}{2} \sum_{k=1}^n (t_k - y_k)^2 \quad (4.163)$$

<sup>i</sup> Mean square error

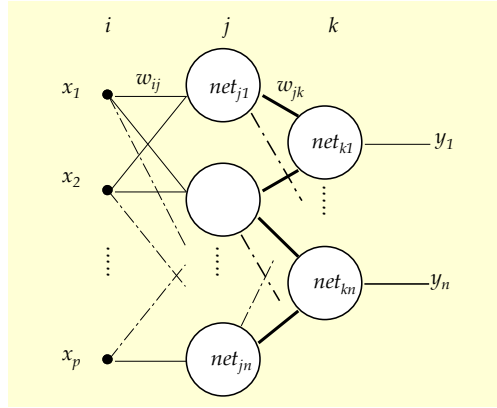
<sup>ii</sup> Propagating the errors or “back propagation”

<sup>iii</sup> “New tools for prediction and analysis in the behavioral sciences”





şeklinde belirlemek mümkündür. Ağ çıktı verisinin hesaplanması adım adım gerçekleştirilebilir.



Şekil 4.57 Yapay sinir ağı ve simgesel gösterim.

Girdi verilerinin saklı katmana giriş değerleri;

$$net_j = \sum_j w_{ij}x_i + b_j \quad (4.164)$$

ifadesi kullanılarak, bu verinin katmandaki sinir hücresinden çıkış değeri ise aktivasyon fonksiyonu dikkate alınarak;

$$o_j = f(net_j) = f\left\{\sum_j w_{ij}x_i + b_j\right\} \quad (4.165)$$

denklemleri ile temsil edilebilir. Eldeki verinin müteakip katmana girdi verisi etkisi;

$$net_k = \sum_j w_{jk}o_j + b_k \quad (4.166)$$

$$net_k = \sum_j w_{jk}f\left\{\sum_j w_{ij}x_i + b_j\right\} + b_k \quad (4.167)$$



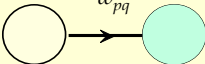
denklemleri ile ifade edilir. Bu net değerinin son katmandaki aktivasyon fonksiyonunda dikkate alınması sonrasında;

$$y_k = f(\text{net}_k) = f \left\{ \sum_j w_{jk} f \left\{ \sum_j w_{ij} x_i + b_j \right\} + b_k \right\} \quad (4.168)$$

şeklindeki ağ çıktı verisini belirlemek mümkündür. Elde edilen ağ çıktı verisinin hata denkleminde yerine yerleştirilmesi neticesinde;

$$E = \frac{1}{2} \sum_{k=1}^n \left[ t_k - f \left\{ \sum_j w_{jk} f \left\{ \sum_j w_{ij} x_i + b_j \right\} + b_k \right\} \right]^2 \quad (4.169)$$

denkleme ulaşılır. Bu denklemde geçen  $w_{jk}$  ve  $w_{ij}$  katsayılarını, uygun bir gradyan esaslı yöntem ile belirlemek mümkündür. Buna göre genel çerçevede ağırlık katsayısı güncelleme işlemi;

$$w_{pq}^{(t+1)} = w_{pq}^{(t)} + \Delta w_{pq}$$


denkleme göre gerçekleştirilebilir. Güncellemeye esas veriyi sağlayan gradyan verisi her katman için;

$$\Delta w_{jk} = -\alpha \frac{\partial E}{\partial w_{jk}} \quad (4.170)$$

$$\Delta w_{ij} = -\alpha \frac{\partial E}{\partial w_{ij}} \quad (4.171)$$

yaklaşımından hareketle;

$$\Delta w_{pq} = \alpha \sum \delta_q o_p \quad (4.172)$$

ifadesine göre belirlenir. Burada  $\alpha$  adım büyüklüğünü göstermekte,  $\delta_q$  terimi katmana göre;







$$\delta_k = (t_k - y_k)f'(net_k) \quad (4.173)$$

$$\delta_j = \left( \sum_k w_{jk} \delta_k \right) f'(net_j) \quad (4.174)$$

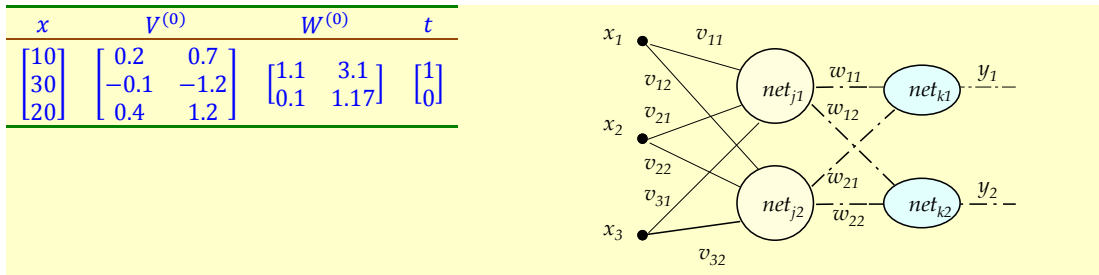
denklemleri dikkate alınarak belirlenmektedir. Görüldüğü üzere güncellemelerde aktivasyon fonksiyonunun 1inci mertebeden türevi gerekmekte olup bu nedenle seçili aktivasyon fonksiyonunun sürekli olması ve türev alma işleminin de kolay olması beklenir. İfade edilen nedenlerden dolayı aktivasyon fonksiyonu olarak genellikle sigmoid fonksiyonu tercih edilmekte olup müteakip denklemde bahse konu fonksiyon ve ona ait türev ifadesi görülmektedir;

$$f(x) = \frac{1}{1 + e^{-x}} \quad f'(x) = f(x)(1 - f(x))$$

Güncelleme denklemi, bazı uygulamalarda ilave momentum terimi içerecek şekilde tanımlanabilmektedir. Bu durumda dikkate alınan denklem;

$$w_{pq}^{(t+1)} = w_{pq}^{(t)} + \mu \Delta w_{pq} + \Delta w_{pq} \quad (4.175)$$

biçiminde olup  $\mu$  simgesi momentum katsayısını temsil etmektedir. Momentum katsayısı genellikle 0.6 ila 0.9 arası bir değer seçilir. Hatanın geriye yayılması işlem adımlarının daha iyi anlaşılması çerçevesinde örnek bir sorunu dikkate alabiliriz<sup>(79)</sup>. Buna göre girdi, çıktı ve rastgele seçilen ilk ağırlık katsayı değerleri müteakip çizelgede verilmektedir;



Şekil 4.58 Örnek yapay sinir ağı ve veri çizelgesi.

Katmanlar arası ağırlık katsayılarının karışmaması için girdi katmanı ile saklı katman arası bölgede  $v_{ij}$  simgesi, saklı katman ile çıktı katmanı arası bölgede ise  $w_{jk}$  simgesi tercih edilmiştir. Ağ



dâhilindeki yapay sinir hücrelerinde aktivasyon fonksiyonunun sigmoid fonksiyonu olduğu var sayılmış, ayrıca eşik değer kavramı  $b = 0$  kabulü dâhilinde dikkate alınmamıştır. Güncellemede adım büyüklüğü değeri  $\alpha = 0.1$  olarak belirlenmiştir. Buna göre takibedilen işlem adımları sıralanmıştır;

### İleri besleme kısmı;

- 1. Gizli katman hücrelerine ait net değerlerin elde edilmesi;

$$net_{j1} = \sum_{i=1}^3 v_{i1}x_i = v_{11}x_1 + v_{21}x_2 + v_{31}x_3 = 0.2 \times 10 + (-0.1) \times 30 + 0.4 \times 20 = 7$$

$$net_{j2} = \sum_{i=1}^3 v_{i2}x_i = v_{12}x_1 + v_{22}x_2 + v_{32}x_3 = 0.7 \times 10 + (-1.2) \times 30 + 1.2 \times 20 = -5$$

- 2. Gizli katman hücrelerine ait çıkış değerlerinin elde edilmesi;

$$o_1 = f(net_{j1}) = \frac{1}{1 + e^{-net_1}} = \frac{1}{1 + e^{-7}} = 0.999$$

$$o_2 = f(net_{j2}) = \frac{1}{1 + e^{-net_2}} = \frac{1}{1 + e^{-(-5)}} = 0.0067$$

- 3. Çıktı katmanına ait net değerlerin elde edilmesi;

$$net_{k1} = \sum_{j=1}^2 w_{j1}o_j = w_{11}o_1 + w_{21}o_2 = 1.1 \times 0.999 + 0.1 \times 0.0067 = 1.0996$$

$$net_{k2} = \sum_{j=1}^2 w_{j2}o_j = w_{12}o_1 + w_{22}o_2 = 3.1 \times 0.999 + 1.17 \times 0.0067 = 3.1047$$

- 4. Çıktı katmanına ait ağ çıkış değerlerinin elde edilmesi;

$$y_1 = f(net_{k1}) = \frac{1}{1 + e^{-net_{k1}}} = \frac{1}{1 + e^{-1.0996}} = 0.75$$

$$y_2 = f(net_{k2}) = \frac{1}{1 + e^{-net_{k2}}} = \frac{1}{1 + e^{-3.1047}} = 0.957$$



### Hatanın geriye yayılması kısmı;

- 1. Çıktı katmanına ait ağ çıkış değerlerinde oluşan hatanın hesaplanması;

$$\delta_k = (t_k - y_k)f'(net_k)$$

$$\begin{aligned}\delta_1 &= (t_1 - y_1)f'(net_{k1}) = (1 - 0.75)0.75(1 - 0.75) = 0.0469 \\ \delta_2 &= (t_2 - y_2)f'(net_{k2}) = (0 - 0.957)0.957(1 - 0.957) = -0.0394\end{aligned}$$

- 2. Çıktı-saklı katmanları arası ağırlık katsayısı değişim değerlerinin elde edilmesi;

$$\Delta w_{jk} = \alpha \delta_k o_j$$

$$\begin{aligned}\Delta w_{11} &= \alpha \delta_1 o_1 = 0.1 \times 0.0469 \times 0.999 = 0.004685 \\ \Delta w_{12} &= \alpha \delta_2 o_1 = 0.1 \times (-0.0394) \times 0.999 = -0.003936 \\ \Delta w_{21} &= \alpha \delta_1 o_2 = 0.1 \times 0.0469 \times 0.0067 = 0.00003142 \\ \Delta w_{22} &= \alpha \delta_2 o_2 = 0.1 \times (-0.0394) \times 0.0067 = -0.0000264\end{aligned}$$

- 3. Saklı katman üzerindeki hataların hesaplanması;

$$\delta_j = \left( \sum_k w_{jk} \delta_k \right) f'(net_j), \quad j = 1,2; k = 1,2$$

$$\begin{aligned}\delta_1 &= \left( \sum_{k=1}^2 w_{1k} \delta_k \right) f'(net_{j1}) = (w_{11} \delta_1 + w_{12} \delta_2) f'(net_{j1}) = (1.1 \times 0.0469 + 3.1 \times (-0.0394)) f'(7) \\ &= -0.0000705\end{aligned}$$

$$\begin{aligned}\delta_2 &= \left( \sum_{k=1}^2 w_{2k} \delta_k \right) f'(net_{j2}) = (w_{21} \delta_1 + w_{22} \delta_2) f'(net_{j2}) = (0.1 \times 0.0469 + 1.17 \times (-0.0394)) f'(-5) \\ &= -0.00259\end{aligned}$$

- 4. Girdi-saklı katmanları arası ağırlık katsayısı değişim değerlerinin elde edilmesi;

$$\Delta v_{ij} = \alpha \delta_j x_i, \quad i = 1,2,3; j = 1,2$$

$$\begin{aligned}\Delta v_{11} &= \alpha \delta_1 x_1 = 0.1 \times (-0.0000705) \times 10 = -0.0000705 \\ \Delta v_{12} &= \alpha \delta_2 x_1 = 0.1 \times (-0.00259) \times 10 = -0.00259 \\ \Delta v_{21} &= \alpha \delta_1 x_2 = 0.1 \times (-0.0000705) \times 30 = -0.0002115 \\ \Delta v_{22} &= \alpha \delta_2 x_2 = 0.1 \times (-0.00259) \times 30 = -0.00777 \\ \Delta v_{31} &= \alpha \delta_1 x_3 = 0.1 \times (-0.0000705) \times 20 = -0.000141 \\ \Delta v_{32} &= \alpha \delta_2 x_3 = 0.1 \times (-0.00259) \times 20 = -0.00518\end{aligned}$$



► 5. Tüm ağırlık katsayılarının güncellenmesi;

$$w_{jk}^{(1)} = w_{jk}^{(0)} + \Delta w_{jk}$$

$$\begin{aligned} w_{11}^{(1)} &= w_{11}^{(0)} + \Delta w_{11} = 1.1 + 0.004685 = 1.104685 \\ w_{12}^{(1)} &= w_{12}^{(0)} + \Delta w_{12} = 3.1 - 0.003936 = 3.096064 \\ w_{21}^{(1)} &= w_{21}^{(0)} + \Delta w_{21} = 0.1 + 0.00003142 = 0.10003142 \\ w_{22}^{(1)} &= w_{22}^{(0)} + \Delta w_{22} = 1.17 - 0.0000264 = 1.1699736 \end{aligned}$$

$$v_{ij}^{(1)} = v_{ij}^{(0)} + \Delta v_{ij}$$

$$\begin{aligned} v_{11}^{(1)} &= v_{11}^{(0)} + \Delta v_{11} = 0.2 - 0.0000705 = 0.1999295 \\ v_{12}^{(1)} &= v_{12}^{(0)} + \Delta v_{12} = 0.7 - 0.00259 = 0.69741 \\ v_{21}^{(1)} &= v_{21}^{(0)} + \Delta v_{21} = -0.1 - 0.0002115 = -0.1002115 \\ v_{22}^{(1)} &= v_{22}^{(0)} + \Delta v_{22} = -1.2 - 0.00777 = -1.20777 \\ v_{31}^{(1)} &= v_{31}^{(0)} + \Delta v_{31} = 0.4 - 0.000141 = 0.399859 \\ v_{32}^{(1)} &= v_{32}^{(0)} + \Delta v_{32} = 1.2 - 0.00518 = 1.19482 \end{aligned}$$

Güncelleme işlemi, eldeki her girdi-çıkı verisi için ayrı ayrı gerçekleştirilir. Ancak bu işlemin bir defa icra edilmesi yeterli değildir. Genellikle aynı süreç birden fazla devirli olarak tekrarlanır. Müteakip algoritmada yapay sinir ağının eğitim sürecine ait işlemler süreci görülmektedir;

Tekrarla  $i = 1: d$

Tekrarla  $j = 1: s$

$$w_{jk}^{(1)} = w_{jk}^{(0)} + \Delta w_{jk}$$

$$v_{ij}^{(1)} = v_{ij}^{(0)} + \Delta v_{ij}$$

Başa dön

Başa dön

Burada  $d$  devir sayısını,  $s$  veri çifti sayısını göstermektedir. Genellikle veri çifti sayısının çok olması daha iyi bir eğitim ve dolayısıyla da daha hassas bir modelleme anlamına gelir. Ancak hassas modelleme her zaman tercih edilen bir durum olmayabilir. Zira hassaslık yanında arzu edilen diğer bir özellik de modelin genelleme yapabilmesidir.





**MATLAB uygulaması**

```
clc; clear

% Veriler
data = [10 30 20 1 0];
input = data(:,1:3);
output = data(:,4:5);

n = 0.1; % optimizasyon katsayısı
m = 0; % devir sayacı
s = size(input); % veri sayısı
d = 1 ; % devir sayısı

% Baslangic ağırlık katsayı değerleri
V = [0.2 -0.1 0.4;0.7 -1.2 1.2]; % girdi katmanı
W = [1.1 0.1;3.1 1.17]; % çıktı katmanı

while m < d
    m = m + 1;

    % veri havuzu dâhilinde tekrarlı işlemler
    for i=1:s(1)
        I = input(i,:)';
        D = output(i,:)';

        % girdi-çıkıtlı işlemleri
        x1 = V*I;
        net = 1./(1+exp(-x1));
        x2 = W*net; O = 1./(1+exp(-x2));

        % hata hesapları
        delta_k = O.*(1-O).*(D-O);
        delta_j = net.*(1-net).*(W'*delta_k);

        % ağırlık katsayı güncellemeleri
        W = W + n*delta_k*(net');
        V = V + n*delta_j*(I');
    end
end

V = V'
W = W'
```

```
V =
    0.1999    0.6997
   -0.1002   -1.2008
    0.3999    1.1995

W =
    1.1047    3.0961
    0.1000    1.1700
```





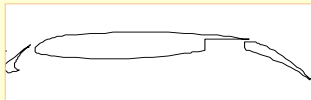
MATLAB uygulaması



Şekil 4.59 Örnek uçak kanadı <sup>(80)</sup>.

Uçak yapılarında çok elementli kanat profili kullanımı yaygın bir uygulama olup bu bağlamda bir kanat ön tarafta slat, ortada ana yapı ve firar kenarında da flap olmak üzere üç parçadan oluşacak şekilde tasarlanabilir. Çok elementli kanat profillerinin değişen  $\alpha$  hücum açısına göre üreteceği taşıma ( $C_L$ ) ve sürükleme ( $C_D$ ) katsayılarını, deneysel ve hesaplamalı yöntemlerle elde etmek mümkündür. Ancak gerek deneysel çalışma ve gerekse de hesaplamalı yöntemler hem zahmetli ve hem de pahalı yöntemlerdir. Bu nedenle elde edilen verilerden hareketle veki model veki kurgulama ve sonrasında da değişik hücum açıları için söz konusu katsayıların öngörülmesi tercih edilebilir bir durumdur. Müteakip çizelgede örnek bir üç-parçalı kanada ait, benzetim veri bazı hücum açısı ( $\alpha^\circ$ )-taşıma katsayısı ( $C_L$ )-sürükleme katsayısı ( $C_D$ ) değerleri görülmektedir.

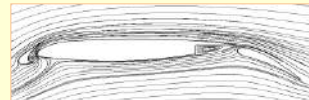
$\alpha^\circ$	14	16	18	20	22	24	26
$C_L$	3.3070	3.4490	3.534	3.6170	3.6580	3.666	3.6080
$C_D$	0.1226	0.1622	0.170	0.1919	0.2286	0.247	0.2744



(a) Üç parçalı kanat profili.



(b) Kanat profili etrafındaki sayısal çözüm ağı.



(c) Sayısal çözüm sonrası elde edilen akım çizgileri.

Şekil 4.60 Örnek üç-parçalı kanat profili ve hesaplamalı akışkanlar mekaniği çözümü.





Eldeki sayısal verilerden hareketle, yapay sinir ağlarını kullanarak, hücum açısına göre değişen iki ayrı vekiil model kurgulamak isteyelim. Bunlardan ilki  $\alpha - C_L$  veri çiftine, diğeri ise  $\alpha - C_D$  veri çiftine ait olsun. Kurgulanan vekiil modellerden yararlanarak  $\alpha = 21^\circ$  için taşıma ve sürüklenme katsayılarını öngörelim.

#### Komut

```
clc; clear
% data [alpha CL CD]
data = [14 3.307 0.1226
16 3.449 0.1622
18 3.534 0.17
20 3.617 0.1919
22 3.658 0.2286
24 3.666 0.247
26 3.608 0.2744];

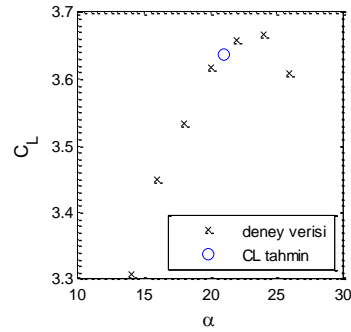
% yapay sinir ağı teşkili
net1 = newgrnn(data(:,1)',data(:,2)');
net2 = newgrnn(data(:,1)',data(:,3)');

% örnek veri tahmini
I = 21;
CL = sim(net1,I)
CD = sim(net2,I)

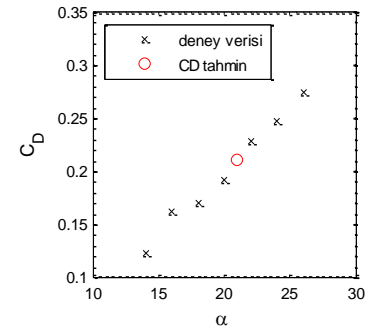
subplot(1,2,1);
plot(data(:,1),data(:,2),'kx')
hold on
plot(I,CL,'bo');
axis square
subplot(1,2,2);
plot(data(:,1),data(:,3),'kx')
hold on
plot(I,CD,'ro');
axis square
```

#### Sonuç

```
CL =
    3.6374
CD =
    0.2102
```



(a)



(b)

**Şekil 4.61** Sayısal veriler ve  $\alpha = 21^\circ$  için öngörülen aerodinamik katsayı değerleri: (a)  $C_L$  taşıma katsayısı, (b)  $C_D$  sürüklenme katsayısı.



#### 4.8 Bulanık mantık

Bulanık mantık kavramı, mantık bilimi üzerine temellendirilmiş bir yaklaşım olup günümüzde özellikle kontrol yöntemleri arasında gittikçe yaygınlık kazanan bir yöntemdir. Yöntemin daha iyi kavranabilmesi bağlamında temellerinin dayandığı mantık ve öncesinde de felsefe bilimine belli bir seviyede değinmekte yarar görülmektedir.

Felsefe, insan aklının erebileceği olgu ve bilgiler bütünü kapsayan bir kavram niteliğindedir. Felsefe, var olanların veya olduğuna inanılanların varlığı, kaynağı, anlamı ve nedeni üzerine düşünme, bilgelik sevgisi, hikmet arayışı anlamlarına gelmektedir. Kelime kökeni olarak Yunanca'da "seviyorum, peşinden koşuyorum, arıyorum" anlamlarına gelen "phileo" ve bilgi, bilgelik anlamlarına gelen "sophia" sözcüklerinden türemiştir<sup>(81)</sup>. Geleneksel olarak felsefenin üç ana dalından bahsedilebilir; söz konusu ana dallar doğa felsefesi<sup>ii</sup>, ahlak felsefesi<sup>iii</sup> ve doğa-ötesi felsefe<sup>iv</sup> şeklinde sıralanabilir. Doğa felsefesi, doğa bilimlerini içermekte olup fizik, kimya ve biyoloji gibi temel fen bilimlerini kapsamaktadır. Günümüzde söz konusu tanım, her ne kadar önemsenmese de geçmiş dönemde yazılan eserler söz konusu başlık etiketini içermekteydi. Nitekim I. Newton tarafından kaleme alınan ve fizik bilimi için temel eserlerden biri olan "Doğa felsefesinin matematiksel ilkeleri"<sup>v</sup> isimli eser ifade edilen husus açısından iyi bir örnektir. Ahlak felsefesi ise daha çok sosyal bilimler ve bu bağlamda ahlak, estetik, siyaset gibi alt bilim dallarını kapsar. Tasnif dâhilinde dikkate alınan son ana dal niteliğindeki doğa-ötesi felsefe, her ne kadar fizik ötesi gibi soyut kavramlar içerse de bir öncekine göre daha geniş bir inceleme alanına sahiptir. Bu bağlamda mantık, matematik ve epistemoloji gibi bilim dalları sıralanabilir. Özellikle epistemolojinin kapsadığı önemli bir alt dal, dilbilimdir<sup>vi</sup>. Dikkate alınan sınıflandırma haricinde farklı kıstaslara göre değişik tasniflere gitmek de mümkündür<sup>(82)</sup>.

Geçmişte felsefenin, doğa bilimlerini de kapsayacak şekilde geniş çerçevede olmasının temel nedenlerinden birisi, doğa olaylarının gözlemlenmesi ve sonrasında da olgusal ilişkilerin sözel söylemlerle veya geometrik tasavvurlarla ifade edilmesiydi. Günümüzde de esas itibarıyla aynen geçerli olan yaklaşım, maalesef gözardı edilen önemli bir ilke niteliğindedir. İfade edilen hususun

<sup>i</sup> Fuzzy logic

<sup>ii</sup> Natural philosophy

<sup>iii</sup> Moral philosophy

<sup>iv</sup> Meta-physical philosophy

<sup>v</sup> "Mathematical principles of natural philosophy."

<sup>vi</sup> Philology



Şekil 4.61 Klasik mantık ile insani düşünce yaklaşımı temalı figür<sup>(104)</sup>.





netleşmesi adına, yine burada Newton'un sözel terimlerle ifade ettiği ünlü hareket ilkelerinden birisi hatırlatılabilir. Newton, kitabında hareket yasalarından ikincisini Latince orijinal haliyle;

- ▶ “Lex II: mutationem motus proportionalem esse vi motrici impressae, et fieri secundum lineam rectam qua vis illa imprimitur.”

şeklinde ifade etmekteydi. Bu ifadenin Türkçesi;

- ▶ “Hareketteki değişim, etki eden kuvvetle orantılı ve aynı doğrultudadır.”

şeklinde tercüme edilebilir<sup>(83)</sup>. Sözel olarak ifade edilen bu ilke, matematiksel simgelerle;

$$F \propto \frac{dP}{dt}$$

biçiminde tanımlanabilir. Hatırlanacağı üzere burada **F** yöneysel nitelikteki kuvveti, **P** yine yöneysel nitelikteki momentumu ve *t* zamanı temsil etmektedir.

Gözlemlerin sözel ifadelere dökümü yanında dikkate alınabilecek diğer bir husus, geometrik tasavvurlardır. Örneğin ikinci dereceden polinomial denklemlerin kökünün, esasen bir seri kare geometrisine sahip şekillerden birisinin kenar mesafesi olduğu gösterilebilir. Hatırlanacağı üzere 2nci dereceden bir polinomial denklemin kökleri;

$$ax^2 + bx + c = 0$$

eşitliğinin çözümü neticesinde;

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

denklemine göre belirlenebilmekteydi. Bu ifadenin elde edilmesini, geometrik tasavvurlarla göstermek mümkündür. Ancak çözüm sürecine başlamadan önce ifade edilen polinomial denklemi farklı bir formda tekrar yazabiliriz;



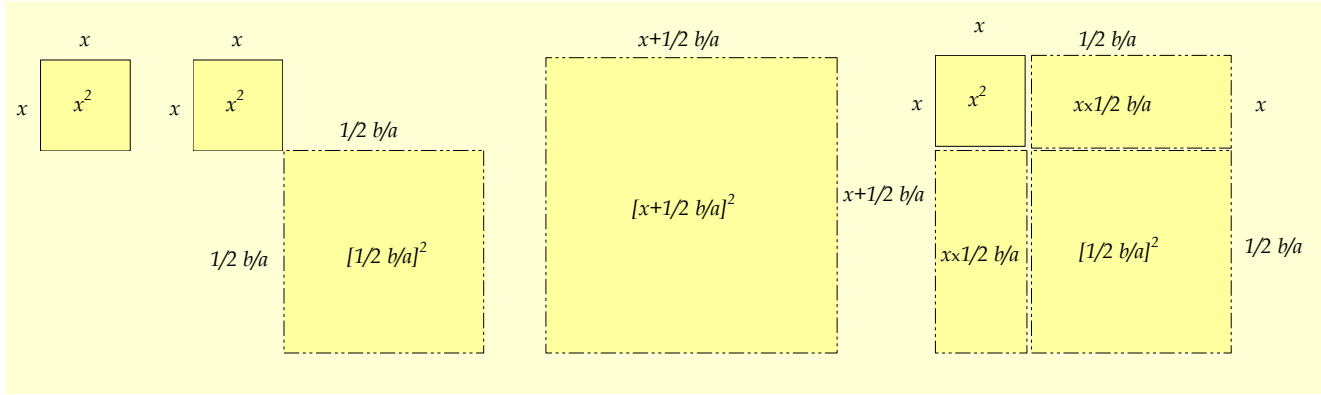
Şekil 4.62 1993 tarihli, I. Newton ve ikinci hareket yasası temalı Alman posta pulu <sup>(84)</sup>.





$$x^2 + \frac{b}{a}x + \frac{c}{a} = 0$$

Bilindiği üzere üssel olarak kare kavramı, geometrik bağlamda alan anlamına gelmektedir. Örneğin müteakip şekilden de görülebileceği üzere kenar uzunluğu  $x$  olan kare şeklinde bir alan düşünelim. İlave olarak kenar uzunluğu  $1/2 b/a$  olan ikinci bir kare daha düşünelim. Tasavvur edilen söz konusu iki kareyi uç uca ekleyerek yeni bir kare daha hayal edelim. Söz konusu daha büyük karenin kenar uzunluğu, tahmin edilebileceği üzere  $x + 1/2 b/a$  mesafesi kadar olacaktır.



Şekil 4.63 Örnek geometrik çıkarsama adımları.

İki küçük karenin yerleşiminden elde edilen büyük karenin alanını, bünyesinde barındırdığı iki küçük kare ve iki dikdörtgenin alanları toplamı şeklinde ifade edebiliriz;

$$\left[x + \frac{1}{2} \frac{b}{a}\right]^2 = x^2 + \frac{b}{a}x + \left[\frac{1}{2} \frac{b}{a}\right]^2$$

Diğer taraftan polinomiyal formdaki denklemi kullanarak;

$$x^2 + \frac{b}{a}x = -\frac{c}{a}$$

eşitliğini yazmak mümkündür. Bu ifade bir önceki alan denkleminde dikkate alınırsa;





$$\left[x + \frac{1}{2} \frac{b}{a}\right]^2 = -\frac{c}{a} + \left[\frac{1}{2} \frac{b}{a}\right]^2$$

eşitliği elde edilir. Elde edilen bu eşitliğin her iki tarafının karekökü alınır ve sonrasında da  $x$  terimini yalnız bırakma adına uygun düzenlemeler yapılırsa;

$$\sqrt[2]{\left[x + \frac{1}{2} \frac{b}{a}\right]^2} = \sqrt[2]{-\frac{c}{a} + \left[\frac{1}{2} \frac{b}{a}\right]^2} \rightarrow x = -\frac{1}{2} \frac{b}{a} + \frac{\sqrt[2]{b^2 - 4ac}}{2a}$$

$$x = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

sonucuna varılır. Görüldüğü üzere 2nci derece denklemler, esasen farklı kenar uzunluklarına sahip kare şeklindeki alanların kenar uzunlukları arasındaki ilişkiyi belirleyen denklemler niteliğindedir.

Gerek sözel ve gerekse de şekilsel tasavvurların doğayı anlamadaki etkinliği, nesiller değişse de kendisi değişmeyen bir ilke niteliğindedir<sup>(85)</sup>. Doğayı tasavvurda kullanılan önemli süzgeçlerden birisi de hiç kuşkusuz mantık bilimidir. Doğa-ötesi felsefenin önemli alt dallarında biri olan mantık kelimesi, köken olarak Arapça'dan gelmekte olup anlamı “konuşma”, “söz söyleme sanatı”dır. Batı literatüründeki hali ile “logic” sözcüğü, köken olarak Yunanca'da “kelime”, “üzerinde konuşulan” veya “sebeplere” anlamlarına gelen “logikos” kelimesinden gelmektedir. Mantık, kavram olarak nedenselleştirme sistematığı, çıkarsamanın geçerliliğini belirlemek adına dikkate alınan ölçüt ve ilkeler ya da tutarlı bir çerçevede düşünme yöntemi anlamlarına gelmektedir<sup>(86)</sup>.

Günümüzde mantık, yapılan çıkarsamaların çeşitliliğine göre değişik tasniflere tabi tutulabilir. Klasik mantık olarak dikkate alınan yaklaşımlarda çıkarsama sonucu, iki farklı değerle somutlaştırılır. Genellikle iki-değerli mantık olarak bilinen mantık yaklaşımının temelinde, yapılan çıkarsamanın doğru veya yanlış ama sadece bahse konu sonuçları doğurabileceği öngörülmüştür. Sayısal olarak söz konusu değerler 0 ve 1 şeklindedir. Klasik mantık anlayışı çerçevesinde geliştirilen en ünlü mantık yaklaşımı, Boolean mantığı olup 1800'lü yılların ortalarında G. Boole (1815-1864) tarafından gündeme getirilmiştir. Söz konusu yaklaşımın günümüz sayısal teknolojilerinde önemli bir yeri vardır. Boolean mantığının temel işlevleri

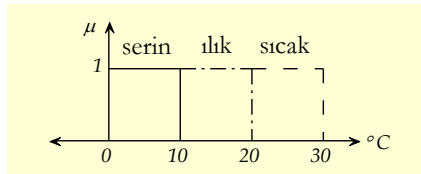
<sup>i</sup> Bivalent, two-level, two-valued logic



arasında “ve”, “veya” ve “değil” işlemleri sayılabilir<sup>(87)</sup>. Müteakip çizelgede, klasik mantık çerçevesinde kullanılan bazı semboller ve işlevleri görülmektedir<sup>(88)</sup>;

Simge	Anlamı	Örnek
$\Rightarrow, \rightarrow, \supset$	Çıkarım <sup>ii</sup> veya imleme; eğer... öyleyse...	$x = 3 \rightarrow x^2 = 9$
$\Leftrightarrow, \equiv, \leftrightarrow$	Denklik <sup>iii</sup>	$x + 1 = y + 2 \leftrightarrow x = y + 1$
$\neg, \sim, !$	Değil	$x = y \rightarrow \neg(x \neq y)$
$\wedge, \cdot, \&$	Kesişim ya da “ve” işlemi	$x < 5 \wedge x > 3 \rightarrow x = 4$ Doğal sayı şartı dikkate alınmaktadır.
$\vee, +, \parallel$	Birleşim ya da “veya” işlemi	$x \leq 5 \vee x \geq 3 \rightarrow x \neq 4$ Doğal sayı şartı dikkate alınmaktadır.
$\forall, ()$	Evensel niteliklendirme <sup>iv</sup> ya da “her”, “tüm”	$\forall x \in Z$
$\exists$	Var olma <sup>v</sup>	$\exists x: x + 1 = 2$

Klasik mantığın, herhangi bir değerlendirmede vereceği cevabı anlama bağlamında müteakip örnek değerlendirilebilir;



Şekil 4.64 Oda sıcaklığı değerleri ve iki-değerli mantığa göre tasnifi.

<sup>i</sup> Conjunction “and”, disjunction “or”, negation “not”

<sup>ii</sup> Implication

<sup>iii</sup> Equivalence

<sup>iv</sup> Universal quantification

<sup>v</sup> Existential quantification





Buna göre; ortam sıcaklığının ölçüldüğünü ve hava sıcaklığın  $11^{\circ}\text{C}$  olduğunu düşünelim. Belirlenen sıcaklığın serin mi, ılık mı yoksa sıcak mı olduğu yönündeki değerlendirmede verilecek cevap, söz konusu sıcaklığın ılık olduğu şeklindedir. Zira  $10 - 20^{\circ}\text{C}$  arası sıcaklık değerleri “ılık” olarak belirlenmiş olup sayısal değeri  $\mu = 1$ 'dir. Buna mukabil aynı sıcaklığın serin mi olduğu sorusuna verilecek cevap, kesinlikle hayır olup sayısal bağlamda değeri  $\mu = 0$ 'dır.

Yüzyıllar boyu iki değerli mantık yaklaşımı ile şekillenen mantık bilimi, son yüzyılda ufku biraz daha genişletmiş ve değerlendirmede ikiden fazla seçeneğin de olabileceği varsayımı dâhilinde çok-değerli mantık<sup>i</sup> yaklaşımı ile zenginleştirilmiştir. Aslında çok-değerli mantığın geçmişi çok eskilere, eski Yunan dönemindeki ünlü düşünür Aristo'ya kadar götürülebilir. Zira onun ifade ettiği üzere “yarın deniz savaşı olacak” önermesi ne doğru ne de yanlış olup doğru da olabilir yanlış da<sup>(89)</sup>. Her ne kadar söz konusu çıkarsama zaman boyutunun özelliğine dayansa da, ilave seçeneğin varlığı inkâr edilemez. Dolayısıyla makul düşünce gereği, doğru ve yanlış seçenekleri haricinde ilave bir seçenek kendisini göstermektedir.

Sistematik bağlamda çok-değerli mantık ilkeleri 20nci yüzyılda, özellikle 1920'li yıllarda J. Łukasiewicz (1878-1956) tarafından “doğru” ve “yanlış” ikilisine “olası<sup>iii</sup>” seçeneği eklenerek üç-değerli mantık<sup>iii</sup> açılımı gündeme getirilmiş ve sayısal değer seçenekleri  $\{0, 1, 1/2\}$  olarak belirtilmiştir. Aynı dönemin bir diğer ünlü matematikçisi E. L. Post (1897-1954), 1921 yılında  $n$ -değerli mantık ( $n \geq 2$ ) yaklaşımını dillendirdi<sup>(90)</sup>. Söz konusu yaklaşım, daha sonra 1950-1960'lı yıllar arasında daha da genişletilerek sonsuz değerli mantık yaklaşımına evrilmiştir.

Özellikle 1965 yılında Azeri bilim insanı L. A. Zadeh (1921- ) tarafından literatüre kazandırılan bulanık mantık yaklaşımı, sonsuz değerli mantık ekollerinden birisidir<sup>(91)</sup>. Aynı yıl yayımladığı “Bulanık kümeler<sup>iv</sup>” isimli makalesi ile Zadeh bulanıklık kavramını, bulanık kümeleri ve bu kümelere ait öğelerle yapılabilecek işlemleri tanımlamıştır. Ona göre bulanık küme, üyelik değerleri 0 ile 1 arasında değişebilen nesnelere kümesidir. Kümenin üyelik değerini, diğer bir ifade ile karakterini belirleyen unsur, üyelik fonksiyonudur<sup>(92)</sup>. Zadeh, belirttiği yaklaşıma göre doğru ya da yanlış hükmünü, kısmen doğru veya kısmen yanlış hükmüne dönüştürmekte ve sayısal bağlamda da kısmilik değerini 0 ile 1 arasındaki bir değerle örtüştürmektedir.

---

<sup>i</sup> Multi-valued logic

<sup>ii</sup> “Possible”, “null”, “unknown”, “not applicable” veya “unavailable”

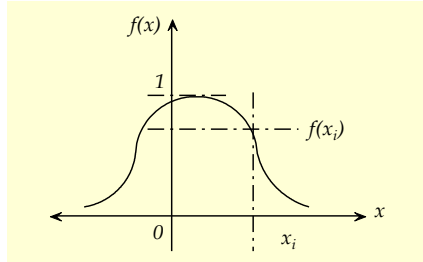
<sup>iii</sup> Three-valued logic

<sup>iv</sup> “Fuzzy sets”



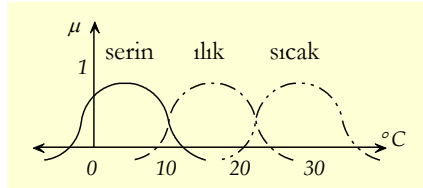


Müteakip şekilde Lütüzade'nin, üyelik fonksiyonu olarak verdiđi misal görölmektedir;



Şekil 4.65 Zadeh tarafından tanımlanan üyelik fonksiyonu.

Buna göre  $x_i$  noktası  $x$  eksenini üzerindeki herhangi bir nokta,  $f(x_i)$  bu noktanın  $f$  üyelik fonksiyonu üzerindeki üyelik deđeridir. Söz konusu deđer  $0 \leq f(x_i) \leq 1$  aralıđında deđişebilmektedir. Daha önce klasik mantık çerçevesinde verilen örneđin bulanık mantık çerçevesindeki hali müteakip şekilde görölmektedir;



Şekil 4.66 Oda sıcaklıđı deđerleri ve bulanık mantıđa göre tasnifi.

Yine bir önceki örnekte belirtilen hava sıcaklıđını ( $11\text{ }^{\circ}\text{C}$ ) yeni tasnife göre deđerlendirdiđimizde, söz konusu sıcaklıđın tamamen ılık olduđunu söylemek mümkün deđildir. Zira aynı sıcaklık, kısmen serin ama büyük oranda ılık olarak karakterize edilebilir. Serinlik ( $S$ ) tanımına göre üyelik fonksiyon deđerini  $\mu(S) = 0.1$ , ılıklık ( $I$ ) tanımına göre üyelik fonksiyon deđerini  $\mu(I) = 0.7$  olarak belirlemek mümkündür. Lütüzade'nin sistematik yaklaşıđına göre herhangi bir deđişkeni ki kendisi deđişken kavramını dilsel deđişken olarak adlandırır, bir beşli ile betimlemek mümkündür. Söz konusu beşli<sup>i</sup> ( $X, T(x), U, G, M$ ) şeklinde olup burada;  $X$  dilsel deđişkeni<sup>ii</sup>,  $U$  dilsel

<sup>i</sup> Quintuple

<sup>ii</sup> Linguistic variable



değişkenin tanım uzayını<sup>i</sup>,  $T(x)$  dilsel değişkene ait tasnif terimlerini<sup>ii</sup>,  $G$  tasnifte dikkate alınan ilkeleri ve  $M$  dilsel değişkene ait bulanık kümenin üretilmesinde dikkate alınan yöntemi ya da diğer bir ifade ile üyelik fonksiyonunu<sup>iii</sup> temsil etmektedir. Örnek üzerinde söz konusu terimlerin değerlendirilmesi yapıldığında; sıcaklık değişkeninin  $X$  dilsel değişkeni olduğu, sıcaklık değişim değerlerinin ki burada  $\pm\infty$  olarak tanımlanan sınırlar olacağı, “serin, ılık, sıcak” betimlemelerinin  $T(x)$  tasnif terimlerini içerdiği, sıcaklık değer araklıklarına göre tasnif edilmesinin tasnifte dikkate alınan  $G$  ilkesini gösterdiği, üyelik fonksiyonu olarak belirlenen Gauss fonksiyonunun ise  $M$  terimini somutlaştırdığı söylenebilir. Tasnifte dikkate alınan ilkeler bağlamında özellikle insani değerler silsilesi dikkate alınabilir. Müteakip çizelgede derecelendirme seviyeleri<sup>iv</sup> ve bu seviyelerin normalize edilmiş değer aralıkları görülmektedir<sup>(93)</sup>.

Seviye	Değer aralığı
Çok Küçük	0 – 0.3
Küçük	0 – 0.4
Biraz Küçük	0.25 – 0.45
Orta	0.3 – 0.7
Biraz Büyük	0.55 – 0.75
Büyük	0.6 – 1
Çok Büyük	0.7 – 1

Tabiatıyla tasnif, tanımı üzerinde çalışılan soruna özgü olarak belirlenebilir. Örneğin konum belirlemede “biraz sağa”, “çok sola” veya “az yukarı” gibi yönlendirme bazlı sınıflandırma terimlerini kullanmak mümkündür. Görüldüğü üzere bulanık mantık yaklaşımının değişken ve tasnif tanımlamalarında dilsel ve sezgisel yaklaşımlar sergilemesi, uygulamalarda insana has düşünce biçiminin hâkim olmasına olanak sağlamaktadır. Dolayısıyla karar verme süreçlerinde, muğlak değerlendirmelerin dikkate alınabilmesi mümkün olmakta, sayısal determinizmle düşüncel kıymetlendirme etkileri bütünleştirilebilmekte, diğer bir ifade ile sayısal veri ve dilsel kanaat kaynaştırılabilmektedir<sup>(94)</sup>. Ancak bu değerlendirme, bulanık mantığın deterministik olmadığı anlamına gelmemektedir. Zira süreç sonunda kesin ve net çıktı verilerine ulaşılır. Daha açık bir ifade ile aynı girdi verileri ve değerlendirme ölçütleri her zaman aynı çıktı verilerini üretir. Bu bağlamda bulanık mantık, stokastik süreçlerden farklıdır.

<sup>i</sup> Universe of discourse

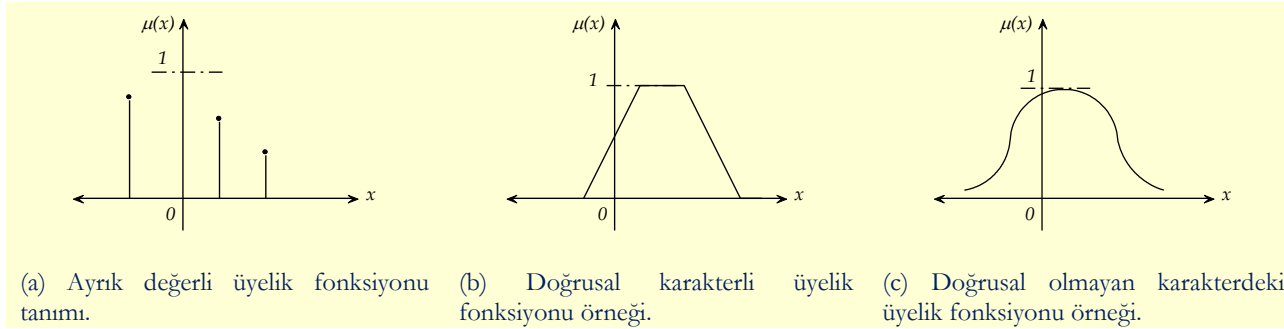
<sup>ii</sup> Linguistic term

<sup>iii</sup> Membership function

<sup>iv</sup> Linguistic hedges



Dilsel deęişkenin nitelięini belirleyen dilsel ya da burada kullanılan haliyle tasnif terimlerini niceliklendirme amacıyla, üyelik fonksiyon tanımlamaları kullanılır. Üyelik fonksiyonu, tekil (tekiz)<sup>i</sup> veya sürekli olabileceęi gibi doğrusal veya doğrusal olmayan karakterde de olabilir. Esasen bulanık mantık süreçlerinin doğrusal olmayan sorunlarda da uygun çıktı verileri üretmesi, büyük oranda doğrusal olmayan üyelik fonksiyonu tanımlamaları sayesinde. Bununla beraber çıkarsama süreçleri de aynı etkiye sahip olabilir. Müteakip şekillerde tekil ya da dięer ifade ile ayrık, doğrusal ve doğrusal olmayan karakterdeki örnek üyelik fonksiyonları görölmektedir<sup>(95)</sup>;



Şekil 4.67 Üyelik fonksiyonu türleri.

Müteakip çizelgede, sıkça kullanılan bazı üyelik fonksiyon tanımlamaları görölmektedir<sup>(90)</sup>;

Fonksiyon tanımı	Fonksiyon denklemi	Şekil
Üçgensel	$Tr[a, m, b](x) = \begin{cases} \frac{x-a}{m-a} & a \leq x \leq m \\ \frac{x-b}{m-b} & m < x \leq b \end{cases}$	

<sup>i</sup> Singleton







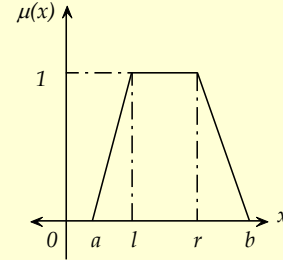
Fonksiyon tanımı

Fonksiyon denklemini

Şekil

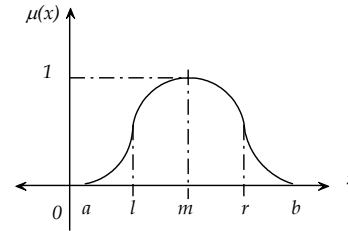
Yamuksal

$$Tp[a, l, r, b](x) = \begin{cases} \frac{x-a}{l-a} & a \leq x \leq l \\ 1 & l < x \leq r \\ \frac{x-b}{r-b} & r < x \leq b \end{cases}$$



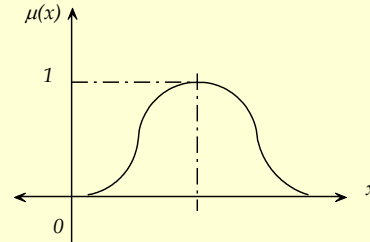
İkilenik

$$Tq[a, l, m, r, b](x) = \begin{cases} \frac{1}{2} \left[ \frac{a-x}{a-l} \right]^2 & a \leq x \leq l \\ 1 - \frac{1}{2} \left[ \frac{m-x}{m-l} \right]^2 & l < x \leq m \\ 1 - \frac{1}{2} \left[ \frac{m-x}{m-r} \right]^2 & m < x \leq r \\ \frac{1}{2} \left[ \frac{b-x}{b-r} \right]^2 & r < x \leq b \end{cases}$$



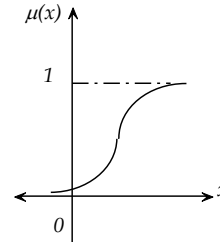
Gauss

$$Tg[m, s, \gamma](x) = \gamma e^{-\frac{(x-m)^2}{s^2}}$$



S-şekilli artan ikilenik

$$Sqi[a, l, m](x) = \begin{cases} \frac{1}{2} \left[ \frac{a-x}{a-l} \right]^2 & a \leq x \leq l \\ 1 - \frac{1}{2} \left[ \frac{m-x}{m-l} \right]^2 & l < x \leq m \end{cases}$$



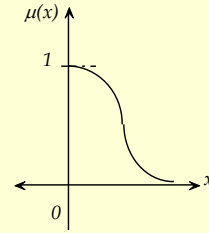
Fonksiyon tanımı

Fonksiyon denklemini

Şekil

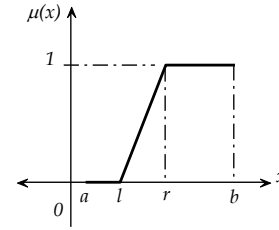
S-şekilli azalan ikilenik

$$Sqd[m, r, b](x) = \begin{cases} 1 - \frac{1}{2} \left[ \frac{m-x}{m-r} \right]^2 & m \leq x \leq r \\ \frac{1}{2} \left[ \frac{b-x}{b-r} \right]^2 & r < x \leq b \end{cases}$$



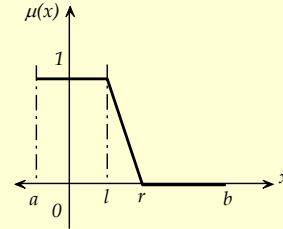
S-şekilli artan doğrusal

$$Sli[a, l, r, b](x) = \begin{cases} 0 & a \leq x \leq l \\ \frac{x-l}{r-l} & l \leq x \leq r \\ 1 & r \leq x \leq b \end{cases}$$



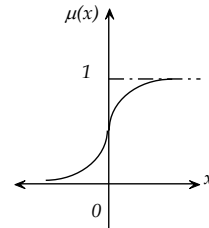
S-şekilli azalan doğrusal

$$Sld[a, l, r, b](x) = \begin{cases} 1 & a \leq x \leq l \\ \frac{x-r}{l-r} & l \leq x \leq r \\ 0 & r \leq x \leq b \end{cases}$$



Sigmoid-artan

$$Si[\beta](x) = \frac{1}{1 + e^{-\beta x}}, \quad \beta \geq 0$$





Fonksiyon tanımı	Fonksiyon denklemleri	Şekil
Sigmoid-azalan	$Si[\beta](x) = \frac{1}{1 + e^{-\beta x}}, \quad \beta < 0$	
Çan	$B[a, b, c](x) = \frac{1}{1 + \left  \frac{x - c}{a} \right ^b}$	

**Çizelge 4.5** Üyelik fonksiyonu tanımlamaları (devamı).

Üyelik fonksiyonlarının belirlenmesi sürecinde analitik bir yöntemden bahsetmek zordur. Genellikle sorun odaklı uzman görüşü, seçimde önem kazanır. Ancak eldeki verilere göre eğri uydurma veya histogram yapılanmasına da gidilebilir. Diğer bir seçenekte ise optimizasyon yöntemlerine dayalı uyarlamalara başvurulabilir. Üyelik fonksiyonları, nihayetinde girdi verisi niteliğindedir. Eldeki verili girdi verisine karşılık çıktı verisinin belirlenmesi için uygun bir vekil modelin kurgulanması gerekir. Bu bağlamda başvuru yaklaşım, kural tabanının<sup>i</sup> teşkilidir. Esasen söz konusu kural tabanını da içeren genel yapıya bulanık çıkarsama sistemi<sup>ii</sup> adı verilir. Literatürde Mamdani, Sugeno, Larsen ve Tsukamoto gibi farklı çıkarsama yaklaşımlarından bahsedilebilir. Ancak özellikle 1975 tarihli Mamdani modeli ile 1985 tarihli Sugeno yaklaşımları, en yaygın çıkarsama modelleridir<sup>(9)</sup>.

<sup>i</sup> “Rule base” veya “FAM (Fuzzy Associative Memories)”

<sup>ii</sup> Fuzzy inference system



Şekil 4.68'de Mamdani modeli görülmektedir. Buna göre bulanık mantık sistemi temelde üç öğeden teşkil edilmekte olup söz konusu öğeler; bulanıklaştırma, çıkarsama ve durulaştırma<sup>ii</sup> şeklindedir. Girdi ve çıktı verilerine yönelik üyelik fonksiyon tanımlamaları ile bu tanımlamalar ışığı altında net değer niteliğindeki girdi verisinin değerlendirilmesi işlemleri, bulanıklaştırma adı altında betimlenmektedir. Müteakip adımda çıkarsama işlemleri dikkate alınır. Bu bağlamda belirleyici unsur, kural tabanı ve bu kurallara göre çıktının belirlenmesi usulüdür. Akabinde net değer<sup>iii</sup> şeklindeki çıktı verisinin elde edilmesi sürecine geçilir. Çıkarsama kapsamındaki kural tabanı, girdi verilerinin aksine sayısal değerden ziyade sayısal olmayan sözel ifadelerden oluşur. Belirli önerme ya da diğer bir ifade ile önceller<sup>iv</sup> çerçevesinde, mantık kurallarına göre sonuç<sup>v</sup> ya da başka bir ifade ile çıkarsamalar yapılır. Hangi kuralların çıkarsamada dikkate alınacağı hususu, girdi verilerinin etkilediği dilsel değişken terimine ve bu terimlerin dikkate alındığı kurallara bağlıdır. Ancak kesin olan husus, çıkarsama sonrası üyelik fonksiyonu bağlamında yine [0,1] aralığında bir değer verisine ulaşılması gereğidir. Daha önce dilsel değişken bağlamında dikkate alınan sıcaklık değişkeni ve onunla ilgili tasnif terimlerinden hareketle örnek bir kural;

Eğer [sıcaklık {x} serin {S}] ise o halde [fanın hızını {v} biraz {B}] aç  

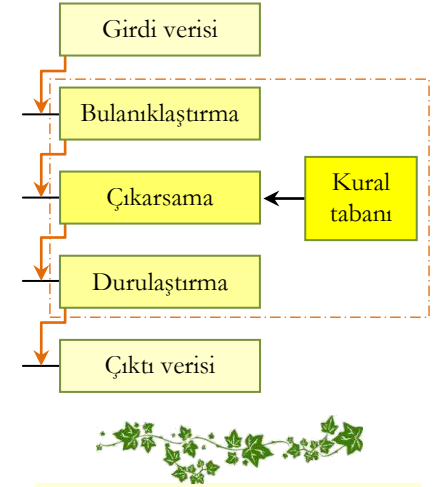
$$x = S \rightarrow v = B$$

şeklinde ifade edilmektedir. Basit bir çıkarsama yanında birden fazla önermenin dikkate alındığı çıkarsamalar mümkün ve hatta daha yaygındır. Örneğin;

Eğer [sıcaklık {x} serin {S}] “ve” [pencere {p} tam açık {TA}] ise o halde [fanın hızını {v} orta {O}] aç  

$$x = S \wedge p = TA \rightarrow v = O$$

çıkarsaması kapsamında sıcaklık değişkenine ilave olarak pencere açıklığı değişkeni dikkate alınmakta ve her iki öncel ile aradaki bağlaç dikkate alınarak bir sonuca varılmaktadır. Kural tanımlamada önerme sayısına göre pratik gösterim yöntemleri kullanılabilir. Örneğin iki önermeli kurgulamada matris formu, üç önermeli kural tabanında tensör formu dikkate alınabilir. Diğer taraftan görüldüğü üzere birden fazla değişkenin ilişkisini gösteren bağlaçlar bu bağlamda önem kazanmaktadır. Daha önce klasik mantık bahsinde yaygın olarak kullanılan mantık sembolleri ifade edilmişti. Aynı bağlaç ve simgeler bulanık mantık için de geçerlidir. Bununla beraber



Şekil 4.68 Mamadani modeli.

<sup>i</sup> Fuzzy logic system

<sup>ii</sup> “Fuzzification”, “inference engine” ve “defuzzification”

<sup>iii</sup> Crisp value

<sup>iv</sup> Antecedent

<sup>v</sup> Consequent

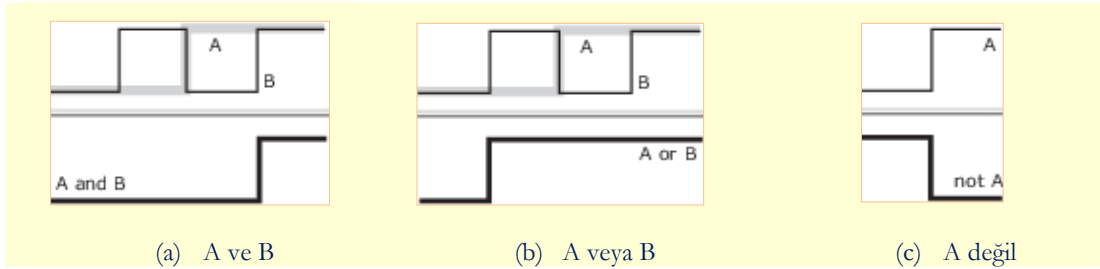




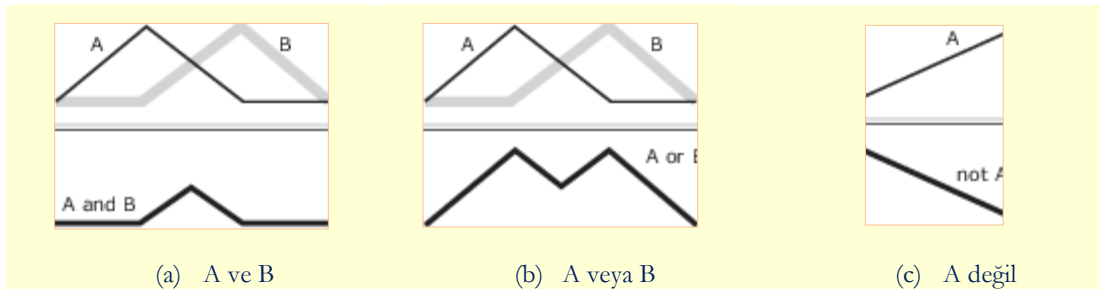
bulanık mantıkta bağlaçların sayısal değerlendirmedeki yaygın karşılıkları müteakip çizelgede ifade edilmektedir;

Sözel tanımlama	Mantıksal simgeleme	Matematiksel ifade	Bulanık mantık karşılığı
“A” ve “B”	$A \wedge B$	$A \cap B$	$\min(\mu(a), \mu(b))$
“A” veya “B”	$A \vee B$	$A \cup B$	$\max(\mu(a), \mu(b))$
“A” değil	$\neg A$	$A^c$	$1 - \mu(a)$

Müteakip şekillerde hem klasik ve hem de bulanık mantığa göre üyelik fonksiyonu bağlamında bağlaçların grafiksel gösterimi görülmektedir;



Şekil 4.69 Klasik mantığa göre önerme ve çıkarsama değerleri.



Şekil 4.70 Bulanık mantığa göre girdi ve çıktı değerleri.

Diğer taraftan nispeten basit çıkarsamalar yerine daha fazla matematiksel işlem içeren çıkarsama yöntemleri mevcut olup söz konusu çıkarsamalar örnek mahiyetinde müteakip çizelgede tanımlanmıştır;



Türü	Tanımı
Aritmetik ortalama	$\frac{\mu(a) + \mu(b)}{2}$
$p$ -ortalama	$\sqrt[p]{\frac{\mu^p(a) + \mu^p(b)}{2}}$
Harmonik ortalama	$\frac{2\mu(a)\mu(b)}{\mu(a) + \mu(b)}$
Geometrik ortalama	$\sqrt{\mu(a)\mu(b)}$
“veya” bağlamında kısıtlı toplam (BSUM <sup>i</sup> )	$\min\{1, \mu(a) + \mu(b)\}$
“ve” bağlamında kısıtlı fark (BDIF <sup>ii</sup> )	$\max\{0, \mu(a) + \mu(b) - 1\}$
“ve” bağlamında ürün (PROD <sup>iii</sup> )	$\mu(a)\mu(b)$
“veya” bağlamında cebirsel toplam (ASUM <sup>iv</sup> )	$\mu(a) + \mu(b) - \mu(a)\mu(b)$

**Çizelge 4.6** İlave çıkarsama yöntemleri.

Girdi verilerinin çıkarsaması bağlamında elde edilen üyelik fonksiyonu değerinin çıktı verisi karşılığı, basitçe öteleme yapılarak belirlenebilir. Müteakip şekilde de görülebileceği üzere elde edilen  $\beta$  üyelik fonksiyonu değeri, çıktı verisine ait üyelik fonksiyonunda  $\{\mu(y)\}$  yerine konularak alansal tutar belirlenebilir;

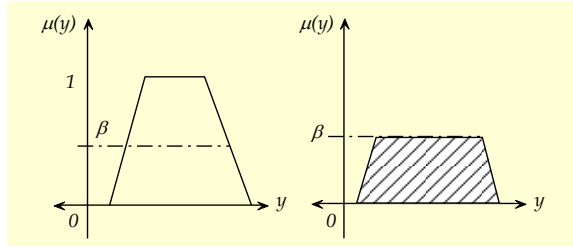
<sup>i</sup> BSUM: bounded summation

<sup>ii</sup> BDIF: bounded difference

<sup>iii</sup> PROD: product

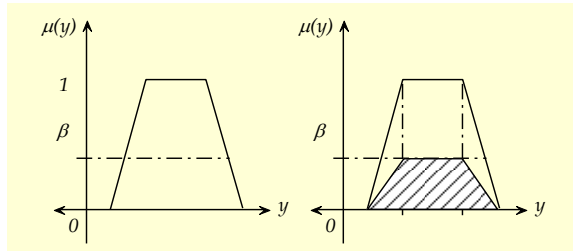
<sup>iv</sup> ASUM: “algebraic summation” veya “Probor: probabilistic or”





**Şekil 4.71** Çıkarsamanın çıktı değişkenine kırma temelli yansıması.

Bu yaklaşımdan daha etkin olduğu değerlendirilen diğer bir yaklaşım, öteleme ile beraber ölçeklendirme işlemini de dikkate alan müteakip yöntemdir. Bu yaklaşımda üyelik fonksiyon değeri seviyesi korunurken fonksiyona ait şeklin mahiyeti de muhafaza edilebilmektedir.

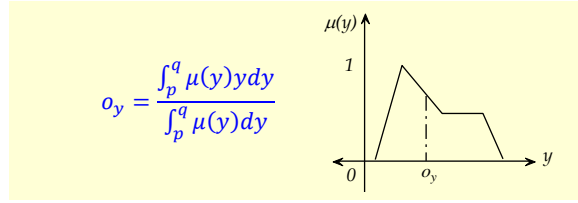


**Şekil 4.72** Çıkarsamanın çıktı değişkenine ölçeklendirme temelli yansıması.

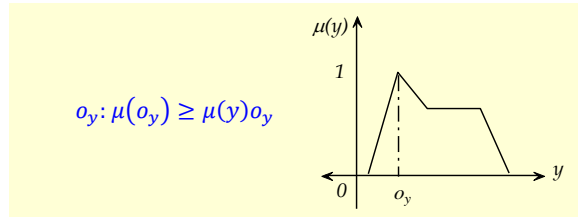
Özellikle birden fazla kuralın etki ettiği çıkarsamalarda ferdi çıkarsamaların toparlanması gerekir. Bu işlemden genellikle “veya” bağlacı dikkate alınır. Doğrudan toplama veya cebirsel toplama toparlamada değerlendirilebilir. Toparlama işlemi sonrasında durulama fazına geçilir. Bu fazda alansal nitelik taşıyan sonuç tek bir değere indirgenir. İndirgeme işleminde Sugeno tarafından gündeme getirilen ağırlık merkezi yaklaşımının kullanımı yaygındır. Yaklaşımın göre toparlanan alanın dikey eksene göre (üyelik fonksiyon değeri eksenine) ağırlık merkezi, müteakip ibare dikkate alınarak belirlenir;

<sup>i</sup> Aggregation

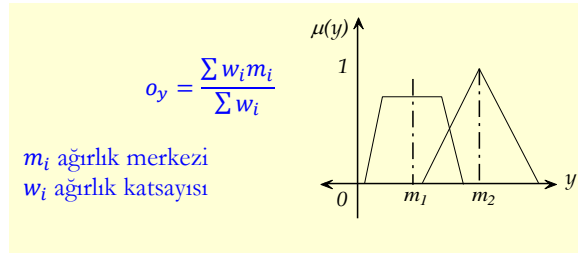




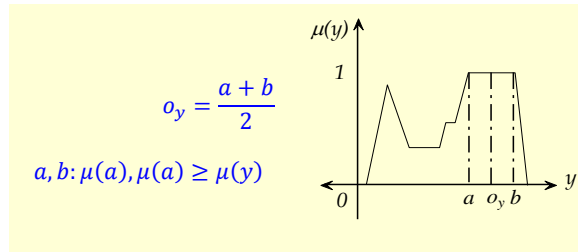
Elde edilen  $o_y$  değeri, net çıktı değeri niteliğindedir. Durulamada dikkate alınan diğer yöntemler arasında azami üyelik ilkesi;



ağırlıklandırılmış ortalama yaklaşımı;



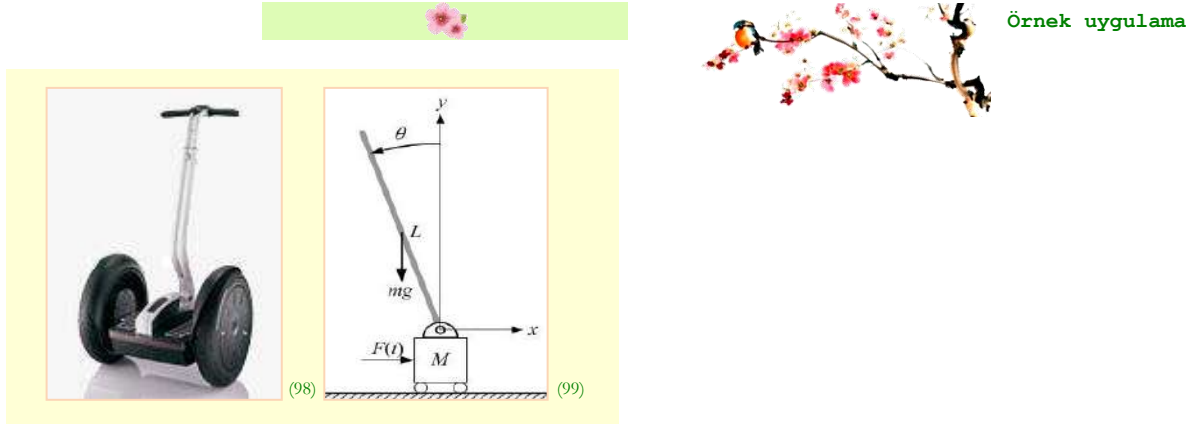
ortalama-azami değer belirlenmesi yöntemi;



sayılabilir<sup>(97)</sup>.

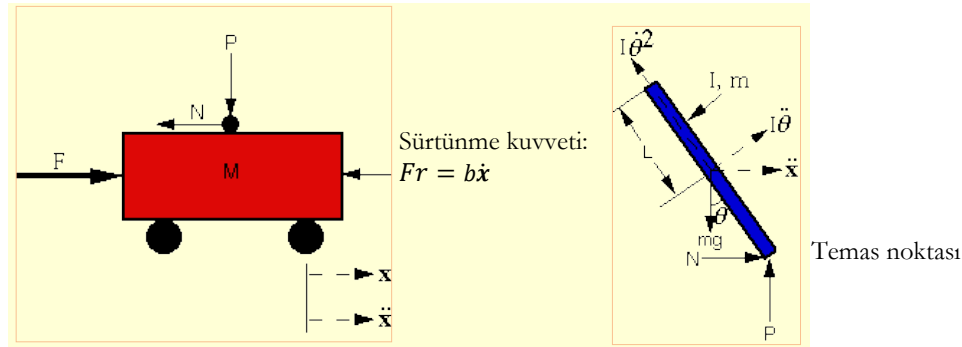






Şekil 4.73 Ters sarkaç uygulamasına örnek mahiyetindeki “elektrikli kaykay” aracı ve kuvvet diyagramı.

Bulanık mantığın uygulamada daha iyi anlaşılabilmesi çerçevesinde Şekil 4.73'te görülen örnek (ters sarkaç sorunu<sup>ii</sup>) ele alınabilir. Buna göre herhangi bir elektrikli kaykayın dengesi, tekerleklere uygun miktarda kuvvet uygulanarak sağlanabilir. Fiziksel düzenek bağlamında ters sarkaç sorunu olarak adlandırılabilen düzenekte etki eden kuvvetler, sarkacın ve tekerlek düzeneğinin (araba<sup>iii</sup>) ağırlığı ile yer kaynaklı sürtünme kuvvetidir. Tekerlek düzeneği ile sarkaç arasındaki temas noktası, etki-tepki ilkesine göre ayrıca değerlendirilebilir.



Şekil 4.74 “Elektrikli kaykay” aracı ve sisteme yönelik detay veri tanımlamaları <sup>(100)</sup>.

<sup>i</sup> “Segway” veya “ginger”

<sup>ii</sup> Inverted pendulum problem

<sup>iii</sup> Cart



Şekil 4.74 dâhilinde tekerlek düzeneği ile sarkaca etki eden kuvvetler ve hareket esnasında dikkate alınan diğer veriler resmedilmmiştir. Burada  $x$  arabanın yatay eksenindeki hareket mesafesi<sup>i</sup>,  $\dot{x}$ ,  $\ddot{x}$  sırasıyla arabaya ait hız<sup>ii</sup> ve ivme değeri,  $F$  dengenin sağlanması amacıyla arabaya etki ettirilen yatay kuvvet,  $b$  arabaya etki eden sürtünme katsayısı,  $M$  arabanın kütlesi,  $P$  ve  $N$  araba ile sarkacın temas (menteşe) noktasında etki eden dikey ve yatay kuvvetler,  $L$  sarkacın yarı boyu,  $I$  sarkaca ait eylemsizlik momentü,  $m$  sarkacın kütlesi,  $g$  yerçekimi ivmesi,  $\theta$  sarkacın dikey eksenle yaptığı açı<sup>iii</sup>,  $\dot{\theta}$ ,  $\ddot{\theta}$  sarkaca ait sırasıyla açısal hız<sup>iv</sup> ve ivme değerleridir. Kurulu düzenek dâhilinde öncelikle sarkacın ağırlık merkezinin konum yöneyini, sarkaç boyu ve açısal değeri dikkate alınarak;

$$\mathbf{r} = (x - L\sin\theta)\mathbf{i} + L\cos\theta\mathbf{j}$$

şeklinde yazılabilir. Burada  $\mathbf{i}$  ve  $\mathbf{j}$  eksenler dâhilindeki birim yöneylerdir. Yazılan bu ifadenin zamana göre ikinci mertebeden türevi alınarak ivme değerine kolayca ulaşılabilir;

$$\begin{aligned}\frac{\partial \mathbf{r}}{\partial t} = \dot{\mathbf{r}} &= (\dot{x} - L\dot{\theta}\cos\theta)\mathbf{i} - L\dot{\theta}\sin\theta\mathbf{j} \\ \frac{\partial^2 \mathbf{r}}{\partial t^2} = \ddot{\mathbf{r}} &= (\ddot{x} + L\dot{\theta}^2\sin\theta - L\ddot{\theta}\cos\theta)\mathbf{i} + (-L\dot{\theta}^2\cos\theta - L\ddot{\theta}\sin\theta)\mathbf{j}\end{aligned}$$

Her eksen dahilindeki ivme değerleri ve mesnet noktası baz alınarak sarkaca etki eden kuvvetler;

$$\begin{aligned}\mathbf{N} &= m\{\ddot{x} + L\dot{\theta}^2\sin\theta - L\ddot{\theta}\cos\theta\} \\ \mathbf{P} &= m\{-L\dot{\theta}^2\cos\theta - L\ddot{\theta}\sin\theta\}\end{aligned}$$

denklemlerine göre belirlenebilir. Diğer taraftan arabaya etki eden yatay kuvvetler dikkate alındığında;

$$\mathbf{F} - b\dot{x} - \mathbf{N} = M\ddot{x}$$

ifadesi kolayca yazılabilir. Dikey eksen için de benzer yaklaşım sergilenebilir, ancak dikey eksen hareketin olmadığı düşünüldüğünde yazılacak denklemin getirisinin olmayacağı anlaşılabilir. Hülasa, daha önce  $\mathbf{N}$  kuvveti için elde edilen denklem ile kuvvetlerin dikkate alındığı son denklem beraberce değerlendirilirse;

$$\mathbf{F} = b\dot{x} + m\{\ddot{x} + L\dot{\theta}^2\sin\theta - L\ddot{\theta}\cos\theta\} + M\ddot{x}$$

eşitliğine veya uygun bir düzenleme ile;

---

<sup>i</sup> Position

<sup>ii</sup> Velocity

<sup>iii</sup> Angle

<sup>iv</sup> Angular velocity



$$F = (M + m)\ddot{x} + b\dot{x} + m\{L\dot{\theta}^2 \sin\theta - L\ddot{\theta} \cos\theta\}$$

sonucuna ulaşılır. Elde edilen bu denklem, eksenel ve açısal konum ile konum değişikliklerinin, etki eden kuvvetle olan ilişkisini veren ikinci mertebeden adi diferansiyel denklemdir. Denklem çözümünü analitik veya sayısal olarak elde etmek mümkündür. Ancak kontrol sürecinde çözümün uygun zaman diliminde elde edilememesi veya zamansal gecikmeler nedeniyle istenilen konuma istenilen anda ulaşamaması gibi sorunlarla karşılaşılabilir.

Diğer taraftan analitik olarak elde edilen bu ifade yerine, aynı sorun başka bir yöntemle, bulanık mantık yaklaşımı ile de kolayca çözülebilir. Bu bağlamda bulanık mantık süreci ve kontrol işlem adımları oldukça hızlı ve esnek çözümler üretebilir.

Bahse konu sorunun çözümü sadedinde bulanık mantık sürecine ait işlem adımlarını algoritmik çerçevede aşağıdaki gibi ifade edebiliriz<sup>(95)</sup>;

1. Dilsel değişkenler ile terimlerin tanımlanması,
2. Değişken ve terimlere ait üyelik fonksiyonlarının belirlenmesi,
3. Kural tabanının ifade edilmesi,
4. Girdi verilerinin bulanıklaştırılması,
5. Girdi verilerine göre kuralların değerlendirilmesi,
6. Etki eden kuralların toparlanması,
7. Toparlanma sonrası durulama işleminin gerçekleştirilmesi.

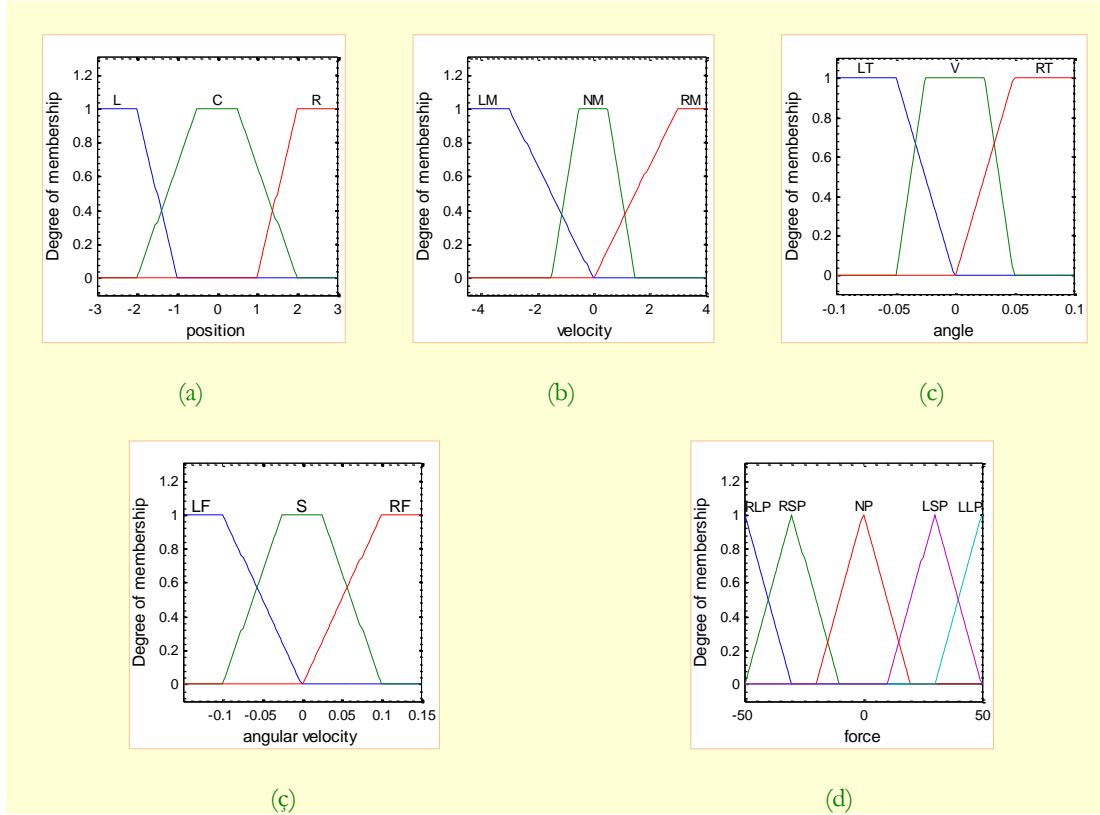
İşlem adımlarından ilki bağlamında dilsel değişkenler olarak  $x, \theta, \dot{x}, \dot{\theta}, F$  değişkenleri dikkate alınabilir. Bu değişkenlerden ilk ikisi  $(x, \theta)$  ölçüm yoluyla, sonraki ikisi  $(\dot{x}, \dot{\theta})$  zamana göre türetilerek kolayca belirlenebilir. Ayrıca bu değişkenlerin girdi verilerini oluşturduğu da söylenebilir. İfade edilen son değişken ise çıktı verisi niteliğindeki dengeleme kuvvetidir.

Eksenel konum değişkeni niteliğindeki  $x$  değişkenine ait terimler; “solda (L)”, “ortada (C)” ve “sağda (R)” olarak tasnif edilebilir. Açısal konum değişkeni niteliğindeki  $\theta$  değişkenine ait terimler; “sola meyilli (LT)”, “dikey (V)” ve “sağa meyilli (RT)” pozisyonu olarak sınıflandırılabilir.

Eksenel hız değişkeni niteliğindeki  $\dot{x}$  değişkenine ait terimler; “sola gitme (LM)”, “hareketsiz (NM)” ve “sağa gitme (RM)” şeklinde tarif edilebilir. Açısal hız değişkeni niteliğindeki  $\dot{\theta}$  değişkenine ait terimler; “sola yatış (LF)”, “durağan (S)” ve “sağa yatış (RF)” biçiminde ifade edilebilir.

Çıktı değişkeni olan  $F$  kuvveti; “Sağa Büyükçe (RLP)”, “Sağa Hafifçe (RSP)”, “İtme (DP)”, “Sola Büyükçe (LLP)”, “Sola Hafifçe (LSP)” terimleri ile tasnif edilebilir.





**Şekil 4.75** Girdi ve çıktı verilerine ait üyelik fonksiyonu tanımlamaları; (a) hareket mesafesi, (b) hız, (c) açı, (ç) açısal hız ve (d) yatay kuvvet.

Sürecin ikinci adımında değişkenlere ait üyelik fonksiyon tanımlamaları yapılır. Şekil 4.75 dâhilinde her bir değişkene ait üyelik fonksiyonu tanımlamaları görülmektedir. Gözlemlenebileceği üzere üyelik fonksiyonu olarak trapez tipi üyelik fonksiyonu tercih edilmiştir. Üyelik fonksiyonlarının belirlenmesi akabinde kural tabanının tanımlanması gerekir. Kural tabanı çerçevesinde hız ve konum ikilisi ayrı, açı ve açısal hız ikilisi ayrı bir kural tabanı olarak dikkate alınabilir. Bu bağlamda konum ve hız girdi verilerine yönelik kuvvet çıktısı terimleri müteakip matris dâhilinde tanımlanabilir;

<sup>1</sup> L: left, C: center, R: right; LM: left motion, NM: no motion, RM: right motion; RLP: right largely push, RSP: right slightly push, DP: don't push, LLP: left largely push, LSP: left slightly push.



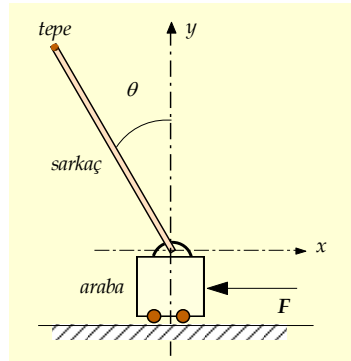


		$x$		
		$r_{ij}^1$	L	C
$\dot{x}$	LM	RLP	RSP	DP
	NM	RSP	DP	LSP
	RM	DP	LSP	LLP

Buna göre belirlene kurallar lâfzî olarak;

$r_{11}^1$ :	Eğer arabanın konumu “solda (L)”	ve	hızı “sola gitme (LM)”	ise	“sağa büyükçe (RLP)” it.
$r_{21}^1$ :	Eğer arabanın konumu “solda (L)”	ve	hızı “hareketsiz (NM)”	ise	“sağa hafifçe (RSP)” it.
$r_{31}^1$ :	Eğer arabanın konumu “solda (L)”	ve	hızı “sağa gitme (RM)”	ise	“itme (DP)”.
$r_{12}^1$ :	Eğer arabanın konumu “ortada (C)”	ve	hızı “sola gitme (LM)”	ise	“sola hafifçe (RSP)” it.
$r_{22}^1$ :	Eğer arabanın konumu “ortada (C)”	ve	hızı “hareketsiz (NM)”	ise	“itme (DP)”.
$r_{32}^1$ :	Eğer arabanın konumu “ortada (C)”	ve	hızı “sağa gitme (RM)”	ise	“sola hafifçe (LSP)” it.
$r_{13}^1$ :	Eğer arabanın konumu “sağda (R)”	ve	hızı “sola gitme (LM)”	ise	“itme (DP)”.
$r_{23}^1$ :	Eğer arabanın konumu “sağda (R)”	ve	hızı “hareketsiz (NM)”	ise	“sola hafifçe (LSP)” it.
$r_{33}^1$ :	Eğer arabanın konumu “sağda (R)”	ve	hızı “sağa gitme (RM)”	ise	“sola büyükçe (RLP)” it.

şeklinde ifade edilebilir. Dikkat edileceği üzere kurallar, arabanın konum ve hızına göre tanımlanmaktadır. Dolayısıyla burada nirengi noktası sarkaç değil, arabadır. Arabanın konumu ifade edilirken, sarkacın tepe noktasına göre konumlandırma tarif edilmektedir. Örneğin Şekil 4.76’da görülebileceği üzere arabanın sağda olması sarkacın tepe noktasına göre arabanın sağda olması anlamına gelmektedir.



Şekil 4.76 Araba ile sarkacın konum tanımlaması.

Diğer taraftan sarkacın tepe noktasının açısal konum ve hız girdi verilerine göre kuvvet çıktısı verileri, müteakip matris dâhilinde tanımlanabilir;



		$\theta$		
		$r_{ij}^2$	LT	V
$\dot{\theta}$	LF	LLP	LSP	DP
	S	LSP	DP	RSP
	RF	DP	RSP	RLP

Buna göre belirlenen kurallar mantık işaretleri yardımıyla lâfzî olarak;

$r_{11}^2$	Eğer tepe noktasının açısıl konumu “sola meyilli (LT)”	$\wedge$	açısıl hızı “sola düşme (LF)”	$\rightarrow$	“sola büyükçe (LLP)” it.
$r_{21}^2$	Eğer tepe noktasının açısıl konumu “sola meyilli (LT)”	$\wedge$	açısıl hızı “durağan (S)”	$\rightarrow$	“sola hafifçe (LSP)” it.
$r_{31}^2$	Eğer tepe noktasının açısıl konumu “sola meyilli (LT)”	$\wedge$	açısıl hızı “sağa düşme (RF)”	$\rightarrow$	“itme (DP)”.
$r_{12}^2$	Eğer tepe noktasının açısıl konumu “dikey (V)”	$\wedge$	açısıl hızı “sola düşme (LF)”	$\rightarrow$	“sola hafifçe (LSP)” it.
$r_{22}^2$	Eğer tepe noktasının açısıl konumu “dikey (V)”	$\wedge$	açısıl hızı “durağan (S)”	$\rightarrow$	“itme (DP)”.
$r_{32}^2$	Eğer tepe noktasının açısıl konumu “dikey (V)”	$\wedge$	açısıl hızı “sağa düşme (RF)”	$\rightarrow$	“sağa hafifçe (RSP)” it.
$r_{13}^2$	Eğer tepe noktasının açısıl konumu “sağa meyilli (RT)”	$\wedge$	açısıl hızı “sola düşme (LF)”	$\rightarrow$	“itme (DP)”.
$r_{23}^2$	Eğer tepe noktasının açısıl konumu “sağa meyilli (RT)”	$\wedge$	açısıl hızı “durağan (S)”	$\rightarrow$	“sağa hafifçe (RSP)” it.
$r_{33}^2$	Eğer tepe noktasının açısıl konumu “sağa meyilli (RT)”	$\wedge$	açısıl hızı “sağa düşme (RF)”	$\rightarrow$	“sağa büyükçe (RLP)” it.

şeklinde ifade edilebilir<sup>1</sup>. Kural tabanının belirlenmesi sonrasında girdi verilerinin bulanıklaştırılması gerekir. Bu bağlamda arabanın konumu ve hızı ile sarkacın açısıl konumu ve hızının;

$x$	$\dot{x}$	$\theta$	$\dot{\theta}$
1.5	1	-0.05	-0.03

şeklinde belirlendiğini öngörelim. Girdi veri değerlerine göre konum “ortada (C)” ve “sağda (R)” üyelik fonksiyonlarını, hız “hareketsiz (NM)” ve “sağa gitme (RM)” üyelik fonksiyonlarını, açı “sola meyilli (LT)” üyelik fonksiyonunu, açısıl hızı değeri ise “sola yatış (LF)” ve “durağan (S)” üyelik fonksiyonlarını etkilemektedir. Buna mukabil etkilenen kurallar;  $r_{22}^1$ ;  $r_{23}^1$ ;  $r_{23}^1$ ;  $r_{33}^1$ ;  $r_{11}^2$ ;  $r_{21}^2$  şeklinde sıralanabilir. Örnek olması açısından; konum ve hız verilerinin bulanıklaştırılması ve akabinde de etki ettiği kurala göre ve mesela  $r_{22}^1$  kuralına göre çıkarsama değerinin belirlenmesi müteakip şekildeki gibi yapılabilir. Buna göre  $x = 1.5$  değerinin “C” ve “R” üyelik fonksiyonlarındaki değerini hesaplamak için öncelikle her bir fonksiyona ait üyelik fonksiyonları tanımlanmalı. “C” üyelik fonksiyonu, trapez tanımı gereği;

$$\mu_C[-2, -0.5, 0.5, 2](x) = \begin{cases} \frac{x+2}{1.5} & -2 \leq x \leq -0.5 \\ 1 & -0.5 < x \leq 0.5 \\ \frac{x-2}{-1.5} & 0.5 < x \leq 2 \end{cases}$$

<sup>1</sup>LT: left trend, V: vertical, RT: right trend; LF: left fall, S: stable, RF: right fall.



şeklinde tanımlanabilir. Bu ifadeye  $x$  değeri yerine koyulursa;

$$\mu_C = \frac{1.5 - 2}{-1.5} = 0.33$$

sonucuna varılır. Aynı yaklaşım “R” üyelik fonksiyonu için de uygulanabilir. “R” üyelik fonksiyonu, trapez tanımını gereği;

$$\mu_R[1,2,3,4](x) = \begin{cases} \frac{x-1}{1} & 1 \leq x \leq 2 \\ 1 & 2 < x \leq 3 \\ \frac{x-4}{-1} & 3 < x \leq 4 \end{cases}$$

şeklinde tanımlanabilir. Bu ifadeye  $x$  değeri yerine koyulursa;

$$\mu_R = \frac{1.5 - 1}{1} = 0.5$$

sonucuna ulaşılır. Benzer yaklaşım hız verisi için de geçerlidir.  $\dot{x} = 1$  hız değerinin “NM” ve “RM” üyelik fonksiyonlarındaki değerini hesaplamak için yine her bir fonksiyona ait üyelik fonksiyonları tanımlanmalıdır. “NM” üyelik fonksiyonu, trapez tanımını gereği;

$$\mu_{NM}[-1.5, -0.5, 0.5, 1.5](\dot{x}) = \begin{cases} \frac{\dot{x} + 1.5}{1} & -1.5 \leq \dot{x} \leq -0.5 \\ 1 & -0.5 < \dot{x} \leq 0.5 \\ \frac{\dot{x} - 1.5}{-1} & 0.5 < \dot{x} \leq 1.5 \end{cases}$$

şeklinde tanımlanabilir. Bu ifadeye  $\dot{x}$  değeri yerine koyulursa;

$$\mu_{NM} = \frac{1 - 1.5}{-1} = 0.5$$

sonucuna varılır. Aynı şekilde “RM” üyelik fonksiyonu, trapez tanımını gereği;

$$\mu_{RM}[0,3,4.5,5](\dot{x}) = \begin{cases} \frac{\dot{x}}{3} & 0 \leq \dot{x} \leq 3 \\ 1 & 3 < \dot{x} \leq 4.5 \\ \frac{\dot{x} - 5}{-0.5} & 4.5 < \dot{x} \leq 5 \end{cases}$$

şeklinde tanımlanabilir.





Bu ifadeye  $x$  değeri yerine koyulursa;

$$\mu_{RM} = \frac{1}{3} = 0.33$$

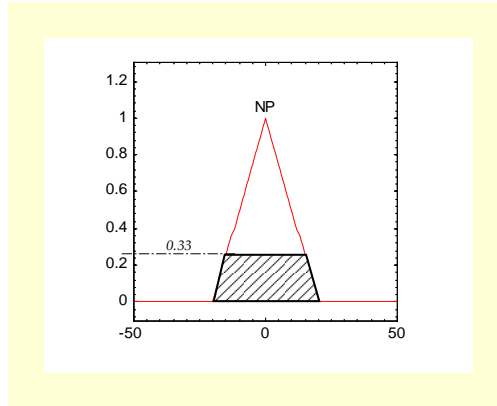
sonucuna ulaşılır. Diğer taraftan örnek olarak dikkate alınan  $r_{22}^1$  kuralı tekrar hatırlanırsa;

$r_{22}^1$ ; Eğer arabanın konumu “ortada (C)” ve hızı “hareketsiz (NM)” şeklinde ise “itme (DP)”.

ve bu kurala göre çıkarsama yapılırsa;

$$\min\{\mu_C, \mu_{NM}\} \rightarrow \min\{0.33, 0.5\} \rightarrow 0.33$$

değeri elde edilir. Hatırlanacağı üzere “ve” bağlacı, bulanık mantık çerçevesinde “minimum” değer olarak dikkate alınmaktaydı. Elde edilen bu değerın kuvvet çıktısındaki karşılığı, yine kuralda ifade edilen değişken terimine göre belirlenir. Dikkate alınan kurala göre etkilenen “NP” terimine ait üyelik fonksiyonunun  $\mu_{NP}(f) = 0.33$  değerine karşılık gelen alan tanımı, müteakip şekilde görülmektedir;



Şekil 4.77  $r_{22}^1$  kuralına göre elde edilen çıkarsama sonucu.

Diğer bir örnek uygulama  $r_{21}^2$  için yapılabilir. Hatırlanacağı üzere söz konusu kural;

$r_{21}^2$ ; Eğer tepe noktasının açıl konumu “sola meyilli (LT)” ve açıl hızı “durağan (S)” şeklinde ise “sola hafifçe (LSP)” it.







şeklindeydi. Buna göre  $\theta = 1.5$  değerinin “LT” üyelik fonksiyonlarındaki değerini hesaplamak için öncelikle bu fonksiyona ait üyelik fonksiyonları belirlenmelidir. “LT” üyelik fonksiyonu, trapez tanımı gereği;

$$\mu_{LT}[-0.15, -0.1, -0.05, 0](x) = \begin{cases} \frac{\theta + 0.15}{0.05} & -0.15 \leq \theta \leq -0.1 \\ 1 & -0.1 < \theta \leq -0.05 \\ \theta & -0.05 < \theta \leq 0 \\ -0.05 & \end{cases}$$

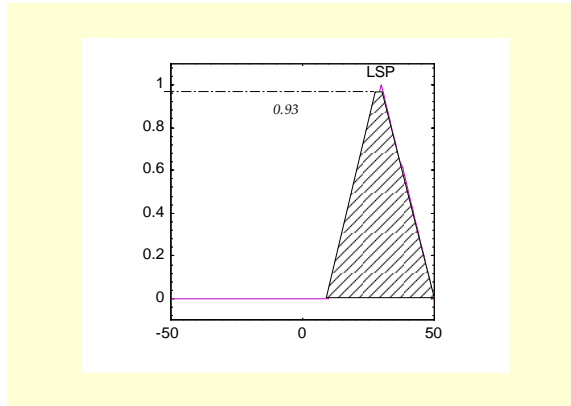
şeklinde tanımlanabilir. Bu ifadeye  $\theta$  değeri yerine koyulursa;

$$\mu_{LT} = 1$$

sonucuna varılır. Benzer şekilde açışal hız değeri  $\dot{\theta} = -0.03$  için “S” üyelik fonksiyonu, trapez tanımı gereği;

$$\mu_S[-0.1, -0.025, 0.025, 0.1](\dot{\theta}) = \begin{cases} \frac{\dot{\theta} + 0.1}{0.075} & -0.1 \leq \dot{\theta} \leq -0.025 \\ 1 & -0.025 < \dot{\theta} \leq 0.025 \\ \frac{\dot{\theta} - 0.1}{-0.075} & 0.025 < \dot{\theta} \leq 0.1 \end{cases}$$

şeklinde ifade edilebilir.



Şekil 4.78  $r_{21}^2$  kuralına göre elde edilen çıkarsama sonucu.



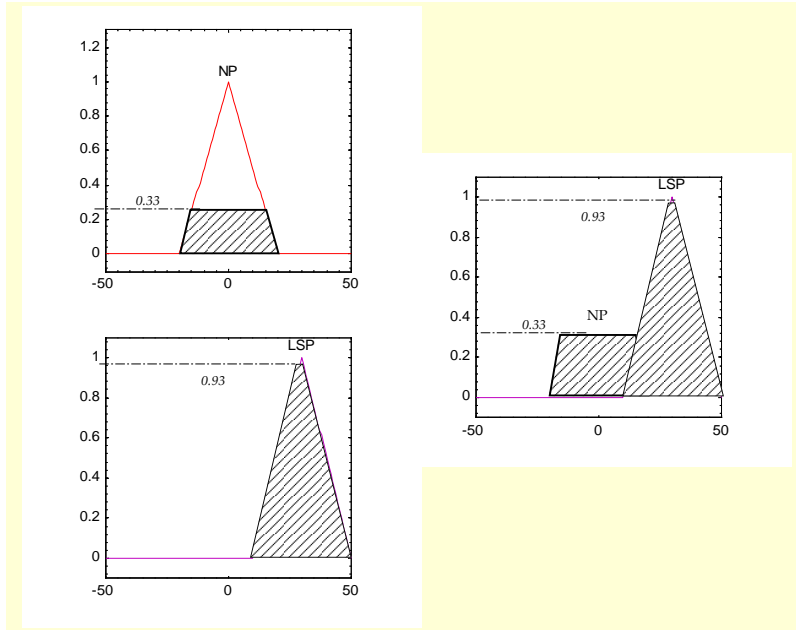
Bu ifadede  $\theta$  değeri yerine koyulursa;

$$\mu_S = \frac{-0.03 + 0.1}{0.075} = 0.93$$

sonucuna ulaşılır. Dikkate alınan kurala göre çıkarsama yapılırsa;

$$\min\{\mu_{LT}, \mu_S\} \rightarrow \min\{1, 0.93\} \rightarrow 0.93$$

değeri elde edilir. Belirlenen bu değerın kuvvet çıktısındaki karşılığı, kuralda ifade edilen değişken terimine göre belirlenir. Dikkate alınan kurala göre etkilenen “LSP” terimine ait üyelik fonksiyonunun  $\mu_{LSP}(f) = 0.93$  değerine karşılık gelen alan tanımı, Şekil 4.78’de görülmektedir. Örnek olması hasebiyle uygulamaları yapılan kurallara ilave olarak etkilenen diğer kurallar;  $r_{32}^1, r_{23}^1, r_{33}^1, r_{11}^2$  olup benzer çıkarsamalar onlar için de tekrar edilebilir. Kural değerlendirmeleri ve çıkarsama işlemlerini takip eden işlem adımı, “etki eden kuralların toparlanması”dır. Bu bağlamda yine örnek olması açısından  $r_{21}^1$  ile  $r_{21}^2$  kurallarının toparlanması örneğini burada ele alabiliriz. Buna göre her iki kural sonucu elde edilen toparlama alanını Şekil 4.79’da görmek mümkündür.



Şekil 4.79 Sadece  $r_{21}^1$  ile  $r_{21}^2$  kurallarının toparlanması sonucu elde edilen alan.

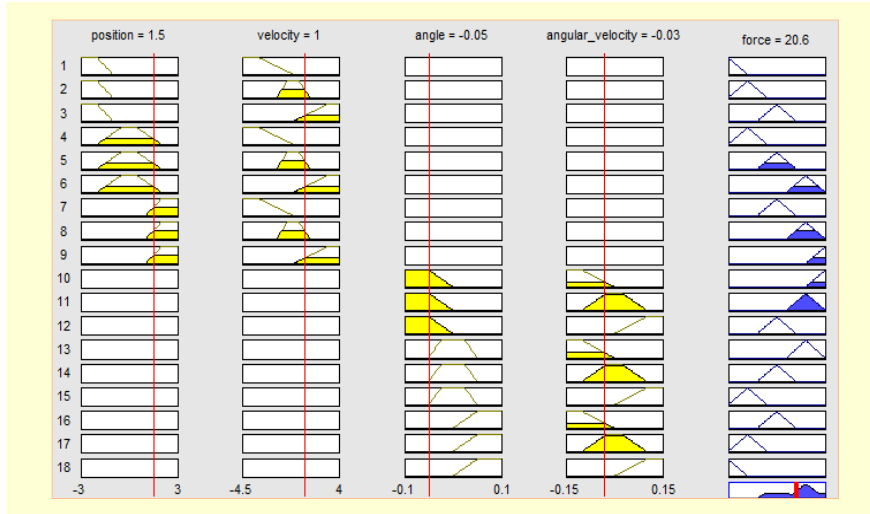




Hatırlanacağı üzere toparlama işlemlerinde “veya” bağlacı ve bu bağlamda azami değer dikkate alınmaktadır. Dolayısıyla kuvvete ait üyelik fonksiyonu değeri;

$$\mu(f) = \max\{\mu_{NP}(f), \mu_{LSP}(f)\}$$

ifadesine göre kolayca belirlenebilir. Benzer işlemler diğer çıkarsamalar da dikkate alınarak yapılırsa Şekil 4.80’deki gibi bir toparlama sonucu elde edilebilir. Buna göre toplamda 18 kural dikkate alınmakta olup bu kurallardan ilk dokuz adedi konum ve hızla ilgili, son dokuz adedi açısal konum ve açısal hızla ilgilidir. Verili girdi değerlerinin değerlendirilmesi sonucu 18 kuraldan sadece altı adedinin etkilendiği gözlemlenmekte ve bu kurallar neticesinde yayılı bir alan elde edilmektedir. Şekilde görülen sütunlar, dilsel değişkenleri ve bu değişkenlerin etkilenen terimlerini (girdi verileri için sarı renkli) temsil etmekte, buna mukabil etkilenen kurallar ve bu kuralların kuvvet terimleri üzerindeki çıkarsamaları ise son sütunda (mavi renkli) gözlemlenmektedir. Dilsel değişkenlere ait grafiklerde görülen dikey hatlar (kırmızı renkli), her bir değişkene ait girdi verisini göstermektedir. Buna göre [1.5 1 -0.05 -0.03] girdi verilerine karşılık sola doğru 20.6 birimlik bir kuvvetle itmek gerekmektedir. Söz konusu kuvvet değeri, toparlama sonucu elde edilen alansal bölgenin ağırlık merkezinin bulunması ile belirlenmiş olup son sütunun son satırı, toparlama alan sonucunu ve ağırlık merkezi hattını (kırmızı dikey kalın çizgi) göstermektedir.



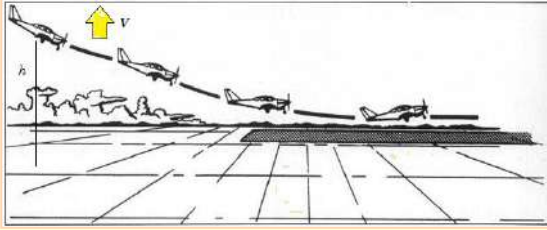
Şekil 4.80 Tanımlı kurallara göre elde edilen çıkarsama sonucu.





## Uçağa ait son yaklaşma ve iniş kontrolü.

MATLAB uygulaması



(a) Örnek iniş hattı <sup>(101)</sup>



(b) Örnek iniş güzergâhı <sup>(102)</sup>



(c) T-41D uçağı <sup>(103)</sup>

**Şekil 4.81** Son yaklaşma ve iniş adımlarına göre uçağın pozisyonu ve bazı görüntü kesitleri.

Uçakların son yaklaşma ve iniş kumandasını, bulanık mantık ile kontrol etmek mümkündür. Bu bağlamda dikkate alınan değişkenleri; uçağın dikey olarak yerden uzaklığı ( $h$ ), dikey hızı ( $v$ ) ve etki edilecek dikey kuvvet ( $f$ ) şeklinde belirleyebiliriz. Genellikle yükseklik ve hız verisine göre sürekli olarak etki eden kuvvetin güncellenmesi ve yumuşak bir inişin gerçekleştirilmesi istenir. Şekil 4.81’de inişe hazırlanan bir uçağa ait grafiksel ve gerçek görüntüler görülmektedir. Newton’ın 2nci ilkesi kapsamında kuvvet;

$$f = m \frac{dv}{dt}$$

ifadesine göre hesaplanabilir. Bu ifadede temsili olarak  $m = 1$  ve  $dt = 1$  değerleri dikkate alınırsa;

$$v_2 = v_1 + f$$

eşitliği elde edilebilir. Benzer şekilde dikey hızın tanımından hareketle;

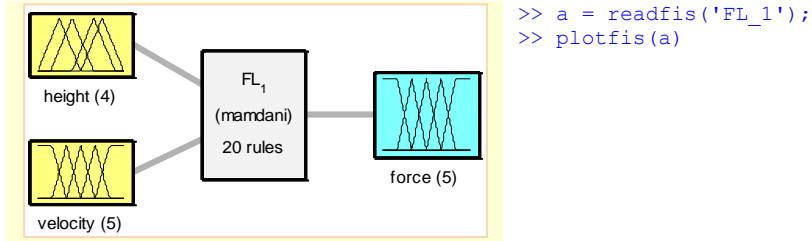
$$v = \frac{dh}{dt}$$



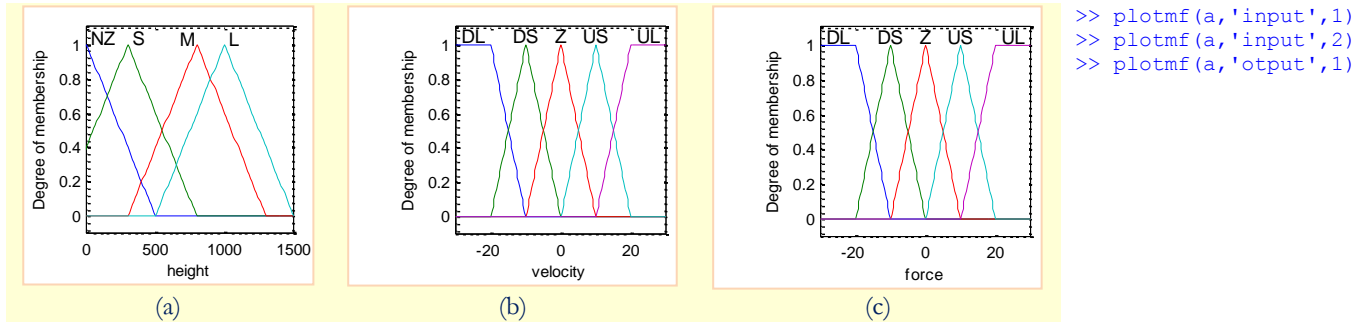
denklemi yazılıp,

$$h_2 = h_1 + v$$

şeklindeki yükseklik denklemine kolayca ulaşılabilir. Basitleştirilmiş yaklaşıma göre kuvvet verisinden hareketle dikey hız ve bu veriden hareketle de yükseklik güncellenmektedir. Tabiatıyla güncel hız ve yükseklik verisine göre de kuvvet değeri güncellenmelidir. Bu bağlamda yükseklik ve hız verileri girdi verisi, kuvvet verisi ise çıktı verisi olacak şekilde bir bulanık mantık sistemi tasarlamak mümkündür. Matlab dâhilindeki “Fuzzy Logic Design” uygulaması veya “FIS (Fuzy Interface System) Editor” kullanılarak söz konusu sistemin tasarlanması yapılabilir. Müteakip şekilde sistem tasarımı görülmektedir;



Program dâhilindeki “Membership Function Editor” kullanılarak her bir dilsel değişken ve terimleri tanımlanabilir.



Şekil 4.82 (a)Yükseklik, (b) hız ve (c) kuvvet için üyelik fonksiyonu tanımlamaları.

Buna göre yükseklik dilsel değişkeni; “yere yakın (NZ)”, “küçük (S)”, “orta (M)” ve “büyük (L)” şeklindeki dört ayrı terimle<sup>i</sup> tasnif edilebilir. Ayrıca ifade edilen her bir terim için üçgensel fonksiyon tanımına da

<sup>i</sup>“Near zero (NZ)”, “small (S)”, “medium (M)”, “large (L)”



gidilebilir. Benzer şekilde hız dilsel değişkeni; “aşağı büyük (DL)”, “aşağı küçük (DS)”, “sıfır (Z)”, “yukarı küçük (US)” ve “yukarı büyük (UL)” şeklindeki beş ayrı terimle<sup>i</sup> tasnif edilebilir. İfade edilen ilk ve son terimler yamuksal, diğer her bir terim içinse üçgensel fonksiyon tanımına gidilebilir. Çıktı niteliğindeki son değişken olan kuvvet, yine hız verisine benzer bir terimsel tanımlama ile betimlenebilir; “aşağı büyük (DL)”, “aşağı küçük (DS)”, “sıfır (Z)”, “yukarı küçük (US)” ve “yukarı büyük (UL)”. Şekil 4.82’de yükseklik, hız ve kuvvete ait üyelik fonksiyon tanımlamaları görülmektedir.

Değişkenlere ait üyelik fonksiyonlarının tanımlanması sonrasında, “Rule Editor” uygulaması yardımı ile kural tabanını oluşturmak mümkündür. Bu bağlamda toplamda 20 ayrı kural tanımlamak mümkün olup söz konusu kurallar matris formunda ve lâfzî olarak müteakip dizinlerde ifade edilmiştir;

		<b>v</b>				
		$r_{ij}$	DL	DS	Z	US
<b>h</b>	L	Z	DS	DL	DL	DL
	M	US	Z	DS	DL	DL
	S	UL	US	Z	DS	DL
	NZ	UL	UL	Z	DS	DS

$r_{11}$ :	Eğer uçağın yüksekliği “büyük (L)”	∧	hızı “aşağı büyük (DL)”	→	“sıfır (Z)” it.
$r_{12}$ :	Eğer uçağın yüksekliği “büyük (L)”	∧	hızı “aşağı küçük (DS)”	→	“aşağı küçük (DS)” it.
$r_{13}$ :	Eğer uçağın yüksekliği “büyük (L)”	∧	hızı “sıfır (Z)”	→	“aşağı büyük (DL)” it.
$r_{14}$ :	Eğer uçağın yüksekliği “büyük (L)”	∧	hızı “yukarı küçük (US)”	→	“aşağı büyük (DL)” it.
$r_{15}$ :	Eğer uçağın yüksekliği “büyük (L)”	∧	hızı “yukarı büyük (UL)”	→	“aşağı büyük (DL)” it.
$r_{21}$ :	Eğer uçağın yüksekliği “orta (M)”	∧	hızı “aşağı büyük (DL)”	→	“aşağı küçük (US)” it.
$r_{22}$ :	Eğer uçağın yüksekliği “orta (M)”	∧	hızı “aşağı küçük (DS)”	→	“sıfır (Z)” it.
$r_{23}$ :	Eğer uçağın yüksekliği “orta (M)”	∧	hızı “sıfır (Z)”	→	“aşağı küçük (DS)” it.
$r_{24}$ :	Eğer uçağın yüksekliği “orta (M)”	∧	hızı “yukarı küçük (US)”	→	“aşağı büyük (DL)” it.
$r_{25}$ :	Eğer uçağın yüksekliği “orta (M)”	∧	hızı “yukarı büyük (UL)”	→	“aşağı büyük (DL)” it.
$r_{31}$ :	Eğer uçağın yüksekliği “küçük (S)”	∧	hızı “aşağı büyük (DL)”	→	“yukarı büyük (UL)” it.
$r_{32}$ :	Eğer uçağın yüksekliği “küçük (S)”	∧	hızı “aşağı küçük (DS)”	→	“yukarı küçük (US)” it.
$r_{33}$ :	Eğer uçağın yüksekliği “küçük (S)”	∧	hızı “sıfır (Z)”	→	“sıfır (Z)” it.
$r_{34}$ :	Eğer uçağın yüksekliği “küçük (S)”	∧	hızı “yukarı küçük (US)”	→	“aşağı küçük (DS)” it.
$r_{35}$ :	Eğer uçağın yüksekliği “küçük (S)”	∧	hızı “yukarı büyük (UL)”	→	“aşağı büyük (DL)” it.
$r_{41}$ :	Eğer uçağın yüksekliği “yere yakın (NZ)”	∧	hızı “aşağı büyük (DL)”	→	“yukarı büyük (UL)” it.
$r_{42}$ :	Eğer uçağın yüksekliği “yere yakın (NZ)”	∧	hızı “aşağı küçük (DS)”	→	“yukarı büyük (UL)” it.
$r_{43}$ :	Eğer uçağın yüksekliği “yere yakın (NZ)”	∧	hızı “sıfır (Z)”	→	“sıfır (Z)” it.
$r_{44}$ :	Eğer uçağın yüksekliği “yere yakın (NZ)”	∧	hızı “yukarı küçük (US)”	→	“aşağı küçük (DS)” it.
$r_{45}$ :	Eğer uçağın yüksekliği “yere yakın (NZ)”	∧	hızı “yukarı büyük (UL)”	→	“aşağı küçük (DS)” it.

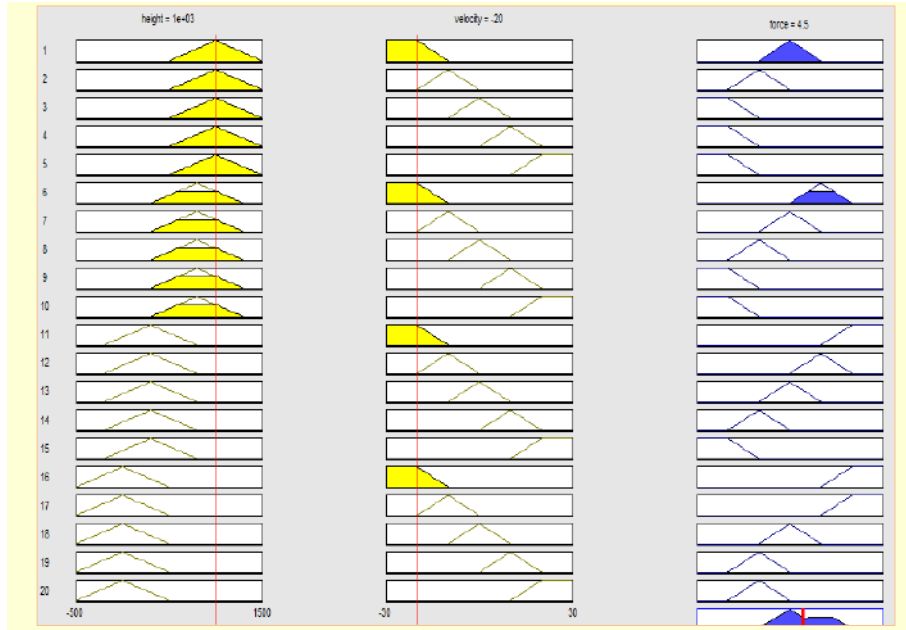
<sup>i</sup>“Down large (DL)”, “down small (DS)”, “zero (Z)”, “up small (US)”, “ up large (UL)”



“Rule Editor” dâhilinde tanımlanan kuralların örnek olması bağlamında yazdırılması müteakip şekilde mümkündür;

```
>> a = readfis('FL_1');
>> showrule(a,1:3)
ans =
1. If (height is L) and (velocity is DL) then (force is Z) (1)
2. If (height is L) and (velocity is DS) then (force is DS) (1)
3. If (height is L) and (velocity is Z) then (force is DL) (1)
```

Buna göre toplamda 20 kural dikkate alınmakta olup girdi verilerine göre uygun çıktı verisinin üretilmesi, “Rule Viewer” dâhilinde mümkündür. Örneğin [1000 -20] girdi verilerine karşılık etkilenen kuralların  $r_{21}$  ve  $r_{11}$  olduğu kolayca söylenebilir. Söz konusu kuralların değerlendirilmesi neticesinde elde edilen kuvvet çıktı verisi, yukarı doğru 4.5 *br* şeklindedir. Müteakip şekilde söz konusu toparlama işlemleri ve elde edilen veri görülmektedir;



```
>> a = readfis('FL_1');
>> ruleview(a)
```

Şekil 4.83 Tanımlı kurallara göre elde edilen çıkarsama sonucu.

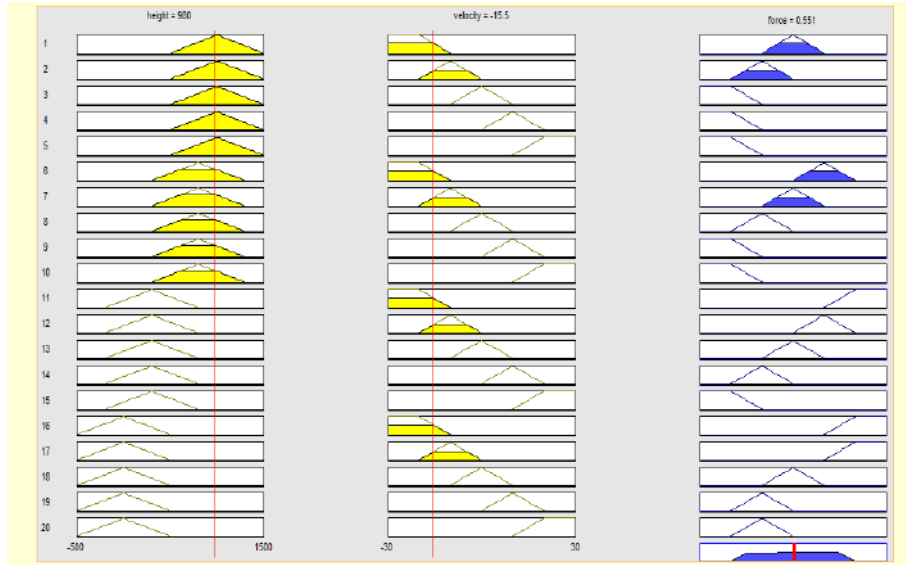


Etki eden kuvvet sonrasında uçağın yeni konum ve hız verileri;

$$h_2 = h_1 + v = 1000 - 20 = 980 \text{ ft}$$

$$v_2 = v_1 + f = -20 + 4.5 = -15.5 \text{ ft/s}$$

şeklinde hesaplanabilir. Elde edilen yeni yükseklik ve hız verilerine göre uygulanması gereken kuvvet, yine "Rule Viewer" aracılığı ile  $0.551 \text{ br}$  olarak kolayca bulunabilir. Müteakip şekilden de görüleceği üzere yeni veriler neticesinde  $r_{11}, r_{12}, r_{21}, r_{22}$  kuralları etkilenmekte ve bu bağlamda toplama işlemleri gerçekleştirilmektedir.



```
>> a = readfis('FL_1');
>> ruleview(a)
```

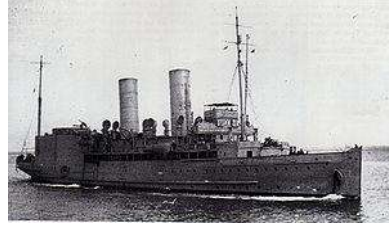
Şekil 4.84 Tanımlı kurallara göre elde edilen yeni çıkarsama sonucu.







## Tarihten Bir Sayfa



“**T**ürk askeri cenge hazırlanıyordu. Biraz sonra kopacak kıyametin heyecanı ile yüreğim çarparken, gözüm batarya dürbününde düşmanı seyrediyordum.”

Mustafa Ertuğrul  
Topçu Zabiti

HMS Ben-my-Chree uçak gemisi, İngiliz Vickers şirketinin 1907 yılında imal ettiği ve sonradan uçak gemisine dönüştürülen bir gemiydi. Bu gemi tarihte iki ilke sahiptir.

Bunlardan ilki, Çanakkale Savaşı esnasında üzerinden kalkan Short 184 tipi bir deniz uçağının dünyada ilk defa havadan atılan torpidoyu ateşleyerek bir Türk savaş gemisine saldırmasıdır.

İkinci ilk ise aynı geminin, yine Dünyada ilk defa kumandanlığını Mustafa Ertuğrul isimli bir Türk topçusunun yaptığı topçu bataryaları tarafından batırılmasıdır. Ocak 1917 tarihlerinde Meis adası açıklarında batan gemi dünyada batırılan ilk uçak gemisidir





- [1] Trees and a lonely girl, Alıntı < <http://alliswall.com/art/trees-and-a-lonely-girl>> [02 Ağustos 2016].
- [2] Alıntı < [www.quotesvalley.com/images/03/try-to-pretend-that-you-are-happy-slowly-it-becomes-your-habit.jpg](http://www.quotesvalley.com/images/03/try-to-pretend-that-you-are-happy-slowly-it-becomes-your-habit.jpg)> [11 Ağustos 2016].
- [3] Y. Kuya, K. Takeda, X. Zhang, and A. I. J. Forrester, Multifidelity surrogate modeling of experimental and computational aerodynamic data sets, *AIAA Journal*, Vol. 49, No. 2, 2011.
- [4] Surrogate model, from Wikipedia: The Free Encyclopedia. Alıntı < [https://en.wikipedia.org/wiki/Surrogate\\_model](https://en.wikipedia.org/wiki/Surrogate_model)> [04 Ağustos 2016].
- [5] Y. S. Ong, P. B. Nair, and A. J. Keane, Evolutionary optimization of computationally expensive problems via surrogate modeling, *AIAA Journal*, Vol. 41, No. 4, 2003.
- [6] P. I. K. Liakopoulos, I. C. Karpolis, and K. C. Giannakoglou, Grid enabled, hierarchical distributed metamodel-assisted evolutionary algorithms for aerodynamic shape optimization, *Future Generation Computer Systems*, vol. 24/7, July 2008, pp. 701-708.
- [7] Y. V. Pehlivanoglu and O. Baysal, Vibrational genetic algorithm enhanced with fuzzy logic and neural networks, *Aerospace Science and Technology* 14 (2010) 56-64.
- [8] M. Vitturi, F. Beux, A discrete gradient-based approach for aerodynamic shape optimization in turbulent viscous flow, *Finite Elements in Analysis and Design*, Volume 43, Issue 1, Nov. 2006, pp. 68-80.
- [9] J.F. Wang, J. Periaux, and M. Sefrioui, Parallel evolutionary algorithms for optimization problems in aerospace engineering, *Journal of Computational and Applied Mathematics* 149, 2002, pp. 155–169.
- [10] R. Duvigneau, Optimization using surrogate models. Alıntı < [https://team.inria.fr/acumes/files/2015/05/cours\\_meta.pdf](https://team.inria.fr/acumes/files/2015/05/cours_meta.pdf)> [04 Ağustos 2016].
- [11] Surrogate models, SUMO – Surrogate Modeling Lab, Alıntı < [sumo.intec.ugent.be/surrogates](http://sumo.intec.ugent.be/surrogates)> [05 Ağustos 2016].
- [12] Z. Qian, C. C. Seepersad, V. R. Joseph, J. K. Allen, and C. F. J. Wu, Building surrogate models based on detailed and approximate simulations, Alıntı < [https://www.me.utexas.edu/~ppmmlab/files/AD-ASME-02-24-2005\\_revised\\_.pdf](https://www.me.utexas.edu/~ppmmlab/files/AD-ASME-02-24-2005_revised_.pdf)> [05 Ağustos 2016].
- [13] S. E. Gano, H. Kim, and D. E. Brown, Comparison of three surrogate modeling techniques: Datascape, Kriging, and second order regression, 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, AIAA–2006–7048, 2006.
- [14] Z-H Han and K-S Zhang, Surrogate-based optimization, Alıntı < [cdn.intechopen.com/pdfs/30305/InTech-Surrogate\\_based\\_optimization.pdf](http://cdn.intechopen.com/pdfs/30305/InTech-Surrogate_based_optimization.pdf)> [05 Ağustos 2016].
- [15] Definition of regress, Alıntı < [www.merriam-webster.com/dictionary/regression](http://www.merriam-webster.com/dictionary/regression)> [09 Ağustos 2016].
- [16] Regression analysis, from Wikipedia: The Free Encyclopedia. Alıntı < [https://en.wikipedia.org/wiki/Regression\\_analysis](https://en.wikipedia.org/wiki/Regression_analysis)> [10 Ağustos 2016].
- [17] Regression methods, Alıntı < <https://onlinecourses.science.psu.edu/stat501/node/251>> [10 Ağustos 2016].



- [18] S. Ray, 7 types of regression techniques you should know, Alıntı < <https://www.analyticsvidhya.com/blog/2015/08/comprehensive-guide-regression/> > [10 Ağustos 2016].
- [19] Linear regression, from Wikipedia: The Free Encyclopedia. Alıntı < [https://en.wikipedia.org/wiki/Linear\\_regression](https://en.wikipedia.org/wiki/Linear_regression) > [10 Ağustos 2016].
- [20] Simple linear regression, from Wikipedia: The Free Encyclopedia. Alıntı < [https://en.wikipedia.org/wiki/Simple\\_linear\\_regression](https://en.wikipedia.org/wiki/Simple_linear_regression) > [10 Ağustos 2016].
- [21] Simple linear regression analysis, Alıntı<[reliawiki.org/index/php/Simple\\_Linear\\_Regression\\_Analysis](http://reliawiki.org/index/php/Simple_Linear_Regression_Analysis)> [10 Ağustos 2016].
- [22] Multiple linear regression analysis, Alıntı < [reliawiki.org/index/php/Multiple\\_Linear\\_Regression\\_Analysis](http://reliawiki.org/index/php/Multiple_Linear_Regression_Analysis) > [10 Ağustos 2016].
- [23] STATS Statistical Datasets, Alıntı<<http://people.sc.fsu.edu/~jburkardt/datasets/stats/asteroids.csv>> [11 Ağustos 2016].
- [24] Vandermonde matrix, from Wikipedia: The Free Encyclopedia. Alıntı < [https://en.wikipedia.org/wiki/Vandermonde\\_matrix](https://en.wikipedia.org/wiki/Vandermonde_matrix) > [12 Ağustos 2016].
- [25] STATS Statistical Datasets, Alligators, Alıntı < <http://people.sc.fsu.edu/~jburkardt/datasets/stats/alligators.csv> > [12 Ağustos 2016].
- [26] Nonlinear regression, from Wikipedia: The Free Encyclopedia. Alıntı < [https://en.wikipedia.org/wiki/Nonlinear\\_regression](https://en.wikipedia.org/wiki/Nonlinear_regression) > [12 Ağustos 2016].
- [27] V. Granville, 10 types of regressions: which one to use, Alıntı < [www.datasciencecentral.com/profiles/blogs/10-types-of-regressions-which-one-to-use](http://www.datasciencecentral.com/profiles/blogs/10-types-of-regressions-which-one-to-use) > [12 Ağustos 2016].
- [28] Introduction to Regression, Data and Statistical Services: Princeton University Library, Alıntı < [dss.princeton.edu/online\\_help/analysis/regression\\_intro.htm](http://dss.princeton.edu/online_help/analysis/regression_intro.htm) > [12 Ağustos 2016].
- [29] Inverse distance weighting, from Wikipedia: The Free Encyclopedia. Alıntı < [https://en.wikipedia.org/wiki/Inverse\\_distance\\_weighting](https://en.wikipedia.org/wiki/Inverse_distance_weighting) > [18 Ağustos 2016].
- [30] E. Brusilovskiy, Spatial interpolation: A brief introduction, Alıntı < [slideplayer.com/slide/763109](http://slideplayer.com/slide/763109) > [18 Ağustos 2016].
- [31] Spatial interpolation, Alıntı < [web.pdx.edu/~jduh/courses/geog492w12/Week6b.pdf](http://web.pdx.edu/~jduh/courses/geog492w12/Week6b.pdf) > [18 Ağustos 2016].
- [32] D. G. Krige, A statistical approach to some mine valuation and allied problems on the Witwatersrand, Thesis, University of Witwatersrand, 1951, Alıntı < <https://core.ac.uk/download/files/979/39676542.pdf> > [19 Ağustos 2016].
- [33] G. Matheron, The theory of regionalized variables and its applications, Published by the Ecole Nationale Supérieure des Mines de Paris, 1971, Alıntı < [cg.ensmp.fr/bibliotheque/public/MATHERON\\_Ouvrage\\_00167.pdf](http://cg.ensmp.fr/bibliotheque/public/MATHERON_Ouvrage_00167.pdf) > [19 Ağustos 2016].
- [34] A. Lichtenstern, Kriging methods in spatial statistics, Bachelor's Thesis, Technische Universität München, 2013, Alıntı < <https://mediatum.ub.tum.de/doc/1173364/1173364.pdf> > [19 Ağustos 2016].
- [35] G. Bohling, Kriging, Alıntı < [people.ku.edu/~gbohling/cpe940/Kriging.pdf](http://people.ku.edu/~gbohling/cpe940/Kriging.pdf) > [19 Ağustos 2016].



- [36] J. Li and A. D. Heap, A review of spatial interpolation methods for environmental scientists, *Geoscience Australia Record* 2008/23, Alıntı < [www.ga.gov.au/webtemp/image\\_cache/GA12526.pdf](http://www.ga.gov.au/webtemp/image_cache/GA12526.pdf) > [19 Ağustos 2016].
- [37] H. Wackernagel, Basics in geostatistics 2 geostatistical interpolation/estimation: Kriging methods, Alıntı < <https://nersc.no/sites/www.nersc.no/files/Basics2kriging.pdf> > [19 Ağustos 2016].
- [38] M. De Smith, M. F. Goodchild, P. A. Longley, *Geospatial analysis*, 5th Edition, Alıntı < [spatialanalysisonline.com/HTML/index.html?kriging\\_interpolation.htm](http://spatialanalysisonline.com/HTML/index.html?kriging_interpolation.htm) > [19 Ağustos 2016].
- [39] Spatial interpolation techniques (1), Alıntı < [www.nuim.ie/staff/dpringle/gis/gis09.pdf](http://www.nuim.ie/staff/dpringle/gis/gis09.pdf) > [19 Ağustos 2016].
- [40] Kriging, from Wikipedia: The Free Encyclopedia. Alıntı < <https://en.wikipedia.org/wiki/Kriging> > [22 Ağustos 2016].
- [41] Coğrafi Bilgi Sistemlerine Giriş: mekansal ve öznitelik verilerinin bütünleşik analizi, Ulusal Açık Ders malzemeleri Konsorsiyumu, Alıntı < [www.acikders.org.tr/pluinfile.php/702/mod\\_resource/content/1/unite7\\_Mekansal\\_Oznitelik\\_Verilerinin\\_Butunlesik\\_Analizi\\_guncel.pdf](http://www.acikders.org.tr/pluinfile.php/702/mod_resource/content/1/unite7_Mekansal_Oznitelik_Verilerinin_Butunlesik_Analizi_guncel.pdf) > [22 Ağustos 2016].
- [42] Spatial interpolation, Alıntı < [personal.colby.edu/personal/m/mgimond/Spatial/Interpolation.html](http://personal.colby.edu/personal/m/mgimond/Spatial/Interpolation.html) > [22 Ağustos 2016].
- [43] I. Clark, *Geostatistics*, Alıntı < [www.kriging.com/RSMA1978/](http://www.kriging.com/RSMA1978/) > [23 Ağustos 2016].
- [44] Kriging interpolation, Alıntı < [lazarus.elte.hu/hun/digkonyv/havas/mellekl/vm25/vma07.pdf](http://lazarus.elte.hu/hun/digkonyv/havas/mellekl/vm25/vma07.pdf) > [24 Ağustos 2016].
- [45] A. A. Daya and H. Bejari, A comparative study between simple kriging and ordinary kriging for estimating and modeling the Cu concentration in Chehlkureh deposit, SE Iran, *Arab J Geosci* (2015) 8:6003–6020.
- [46] Universal Kriging algorithm, Alıntı < [spatial-analysis.net/ILWIS/htm/ilwisapp/universal\\_kriging\\_algorithm.htm](http://spatial-analysis.net/ILWIS/htm/ilwisapp/universal_kriging_algorithm.htm) > [29 Ağustos 2016].
- [47] Kriging and cokriging, Alıntı < [petrowiki.org/Kriging\\_and\\_cokriging](http://petrowiki.org/Kriging_and_cokriging) > [29 Ağustos 2016].
- [48] H. Xu, J. Sun, B. Russell, and C. Innanen, Porosity prediction using cokriging with multiple secondary datasets, Alıntı < [www.cggv.com/technicalDocuments/cggv\\_0000025392.pdf](http://www.cggv.com/technicalDocuments/cggv_0000025392.pdf) > [29 Ağustos 2016].
- [49] CoKriging algorithm, Alıntı < [spatial-analysis.net/ILWIS/htm/ilwisapp/cokriging\\_algorithm.htm](http://spatial-analysis.net/ILWIS/htm/ilwisapp/cokriging_algorithm.htm) > [29 Ağustos 2016].
- [50] Cross Variogram Algorithm, Alıntı < [spatial-analysis.net/ILWIS/htm/ilwisapp/cross\\_variogram\\_algorithm.htm](http://spatial-analysis.net/ILWIS/htm/ilwisapp/cross_variogram_algorithm.htm) > [29 Ağustos 2016].
- [51] S. Sakata, F. Ashida, and M. Zako, An efficient algorithm for Kriging approximation and optimization with large-scale sampling data, *Computer methods in applied mechanics and engineering*, 193 (2004) 385–404.
- [52] Support Vector Machines (SVMs), Alıntı < [www.svms.org/history.html](http://www.svms.org/history.html) > [27 Ekim 2016].
- [53] P. Paisitkriangkrai, *Linear Regression and Support Vector Regression*, Alıntı < [cs.adelaide.edu.au/~chshen/teaching/ML\\_SVR.pdf](http://cs.adelaide.edu.au/~chshen/teaching/ML_SVR.pdf) > [01 Kasım 2016].



- [54] B. Diri, Makine Öğrenmesine Giriş (Machine Learning - ML), Alıntı < <https://www.ce.yildiz.edu.tr/personal/banud/file/2634/Makine+Ogrenmesi-ML-10.pdf>> [02 Kasım 2016].
- [55] D. Özer, Destek Vektör Makineleri, Alıntı < <https://prezi.com/rca9jdnskzph/destek-vektor-makineleri/>> [01 Kasım 2016].
- [56] T. Fletcher, Support Vector Machines Explained, Alıntı < [www.tristanfletcher.co.uk/SVM%20Explained.pdf](http://www.tristanfletcher.co.uk/SVM%20Explained.pdf)> [09 Kasım 2016].
- [57] M. Law, A Simple Introduction to Support Vector Machines, Alıntı < [www.cise.ufl.edu/class/cis4930sp11dtm/notes/intro\\_svm\\_new.pdf](http://www.cise.ufl.edu/class/cis4930sp11dtm/notes/intro_svm_new.pdf)> [14 Kasım 2016].
- [58] Iris (plant), from Wikipedia: The Free Encyclopedia. Alıntı < [https://en.wikipedia.org/wiki/Iris\\_\(plant\)](https://en.wikipedia.org/wiki/Iris_(plant))> [17 Kasım 2016].
- [59] Alıntı <[http://www.plant-world-seeds.com/images/seed\\_images/IRIS\\_VERSICO\\_LOR/size3\\_500x500/IRIS\\_VERSICOLOR.JPG](http://www.plant-world-seeds.com/images/seed_images/IRIS_VERSICO_LOR/size3_500x500/IRIS_VERSICOLOR.JPG)> [17 Kasım 2016].
- [60] Alıntı <[https://upload.wikimedia.org/wikipedia/commons/2/27/Southern\\_Blue\\_Flag\\_Iris\\_\(iris\\_virginica\)\\_-\\_Flickr\\_-\\_Andrea\\_Westmoreland.jpg](https://upload.wikimedia.org/wikipedia/commons/2/27/Southern_Blue_Flag_Iris_(iris_virginica)_-_Flickr_-_Andrea_Westmoreland.jpg)> [17 Kasım 2016].
- [61] Alıntı < [https://www.math.umd.edu/~petersd/666/html/iris\\_with\\_labels.jpg](https://www.math.umd.edu/~petersd/666/html/iris_with_labels.jpg) > [17 Kasım 2016].
- [62] Support vector machine, from Wikipedia: The Free Encyclopedia. Alıntı < [https://en.wikipedia.org/wiki/Support\\_vector\\_machine](https://en.wikipedia.org/wiki/Support_vector_machine) > [15 Kasım 2016].
- [63] S. M. Clarke, J. H. Griebisch and T. W. Simpson, Analysis of Support Vector Regression for Approximation of Complex Engineering Analysis, J. Mec. Des. 127(6), 1077-1087, 2004.
- [64] Alıntı < [us.codeforge.com/read/210432/svr.m.html](http://us.codeforge.com/read/210432/svr.m.html)> [21 Kasım 2016].
- [65] Artificial neural network, from Wikipedia: The Free Encyclopedia. Alıntı < [https://en.wikipedia.org/wiki/Artificial\\_neural\\_network](https://en.wikipedia.org/wiki/Artificial_neural_network)> [08 Aralık 2016].
- [66] Alıntı<[http://www.chaz.org/Courses/Sapir\\_Whorf/Brain\\_language/Language\\_in\\_the\\_brain.html](http://www.chaz.org/Courses/Sapir_Whorf/Brain_language/Language_in_the_brain.html)> [08 Aralık 2016].
- [67] Alıntı<<http://cdn.iflscience.com/images/8828ec0f-e640-59a8-ae18-9ad858bb353f/large-1464365309-790-scientists-create-artificial-neuron-that-functions-like-the-real-thing.jpg>> [08 Aralık 2016].
- [68] Neuron, from Wikipedia: The Free Encyclopedia. Alıntı < <https://en.wikipedia.org/wiki/Neuron>> [08 Aralık 2016].
- [69] Alıntı<<https://classconnection.s3.amazonaws.com/811/flashcards/141811/jpg/neuron21320853416098.jpg>> [12 Aralık 2016].
- [70] R. Rojas, Neural Networks-a systematic introduction, Alıntı < [page.mi.fu-berlin.de/rojas/neural/index.html.html](http://page.mi.fu-berlin.de/rojas/neural/index.html.html) > [12 Aralık 2016].
- [71] Artificial neuron, from Wikipedia: The Free Encyclopedia. Alıntı < [https://en.wikipedia.org/wiki/Artificial\\_neuron](https://en.wikipedia.org/wiki/Artificial_neuron) > [12 Aralık 2016].
- [72] W. S. McCulloch and W. Pitts, A logical calculus of the ideas immanent in nervous activity”, Bulletin of Mathematical Biophysics, Vol. 5, 1943. Alıntı < [www.minicomplexity.org/pubs/1943-mcculloch-pitts-bmb.pdf](http://www.minicomplexity.org/pubs/1943-mcculloch-pitts-bmb.pdf) > [12 Aralık 2016].
- [73] Alıntı < <http://www.extremetech.com/wp-content/uploads/2015/07/neural-net-head.jpg>> [13 Aralık 2016].

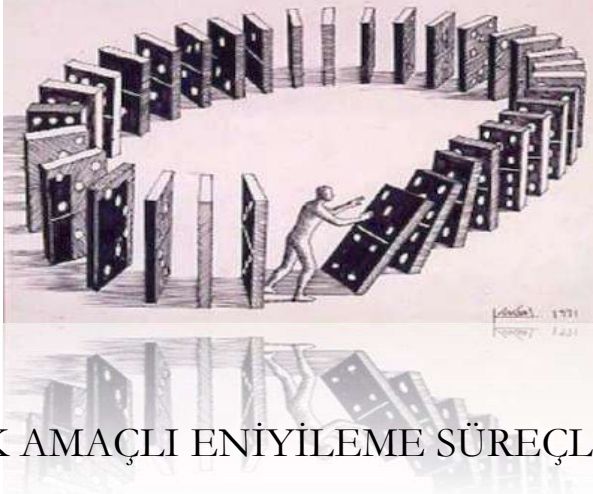


- [74] Nomogram, from Wikipedia: The Free Encyclopedia. Alıntı < <https://en.wikipedia.org/wiki/Nomogram> > [14 Aralık 2016].
- [75] İ. Çayiroğlu, İleri Algoritma Analizi-5: Yapay Sinir Ağları, Alıntı < [www.ibrahimcayiroglu.com/dokumanlar/ilerialgoritmaanalizi-5.hafta-yapaysiniraglari.pdf](http://www.ibrahimcayiroglu.com/dokumanlar/ilerialgoritmaanalizi-5.hafta-yapaysiniraglari.pdf) > [14 Aralık 2016].
- [76] Neural Networks, Alıntı <<https://www.cs.cmu.edu/afs/cs.cmu.edu/academic/class/15381-s06/www/nn.pdf>> [19 Aralık 2016].
- [77] Paul Werbos, from Wikipedia: The Free Encyclopedia. Alıntı < [https://en.wikipedia.org/wiki/Paul\\_Werbos](https://en.wikipedia.org/wiki/Paul_Werbos) > [19 Aralık 2016].
- [78] Neural Networks: Backpropagation Algorithm, Alıntı <[Sydney.edu.au/engineering/it/courses/comp4302/ann4-6s.pdf](http://Sydney.edu.au/engineering/it/courses/comp4302/ann4-6s.pdf)> [19 Aralık 2016].
- [79] Multi-Layer Artificial Neural Networks, Alıntı <[www.doc.ic.ac.uk/~sgc/teaching/pre2012/v231/lecture13.html](http://www.doc.ic.ac.uk/~sgc/teaching/pre2012/v231/lecture13.html)> [19 Aralık 2016].
- [80] A. Spampinato, The materials used in the design of aircraft wings, Alıntı <[http://www.azom.com/images/Article\\_Images/ImageForArticle\\_12117\(1\).jpg](http://www.azom.com/images/Article_Images/ImageForArticle_12117(1).jpg) > [20 Aralık 2016].
- [81] Felsefe, from Wikipedia: The Free Encyclopedia. Alıntı < <https://tr.wikipedia.org/wiki/Felsefe> > [29 Aralık 2016].
- [82] Philosophy, from Wikipedia: The Free Encyclopedia. Alıntı < <https://en.wikipedia.org/wiki/Philosophy> > [29 Aralık 2016].
- [83] Newton's laws of motion, from Wikipedia: The Free Encyclopedia. Alıntı < [https://en.wikipedia.org/wiki/Newton's\\_laws\\_of\\_motion](https://en.wikipedia.org/wiki/Newton's_laws_of_motion) > [29 Aralık 2016].
- [84] Alıntı <[https://c1.staticflickr.com/5/4007/4497377387\\_d7ba144e71\\_b.jpg](https://c1.staticflickr.com/5/4007/4497377387_d7ba144e71_b.jpg)> [29 Aralık 2016].
- [85] Z. Şen, Mühendislikte Felsefe, Mantık, Bilim ve Etik, 2011, Su Vakfı, Sitanbul.
- [86] Logic, from Wikipedia: The Free Encyclopedia. Alıntı < <https://en.wikipedia.org/wiki/Logic> > [30 Aralık 2016].
- [87] Boolean algebra, from Wikipedia: The Free Encyclopedia. Alıntı < [https://en.wikipedia.org/wiki/Boolean\\_algebra](https://en.wikipedia.org/wiki/Boolean_algebra) > [30 Aralık 2016].
- [88] List of logic symbols, from Wikipedia: The Free Encyclopedia. Alıntı < [https://en.wikipedia.org/wiki/List\\_of\\_logic\\_symbols](https://en.wikipedia.org/wiki/List_of_logic_symbols) > [30 Aralık 2016].
- [89] G. Malinowski, Many-Valued Logic, Alıntı < [www.thatmarcusfamily.org/philosophy/Courses\\_websites/Readings/Malinowski%20-%20Many-Valued%20Logic.pdf](http://www.thatmarcusfamily.org/philosophy/Courses_websites/Readings/Malinowski%20-%20Many-Valued%20Logic.pdf) > [03 Ocak 2017].
- [90] M. J. Wierman, An introduction to the mathematics of uncertainty, Alıntı < [https://www.creighton.edu/fileadmin/user/CCAS/programs/fuzzy\\_math/docs/MOU.pds](https://www.creighton.edu/fileadmin/user/CCAS/programs/fuzzy_math/docs/MOU.pds) > [03 Ocak 2017].
- [91] Non-classical logic, from Wikipedia: The Free Encyclopedia. Alıntı < [https://en.wikipedia.org/wiki/Non-classical\\_logic](https://en.wikipedia.org/wiki/Non-classical_logic) > [03 Ocak 2017].
- [92] L. A. Zadeh, Fuzzy Sets, Information and Control 8, 338–353 (1965).
- [93] B. Hamed, Fuzzy Logic Control, Alıntı < [slideshare.com/slide/4858990](http://slideshare.com/slide/4858990) > [03 Ocak 2017].
- [94] Fuzzy Logic, Alıntı < [www.massey.ac.nz/~nhreyes/MASSEY/159741/Lectures/Lec2012-3-159741-FuzzyLogic-v.2.pdf](http://www.massey.ac.nz/~nhreyes/MASSEY/159741/Lectures/Lec2012-3-159741-FuzzyLogic-v.2.pdf) > [04 Ocak 2017].



- [95] A Short Fuzzy Logic Tutorial, Alıntı < [cs.bilkent.edu.tr/~zeynep/files/short\\_fuzzy\\_logic\\_tutorial.pdf](http://cs.bilkent.edu.tr/~zeynep/files/short_fuzzy_logic_tutorial.pdf)> [04 Ocak 2017].
- [96] What is Mamdani-Type Fuzzy Inference, Alıntı < [https://www.mathworks.com/help/fuzzy/what-is-mamdani-type-fuzzy-inference.html#bu\\_nsiq-1](https://www.mathworks.com/help/fuzzy/what-is-mamdani-type-fuzzy-inference.html#bu_nsiq-1) > [04 Ocak 2017].
- [97] B. Kosko, Defuzzification, Alıntı < [www.csee.wvu.edu/classes/cpe521presentations/DEFUZZ.pdf](http://www.csee.wvu.edu/classes/cpe521presentations/DEFUZZ.pdf)> [05 Ocak 2017].
- [98] Alıntı<<http://ovaktechnologies.com/wp-content/uploads/2014/05/Inverted-Pendulum.jpg>> [06 Ocak 2017].
- [99] Alıntı<<http://dsc.utrgv.edu/VirtualLab/PendulumCart/CartPendulumSchem.jpg>> [06 Ocak 2017].
- [100] Alıntı<<http://www.dii.unisi.it/~control/ctm/examples/pend/invpen.html>> [06 Ocak 2017].
- [101] Alıntı<<http://www.flightsimbooks.com/flightsimhandbook/105-1.jpg>> [13 Ocak 2017].
- [102] Alıntı<<http://www.x-plane.com/wp-content/uploads/2014/08/cessna-from-above-in-seatac-landing.png>> [13 Ocak 2017].
- [103] Alıntı<[https://www.aviationsmilitaires.net/media/pictures/9691558409\\_e063925a3c\\_c.jpg](https://www.aviationsmilitaires.net/media/pictures/9691558409_e063925a3c_c.jpg)> [13 Ocak 2017].
- [104] Alıntı<<https://www.cs.ox.ac.uk/philosophyplus/Thinking.jpg>> [13 Ocak 2017].





(1)



5

## ÇOK AMAÇLI ENİYİLEME SÜREÇLERİ

“ Ey yüzü nurlu çocuk! “Rabbimiz, biz nefsimize zulmettik!” demeyi babandan öğren. O, ne bahaneler buldu, ne hileye kalkıştı, ne de tuzak/düzen bayrağını yüceltti. Fakat İblis, bahse girişti, benzin kırmızı, beni sen sararttın. Renk, senin verdiği renktedir, beni boyayan sensin; suçumun da aslı sensin, uğradığım âfetin, dağıldığım dağın da, dedi! Kendine gel de (Hicr suresi 39’uncu ayeti) oku... Oku da **Cebrî** olma, ters bir kumaş dokumaya kalkışma...”

Mevlana (2)

“İblis dedi ki: "Rabbim, beni kıskırttığın şeye karşılık and olsun, ben de yeryüzünde onlara, sana başkaldırmayı ve dünya tutkularını süsleyip çekici göstereceğim ve onların tümünü mutlaka kıskırtıp saptıracağım.”

Hicr Suresi 39 (3)

		Sayfa
5.1	Çok amaçlı eniyilemede tarihsel gelişim ve temel kavramlar	714
5.2	Genel maksat fonksiyonu	721
5.3	Pareto hattı	723
	Bilim Damlası	736
	Kaynakça	738





## 5.1 Çok amaçlı eniyilemede tarihsel gelişim ve temel kavramlar

Ç

ok amaçlı eniyileme<sup>i</sup> kavramının endüstriyel gelişimle beraber gündeme geldiğini, özellikle adil gelir paylaşımı düşüncesi üzerinde oldukça yaygın bir şekilde tartışmaların yaşandığı 1800'lü yılların sonlarında filizlendiğini söylemek mümkündür. 1881 yılında ekonomi profesörü F. Y. Edgeworth (1845-1926) tarafından ifade edilen ve bir manada çok ölçütlü karar verme<sup>ii</sup> sürecine ait optimal nokta tanımlaması niteliğindeki;

- ▶ “Öyle bir  $(x, y)$  noktası bulunmalı ki herhangi bir yönde atılan küçük bir adım sonrasında  $p$  ve  $\pi$ 'nin (tüketici) maliyeti aynı anda artmamalı, birisinin artarken birisinin azalmalı...”

düşüncesi, daha sonra 1896 yılında V. Pareto (1848-1923) tarafından geliştirilerek iktisat alanında ünlü bir yaklaşım haline geldi. Döneminde Pareto eniyilemesi olarak adlandırılan yaklaşımın mühendislik ve diğer doğa bilimlerindeki ilk uygulamalarına ise 1970'li yıllarda rastlanmaktadır<sup>(4)</sup>. Çok amaçlı eniyileme disiplini, çok ölçütlü karar verme bilim dalının bir alt dalı niteliğinde olup temelde birden fazla amacın güdüldüğü eniyileme süreçlerini kapsar<sup>(5)</sup>. Çok ölçütlü karar verme süreci, karar verme sürecinde birden fazla ölçütün söz konusu olduğu sorunlarda karar vericinin daha iyi karar vermesine yardımcı olmayı hedefleyen analitik yöntemleri konu edinir. Genellikle iki tür sorun türünün söz konusu olduğu karar verme problemlerinin ilki çok ölçütlü değerlendirme sorunları, diğeri çok ölçütlü tasarım problemleridir. Çok ölçütlü değerlendirmeden kastedilen, az veya çok ama sınırlı sayıdaki seçenekler arasında en uygun olanının seçimidir. Tabiatıyla seçimde dikkate alınan seçeneklerin önceden bilindiği düşünülebilir. Çok ölçütlü tasarım sorunu ise sürekli değişkenleri barındırması nedeniyle genellikle sonsuz sayıda seçenek içerebilir. Ayrıca olası tasarım seçeneklerinin önceden kestirilmesi de pek mümkün değildir<sup>(6)</sup>. Tasarım sürecini kendi içerisinde, bir veya birden fazla disiplin dâhilinde tasarım süreci olarak iki ayrı gruba ayırmak mümkündür. İlki genellikle çok ölçütlü eniyileme süreci olarak adlandırılabilir. İkincisi, çok disiplinli tasarım optimizasyon<sup>iii</sup> sorunu olarak isimlendirilir. Esasen her iki tasarım grubunda da birden fazla amacın söz konusu olduğu sorunlarla uğraşılır. Ancak tahmin edilebileceği üzere ayrı bilim disiplinlerindeki birden fazla hedefle ilgilenmek aynı disiplindeki çoklu amaçların gerçekleştirilmesinden daha zordur.



Şekil 5.1 Aerodinamik açıdan mükemmel yapıdaki eliptik kanat tasarımı<sup>(7)</sup>.

<sup>i</sup> Multi-objective optimization, multi-objective programming, Pareto optimization

<sup>ii</sup> Multi-criteria decision making

<sup>iii</sup> Multi-disciplinary design optimization



Örneğin uçak kanadı tasarımında aerodinamik biliminin temel ilkeleri dikkate alınarak iki ayrı amaç tanımlanabilir ve belirlenen hedefler çerçevesinde kanat tasarımı şekillendirilebilir;

$$\begin{aligned} \min_x C_L \\ \min_x C_D \end{aligned}$$

Burada  $C_L$  taşıma katsayısını ve  $C_D$  sürüklenme katsayısını ifade etmektedir. Sadece aerodinamik disiplininin dikkate alındığı bir kanat tasarımında olası mükemmel kanat tasarımı Şekil 5.1'deki gibi sonuçlanabilir. Diğer taraftan günümüz uçak tasarımında sadece aerodinamik kaygılar değil, aynı zamanda elektromanyetik kaygılar da büyük önem arz etmektedir. Elektromanyetik kaygılardan kastedilen, radara yakalanmama özelliği olup söz konusu disiplinin dikkate alındığı bir uçak tasarımında amaç fonksiyonu;

$$\min_x \sigma$$

şeklinde ifade edilebilir. Burada  $\sigma$ , belirli bir açı değerindeki radar kesit alanı<sup>i</sup> değerini göstermektedir. Elektromanyetik kaygıların dikkate alındığı bir tasarım sürecinin olası sonucu Şekil 5.2'deki gibi olabilir. Buna mukabil her iki disiplinin dikkate alındığı bir uçak tasarımında çoklu amaç fonksiyonu;

$$\begin{aligned} \max_x C_L/C_D \\ \min_x \sigma \end{aligned}$$

denklemleri ile tarif edilebilir. Gerek aerodinamik ve gerekse de elektromanyetik kaygıların ön planda olduğu bir tasarım sürecinin olası uçak şekli Şekil 5.3'teki gibi sonuçlanabilir. Görüldüğü gibi eniyileme sürecinde farklı disiplinlerin dikkate alınması tasarım sonucunu ciddi manada etkilemektedir. Diğer taraftan örneklerden de gözlemlenebileceği üzere ister tek disiplin olsun isterse de birden fazla disiplin olsun, her iki sorunda da çok amaçlı tasarım eniyileme süreçleri söz konusudur. Genel çerçevede çok amaçlı eniyileme sorununu;

$$\min/\max_x f_i(x), \quad i = 1, 2, \dots, I \quad (5.1)$$

Kısıt

$$\begin{aligned} h_j(x) &= 0, & j &= 1, 2, \dots, J \\ g_k(x) &\leq 0, & k &= 1, 2, \dots, K \\ x_p^L &\leq x_p \leq x_p^U, & p &= 1, 2, \dots, P \end{aligned}$$



Şekil 5.2 Elektromanyetik açıdan mükemmel yapıdaki uçak tasarımı<sup>(8)</sup>.



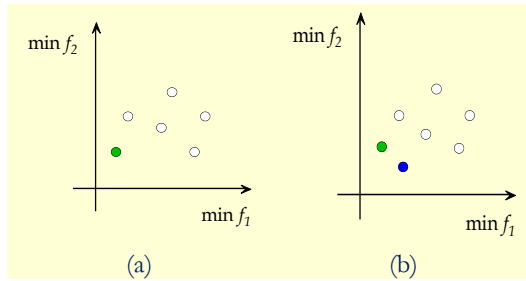
Şekil 5.3 Aerodinamik ve elektromanyetik açıdan mükemmel yapıdaki uçak tasarımı<sup>(9)</sup>.

<sup>i</sup> Radar cross section



denklemleri ile betimlemek mümkündür<sup>(10)</sup>. Burada  $f_i(\mathbf{x})$  birden fazla sayıdaki amaç fonksiyonunu,  $h_j(\mathbf{x})$  ile  $g_k(\mathbf{x})$  sırasıyla eşitlik ve eşitsizlik kısıtlarını temsil etmektedir. Ayrıca sorunda tasarım yöneyinin her ögesine ait alt ve üst sınırlardan bahsedilebilir. Diğer taraftan amaç fonksiyonlarında aynı gayenin güdülmesi zorunlu değildir. Dolayısıyla eniyileme sürecinde bazı amaç fonksiyonlarının minimizasyonu, bazılarının ise maksimizasyonu hedeflemesi mümkündür.

Birden fazla amaç fonksiyonunun dikkate alındığı süreçlerde amaç fonksiyonları arasındaki etkileşim, çözüme ulaşmada önemli bir etkiye sahiptir. Amaç fonksiyonları arasında temelde üç çeşit etkileşim mümkün olup söz konusu etkileşimler; işbirliği, çatışma ve ilgisizlik şeklindedir. İşbirliğinden kastedilen, amaç fonksiyonlarının tasarım yöneyi değişimlerine verdiği tepkinin benzer olmasıdır. Örneğin tasarım yöneyi azaldıkça amaç fonksiyonları beraberce azalmakta ise aralarında işbirliği niteliğindeki bir etkileşimden bahsedilebilir. Buna mukabil tasarım yöneyi değişimlerine verilen tepkinin farklı olması, işbirliğinden ziyade bir çatışmanın göstergesidir. Örneğin tasarım yöneyindeki azalmaya karşılık amaç fonksiyonlarından birinin azalması, diğerinin artması durumunda amaç fonksiyonları arasında bir rekabetin varlığından bahsedilebilir. Öte yandan etkileşimin, tasarım uzayının farklı bölgelerinde farklı özellikler göstermesi mümkündür. Örneğin her iki farklı amaç fonksiyonu tasarım yöneyinin belirli değerlerinde azalma, belirli değerlerinde artma göstererek işbirliği içinde olabilir. Ancak belirli bir bölgede çatışma olabilir. Son etkileşim türü olan ilgisizlik, esasen amaç fonksiyonları arasında etkileşimsizliğin olduğunu, birbirinden bağımsız bir ilişkinin varlığını gösterir. Genellikle ortak tasarım yöneyi ögesine sahip olmayan amaç fonksiyonları arasında bir etkileşimden bahsedilemez. Tasarım uzayı boyunca işbirliğinin söz konusu olduğu sorunlarda genellikle tekil ve yalın çözümlerden bahsedilebilir. Örneğin Şekil 5.4(a)'daki gibi çözüm noktalarının bulunduğu bir sorunda optimal noktanın hangisi olduğu kolayca belirlenebilir;



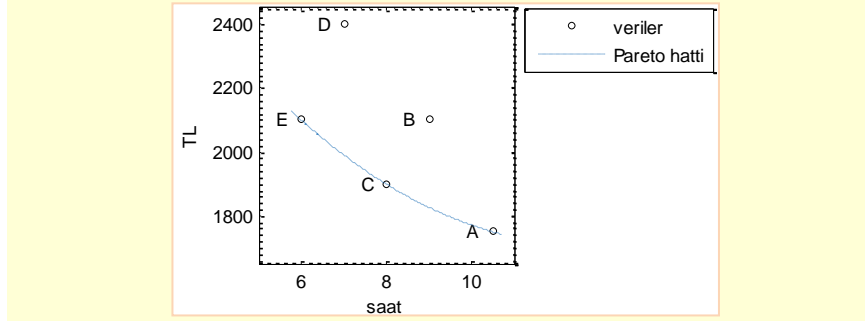
Şekil 5.4 İki amaçlı bir sorunda olası çözüm kümesi noktalarının dağılımları.



Buna göre her iki amaç fonksiyon değerinin de minimum olduğu nokta (yeşil), mevcut çözüm noktaları arasındaki eniyi noktadır. Ancak Şekil 5.4(b)'deki gibi bir çözüm uzayı kümesi için aynı mülahazayı yapmak mümkün değildir. Zira şekilden de görüldüğü üzere burada olası optimal nokta sayısı birden fazladır, (yeşil ve mavi noktalar). Yeşil nokta,  $f_1$  amaç fonksiyonu değeri açısından daha iyi bir çözüm sunarken; mavi nokta,  $f_2$  amaç fonksiyonu açısından daha minimal bir çözümle sonuçlanmaktadır. Hangisinin en iyi nokta olduğunu belirlemek ise nesnel olmaktan ziyade öznel bir süreçtir. Daha açık bir ifade ile çözüm noktasının seçimi, tasarımcının tercihinine göre şekillenir.

Diğer taraftan elde edilen çözüm kümesi noktaları arasındaki karşılaştırmada nesnel ölçütlerden bahsedilebilir. Baskınlık özelliği olarak adlandırılan tanımlama çerçevesinde bir çözüm noktasının diğer başka bir çözüm noktasına göre daha iyi olup olmadığı göreceli olarak belirlenebilir. Buna göre herhangi bir  $x_i$  noktasının  $x_j$  noktasına göre daha baskın olarak tanımlanabilmesi için müteakip şartların sağlanması gerekir; öncelikle  $x_i$  noktası  $x_j$  noktasına göre tüm amaç fonksiyon değerleri açısından daha kötü olmamalı ve ayrıca  $x_i$  noktası  $x_j$  noktasına göre amaç fonksiyon değerlerinden en az birisine göre belirli bir şekilde daha iyi olmalı. Çözüm noktaları arasındaki baskınlık kavramının daha iyi anlaşılması için müteakip örnek üzerinde durulabilir. Örneğin uzak bir kente uçak yolculuğu yapacağımızı, kısa bir araştırma neticesinde fiyatı ve seyahat süresi değişen farklı biletleri belirlediğimizi düşünelim;

Bilet	A	B	C	D	E
$f_1$ : Seyahat süresi (saat)	10.5	9	8	7	6
$f_2$ : Bilet fiyatı (TL)	1750	2100	1900	2400	2100



Şekil 5.5 İki amaçlı örnek problemde Pareto hattı.

<sup>1</sup>Dominance



Eldeki olası biletlerin uçuş süresi ve maliyet değerlerine göre grafiksel gösterimi Şekil 5.5'te görülmektedir. Buna göre E ve B biletleri arasında yapılan bir karşılaştırmada, birinin diğerine göre üstün olduğunu nesnel bir çerçevede söyleyebiliriz. Zira  $f_2$  amaç fonksiyonu açısından E bileti B biletinden daha kötü değildir ( $f_2^E = f_2^B$ ), ilave olarak aynı çözüm noktalarının  $f_1$  amaç fonksiyonu açısından değerlendirilmesi neticesinde E biletinin B biletine göre belirgin bir şekilde daha düşük bir maliyete sahip olduğu aşikârdır. Dolayısıyla E noktası çözümünün B noktasına göre baskın bir çözüm olduğu rahatlıkla söylenebilir. Buna mukabil B ve C biletlerinin karşılaştırılması neticesinde kesinlikle C biletinin süre ve maliyet açısından avantajlı olduğunu gözlemleyebiliriz. Ayrıca C çözümünün B çözümüne göre baskın olduğu kolayca söylenebilir. Benzer mütalaa D ve E biletleri için de geçerlidir. Verilere göre E bileti D biletine göre süre ve fiyat açısından daha baskındır. Bununla beraber A, C ve E biletleri arasında herhangi bir baskınlık karakterine sahip çözüm noktası yoktur. Bazı biletlerde fiyat, bazılarında süre avantajı söz konusudur. Örneğin C ve A noktaları arasında  $f_1^{(C)} < f_1^{(A)}$  olmasına karşılık  $f_2^{(C)} > f_2^{(A)}$  eşitsizliği baskınlık karakterini zedelemektedir. Benzer durum E ve C noktaları için de geçerlidir. Baskınlık açısından belirsizlik içeren bu tür noktalara baskılanamayan<sup>i</sup> noktalar adı verilmektedir. Literatürde bahsi geçen baskılanamayan noktalar kümesine genellikle Pareto optimal kümesi<sup>ii</sup> ve bu kümeye ait noktaların bulunduğu hatta da Pareto hattı<sup>iii</sup> adı verilmektedir. Dolayısıyla örnekteki bahse konu bu üç nokta, Pareto hattını oluşturmaktadır<sup>(1)</sup>.

Gerek Pareto optimal nokta veya noktaların belirlenmesi ve gerekse de belirlenen olası çözüm noktaları arasından seçim yapılması sadedinde değişik yaklaşımlardan bahsedilebilir. Bununla beraber hat üzerindeki noktaların belirlenmesi daha çok matematiksel temellere, çözümde tercih yapılması ise konusunda uzman karar vericilere<sup>iv</sup> dayandırılır. Çok amaçlı eniyileme yöntemlerini, öncelikle karar vericiye dayalı veya tamamen bağımsız yöntemler<sup>v</sup> olarak iki ayrı gruba tasnif etmek mümkündür. Bağımsız yöntemlerde çözüm noktasının seçimi, uzman görüşü olmadan tanımlanmaya çalışılır. Karar vericiye dayalı yöntemler üç grupta toparlanabilir. Bunlardan ilki, önsel yöntemler<sup>vi</sup> grubu olup sorunun çözüm süreci öncesinde konu hakkında uzman bir karar verici tarafından çözüme yönelik tercihler<sup>vii</sup> belirlenir. Akabinde, belirlenen öncelikleri en uygun

---

<sup>i</sup> Non-dominated points

<sup>ii</sup> The set of Pareto optimal outcomes

<sup>iii</sup> Pareto front

<sup>iv</sup> A human decision maker (DM)

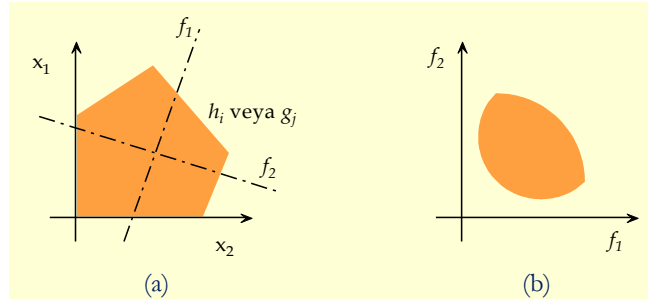
<sup>v</sup> No preference methods

<sup>vi</sup> A priori methods

<sup>vii</sup> Preferences



şekilde karşılayan çözüm noktası aranır. İkinci yöntem grubu, sonrasali (sonradan gelen) yöntemler olup önsel yöntemlerin aksine uzman karar vericinin tercihlerine çözüm süreci sonrasında başvurulur. Daha açık bir ifade ile önce Pareto optimal çözüm noktalarının tespiti yapılır ve sonrasında da elde edilen olası çözüm noktaları arasından uzman kişinin seçimi dikkate alınır. Son grup, etkileşimli yöntemler<sup>ii</sup> topluluğudur. Bu yaklaşımda uzman görüşü, çözüm sürecinin her adımında başvurulmuş bir kaynaktır. Elde edilen her çözüm, karar verici tarafından değerlendirilerek yeni ve belki de daha iyi bir çözüme ulaşılmaya çalışılır<sup>(5)</sup>.



Şekil 5.6 İki boyutta değişken (a) ve gaye (b) uzayı örnekleri.

Tabiatıyla karar verme sürecini etkileyen önemli bir diğer husus da çözümün veya çözüm noktalarının görselleştirilmesidir. Zira karar aşamasında soruna yönelik çözümün diğer çözümlere göre konumu, kıyas açısından önem arz eder. Bu meyanda iki olası görsel ortamdan bahsedilebilir. Bunlardan ilki değişken uzayı, ikincisi gaye uzayıdır<sup>iii</sup>. Değişken uzayı, çözüm sürecinde dikkate alınan tasarım yöneyi olası çözüm bölgesini ve bu bölgeyi çevreleyen eşitlik ve/veya eşitsizlik kısıtlarını içerir. Gaye uzayı, tasarım yöneyi değerine göre belirlenen amaç fonksiyonlarının grafiğe dökülmüş halidir. Şekil 5.6 dâhilinde iki boyutlu tasarım yöneyi ve iki amaç fonksiyonu içeren değişken uzayı ile iki farklı amaç fonksiyonu barındıran gaye uzayı örnekleri görülmektedir. Genellikle tercih aşamasında gaye uzayı daha açıklayıcı görselliğe sahiptir. Ancak amaç fonksiyon sayısının ikiden fazla olması halinde görselliği sağlamak mümkün olmayabilir. Bu durumda ikili çiftlerden oluşan amaç fonksiyonu değişimleri dikkate alınabilir.

<sup>i</sup> A posteriori methods

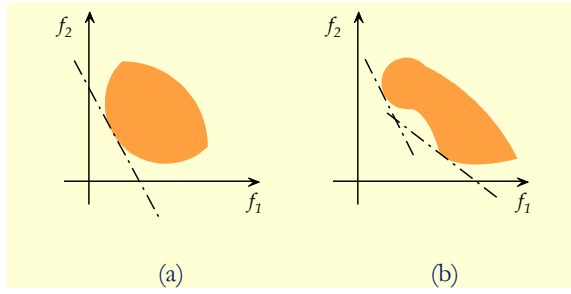
<sup>ii</sup> Interactive methods

<sup>iii</sup> Variable space and criterion space



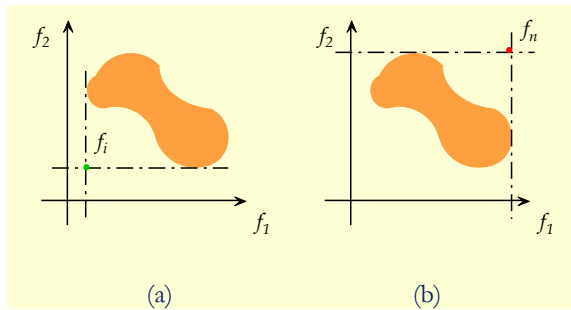


Gaye uzayındaki olası çözüm bölgesinin niteliği, çözüme ulaşmada önemli bir rol oynar. Tahmin edileceği üzere dışbükey çözüm alanları, çözüme ulaşmada büyük kolaylık sağlar. Ancak dışbükey olmayan çözüm havzaları da uygulamalarda sıkça rastlanan bir durum olup bu tür sorunların çözümü daha karmaşık yaklaşımların sergilenmesini gerektirir. Tabiatıyla çözüm havzasının niteliği, aynı zamanda Pareto hattı veya eğrisinin dışbükey veya dışbükey olmayan karakteri anlamına da gelmektedir. Müteakip grafiklerde dışbükey ve dışbükey olmayan gaye uzayları görülmektedir;



Şekil 5.7 İki boyutta dışbükey (a) ve dışbükey olmayan (b) gaye uzayı örnekleri.

Amaç fonksiyonlarının ölçeksel olarak birbirinden oldukça farklı olduğu sorunlarda başvurulan yaygın bir yöntem, normalizasyondur. Normalizasyon tanımlamasında kullanılan limit değerler, her bir fonksiyona ait azami ve asgari değerlerdir.



Şekil 5.8 İki boyutlu gaye uzayında (a) ideal ( $f_i$ ) ve (b) beter nokta ( $f_n$ ) tanımlamaları.





Her bir amaç fonksiyonunun azami değerlerini temsil eden noktaya ideal nokta<sup>i</sup>, buna karşılık en kötü değerlerini temsil eden noktaya betor nokta<sup>ii</sup> adı verilmektedir. Normalizasyon haricinde ideal ve betor nokta tanımlamalarının diğer bir getirisi de karar verici açısından olası çözüme ait alt ve üst limit değerleri hakkında bir fikir vermesidir. Şekil 5.8 dâhilinde gaye uzayındaki ideal ve betor nokta görsel tanımlamaları görülmektedir. Normalizasyonda dikkate alınabilecek örnek bir denklemi;

$$f_i^{norm} = \frac{f_i - f_{i,i}}{f_{i,n} - f_{i,i}} \quad (5.2)$$

şeklinde tanımlayabiliriz Burada  $f_{i,i}$   $i$ 'ncinci amaç fonksiyonuna ait ideal nokta değerini,  $f_{i,n}$  aynı amaç fonksiyonuna ait betor nokta değerini temsil etmektedir.

## 5.2 Genel maksat fonksiyonu

Çok sayıdaki amaç fonksiyonunu dikkate alan çözüm sürecinde ister önsel, ister sonrasal, isterse de etkileşimli yöntem tercih edilsin, her üç yöntemde de yapılması gereken ilk işlemlerden birisi, amaç fonksiyonlarını tek çatı altında toparlayan bir genel maksat fonksiyonu tanımlamaktır. Buna göre genel maksat fonksiyonu ve kısıtlarla beraber çok amaçlı optimizasyon problemi;

$$\begin{aligned} & \min_{\mathbf{x}} U[f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_I(\mathbf{x})] & (5.3) \\ \text{Kısıt} & \quad \quad \quad \begin{aligned} h_j(\mathbf{x}) &= 0, & j &= 1, 2, \dots, J \\ g_k(\mathbf{x}) &\leq 0, & k &= 1, 2, \dots, K \\ \mathbf{x}_p^L &\leq \mathbf{x}_p \leq \mathbf{x}_p^U, & p &= 1, 2, \dots, P \end{aligned} \end{aligned}$$

şeklindeki tek amaçlı eniyileme sorunu olarak yeniden tanımlanabilir. Doğal olarak genel maksat fonksiyonunun içinde bulunan doruklaştırma veya minimizasyon gibi farklı amaçların, uygun yaklaşımlarla aynı amaca dönüştürülmesi gerekir. Literatürde değişik genel maksat fonksiyonu tanımlamaları mevcuttur. Ancak genellikle bu yöntemlerde başvuru olan temel yaklaşım, amaç fonksiyonlarına değişik önem atfeden ağırlık katsayıları bazlı doğrusal veya doğrusal olmayan toparlama fonksiyonu tanımlamasıdır.

<sup>i</sup> Ideal or utopia point

<sup>ii</sup> Nadir point





En yaygın amaç fonksiyonu tanımlaması, ağırlıklandırılmış toplama yöntemi<sup>i</sup> olup genel maksat fonksiyonu müteakip denklemdeki gibi betimlenebilir;

$$\begin{aligned} \min_x \sum_{i=1}^l w_i f_i(x) & \quad (5.4) \\ \sum_{i=1}^l w_i &= 1 \\ w_i &> 0 \end{aligned}$$

burada  $w_i$  ağırlık katsayısı olarak adlandırılır ve genellikle kullanıcı tarafından belirlenir. Diğer bir yöntem, ağırlıklandırılmış üssel toparlama yaklaşımıdır<sup>ii</sup>. Yöntemin bir öncekinden farkı, amaç fonksiyonlarının üssel değerlerinin dikkate alınmasıdır. Bu meyanda genel maksat fonksiyonu;

$$\begin{aligned} \min_x \sum_{i=1}^l w_i f_i^p(x) & \quad (5.5) \\ \sum_{i=1}^l w_i &= 1 \\ w_i, p &> 0 \end{aligned}$$

şeklinde belirlenebilir. Burada ağırlık katsayısı parametresine ilave olarak  $p$  parametresi yine kullanıcı tarafından tanımlanır. Doğrusal olmayan karakterin amaç fonksiyonu yanında ağırlık katsayısında da dikkate alınması ile üçüncü bir yaklaşım daha tanımlamak mümkündür. Literatürde Tchebycheff yöntemi olarak geçen yaklaşımda genel maksat fonksiyonu;

$$\begin{aligned} \min_x \left[ \sum_{i=1}^l w_i^p (f_i(x) - f_{i,i})^p \right]^{1/p} & \quad (5.6) \\ \sum_{i=1}^l w_i &= 1 \\ w_i, p &> 0 \end{aligned}$$

denklemleri ile tanımlanır. Denklemden geçen  $f_{i,i}$  değeri,  $i$ 'nci amaç fonksiyona ait ideal nokta değeridir<sup>(12)</sup>.

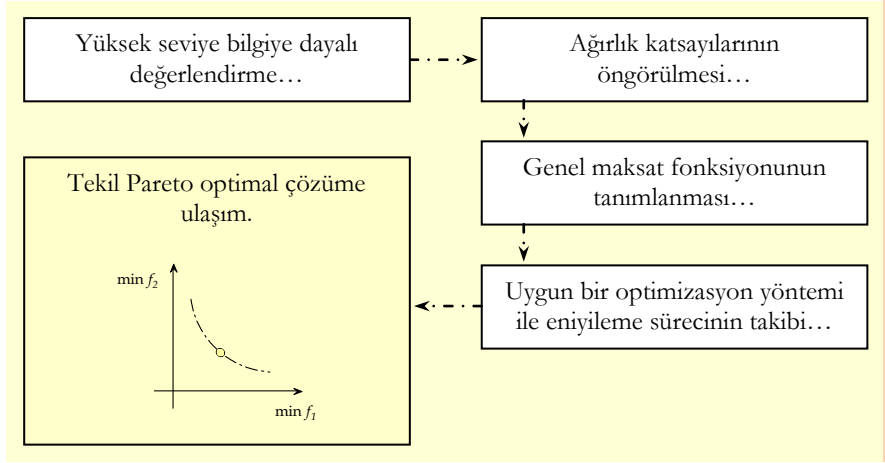
<sup>i</sup> Weighted sum method

<sup>ii</sup> Weighted exponential method



### 5.3 Pareto hattı

Genel maksat fonksiyonunun tanımlanması sonrasında çözüm yöntemi seçimine gidilir. Daha önce bahsedilen önsel yöntemlerde müteakip akış diyagramı izlenebilir;



Şekil 5.9 Önsel yöntem dâhilinde çok maksatlı eniyileme akış diyagramı.

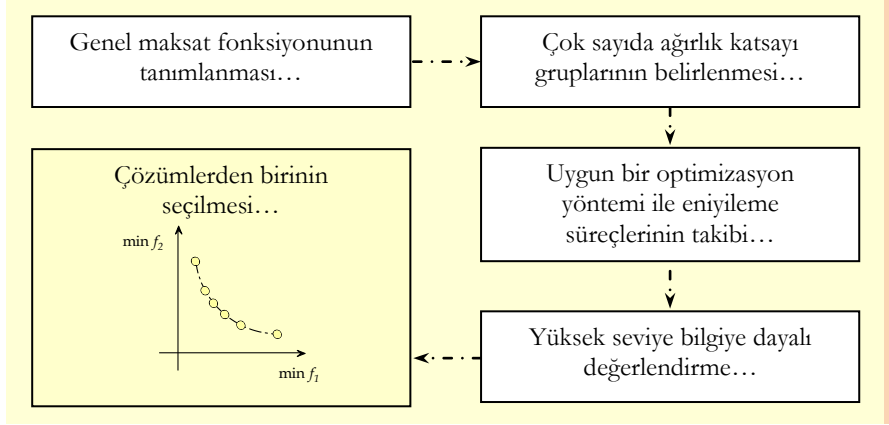
Diyagramdan da görüldüğü üzere genel maksat fonksiyonunun tanımlanmasında ihtiyaç duyulan ağırlık katsayıları, uzman bir karar vericinin tercihleri dâhilinde öngörülebilir. Öngörü sonrasında eldeki minimizasyon bazlı optimizasyon sorunu, bilinen eniyileme yöntemlerinden biri kullanılarak çözülebilir. Nihayetinde tekil bir çözüme ulaşılır ve söz konusu çözüm Pareto optimal çözüm noktalarından birisidir.

Buna karşılık sonrasal yöntemlerde, müteakip akış diyagramı (Şekil 5.10) dikkate alınır. Çözüm sürecinin başlangıcında, genel maksat fonksiyonunun tanımlanması gerekir. Ancak fonksiyona ait katsayıların önsel yöntemlerde olduğu gibi önceden öngörülmesi yerine, çok sayıda katsayı setleri dikkate alınır. Tabiatıyla her bir katsayı seti, farklı bir genel maksat fonksiyonu anlamına gelir. Eldeki her bir genel maksat fonksiyonu baz alınarak çok sayıda eniyileme sorunu tanımlanmış olur. Tanımlanan söz konusu optimizasyon sorunları, bilinen eniyileme yöntemlerinden biri kullanılarak çözülebilir. Nihayetinde tekil bir çözüm yerine, çok sayıda çözüm noktalarına ulaşılır ve söz konusu Pareto optimal çözüm noktalarından birisi, uzman



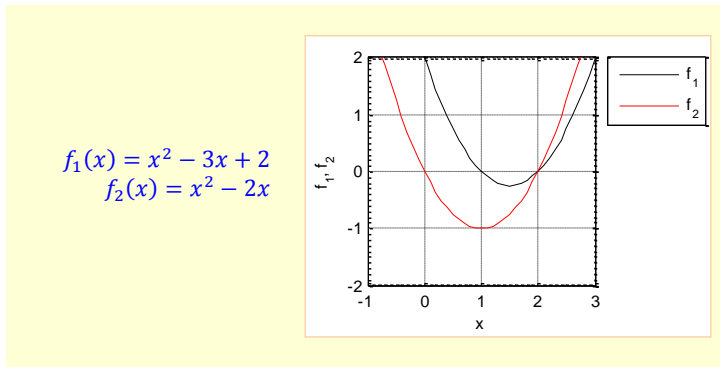


görüşü çerçevesinde seçilir. Çok amaçlı eniyileme süreçlerinde esasen yalnız bir optimal noktanın bulunmasından ziyade sonrasal yöntemlerde olduğu gibi çok sayıda çözüm noktası barındıran Pareto optimal noktalar kümesinin tespit edilmesi ve bu sayede çözüm çeşitliliği ile beraber uygun seçeneklerin belirlenmesi hedeflenir.



Şekil 5.10 Sonrasal yöntem dâhilinde çok maksatlı eniyileme akış diyagramı.

Önsel yöntem kullanılarak optimal çözüme ulaşılması sürecinin daha iyi anlaşılması için iki farklı amaç fonksiyonunu dikkate alan bir çok amaçlı eniyileme sorununu, ağırlıklandırılmış toplama yöntemi ile çözebiliriz. Buna göre örnek amaç fonksiyonlarını müteakip şekilde tanımlayalım;

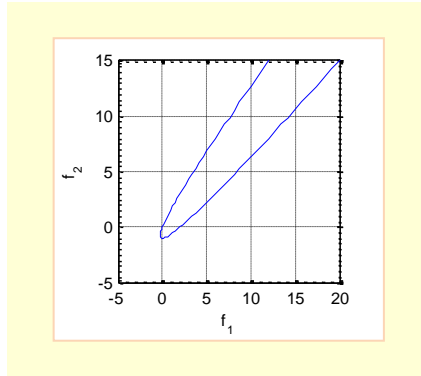


Şekil 5.11 İki farklı amaç fonksiyonu.





Şekilden de görüldüğü üzere tanımlanan amaç fonksiyonlarının  $x$  tasarım değerine göre değişimi, bölgesel olarak farklı etkileşimler sergilemektedir. Örneğin  $-\infty < x \leq 1$  aralığında amaç fonksiyonlarının her ikisi de azalmakta, dolayısıyla işbirliği etkileşimi gözlenmektedir. Ancak  $1 < x \leq 1.5$  arasında  $f_1(x)$  fonksiyonu azalmaya devam ederken,  $f_2(x)$  fonksiyonu artmakta, bu nedenle söz konusu aralıkta amaç fonksiyonları çatışmaktadır. Öte yandan tasarım yöneyi değişimine göre  $f_1(x)$  ve  $f_2(x)$  amaç fonksiyonlarının alacağı değerleri, müteakip gaye uzayında gözlemek mümkündür;



Şekil 5.12 Gaye uzayında amaç fonksiyonlarının değişimi ( $-5 \leq x \leq 5$ ).

Tanımlı amaç fonksiyonlarının dikkate alındığı çok amaçlı optimizasyon problemini, ağırlıklandırılmış toplama yaklaşımına göre;

$$\min_x f(x)$$

$$f(x) = \sum_{i=1}^2 w_i f_i(x) = w_1[x^2 - 3x + 2](x) + w_2[x^2 - 2x]$$

$$\sum_{i=1}^2 w_i = 1$$

$$w_i > 0$$

şeklinde tanımlamak mümkündür. İfade edilen genel maksat fonksiyonundaki ağırlık katsayılarının, uzman görüşü dâhilinde  $w_1 = 0.5, w_2 = 0.5$  olarak belirlendiğini düşünelim. Dolayısıyla aynı soruna ait genel maksat fonksiyonu;



$$f(x) = \frac{1}{2}[x^2 - 3x + 2](x) + \frac{1}{2}[x^2 - 2x] = x^2 - \frac{5}{2}x + 1$$

denklemine ve eniyileme sorunu;

$$\min_x f(x) = x^2 - \frac{5}{2}x + 1$$

ifadesine indirgenebilir. Belirlenen bu sorunun çözümü oldukça kolay olup analitik bir yaklaşımla optimal tasarım yöneyi;

$$\frac{\partial f(x)}{\partial x} = 2x - \frac{5}{2} = 0$$

$$x^* = 1.25$$

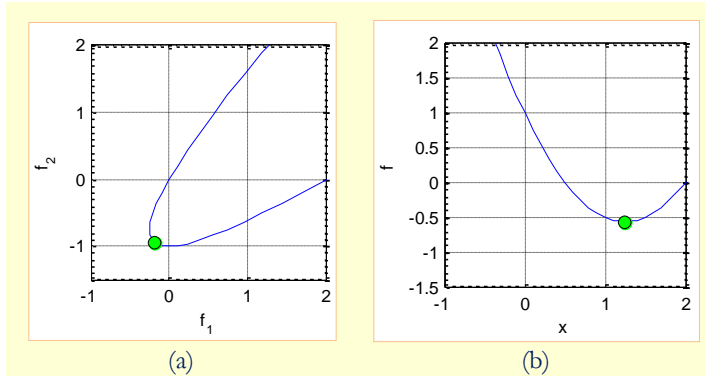
olarak elde edilebilir. Optimal noktadaki amaç fonksiyon değerleri ile genel maksat fonksiyon değeri;

$$f_1(x^*) = f_1(1.25) = -0.1875$$

$$f_2(x^*) = f_2(1.25) = -0.9375$$

$$f(x^*) = f(1.25) = -0.5625$$

biçiminde belirlenebilir. Önsel bir yaklaşımla öngörülen ağırlık katsayılarına göre ulaşılmış optimal noktanın gaye uzayındaki konumu ve genel maksat fonksiyonunun tasarım yöneyine göre değişimi, müteakip grafiklerde görülmektedir;



**Şekil 5.13**(a) Gaye uzayında amaç fonksiyonlarının değişimi ve optimal nokta, (b) Genel maksat fonksiyonunun tasarım yöneyine göre değişimi ve optimal nokta;  $x^* = 1.25$ .



Tanımlanan çok maksatlı eniyileme sorununun sonrasal yaklaşımla da çözümü mümkündür. Bu bağlamda sorunun tanımı açısından yine ağırlıklandırılmış toplama yöntemi dikkate alınabilir ve aynı eniyileme sorunu dikkate alınabilir;

$$\begin{aligned} \min_x f(x) \\ f(x) = \sum_{i=1}^2 w_i f_i(x) = w_1[x^2 - 3x + 2](x) + w_2[x^2 - 2x] \\ \sum_{i=1}^2 w_i = 1 \\ w_i > 0 \end{aligned}$$

Ancak bu yaklaşımda ağırlık katsayılarının önceden belirlenmesi yerine, katsayılar seti dikkate alınarak çok sayıda eniyileme problemi tanımlanır. Örneğin bu sorun için müteakip ağırlık katsayı seti dikkate alınabilir;

$$W = \begin{bmatrix} 0 & 0.25 & 0.5 & 0.75 & 1 \\ 1 & 0.75 & 0.5 & 0.25 & 0 \end{bmatrix}$$

Dikkate alınan set değerlerine göre genel maksat fonksiyonları ve eniyileme sorunları;

$$\begin{aligned} \min f^{(1)}(x) &= 0 \times [x^2 - 3x + 2](x) + 1 \times [x^2 - 2x] = x^2 - 2x \\ \min f^{(2)}(x) &= 0.25 \times [x^2 - 3x + 2](x) + 0.75 \times [x^2 - 2x] = x^2 - \frac{9}{4}x + \frac{1}{2} \\ \min f^{(3)}(x) &= 0.50 \times [x^2 - 3x + 2](x) + 0.50 \times [x^2 - 2x] = x^2 - \frac{5}{2}x + 1 \\ \min f^{(4)}(x) &= 0.75 \times [x^2 - 3x + 2](x) + 0.25 \times [x^2 - 2x] = x^2 - \frac{11}{4}x + \frac{3}{2} \\ \min f^{(5)}(x) &= 1 \times [x^2 - 3x + 2](x) + 0 \times [x^2 - 2x] = x^2 - 3x + 2 \end{aligned}$$

olarak belirlenebilir. Her bir eniyileme sorunun uygun bir eniyileme algoritması ile ve mesela analitik yaklaşımla çözülmesi neticesinde olası optimal nokta değerleri;

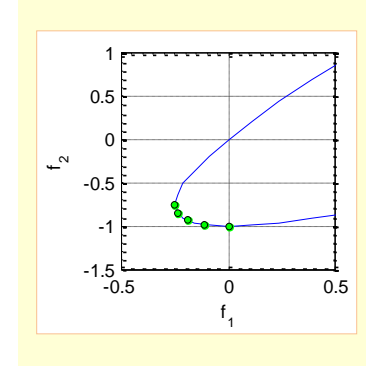
$$\begin{aligned} \frac{\partial f^{(1)}(x)}{\partial x} = 2x - 2 = 0 & \quad x^{(1),*} = 1 \\ \frac{\partial f^{(2)}(x)}{\partial x} = 2x - \frac{9}{4} = 0 & \quad x^{(2),*} = 1.125 \\ \frac{\partial f^{(3)}(x)}{\partial x} = 2x - \frac{5}{2} = 0 & \quad x^{(3),*} = 1.25 \\ \frac{\partial f^{(4)}(x)}{\partial x} = 2x - \frac{11}{4} = 0 & \quad x^{(4),*} = 1.375 \\ \frac{\partial f^{(5)}(x)}{\partial x} = 2x - 3 = 0 & \quad x^{(5),*} = 1.5 \end{aligned}$$





şeklinde elde edilebilir. Olası optimal noktaların amaç fonksiyon değerleri ve gaye uzayındaki konumları, müteakip satırlar ve grafikte görülmektedir;

$f_1(x^{(1),*}) = f_1(1) = 0$	$f_2(x^{(1),*}) = f_2(1) = -1$	$f(x^{(1),*}) = -1$
$f_1(x^{(2),*}) = f_1(1.125) = -0.1094$	$f_2(x^{(2),*}) = f_2(1.125) = -0.9844$	$f(x^{(2),*}) = -0.7657$
$f_1(x^{(3),*}) = f_1(1.25) = -0.1875$	$f_2(x^{(3),*}) = f_2(1.25) = -0.9375$	$f(x^{(3),*}) = -0.5625$
$f_1(x^{(4),*}) = f_1(1.375) = -0.2344$	$f_2(x^{(4),*}) = f_2(1.375) = -0.8594$	$f(x^{(4),*}) = -0.3906$
$f_1(x^{(5),*}) = f_1(1.5) = -0.25$	$f_2(x^{(5),*}) = f_2(1.5) = -0.75$	$f(x^{(5),*}) = -0.25$



Grafikten de anlaşılacağı üzere az sayıda olası çözüm noktası olmasına rağmen soruna ait Pareto hattı belli oranda gözlemlenebilmektedir. Elde edilen çözümler arasından hangisinin seçileceği hususu, uzman görüşüne veya diğer bir ifade ile karar vericiye bağlıdır. Mevcut verilere göre genel maksat fonksiyon değeri en düşük olan  $x^{(2),*} = 1.125, f(x^{(2),*}) = -0.7657$  noktasının çok amaçlı optimizasyon sorununun çözümü olarak dikkate alınması tercih edilebilir. Böyle bir tercihte, ilk amaç fonksiyonuna %25 oranında, diğer amaç fonksiyonuna ise %75 oranında önem atfedilmektedir.

Diğer taraftan aynı soruna ait Pareto optimal noktalar kümesi, sorunun basit olması vesilesi ile, analitik olarak da elde edilebilir. Şöyle ki; genel maksat fonksiyonu, ağırlık katsayıları dikkate alınarak tekrar yazılır ve katsayılar arasındaki toplamsal ilişki değerlendirilerek iki ağırlık katsayısı yerine sadece bir ağırlık katsayısı dikkate alınırsa;

$$\sum_{i=1}^2 w_i = 1$$

$$w_1 + w_2 = 1 \rightarrow w_2 = 1 - w_1$$

$$w_1 \rightarrow w$$

$$f(x) = w[x^2 - 3x + 2](x) + (1 - w)[x^2 - 2x]$$

denkleme ulaşılır. Elde edilen denklemin tasarım yöneyine göre optimal nokta değeri, analitik yolla belirlenirse;

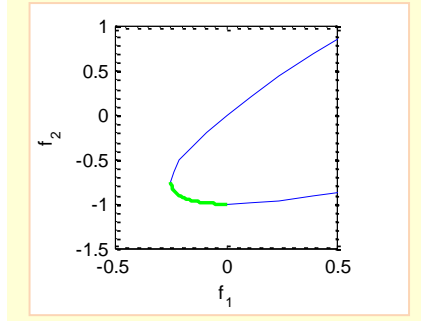




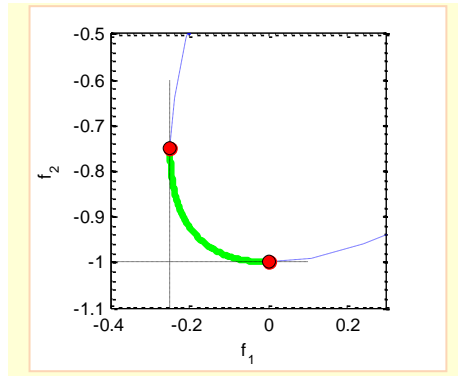
$$\frac{\partial f(x)}{\partial x} = 2x - 2 - w = 0$$

$$x^* = \frac{2 + w}{2}$$

sonucuna ulaşılır. Görüldüğü üzere en uygun nokta, doğrudan doğruya ağırlık katsayı değerine bağlıdır. Ağırlık katsayısı değerlerinin belli bir sistematik dahilinde değiştirilmesi halinde genel maksat fonksiyonuna ait olası optimal noktalar ve dolayısıyla Pareto hattı ile bu noktalara bağlı olarak amaç fonksiyon değerleri kolayca belirlenebilir. Örneğin  $w = 0.25$  değerine karşılık optimal nokta olarak  $x^* = (2 + 0.25)/2 = 1.125$  değerine ulaşılır ki bu değer önceki çözümle örtüşmektedir. Müteakip gaye uzayında, ağırlık katsayısı değerinin  $[0,1]$  aralığında ve eşit adımlarla değiştirilmesi ile elde edilen Pareto hattı ve diğer olası çözüm noktaları görülmektedir.



Şekil 5.14 Gaye uzayında Pareto hattı (yeşil çizgi) ve amaç fonksiyonlarının değişimi,  $(-5 \leq x \leq 5)$ .



Şekil 5.15 Gaye uzayında Pareto hattı (yeşil çizgi) ve ideal noktalar,  $(-5 \leq x \leq 5)$ .







Öte yandan Şekil 5.14'teki Pareto hattına daha yakından bakılarak ideal noktanın konumu da değerlendirilebilir. İdeal nokta, amaç fonksiyonlarının aşgari değerlerini aldığı noktalardır. Buna göre  $f_1(x)$  amaç fonksiyonunun minimum nokta değeri  $-0.25$ ,  $f_2(x)$  amaç fonksiyonunun minimum nokta değeri ise  $-1$  olarak gözlemlenebilir. Şekil 5.15'te Pareto hattını ve ideal noktaları (kırmızı renkli) daha detaylı olarak görmek mümkündür.

Hatırlanacağı üzere ideal ve beter noktalar, çözümün sınırları hakkında önemli fikirler vermekteydi. Burada da görüldüğü üzere söz konusu noktalar, olası Pareto optimal çözüm noktalarının sınırlarında konumlanmakta olup karar aşamasında bahse konu aralıkta kalmak gerekir. Söz konusu aralık dışındaki noktalarda ise amaç fonksiyon değerlerinin çatışmasından ziyade işbirliği ilişkisi görülmekte olup sınıra doğru bir sonraki nokta bir önceki noktadan daha baskın bir karaktere sahiptir.

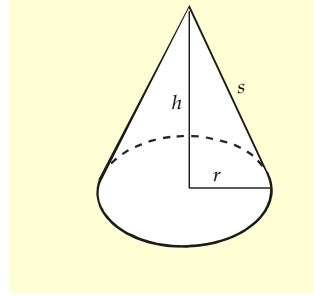
Çözümü üzerinde durulan sorunun gaye uzayının diğer bir özelliği de dışbükey olmasıdır. Genel bir ilke olarak, tüm amaç fonksiyonlarının dışbükey olduğu ve ayrıca tasarım uzayının dışbükey olduğu çok amaçlı eniyileme sorununun kendisi de dışbükey olarak düşünülebilir<sup>(13)</sup>. Bu tür problemlerde ağırlıklandırılmış toplama yönteminin gayet iyi bir performans gösterdiği söylenebilir. Ancak gaye uzayının dışbükey olmaması durumunda, bahse konu yöntem Pareto hattındaki her noktayı belirleyemeyebilir. Dolayısıyla karar verme sürecinde eksik veri ile hareket edilmiş olabilir. Ayrıca ağırlık katsayılarının yeknesak dağılımı dikkate alınsa dahi benzer homojen dağılım, Pareto hattı çözüm noktaları için elde edilemeyebilir<sup>(14)</sup>. Sözü edilen sorunlardan dolayı günümüzde çok amaçlı eniyileme sorunlarının çözümünde sezgisel yöntemlerin kullanımı gittikçe yaygınlaşmaktadır. Bu bağlamda özellikle genetik algoritma yaklaşımının başarılı sonuçlar verdiği söylenebilir<sup>(15)</sup>.





**MATLAB uygulaması**

Taban yarıçapı  $r$ , yüksekliği  $h$  ve yanal uzunluğu  $s$  olan bir koninin yanal yüzey alanı ile toplam yüzey alanını, belirli bir hacim değeri kısıtını da dikkate alarak optimize edelim. Burada tasarım değişkenleri olarak  $r$  ve  $h$  parametrelerini göz önünde bulundurabiliriz.



Koninin yanal mesafe değerini;  $s = \sqrt{h^2 + r^2}$ , yanal yüzey alanını;  $S = \pi r s$ , taban ve yanal yüzey alanlarından oluşan toplam yüzey alanını;  $T = \pi r s + \pi r^2$ , hacim değerini de;  $V = \pi/3 r^2 h$  denklemlerine göre belirlememiz mümkündür. Soruna ait kısıtlar kapsamında hacim kısıtını;

$$V \geq 200 \text{ cm}^3$$

olarak tanımlayalım. İlave olarak tasarım değişkenlerinin de alt ve üst sınırlarının var olduğunu düşünelim;

$$\begin{aligned} 0 &\leq r \leq 20 \text{ cm} \\ 0 &\leq h \leq 20 \text{ cm} \end{aligned}$$

İfade edilen tanımlamalara ve verilere göre çok amaçlı eniyileme sorununu;

$$\min_{r,h} \{S = \pi r s\}$$

$$\min_{r,h} \{T = \pi r (r + s)\}$$

Kısıt

$$\begin{aligned} -\frac{\pi}{3} r^2 h + 200 &\leq 0 \\ 0 &\leq r \leq 20 \text{ cm} \\ 0 &\leq h \leq 20 \text{ cm} \end{aligned}$$

şeklinde tanımlayabiliriz.



Matlab kapsamında söz konusu sorunun çözümü için “fgoalattain” komutu kullanılabilir. Söz konusu komut dâhilinde;

$$\min \left\{ \frac{f_i(x) - f_i^*(x)}{w_i} \right\}$$

amacı güdülür. Burada  $f_i^*(x)$  ifadesi, hedef değer olarak tanımlanmakta olup genellikle o amaç fonksiyonuna ait ideal nokta değeridir. Dolayısıyla çözümün ilk aşamasında, her bir amaç fonksiyonuna ait hedef değerın belirlenmesi gerekir;

<p><i>min S</i></p> <p>Kıst</p> $-\frac{\pi}{3}r^2h + 200 \leq 0$ $0 \leq r \leq 20 \text{ cm}$ $0 \leq h \leq 20 \text{ cm}$	<pre> clc; clear x0 = [1,1]; [x, fval] = fmincon(@myfun, x0, [], [], [], [], [0,0], [10,20], @mycon)  function f = myfun(x) f = pi * x(1)*sqrt(x(1)^2+x(2)^2);  function [c,ceq] = mycon(x) c = -pi/3*x(1)^2*x(2)+200; ceq = []; </pre>	<pre> x =     5.1305    7.2557  fval =     143.2303 </pre>
<p><i>min T</i></p> <p>Kıst</p> $-\frac{\pi}{3}r^2h + 200 \leq 0$ $0 \leq r \leq 20 \text{ cm}$ $0 \leq h \leq 20 \text{ cm}$	<pre> clc; clear x0 = [1,1]; [x, fval] = fmincon(@myfun2, x0, [], [], [], [], [0,0], [10,20], @mycon)  function f = myfun2(x) f = pi * x(1) * (x(1)+sqrt(x(1)^2+x(2)^2));  function [c,ceq] = mycon(x) c = -pi/3*x(1)^2*x(2)+200; ceq = []; </pre>	<pre> x =     4.0721    11.5176  fval =     208.3759 </pre>

Buna göre yapılan tarama neticesinde ideal nokta değerleri ilk amaç fonksiyonu için;

$$r = 5.131 \text{ cm}, \quad h = 7.256 \text{ cm}, \quad V = 200 \text{ cm}^3, \quad T = 225.92 \text{ cm}^2$$

$$S^* = 143.23 \text{ cm}^2$$

sonuçlarına kolayca ulaşılabilir. İkinci amaç fonksiyonu için ideal nokta değerleri;

$$r = 4.072 \text{ cm}, \quad h = 11.517 \text{ cm}, \quad V = 200 \text{ cm}^3, \quad S = 156.28 \text{ cm}^2$$

$$T^* = 208.38 \text{ cm}^2$$

şeklinde elde edilebilir.



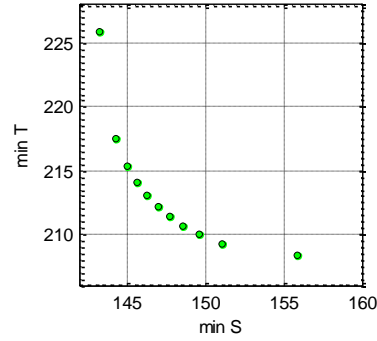
Hedeflerin belirlenmesi sonrasında her iki gayenin dikkate alındığı çok amaçlı optimizasyon problemi kolayca çözülebilir;

```

clc; clear; x0 = [1,1]; % başlangıç noktası
goal = [143.23,208.38]; % hedefler
n = 10; % Pareto hattı nokta sayısı
ds = 1/n ; x = zeros(n,2);
fval = zeros(n,2);
weight = zeros(n,2);

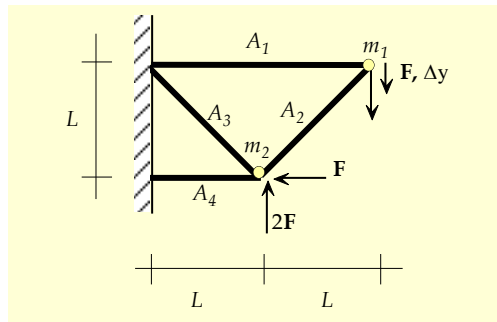
for i=0:n
    t = ds * i ;
    weight(i+1,:) = [t,1-t];
    [x(i+1,:),fval(i+1,:)] = fgoalattain(@myfun3,x0,
    goal,weight(i+1,:),[],[],[],[0,0],[10,20],@mycon);
end

plot(fval(:,1),fval(:,2),'go')
xlabel('min S')
ylabel('min T')
    
```



MATLAB uygulaması

Müteakip şekilde görüldüğü gibi dört kirişten oluşan bir kafes sistemini tasarladığımızı düşünelim.



Şekil 5.16 Dört-parçalı kafes kiriş yapısı.



Kafes sisteminin  $m_1$  ve  $m_2$  gibi iki mafsal noktasına, şekildeki gibi kuvvetlerin etki ettiğini, her bir kirişin kesit alanının  $A_i$ ,  $i = 1,2,3,4$  olarak tanımlandığını öngörelim. Kafes kiriş sisteminin tasarımında iki ayrı gaye güdülmekte olup bunlardan ilki, kirişlerin toplam hacminin asgari seviyede tutulması, diğeri ise  $m_2$  mafsalındaki dikey sehimin ( $\Delta y$ ) minimize edilmesidir. Ayrıca eniyilemede dikkate alınması istenen diğeri bir husus da her bir kirişe ait kesit alan değerinin belirli sınırlar içinde tutulması gereğidir. Buna göre amaç fonksiyonlarını;

toplam hacim miktarı;

$$f_1(\mathbf{A}) = L\{2A_1 + \sqrt{2}A_2 + \sqrt{2}A_3 + A_4\}$$

sehim miktarı;

$$f_2(\mathbf{A}) = \frac{FL}{E} \left( \frac{2}{A_1} + \frac{2\sqrt{2}}{A_2} - \frac{2\sqrt{2}}{A_3} + \frac{2}{A_4} \right)$$

olacak şekilde tanımlayabiliriz<sup>(16)</sup>. Burada  $E$  elastisite modülünü temsil etmektedir. Tanımlanan ifadelere göre çok amaçlı optimizasyon problemini;

$$\begin{aligned} & \min_{A_i} \{f_1, f_2\} \\ \text{Kısıt} & \\ & \frac{F}{\sigma} \leq A_i \leq 3 \frac{F}{\sigma}, \quad i = 1,4 \\ & \sqrt{2} \frac{F}{\sigma} \leq A_i \leq 3 \frac{F}{\sigma}, \quad i = 2,3 \end{aligned}$$

şeklinde ifade etmek mümkündür. Kısıtlarda geçen  $\sigma$  terimi, normal gerilimi simgelemektedir. Soruna ait terimlerin sayısal değerleri;

$$F = 10 \text{ kN}, \quad E = 2 \times 10^5 \frac{\text{kN}}{\text{cm}^2}, \quad \sigma = 10 \frac{\text{kN}}{\text{cm}^2}, \quad L = 200 \text{ cm}$$

olarak verilmektedir Matlab kapsamında söz konusu sorunun çözümü için “gamultiobj” komutu kullanılabilir. Söz konusu komut dâhilinde elitist yaklaşıma dayalı genetik algoritma kullanılmakta olup sıralamada baskınlık ön planda tutulmaktadır. Buna göre her iki gayenin dikkate alındığı çok amaçlı optimizasyon problemi kolayca çözülebilir;

```
clc; clear
[x,fval]= gamultiobj(@myfun3,4, [], [], [], [], [1 sqrt(2) sqrt(2) 1], [3 3 3 3])

function f = myfun3(x)
f(1) = 200 * (2*x(1)+sqrt(2)*x(2)+sqrt(2)*x(3)+x(4));
f(2) = 10*200/2e05 * (2/x(1)+2*sqrt(2)/x(2)-2*sqrt(2)/x(3)+2/x(4));
```



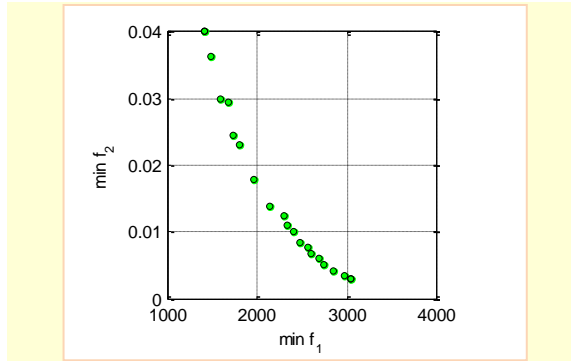


### Sonuç

$x =$	1.0000	1.4142	1.4142	1.0000	$fval =$	1400.000000000000	0.0400000000000000
	2.9990	2.9977	1.4216	2.9908		3047.70556288075	0.00289535582783927
	1.0784	1.5000	1.4302	1.0690		1473.95113143184	0.0363352017109516
	1.4464	1.6955	1.4393	1.6305		1791.33681160686	0.0231245789720858
	2.8246	2.9746	1.4195	2.9812		2968.90106248739	0.00337247206218228
	2.9990	2.9977	1.4216	2.9908		3047.70556288075	0.00289535582783927
	1.0141	2.2226	1.4155	1.1789		1670.42582074926	0.0294314611488520
	1.8088	2.5049	1.4520	2.4290		2328.51134307353	0.0111027473667499
	1.0000	1.4142	1.4142	1.0000		2395.26321010898	0.0100658543806017
	2.1690	2.2405	1.4208	2.4605		1581.82763803572	0.0298510685259641
	1.1081	1.5050	1.4230	1.5522		1957.74446570802	0.0177687279609973
	1.4656	2.0014	1.4201	2.0188		2847.70556288075	0.00422970383890743
	2.4990	2.9977	1.4216	2.9908		2475.50396957544	0.00849501667420493
	1.7284	2.7801	1.4176	2.9841		2558.48096202829	0.00758186688025934
	2.2440	2.4841	1.4209	2.7819		2143.00713709327	0.0138997925719446
	1.6755	2.3356	1.4214	2.0509		1728.34010418051	0.0245813312013369
	1.2626	1.8302	1.4218	1.5174		2683.78282193663	0.00605711693644264
	2.2065	2.9673	1.4337	2.7820		2288.74851273133	0.0123266601170917
	1.7145	2.2739	1.4766	2.7108		2606.50475359249	0.00668728253231578
	2.1080	2.9015	1.4187	2.7069		2747.70556288075	0.00511937173892596
	2.2490	2.9977	1.4216	2.9908			

Bulunan amaç fonksiyon değerlerinin grafik ortamına aktarılması ile elde edilen Pareto hattının gaye uzayındaki gösterimi ise müteakip şekilde görülmektedir;

```
plot(fval(:,1),fval(:,2),'ko')
```



Şekil 5.17 Kafes giriş sorununa ait gaye uzayında Pareto optimal noktalar.



## Bilim Damlası



(17)

**B**

çinde bulunduğumuz evren, gökada, güneş sistemi ve Dünya, ister tekil isterse de beraberce değerlendirilsin, oldukça çok sayıda kayda değer özellikler içermekte. Söz konusu özelliklerin teolojik bir bakış açısıyla zekâ ürünü veya materyal bir yaklaşımla rastlantısal sürecin sonucu olduğu düşünülebilir, ancak her halükarda gerçek olan olgu, hayatın kendisinin bir mucize olduğudur...

Örneğin güneş sistemimiz, Samanyolu gökadasının merkezinden belirli bir mesafe uzağında ve ayrıca onun spiral kollarının da arasında konuşlanmış durumda. Takip ettiği yörünge, ortak-dönü yarıçapına (co-rotation radius) sahip. Söz konusu yarıçapın özelliği, spiral kollar ile aynı hızda dönme imkânına sahip olunması. Zira galaksinin merkezine daha yakın dönü yarıçapı kolları göre çok daha hızlı bir dönme, daha uzak dönü yarıçapı ise yine spiral kolları göre daha yavaş bir dönme anlamına geliyor. Böyle bir konumun hayat için önemi, çok uzun zamandır ve neredeyse 4.5 milyar yıldır dengeli bir yörüngeye sahip olunması. Aksi durum, zaman içerisinde yörüngenin, spiral kolların içine doğru kayması ve Güneş sistemimizin bu bölgelerdeki yoğun gök cismi, gaz ve toz yığınlarıyla karşılaşması anlamına gelecekti... Diğer bir ifade ile zaman içerisinde Güneş sistemi ve gezegenler, pek çok çarpışma ve yörünge kaymasına maruz kalabilecekti... Ortak-dönü yarıçapının diğer bir getirisi, ufkumuzun açık olması ve bu sayede göz yüzüne baktığımızda Samanyolu galaksisini rahatça gözlemleyebilme imkânı sunması. Zira spiral kolların madde yoğun kütlelerinden uzakta ve güven içerisindeyiz...

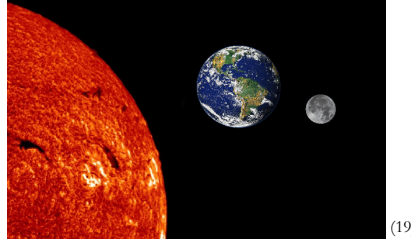


(18)

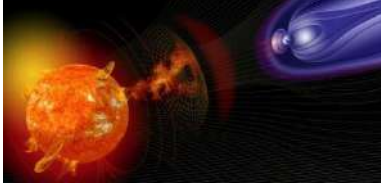
Güneş sistemimiz içerisindeki en büyük gezegen olan Jüpiter, evrendeki yolculuğumuzda adeta yeryüzünün önünü açan bir büyük çekim kütlesi niteliğinde. Zira büyüklüğü sayesinde, Güneş sisteminin yörüngesi üzerindeki irili ufaklı pek çok gök cismini toparlamakta ve Dünya gezegeninin istenmeyen çarpışmalardan uzak durmasını sağlamakta...



Yeryüzüne yakın konumdaki Mars ve Venüs gezegenleri ile komşumuz Ay'ın kütesel olarak mevcudiyeti ve sahip oldukları yörüngeleri, Dünya gezegeninin çok nadir bir şekilde eliptikliğini oldukça düşük seviyelerde (0.03) olmasını sağlamakta... Normalde sahip olunan kütesel büyüklüğe göre olması gereken eliptikliğın beşte biri mertebesindeki bu değerin getirisi, yörüngesinin daha da dengeli ve kararlı olması. Bu sayede yaşamın var olması için çok daha uzun denge dönemleri ve buzul çağların uzun dönemler sonrasında ortaya çıkması mümkün olmaktadır.



(19)



(20)

Gezegeneimizin kutuplar arası eksenini, dönüş yörüngesine göre yaklaşık 23.5° eğime sahiptir. İlk anda mükemmel olmayan bu eğikliğın yaşam için önemi, Güneş ışınlarından çok daha fazla faydalanmayı sağlaması. Karasal alanının kuzey yarım küresinde toplandığı bir kütle dağılımına sahip gezegenin, Güneş ışınlarını çok daha fazla miktarda soğurması ve yaşam için mükemmel bir ortam sağlaması, söz konusu eğikliğın bir getirisi...

Zamanımızdan yaklaşık 4.25 milyar yıl önce Dünya'ya göre daha küçük ve muhtemelen Mars büyüklüğünde başka bir gök cisminin gezegenimize çarptığını ve çarpmanın etkisi ile bazı ciddi değişimlerin olduğunu düşünmekteyiz... Bunlardan ilki, Ay gibi bir uydumuzun olması. Ancak çarpışmanın daha da ilginç yönü, Ay'ın oluşmasının yanında gezegenimizdeki çok daha kalın miktardaki atmosferin önemli bir kısmının uzaya savrulması ve bu sayede daha ince bir hava tabakasına sahip olunması... İlk anda kulağa olumsuz gibi gelen bu oluşumun hayata yansması, sera etkisinin bertaraf edilmesi. Yapılan öngörülere göre Dünya'dan çok daha kalın bir atmosfere sahip olan Venüs gezegeni, oluşan sera etkisi nedeniyle oldukça sıcak bir yüzeye sahip ve bu da hayatı imkânsız hale getirmekte...

Ay oluşumunun diğer bir ayrıntısı, gezegenimizin dönüş hızını düşürmesi. Ay olmadan var olan bir dünyada günün 8 saat olması mümkündür. Bu durumda yeryüzünde, hızı saatte 500 mil'i bulan rüzgârlar hüküm sürerdi. Ancak Ay sayesinde, yıllar içerisinde dönüş hızı oldukça mutedil seviyelere geldi ve 24 saatte bir tur atan gezegen döngüsü oluştu. Çarpışma ile meydana gelen kütesel artış, yeryüzünün çekirdeğinin çok daha yoğun bir kıvama gelmesini ve merkezinde ağır nikel-demir kitlesinin oluşmasını sağladı. Bu sayede oluşan manyetik alan, gezegenin etrafında bir manyetik kalkanın oluşmasını ve bu vesile ile Güneşten gelen radyasyonun engellenmesini sağladı. Oysa yaşam, radyasyonun yoğun olduğu bir ortamda mümkün olmazdı<sup>(21)</sup>...







- [1] Fate versus Free Will, Alıntı: <<https://userscontent2.emaze.com/images/fb81f8ff-358d-4083-b8f7-9b2f6734adfa/0f75febe-a303-4b7b-817c-93b7dffa3f7d.jpg>> [12 Şubat 2017].
- [2] R. Altıntaş, Mevlana'da irade hürriyeti, Alıntı: < <http://dusundurensozler.blogspot.com.tr/2010/09/mevlanada-irade-hurriyeti-1.html>> [12 Şubat 2017].
- [3] Düşünce Okulu, Sureler, Hicr 39, Alıntı: < <http://www.dusunceokulu.net/meal.php?surenno=15&ayetno=39>> [12 Şubat 2017].
- [4] A. Alfaris, Multiobjective optimization, Alıntı: < [https://ocw.mit.edu/courses/engineering-systems-division/esd-77-multidisciplinary-system-design-optimization-spring-2010/lecture-notes/MITESD\\_77S10\\_lec14.pdf](https://ocw.mit.edu/courses/engineering-systems-division/esd-77-multidisciplinary-system-design-optimization-spring-2010/lecture-notes/MITESD_77S10_lec14.pdf)> [12 Şubat 2017].
- [5] Multiobjective optimization, from Wikipedia: The Free Encyclopedia. Alıntı < [https://en.wikipedia.org/wiki/Multi-objective\\_optimization](https://en.wikipedia.org/wiki/Multi-objective_optimization)> [12 Şubat 2017].
- [6] Multiple-criteria decision analysis, from Wikipedia: The Free Encyclopedia. Alıntı < [https://en.wikipedia.org/wiki/Multiple-criteria\\_decision\\_analysis](https://en.wikipedia.org/wiki/Multiple-criteria_decision_analysis)> [12 Şubat 2017].
- [7] Spitfire underside, Alıntı < [http://pre01.deviantart.net/2778/th/pre/i/2013/019/9/1/spitfire\\_underside\\_fas\\_2012\\_by\\_somebody\\_somewhere-d5s0ilp.jpg](http://pre01.deviantart.net/2778/th/pre/i/2013/019/9/1/spitfire_underside_fas_2012_by_somebody_somewhere-d5s0ilp.jpg)> [12 Şubat 2017].
- [8] F-117, Alıntı < [http://nevskaa-bastion.ru/VVT/F-117\\_07.jpg](http://nevskaa-bastion.ru/VVT/F-117_07.jpg)> [29 Mayıs 2017].
- [9] B-2 Spirit, Alıntı < [http://www.science-at-home.net/gallery/history/albums/userpics/normal\\_Tarnkappen\\_flugzeug.jpg](http://www.science-at-home.net/gallery/history/albums/userpics/normal_Tarnkappen_flugzeug.jpg)> [12 Şubat 2017].
- [10] Multi-objective optimization, Alıntı <<https://engineering.purdue.edu/~sudhoff/ee630/Lecture09.pdf>> [13 Şubat 2017].
- [11] Multi-Objective Optimization, Alıntı < [www.polymtl.ca/namp/docweb/Modules\\_Web/M15\\_Part1\\_Tier1\\_MultiObjective.ppt](http://www.polymtl.ca/namp/docweb/Modules_Web/M15_Part1_Tier1_MultiObjective.ppt)> [14 Şubat 2017].
- [12] K. Duh, Multi-objective optimization, Alıntı < <http://cl.naist.jp/~kevinduh/notes/duh11multiobj-handout.pdf>> [16 Şubat 2017].
- [13] G. Narzisi., Multi-Objective Optimization: a quick introduction, Courant Institute of Mathematical Sciences, New York University, 2008.
- [14] S. S. Majeed, Multi-Objective Optimization, Alıntı < <http://www.slideshare.net/SEMEDARSALIM/multi-objective-optimization>> [18 Şubat 2017].
- [15] A simple multi-objective optimization problem, Alıntı < [http://www.math.unipd.it/~marcuzzi/DIDATTICA/LEZ&ESE\\_PIAI\\_Matematica/3\\_cones.pdf](http://www.math.unipd.it/~marcuzzi/DIDATTICA/LEZ&ESE_PIAI_Matematica/3_cones.pdf)> [18 Şubat 2017].
- [16] C.A. Coello Coello, A short tutorial on evolutionary multiobjective optimization, Lecture Notes in Computer Science, Evolutionary Multi-Criterion Optimization Proceedings, 2001, pp. 23-24.
- [17] Alıntı<[https://s3.amazonaws.com/user-media.venngage.com/981469-e2b6629f26488238c53dc52ece\\_a2a642.jpg](https://s3.amazonaws.com/user-media.venngage.com/981469-e2b6629f26488238c53dc52ece_a2a642.jpg)> [05 Nisan 2017].
- [18] Alıntı < <http://wallpaper-gallery.net/single/solar-system-wallpaper/solar-system-wallpaper-4.html> > [05 Nisan 2017].





- [19] Alıntı < [http://4.bp.blogspot.com/\\_RgAasUT4r0Y/SpuJjmRAhUI/AAAAAAAAAw/OgycoSGVojg/s800/sun%2Bmoon%2Bearth.gif](http://4.bp.blogspot.com/_RgAasUT4r0Y/SpuJjmRAhUI/AAAAAAAAAw/OgycoSGVojg/s800/sun%2Bmoon%2Bearth.gif) > [05 Nisan 2017].
- [20] Alıntı < <http://i1.wp.com/locklip.com/wp-content/uploads/2014/11/magnetic-field.jpg> > [05 Nisan 2017].
- [21] Alıntı < <http://www.godandscience.org/apologetics/designss.html#top> > [05 Nisan 2017].



