



Selçuk Üniversitesi  
Teknik Eğitim Fakültesi  
Elektronik-Bilgisayar Eğitimi

8061011007 Genetik Algoritmalar ve Uygulamaları

Ders Notları

Bahar, 2006

**Yrd.Doç.Dr. Mehmet ÇUNKAŞ**  
mcunkas@yahoo.com

# İÇİNDEKİLER

<b>1. GENETİK ALGORİTMALAR İLE OPTİMİZASYON .....</b>	<b>1</b>
1.1. OPTİMİZASYON .....	1
1.2. GENETİK ALGORİTMALAR VE EVRİMSEL HESAPLAMA.....	5
1.3. RASGELE ARAŞTIRMA ALGORİTMASI .....	7
1.4. İKİLİ KODLU GENETİK ALGORİTMALAR .....	8
1.4.1. İKİLİ KODLU GENETİK ALGORİTMALARIN OPERATÖRLERİ .....	10
1.4.2. AMAÇ FONKSİYONU VE PARAMETRELER .....	12
1.4.2.1. PARAMETRELERİN TANIMLANMASI VE KODLAMA .....	13
1.4.2.2. BAŞLANGIÇ POPÜLASYONU .....	16
1.4.2.3. DOĞAL SEÇİM.....	18
1.4.2.4. ÇAPRAZLAMA .....	22
1.4.2.5. MUTASYON .....	23
1.4.2.6. GELECEK NESİL .....	25
1.4.2.7. YAKINSAMA.....	28
1.5. GERÇEK KODLU GENETİK ALGORİTMA .....	29
1.5.1. AMAÇ FONKSİYONU VE PARAMETRELER .....	30
1.5.1.1. BAŞLANGIÇ POPÜLASYONU .....	31
1.5.1.2. DOĞAL SEÇİM.....	32
1.5.1.3. SEÇİM.....	34
1.5.1.4. ÇAPRAZLAMA .....	34
1.5.1.5. MUTASYON .....	37
<b>2. ÖRNEKLER .....</b>	<b>42</b>
2.1. BİR AÇIKLAYICI ÖRNEK – GA NASIL ÇALIŞIR .....	42
2.2. BİR GANIN NASIL ÇALIŞTIĞINI GÖSTEREN DİĞER BİR ÖRNEK.....	46
2.3. İKİ DEĞİŞKEN İÇEREN BİR BAŞKA ÖRNEK .....	49
2.4. İKİ DEĞİŞKEN İÇEREN DİĞER BİR ÖRNEK.....	51
2.5. GEZGİN SATICI PROBLEMİNİN GENETİK ALGORİTMALARLA UYGULAMASI.....	62
<b>3. ŞEKİL LİSTESİ .....</b>	<b>789</b>

# 1. GENETİK ALGORİTMALAR İLE OPTİMİZASYON

## 1.1. Optimizasyon

Optimizasyon bir şeyin daha iyisini gerçekleştirme sürecidir. Bir mühendis veya bilim adamı yeni bir fikir ortaya koyar. Optimizasyon bu fikrin geliştirilmesine yardımcı olur.

Optimizasyon; fikirleri ilerletmek için ortaya atılan bilgileri kullanarak, başlangıç kavramlarını (parametrelerini) değiştirmektir. Eğer fikri etkileyen parametreler elektronik formata dönüştürülürse bilgisayar mükemmel bir optimizasyon aracı olur. Bir takım bilgiler bilgisayara girilir ve bir çözüm elde edilir. Elde edilen bu çözümün değerlendirilmesi optimizasyon aracılığıyla gerçekleştirilir.

Bir problemin birden fazla çözümü varsa, en iyi çözümü bulmak gerekir. Aslında “en iyi” tanımlaması izafi kavramdır. Optimal çözüm, problemi formülize eden kişiye bağlıdır. Eğitim, fikirler, ideolojiler ve sosyal durumlar “en iyi” tanımlamasında etkili olan faktörlerdir. Bazı problemlerin tam cevabı bulunurken, bazıları optimal noktalar olarak bilinen, değişik minimum ve maksimum noktalarına sahiptir. İşte burada “en iyi” izafi olabilir.

Yer kürede yaşayan insanoğlunun hayatı, optimizasyon problemleri ve çözümleri ile doludur. Bununla ilgili birkaç örnek verilebilir. Hem işe zamanında gitmek, hem de uyku miktarını artırmak için ne zaman kalkılmalıdır? İşe giderken en iyi yol neresidir? Bir projeyi yürütürken ilk olarak nereden başlanmalıdır? Bir şeyin uzunluğunu kısaltmak, ağırlığını azaltmak bir ürünün cazibesini artırmak ve maliyetini düşürmek optimizasyon tekniğinin ilgi alanı içerisindedir. Optimizasyonun temel aşamaları Şekil 1’de gösterilmiştir.



Şekil 1: Optimizasyonun temel aşamaları

Bir deney düzeneğinde, maksimum veya minimum sonuç/çıkış elde edebilmek için, cihazın giriş karakteristiklerinin ayarlanması işlemi de bir optimizasyon sürecidir. Bir fonksiyonun girişinde, çeşitli parametreler çıkışında, uygunluk veya maliyet değerleri vardır. Fonksiyon veya süreç; maliyet (cost) fonksiyonu, amaç (objective) fonksiyonu veya uygunluk (fitness) fonksiyonu olarak tanımlanır. Eğer süreç deneysel ise, giriş parametreleri fiziksel büyüklüklerden oluşur (Pierre 1992).

Hayat; tesadüf gibi görünen olaylarda, verilen kararlar nedeniyle çok ilginçtir. Kuantum teorisi, sonsuz boyut olduğunu ve her boyutun bir kararı temsil ettiğini söylemektedir. Hayat, lineer olmadığından kaos önemli bir rol oynar. Başlangıç şartlarında küçük bozucu etkiler, çok farklı ve tahmin edilemeyen çözümlere sebep olmaktadır. Bir ürün tasarımında yüksek dereceli karmaşıklıklar ortaya çıkmaktadır. Bilimin gelişmesiyle birlikte geçmişte çözülemeyen karmaşıklıkların yeni çözümleri üretilmiştir. Gelişmenin bu sürecinde optimizasyon önemli bir rol oynamıştır.

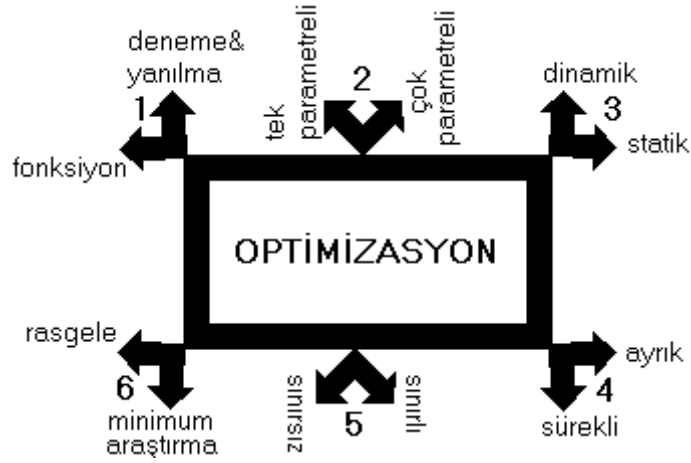
Çoğu optimizasyon metotları, kök veya sıfır araştırma işlemini kullanır. Matematiksel yaklaşımda, kök bulmak için fonksiyonun sıfır olduğu yerler araştırılırken, optimizasyon da türevin sıfır olduğu yerleri araştırmak gerekmektedir. Türev hesaplama her zaman kolay bir iş değildir. Teknik problemlerin birçoğu, köklerini bulmak üzere formülize edilebilir. Fakat bir kısım optimizasyon yöntemleri bu kökleri bulmada yetersiz kalmaktadır

Optimizasyonda diğer bir zorluk; elde edilen bir sonucun, global veya lokal bir çözüm olup olmadığının belirlenmesidir. Örneğin; bulunan kökün optimal bir çözüm olduğunu anlamak zordur. Çünkü bütün kökler, fonksiyonu sıfır yapmaktadır.

Lineer olmayan bir fonksiyonun da minimumunu bulmak oldukça zordur. Bu tip problemler ya lineer bir yaklaşımla veya optimizasyon bölgesini küçük bir bölge ile sınırlamakla çözülür.

Optimizasyon algoritmaları Şekil 2’de gösterildiği gibi altı grupta ele alınabilir. Ancak kesin hatlarıyla altı gruba ayrıldığı söylenemez. Örneğin; dinamik optimizasyon problemi sınırlı veya sınırsız olabilir. Bazı parametreler ayrık veya sürekli olarak tanımlanabilir (Broyden 1965). Şekil 2’de verilen optimizasyon algoritmaları aşağıdaki gibi açıklanabilir.

- 1) *Deneme-yanılma optimizasyonu*; İşlem hakkında çok fazla bilgi olmaksızın çıkışı etkileyen parametrelerin ayarlanmasıdır. Örneğin TV 'de en iyi görüntü ve ses, deneme yanılma yoluyla ayarlanır. TV'deki görüntü ve sesin, antenin hangi eğiminde iyileşeceği anten uzmanları tarafından sadece tahmin edilir. Deneysel çalışma yapanlar ve çoğu büyük kâşifler bu yolu kullanmışlardır. Bunun aksine, matematiksel fonksiyonun optimizasyonunda, matematiksel formül ile süreç tanımlanır. Fonksiyonun optimum çözümünü bulmada değişik metotlar uygulanır. Bu yaklaşım teorisyenler tarafından tercih edilir.



Şekil 2: Optimizasyon algoritmalarının grupları

- 2) *Tek ve çok parametrelili optimizasyon*; Sadece bir parametre varsa, optimizasyon bir boyutludur. Birden fazla parametreye sahip fonksiyon için çok boyutlu optimizasyon gereklidir. Boyut sayısı artarsa, optimizasyonun zorluk derecesi de artar. Çok boyutlu optimizasyon metodunda, bir boyutlu optimizasyon metodu yaklaşımı kullanılır.
- 3) *Statik ve dinamik optimizasyon*; Statik optimizasyon zamandan bağımsızdır, dinamik optimizasyon ise zamana bağlı olarak çıkış üretir. Örneğin; bir şehrin kenar mahallesinde oturan bir insanın merkezdeki işine gitmesi için birçok yol olduğunu kabul edilirse en iyi yolun hangisi olduğu sorgulanabilir. Mesafe açısından bakılacak olursa problem statiktir. Çözüm, haritayı ve arabanın kilometre/saat'i kullanılarak bulunabilir. Pratikte değişkenlerin

çokluğu nedeniyle problem, pek de basit değildir. En kısa yol, en hızlı yol değildir. En hızlı yolu bulmak dinamik bir problemdir ve zamana, havanın durumuna, kazalara vb. bağlıdır.

- 4) *Sürekli ve ayrık parametrelili optimizasyon*; Sürekli parametreler sonsuz değer alırken ayrık parametreler sınırlı değerler alır. Örneğin yapılacak işler bir liste halinde verilmiştir. Bu işlerin yapılması bir birinden bağımsız olduğundan ayrık parametrelili düşünülebilir. Ayrık parametrelili optimizasyon kombinasyonel bir optimizasyon olarak da adlandırılabilir. Bir çizgide  $f(x)$ 'in minimum değerini bulmaya çalışmak, sürekli parametrelili optimizasyon olarak tanımlanır.
- 5) *Sınırlı ve sınırsız optimizasyon*; Sınırlı optimizasyon, parametreleri bir tanım aralığında değerlendirir. Sınırsız optimizasyonda ise parametreler her hangi bir değerde olabilir. Değişkenlerin sınırları kaldırılarak sınırlı parametreler sınırsız parametrelere çevrilirler. Çoğu nümerik optimizasyon rutinleri sınırsız parametrelilerle çalışırlar. Örnek olarak  $f(x)$  fonksiyonunu ele alalım ve sınırlar  $-1 \leq x \leq 1$  arasında olsun. Bu fonksiyon  $x = \sin(u)$  tanımı kullanılarak sınırsız optimizasyona dönüştürülür. Burada  $u$ 'nun değeri ne olursa olsun  $x$ ;  $(-1,1)$  aralığında değişecektir. Sınırlı optimizasyon, lineer denklemler ve lineer sınırlarla parametreleri optimize ettiği zaman, program lineer program olarak adlandırılır. Sınırlar ve maliyet denklemleri nonlineer ise, program da nonlineer programlama problemi olur.
- 6) *Rasgele ve minimum araştırma algoritmaları*: Bazı algoritmalar parametrelerin başlangıç değerlerini ayarlayarak uygunluk değerlerini minimize etmeye çalışır. Bu araştırma tekniği, hızlı olmakla beraber lokal minimumlara ulaşabilir. Bunlar nümerik metotlara dayanan klasik optimizasyon algoritmalarıdır. Bir parametreden hareketle diğer parametreyi tespit etmek, bazı deterministik adımlarla gerçekleştirilmektedir. Diğer taraftan rasgele metotlar; parametrelerin optimum çözümünü bulmada ihtimal hesaplarını kullanırlar. Bu metotlar yavaş olmakla birlikte global minimumu bulmada daha başarılıdırlar.

Yukarıdaki gruplandırmanın sonucunda optimizasyon metotları; Deterministik metotlar, İstatistiksel metotlar olmak üzere iki ana gruba ayrılabilir (Haataja 1994). Deterministik optimizasyon metotları, lokal maksimum veya minimuma yakınsayan algoritmalarıdır. Türevsel hesaplamalar veya türevsel yaklaşımlar deterministik metotlara örnek verilebilir. Rasgele araştırma algoritmaları gibi istatistiksel metotlar ise global minimum veya maksimumu bulmada bazı stratejileri ve rasgele sayıları kullanırlar (Palko 1996). Son yıllarda PC'lerin hızlarındaki artış bu algoritmaların uygulama sahasında sıkça görülmesine neden olmuştur (Wurtz F. ve ark. 1997).

## **1.2. Genetik Algoritmalar ve Evrimsel Hesaplama**

Evrimsel hesaplama bir optimizasyon işlemidir. Öyle ki, amaç, hayatta kalacak bireylerin yeteneklerini geliştirmektir. Evrimsel hesaplama (EC), bir arama işleminde doğal seçimin benzetimidir. Doğada, organizmalar hayatta kalmak ve çoğalmak için yeteneklerini etkileyen belli karakteristiklere sahiptir. Bu karakteristikler organizmanın kromozomlarında içerilen bilginin uzun stringleri şeklinde temsil edilirler. Eşlenerek çoğaltmadan sonra, yavru kromozomlar her çiftten gelen bilginin bir kombinasyonundan oluşur. Umut verici bir biçimde sonuç, her çiftin en iyi karakteristiklerini içeren yavru kromozomlar olacaktır. Doğal seçim işlemi, uygunluk değeri en yüksek bireyin seçilmesini sağlar.

Evrim doğal seçim aracılığıyla, popülasyon içinden rasgele bir şekilde seçilmiş bireyler, uygun kromozom değerleri arama olarak düşünülebilir. Bu amaçla, bir evrim algoritması (EA) verilen bir probleme en uygun çözüm için olasılıklı bir aramadır.

Evrim algoritmaları; genetik algoritmalar (GA), genetik programlama (GP), evrimsel programlama (EP), evrimsel strateji (EV) ve benzerlerini içeren alt bölümlere ayrılabilir.

Genetik algoritma (GA)'nın terminolojisinin anlaşılması için "doğal seçim" in (seleksiyonun) anlaşılması gerekir. Dünyayı gözlemleyecek olursak, olup biten olaylarda doğal seçim göze çarpar. Birbirinden ayrı muazzam organizmalar ve bu organizmalardaki karmaşıklık, inceleme ve araştırma konusudur. Organizmaların

niçin böyle olduđu ve nasıl bu aşamaya geldiđi sorgulanabilir. Bir başka açıdan meseleye şöyle bakılabilir. Sanki bu günün dünyasında, çok büyük optimizasyon algoritmaları oluşturulmuş ve binlerce iterasyon sonunda yer yüzünde optimum çözüm elde edilmiştir. Amaç fonksiyonu, yaşam mücadelesini temsil eder. İnsanođlu bunu maksimize etmek ister (Grant 1985). Adaptasyon ve uygunluđun seviyesi, dünyada uzun süre yaşayabilmenin göstergesi haline gelmiştir. Evrim süreci, hayat şartlarına en uygun olanın yaşamasını sağlayan büyük bir algoritmadır. Eđer çevreyi deđiştirme zeka ve yeteneđine sahip olunursa hayatta, global maksimum elde edilebilir (Haupt 1998).

Gen, temel kalıtım birimidir. Organizmanın genleri; DNA (Deoxyribo Nucleic Acid) formunda bir çift kromozomdur. Organizmanın her bir hücresi benzer miktarda kromozomlar içerirler. Her bir vücut hücresindeki kromozom miktarı farklıdır. Örneđin; sivrisinek de 6, kurbađada 26, insanda 46 ve alabalıkta 94 adet kromozom vardır. Genler, her biri farklı karakteristiđi temsil eden iki fonksiyonel formda meydana gelir. Bu formların her biri “allel” olarak bilinir (Curtis 1975). Örneđin allel deđerine göre insanın gözünün biri mavi olurken diđeri kahverengi olmaktadır. Kromozomlardaki allel’lerin kombinasyonları, fertlerin kişisel özelliklerini belirler. Fert üzerinde allel’lerden biri baskın olurken diđeri pasif olmaktadır.

Genetik algoritmalar, biyolojik süreci modelleyerek fonksiyonları optimize eden evrim algoritmalarıdır. GA parametreleri, biyolojideki genleri temsil ederken, parametrelerin toplu kümesi de kromozomu oluşturmaktadır. GA’ların her bir ferdi kromozomlar (bireyler) şeklinde temsil edilen popülasyonlardan oluşur. Popülasyonun uygunluđu, belirli kurallar dâhilinde maksimize veya minimize edilir. Her yeni nesil, rasgele bilgi deđişimi ile oluşturulan diziler içinde hayatta kalanların birleřtirilmesi ile elde edilmektedir (Angeline 1995).

Bu metot, uzun çalışmaların neticesinde ilk defa John Holland (1975) tarafından uygulanmaya başlandı. En son onun öğrencisi olan David Goldberg popüler oldu. David Goldberg tezinde; gaz boru hattının kontrolünü içeren bir problemin çözümünü genetik algoritma ile gerçekleřtirdi (Goldberg,1989).

GA'ların avantajları;

- Sürekli ve ayrık parametreleri optimize etmesi
- Türevsel bilgiler gerektirmemesi
- Amaç fonksiyonunu geniş bir spektrumda araştırması
- Çok sayıda parametrelerle çalışma imkânı olması
- Paralel PC 'ler kullanılarak çalıştırılabilmesi
- Karmaşık amaç fonksiyonu parametrelerini, lokal minimum veya maksimumlara takılmadan optimize edebilmesi
- Sadece tek çözüm değil, birden fazla parametrelerin optimum çözümlerini elde edebilmesi olarak sıralanabilir.

GA lar arama ve optimizasyon için sezgisel yöntemlerdir. Geniş arama algoritmalarının aksine, genetik algoritmalar en iyiyi seçmek için tüm farklı durumları üretmez. Bundan dolayı, mükemmel çözüme ulaşamayabilir. Fakat zaman kısıtlamalarını hesaba katan en yakın çözümlerden biridir. GA lar şartlara uyum sağlayabilir. Bunun anlamı, önceden hiç bilgisi olmamasına karşın, olayları ve bilgiyi öğrenme ve toplama yeteneğine sahip olmasıdır.

Her problemin çözümü için GA kullanmak iyi bir yol değildir. Birkaç parametrelili analitik fonksiyonun çözümünde klasik metotlar daha hızlıdır. Böyle durumlarda, nümerik metotlar tercih edilmelidir. Paralel bilgisayarlar kullanılırsa GA daha hızlı sonuç verebilir.

Gelecek bölümde GA'larla benzerlik gösteren Saf Rasgele Araştırma Algoritması (pure random search) kısaca anlatılacaktır. GA'lara özel operatörler tanıtılacak ve birkaç örnekle çalışması gösterilecek. Ayrıca, ikili kodlarla çalışan (ikili kodlu) GA ve gerçek kodlarla çalışan (gerçek kodlu) GA'ların çalışma ve işleyişi anlatılacaktır.

### **1.3. Rasgele Araştırma Algoritması**

Rasgele Arama, belki de en basit arama işlemidir. Bir başlangıç arama noktasından veya başlangıç noktalarının kümesinden başlayan arama işlemi, arama uzayında rasgele noktaları araştırır ve kabul edilebilir bir çözüme ulaşıncaya veya maksimum iterasyon sayısı ulaşıncaya kadar devam eder. Rasgele aramayı

gerçekleştirmek son derece basit olmakla beraber, verimsiz olabilir. Uygun çözüm elde edinceye kadar geçen zaman çok uzun olabilir.

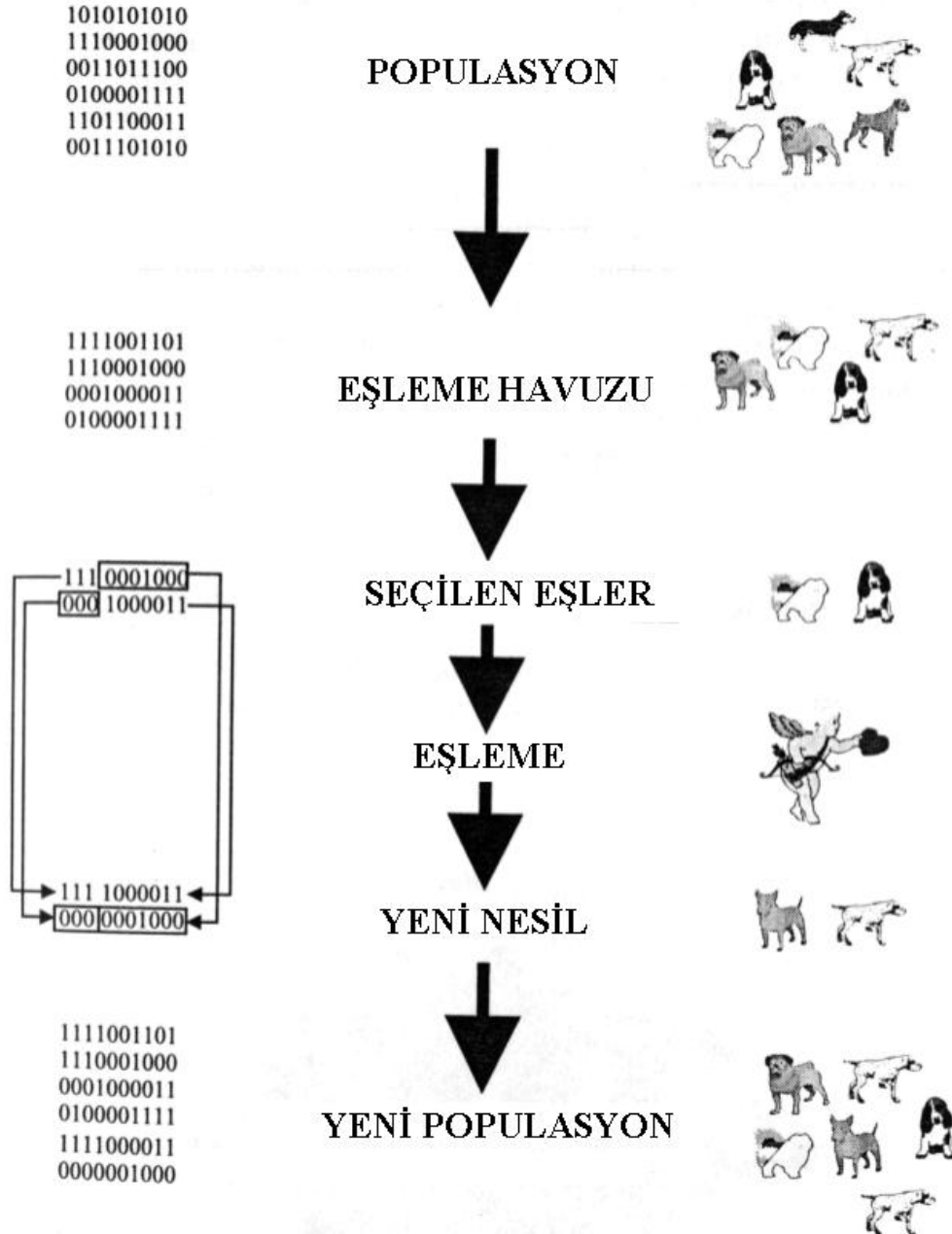
Rasgele araştırma için bir algoritma çalışması ve işleyişi aşağıda sunulmuştur.

- Adım 1.  $N$  başlangıç arama noktaları kümesini seç.  $C_g = \{ C_{g,n} = | n=1,2,\dots,N \}$ . Burada,  $C_{g,n}$   $I$  değişkenlerinin vektörü ve  $g=0$  dır. Her bir  $C_{g,n}$  elemanı,  $U(\min, \max)$  değişken değerlerin sınırı olmak üzere, verilen aralıklarda üretilir.
- Adım 2. Her bir  $C_{g,n}$  vektörünün (“uygunluk”)  $F(C_{g,n})$  doğruluğunu değerlendir.
- Adım 3. En iyi noktayı bul  $C_{g, best} = \min\{ F(C_{g,n}) \}$
- Adım 4. if  $C_{g, best} < C_{best}$  then  $C_{best} = C_{g, best}$   $C_{best}$  tümünün en iyi çözümü
- Adım 5. if  $C_{best}$  kabul edilebilir bir çözüm ise veya maksimum iterasyon sayısı aşılmış ise o zaman “dur” ve çözüm olarak  $C_{best}$  “dön”
- Adım 6. Her bir  $C_{g,n}$ ,  $\Delta C_{g,n}$  ile karıştır. Burada,  $\Delta C_{g,n} \approx N(0, \sigma^2)$  ve  $\sigma^2$  li küçük bir değişimi ifade eder.
- Adım 7.  $g = g + 1$  artır ve adım 2 ye git.

#### 1.4. İkili kodlu genetik algoritmalar

Genetik Algoritmalar (GAs), verilen bir veri kümesi için en iyi çözümü (parametreler kümesini) bulacak, doğrusal olmayan optimizasyon aracıdır. GA algoritma, mümkün çözümler kümesinin rasgele üretilmesiyle başlar. Parametreleriyle her bir çözüm, arama uzayında (kromozom veya nesil uzayı), uygunluk fonksiyonunun özel bir noktasını üretir. Her iterasyondaki bu farklı nesiller kümesi popülasyon olarak adlandırılır. Netice olarak, bir popülasyonun en iyi çözümünün bir parçasından (yarısı veya dörtte biri denebilir), çocuklar (yeni nesil) üretilir. Bu yeni nesil eskilerden daha iyi olması beklenir.

Biyolojik evrim ile ikili kodlarla çalışan GA arasındaki benzerlik Şekil 3’de görülmektedir. Her ikisinde de popülasyonun üyeleri rasgele başlar. Köpeklerin her birinin karakteristik özellikleri, sol taraftaki satırlarda verilmiştir.

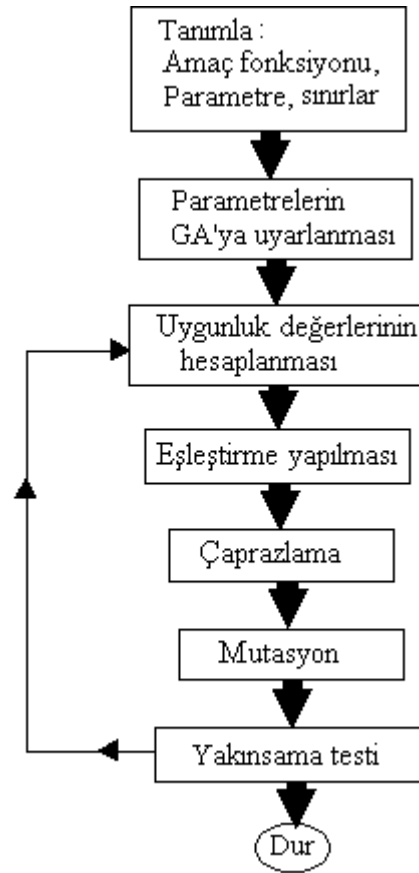


Şekil 3: İkili kodlu GA ile biyolojik evrim arasındaki benzetim (Haupt 1998)

En iyi havlayan köpekler üretilmek istenirse sadece birkaç köpeğin elde tutulması gerekir. Her bir köpek ikili sayı sistemiyle kodlanır. Yeni yavruların üretilmesi için bu popülasyonda rasgele iki adet köpek seçilir ve eşleştirilir. Eşleştirmenin sonucunda iyi havlayan köpeğin çıkma olasılığı yüksektir. Oluşan yeni nesil, eşleştirme havuzuna tekrar atılır. Başa dönülerek yeniden eşleştirme prosedürü tekrarlanır (Goldberg 1993). Bu işleme, en iyi havlayan köpek elde edilene kadar devam edilir.

#### 1.4.1. İkili kodlu genetik algoritmaların operatörleri

Diğer optimizasyon metotlarında olduğu gibi ikili kodlu GA'da da amaç fonksiyonu, parametreler ve sınırlar tanımlanır. Aynı şekilde yakınsama kontrol edilerek algoritma son bulur. İkili kodlu GA'nın akış diyagramı Şekil 4'de verilmektedir.

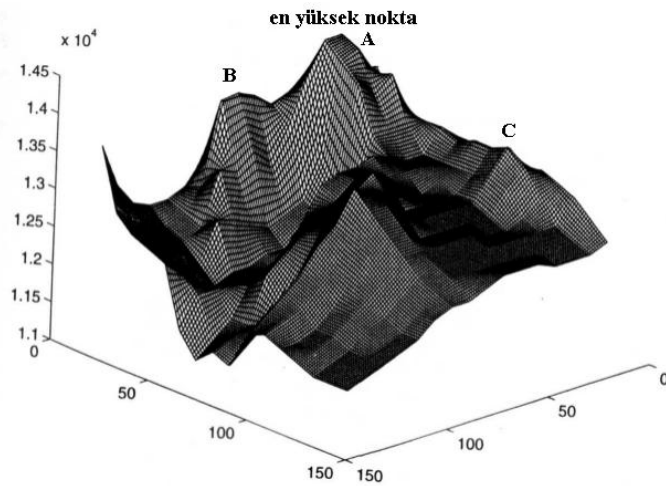


Şekil 4: İkili kodlu GA'nın akış diyagramı

Genel bir GA ařađıdaki program kod ile zetlenebilir.

1.  $y = 0$  ata.
2.  $C_g$  ilk neslini oluřtur.
3. While( yakınsama olmazken)
  - a. Her bireyin uygunluđunu deđerlendir.
  - b.  $g = g + 1$
  - c.  $C_{g-1}$  den ebeveynleri se.
  - d. Yavru  $O_g$ yi řekillendirmek iin aprazlama seilen ebeveynleri yeniden birleřtir.
  - e.  $O_g$  deki yavruyu mutasyona uđrat.
  - f. nceki nesil  $C_{g-1}$  ve yavru  $O_g$  den yeni nesli se.

GA ile topođrafik haritadaki ykseltiler veya vadiler arařtırılabilir. Optimizasyon algoritması vadiyi bulmak iin, ama fonksiyonunun minimum deđerini ve en yksek blgeyi bulmak iin ama fonksiyonunun maksimum deđerini arar. řekil 5’de en yksek tepe, deniz seviyesinden 14255 m yksekliginde olan A noktasıdır. B noktası 13281 m ve C noktası 11800 m yksekliginde dir. A noktasına ulařılıncaya kadar arada noktalar vardır. Bu yzden klasik metotlarda en yksek nokta olan A’yı bulmak zordur. GA ise bu noktayı rahatlıkla bulabilir.



řekil 5: Dađın  boyutlu grnm (Haupt 1998)

### 1.4.2. Amaç fonksiyonu ve parametreler

Amaç fonksiyonu, giriş parametrelerine göre çıkış üreten bir fonksiyondur. Bu fonksiyon matematiksel veya deneysel olabilir. Burada amaç, giriş parametreleri için uygun değer bulununcaya kadar çıkış üretmektir. Örneğin banyo küveti doldurulurken farkında olmadan optimizasyon yapılır. Giriş parametreleri, sıcak ve soğuk su çeşmeleridir. Amaç fonksiyonu deneysel sonuçlardır. Çünkü el, suyun içine sokularak suyun sıcaklığı ayarlanır ( Haupt 1998).

GA'ya, optimize edilecek parametre dizileri ve kromozomlar (bireyler) tanımlanarak başlanır. Kromozom,  $N_{par}$  adet parametreye sahip ise optimizasyon problemi  $N_{par}$  boyutlu olur. Parametreler  $P_1, P_2, P_3, \dots, P_{N_{par}}$  ise;

Kromozom =  $[P_1, P_2, P_3, \dots, P_{N_{par}}]$  olarak tanımlanır.

Örneğin; topoğrafik bir haritayı ele alalım. Amaç fonksiyonu giriş parametreleri, enlem (x) ve boylam (y) kabul edilirse, kromozom =  $[x,y]$  şeklinde tanımlanır. Burada  $N_{par}=2$ ' dir. Her bir kromozom;  $P_1, P_2$  parametrelerinin değerlerine göre hesaplanır.

$$F(\text{kromozom}) = F(P_1, P_2, \dots, P_{N_{par}})$$

Çoğu zaman amaç fonksiyonu oldukça karmaşıktır. Parametrelerin hangisinin önemli olduğuna karar verilmelidir. Parametreler çok fazla olursa GA, iyi sonuç vermeyecektir. Örneğin; arabalarda yakıt tasarrufu için önemli parametreler araba boyutları, motor boyutları ve malzeme ağırlıkları şeklinde belirlenebilir. Boya rengi veya koltuk tipi gibi diğer parametreler ya çok az etkili ya da hiç etkili değildir. Bazen parametrelerin seçimi ve değer tespiti, tecrübeyle veya deneme-yanılma yoluyla yapılır. Örneğin;  $F(w, x, y, z) = 2x + 3y + z/100000 + \sqrt{w}/9876$  şeklinde analitik bir fonksiyonun parametrelerinin, 0–10 arasında değiştiğini kabul edelim. Amaç fonksiyonu içerisinde w ve z parametreleri yüksek değerlere bölündüğünden 0–10 arasında değişmesi, sonucu pek fazla etkilemeyecektir. Eğer w ve z parametreleri, ihmal edilirse, 4 boyutlu problem 2 boyutlu olarak modellenebilir.

Optimizasyon problemlerinin çoğunda parametrelerin belirli sınırlar içerisinde kalması istenir. Örneğin; otomobilin ağırlığı sıfır olmayacak, boyu 10 m'yi geçmeyecek vb. sınırlamalar getirilebilir. Sınırsız parametreler ise herhangi bir değer

alabilir. İki türlü sınır belirlenebilir. Birincisi;  $0 \leq x \leq 10$  arasında değişen bir parametrede, eğer  $x=11$  değerini alırsa  $x=10$  yapılır ve  $x=-1$  değerini alırsa  $x=0$  yapılır. İkincisi;  $x = 5\sin y + 5$  olarak tanımlanırsa,  $y$ 'nin herhangi bir değerinde  $0 \leq x \leq 10$  arası değişecektir. Böyle bir dönüştürmeyle sınırlı optimizasyon problemi sınırsız optimizasyon problemine çevrilmiş olur (Holland, J.H. 1992).

Bağımlı parametreler, optimizasyon algoritmaları için özel problemler meydana getirir. Çünkü bir parametrenin değişimi diğer parametreleri de etkileyecektir. Örneğin; arabanın boyunu artırmak, ağırlığını da artıracaktır. Fourier serisi katsayılarında olduğu gibi bağımsız parametreler birbirlerini etkilemezler. GA literatüründe parametre etkileşimi, “epistasis” olarak adlandırılır. Biyolojide epistasis, genlerin birbirini etkilemesi olarak tanımlanır. Epistasis çok az olursa, minimum araştırma algoritmaları iyi sonuçlar üretir. GA’da Epistasis orta ve yüksek derecede olursa iyi sonuç vermemektedir, çok yüksek veya çok düşük olursa iyi sonuç vermemektedir (Haupt 1998).

#### **1.4.2.1. Parametrelerin tanımlanması ve kodlama**

Şekil 4’den, GA nın üç işlem için döngüye girdiğini görebiliriz.

- En iyi gen stringinin seçimi
- Genetik operasyon ( çaprazlama, mutasyon )
- Yeni gen stringleri (çocuklar) ile eski populasyonun kötü gen stringlerinin yer değiştirilmesi.

Optimizasyon döngüsü başlamadan önce, optimize edilmesi gereken parametreler istenilen şekle dönüştürülmek zorundadır. Buna kodlama (encoding) denir. Kodlama GA için önemli bir konudur. Çünkü sistemden gözlemlenen bilgiye bakış açısı büyük ölçüde sınırlandırılabilir. Gen stringi probleme özel bilgiyi depolar. Gen olarak adlandırılan her bir öge, genellikle değişkenler stringi olarak ifade edilir. Değişkenler ikili veya reel sayı şeklinde gösterilebilir ve aralığı probleme özel olarak tanımlanır.

İkili kodlama

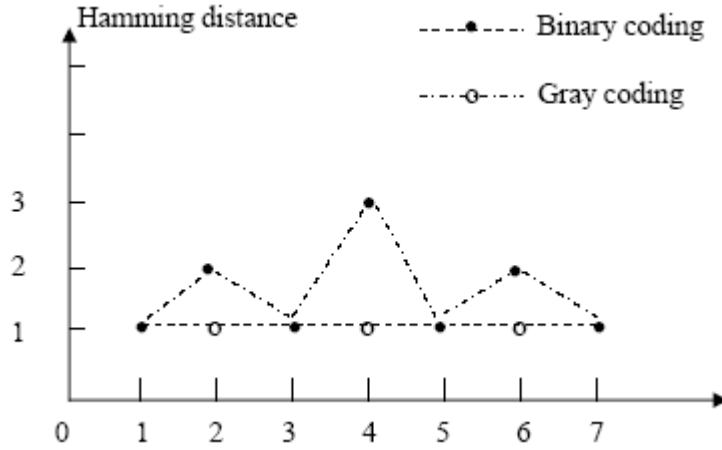
1	1	0	1	0	0	1	0
---	---	---	---	---	---	---	---

Reel sayı kodlaması

43	4	9	16	30	22	59	12
----	---	---	----	----	----	----	----

Şekil 6: Gen stringlerinde parametrelerin kodlanması

İkili kodlama sıklıkla kullanılırken, şekil 7’de gösterildiği gibi Hamming cliffs dezavantajına sahiptir. Bir Hamming cliff, iki bitişik sayısal değer ayrı bit gösterimlerine sahip olduğunda şekillenir. Mesela onluk sayılar 7 ve 8 i düşünün. İkili gösterimlerle ilgili olarak (4 bit gösterimini kullanan)  $7=0111$  ve  $8=1000$  4 hamming mesafesidir. (Hamming mesafesi birbirine benzemeyen ilgili bitlerin sayısıdır.)



Şekil 7: İkili ve Gray kodlama için Hamming mesafesi

Bu, değişkenlerdeki küçük bir değişimin uygunluktaki küçük bir değişiklikte sonuç vermesi gerektiği zaman, bir problemi ortaya koyar. Eğer örneğin, uygun çözüm 7 olsun. Fakat yeni en iyi çözüm 8 olduğunda; 8 i elde etmek için birçok bitin değiştirilmeye ihtiyaç duyulduğu aşikârdır. Hâlbuki uygunluk değerinde küçük bir değişim söz konusudur.

Alternatif bir bit gösterimi Gray Kodlama kullanmaktır. Ardışık sayısal değerlerin gösterimleri arasındaki Hamming Mesafesi bunlardan biridir. ( Tablo 1 de görülen) Tablo 1 İkili ve Gray kodlamayı 3-bit için karşılaştırır.

**Tablo 1** İkili ve Gray kodlama

	Binary	Gray
0	000	000
1	001	001
2	010	011
3	011	010
4	100	110
5	101	111
6	110	101
7	111	100

İkili sayılar dönüşüm kullanılarak kolaylıkla Gray kodlamaya çevrilebilirler.

$$g_1 = b_1$$
$$g_k = b_{k-1}\bar{b}_k + \bar{b}_{k-1}b_k$$

$b_k$  da k ikili sayıdaki bitin sırasıdır.  $b_1$  en anlamlı biti(yani sol en baştaki biti),  $\bar{b}_k$ ;  $b_k$  nin değilini, + lojik OR lamayı ve çarpımlar da lojik AND lemeyi gösterir.

GA'lar ayrıca integer veya reel değerleri temsil eder ve bir kromozomda değişkenlerin sırasının önemli bir rol oynadığı yerdeki komut tabanlı temsiller kullanılarak geliştirilmiştir.

Gaz için klasik temsil şeması, sabit uzunluklu ikili vektörlerdir. Bir I-boyutlu arama uzayının durumunda, her bir birey bir bit stringi olarak kodlanan her bir değişkenle I değişkenlerinden oluşur. Eğer değişkenlerin ikili değeri varsa, her bir kromozomun uzunluğu I bittir. Nominal değerli değişkenlerin durumunda, her bir nominal değer D boyutlu bit vektörleri olarak kodlanır. Bu değişken için 2D, her bir D-bit stringi farklı bir nominal değerle temsil edilir. Ayrık nominal değerlerinin toplam sayısıdır. Sürekli değerli değişkenlerin durumunda, her bir değişken D-boyutlu bir bit vektöre haritalanmalıdır.

$$\phi = R \rightarrow \{0,1\}^D$$

Sürekli uzay aralığı, sonlu bir  $[\alpha, \beta]$  aralığına sınırlandırılması gerekir. Standart ikili kod çözmeyi kullanarak her bir kromozom  $C_n$  in sürekli değişkeni  $C_{n,i}$  sabit uzunluklu bir bit stringi kullanılarak kodlanır. Örneğin, eğer  $z \in [z_{min}, z_{max}]$  bir 30-bit gösterimine dönüştürülmesi gerekirse, aşağıdaki formül kullanılabilir.

$$(2^{30} - 1) \frac{z - z_{\min}}{z_{\max} - z_{\min}}$$

Amaç fonksiyonunun uygunluk değerinin hesaplanması için her iterasyonda kromozomun 10'luk sayı sistemine çevrilmesi gerekir. On bitlik genlerden oluşan bir kromozom örneği aşağıda verilmiştir.

$$\text{Kromozom} = [\underbrace{1111001001}_{\text{gen1}} \underbrace{0011011111}_{\text{gen2}} \dots \dots \dots \underbrace{0000101010}_{\text{genNpar}}]$$

Bu kromozomun bit sayısı, toplam  $N_{\text{bit}} = N_{\text{gen}} \times N_{\text{par}}$  bit olarak bulunur. Şekil 5'de verilen topoğrafik harita, 128x128 değerlendirme noktasından oluşmaktadır. Eğer x ve y, iki gen olarak kodlanırsa her bir gen, "7" bittir ve x, y parametreleri,  $2^7$  'ye kadar değer alabilir. Bu değer aralıkları  $(40^0.15') \leq y \leq (40^0.16')$  ve  $(105^0.37'.30'') \geq x \geq (105^0.36')$ 'dir. Böylece kromozom aşağıdaki gibi ifade edilir (Haupt 1998).

$$\text{Kromozom} = [\underbrace{1100011}_x \underbrace{0011001}_y]$$

Genler, aşağıdaki denklem kullanılarak onluk sayı sistemine çevrilir. Buradaki ikili sayı, her bir gen için üretilen ikili sayıdır.

$$X_i = a_i + \text{decimal}(1001 \dots 001_2) (b_i - a_i) / (2^{m_i} - 1) \quad (3.1)$$

Burada;

$a_i$  = Parametrelerin alt sınır değeri

$b_i$  = Parametrelerin üst sınır değeri

$m_i$  = Gen'in bit sayısı'dır

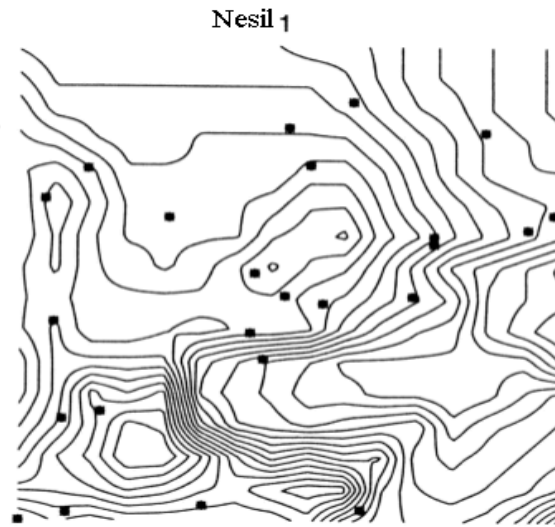
#### 1.4.2.2. Başlangıç popülasyonu

Kromozom, başlangıçta verilen popülasyon miktarına göre rasgele üretilir. Bu durumda  $N_{\text{ipop}} \times N_{\text{bit}}$  adet bitlerle matris oluşturulur.  $N_{\text{ipop}}$ , başlangıç popülasyonunun toplam sayısıdır,  $N_{\text{bit}}$ 'de kromozomun bit sayısıdır. Matrisin her bir satırı, bir kromozomu temsil eder. Kromozomlar ise, enlem ve boylamları temsil eder. Rasgele üretilen popülasyondaki parametreler, amaç fonksiyonunda yerlerine konularak ve uygunluk değerleri hesaplanarak değerlendirilir. Yirmi dört adet kromozomun

uygunluk deęerleri izelge 1’de verilmiřtir. Kromozomların yerleřimleri Őekil 8’da topoęrafik harita üzerinde gsterilmiřtir.

**izelge 1:** Bařlangı poplasyonunun sıralanıřı (Haupt 1998)

Kromozomlar	Uygunluk deęerleri
0000000000000	-13000
11111011010010	-11800
00010110000010	-13255
11000011001010	-12347
01111111101001	-12560
01000111010001	-12700
01010110000100	-13338
11101111001110	-11890
01111100111100	-12953
00100001011110	-12891
10001110111010	-12759
10111000111100	-12320
11011011101000	-11797
00100110011101	-13778
00010100011011	-13360
01110010101011	-12220
11000011001100	-12452
10011101110000	-12335
10100000000011	-12857
00001101010110	-13166
00010000110101	-13164
01101100110010	-12927
01101111000010	-13079
10001001011111	-12756



**Őekil 8:** Bařlangı poplasyonunun topoęrafik harita üzerinde daęılımı(Haupt 1998)

### 1.4.2.3. Doğal seçim

Başlangıç popülasyonu fazla ise bazı iteratif adımların uygulanması gerekir. Kromozom olarak en uygun olanı hayatta kalmalı diğerleri de yok olmalıdır. Doğal seçim, algoritmanın her bir iterasyonunda (jenerasyonunda) meydana gelir. Burada ne kadar kromozomun elde tutulacağına karar vermek biraz keyfidir. Kromozomlarda gelecek nesil içerisinde yer alma mücadelesi başlamaktadır. Popülasyon içerisinde bütün kromozomların seçilmesi gelecek nesile çok fazla katkısı olmamaktadır. GA'da değişik seçim metotları vardır.

- a) *Rasgele seçim*: Kromozomların eşleştirilmesinde rasgele sayı üretici kullanılır. Kromozomlar 1'den başlayarak  $N_{iyi}$ 'ye kadar sıralanır. Birinci eşleştirmeyi bulmak için iki adet rasgele sayı üretilir.

$$\text{Kromozom} = \text{roundup}\{N_{iyi} \times \text{rasgele sayı}\}$$

Burada roundup fonksiyonu, sayıları en yüksek sayıya yuvarlar. Örneğin; rasgele olarak üretilen 6 sayı 0.1535, 0.6781, 0.0872, 0.1936, 0.7021 ve 0.3933 ise bu sayılar, 6 ile çarpılıp bir üst tamsayıya yuvarlanarak; 1, 5, 1, 2, 5, 3 değerleri elde edilir. Buna göre kromozom<sub>1</sub>-kromozom<sub>5</sub>, kromozom<sub>1</sub>-kromozom<sub>2</sub> ve kromozom<sub>5</sub>-kromozom<sub>3</sub> eşleştirilecektir.

- b) *Ağırlıklı seçim*: İlk olarak amaç fonksiyonunda, kromozomların uygunluk değerleri hesaplanır. Hesaplanan uygunluk değerleri en küçükten en büyüğe doğru sıraya konur. Çizelge 2 'de görüldüğü gibi  $N_{pop} \leq N_{ipop}$  ise tutulur, geriye kalanlar atılır. Burada  $N_{pop}$  değeri  $N_{ipop}$ 'a kadar olabilir. Genelde popülasyonun %50'sinin seçilmesi ( $N_{pop} = N_{ipop}/2$ ) uygun seçenektir. Seçilen  $N_{pop}$ 'un yarısı  $N_{iyi}$ , yarısı da  $N_{kötü}$  olarak ayrılır.  $N_{iyi}$  olanlar eşleştirme havuzuna konurken  $N_{kötü}$  olanlar eşleştirme havuzundan atılır. Toplam popülasyon sayısı  $N_{ipop} = 24$ 'dür. GA'nın her bir iterasyonunda popülasyonun 12'si tutulur ve bu kromozomların altı tanesi eşleştirme havuzuna atılır. Doğal seçim sonuçları Çizelge 2'de gösterilmiştir. Bunlar amaç fonksiyonunda hesaplanan uygunluk değerlerine göre sıralanmış olup ilk altı tanesi gelecek nesil için kopyalanacaktır.

**Çizelge 2:** Kromozomların uygunluk değerlerine göre sıralanışı (Haupt 1998)

Kromozomlar		Uygunluk Değerleri
N <sub>iyi</sub>	00100110011101	-13778
	00010100011011	-13360
	01010110000100	-13338
	00010110000010	-13225
	00001101010110	-13166
	00010000110101	-13164
N <sub>kötü</sub>	01101111000010	-13079
	00000000000000	-13000
	01111100111100	-12953
	01101100110010	-12927
	00100001011110	-12891
	10100000000011	-12857
N <sub>ipop</sub>	10001110111010	-12759
	10001001011111	-12756
	01000111010001	-12700
	01111111101001	-12560
	11000011001100	-12452
	11000011001010	-12347
	10011101110000	-12335
	10111000111100	-12320
	01110010101011	-12220
	11101111001110	-11890
	11111011010010	-11800
	11011011101000	-11797

- c) *Ağırlıklı rasgele seçim:* Bu yöntem, eşleştirme havuzundaki kromozomların, amaç fonksiyonunda hesaplanan uygunluk değerlerine göre seçilme ihtimallerini dikkate alır. En düşük uygunluk değerine sahip kromozom, eşleştirme ihtimali en yüksek kromozom olurken en yüksek uygunluk değerine sahip kromozom eşleştirme ihtimali en düşük olarak düşünülür. Üretilen rasgele sayı, hangi kromozomun seçileceğini belirler. Bu yöntem kendi arasında ikiye ayrılır.

*Sıra Ağırlık:* Kromozomlar büyükten küçüğe doğru sıralanır ve denklem (3.2) kullanılarak kromozomların sıralamasındaki yerlerine göre P<sub>n</sub> olasılıkları hesaplanır.

$$P_n = \frac{N_{iyi} - n + 1}{\sum_{n=1}^{N_{iyi}} n} = \frac{6 - n + 1}{1 + 2 + 3 + 4 + 5 + 6} = \frac{7 - n}{21} \quad (3.2)$$

Burada n, sıra sayısını göstermektedir. Bu yöntemle uygunluk değerlerinden bağımsız olarak, sadece sıralamayı dikkate alan  $P_n$  normalize değerleri elde edilmektedir. Bu metot uygulanarak alınan sonuç Çizelge 3 'de gösterilmiştir.

**Çizelge 3:** Eşleştirme olasılıkları (Haupt 1998)

n	Kromozom	Pn	$\sum_{i=1}^n P_i$
1	00100110011101	0.2857	0.2857
2	00010100011011	0.2381	0.5238
3	01010110000100	0.1905	0.7143
4	00010110000010	0.1429	0.8572
5	00001101010110	0.0952	0.9524
6	00010000110101	0.0476	1.0000

Kromozomların seçilmesinde kümülatif ihtimaller kullanılır. Rasgele olarak 0–1 arasında bir sayı üretilir. Bu sayı, kümülatif ihtimallerin en başından başlanarak sırayla karşılaştırılır. Kümülatif ihtimal değeri, üretilen rasgele sayıdan büyükse, eşleştirme havuzu için kromozom seçilir. Benzer tarzda 6 tane rasgele sayı üretilerek eşleştirme gerçekleştirilir. Üretilen rasgele sayılar 0.1535, 0.6781, 0.0872, 0.1936, 0.7021 ve 0.3933'dir. Bu sayılara göre kromozom<sub>1</sub>-kromozom<sub>3</sub>, kromozom<sub>1</sub>-kromozom<sub>1</sub> ve kromozom<sub>3</sub>-kromozom<sub>2</sub> eşleştirilirler. Kromozomlar kendisi ile eşleştirilirse, bu kromozomun gelecek nesilde daha etkin olarak yer aldığı söylenebilir. Bu yaklaşım tarzı doğaya daha uygundur.

*Değer ağırlık:* Kromozomların uygunluk değerleri amaç fonksiyonundan hesaplanır. Eşleştirme havuzunda  $N_{iyi+1}$ 'inci sırada bulunan kromozomun amaç fonksiyonunda hesaplanan uygunluk değeri, bütün eşleştirme havuzundaki kromozomlardan çıkartılır.  $N_{iyi+1}$ 'inci, sıradaki kromozomun uygunluk değeri – 13079'dur. Eşleştirme havuzundaki her bir kromozomun normalize uygunluk değerleri ve seçilme ihtimalleri aşağıdaki gibi hesaplanır.

$$C_n = f(\text{kromozom}_n) - f(\text{kromozom}_{N_{iyi+1}}) \quad (3.3)$$

$$P_n = \left| \frac{C_n}{\sum_{p=1}^{N_{iyi}} C_p} \right| \quad (3.4)$$

Buna göre Çizelge 4’de üstteki kromozomların daha ağırlıklı olarak sıralamada yer aldığı görülmektedir. Daha önce üretilen rasgele sayılar kullanılırsa kromozom<sub>1</sub>-kromozom<sub>3</sub>, kromozom<sub>1</sub>-kromozom<sub>1</sub> ve kromozom<sub>3</sub>-kromozom<sub>1</sub> eşleştiği görülür.

**Çizelge 4:** Eşleştirme havuzunda kromozomların sıralanışı (Haupt 1998)

<b>n</b>	<b>Kromozom</b>	<b>C<sub>n</sub></b>	<b>P<sub>n</sub></b>	$\sum_{i=1}^n P_i$
1	00100110011101	-13778+13079= -699	0.4401	0.4401
2	00010100011011	-13360+13079= -281	0.1772	0.6174
3	01010110000100	-13338+13079= -259	0.1632	0.7805
4	00010110000010	-13255+13079= -176	0.1109	0.8915
5	00001101010110	-13166+13079= -87	0.0547	0.9461
6	00010000110101	-13164+13079= -85	0.0539	1.0000

- d) *Eşik değer seçim:* Eşik değerden küçük olan kromozomlar yaşamaya devam eder, eşik değeri aşan kromozomlar ölür. Bu tekniğin avantajı, popülasyonun sıralanmasına gerek olmamasıdır.
- e) *Rulet tekerleği seçim:* Rulet tekerleği seçim metodunun uygulanması için, ilk önce kromozomların toplam uygunluk değeri hesaplanır. Kromozomlar, toplam uygunluk değerine bölünerek her bir kromozom için 0–1 arasında değişen seçim ihtimalleri bulunur. Daha sonra, kümülatif ihtimaller hesaplanır. Popülasyon sayısı kadar “rasgele” 0-1 arasında sayılar üretilir. Üretilen rasgele sayı, birinci kromozomun kümülatif seçim ihtimalinden küçük ise, birinci kromozom seçilir. Eğer değilse, ikinci kromozomun veya diğerlerinin kümülatif ihtimalleriyle karşılaştırılarak hangisinden küçükse o kromozom seçilir. Böylece Rulet seçim metodu gerçekleştirilmiş olur.
- f) *Turnuva Seçimi:* Bu yöntemde bir grup k bireyi rasgele olarak seçilir. Bu k bireyleri daha sonra bir turnuvaya katılır ve en iyi uygunluklu birey seçilir. Çaprazlama için iki turnuva tutulur. Bunlardan biri her bir ebeveyni seçmek içindir. Bir kereden daha fazla üretmek için seçilebilecek bir ebeveyn mümkündür. Turnuva seçiminin avantajı, popülasyonun daha kötü bireylerinin seçilmeyecek olması ve bundan dolayı sonraki neslin genetik

yapısına katılmayacak olmasıdır. Ayrıca en iyi bireyler tekrar üretim sürecinde baskın olmayacaktır.

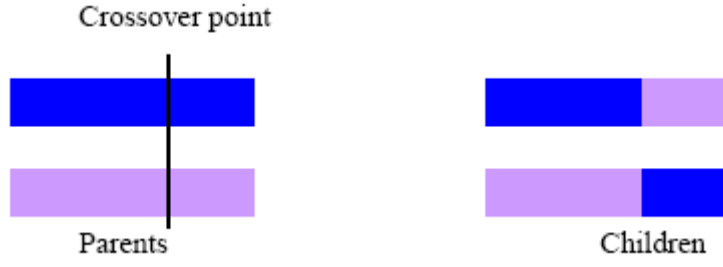
- g) *Seçkinlik(elitizm)*: Bu işlem, şu anki nesilden sonraki nesle aktarılan bir grup bireyin seçimini kapsar. Mutasyona uğratılmaksızın, sonraki nesle aktarılan bireylerin sayısı, nesil boşluğu olarak işaret edilir. Eğer nesil boşluğu sıfırsa; yeni nesil, yeni bireylerin girişleriyle oluşur. Pozitif nesil boşlukları için, diyelim ki  $k$ ,  $k$  bireyleri sonraki nesle aktarılır.

Seçilen genler sonraki yeni nesli oluşturmak için çaprazlama ve mutasyon genetik operasyonlarından geçer.

#### 1.4.2.4. Çaprazlama

İki adet yeni nesil elde etmek için kromozomların bulunduğu eşleme havuzundan iki adet kromozom seçilir. Eşleme sürecinde, seçilen kromozomlardan bir ve birden fazla yeni nesil oluşturma olayına “çaprazlama” denir. En yaygın olarak kullanılan iki kromozomdan iki tane yeni nesil elde edilmesidir. Öncelikle kromozomun ilk biti ile son biti arasında çaprazlama noktası rasgele seçilir. Birinci kromozomun çaprazlama noktasının solundaki bitlerin hepsi ikinci kromozoma geçerken, benzer tarzda ikinci kromozomun çaprazlama noktasının solundaki bitler de birinci kromozoma geçerler. Sonuçta meydana gelen yeni kromozomlar, her iki kromozomun karakterlerini içermektedirler. Çaprazlama oranı, bir popülasyonda ne kadar kromozomun çaprazlamaya tabi tutulacağını belirler. Şekil 9 ve 10 da çaprazlamanın nasıl gerçekleştiği gösterilmiştir.

$$\begin{array}{l} \text{Kromozom}_1 = \overbrace{00100110011}^{\text{yer de\u0131secek}} 101 \\ \quad \quad \quad \downarrow \uparrow \\ \text{Kromozom}_2 = \overbrace{01010110000}^{\text{yer de\u0131secek}} 100 \end{array} \quad \begin{array}{l} \text{Nesil}_1 = \mathbf{01010110000}101 \\ \text{Nesil}_2 = \mathbf{00100110011}100 \end{array}$$



Şekil 9: Tek noktalı çaprazlama



Şekil 10: Gerçek kodlu GA'da çaprazlama

Gerçek kodlu GA'da sonraki neslin reel sayılarını değiştirmedigine dikkat edin. Çünkü çaprazlama noktası her zaman reel sayılar arasındadır. (1,5,10 ve 14)

Sonuçta meydana gelen nesiller her iki kromozomun karakterlerini içermektedirler. Bu durum Çizelge 5 'de gösterilmiştir.

Çizelge 5: Eşleştirmesi yapılan kromozom çiftleri (Haupt 1998)

İşlem Sırası	6xrasgele(2-1)	Kromozom
1	2-1	1
2	5-5	5
3	6-3	3
4	4-5	4
5	1-1	1
6	4-5	4

#### 1.4.2.5. Mutasyon

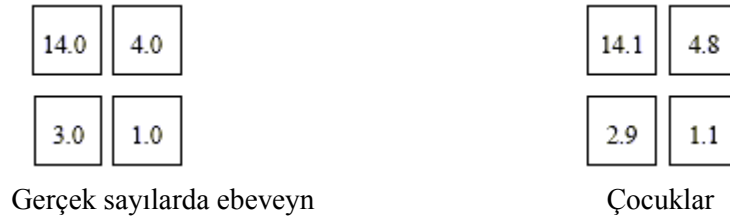
GA operatörlerinin ikincisi olan mutasyon, kromozomdaki bitlerin küçük bir yüzdesini değiştirir. Mutasyon ile kromozomdaki bitler "1" ise "0", "0" ise "1" yapılıır.  $N_{pop} \times N_{bit}$ 'lik bir popülasyon matrisinde, mutasyon noktaları rasgele seçilir. Mutasyonun % değeri artırılırsa, algoritma daha geniş parametre uzayını tarama

yapar. Son iterasyonda mutasyon meydana gelmez. Verilen örneğe, en iyi kromozom hariç tutularak %5’lik mutasyon değeri uygulanırsa, rasgele sayı üretici 7 adet kromozom için, satır ve sütunlara karşılık gelecek tam sayılar üretir. Böylece üretilen tamsayılara göre dördüncü satır ve on birinci sütundaki (4–11) bit “0” iken “1” yapılır.

00010110000010 → 00010110001010

(5-3), (2-2), (2-1), (5-14), (8-10) ve (5-8) satır ve sütun numaralarına göre altı kez meydana gelir. Çoğu mutasyonlar, kromozomların uygunluk değerini artırarak daha iyi sonuç elde edilmesine yardımcı olur.

Çaprazlamada olduğu gibi, mutasyonda gerçek kodlu ve ikili kodlu GA’da farklı etkilere sahiptir. ES mutasyonunda, gen stringlerinin değerleri, normal bir şekilde dağıtılmış küçük bir sayı ekleyerek değiştirilmesiyle iyi bir ayarlama olarak anlaşılabilir. GA da bir bitin terslenmesinin geniş bir etkisi olabilir. Şekil 11’de mutasyonun nasıl gerçekleştiği gösterilmiştir.



a) Gerçek kodlu GA da mutasyon



b) İkili kodlu GA da mutasyon

Şekil 11. (a), (b) İkili ve gerçek kodlamada mutasyon

#### 1.4.2.6. Gelecek nesil

İterasyon sonucunda, çaprazlamaya ve mutasyona uğramış kromozomların uygunluk değerleri hesaplanarak Çizelge 6'da verilen yeni nesil elde edilir. İtalik yazılan bitler, mutasyona uğrayanlardır. Çizelge 7'de görüldüğü gibi bir sonraki adımda kromozomlar uygunluk değerlerine göre yeniden sıralanır ve eşleştirme havuzuna konur. Başlangıç popülasyonun ortalama değeri, -12738 iken birinci nesilden sonra bu değer, -13334'e ulaşmıştır. Çizelge 7'deki kromozomların topoğrafik harita üzerinde yerleşimi Şekil 12'de verilmiştir.

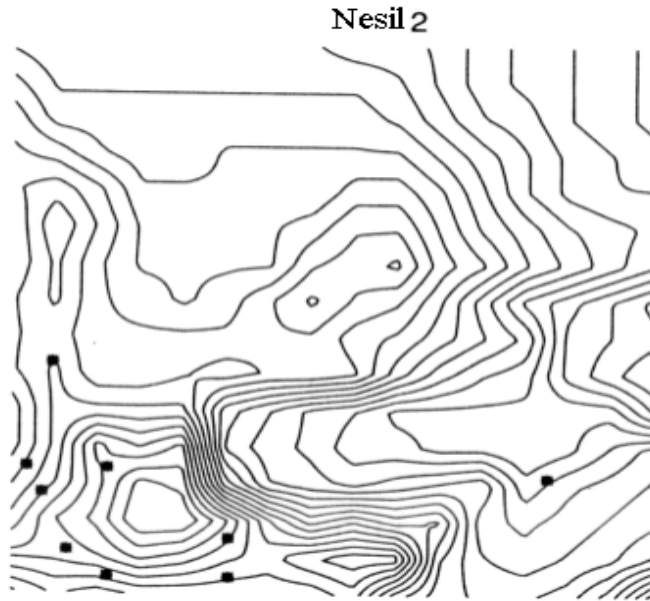
Eşleştirme havuzunda kromozomların sadece altı tanesi tutulmaktadır. Tekrar çaprazlama, mutasyon ve sıralamadan sonra elde edilen üçüncü nesil Çizelge 8'de gösterilmektedir. Üçüncü neslin sonunda popülasyon genel ortalaması -13403 olmaktadır. Dördüncü neslin sonunda ise -13676 ve yerleşimi Şekil 13' de görülmektedir.

**Çizelge 6:** Çaprazlamaya ve mutasyona uğramış yeni nesil (Haupt 1998)

<b>Kromozomlar</b>	<b>Uygunluk değerleri</b>
00100110011101	-13778
<i>11</i> 010100011011	-11956
01010110000100	-13338
0001011000 <i>1</i> 010	-13553
0000110001011 <i>1</i>	-13289
00010000110101	-13164
00100110000100	-13372
0101011000 <i>0</i> 1101	-13632
00000110011101	-13036
00100110011101	-13778
00100110011101	-13778
01010110000100	-13338

Çizelge 7: İkinci nesilden sonra sıralama (Haupt 1998)

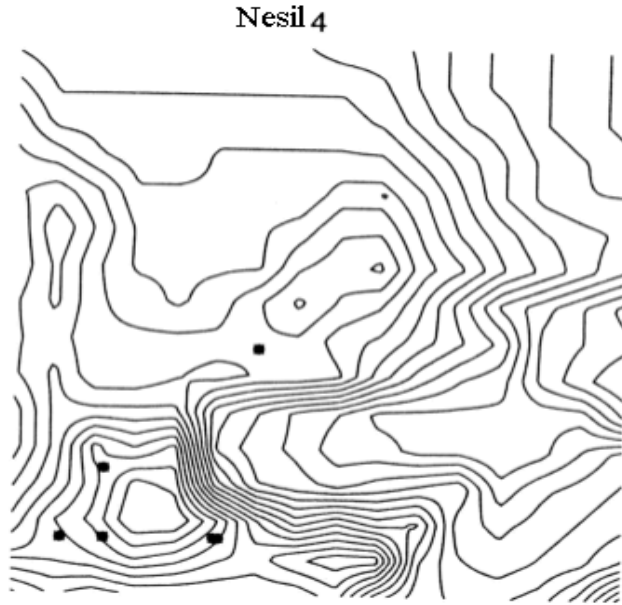
Kromozomlar	Uygunluk değerleri
00100110011101	-13778
00100110011101	-13778
00100110011101	-13778
01010110001101	-13632
00010110001010	-13552
00100110000100	-13372
01010110000100	-13338
01010110000100	-13338
00000110011101	-13036
00001100010111	-13289
00010000110101	-13164
11010100011011	-11956



Şekil 12: İkinci neslin topoğrafik harita üzerinde dağılımı (Haupt 1998)

**Çizelge 8:** Üçüncü nesilden sonra sıralama (Haupt 1998)

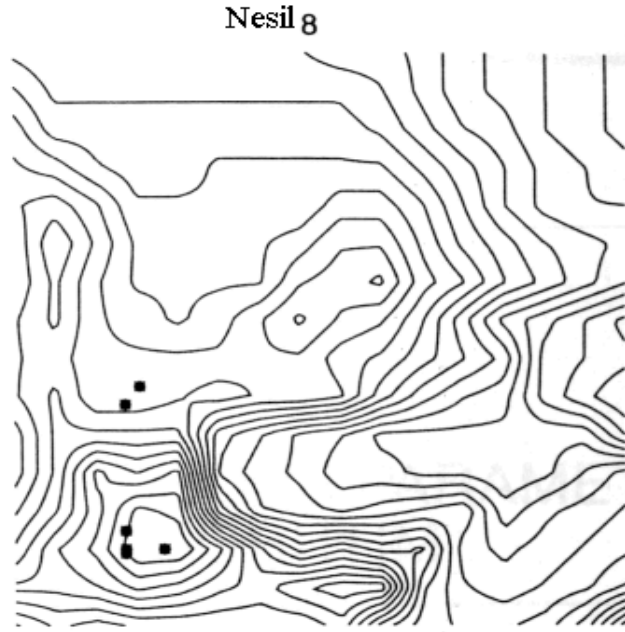
<b>Kromozomlar</b>	<b>Uygunluk Değerleri</b>
00100110011101	-13778
00100110011101	-13778
00100110011101	-13778
00100110011101	-13778
00100110011101	-13778
01010110001101	-13632
00010110001100	-13584
00010110001010	-13553
01010110001010	-13539
00100111010100	-13921
01101111011101	-13602
01100110011101	-13255



**Şekil 13:** Dördüncü neslin topoğrafik harita üzerinde dağılımı (Haupt 1998)

#### 1.4.2.7. Yakınsama

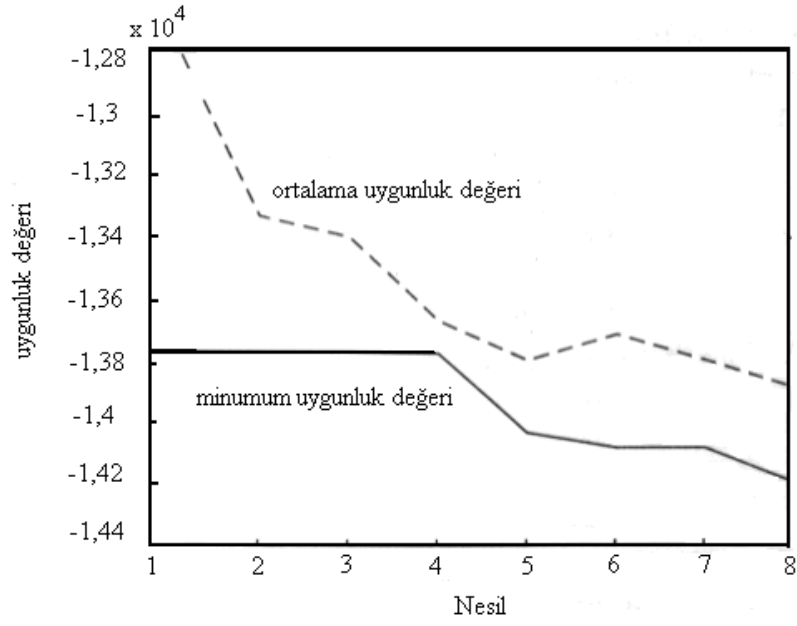
İterasyon sayısına veya uygun bir çözüme ulaşıncaya kadar işleme devam edilir. Bütün kromozomların uygunluk değerleri değişmiyorsa algoritma durdurulmalıdır. Çoğu GA'lar genel ortalama, standart sapma ve minimum uygunluk değeri gibi istatistikleri tutarlar. Bu istatistikler, yakınsama testi için bir ölçü olabilir.



Şekil 14: Sekizinci neslin topoğrafik harita üzerinde dağılımı (Haupt 1998)

Son popülasyondaki kromozomların yerleşimi Şekil 14'de gösterilmektedir. Sekizinci neslin sonunda topoğrafik harita üzerinde iki minimum noktası etrafında yoğunlaşma görülmektedir. Örnekte dokuzuncu iterasyonun sonunda global minimum -14199 olarak elde edilmiştir.

Algoritmanın minimuma yakınsaması ve her bir neslin genel ortalaması Şekil 15'de verilmektedir. Popülasyonun ortalama uygunluk değeri, algoritmanın yakınsaması hakkında bilgi vermesi bakımından önemli bir kriterdir. Örneğin maksimizasyon probleminde, farklı iki seçim metodu karşılaştırılıyor ise ortalama uygunluk değeri yüksek olan seçim metodu daha iyi sonuç vermektedir denilebilir.

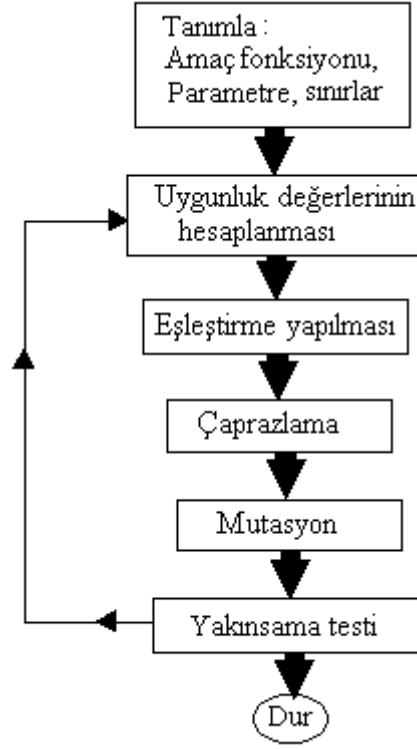


Şekil 15: İlk sekiz neslin ortalama ve minimum uygunluk değerleri (Haupt 1998)

## 1.5. Gerçek kodlu genetik algoritma

Problemlerin çözümünde rakamların hassasiyeti önemli bir yer tutmaktadır. İkili kodlu GA'da, parametrelerin "1" ve "0" larla ifade edilmesi, kromozomların boyutlarını oldukça artırdığından sınırlı hassasiyete sahiptir. Bunun yerine gerçek rakamlarla kodlama yapabilen, gerçek kodlu GA'yı kullanmak avantajlıdır. Gerçek kodlu GA, hem daha hassas hem de PC belleğinde daha az yer kaplamaktadır. Çunkaş ve Akkaya (2002), değişik test fonksiyonları kullanarak ikili kodlu GA ve gerçek kodlu GA'nın temel farklılıklarını irdelemişler. Gerçek kodlu GA'ların ikili kodlu GA'ya göre hızlı çalıştığını ve global optimumu daha kısa sürede bulduğunu göstermişlerdir.

Gerçek kodlu GA bazı literatürlerde "Sürekli Parametrelili GA" olarak da bilinmektedir. Bu çalışmada "Gerçek Kodlu GA" tanımlaması kullanılmıştır. Gerçek kodlu GA, ikili kodlu GA'ya çok benzemektedir. Aralarındaki en önemli farklılık parametrelerin "1" ve "0" lar yerine gerçek rakamlarla ifade edilmesidir. Gerçek kodlu GA'nın akış diyagramı Şekil 16'da verilmiştir.



Şekil 16: Gerçek kodlu GA'nın akış diyagramı

### 1.5.1. Amaç fonksiyonu ve parametreler

Optimizasyonda amaç, parametrelerin en uygun değerlerini bulmaktır. Optimize edilecek parametre değerleri, bir dizi haline getirilip kromozom şekline sokularak GA ile çözülür.  $N_{par}$  parametreye sahip bir kromozom  $1 \times N_{par}$  boyutlu matris biçiminde yazılır.

$$\text{Kromozom} = [P_1, P_2, P_3, \dots, P_{N_{par}}]$$

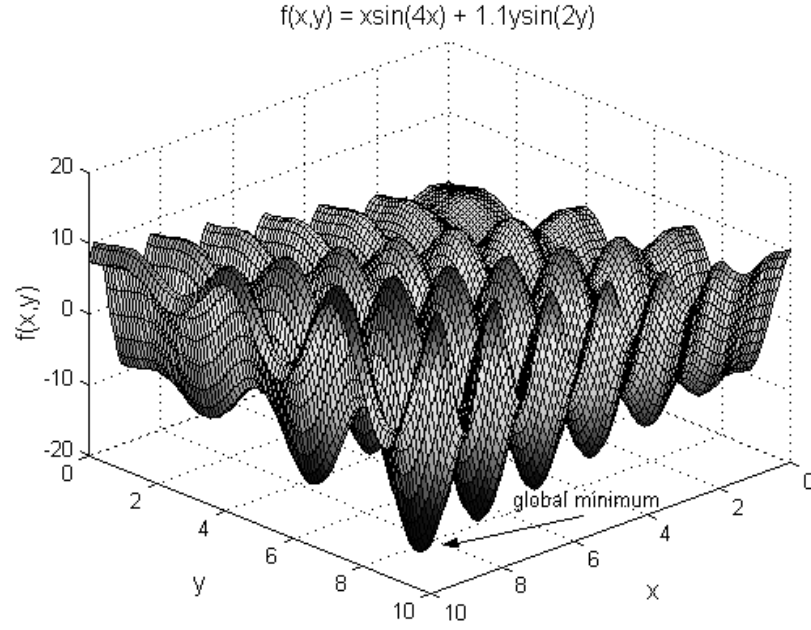
Gerçek kodlu GA'da parametrelerin hepsi “ondalık” rakamlarla ifade edilir ve her bir kromozomun bir uygunluk değeri (C) vardır.

$$C = f(\text{kromozom}) = f(P_1, P_2, P_3, \dots, P_{N_{par}})$$

Örneğin; amaç fonksiyonu

$$C = f(x, y) = x \sin(4x) + 1,1y \sin(2y)$$

ve parametre sınırları  $0 \leq x \leq 10$ ,  $0 \leq y \leq 10$  olarak tanımlanırsa Kromozom= $[x,y]$  şeklinde ifade edilebilir. Buradaki problem iki parametrelili bir optimizasyon problemi olup Şekil 17’de grafik olarak gösterilmiştir. Bu problem, Bölüm bir önceki örneğe göre daha zor bir optimizasyon problemidir ve klasik metotlarla bu fonksiyonun global minimumunu bulmak zordur. GA kullanılırsa global minimumu bulmak kolaylaşır.



Şekil 17:  $f(x, y) = x \sin(4x) + 1,1 y \sin(2y)$  ’nin üç boyutlu görünümü

### 1.5.1.1. Başlangıç Popülasyonu

GA’nın çalışmaya başlayabilmesi için,  $N_{ipop}$  kromozomdan oluşan başlangıç popülasyonuna ihtiyaç vardır. Popülasyonun kromozomları  $N_{par} \times N_{ipop}$ ’luk bir matris ile temsil edilir. Kromozomlar denklem (3.5) kullanılarak rasgele üretilir.

$$IPOP = (P_H - P_L) \times \text{rasgele}\{N_{ipop}, N_{par}\} + P_L \quad (3.5)$$

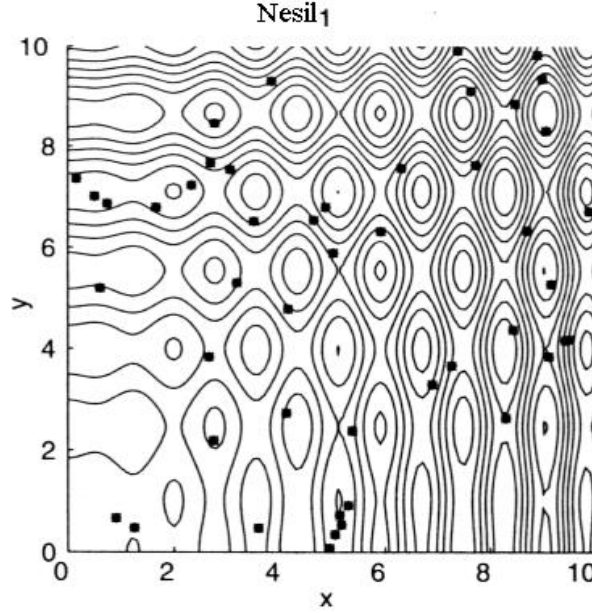
Burada;

$P_H \rightarrow$  Parametrenin üst sınır değeri

$P_L \rightarrow$  Parametrenin alt sınır değeri

$\text{rasgele}\{N_{\text{ipop}}, N_{\text{par}}\} \rightarrow N_{\text{ipop}} \times N_{\text{par}}$  olacak şekilde matris formunda 0-1 arasında üretilen rasgele sayıdır.

Bu tanımlamalar her parametre için kullanılabilir. Parametrelerin bu şekilde değerleri belirlendikten sonra amaç fonksiyonunda yerlerine konularak fertlerin uygunluk değerleri hesaplanır. Denklem (3.4)'de görüldüğü gibi x ve y den oluşan iki tane parametre vardır. Sınır değerleri  $P_L = 0$  ve  $P_H = 10$  dur. Amaç fonksiyonu biraz karmaşık olduğundan, başlangıç popülasyonunun yüksek tutulması iyi sonuç vermektedir. Başlangıç popülasyon sayısı  $N_{\text{ipop}}=48$  seçilir ve bu durumda popülasyon matrisi  $48 \times 2$ 'dir. Başlangıç popülasyonun büyüklüğü araştırma uzayının daha geniş seçilmesini sağlar. Başlangıç popülasyonunun amaç fonksiyonunda hesaplanan uygunluk değerlerinin topoğrafik harita üzerinde dağılımı Şekil 18'de gösterilmiştir.



Şekil 18: Birinci neslin topoğrafik harita üzerinde dağılımı (Haupt 1998)

### 1.5.1.2. Doğal seçim

Gelecek nesilde, hangi kromozomun yer alacağını belirlemek gerekir. İkili kodlu GA'da yapıldığı gibi kromozomlar, büyükten küçüğe doğru sıralanır. Gerçek kodlu GA'da gelecek iterasyonda kullanılmak üzere  $N_{\text{pop}}$  kadar kromozom tutulur ve geri kalanı dikkate alınmaz. Doğal seçim, en iyi kromozom uygunluk değeri

bulununcaya kadar devam etmelidir.  $N_{pop}$  kromozomları, kendi aralarında  $N_{iyi}$  ve  $N_{kötü}$  olarak büyükten küçüğe doğru sıralamaya tabi tutulur.  $N_{iyi}$ , eşleştirme havuzuna atılırken,  $N_{kötü}$  hariçte tutulur. Örnekte 48 adet kromozomun ortalama uygunluk değeri 0.9039 ve en iyi kromozomun uygunluk değeri -16.26'dır. Alt sıradan başlanarak 24 tanesi atılırsa, geri kalan kromozomların ortalama uygunluk değeri -4.27 olmaktadır. Çizelge 3.10'da gösterildiği gibi her bir iterasyonda  $N_{pop}= 24$  kromozom kullanılır ve bunun 12 tanesi  $N_{iyi}$ , 12 tanesi  $N_{kötü}$  olarak tanımlanır.

**Çizelge 10 :**Yirmidört adet kromozomun sıralanması (Haupt 1998)

No	x	y	Uygunluk Değerleri
1	9.0465	8.3097	-16.2555
2	9.1382	5.2693	-13.5290
3	7.6151	9.1032	-12.2231
4	2.7708	8.4617	-11.4863
5	8.9766	9.3469	-10.3505
6	5.9111	6.3163	-5.4305
7	4.1208	2.7271	-5.0958
8	2.7491	2.1896	-5.0251
9	3.1903	5.2970	-4.7452
10	9.0921	3.8350	-4.6841
11	0.6056	5.1942	-4.2932
12	4.1539	4.7773	-3.9545
13	8.4598	8.8471	-3.3370
14	7.2541	3.6534	-1.4709
15	3.8414	9.3044	-1.1517
16	8.6825	6.3264	-0.8886
17	1.2537	0.4746	-0.7724
18	7.7020	7.6220	-0.6458
19	5.3730	2.3777	-0.0419
20	5.0071	5.8898	0.0394
21	0.9073	0.6684	0.2900
22	8.8857	9.8255	0.3581
23	2.6932	7.6649	0.4857
24	2.6614	3.8342	1.6448

### 1.5.1.3. Seçim

Kromozomların eşleştirilmesinde ikili kodlu GA'da tanımlanan Ağırlıklı eşleştirme metodu kullanılmış ve uygunluk değerlerine göre kromozomların seçilme ihtimalleri Çizelge 3.11'de verilmiştir. Çizelgede görüldüğü gibi eşleştirme çoğunlukla yukarıdan aşağıya doğru yapılmaktadır. Çünkü sıralamanın altındaki kromozomların uygunluk değerleri küçüktür. Rasgele üretilen 6 tane sayı aşağıya sıralanmıştır. (0.4679,0.5344), (0.2872,0.4985), (0.1783,0.9554), (0.1537,0.7483), (0.5717,0.5546), (0.8024,0.8907). Üretilen bu sayılar kullanılarak eşleştirme aşağıdaki gibi gerçekleştirilir.

Anne=[3, 2, 1, 1, 4, 5 ]

Baba=[3, 3, 10, 5, 3, 7]

**Çizelge 11:** Kromozomların uygunluk değerlerine göre seçilme ihtimalleri

n	Pn	$\sum_{i=1}^n P_i$
1	0.2265	0.2265
2	0.1787	0.4052
3	0.1558	0.5611
4	0.1429	0.7040
5	0.1230	0.8269
6	0.0367	0.8637
7	0.0308	0.8945
8	0.0296	0.9241
9	0.0247	0.9488
10	0.0236	0.9724
11	0.0168	0.9892
12	0.0108	1.0000

### 1.5.1.4. Çaprazlama

Çaprazlama için çok farklı yaklaşımlar vardır. Adewuya (1996) ve Michalewicz (1994) tarafından ilginç metotlar ortaya konulmuştur. Kromozomun bir veya birden fazla yerinden bölünerek işaretleme yapılması ve işaretlenen yerlere göre

kromozomun karşılıklı genlerinin yerlerinin değiştirilmesi “tek düzen metot” olarak adlandırılır. Aşağıdaki iki adet kromozom ele alınır;

$$\text{Kromozom}_1 = [P_{a1}, P_{a2}, P_{a3}, P_{a4}, P_{a5}, P_{a6}, \dots, P_{a_{N_{\text{par}}}}]$$

$$\text{Kromozom}_2 = [P_{b1}, P_{b2}, P_{b3}, P_{b4}, P_{b5}, P_{b6}, \dots, P_{b_{N_{\text{par}}}}]$$

Çaprazlama noktaları rasgele seçilir ve parametrelerin genleri karşılıklı değiştirilir.

$$\text{Nesil}_1 = [P_{a1}, P_{a2}, P_{b3}, P_{b4}, P_{a5}, P_{a6}, \dots, P_{a_{N_{\text{par}}}}]$$

$$\text{Nesil}_2 = [P_{b1}, P_{b2}, P_{a3}, P_{a4}, P_{b5}, P_{b6}, \dots, P_{b_{N_{\text{par}}}}]$$

Burada parametrelerin değerleri değişmemektedir ve parametreler sadece gelecek nesil içerisinde farklı yerlerde yer almaktadır. Bu işlem, ikili kodlu GA için iyi bir yöntem kabul edilmesine rağmen gerçek kodlu GA için iyi sonuç vermez.

Karıştırma metotları, gerçek kodlu GA için daha iyi sonuç vermektedir. Burada iki parametrenin değerleri karıştırılır ve aşağıdaki formül kullanılarak yeni nesil elde edilir (Radcliff 1991).

$$P_{\text{yeni}} = \beta P_{\text{an}} + (1 - \beta) P_{\text{bn}} \quad (3.6)$$

$\beta = 0$  ve  $1$  arasında üretilen rasgele sayı

$P_{\text{an}}$  = Anne kromozomun n. parametresi

$P_{\text{bn}}$  = Baba kromozomun n. parametresi

İkinci nesil,  $\beta$  yerine  $1 - \beta$  konularak ve birinci neslin tümleyeni alınarak bulunur.  $\beta = 1$  ise,  $P_{\text{an}}$  baskın gelir ve  $P_{\text{bn}}$  ölür.  $\beta = 0$  ise  $P_{\text{bn}}$  baskın gelir ve  $P_{\text{an}}$  ölür.  $\beta = 0.5$  olduğu zaman (Davis, 1991), sonuç iki parametrenin ortalaması olarak ortaya çıkar. Bu metodun iyi işlediği Michalewicz (1994) tarafından gösterilmiştir. Karıştırma işlemi için hangi parametrenin seçileceği belirlenmelidir. Bazen çaprazlama noktalarının solundan veya sağından başlanarak bütün parametreler için lineer bir kombinasyon işlemi yapılır. Çaprazlama noktası rasgele seçilebilir. Parametreler, farklı  $\beta$  değerleri kullanılarak karıştırılabilir. Parametrelerin sınır aralıkları kullanılırsa, iki kromozomun bilgileri etkili olarak karıştırılır. Gerçek kodlu GA karıştırma metotlarının en basiti, “lineer çaprazlama metodu” dur (Wright, 1991). Bu metotla iki kromozomdan üç tane nesil üretilir.

$$P_{yeni1}=0.5P_{an}+0.5P_{bn} \quad (3.7)$$

$$P_{yeni2}=1.5 P_{an} -0.5 P_{bn} \quad (3.8)$$

$$P_{yeni3}= -0.5 P_{an} +1.5 P_{bn} \quad (3.9)$$

Parametre sınırları dışında kalan herhangi bir nesil dikkate alınmaz. Sonraki popülasyonda kromozomun hayatını diğer iki nesil devam ettirir. Dikkat edilecek olursa  $\beta$  pek fazla değişmemektedir.

Diğer bir çaprazlama metodu olan “Heuristic çaprazlama” da ise  $\beta$ , 0-1 arasında değer almaktadır ve yeni nesil aşağıdaki formüle göre elde edilir(Michalewicz 1994).

$$P_{yeni}=\beta(P_{an}-P_{bn})+P_{an} \quad (3.10)$$

Her parametre için farklı  $\beta$  değeri üretilerek yeni nesiller elde edilir. Denklem (3.10) göre, izin verilen bazı değerlerin dışında üretilen yeni nesiller ihmal edilir. Algoritma, yeni  $\beta$  değeri üreterek işleme devam eder.

“Karma çaprazlama metodu” (blend crossover, BLX- $\alpha$ ) bazı parametrelerin  $\alpha$  değerlerini belirleyerek işleme başlar (Eshelman ve Shafter1993).  $\alpha$  değeri, parametrelerin dış sınırlarını belirler ve bu sınırlara göre yeni nesiller üretilir.

“Kuatratik çaprazlama” gibi metotlar amaç fonksiyonu için sayısal uygunluk gerçekleştirmeye çalışırlar. Sayısal uygunluğun gerçekleştirilebilmesi için üç adet kromozoma ihtiyaç duyulmaktadır (Adewuya 1996).

Heuristik çaprazlama metoduyla extrapolasyon metodunun kombinasyonundan oluşan metot, en çok kullanılanıdır (Haupt 1998). Bu metotla, ikili kodlu GA’daki çaprazlamaya daha çok yaklaşılmaktadır. Burada ilk olarak çaprazlama noktası rasgele seçilir.

$$\alpha=\text{roundup}\{\text{rasgele} \times N_{\text{par}}\} \quad (3.11)$$

$$\text{Kromozom}_1=[P_{a1},P_{a2}\dots\dots\dots P_{a\alpha}\dots\dots\dots P_{aN_{\text{par}}}]$$

$$\text{Kromozom}_2=[P_{b1},P_{b2}\dots\dots\dots P_{b\alpha}\dots\dots\dots P_{bN_{\text{par}}}]$$

Burada m ve n indisleri anne ve baba kromozomlarını tanımlamak amacıyla kullanılmaktadır. Bu metotta, denklem (3.12) ve denklem (3.13) kullanılarak yeni nesiller elde edilir.

$$P_{yeni1} = P_{a\alpha} - \beta [P_{a\alpha} - P_{b\alpha}] \quad (3.12)$$

$$P_{yeni2} = P_{b\alpha} - \beta [P_{a\alpha} - P_{b\alpha}] \quad (3.13)$$

Burada  $\beta$ , 0-1 arasında bir değerdir. Üretilen bu yeni nesiller kromozomda yerlerine konularak işlem tamamlanır.

$$Nesil_1 = [ P_{a1}, P_{a2}, \dots, P_{yeni1}, \dots, P_{b_{N_{par}}} ]$$

$$Nesil_2 = [ P_{b1}, P_{b2}, \dots, P_{yeni2}, \dots, P_{a_{N_{par}}} ]$$

Eğer kromozomun birinci parametresi seçilirse, seçilen parametrenin sağındaki parametreler değiştirilir. Kromozomun en son parametresi seçilirse, seçilen parametrenin solundaki parametreler değiştirilir. Bu yöntemde  $\beta > 1$ 'den büyük olmadıkça sınır değerleri aşan herhangi bir parametre üretilmemektedir. Örneğin;

$$Kromozom_2 = [5.2693, 9.1382] \quad Kromozom_3 = [9.1032, 7.6151]$$

kromozomlarını ele alalım. Rasgele sayı olarak üretilen  $\alpha=1$  için çaprazlama noktası olarak 1. parametre seçilmektedir. İkinci rasgele sayı olarak  $\beta=0.7147$  üretilmektedir. Denklem (3.12) ve (3.13)' göre yeni nesil aşağıdaki gibi bulunur.

$$Nesil_3 = [5.2693 - 0.7147 \times 5.2693 + 0.7147 \times 9.1032, 7.6151] = [8.0094, 7.6151]$$

$$Nesil_4 = [9.1032 + 0.7147 \times 5.2693 - 0.7147 \times 9.1032, 9.1382] = [6.3631, 9.1382]$$

Diğer kromozomlara da aynı kurallar uygulanarak yeni nesiller elde edilir.

### 1.5.1.5. Mutasyon

Genetik algoritmalar, bazen çok hızlı yakınsarlar. Yakınsama sonucunda global maksimum bulunursa sonuç iyi, lokal minimumlar bulunursa beklenen sonuç elde edilememiş demektir. Hızlı yakınsamadan kurtulmanın yolu, araştırma uzayında, mutasyon aracılığıyla yeni çözümler elde etmektir. İkili kodlu genetik

algoritmada'da, mutasyon oranı %1-%5 arasında değiştirilirse iyi sonuçlar elde edilir (Çunkaş ve Akkaya 2004). Gerçek kodlu GA'da mutasyon oranı ise daha yüksektir. Parametrelerin toplam sayısı ile mutasyon oranı çarpılarak mutasyona girecek parametre sayısı tespit edilir. Bir matriste, satır ve sütunu ifade edecek şekilde rasgele rakamlar üretilerek mutasyon gerçekleştirilir. Örneğin; mutasyon değerini  $\mu=0.04$  olarak alınırsa, mutasyona uğrayacak parametre sayısı  $0.04 \times 24 \times 2 = 2$  olarak hesaplanır.

Çizelge 10'daki kromozom<sub>7</sub>'nin birinci parametresi ile kromozom<sub>22</sub>'nin ikinci parametresi mutasyona uğrarsa, kromozom<sub>7</sub>'nin birinci parametresi ile kromozom<sub>22</sub>'nin ikinci parametresi silinir ve yerine 0–10 arasında yeni bir rasgele sayı üretilerek konur.

$$\text{Kromozom}_7=[4.1208, 2.7271] \quad \Rightarrow \quad \text{Kromozom}_7=[8.6750, 2.7271]$$

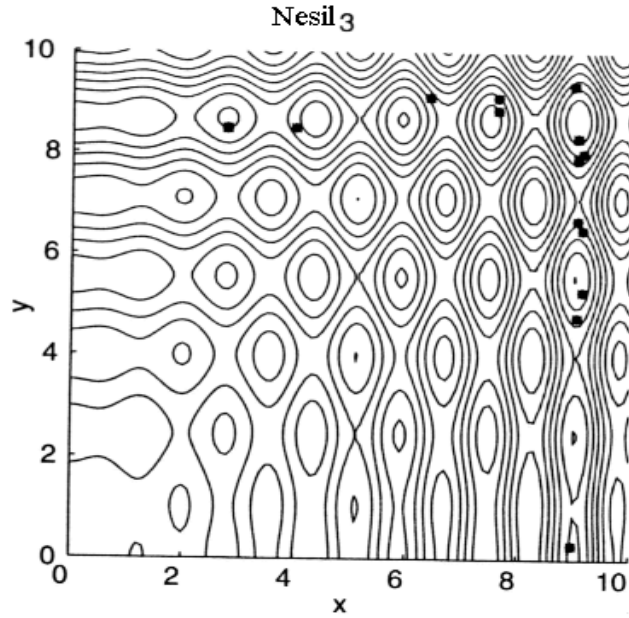
Mutasyon süreci tamamlandıktan sonra, yeni popülasyonun sıralaması Çizelge 3.12'deki gibi olur. Algoritma, yedinci iterasyondan sonra minimum uygunluk değerini -18.5 olarak bulmaktadır. Popülasyon üyelerinin topoğrafik harita üzerinde dağılımı Şekil 14, Şekil 15 ve Şekil 16'da gösterilmiştir.

Şekil-14 'de görüldüğü gibi popülasyon üyeleri üçüncü iterasyondan sonra iki minimum etrafında toplanmaya başlamaktadır. Şekil 15'de görüldüğü gibi beşinci iterasyondan sonra, bir minimum etrafında toplanmakta ve Şekil 16'daki gibi nihayet yedinci iterasyondan sonra global minimuma (-18.5) ulaşmaktadır. Şekil 18'de gerçek kodlu GA'nın yakınsama grafiği verilmiştir.

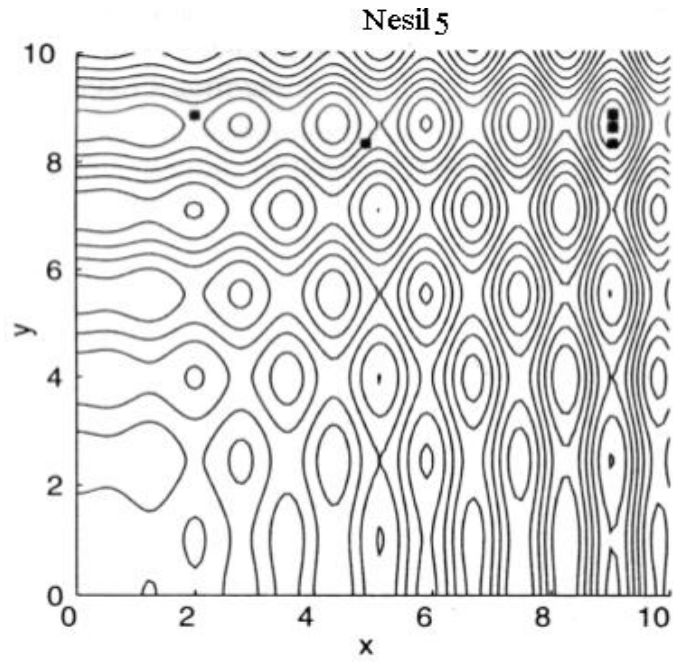
**Çizelge 12:** Gerçek kodlu GA'nın ikinci nesilden sonraki sıralanışı (Haupt 1998)

No	x	y	Uygunluk değerleri
1	9.0465	8.3128	-16.2929
2	9.0465	8.3097	-16.2555
3	9.1382	5.2693	-13.5290
4	7.6151	9.1032	-12.2231
5	7.6151	9.1032	-12.2231
6	7.6151	9.1032	-2.2231
7	2.7708	8.4789	-11.6107
8	2.7708	8.4617	-11.4863
9	9.1382	8.0094	-11.0227
10	8.9766	9.3438	-10.4131
11	8.9766	9.3469	-10.3505

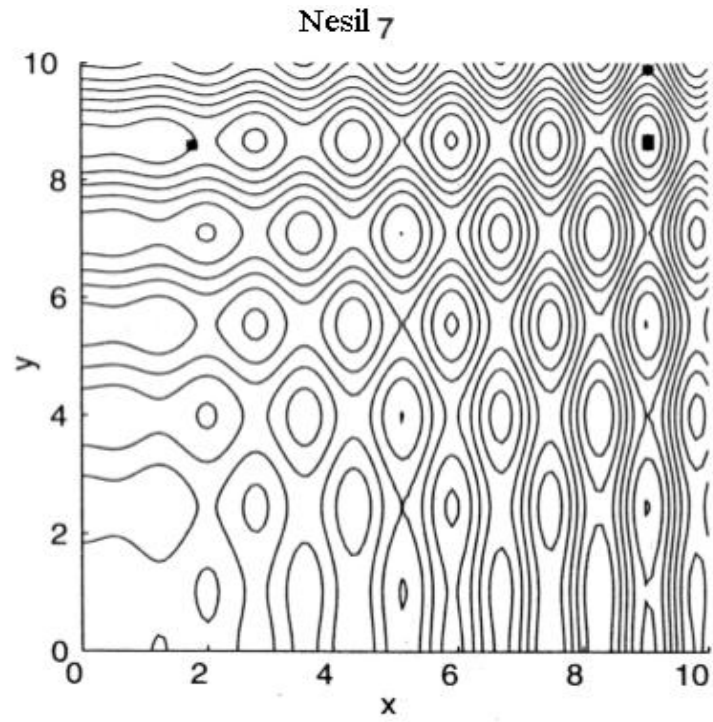
12	9.0465	7.9346	-9.8737
13	1.5034	9.0860	-6.6667
14	4.4224	9.3469	-5.6490
15	5.9111	6.3163	-5.4305
16	7.6151	6.3631	-5.1044
17	9.0921	4.2422	-5.0619
18	2.7491	2.1896	-5.0251
19	3.1903	5.2970	-4.7452
20	9.0921	3.8350	-4.6841
21	0.6956	5.1942	-4.2932
22	4.1539	4.7773	-3.9545
23	8.6750	2.7271	-3.4437
24	4.1208	3.1754	-2.6482



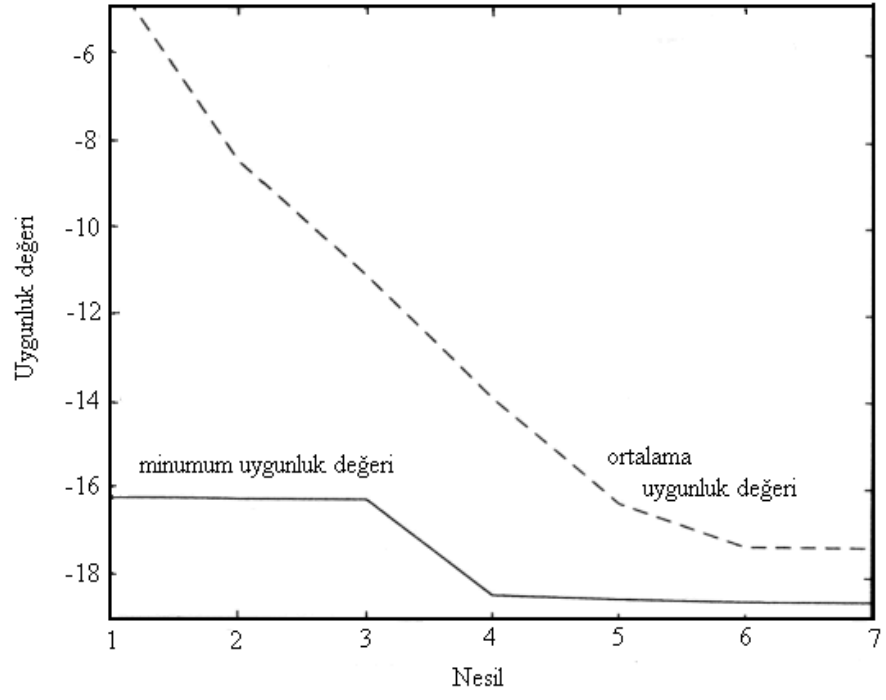
Şekil 19: Üçüncü neslin topoğrafik harita üzerinde dağılımı (Haupt 1998)



Şekil 20: Beşinci neslin topoğrafik harita üzerinde dağılımı



Şekil 21: Yedinci neslin topoğrafik harita üzerinde dağılımı



Şekil 22: Yedi tane neslin ortalama ve minimum uygunluk değerleri (Haupt 1998)

## 2. ÖRNEKLER

### 2.1. Bir Açıklayıcı Örnek – GA Nasıl Çalışır

X parametresinin 0 ve 15 arasında değişken olduğu  $(15x-x^2)$  fonksiyonun maksimum değerini bulalım. Kolaylık için biz x'i sadece integer değerlerden alacağız. Böylece, kromozomlar dört bit'le ifade edilebilir.

Integer	Binary code	Integer	Binary code	Integer	Binary code
1	0001	6	0110	11	1011
2	0010	7	0111	12	1100
3	0011	8	1000	13	1101
4	0100	9	1001	14	1110
5	0101	10	1010	15	1111

Farz edelim ki; kromozom popülasyon boyutu 6, çaprazlama olasılığı 0,7 ve mutasyon olasılığı 0.001 dir.( seçilen bu olasılıklar GAlarda yeterince tipiktir.) Uygunluk fonksiyonu şöyle tanımlanır,

$$f(x) = 15x - x^2$$

GA rasgele bir şekilde üretilen birler ve sıfırlarla altı adet 4-bit string ile doldurularak kromozomların başlangıç popülasyonunu oluşturur. Başlangıç popülasyonu Tablo 2 de gösterilen gibi görünebilir. (Gerçek pratik bir problem, tipik olarak binlerce kromozomlu bir popülasyona sahip olacaktır.)

Sonraki adım her bir birey kromozomun uygunluğunu hesaplamaktır. Bu durumda başlangıç popülasyonunun ortalama uygunluğu 36 dır. Bunu geliştirmek için, başlangıç popülasyonu genetik operatörler kullanılarak değiştirilir: seçim, çaprazlama ve mutasyon.

**Tablo 2** Rasgele bir şekilde üretilen başlangıç kromozomlarının popülasyonu

Chromosome label	Chromosome string	Decoded integer	Chromosome fitness	Fitness ratio, %	Cumulative ratio
$x_1$	1100	12	36	16.5 %	16.5
$x_2$	0100	4	44	20.2 %	36.7
$x_3$	0001	1	14	6.4 %	43.1
$x_4$	1110	14	14	6.4 %	49.5
$x_5$	0111	7	56	25.7 %	75.2
$x_6$	1001	9	54	24.8 %	100.0

Doğal seçimde, sadece en uygun türler hayatta kalabilir, üreyebilir ve sonraki nesle onların genleri ile geçilebilir. GA lar benzer bir yaklaşım kullanır. Fakat doğadan farklı bir şekilde, kromozom popülasyonunun boyutu bir nesilden sonrakine değiştirilmemiş kalır.

Tablo 2 deki son kolon, popülasyonun toplam uygunluğuna bireysel kromozomların uygunluk oranını gösterir. Bu oran, eşleşme için kromozomların seçilme şansını tanımlar. Böylece,  $x_3$  ve  $x_4$  kromozomlarının çok düşük bir seçilme olasılığı varken,  $x_5$  ve  $x_6$  kromozomlarının yeterli şansı vardır.

### **Seçim**

Eşleşme için bir kromozom seçminde, rulet tekerleği seçimi tekniğini kullanarak, [0,100] aralığında rasgele bir sayı üretilir ve kümülatif oranının rasgele sayıyı içerdiği kromozom seçilir. Bu, uygunluklarıyla orantılı olarak tekerlek üzerinde bir alana sahip her bir kromozomun yer aldığı bir tekerleğin dönmesi gibidir.

Örneğimizde, biz altı kromozomlu bir başlangıç popülasyonuna sahibiz. Böylece, sonraki nesilde aynı popülasyon boyutunu kurmak için, altı rasgele sayı üretilecektir.(bu, rulet tekerleğinin altı kez dönmesi gibidir.) İlk iki dönme,  $x_6$  ve  $x_2$  ebeveyn olmak için seçebilir; ikinci dönme çifti,  $x_1$  ve  $x_5$  kromozomlarını seçebilir ve son iki dönüş  $x_2$  ve  $x_5$  kromozomlarını seçebilir.

### **Çaprazlama**

Bir çift ebeveyn kromozom seçildikten sonra, çaprazlama operatörü uygulanır. Öncelikle çaprazlama operatörü rasgele bir şekilde, iki ebeveyn kromozomu kırarak

ve bu noktadan sonra kromozomların parçalarını değiştirecek bir çaprazlama noktası seçer. Sonuç olarak iki yeni yavru oluşturulur.

Mesela, iki kromozom  $x_6$  ve  $x_2$  ikinci genden sonra çaprazlanabilir. Her biri iki yavru üretir. Aşağıdaki gibi gösterilir:



Eğer bir çift kromozom çaprazlanmazsa, yavrular bireylerin tam kopyası olarak oluşturulur. 0.7 çaprazlama olasılığı genellikle iyi sonuç üretir.

### **Mutasyon**

Doğada nadiren olan Mutasyon, gendeki bir değişikliği temsil eder. Bu, uygunlukta önemli bir gelişmeye neden olabilir. Fakat çoğunlukla, oldukça faydalı sonuçları vardır. Mutasyonun rolü, arama algoritmasının bir local optimuma takılmamasının garantisini sağlamaktır. Seçim sırası ve çaprazlama operatörleri herhangi bir homojen çözüm kümesinde durgunlaşabilir. Böyle şartlar altında, tüm kromozomlar özdeşir ve bu yüzden popülasyonun ortalama uygunluğu geliştirilemeyebilir. Çözüm sadece optimal ( veya local olarak oldukça optimal) olmak görünebilir. Çünkü arama algoritması daha fazla ilerlemeyebilir. Mutasyon rasgele bir aramaya eşdeğerdir ve genetik farklılıkların kaybının korunmasında bize yardım eder.

Mutasyon operatörü bir kromozomda rasgele bir şekilde seçilen geni çevirir. Mesela, aşağıda gösterildiği gibi,  $x_1'$  ikinci geninde ve kromozom  $x_2$  de üçüncü geninde mutasyona uğratılmış olabilir.



Mutasyon, bazı olasılıklı kromozomda, herhangi bir gende meydana gelebilir. Mutasyon olasılığı doğada oldukça küçüktür ve GALarda oldukça küçük tutulur. Tipik olarak 0.001 ile 0.01 aralığında.

Genetik algoritmalar, popülasyonun ortalama uygunluğunun sürekli geliştirilmesini temin eder ve bir nesil sayısından sonra (tipik olarak birkaç yüz) popülasyon bir yakın-optimal çözümü geliştirir. Örneğimizde son çözüm sadece 

0	1	1	1
---	---	---	---

 ve 

1	0	0	0
---	---	---	---

 kromozomlarından oluşur.

Bu  $f(x) = 15x - x^2 = 56$  dır.

Bu örnekte, problem sadece bir değişkene sahiptir. Çözümünü göstermek kolaydır.

## 2.2. Bir GA'nın Nasıl Çalıştığını Gösteren Diğer Bir Örnek

(30x-x2) fonksiyonunun [0, 31] aralığında maksimumunu bulalım. İlk adım değişken kümesi x'i kodlamaktır. Örneğin, beş-dijit ikili string [00000 – 11111] şeklindedir. Sonra başlangıç popülasyonunu üretiriz. Bu örnekte, aşağıda gösterildiği gibi popülasyon, dört potansiyel çözüm içerir.

### Nesil 0

Chromosome label	Chromosome string	Decoded integer	Chromosome fitness	Fitness ratio, %
1	01000	8	176	24.6
2	01101	13	221	30.8
3	11010	26	104	14.5
4	10010	18	216	30.1
			717	100.0

Başlangıç popülasyonu rasgele bir sayı üretici kullanılarak üretilir. Bu popülasyondan stringler x değişkeninin değerleriyle ilgilidir ve bu değerleri kullanarak biz f(x) fonksiyonunun değerlerini değerlendiririz. Maksimizasyon problemi için, bu fonksiyon seçim işlemi yapabileceğimiz temelde uygunluk fonksiyonu olarak işleme geçirilebilir. Bu popülasyonun toplam uygunluğu 717 dir ve yukarıdaki tablonun son kolonunda her bir kromozomun uygunluğu yüzde olarak ifade edilir.

Seçim işlemi, kromozom çiftlerini bağıl kuvvetlerine göre eşleşme için değerlendirir ve seçer. Her bir kromozomun uygunluğuna dayanarak, bir seçim mekanizması, genetik manipülasyon işlemi için eşleşmeleri seçer. Seçim stratejisi, sonunda en iyi uygunluklu kromozomların(bireylerin) kurtarılmasından sorumludur. Bu örnekte, ilk string bir kopyası, ikincinin iki kopyası ve dördüncü stringin bir kopyası seçilir. [ikinci string en yüksek uygunluğa sahiptir. Eşleşme için seçilmeyen üçüncü string ise en düşük uygunluğa sahiptir.] genellikle seçim işleminde ya orantılı seçim(rulet tekerleği seçimi), ya dereceli seçim ya da turnuva seçimi uygulanır.

Genetik operasyon( veya manipülasyon işlemi) genetik operatörleri kullanır: çaprazlama (Ayrıca, iki ebeveynin gen stringlerinin alt parçalarını birleştiren yeniden

birleştirme operatörü de denir.) ve mutasyon, genetik bilgiyi işleyerek, yeni bir kromozom popülasyonu (yavrular) üretmek. Çaprazlamada, öncelikle, çaprazlamanın meydana gelip gelmeyeceğine karar verilir ve eğer yapılacaksa, rasgele sayı üreticisi tarafından tanımlanan çaprazlama noktasında stringler yer değiştirir. Aksi takdirde stringler basitçe kopyalanır. Örneğimizde, popülasyon boyutu dörtte korunur ve çaprazlama ilk çift için ikinci pozisyonda ve diğer çift için üçüncü pozisyonda meydana gelir.

### Çaprazlama

Chromosome label	Chromosome Strings (before)	Chromosome Strings (after)
1	01 000	01101
2	01 101	01000
2	011 01	01110
4	100 10	10001

Mutasyon operatörü bazı kromozomları rasgele bir şekilde değiştirerek popülasyonda yeni genetik yapıyı ortaya çıkarır. Arama algoritmasının lokal en iyiye takılmaktan kurtulmasına yardım eder. Mutasyon bazen rasgele bir string değiştirilerek gerçekleştirilir. Yukarıdaki tabloda ikinci kromozomun dördüncü biti mutasyona uğratılmıştır.

### Mutasyon

Chromosome label	Chromosome Strings (before)	Chromosome Strings (after)
1	01101	01101
2	01000	01010
2	01110	01110
4	10001	10001

Çaprazlama ve mutasyondan sonra yeni bir nesil oluşturulur ve sonlanma ölçütü istenilene ulaşıncaya kadar işlem devam eder. Sonlanma ölçütü simülasyon zamanının dolması, nesil sayısını aşması veya bir yakınsama ölçütünü sağlaması durumu olabilir.

### Nesil 1

Chromosome label	Chromosome string	Decoded integer	Chromosome fitness	Fitness ratio, %
1	01101	13	221	25.5
2	01010	10	200	23.1
3	01110	14	224	25.9
4	10001	17	221	25.5
			866	100.0

### Çaprazlama

Chromosome label	Chromosome Strings (before)	Chromosome Strings (after)
1	011 01	01110
3	011 10	01101
3	0111 0	01111
4	1000 1	10000

### Mutasyon

Chromosome label	Chromosome Strings (before)	Chromosome Strings (after)
1	0111 0	01100
3	01101	01101
3	01111	01111
4	10000	10000

### Nesil 2

Chromosome label	Chromosome string	Decoded integer	Chromosome fitness	Fitness ratio, %
1	01100	12	216	24.4
2	01101	13	221	24.9
3	01111	15	225	25.4
4	10000	16	224	25.3
			886	100.0

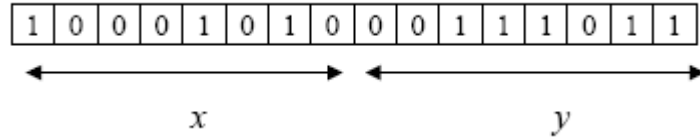
### 2.3. İki Değişken İçeren Bir Başka Örnek

Farz edelim ki, iki değişkenin “peak” fonksiyonunun maksimumunu bulunması istensin.

$$f(x,y) = (1-x)^2 e^{-x^2-(y+1)^2} - (x-x^3-y^3)e^{-x^2-y^2}$$

x ve y parametreleri -3 ve 3 arasındadır.

İlk adım kromozom olarak problem değişkenlerini temsil etmektir. x ve y parametrelerini birleştirilmiş bir ikili string olarak temsil ederiz:



Her bir parametre sekiz ikili bitlerle gösterilir. Sonra kromozom popülasyonunun boyutunu seçeriz. Örnek için 6 ve rasgele bir şekilde başlangıç popülasyonu üretilir.

Sonraki adım her bir kromozomun uygunluğunu hesaplamaktır. Bu iki seviyede yapılır. Önce, bir kromozom iki reel sayı içine dönüştürülerek kodlanır. x ve y, bunlar -3 ve 3 arasındaki aralıktadır. Sonra kodlanan x ve y peak fonksiyonunun içine koyulur.

#### Kodlama

Önce, 16 bit bir string olan bir kromozom iki 8 bit stringe parçalanır.



Sonra, bu stringler ikiliden (2 tabanından) onluğa (10 tabanına) çevrilir.

$$(10001010)_2 = (138)_{10}$$

ve

$$(00111011)_2 = (59)_{10}$$

Şimdi, 8-bit olarak ele alınabilen, 0'dan  $(2^8 - 1)$  'e, integer'ların aralığı x ve y parametrelerinin gerçek aralığına haritalanır. Bu ise -3 den 3 edir:

$$= 6/(2^8 - 1) = 0.0235294$$

x ve y nin gerçek değerlerini elde etmek için, onluk değerlerini 0.0235294 ile çarpar ve sonuçtan 3 ü çıkarırız:

$$x = (138)_{10} (0.0235294) - 3 = 0.2470588 \quad \text{ve}$$

$$y = (59)_{10} (0.0235294) - 3 = 1.6117647$$

Matematiksel fonksiyonda kodlanan x ve y değerlerini giriş olarak kullanarak, GA her bir kromozomun uygunluğunu hesaplar.

Maksimum “peak” fonksiyonunu bulmak için, 0.7 olasılıkla çaprazlama, 0.001 olasılıkla mutasyon ve 100 nesil sayısı kullanacağız. Bunun anlamı, GA durmadan önce 6 kromozomun 100 neslini oluşturacak olmasıdır.

GA’ların amacındaki en ciddi problem sonuçların kalitesiyle ilgilidir. Sonuçların özellikle en uygun çözüme ulaşıp ulaşılmadığını kontrol etmenin bir yolu farklı mutasyon oranları altında elde edilen sonuçları karşılaştırmaktır. Sabit sonuçların olumlu olması için, biz ayrıca kromozomu yani popülasyonu boyutunu artırabiliriz.

## 2.4. İki deęişken ieren dięer bir rnek

İkili kodlu bir GA'nın iřlem basamakları řoye tanımlanabilir.

- 1 Ama (Objectif) fonksiyonunu tanımlayınız.
- 2 Parametrelerin sınırlarını tanımlayınız.
- 3 Popülasyon sayısı kadar rasgele kromozom üretiniz.
- 4 Parametrelerin ikilik deęerlerini ařaęıdaki formülü kullanarak 10'luk sisteme eviriniz.

$$X_i = a_i + \frac{\text{decimal}(1001\dots\dots001)(b_i - a_i)}{2^{mi} - 1}$$

Parametreleri ama fonksiyonunda yerine koyarak uygunluk (maliyet) deęerini hesaplayınız. Burada  $m$  parametrenin bit sayısıdır.

- 5 Popülasyon toplam uygunluk deęerini bulunuz.

$$F = \sum_{i=1}^{\text{pop-size}} \text{eval}(v_i)$$

- 6 Tabii seleksiyon ihtimalini her bir kromozom iin hesaplayınız.

$$P_i = \text{eval}(v_i) / F$$

- 7 Her bir kromozom kümü latif ihtimalini hesaplayınız.

$$q_i = \sum_{j=1}^i p_j$$

- 8 Seleksiyon sūreci Roulette tekerleęinin popülasyon sayısı(pop-size) kadar dōndürölmesine dayandırılır.

- 9 Random olarak popülasyon sayısı kadar 0–1 arasında float bir sayı üretiniz.( $r$ ) Hassasiyet virgülden sonra altı rakam olacaktır.

- 10 Üretilen random sayı  $r < q_1$  ise 1. kromozomu seiniz. Aksi takdirde  $V_i$  Kromozom seilir. ( $2 \geq i \geq \text{pop-size}$ ) Üretilen random sayı  $q_{i-1} < r \leq q_i$  olursa  $V_i$  seilir.

### aprazlama ve Mutasyon:

- 1 0-1 arasında random bir sayı float olarak üretiniz.( $r$ )
- 2  $P_c$  aprazlama oranı,  $r < P_c$  ise aprazlama iin verilen kromozomu seiniz.
- 3 aprazlama iin iki kromozom seilir.

- 4 Çaprazlama noktasının belirlenmesi için Kromozom genişliğine göre bir sayı üretilir. (1.....m-1). m: kromozom genişliği
- 5 Üretilen bu sayıyla çaprazlama noktası belirlenmiş olur.  
 B1b2 bpos bpos+1.....bm  
 C1c2 cpos cpos+1.....cm  
 yeni nesil  
 b1b2 bpos cpos+1.....cm  
 c1c2 cpos bpos+1.....bm
- 6  $P_c \times \text{pop-size} = \text{Çaprazlamaya maruz kalacak kromozom sayısını verir.}$   
 $P_m = \text{mutasyon yüzdesi}$   
 $P_m \times \text{pop-size} = \text{Mutasyona maruz kalacak bit sayısını verir.}$
- 7 0–1 arasında rasgele bir sayı üretiniz.(r)
- 8 Eğer  $r < P_m$  ise biti mutasyona uğratınız.

### Uygulama:

Amaç fonksiyonu:

$$F(x_1, x_2) = 21.5 + x_1 \sin(4\pi x_1) + x_2 \sin(20\pi x_2) \text{ olarak tanımlansın.}$$

Parametrelerin tanım aralığı:  $-3 \leq x_1 \leq 12.1$  ve  $4.1 \leq x_2 \leq 5.8$

Toplam popülasyon: pop-size=20

Çaprazlama ve mutasyon oranları:  $P_c=0.25$   $P_m=0.01$

Amaç: Yukarıdaki iki değişkenli fonksiyonun maksimizasyonudur.

Her bir değişken için hassasiyet 4 decimal olarak kabul edilsin.

- $X_1 = [-3, 12.1]$  in toplam genişliği  $(-3) - (12.1) = 15.1$  ve  $15.1 \times 10000$  eşit değere bölünmesi gerekir.  
 Bunun anlamı  $2^{17} < 151000 \leq 2^{18}$  dir. Buna göre  $x_1$  i ifade etmek için 18 bit gereklidir.
- $X_2 = [4.1, 5.8]$  nin toplam genişliği  $5.8 - 4.1 = 1.7$  ve  $1.7 \times 10000$  eşit değere bölünebilmesi gerekir.  
 Bunun anlamı  $2^{14} < 17000 \leq 2^{15}$  dir. Buna göre  $X_2$  yi ifade etmek için 15 bit gereklidir.
- Buna göre iki parametrelili kromozom toplam genişliği  $18 + 15 = 33$  bit dir. İlk 18 biti  $X_1$  geri kalanı da  $X_2$  yi temsil etmek amacıyla kullanılır.  
 Rasgele üretilen bir kromozom ele alarak işleme devam edelim.

$$\left[ \begin{array}{c} 010001001011010000111110010100010 \\ \underbrace{\hspace{10em}}_{X_1} \quad \underbrace{\hspace{10em}}_{X_2} \end{array} \right]$$

▪ İlk 18 biti ele alırsak;

$$\text{▪ } X_1 = -3 + \text{desimal}(010001001011010000)_2 \frac{12.1 - (-3)}{2^{18} - 1},$$

$$\text{▪ } X_1 = -3 + 70352 \times 15.1 / 262143 = 1.052426$$

Aynı yöntemle  $X_2$  şöyle bulunur.

$$\text{▪ } X_2 = 4.1 + \text{desimal}(111110010100010)_2 \frac{5.8 - (-4.1)}{2^{15} - 1} = 5.755330$$

Buna göre kromozomdaki genlerin değeri şöyledir.

$$\text{▪ } [X_1, X_2] = [1.052426, 5.755330]$$

$$\text{▪ } \text{Kromozomun uygunluk değeri } f(X_1, X_2) = f(1.052426, 5.755330) = 20,252640$$

Örneğimizde popülasyon sayısı 20 olduğundan, başlangıçta 1 ve 0 ‘lardan oluşan rasgele olarak 20 adet kromozom üretilir. Biz yukarıda sadece 1 kromozom için hesaplamamın nasıl yapılacağını gösterdik. Geriye kalan 19 adet kromozomun uygunluk değerleri de yukarıdaki gibi hesaplanmalıdır. Rasgele üretilen kromozomlar;

$$V1 = (100110100000001111111010011011111)$$

$$V2 = (111000100100110111001010100011010)$$

$$V3 = (000010000011001000001010111011101)$$

$$V4 = (100011000101101001111000001110010)$$

$$V5 = (000111011001010011010111111000101)$$

$$V6 = (000101000010010101001010111111011)$$

$$V7 = (001000100000110101111011011111011)$$

$$V8 = (100001100001110100010110101100111)$$

$$V9 = (010000000101100010110000001111100)$$

$$V10 = (000001111000110000011010000111011)$$

$$V11 = (011001111110110101100001101111000)$$

$$V12 = (110100010111101101000101010000000)$$

$$V13 = (111011111010001000110000001000110)$$

$$V14 = (010010011000001010100111100101001)$$

V15= (111011101101110000100011111011110)  
V16= (110011110000011111100001101001011)  
V17= (011010111111001111010001101111101)  
V18= (011101000000001110100111110101101)  
V19= (000101010011111111110000110001100)  
V20= (101110010110011110011000101111110)

- Uygunluk değerleri aşağıdaki gibi hesaplanmıştır.

Eval(V<sub>1</sub>) = f(6.084492,5.652242) = 26.019600  
Eval(V<sub>2</sub>) = f(10.348434,4.380264) = 7.580015  
Eval(V<sub>3</sub>) = f(-2.516603,4.390381) = 19.526329  
Eval(V<sub>4</sub>) = f(5.278638,5.593460) = 17.406725  
Eval(V<sub>5</sub>) = f(-1.255173,4.734458) = 25.341160  
Eval(V<sub>6</sub>) = f(-1.811725,4.391937) = 18.100417  
Eval(V<sub>7</sub>) = f(-0.991471,5.680258) = 16.020812  
Eval(V<sub>8</sub>) = f(4.910618,4.703018) = 17.959701  
Eval(V<sub>9</sub>) = f(0.795406,5.381472) = 16.127799  
Eval(V<sub>10</sub>) = f(-2.554851,4.793707) = 21.278435  
Eval(V<sub>11</sub>) = f(3.130078,4.996097) = 23.410669  
Eval(V<sub>12</sub>) = f(9.356179,4.239457) = 15.011619  
Eval(V<sub>13</sub>) = f(11.134646,5.378671) = 27.316702  
Eval(V<sub>14</sub>) = f(1.335944,5.151378) = 19.876294  
Eval(V<sub>15</sub>) = f(11.089025,5.054515) = 30.060205  
Eval(V<sub>16</sub>) = f(9.211598,4.993762) = 23.867227  
Eval(V<sub>17</sub>) = f(3.367514,4.571343) = 13.696165  
Eval(V<sub>18</sub>) = f(3.843020,5.158226) = 15.414128  
Eval(V<sub>19</sub>) = f(-1.746635,5.395584) = 20.095903  
Eval(V<sub>20</sub>) = f(7.935998,4.757338) = 13.666916

Yukarıdaki kromozomlar içerisinde uygunluk değeri en zayıf kromozom V<sub>2</sub>, en güçlü olanı da V<sub>15</sub> dir.

- Şimdi bir seleksiyon süreci için rolette tekerleğini uygulayalım.
- Popülasyonun toplam uygunluğu;  $F = \sum_{i=1}^{20} eval(v_i) = 387.776822$
- Her bir kromozomun seleksiyon ihtimali P<sub>i</sub> (i.....20)

P <sub>1</sub> = eval(V <sub>1</sub> )/F = 0.067099	P <sub>2</sub> = eval(V <sub>2</sub> )/F = 0.019547
P <sub>3</sub> = eval(V <sub>3</sub> )/F = 0.050355	P <sub>4</sub> = eval(V <sub>4</sub> )/F = 0.044889
P <sub>5</sub> = eval(V <sub>5</sub> )/F = 0.065350	P <sub>6</sub> = eval(V <sub>6</sub> )/F = 0.046677
P <sub>7</sub> = eval(V <sub>7</sub> )/F = 0.041315	P <sub>8</sub> = eval(V <sub>8</sub> )/F = 0.046315
P <sub>9</sub> = eval(V <sub>9</sub> )/F = 0.041590	P <sub>10</sub> = eval(V <sub>10</sub> )/F = 0.054873
P <sub>11</sub> = eval(V <sub>11</sub> )/F = 0.060377	P <sub>12</sub> = eval(V <sub>11</sub> )/F = 0.038712
P <sub>13</sub> = eval(V <sub>13</sub> )/F = 0.070444	P <sub>14</sub> = eval(V <sub>14</sub> )/F = 0.051257
P <sub>15</sub> = eval(V <sub>15</sub> )/F = 0.077519	P <sub>16</sub> = eval(V <sub>16</sub> )/F = 0.061549
P <sub>17</sub> = eval(V <sub>17</sub> )/F = 0.035320	P <sub>18</sub> = eval(V <sub>18</sub> )/F = 0.039750
P <sub>19</sub> = eval(V <sub>19</sub> )/F = 0.051823	P <sub>20</sub> = eval(V <sub>20</sub> )/F = 0.035244

- Her bir kromozomun kümülatif ihtimali  $q_i$  ise;  $q_i = \sum_{j=1}^i p_j$  kullanılarak aşağıdaki gibi hesaplanır.

$q_1=0.067099$ ,  $q_2=0.086647$ ,  $q_3=0.137001$ ,  $q_4=0.181890$ ,  $q_5=0.247240$ ,  
 $q_6=0.293917$ ,  $q_7=0.335232$ ,  $q_8=0.381546$ ,  $q_9=0.423137$ ,  $q_{10}=0.478009$ ,  
 $q_{11}=0.538381$ ,  $q_{12}=0.577093$ ,  $q_{13}=0.647537$ ,  $q_{14}=0.698794$ ,  $q_{15}=0.776314$ ,  
 $q_{16}=0.837863$ ,  $q_{17}=0.873182$ ,  $q_{18}=0.912932$ ,  $q_{19}=0.964756$ ,  $q_{20}=1.000000$

- Rulette tekerleğini 20 kez döndürelim. Yeni popülasyon için her döndürüşümüzde bir kromozom seçeceğiz. 20 kez döndürmek demek 0-1 arasında 20 adet sayı üretmek anlamındadır.

- Random olarak 0-1 arasında 20 tane sayı üretildi.

$r_1=0.513870$ ,  $r_2=0.175741$ ,  $r_3=0.308652$ ,  $r_4=0.534534$ ,  $r_5=0.947628$ ,  $r_6=0.171736$ ,  
 $r_7=0.702231$ ,  $r_8=0.226431$ ,  $r_9=0.494773$ ,  $r_{10}=0.424720$ ,  $r_{11}=0.703899$ ,  
 $r_{12}=0.389647$ ,  $r_{13}=0.277226$ ,  $r_{14}=0.368071$ ,  $r_{15}=0.983437$ ,  $r_{16}=0.005398$ ,  
 $r_{17}=0.765682$ ,  $r_{18}=0.646473$ ,  $r_{19}=0.767139$ ,  $r_{20}=0.780237$ .

ilk sayı  $r_1=0.513870$   $q_{10}$  dan büyük  $q_{11}$  den daha küçüktür. Bunun anlamı  $v_{11}$  kromozomu yeni popülasyon için seçilir. İkinci sayı  $r_2 = 0.175741$   $q_3$  ten büyük  $q_4$  ten küçüktür yani  $v_4$  kromozomu yeni popülasyon için seçilir.

- Sonuç olarak yeni popülasyon aşağıdaki kromozomları içerir:

$V^1=(011001111110110101100001101111000)$  (v11)  
 $V^2=(100011000101101001111000001110010)$  (v4)  
 $V^3=(001000100000110101111011011111011)$  (v7)  
 $V^4=(011001111110110101100001101111000)$  (v11)  
 $V^5=(00010101001111111110000110001100)$  (v19)  
 $V^6=(100011000101101001111000001110010)$  (v4)  
 $V^7=(111011101101110000100011111011110)$  (v15)  
 $V^8=(00011101100101001101011111000101)$  (v5)  
 $V^9=(011001111110110101100001101111000)$  (v11)  
 $V^{10}=(000010000011001000001010111011101)$  (v3)  
 $V^{11}=(111011101101110000100011111011110)$  (v15)  
 $V^{12}=(010000000101100010110000001111100)$  (v9)  
 $V^{13}=(00010100001001010100101011111011)$  (v6)  
 $V^{14}=(100001100001110100010110101100111)$  (v8)  
 $V^{15}=(101110010110011110011000101111110)$  (v20)

V'16=(100110100000001111111010000111011) (v1)  
V'17=(000001111000110000011010000111011) (v10)  
V'18=(111011111010001000110000001000110) (v13)  
V'19=(111011101101110000100011111011110) (v15)  
V'20=(110011110000011111100001101001011) (v16)

### Şimdi çaprazlama operatörünü uygulayalım

- Çaprazlama operatörü  $P_c=0.25$  verildiğinden ortalama olarak kromozomların %25'i çaprazlamaya maruz kalacaktır. Pop-size=20 olduğundan çaprazlamaya maruz kalan popülasyon sayısı 5 dir.
- Yeniden random olarak 0-1 arasında 20 tane sayı üretilir. Eğer üretilen  $r<0.25$  ise çaprazlama için o kromozomu seçeriz. Rasgele üretilen sayılar aşağıdaki gibi olsun.

r1=0.822951	r2= 0.151932	0.625447	0.314685	0.346901
0.917204	0.519760	0.401254	0.606758	0.785402
0.031523	0.869921	0.166525	0.674520	0.758400
0.581893	0.389248	0.200232	0.355635	r20= 0.826927

- $P_c=0.25$  olduğuna göre; yukarıda üretilen rasgele sayılardan 4 tanesi 0.25 altında olduğu için bunlar çaprazlamaya tabii tutulacak kromozomlardır. (V2',V11',V13',V18')
- Şans eseri 4 tane kromozom seçildi. Yani çift sayılıdır. Eğer sayı tek olsaydı ya bir tane kromozom ilave edecektik Veya bir tane kromozom kaldıracaktık. Bu seçimde rasgele olarak yapılır.
- Rasgele eleştirelim (V2',V11'), (V13',V18')
- Bu iki çift için rasgele popülasyonun toplam uzunluğu(1-32) içinde kalacak şekilde bir tam sayı üretilir.(kromozom toplam uzunluğu=33).
- *Pos* sayısı çaprazlama (crossing) yapılacak pozisyonu belirtmektedir. İlk kromozom çifti için;

V2' = (100011000101101001111000001110010)  
V11'=(111011101101110000100011111011110)

ve pos sayısı 9 olsun, kromozomlar 9. pozisyonundan itibaren kesilir ve yer değiştirilir. Ortaya çıkan kromozomlar şöyledir:

$$V2'' = (10001100 \text{ 1101110000100011111011110})$$
$$V11'' = (111011100101101001111000001110010)$$

İkinci çift kromozom;

$$V13' = (000101000010010101000000001000110)$$
$$V18' = (111011111010001000111010111111011)$$

İkinci çift için üretilen pozitif tam sayı 20 dir. 20. bitten sonraki bitler karşılıklı olarak yer değiştirirler.

$$V13'' = (0001010001001010101101011111011)$$
$$V18'' = (1110111110100010001010101111011)$$

- Çaprazlamadan sonra ortaya yeni nesil çıkar.(V13'',V2'',V18'',V11'').
- Popülasyonun yeni versiyonu aşağıdaki gibi dizilirler.

$$V^1=(011001111110110101100001101111000)$$
$$V''2=(10001100 \text{ 1101110000100011111011110})$$
$$V^3=(00100010000011010111101101111011)$$
$$V^4=(011001111110110101100001101111000)$$
$$V^5=(000101010011111111110000110001100)$$
$$V^6=(100011000101101001111000001110010)$$
$$V^7=(111011101101110000100011111011110)$$
$$V^8=(00011101100101001101011111000101)$$
$$V^9=(011001111110110101100001101111000)$$
$$V^10=(000010000011001000001010111011101)$$
$$V''11=(111011100101101001111000001110010)$$
$$V^12=(010000000101100010110000001111100)$$
$$V''13=(0001010001001010101101011111011)$$
$$V^14=(100001100001110100010110101100111)$$
$$V^15=(101110010110011110011000101111110)$$
$$V^16=(100110100000001111111010000111011)$$
$$V^17=(000001111000110000011010000111011)$$
$$V''18=(1110111110100010001010101111011)$$
$$V^19=(111011101101110000100011111011110)$$
$$V^20=(110011110000011111100001101001011)$$

Şimdi de Mutasyon operatörünü uygulayalım.

- Her bir generasyonda; mutasyon operatörü  $P_m=0.01$  olduğundan, kromozomun toplam bit sayısı  $20 \times 33=660$  den, ancak 5 veya 6 bitin ( $660 \times 0.01=6.6$  olduğu için) mutasyona maruz kalacağı tahmin edilir. Mutasyon için her bir bit eşit şansa sahiptir.

- 0–1 arası kromozomun toplam bit sayısı=660 kadar rasgele sayı üretilir.(r).
- Kromozomda eğer  $r_i < P_m$  (0.01) ise mutasyon gerçekleştirilir.
- Rastgele üretilen 660 tane sayının 5 tanesi 0.01 den küçüktür.

Bit pozisyonu	Kromozom numarası	Kromozom içindeki bit numarası
112	4	13
349	11	19
418	13	22
429	13	33
602	19	8

Bit pozisyonu	$r_i$
112	0.000213
349	0.009945
418	0.008809
429	0.005425
602	0.002836

Aşağıdaki tabloda da hangi kromozomun kaçınıcı biti mutasyona uğradığı mavi renk ile verilmiştir.

$V^1=(011001111110110101100001101111000)$   
 $V^2=(10001100 \text{ 1101110000100011111011110})$   
 $V^3=(00100010000011010111101101111011)$   
 $V^4=(011001111110110101100001101111000)$   
 $V^5=(000101010011111111110000110001100)$   
 $V^6=(100011000101101001111000001110010)$   
 $V^7=(111011101101110000100011111011110)$   
 $V^8=(00011101100101001101011111000101)$   
 $V^9=(011001111110110101100001101111000)$   
 $V^{10}=(000010000011001000001010111011101)$   
 $V^{11}=(11101110010110100111000001110010)$   
 $V^{12}=(010000000101100010110000001111100)$   
 $V^{13}=(0001010001001010101101011111011)$   
 $V^{14}=(100001100001110100010110101100111)$

$V^{*15}=(101110010110011110011000101111110)$   
 $V^{*16}=(100110100000001111111010000111011)$   
 $V^{*17}=(000001111000110000011010000111011)$   
 $V^{*18}=(111011111010001000010101011111011)$   
 $V^{*19}=(1110111011011110000100011111011110)$   
 $V^{*20}=(110011110000011111100001101001011)$

Yukarıdaki tabloya göre 13. kromozomda iki defa mutasyon gerçekleştirilir. Buraya kadar sadece 1 iterasyon/generasyon tamamlandı. İkinci iterasyon için, popülasyonun uygunluk değerleri amaç fonksiyonunda yerine konularak hesaplanır. Yeni uygunluk değerleri aşağıdaki gibidir.

$eval(v_1)=f(3.130078,4.996097)=23.410669$   
 $eval(v_2)=f(5.279042,5.054515)=182010839$   
 $eval(v_3)=f(-0.991471,5.680258)=16.020812$   
 $eval(v_4)=f(3.128235,4.996097)=23.412613$   
 $eval(v_5)=f(-1.746635,5.395584)=20.095903$   
 $eval(v_6)=f(5.278638,5.593460)=17.406725$   
 $eval(v_7)=f(11.089025,5.054515)=30.060205$   
 $eval(v_8)=f(-1.255173,4.734458)=25.341160$   
 $eval(v_9)=f(3.130078,4.996097)=23.410669$   
 $eval(v_{10})=f(-2.516603,4.390381)=19.526329$   
 $eval(v_{11})=f(11.088621,4.743434)=33.351874$   
 $eval(12)=f(-0.795406,5.381472)=016.127799$   
 $eval(13)=f(-1.811725,4.209937)=22.692463$   
 $eval(14)=f(4.910618,4.703018)=17.959701$   
 $eval(15)=f(7.935998,4.757338)=013.666916$   
 $eval(16)=f(6.084492,5.652242)=26.019600$   
 $eval(17)=f(-2.554851,4.793707)=21.278435$   
 $eval(18)=f(11.134646,5.666975)=27.591064$   
 $eval(19)=f(11.059532,5.054515)=27.608441$   
 $eval(20)=f(9.211598,4.993762)=23.867227$

Yeni popülasyonun toplam uygunluğu  $F=447.049688$  dir. Önceki popülasyonun uygunluk değeri  $387.776822$  idi. Aynı zamanda önceki popülasyondaki en iyi kromozom  $v_{15}$   $30.060205$  dan daha iyi kromozom  $v_{11}$   $33.351874$  olarak hesaplanmıştır.

Şu anda seçim işlemi tekrar çalıştırılır ve genetik operatörler uygulanır, bir sonraki jenerasyonun uygunluğu hesaplanır. 1000 jenerasyondan sonra popülasyon;

$V_1=(1110111101100110111001010111011)$

$V_2=(111001100110000100010101010111000)$   
 $V_3=(111011110111011011100101010111011)$   
 $V_4=(11100110001000110000101010111001)$   
 $V_5=(1110111101110110111001010110111011)$   
 $V_6=(111001100110000100000100010100001)$   
 $V_7=(11010110001001001000110010110000)$   
 $V_8=(111101100010001010001101010010001)$   
 $V_9=(1110011000100100110010001100010110001)$   
 $V_{10}=(111011110111011011100101010111001)$   
 $V_{11}=(110110110000010010001100010110000)$   
 $V_{12}=(110101100010010010001100010110001)$   
 $V_{13}=(1110111011101101110010101011101)$   
 $V_{14}=(11100110011000010000101010111011)$   
 $V_{15}=(1110011010101111001010100110110001)$   
 $V_{16}=(111001100110000101000100010100001)$   
 $V_{17}=(111001100110000100000101010111011)$   
 $V_{18}=(111001100110000100000101010111001)$   
 $V_{19}=(11110110001000101000111000001001)$   
 $V_{20}=(11100110011000010000101010111001)$

Uygunluk deęerleri:

$Eval(v_1) = f(11.120940,5.092514) = 30.298543$   
 $Eval(v_2) = f(10.588756,4.667358) = 26.869724$   
 $Eval(v_3) = f(11.124627,5.092514) = 30.316575$   
 $Eval(v_4) = f(10.574125,4.242410) = 31.933120$   
 $Eval(v_5) = f(11.124627,5.092514) = 30.316575$   
 $Eval(v_6) = f(10.588756,4.214603) = 34.356125$   
 $Eval(v_7) = f(9.631066,4.427881) = 35.458636$   
 $Eval(v_8) = f(11.518106,4.452835) = 23.309078$   
 $Eval(v_9) = f(10.574816,4.427933) = 34.393820$   
 $Eval(v_{10}) = f(11.124627,5.092514) = 30.316575$   
 $Eval(v_{11}) = f(9.623693,4.427881) = 35.477938$   
 $Eval(v_{12}) = f(9.631066,4.427933) = 35.456066$   
 $Eval(v_{13}) = f(11.124627,5.092514) = 30.316575$   
 $Eval(v_{14}) = f(10.588756,4.242514) = 32.932098$   
 $Eval(v_{15}) = f(10.606555,4.653714) = 30.746768$   
 $Eval(v_{16}) = f(10.588814,4.214603) = 34.359545$   
 $Eval(v_{17}) = f(10.588756,4.242514) = 32.932098$   
 $Eval(v_{18}) = f(10.588756,4.242410) = 32.956664$   
 $Eval(v_{19}) = f(11.518106,4.472757) = 19.669670$   
 $Eval(v_{20}) = f(10.588756,4.242410) = 32.956664$

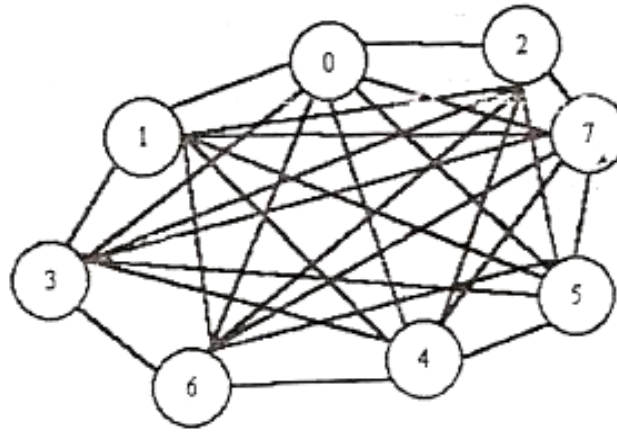
Bununla beraber eęer geliřmeye dikkatlice bakılırsa 1000 ninci jenerasyonda en iyi kromozomun 35,477938 deęerinden daha iyi kromozomların uygunluk deęerlerini bulunduran önceki jenerasyonlar meydana gelmiř olabilir. Örneęin 396.jenerasyonundaki 38.827553 deęerlerine sahip kromozom en iyisidir.

Gelişim sürecindeki sadece bir tane en iyi saklanabilir. Genetik algoritma uygulamalarında da alışılmış olarak daima en iyiyi saklanır. Son popülasyondaki en iyi değerinin tersine tüm sürecin sonundaki en iyi değeri rapor edilir.

- **DİKKAT:** En son iterasyonda mutasyon operatörü uygulanmamalıdır.

## 2.5. Gezgin Satıcı Probleminin Genetik Algoritmalarla Uygulaması

GA'ların uygulanmasına ilişkin incelenecek olan GSP'de 8 şehir olduğunu varsayalım. Hatırlatalım ki problemde amaç, satıcının her şehre yalnızca bir kere uğrayarak turunu en kısa yoldan tamamlamasıdır. Satıcının başladığı şehre geri dönme zorunluluğunun olmadığını varsayalım. (Yani burada amaç döngü değil, Hamilton yolunun bulunmasıdır.) Şehirler birbirine bağlı olmakta ve uzaklıklar verilmektedir bu şartlar içerisinde toplam alınan yolun minimum değerini bulmaya çalışacağız.[20].



Şekil 23: 8 Şehirli GSP Şeması

### 2.5.1. Başlangıç popülasyonun oluşturulması:

GSP'de 8 şehir olduğunu kabul etmiştik. Şehirleri "0" dan "7" ye kadar numaralandırdığımızı düşünürsek, her gen bir şehre karşılık gelecektir. Bu durumda kromozomlarımız ve başlangıç popülasyonumuz rastlantısal olarak aşağıdaki gibi oluşturulabilir.(Popülasyon büyüklüğü "5" olarak seçilmiştir.)

10275463

71302564

24170536

41653720

60534217

### 2.5.2. Popülasyon içindeki her kromozomun amaç fonksiyonunun değerinin hesaplanması:

İlk kromozomumuza bakarsak, bu kromozomun amaç fonksiyonu değeri, şartta verilen şehirlerarası yol uzunluklarına bağlı olarak aşağıdaki şekilde hesaplanır:

10275463

1'den 0'a 5 km

2'den 7'ye km

5'den 4'e 3 km

6'dan 3'e 5 km

0'dan 2'ye 3 km

7'den 5'e 4 km

4'den 6'ya 6 km

İse; bu kromozomun gösterdiği yolun toplam uzunluğu 27 km, yani amaç fonksiyonun değeri "27" olacaktır. Artık popülasyonumuzun GA operatörlerinin uygulanması için hazırdır. Bu operatörleri uygulayarak popülasyonumuzun her yeni nesilde evrilmesini ve daha iyi sonuçlar elde edilmesini sağlayabiliriz.

### 2.5.3. Tekrar üretme, çaprazlama ve mutasyon operatörlerinin uygulanması:

Goldberg probleminde uygulanan çaprazlama işlemini GSP'nin kromozomlarına uygularsak bazı güçlüklerle karşılaşabiliriz. Örneğin, rastgele bir şekilde üçüncü ve dördüncü kromozomları seçelim ve çaprazlama noktası 5 olsun. Bu durumda çaprazlama şu şekilde olacaktır.

Çaprazlama öncesi durum;

2 4 1 7 0 **5 3 6**

4 1 6 5 3 **7 2 0**

Çaprazlama sonrası durum;

2 4 1 7 0 **7 2 0**

4 1 6 5 3 **5 3 6**

Görüldüğü gibi yeni oluşan kromozomlar, GSP problemi için olumlu bir sonuç vermez. Çünkü satıcımız, tüm şehirleri gezmeli ve her şehre yalnızca bir kez uğramalıydı, fakat çaprazlama sonrası kromozomlarda bazı şehirlere hiç gidilmemekte yada şehirlere uygun gelen düğümlere iki kere başvurulmaktadır. Örneğin çaprazlama sonrası oluşan ilk kromozomumuza dikkat edecek olursak, 3., 5. ve 6. şehirlere hiç gidilmediğini ve 0., 2. ve 7. şehirlere iki defa uğrandığını gözlemleriz. Böyle bir olumsuz durumda ikinci bir işleme gerek duyulur. Yani kısıtlamaları sağlamayan uygunsuz kromozomların standartlaştırılması gerekmektedir. Standartlaştırma için aşağıdaki basit kural uygulanabilir: "Kromozom içinde tekrar eden ilk şehir ziyaret edilmeyen en küçük numaralı şehir ile değiştirilsin". Bu

standartlaştırma kuralı akla ilk gelen kurallardan biridir. Herhangi başka bir kural da standartlaştırma amacı için uygulanabilirdi. Evrimsel programlama böyledir; hemen hemen her şey plansız programsızdır. Tıpkı doğadaki evrim gibi... Sorunlu kromozomların standartlaştırılması ile şu sonucu elde ederiz:

Çaprazlama ve standartlaştırma sonrası durum;

34156720

41027536

Böyle bir durumda ikili düzende kodlanmış olan kromozomların daha avantajlı olabileceği görülmektedir. Ancak ikili düzende kodlama da kromozomların çok uzun olması, ya da problemin yapısına uygun olmaması gibi sorunlar çıkabilmektedir.

Şimdi de Goldberg problemindeki gibi mutasyon operatörüne geçelim. Ancak bu rasgele seçilen kromozomlar üzerindeki iki genin yerlerini değiştirmekle gerçekleştirilir. Rastgele olarak seçtiğimiz 5. kromozomumuzun ikinci ve yedinci genlerini seçtiğimizi varsayalım. Bu durumda; mutasyondan önceki durum mutasyondan sonraki durum

60534217

67534210

*4. ve 5. Yeni kromozomların amaç fonksiyonu değerlerinin bulunması ve kötü kromozomların popülasyona çıkartılması:*

Yeni oluşturulan fonksiyonların değerleri 2. adımda yapılanlara benzer olarak hesaplanır. Seçici bir model söz konusu ise, bu adımdan sonra popülasyonun içindeki en kötü değerli kromozom, popülasyondan çıkartılır.

#### **2.5.4 (2-3) arasındaki adımların tekrarı.**

Konunun başında optimizasyon problemlerinde, optimuma her zaman ulaşamadığını ya da tüm olası sonuçların incelenmesini, bilgisayarlarla dahi mümkün olmadığını söylemiştik. Burada anlatmaya çalıştığımız GENETİK Algoritmalar, Evrim Kuramını temel olarak, bu tür problemler için optimum sonuçları ya da optimumlara yakın sonuçları elde etmek için kullanılmaktadır. Örneğin bu yöntemle 200–300 şehir için bile yaklaşık 20 dk içinde optimuma çok yakın sonuçlar elde edilebilmektedir.

## 2.6 Gerçek kodlama kullanılarak yapılan bir uygulama

Aşağıda verilen problemde amaç fonksiyonu ile birlikte kısıt fonksiyonları da verilmiştir. Bu örnekte penaltı fonksiyonu kullanılarak çözüm yapılacaktır.

### 2.6.1 Problemin tanımlanması

Amaç fonksiyonu:

$$f(x) = 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141$$

Kısıt fonksiyonları:

$$g_1(x) \equiv 85.334407 + 0.0056858x_2x_5 + 0.0006262x_1x_4 - 0.0022053x_3x_5 \geq 0$$

$$g_2(x) \equiv 92.0 - (85.334407 + 0.0056858x_2x_5 + 0.0006262x_1x_4 - 0.0022053x_3x_5) \geq 0$$

$$g_3(x) \equiv 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 + 0.0021813x_3^2 - 90.0 \geq 0$$

$$g_4(x) \equiv 110.0 - (80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 + 0.0021813x_3^2) \geq 0$$

$$g_5(x) \equiv 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 - 0.0019085x_3x_4 - 20 \geq 0$$

$$g_6(x) \equiv 25 - (9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 - 0.0019085x_3x_4) \geq 0$$

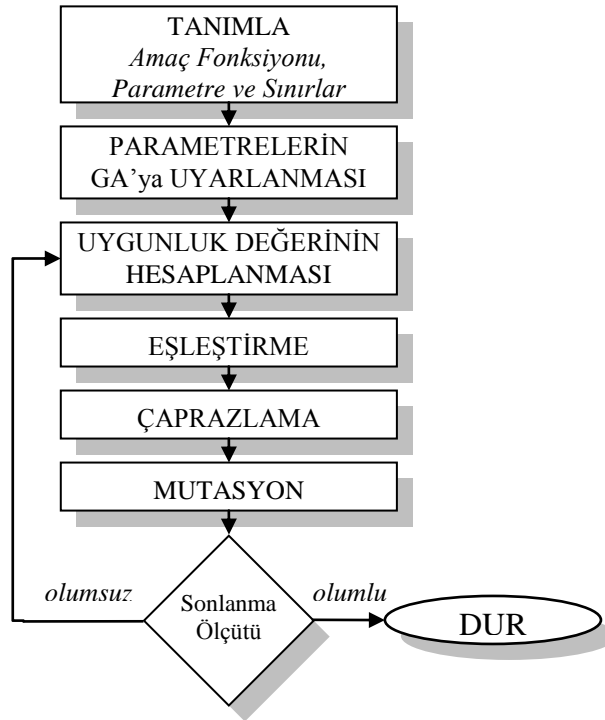
Karar Değişkenlerinin (parametrelerin) alt ve üst sınırları:

$$78 \leq x_1 \leq 102 \quad 33 \leq x_2 \leq 45 \quad 27 \leq x_3 \leq 45 \quad 27 \leq x_4 \leq 45 \quad 27 \leq x_5 \leq 45$$

Yukarıda amaç fonksiyonu, sınır fonksiyonları ve aralıkları verilen problemin minimize edilmesi istenilmektedir.

## 2.6.2 Çözüm adımları

Bilindiği gibi genetik algoritmalarda çözüme ulaşırken problemin türüne göre kromozomların gösterimi, ikili kodlu veya gerçek kodlu olmak üzere iki şekilde seçilebilir. Bu örnek için değişkenlerin (parametrelerin) gösteriminde gerçek kodlu genetik algoritma kullanılmıştır. Tek kriterli bir genetik algoritmanın işlem adımları aşağıdaki akış diyagramında görülmektedir (bkz şekil-1).



Şekil-1: Genetik Algoritma Akış Diyagramı

### 1.Adım

Buna göre; amaç fonksiyonu, kısıt fonksiyonları ve aralıklar tanımlı olduğundan ve popülasyonun gösterim şekli seçildiğinden, yapılması gereken ilk işlem başlangıç popülasyonunu oluşturmaktır. Aralıkları problemde belirtilen parametrelerin (genlerin) ilk değerleri rasgele olarak bu değerler arasında üretilir ve başlangıç popülasyonu oluşturulmuş olunur.

Problemde verilen aralıklara uygun olarak programda üretilen bir başlangıç popülasyonu aşağıda verilmiştir.(Şekil-2 )

No	X1	X2	X3	X4	X5	G1(x)	G2(x)	G3(x)	G4(x)	G5(x)	G6(x)	violated	F(x)
1	78,9016426	37,5093249	27,5157629	27,8746018	44,0881481	93,439052	-1,439052	12,8231251	7,176874949	-1,00682459	6,006824595	-27801,5018	---
2	82,2043581	41,7773139	32,2786403	30,5454061	43,3910457	94,1250201	-2,12502011	16,0006572	3,999342761	2,152184345	2,847815655	-24647,511	---
3	81,1992084	41,4222327	34,3811635	42,228625	43,8332173	94,4816698	-2,48166977	16,1149609	3,885039115	6,09376669	-1,09376669	-21101,2163	---
4	83,5275605	39,991065	36,3604798	27,3047626	30,7598439	91,29029	0,709710045	12,1752287	7,824771323	1,780919947	3,219080053	0	-28446,4595
5	81,2038588	39,5429571	38,8843823	27,1603661	27,6662068	90,5633685	1,436631497	11,2313832	8,768616764	2,345304601	2,654695399	0	-27785,2741
6	78,4614999	40,0169407	40,9917562	40,9573664	32,1780798	91,7593029	0,240697068	12,7662835	7,233716485	8,434312724	-3,43431272	-14958,7234	---
7	93,475184	41,3920067	42,4394935	42,8084784	27,095164	91,6810364	0,31896358	14,0295849	5,970415096	11,96682636	-6,96682636	22997,18083	---
8	99,4576479	37,316405	41,967001	27,9218148	35,8081526	91,3568951	0,643104923	15,0013659	4,99863413	8,079808636	-3,07980864	-15185,1933	---
9	96,9419587	38,0358566	39,5186384	42,7116903	40,1601727	93,1124558	-1,11245583	15,8581558	4,141844232	12,24874772	-7,24874772	28226,03351	---
10	85,5559514	43,3079966	31,042282	36,5746174	28,9349692	92,4380446	-0,43804457	12,6503825	7,349617496	0,74540615	4,25459385	-30177,8484	---
11	80,2241412	40,3584521	34,8253343	31,6664242	32,2588493	91,8501641	0,149835863	12,141459	7,858541014	1,961778271	3,038221729	0	-29139,5838
12	89,8157283	41,5864034	27,3553968	27,208395	43,7420232	94,568747	-2,56874702	16,3064127	3,693587296	-0,69301537	5,693015371	-23071,3225	---
13	92,1704767	44,4494765	36,3212739	43,3673263	31,7846577	93,3244796	-1,3244796	15,7382363	4,261763689	7,35716846	-2,35716846	-20527,8062	---
14	88,4151951	43,1292371	36,5726852	42,1334611	27,7399815	92,232333	-0,23233303	13,3851885	6,614811484	6,003499806	-1,00349981	-27217,7613	---
15	98,8947504	42,0130643	35,438665	36,4966101	34,5533748	93,1481561	-1,14815609	16,0509402	3,949059835	5,913982315	-0,91398232	-25365,8123	---
16	95,1321455	43,1971982	41,172692	32,0001934	39,9992432	93,433097	-1,43309703	18,8425884	1,157411642	9,40305868	-4,40305868	-3541,10185	---
17	81,8455066	37,322086	31,4171388	39,0223972	30,5099739	91,6949068	0,305093155	9,93653952	10,06346048	1,357306579	3,642693421	0	-30364,6554
18	78,7582798	36,1751171	37,7730693	37,5496291	36,2333678	91,6206634	0,379336569	11,5070779	8,492922123	5,841273173	-0,84127317	-27117,8491	---
19	82,2848411	37,0794738	41,758606	29,1797916	43,180295	91,9650123	0,034987726	14,8742894	5,125710582	7,806411479	-2,80641148	-17535,3278	---
20	98,0430058	43,821674	41,1513121	37,9270326	32,3487765	92,7872983	-0,78729827	17,1859997	2,814000282	10,63977205	-5,63977205	7014,669626	---
21	89,2974773	36,9262065	33,9088209	35,8346359	31,2639064	91,5643378	0,435662201	11,1312237	8,86877633	3,096772832	1,903227168	0	-28968,3771
22	86,1980125	34,7326754	43,3287243	42,1130493	37,4097402	91,420734	0,579266018	12,8423113	7,157688653	13,24264237	-8,24264237	43117,13662	---
23	93,4213318	44,137942	40,3552035	32,2091517	40,8578933	93,8361717	-1,83617168	19,2777391	0,722260945	8,789023435	-3,78902343	-7664,6138	---
24	83,5263046	41,5047173	42,2988767	39,8573524	31,2159478	91,8738099	0,126190106	14,039763	5,960236995	9,773618727	-4,77361873	-3124,55584	---
25	82,0649608	35,9471908	44,0230561	32,9000933	42,5263506	91,5883809	0,411619075	14,4789031	5,521096868	10,63477837	-5,63477837	7319,225667	---
26	90,7358028	44,1739936	33,2588735	38,316315	37,5211205	94,1834493	-2,18344934	16,7522838	3,247716242	4,378321605	0,621678395	-23869,1417	---
27	97,4887974	33,7629902	30,2456594	44,6514169	31,5261353	92,0095066	-0,00950663	9,95877101	10,04122899	3,320627688	1,679372312	-29686,5681	---
28	97,9930936	44,7162054	40,5825029	34,7584908	42,3614645	94,4463873	-2,44638731	20,7400582	-0,74005823	10,81814781	-5,81814781	15538,79743	---
29	99,2929289	37,289301	40,66002	27,1398889	42,6673689	92,2423266	-0,24232662	16,5565206	3,443479433	8,127555974	-3,12755597	-14850,599	---
30	84,0089571	40,1760048	36,5494543	33,3218983	39,0355666	92,8579887	-0,85798867	14,7212477	5,27875226	4,964956905	0,035043095	-27025,1662	---
.													---
.													---
.													---
.													---
100													---

## 2.Adım

Bu örnekte, elde edilen popülasyondan doğal seçim işlemini yapmak için turnuva seçimi tercih edilmiştir. Turnuva seçiminde, eşleşecek çiftler (popülasyon boyutu kadar) rasgele seçilir. Çiftlerden popülasyona alınacak olan bireylerin (kromozomların) belirlenmesinde kısıt fonksiyonlarından elde edilen bir penalty (cezalandırma –  $\psi(x)$ ) fonksiyonundan faydalanılır.

Penalty fonksiyonu, problemde verilen  $g_1(x), g_2(x), g_3(x), g_4(x), g_5(x), g_6(x)$  kısıt fonksiyonlarının istenen aralık dışında olması durumunda, üretilen kromozomların olması gereken değerlere yakınlığını değerlendiren bir fonksiyondur. Buna göre çiftlerden penalty fonksiyonu sonucu küçük olan birey popülasyona alınır. Aşağıda, kullanılan penalty fonksiyonu görülmektedir.

$$\phi(x, r) = f(x) + \sum_{k=1}^K r_{kl} G_k [g_k(x)]^2 \quad (1)$$

Turnuva seçinde, seçim sürecini üç maddeyle özetleyebiliriz:

- Eğer kromozom çiftinin her ikisi de uygun aralıkta değilse, penalty fonksiyonu sonucu küçük olan birey alınır. Yani uygun aralığa daha yakın olan birey tercih edilmiş olur. Ayrıca her iki bireyin de uygunlukları hesaplanmaz.
- Eğer kromozomlardan yalnız biri uygun aralıkta ise, yani penalty fonksiyonu sonucu sıfır ise, bu kromozom popülasyona alınır. Diğer kromozomun uygunluğu hesaplanmaz.
- Eğer her iki kromozom da uygun aralıkta ise, kromozomların uygunluk değerleri amaç fonksiyonu yardımıyla hesaplanır ve daha uygun olan birey popülasyona alınır. Bu örnek için minimizasyon yapıldığından dolayı uygunluk değeri daha küçük olan birey seçilir.

Bu anlatılanlara göre program çıktısının devamı olarak turnuva seçimi sonucu elde edilen popülasyon aşağıdaki tablodaki gibidir.(Şekil-3)

### Şekil-3: Turnuva Seçimi Sonuçları

No	birinci	ikinci	index	$\phi$ (birinci)	$\phi$ (ikinci)	F(x)
1	7	38	38	22997,18	16715,489	---
2	38	26	26	16715,489	23869,141	---
3	14	1	1	-27217,761	-27801,501	---
4	16	6	6	-3541,101	-14958,7234	---
5	46	21	21	19,29748	0	-28968,4
6	14	37	14	27217,761	152150,793	---
7	34	12	12	-15210,681	-23071,322	---
8	34	27	27	15210,681	-29686,568	---
9	45	24	45	-18192,094	-3124,555	---
10	45	4	4	-18192,094	0	-28446,5
11	17	46	17	0	-3732,209	-30364,7
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
100						

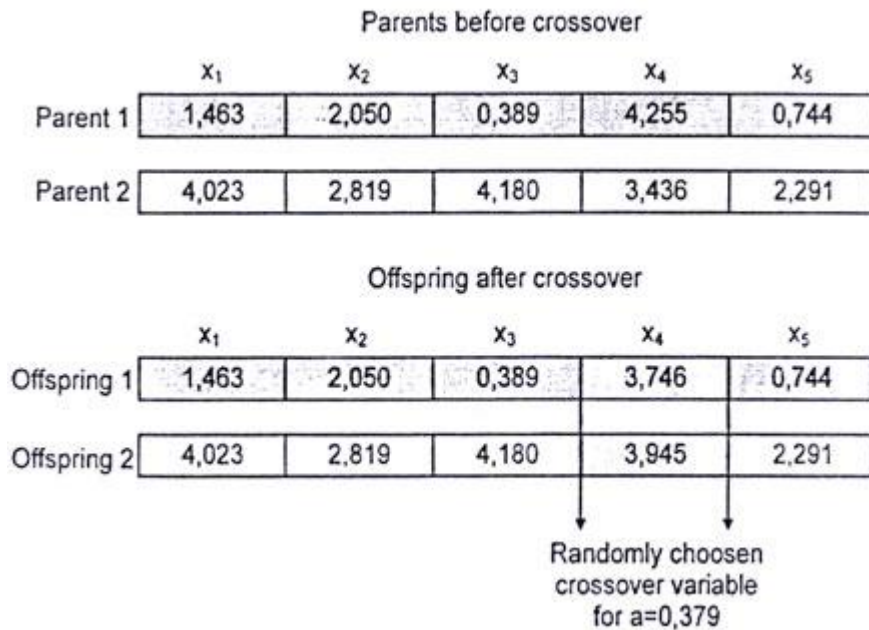
### 3.Adım

Seçim işleminden sonra sıra çaprazlama işlemindedir. Gerçek kodlu genetik alırtmada da ikili kodludaki gibi çeşitli çaprazlama yöntemleri mevcuttur. Tek düzen metodu, lineer çaprazlama, heuristic çaprazlama, karma çaprazlama, kuadratik çaprazlama vb. Bu problem için tercih edilen metot, aritmetiksel çaprazlamadır.

Aritmetiksel çaprazlamada eşleşen kromozomların, karşılıklı her bir geni için rasgele [0,1] aralığında bir sayı üretilir. Bu sayı çaprazlama oranından küçük ise genler çaprazlama işlemine alınabilir. Daha sonra her bir gen için  $\rho$  adı verilen [0,1] aralığında rasgele üretilen bir katsayı ile kromozomlar (2) deki formüle göre çaprazlanır.

$$x_n^{1,t+1} = \rho \cdot x_n^{2,t} + (1 - \rho) \cdot x_n^{1,t} \quad \text{ve} \quad x_n^{2,t+1} = \rho \cdot x_n^{1,t} + (1 - \rho) \cdot x_n^{2,t} \quad (2)$$

Aşağıdaki şemada buna rasgele bir örnek verilmiştir.(şekil-4)



Şekil-3: Aritmetiksel Çaprazlama Örneği

Programdan alınan çaprazlama sonuçları aşağıda görölmektedir. (Şekil-5)

**Çaprazlama Sonuçları**

1	0,51353943	0,65316118	0,98497348	0,10465259	0,14408535	96,2439517	44,9003504	-1	42,351361	27,4738135	0,46922669	0,89382909	-1	0,5642038	0,6110214	93,3
2	-1	-1	-1	-1	-1	90,7358028	44,1739936	-1	38,316315	37,5211205	-1	-1	-1	-1	-1	93,6
3	0,66986518	0,39501403	0,99237996	0,20909799	0,05305935	78,9016426	37,5093249	-1	27,8746018	44,0881481	0,20102997	0,31442377	-1	0,34362779	0,97511112	78,5
4	-1	-1	-1	-1	-1	78,4614999	40,0169407	-1	40,9573664	32,1780798	-1	-1	-1	-1	-1	78,8
5	0,91796166	0,76382698	0,50723307	0,62588287	0,07856239	-1	-1	33,9088209	35,8346359	31,2639064	-1	-1	0,67850336	0,66589213	0,25702837	-1
6	-1	-1	-1	-1	-1	-1	-1	36,5726852	42,1334611	27,7399815	-1	-1	-1	-1	-1	-1
7	0,13939484	0,23583775	0,11343052	0,15529161	0,62551065	89,8157283	41,5864034	27,3553968	27,208395	43,7420232	0,56592907	0,94612352	0,72674981	0,90938412	0,39884408	93,1
8	-1	-1	-1	-1	-1	97,4887974	33,7629902	30,2456594	44,6514169	31,5261353	-1	-1	-1	-1	-1	94,1
9	0,18128151	0,21475358	0,0047471	0,91391722	0,69817821	92,0946506	44,2844527	43,3524256	-1	35,7345066	0,40674803	0,89879839	0,36403574	-1	0,6398909	87,0
10	-1	-1	-1	-1	-1	83,5275605	39,991065	36,3604798	-1	30,7598439	-1	-1	-1	-1	-1	88,6
11	0,93730959	0,13824992	0,2050839	0,70539971	0,38659602	-1	37,322086	31,4171388	-1	30,5099739	-1	0,2410118	0,66842942	-1	0,30647012	-1
12	-1	-1	-1	-1	-1	-1	44,0437203	29,2742175	-1	34,9739456	-1	-1	-1	-1	-1	-1
13	0,45462358	0,48564141	0,47162657	0,96519222	0,20099325	97,9930936	44,7162054	40,5825029	-1	42,3614645	0,43876046	0,88705698	0,10393612	-1	0,26563015	93,1
14	-1	-1	-1	-1	-1	89,2974773	36,9262065	33,9088209	-1	31,2639064	-1	-1	-1	-1	-1	94,1
15	0,02382917	0,71811714	0,7938296	0,63035643	0,65214497	78,2346437	-1	-1	41,3459867	27,1101784	0,29971387	-1	-1	0,8650286	0,14319732	79,6
16	-1	-1	-1	-1	-1	80,2241412	-1	-1	31,6664242	32,2588493	-1	-1	-1	-1	-1	78,8
17	0,76458613	0,96771393	0,31971877	0,42767914	0,74103962	-1	-1	44,0230561	32,9000933	-1	-1	-1	0,06567059	0,99482595	-1	-1
18	-1	-1	-1	-1	-1	-1	-1	29,2742175	40,0464366	-1	-1	-1	-1	-1	-1	-1
19	0,0213501	0,5227881	0,9442205	0,92110662	0,42717182	78,7582798	36,1751171	-1	-1	36,2333678	0,1590225	0,31904053	-1	-1	0,90081098	83,2
20	-1	-1	-1	-1	-1	84,0938938	44,4497154	-1	-1	33,2462512	-1	-1	-1	-1	-1	79,6
21	0,39884352	0,05433648	0,61606173	0,35560711	0,44718219	78,9016426	37,5093249	27,5157629	27,8746018	44,0881481	0,46502672	0,43254806	0,63759572	0,48403466	0,69387027	79,6
22	-1	-1	-1	-1	-1	80,3034989	43,8967322	37,4781501	36,8156023	32,1453368	-1	-1	-1	-1	-1	79,5
23	0,03807299	0,34412413	0,54188399	0,47104654	0,56733821	95,5909693	44,7951168	39,5364205	29,9868689	28,9741685	0,84037111	0,2433987	0,08407044	0,04696993	0,40595598	94,4
24	-1	-1	-1	-1	-1	88,4151951	43,1292371	36,5726852	42,1334611	27,7399815	-1	-1	-1	-1	-1	89,5
25	0,67625698	0,70785088	0,24428005	0,77692577	0,70999477	97,4887974	-1	30,2456594	-1	-1	0,97036627	-1	0,86480667	-1	-1	97,2
26	-1	-1	-1	-1	-1	89,2974773	-1	33,9088209	-1	-1	-1	-1	-1	-1	-1	89,5
27	0,59010701	0,43785678	0,7443538	0,31621666	0,82260975	83,4030956	37,5775024	-1	38,3082853	-1	0,30265983	0,751334	-1	0,53516587	-1	84,9
28	-1	-1	-1	-1	-1	85,5559514	43,3079966	-1	36,5746174	-1	-1	-1	-1	-1	-1	84,0
29	0,71006348	0,54764376	0,40692811	0,81489836	0,77715654	-1	43,8967322	37,4781501	-1	-1	-1	0,04716935	0,18097899	-1	-1	-1
30	-1	-1	-1	-1	-1	-1	42,9335704	39,4939493	-1	-1	-1	-1	-1	-1	-1	-1
31	0,27147452	0,76035839	0,224554	0,75717584	0,18826801	95,7220739	-1	37,7760033	-1	40,9242095	0,87413669	-1	0,42847725	-1	0,75125506	94,4
32	-1	-1	-1	-1	-1	85,5559514	-1	31,042282	-1	28,9349692	-1	-1	-1	-1	-1	86,8
33	0,86066995	0,77403988	0,10376326	0,40251434	0,49904489	-1	-1	36,3604798	27,3047626	30,7598439	-1	-1	0,73541316	0,59279014	0,24232458	-1
34	-1	-1	-1	-1	-1	-1	-1	31,042282	36,5746174	28,9349692	-1	-1	-1	-1	-1	-1
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.

#### 4.Adım

Çaprazlamadan sonra mutasyon işlemine geçilir. Popülasyondaki her bir kromozomun her bir geni(parametresi) için [0,1] aralığında rasgele bir  $\rho$  sayısı üretilir. Eğer  $\rho$  mutasyon oranından küçükse, gen tekrar üretilerek kromozom mutasyona uğratılmış olunur.

Mutasyon işleminin sonuçları da aşağıda görülmektedir.(Şekil-6)

Şekil-6: Mutasyon Sonuçları

No	X1 önce	X1 için rnd	X1 sonra	X2 önce	X2 için rnd	X2 sonra	X3 önce	X3 için rnd	X3 sonra	X4 önce	X4 için rnd	X4 sonra	X5 önce	X5 için rnd	X5 sonra
1	93,3203732	0,0661938	-1	44,8232324	0,6671582	-1	41,5345746	0,0913176	-1	40,5929033	0,03483849	35,0260068	31,3820009	0,31012182	-1
2	93,6593812	0,06859202	-1	44,2511116	0,22193091	-1	33,2588735	0,37000794	-1	40,0747727	0,45831951	-1	33,6129331	0,41892258	-1
3	78,5499817	0,24540944	-1	39,2284867	0,53573052	-1	27,5157629	0,55561314	-1	36,4617649	0,47840323	-1	43,7917199	0,79259125	-1
4	78,8131607	0,15609808	-1	38,2977789	0,59717618	-1	40,9917562	0,83489095	-1	32,3702033	0,34231481	-1	32,4745081	0,74661103	-1
5	89,2974773	0,49677104	-1	36,9262065	0,81138117	-1	34,7652444	0,38296382	-1	37,939123	0,92893361	-1	28,6457301	0,73480235	-1
6	88,4151951	0,37921886	-1	43,1292371	0,27173122	-1	35,7162618	0,84013053	-1	40,028974	0,87354667	-1	30,3581577	0,64594174	-1
7	93,1463846	0,24752228	-1	41,1649054	0,16664672	-1	28,1451616	0,17179941	-1	28,7890097	0,32159915	-1	36,3983699	0,52595352	-1
8	94,1581411	0,8945558	-1	34,1844881	0,03117165	37,4046229	29,4558946	0,87304652	-1	43,0708021	0,25405174	-1	38,8697886	0,68138944	-1
9	87,0122075	0,78299754	-1	43,8499549	0,60544255	-1	38,905798	0,52343693	-1	27,2092029	0,31759846	-1	33,9430853	0,73522187	-1
10	88,6100036	0,16166231	-1	40,4255628	0,19914586	-1	40,8071074	0,49224517	-1	27,3047626	0,671702	-1	32,5512652	0,12001568	-1
11	81,8455066	0,25219717	-1	42,4237271	0,41523363	-1	30,7066091	0,69762916	-1	39,0223972	0,56942276	-1	33,6058716	0,85577703	-1
12	88,8380887	0,70835378	-1	38,9420792	0,10590488	-1	29,9847472	0,57531946	-1	40,0464366	0,06667041	-1	31,8780479	0,7968608	-1
13	93,1127699	0,38180038	-1	43,8363794	0,49646127	-1	34,6024575	0,81237777	-1	34,7584908	0,1714003	-1	34,2117524	0,07876272	-1
14	94,177801	0,36393111	-1	37,8060325	0,23401399	-1	39,8888663	0,9524497	-1	35,8346359	0,89803545	-1	39,4136185	0,12492643	-1
15	79,6278612	0,72412966	-1	42,6021096	0,58083921	-1	32,7316608	0,66092538	-1	40,0395225	0,37955912	-1	31,5215734	0,7137735	-1
16	78,8309237	0,20593862	-1	40,3584521	0,85952689	-1	34,8253343	0,61133515	-1	32,9728883	0,99958048	-1	27,8474543	0,21755655	-1
17	82,0649608	0,07258536	-1	35,9471908	0,34788537	-1	30,2427824	0,20184467	-1	32,9370688	0,52592606	-1	42,5263506	0,99532511	-1
18	88,8380887	0,75141308	-1	44,0437203	0,47417025	-1	43,0544912	0,75217639	-1	40,009461	0,67017649	-1	34,9739456	0,18620202	-1
19	83,2454111	0,92526677	-1	41,8097832	0,29772706	-1	37,7730693	0,18738133	-1	37,5496291	0,88452864	-1	35,9370786	0,98525153	-1
20	79,6067624	0,28653628	-1	38,8150493	0,79306478	-1	34,5403859	0,35203664	-1	42,4958878	0,23883574	-1	33,5425404	0,37129076	-1
21	79,6515983	0,92764832	-1	41,1338716	0,24669615	-1	31,1261746	0,59420972	-1	32,4878482	0,54677557	-1	40,4320986	0,39588602	-1
22	79,5535432	0,67135796	-1	40,2721855	0,75430825	-1	33,8677384	0,31807004	-1	32,2023559	0,85564926	-1	35,8013864	0,64983841	-1
23	94,4455084	0,84882689	-1	43,53471	0,49456973	-1	36,8218477	0,57573898	-1	41,5629365	0,84911386	-1	28,2410071	0,72427544	-1
24	89,5606559	0,03391501	99,3542607	44,3896438	0,29461659	-1	39,2872579	0,28645171	-1	30,5573935	0,17607519	-1	28,4731429	0,32734964	-1
25	97,2460581	0,48183759	-1	33,7629902	0,28227321	-1	30,7408944	0,71183343	-1	44,6514169	0,19965967	-1	31,5261353	0,4264026	-1
26	89,5402167	0,39345788	-1	36,9262065	0,77639674	-1	33,4135859	0,39430759	-1	35,8346359	0,42723722	-1	31,2639064	0,41287384	-1
27	84,9043684	0,50749025	-1	39,0024815	0,37249942	-1	28,0087277	0,62828972	-1	37,5024173	0,28990823	-1	39,7120543	0,82588921	-1
28	84,0546786	0,75367565	-1	41,8830175	0,65506911	-1	31,042282	0,13004004	-1	37,3804854	0,32396715	-1	28,9349692	0,99710483	-1
29	80,3034989	0,15610021	-1	42,9790021	0,89652714	-1	39,129132	0,02033808	39,9957854	36,8156023	0,33737513	-1	32,1453368	0,43069703	-1
.	.	.													
.	.	.													

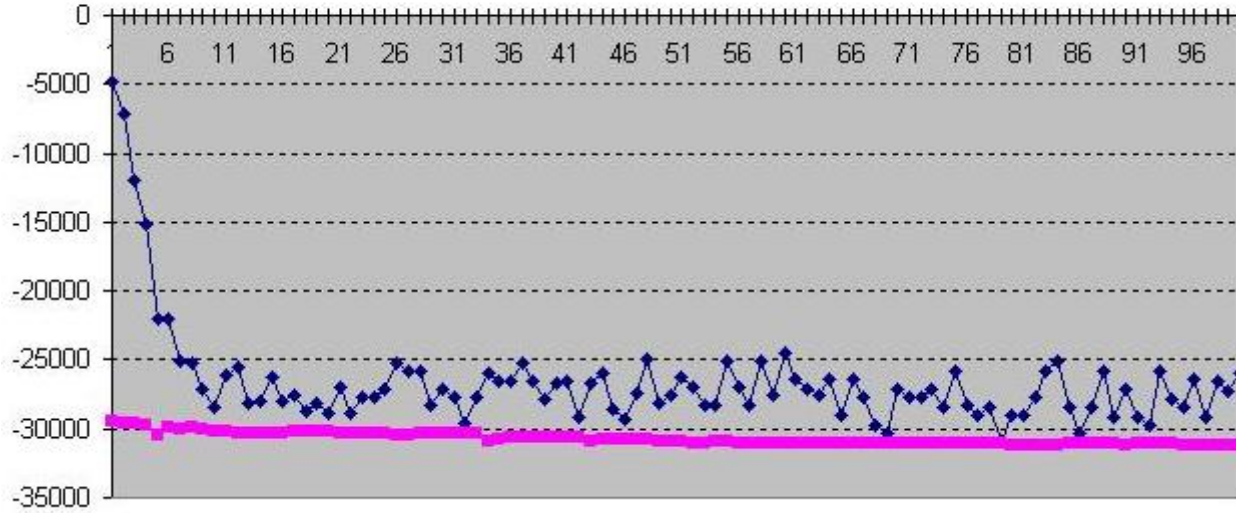
## 5.Adım

Seçilen sonlanma ölçütüne göre kontrol yapılarak sürecin devam edip etmemesine karar verilir. Sonlandırma belli bir iterasyon sayısına ulaşmak yada bir uygunluk değerine yakınsamak olabilir.

Sonuç olarak;

İşlemler istenilen sonlandırma ölçütüne kadar devam ettirildikten sonra problemin çözümüne en uygun olan bireyler popülasyonda birikmiş olur. Buna göre 100 iterasyon için, 0.7 çaprazlama oranında, 0.05 mutasyon oranı ile, 48 popülasyon boyutlu ve penalty parametresi 1000 seçilen bir örnek için yakınsama grafiği aşağıdaki gibi olmaktadır.

Yakınsama Grafiği



**X1**                      **X2**                      **X3**                      **X4**                      **X5**                      **F(x)**  
78,14143125099568 36,7138324693871 28,942713513039507 44,908168128648853 27,029791714912861 -31624,7247373866

Yazılan program ile yapılan denemelerde en uygun sonuçlardan biri olarak yukarıdaki tablada verilen sonuçlar elde edilmiştir. Aynı problemin değişik algoritmalarla çözümünün sonuçları aşağıdaki gibidir. Bu da elde edilen sonuçların tutarlı ve hatta daha iyi olduğunu göstermektedir.

Items	Reference Solution	General Reduced Gradient Method	GA Solution by Gen & Cheng, 1997 with proportional selection	The best GA solution with tournament selection
f(x)	-30665.5	-30373.950	-30182.269	-30573.244
x <sub>1</sub>	78.00	78.62	81.49	78.30
x <sub>2</sub>	33.00	33.44	34.09	33.20
x <sub>3</sub>	29.995	31.07	31.24	30.44
x <sub>4</sub>	45.00	44.18	42.20	44.91
x <sub>5</sub>	36.776	35.22	34.37	35.69

### 2.6.3: Program Arayüzü

Yapılan programın arayüzü aşağıda görülmektedir.

<< genetic tahirs >> .. jenerasyon: 1

no	x1	x2	x3	x4	x5	g1( x )	g2( x )	g3( x )	g4( x )
1	93,320373232244989	44,823232422400324	41,534574585284375	35,026006778714248	31,3820009157158	92,504646617978949	-0,504646617978949	16,837189275307054	3,162810724692946
2	93,659381196590488	44,251111597505506	33,258873484217972	40,074772748361866	33,612933149083624	93,676519126140036	-1,676519126140036	15,948045047769895	4,051954952230104
3	78,549981743706041	39,228486697491228	27,515762935632729	36,461764870094441	43,791719855404551	94,238122847670013	-2,238122847670013	13,645739702231694	6,354260297768306
4	78,8131606988869	38,297778925336118	40,991756173778214	32,3702032692187	32,474508095992746	91,067738426303663	0,93226157369633711	12,08901147123197	7,91098852876803
5	89,297477306470967	36,92620652165553	34,765244350045336	37,939122961072947	28,645730116620886	91,273999990416243	0,72600000958375688	10,570026932320758	9,429973067679242
6	88,415195060155909	43,129237056770009	35,716261756622586	40,028974029050488	30,358157701110191	92,604027914739831	-0,60402791473983086	14,055455515323999	5,944544484676001
7	93,146384581500016	41,164905426317617	28,1451616325768	28,789009745483387	36,3983698598134	93,273662609697709	-1,2736626096977091	14,411917720617609	5,588082279382391
8	94,158141136269052	37,404622879067723	29,455894584453382	43,070802119244632	38,869788628752723	93,615634604007312	-1,6156346040073117	13,323943267514437	6,676056732485562
9	87,012207523368858	43,849954927848593	38,90579798690068	27,209202932291294	33,943085294420811	92,367435169407429	-0,36743516940742893	15,858355427957562	4,141644572042437
10	88,610003595615268	40,425562775313686	40,807107400359762	27,304762621552108	32,551265183862469	91,402094155806935	0,5979058441930647	14,25966966636048	5,74033033363952

Başla Bitir Adım Adım Yavaş Yavaş Ayar  izleme ekranı  Tam ekran izleme grafik

Başlangıç Populasyonu

no	x1	x2
1	78,901642576279883	37,50932491966957
2	82,204358117750076	41,7773139200999
3	81,199208447336787	41,42223271374695
4	83,527560489963534	39,99106500809595
5	81,203858771922	39,54295714876752
6	78,461499866313062	40,01694070315777
7	93,475184034311766	41,39200669359043
8	99,457647855141033	37,31640504129063
9	96,941958700745346	38,03585663672343
10	85,55595142001097	43,30799663174338
11	80,224141203902732	40,35845213353562
12	89,815728278744842	41,58640335713811
13	92,170476711434532	44,44947647091442
14	88,415195060155909	43,12923705677000
15	98,894750407382261	42,01306433464077

Turnuva Seçimi

No	first	secc	inde	x1	x2
1	7	38	38	96,243951656969244	44
2	38	26	26	90,735802771866233	44
3	14	1	1	78,901642576279883	37
4	16	6	6	78,461499866313062	40
5	46	21	21	89,297477306470967	36
6	14	37	14	88,415195060155909	43
7	34	12	12	89,815728278744842	41
8	34	27	27	97,488797439024225	33
9	45	24	45	92,0946506290206	44
10	45	4	4	83,527560489963534	39
11	17	46	17	81,8455066028263	37
12	39	37	39	88,838088662753847	44
13	28	37	28	97,993093559515245	44
14	36	21	21	89,297477306470967	36
15	44	33	44	78,234643723924947	42

Çaprazlama

No	x1 için	x2 için
1	0,51353943045881545	0,653161177250166
2	-1	-1
3	0,66986518430982955	0,395014027317526
4	-1	-1
5	0,91796165747473091	0,763826980145567
6	-1	-1
7	0,13939483703085911	0,235837750246672
8	-1	-1
9	0,18128151361890207	0,214753580379650
10	-1	-1
11	0,93730959339873376	0,138249924936448
12	-1	-1
13	0,45462358438159506	0,485641411266122
14	-1	-1
15	0,02382916958249601	0,718117143361883

Mutasyon

No	x1 önce	rndMO x1
1	93,320373232244989	0,066193797190763
2	93,659381196590488	0,068592020808063
3	78,549981743706041	0,245409436638192
4	78,8131606988869	0,156098082734317
5	89,297477306470967	0,496771040603877
6	88,415195060155909	0,379218859774628
7	93,146384581500016	0,247522282529399
8	94,158141136269052	0,894555799613965
9	87,012207523368858	0,782997541028539
10	88,610003595615268	0,161662309971480
11	81,8455066028263	0,252197172144519
12	88,838088662753847	0,708353778677225
13	93,11276990596839	0,381800376056600
14	94,177800959389373	0,363931108901245
15	79,627861214456559	0,724129656666950

## SONUÇLAR

Evrimin modellenmesiyle birlikte Genetik algoritmaların ele alınmaları çok yakın tarihe dayanmaktadır. Son zamanlarda bu konu ile ilgili çalışmalar oldukça artmaktadır. Bilgisayar teknolojisindeki gelişmelere paralel olarak, GA'ların pratikte uygulanabilirliği artmıştır. Her geçen gün GA'nın adına yeni sempozyumlar düzenlenirken yeni uygulama alanlarına tatbik edildiği de görülmektedir. GA'nın geliştirilmesi alanında; hibridizasyon, parametrelerin optimal seçimi ve adaptasyonu gibi konularda araştırmalar devam etmektedir(Wurtz 1997).

İlk bölümde de anlatıldığı gibi klasik metotlar bazı sınırlamalara sahiptir. Bu metotlardaki ciddi sınırlamalardan biri optimizasyon algoritmasının seri çalışmasıdır. GA'da ise PC'leri paralel çalıştırılarak hızlı sonuçlar elde edilebilir. Birkaç tane algoritmanın aynı anda çalıştırılması ve popülasyonları arasında göç olabilmesi paralel işlemcilerle yapılır. Buna ek olarak popülasyondaki fertlerin maliyet değerleri aynı anda hesaplanarak GA'nın yakınsaması hızlandırılır.

GA karar mekanizmasında kullanılabilir mi? sorusu sorulduğunda subjektif maliyet fonksiyonlu GA akla gelmektedir. Bu algoritma market stoklarının değerlendirilmesi, savaşta askeri stratejilerin geliştirilmesi gibi karar organlarında kullanılabilir.

GA geçici bir heves veya bir moda olmayıp uzun süre gündemde kalacak bir yöntemdir. Tabiatın hesaplama algoritmaları için iyi bir model olduğu bilinmektedir. Bütün bunlardan sonra tabiatta iyilikler ve güzellikler azalmaktadır ve belki belli bir müddet sonra tabiatta global optimuma ulaşacaktır.

## KAYNAKLAR

1. Adewuya, A.A., 1996, New methods in Genetic Search with Real Valued Chromosomes, Master's Thesis, Cambridge:Massachusetts Institute of Technology.
2. Angeline, P.J., 1995, "Evolution revolution: An introduction to the special track on genetic and evolutionary programming," IEEE Expert Intelligent Systems and their Applications 10, June pp.6-10
3. Broyden, G.C., 1965 Oct., "A class of methods for solving nonlinear simultaneous equations," Math. Comput., pp.577-593
4. Curtis, H., 1975, Biology, 2<sup>nd</sup> Ed., New York: Worth publisher.
5. Davis, L., 1991, "Hybridization and numerical representation," in L.Davis(Ed.), The Handbook of Genetic Algorithms, New York: Van Nostrand Reinhold, pp. 61-71
6. Eshelman, L.J., and D.J. Shafter, 1993, "Real-coded genetic algorithms and interval-schemata, " in D.L. Whitley (Ed.), Foundations of Genetic Algorithms 2, San Mateo, CA: Morgan Kaufman, pp.187-20
7. Goldberg, D.E., 1993 Feb., "Making genetic algorithms fly: A lesson from the Wright brothers," Adv. Technol. Dev. 2, pp.1-8.
8. Goldberg, D.E.,1989, Genetic Algorithms in Search, Optimization, and Machine Learning, New York: Addison Wesley.
9. Grant, V., 1985, The Evolutionary process, New York: Columbia University Press.
10. Haataja,1994, Solving Optimization Problems. CSC- Center for Scientific Computing Ltd, Yliopistopaino. ISBN 952-9821-02-6. Ed. 1. 232p.(In Finnish)
11. Haupt Randy L., Haupt Sue E., 1998, Practical Genetic Algorithms, A Willey-Interscience Publication, USA
12. Holland, J.H., 1975, Adaptation in Natural and Artificial Systems, Ann Arbor: The University of Michigan Press.
13. Holland, J.H., 1992 july, "Genetic algorithms,"" Sci. Am. Pp. 66-72
14. Michalewicz, Z., 1994, Genetic Algorithms + Data structures = Evolution Programs, 2<sup>nd</sup> ed., New York: Springer-Verlag.
15. Palko S., 1996, "Structural Optimization of Induction Motor using a Genetic Algorithm and a Finite Element Method", Acta Polytechnica Scandinavica, Electrical Engineering Series No. 4, Helsinki, pp. 10-12
16. Pierre, D.A., 1992, "Optimization," in McGraw-Hill Encyclopedia of Science and Technology 12, New York: McGraw-Hill, pp.476-482

17. Radcliff, N.J., 1991, "Forma analysis and random respectful recombination." In Proc. of Fourth International Conference on Genetic Algorithms, San Mateo, CA: Morgan Kauffman.
18. Wright, A., 1991, " Genetic Algorithms for real parameter optimization," in G.J.E. Rawlins (Ed.), Foundations of Genetic Algorithms 2, San Mateo, CA: Morgan Kaufman, pp.205-218
19. Wurtz F., Richomme M., Bigeon J., Sabonnadiere J.C, 1997, "A Few Results for Using genetic algorithms in Design of Electrical Machines", IEEE Transactions on Magnetics, Vol.33, No.2
20. NabiyeV V.V. 2003. "Yapay Zeka", Seçkin yayınları,

### 3. ŐEKİL LİSTESİ

Őekil 1: Optimizasyonun temel aŐamaları	1
Őekil 2: Optimizasyon algoritmalarının grupları	3
Őekil 3: İkili kodlu GA ile biyolojik evrim arasındaki benzetim (Haupt 1998)	9
Őekil 4: İkili kodlu GA'nın akıŐ diyagramı	10
Őekil 5: Dađın üç boyutlu görünümü (Haupt 1998)	11
Őekil 6: Gen stringlerinde parametrelerin kodlanması	14
Őekil 7: ikili ve gray kodlama için Hamming mesafesi	14
Őekil 8: BaŐlangıŐ popülasyonunun topođrafik harita üzerinde dađılımı (Haupt 1998)	17
Őekil 9: Tek noktali ıaprazlama	23
Őekil 10: Gerıek kodlu GA'da ıaprazlama	23
Őekil 11:	24
Őekil 12: İkinci neslin topođrafik harita üzerinde dađılımı (Haupt 1998)	26
Őekil 13: Dördüncü neslin topođrafik harita üzerinde dađılımı (Haupt 1998)	27
Őekil 14: Sekizinci neslin topođrafik harita üzerinde dađılımı (Haupt 1998)	28
Őekil 15: İlk sekiz neslin ortalama ve minimum uygunluk deđerleri (Haupt 1998)	29
Őekil 16: Gerıek kodlu GA'nın akıŐ diyagramı	30
Őekil 17: $f(x, y) = x \sin(4x) + 1,1 y \sin(2y)$ 'nin üç boyutlu görünümü	31
Őekil 18: Birinci neslin topođrafik harita üzerinde dađılımı (Haupt 1998)	32
Őekil 19: Üçüncü neslin topođrafik harita üzerinde dađılımı (Haupt 1998)	39
Őekil 20: BeŐinci neslin topođrafik harita üzerinde dađılımı	40
Őekil 21: Yedinci neslin topođrafik harita üzerinde dađılımı	40
Őekil 22: Yedi tane neslin ortalama ve minimum uygunluk deđerleri (Haupt 1998)	41
Őekil 23: 8 Őehirli GSP Őeması	62