EGE ÜNİVERSİTESİ

GENETİK ALGORİTMA

Hazırlayanlar

Dilay GÖK (91090017986)

Ceren ÖCAL (91080016905)

Şerife BAŞ (91080016731)

Danışman

Yrd.Doç.Dr.Aybars UĞUR

GENETİK ALGORİTMA

- Genetik Algoritma Nedir?
- GA Tarihçesi
- Neden GA?
- GA Amacı
- Uygulama Alanları
- Temel Kavramlar
- GA Akış Diagramı
- Genetik Programlama Taslağı
- Genetik Operatörler
- Genetik Algoritmalarda Parametre Seçimi
- Örnek Problem Çözümü
- Diğer Yöntemlerden Farkı
- Sonuçlar
- A Fast TSP Solver Using GA on JAVA

Genetik Algoritma Nedir?

- Genetik algoritmalar, doğal seleksiyon ve genetik kuralları üzerine kurulmuş arama algoritmalarıdır (Goldberg – 1989).

- Genetik algoritmalar genetik ve evrim konularının anlaşılmasından sonra modellenen bir yöntemdir.

Genetik Algoritma Nedir?

- **Genetik Algoritmalar, doğada gözlemlenen evrimsel sürece benzer bir şekilde çalışan arama ve eniyileme yöntemidir.**

- **GA, doğadaki canlıların geçirdiği süreci örnek alır ve iyi nesillerin kendi yaşamlarını korurken, kötü nesillerin yok olması ilkesine dayanır.**

Genetik Algoritma Nedir?

- GA evrimsel yaklaşım ilkeleri ışığında raslantısal araştırma yöntemlerini kullanarak kendi kendine öğrenme ve karar verme sistemlerinin düzenlenmesini hedef alan bir araştırma tekniğidir.

- Genetik Algoritma, anne ve baba bireyden (bir önceki nesilden) doğan yeni bireylerin şartlara uyum sağlayıp yaşamlarını devam ettirmesine dayanır.

Genetik Algoritma Nedir?

- **Gentik algoritmalar, doğal seçilim ve genetiğe dayalı heuristic (sezgisel) arama algoritmalarına kolayca uyum sağlarlar.**

- **Heuristic nedir?**

  Bütün olasılıklar arasından çözümler bulan algoritmalar için kullanılır. Bu algoritmalar, genellikle en iyiye yakın çözümler bulur ve bu işlemi hızlı ve kolayca yapar.

Genetik Algoritma Nedir?

- Genetik algoritmaların önemli bir üstünlükleri, çözümlerden oluşan populasyonu eş zamanlı incelemeleri ve böylelikle yerel en iyi çözümlere takılmamalarıdır.

- "Reeves; ulusal hükümetler ve organizasyonlar tarafından genetik algoritma tabanlı projelere, tavlama benzetimi (simulated annealing) ve yasaklı arama (tabu search) tabanlı projelere göre daha fazla kaynak ayrılmakta olduğunu belirtmektedir" (Yeniay, 2001: 37).

GA Tarihçesi

- Genetik algoritmalar, Darwin' in evrim teorisinden etkilenerek geliştirilmiştir. Problemler evrimsel bir süreç kullanılarak bu süreç sonunda en iyi sonucu veren çözüme erişmeye çalışmaktadır. Başka bir ifadeyle çözüm evrimleşmektedir.

- İlk bilgisayar bilimcileri (Alan Turing, John von Neumann, Robert Wiener ve diğerleri) bilgisayar programlarına zeka ve yaşamsal yetenekler ile çevreleriyle etkileşimde bulunma özellikleri katmaya çalışmışlardır.

GA Tarihçesi

- 1950 ve 1960'larda bazı bilgisayar mühendisleri evrimin mühendislik problemleri için bir optimizasyon aracı olarak kullanılabileceğini düşünüp evrimsel sistemler üzerinde çalışmaya başlamışlardır.

- Evrimsel hesaplama 1960'larda I.Rechenberg'in Evrim Stratejileri ("Evolutions strategie") adlı çalışmasında tanıtılmıştır.

GA Tarihçesi

- Genetik algoritmaların temel ilkeleri ilk kez Michigan Üniversitesi'nde John Holland tarafından ortaya atılmıştır.
- Mekanik öğrenme (machine learning) konusunda çalışan Holland, Darwin'in evrim kuramından etkilenerek canlılarda yaşanan genetik süreci bilgisayar ortamında gerçekleştirmeyi düşünmüştür.
- 1975 yılında,John Holland, yaptığı çalışmaları "Adaptation in Natural and Artificial Systems" adlı kitabında bir araya getirmiştir.

GA Tarihçesi

- 1985 yılında Holland'ın öğrencisi olan David E. Goldberg'in gaz boru hatlarının denetimi üzerine yaptığı doktora tezi ile "1985 National Science Foundation Genç Araştırmacı" ödülünü kazanmıştır.

- Böylece genetik algoritmaların pratik kullanımının da olabilirliğini kanıtlamış,ayrıca genetik algoritmalara dayalı tam 83 uygulamaya yer vererek GA'nın dünyanın her yerinde çeşitli konularda kullanılmakta olduğunu göstermiştir.

GA Tarihçesi

- 1987 yılında, Liepis ilk defa genetik algoritma yapısını iki makineli çizgeleme problemine uygulamıştır.

- 1990 yılında, Biegal ve Daven atölye çizgelemede genetik algoritmayı bütünleşmiş imalat çevrimi içinde kullanmışlar ve bu yapıyı tek, çift ve çok makineli sistemlerde uygulamışlardır.

GA Tarihçesi

- 1992 yılında John Koza, genetik algoritmaları kullanarak, programları evrimleştirerek belli işleri yapmakta kullanmıştır.

- Bu kullandığı yönteme “genetik programlama” adını vermiş ve LISP dilinde programlar “Ayrıştırma Ağaçları”(“Parse Tree”) şeklinde ifade edildiği için LISP diliyle geliştirilmiştir.

GA Tarihçesi

- GA, sonraları gezgin satıcı,karesel atama, yerleşim, atölye çizelgeleme, ders/sınav programı hazırlanması gibi problemlerde başarıyla uygulanmıştır.

- Son yıllarda üretim planlama, tasarım, elektronik ve finansman gibi farklı ve çok geniş alanları kapsayan gerek teorik, gerekse uygulamalı GA çalışmaların sayısı giderek artmaktadır.

Neden GA?

- **Problemin zorluk derecesinin bilinmesi problemin çözümü için en iyi yöntemin uygulanmasını sağlar.**

- **GA, problemin gerçek olmayan ancak geçerli olan çözümünü kısa sürede bulabilir.**

GA Amacı

- Genetik algoritmaların asıl amacı, hiçbir çözüm tekniği bulunmayan problemlere çözüm aramaktır.

-

- Kendilerine has çözüm teknikleri olan özel problemlerin çözümü için mutlak sonucun hızı ve kesinliği açısından genetik algoritmalar kullanılmazlar.

GA Amacı

Genetik algoritmalar ancak;

- Arama uzayının büyük ve karmaşık olduğu,
- Mevcut bilgiyle sınırlı arama uzayında çözümün zor olduğu,
- Problemin belirli bir matematiksel modelle ifade edilemediği,
- Geleneksel eniyileme yöntemlerinden istenen sonucun alınmadığı alanlarda etkili ve kullanışlıdır.

Genel Uygulama Alanları

- Optimizasyon
- Otomatik Programlama (automatic programming)
- Makine öğrenmesi (machine learning)
- Ekonomi (economics)
- Ekoloji (ecology)
- Bağışık sistemler (Immune systems)
- Popülasyon genetiği (population genetics)
- Evrim ve öğrenme (evolution and learning)
- Sosyal sistemler (social systems)

Optimizasyon

- **<u>Genetik algoritmaların uygulandığı optimizasyon problemleri iki başlık altında toplanabilir:</u>**

  - *Fonksiyon optimizasyonu* : Geleneksel optimizasyon tekniklerine göre zor, süreksiz ve gürültü (noisy) içeren fonksiyonları çözmekle ilgili.

  - *Birleşi (combinatorial) optimizasyonu*: İstenen amaçlara ulaşmak üzere, sınırlı kaynakların etkin tahsis edilmesiyle ilgili.

Otomatik Programlama ve Bilgi Sistemleri

- Genetik algoritmaların yaygın olarak kullanıldığı alanlardan biri, belirli ve özel görevler için gerekli olan bilgisayar programlarını geliştirmedir.

- Bunlara örnek olarak, bilgisayar çipleri tasarımı, ders programı hazırlanması ve ağların çizelgelenmesi verilebilir.

Mekanik Öğrenme

- Sınıflama sistemi, genetik algoritmaların mekanik öğrenme alanında bir uygulamasıdır.
- Basit dizi kurallarını öğrenen bir mekanik öğrenme sistemi olan sınıflama sisteminin kural ve mesaj sistemi, özel bir üretim sistemi olarak adlandırılabilir.
- Genetik algoritmalar, sınıflama sistemlerinde kural-bulma mekanizması olarak kullanılmaktadırlar.

Ekonomik ve Sosyal Sistem Modelleri

- **Bir sistemi ölçen ampirik olarak gözlenmiş değişkenler arasındaki matematiksel ilişkiyi keşfetme problemi ekonomide en önemli problemlerden biridir. Genetik algoritmaların kullanıldığı genetik programlamayla bu tip problemlere tatmin edici çözümler çok daha kolay getirilebilmektedir.**

- **Genetik algoritmalar yenilik sürecinin modellenmesi amacıyla da kullanılmaktadır.**

İşletmelerdeki Uygulama Alanları

- Finans
- Pazarlama
- Üretim/işlemler
- Montaj hattı dengeleme
- Çizelgeleme problemi
- Tesis yerleşim problemi

Bazı GA Uygulama Tipleri

Domain (Tanım Kümesi)	Uygulama Tipleri
Kontrol	Gaz boru hattı, füze atılması, takip
Tasarım	Uçak tasarımı, klavye tasarımı, iletişim ağları
Zamanlama	Üretim, kaynakların dağıtımı
Robot Bilimi	Rota planlaması
Makine Öğrenmesi	Sinir ağları tasarlanması, sınıflandırma algoritmalarının gelişimi, sınıflandırıcı sistemler
Sinyal İşleme	Filtre tasarımı
Oyun	poker, dama
Kombinasyonal Optimizasyon	Gezgin satıcı, yönlendirme, grafik renklendirme

Temel Kavramlar

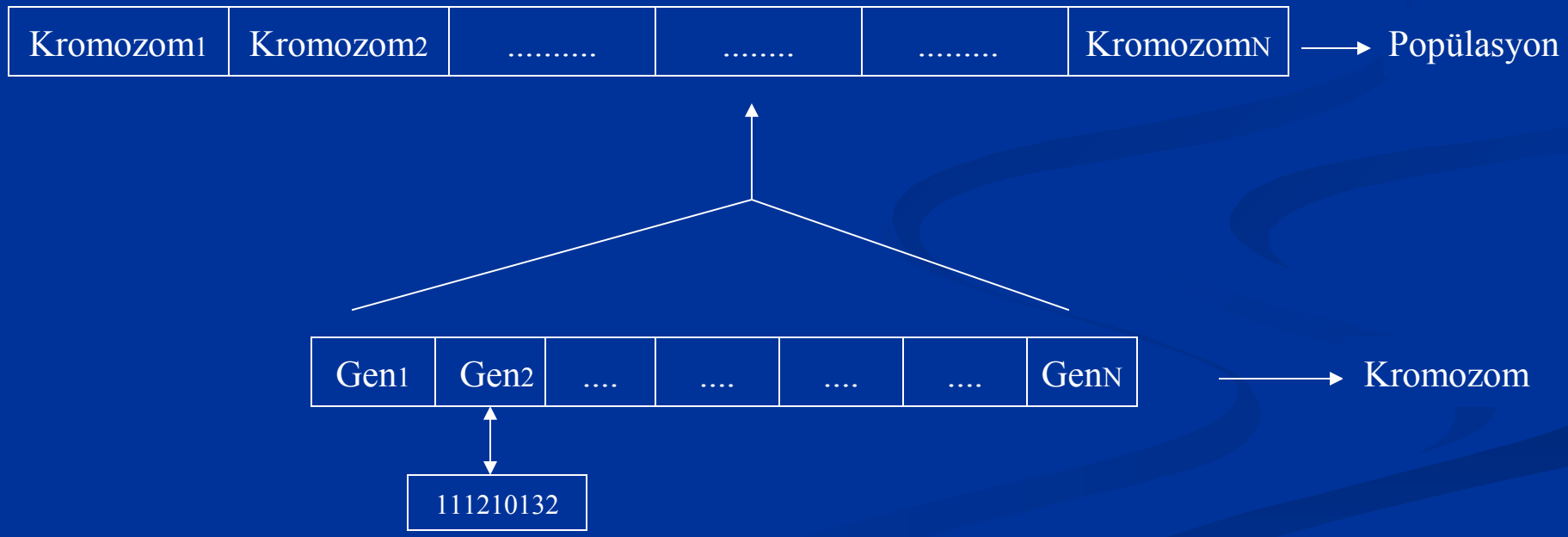
[1] Gen;kromozom yapısında kendi başına birer genetik bilgi taşıyan en ufak yapı birimidir.

[2] Kromozom;bir yada birden fazla gen yapısının bir araya gelerek problemin çözümüne ait tüm bilgiyi içeren dizilerdir.

[3] Popülasyon(Yığın);çözüm bilgilerini içeren kromozomların bir araya gelmesiyle oluşan olası çözüm yığınıdır.

Temel Kavramlar

- **Popülasyonun Yapısı**



Temel Kavramlar

[4] Başlangıç Popülasyonunun Oluşturulması;

- GA'nın ilk adımı başlangıç popülasyonunun oluşturulmasıdır.

- Genelde başlangıç popülasyonu rasgele oluşturulur. Ancak bu olay kısıtlı en iyileme problemlerinde popülasyonun uygun olmayan çözümlere doğru yönelmesine sebep olabilir. Bu durumu ortadan kaldırmak için probleme özgü çeşitli sezgisel yöntemler geliştirilebilir.

Temel Kavramlar

[5] Uygunluk değeri(fitness value);

- Popülasyon içindeki her bireyin problem için çözüm olup olmayacağına karar veren bir uygunluk(fitness) fonksiyonu vardır.

- Uygunluk fonksiyonundan dönen değere göre yüksek değere sahip olan bireylere, nüfustaki diğer bireyler ile çoğalmaları için fırsat verilir.

Fitness Value

- **Bu bireyler çaprazlama işlemi sonunda çocuk adı verilen yeni bireyler üretirler.**

- **Çocuk kendisini meydana getiren ebeveynlerin (anne, baba) özelliklerini taşır.**

- **Yeni bireyler üretilirken düşük uygunluk değerine sahip bireyler daha az seçileceğinden bu bireyler bir süre sonra popülasyonun dışında bırakılır.**

Fitness Value

- Yeni popülasyon, bir önceki popülasyonda yer alan uygunluğu yüksek bireylerin bir araya gelip çoğalmalarıyla oluşur.
- Aynı zamanda bu popülasyon önceki popülasyonun uygunluğu yüksek bireylerinin sahip olduğu özelliklerin büyük bir kısmını içerir.
- Böylelikle, pek çok nesil aracılığıyla iyi özellikler popülasyon içersinde yayılırlar ve genetik işlemler aracılığıyla da diğer iyi özelliklerle birleşirler.

Fitness Value

- **Uygunluk değeri yüksek olan ne kadar çok birey bir araya gelip, yeni bireyler oluşturursa arama uzayı içerisinde o kadar iyi bir çalışma alanı elde edilir.**

- **Uygunluk değeri, popülasyondaki diziler (kromozomlar) bir değerlendirme fonksiyonu yardımıyla hesaplanır. GA'da kullanılan değerlendirme işlevi veya uygunluk fonksiyonu problemin amaç işlevini oluşturmaktadır.**

Temel Kavramlar

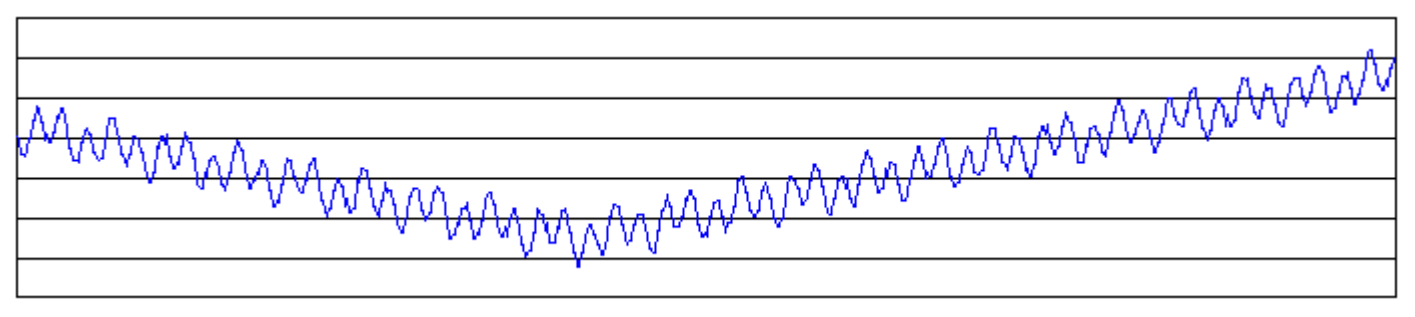
[6] Arama Uzayı;

- Mümkün tüm çözümlerin uzayına yani istenen çözümün aralarından bulunduğu çözümler kümesine arama uzayı (durum uzayı) adı verilir.

- Arama uzayındaki her nokta bir olası çözümü temsil eder. Her olası çözüm değeri (uygunluğu) ile problem için işaretlenebilir.

- Genetik algoritmalar yardımıyla arama uzayındaki olası çözümler arasından en iyi çözüm aranır.

Temel Kavramlar



http://www.obitko.com/tutorials/genetic-algorithms/ga-basic-description.php

-

Temel Kavramlar

[7] Tekrar Üretim;

- **Tekrar üretim sırasında, yeniden birleşme (veya çaprazlama) ilk önce ortaya çıkar.**
- **Atalardan gelen genler yepyeni bir kromozom üretmek için bir araya gelirler. Bu yeni yaratılmış nesil daha sonra mutasyona uğrayabilir.**
- **Mutasyon DNA elemanlarının değişmesidir. Bu değişimler genellikle atalardan gen kopyalanması sırasındaki hatalardan kaynaklanır. Bir organizmanın uygunluğu ("fitness") organizmanın yaşamındaki başarısıyla (hayatta kalmasıyla) ölçülür.**

Temel Kavramlar

[8] Seçim Mekanizması;

- Bir nesildeki dizilerden bir kısmının bir sonraki nesle aktarılırken bir kısmı da yok olur. Bu aşamada hangi dizilerin bir sonraki nesle aktarılacağı kurulan seçim mekanizmaları ile sağlanır.

- GA'da kullanılan en basit ve en yaygın olan seçim mekanizması "Rulet Tekerleği" seçimidir.

Rulet Tekerleği (Roulette Wheel)

- **Bir nesildeki dizilerden bir kısmının bir sonraki nesle aktarılırken bir kısmı da yok olur. İşte bu aşamada hangi dizilerin bir sonraki nesle aktarılacağı kurulan seçim mekanizmaları ile sağlanır.**

- **GA'da kullanılan en basit ve en yaygın olan seçim mekanizması RULET TEKERLEĞİ seçimidir.**

Rulet Tekerleği (Roulette Wheel)

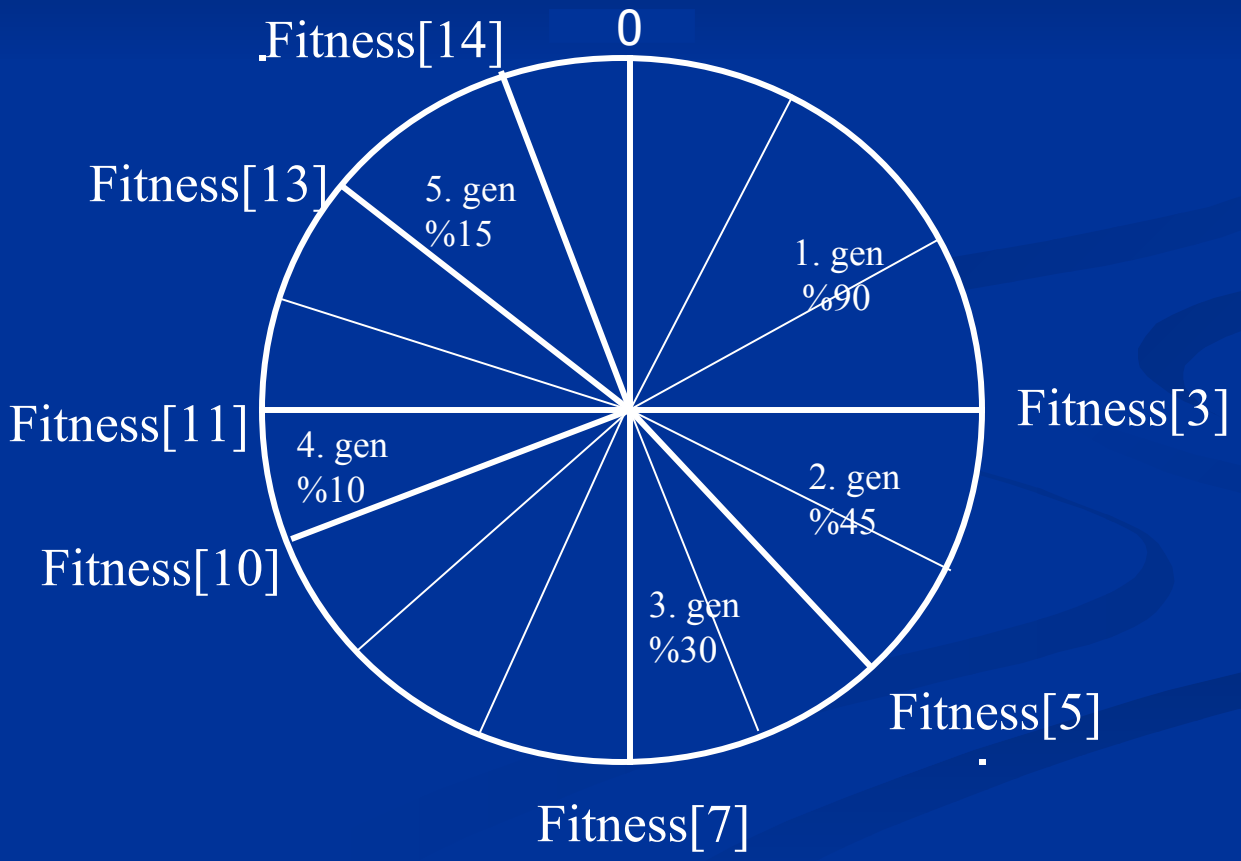
- Çember n tane parçacığa bölünür.
- Her aralık bir diziyi temsil eder.
- Her dizinin uygunluk değeri toplam uygunluk değerine bölünür. Bu sayede popülasyonun içindeki her dizinin çözüm kümesi içindeki [0-1] değerleri arasındaki yeri bulunur.
- Diziler uygunluk değerlerine göre yüzdelik olarak çemberde gösterilir.

Rulet Tekerleği

- Tekrar üreme için rulet tekerleğinin döndürülmesi gerekir. Bunun için sıfırla toplam uygunluk arasında rasgele bir sayı seçilerek bu sayının tekerleğin hangi parçasına karşılık geldiğine bakılarak kromozom seçilir.
- Bu sayede çemberin bir defa döndürülmesiyle bir sonraki nesle aktarılacak olan dizilerden bir tanesi seçilmiş olur.
- Benzer şekilde diğer kromozomlarında belirlenmesi ile uygunluk değeri en başarılı olan bireyler eşleştirme havuzuna alınır.
- Bundan sonra artık diğer nesle ait diziler ve genetik operatörlerin uygulanmasıyla yeni nesil elde edilir.
- Aynı işlem her döngüde devam ederek nesil devamı sağlanır.

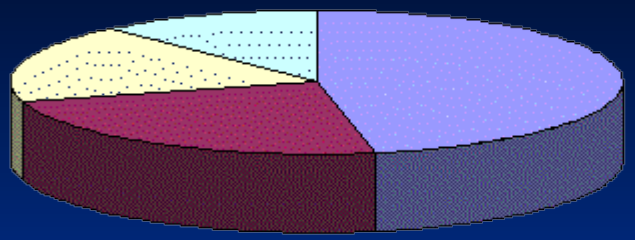
Rulet Tekerleği

Örnek:Rulet Tekerleği seçme operatörü

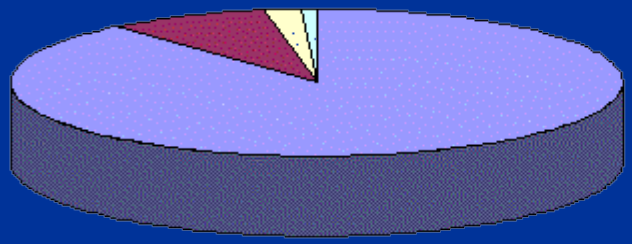


Rank Seçimi

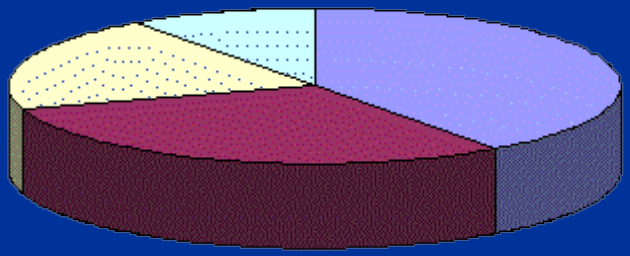
- En iyi kromozomun fitness değeri çok yüksek ise, Rulet-Çemberi seçim yöntemi problem yaratabilir (Sürekli yüksek olasılığa sahip kromozom seçilecek). Bu nedenle, Rank seçim yöntemi uygulanabilir.
- Rank seçim yönteminde, popülasyon uygunluk değerine göre tersten sıralanır. Yani en iyi kromozom N adetlik bir popülasyonda N değerini alır.
- Seçim bu değerlere göre yapılır.



Rulet-Çember Seçimi

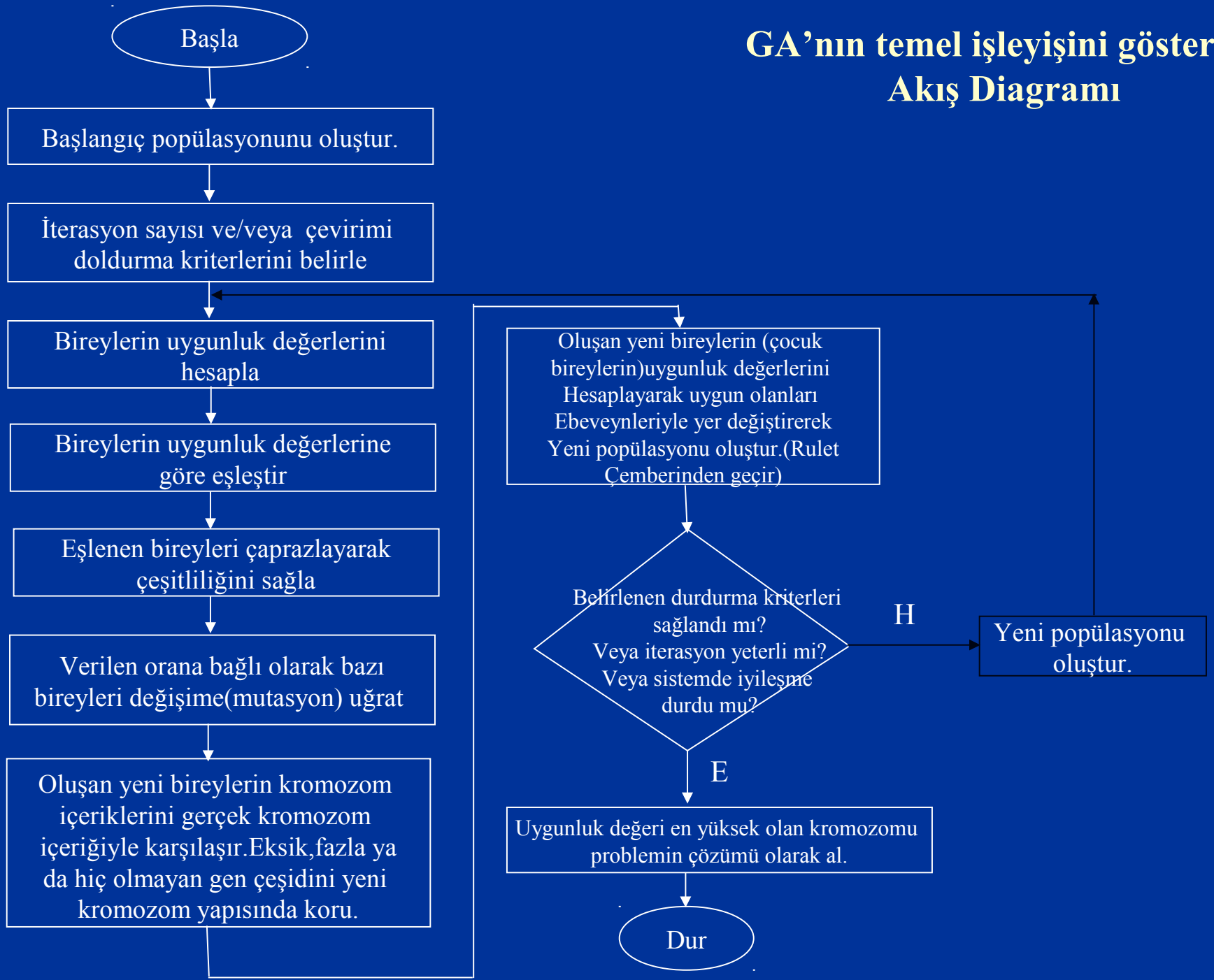


Rank Seçimi Öncesi



Rank Seçimi Sonrası

GA'nın temel işleyişini gösteren Akış Diagramı



Genetik Programlama Taslağı

1. Başlangıç: n kromozom oluşan rasgele toplumu oluşturulur.
2. Uygunluk: Toplumdaki her x kromozomu için f(x) uygunluk değeri hesaplanır.
3. Yeni Toplum: Aşağıdaki adımlar izlenerek yeni toplum üretilir;
   - Seçim: Toplumdan uygunluklarına göre iki ata seçilir.
   - Çaprazlama: Çaprazlama olasılığı ile ataları yeni yavru oluşturmak için birbirleriyle eşleştirilir. Eğer çaprazlama yapılmazsa, yavru ataların tıpatıp aynısı olacaktır.
   - Mutasyon: Mutasyon olasılığı ile yeni yavru üzerinde her yörünge için mutasyon işlemi yapılacaktır.
   - Kabul: Yeni yavru, yeni topluma eklenir.
4. Değiştir: Yeni toplum algoritmanın tekrar işlenmesinde kullanılır.

5. Deney: Eğer bitiş durumu sağlandıysa, durup toplumdaki en iyi çözüm döndürülür.

6. Döngü: Adım 2'ye gidilir.

Probleme ait en iyi çözümün bulunabilmesi

- Bireylerin gösterimi doğru bir şekilde yapılmalı,
- Uygunluk fonksiyonu etkin bir şekilde oluşturulmalı,
- Doğru genetik işlemciler seçilmelidir.

Genetik Operatörler

- Üreme
- Çaprazlama
- Mutasyon
- Kodlama Çeşitleri
- Elitizm

ÜREME (Reproduction)

- Bir dizinin bir sonraki nesile aktarılmasıdır. Dizinin kopyalanarak bir sonraki nesile aktarılması olasılığı dizinin fitness function ile hesaplanan fitness değerine bağlıdır.

- Her yeni oluşacak nesil için Reproduction operatörü bir mating pool (eşleştirme havuzu)'da bulunan diziler arasından seçim yapar (roulette wheel).

Reproduction

- **10000 dizisi, reproduction için %46 oranında seçilmesi gereken dizidir.**
- **01001 dizisi ise sadece %19 olasılıkla seçilebileceği için en zayıf reproduction olasılığı olan dizidir.**

Mating pool

String	Fitness Value	Percentage
01001	5	19%
10000	12	46%
01110	9	35%

GA'da çözüm yığını incelenirken belirli noktalardan sonra nesil çeşitliliği olmadığı için çözüme gidilememektedir.

Nesil çeşitliliğini sağlayarak çözüm uzayında algoritma istenen kısıtları sağlayacak olan çözüm yığınına ulaşabilir.

Bunun için dizilere çaprazlama ve değişim ya da mutasyon operatörleri uygulanarak nesil çeşitliliği sağlanır.

Böylelikle sistemin belirli noktalara gelip takılması önlenmiş olur.

ÇAPRAZLAMA (Crossover)

- İki dizinin bir araya gelerek karşılıklı gen yapılarının değişimi ile yeni dizilerin oluşumunu sağlayan operatördür.
- Çaprazlama, atalardaki seçili genler üzerinde işlem yapar ve yeni yavrular oluşturur.
- Çaprazalamada bir önemli unsurda ne tür bir çaprazlamanın yapılacağıdır. Örneğin, eş kromozom seçiminde ilk kromozom en yüksek uygunluk değerine sahip kromozom seçilirken ikinci kromozom rasgele olarak seçilebilir.

GA’da Çaprazlama Yöntemleri

[1] Tek noktalı Çaprazlama

[2] İki noktalı Çaprazlama

[3] Çok Noktalı Çaprazlama

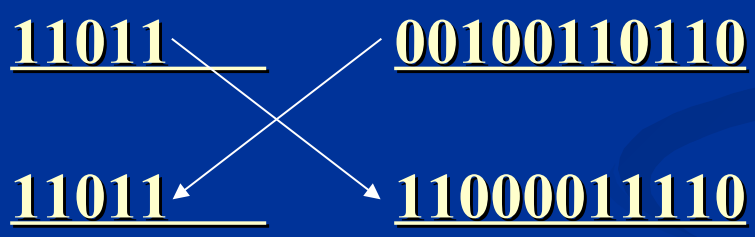
[1] Tek Noktalı Çaprazlama

- **Tek bir kesme noktası seçilir, ilk atanın kromozomundan kesme noktasına kadar baştan itibaren alınır.**
- **Geri kalan kısım ikinci atanın kesme noktasından sonraki kısmıyla birleştirilip yavrunun kromozomu oluşturulur.**

Tek Noktalı Çaprazlama

Ör:

Kromozom 1: 1101100100110110
Kromozom 2: 1101111000011110



Yavru 1 : 11011 | 11000011110
Yavru 2 : 11011 | 00100110110

[2] İki Noktalı Çaprazlama

- **İki kesme noktası seçilir.**

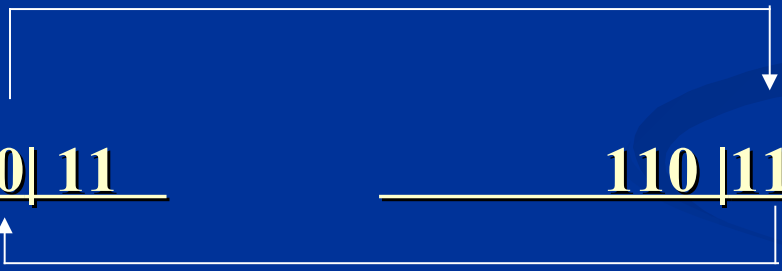
- **Kromozomun başından ilk kesme noktasına kadar olan ikili karakter dizisi ilk atadan, iki kesme noktası arasındaki kısım ikinci atadan ve ikinci kesme noktasından sonraki kısım tekrar ilk atadan alınarak yeni yavru oluşturulur.**

[2] İki Noktalı Çaprazlama

Ör:

Kromozom1:11001011

Kromozom2:11011101



Yavru1= 11011111

Yavru2= 11001001

[2] Çok Noktalı Çaprazlama

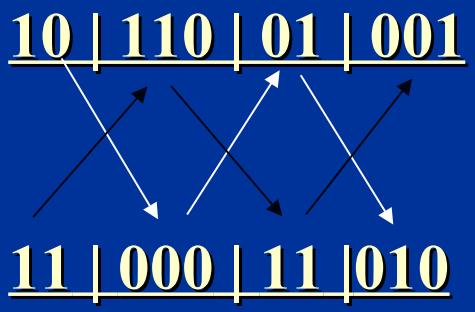
- **Rasgele çoklu bölge seçilir.**
- **Eşlenen iki dizide bu rasgele seçilen çaprazlama noktaları arasında kalan bölümler yer değiştirerek iki tane yeni birey elde edilir.**

[2] Çok Noktalı Çaprazlama

Ör:

Kromozom 1: 1011001001

Kromozom 2: 1100011010



Yavru 1: 1000001010

Yavru 2: 1111011001

ÇAPRAZLAMA (Crossover)

Tek noktalı, iki noktalı ve çok noktalı çaprazlama işlemleri GA'da en yaygın çaprazlama yöntemleridir. Ancak problemin özelliğine göre farklı tiplerde çaprazlama yapmakta mümkündür. Bu çaprazlama yöntemlerinden bazıları;

- Pozisyona dayalı çaprazlama
- Sıraya dayalı çaprazlama

MUTASYON (Mutation)

- GA'da sistem belli döngü değerine geldikten sonra diziler birbirlerine gitgide benzemektedir. Bu da çözüm uzayının daralmasına neden olmaktadır.
- Dizilere ne kadar çaprazlama operatörü uygulansa da ilerleyen nesillerde dizi çeşitliliği sağlanamamaktadır. Bu durumda dizinin kendi içindeki genler rasgele yerdeğiştirilir. Böylelikle dizi çeşitliliğinin devamı sağlanmış olur.

MUTASYON (Mutation)

- **Çaprazlama işlemi gerçekleştirildikten sonra, mutasyon işlemi yapılır.**

- **Mutasyon işlemi çaprazlama sonucu oluşan yavruyu rasgele değiştirmektedir. İkili kodlamada rasgele seçilmiş bir kaç biti 1'i 0'a, 0'ı 1'e şeklinde değiştirmek bir mutasyondur.**

MUTASYON (Mutation)

Ör:

Asıl Yavru 1: 1101111000011110

Mutasyon Geçirmiş Yavru 1: 1100111000011110

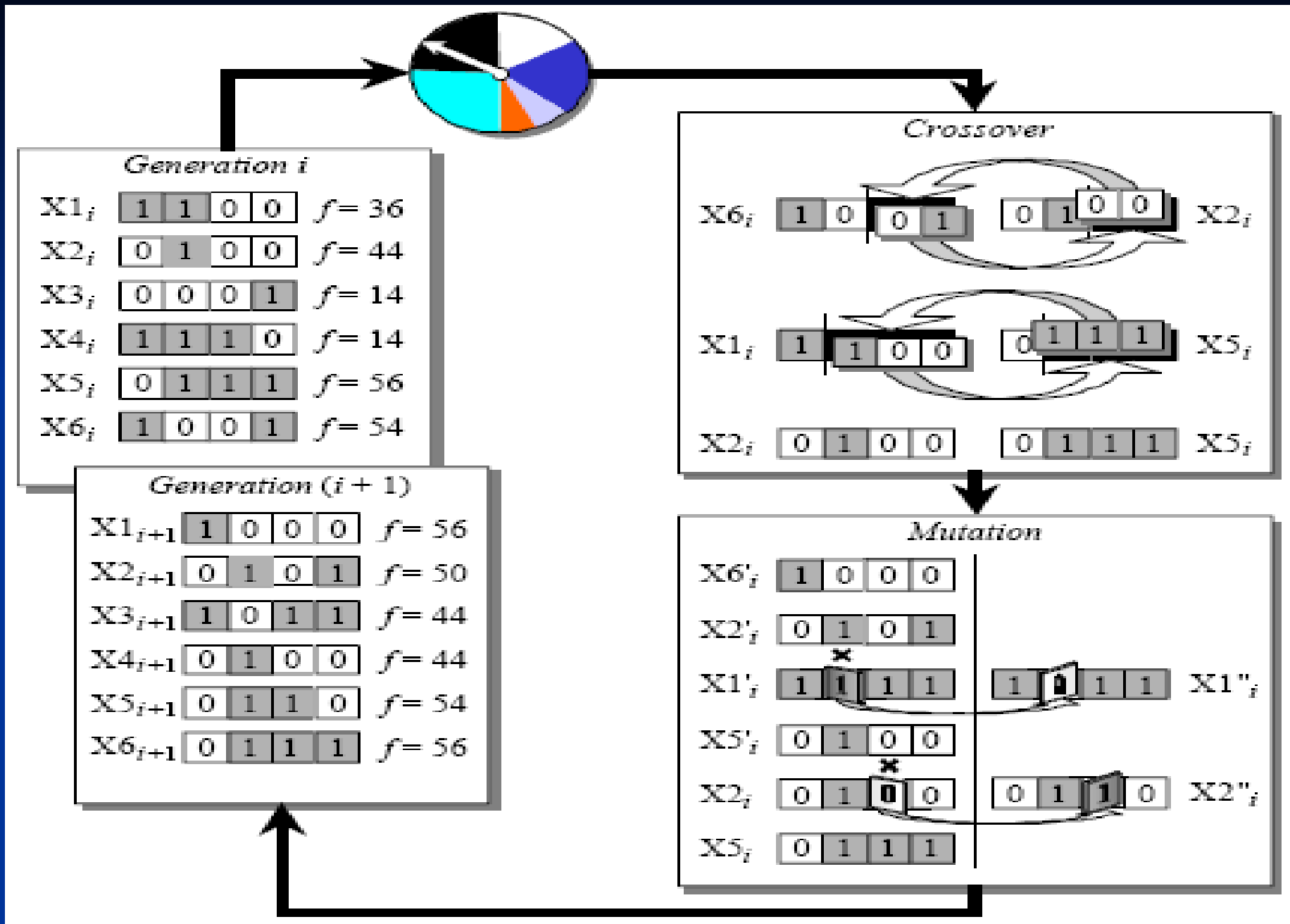
Asıl Yavru 2: 1101100100110110

Mutasyon Geçirmiş Yavru 2: 1101101100110110

MUTASYON (Mutation)

- **Hem mutasyon tekniği hem de çaprazlama tekniği çoğunlukla kromozomların kodlamasına bağlıdır.**

- **Örneğin; permütasyon şeklinde kodlamada mutasyon rasgele seçilen iki genin yer değiştirmesi olarak gerçekleştirilir.**


Generation i
X1i 1 1 0 0 f = 36
X2i 0 1 0 0 f = 44
X3i 0 0 0 1 f = 14
X4i 1 1 1 0 f = 14
X5i 0 1 1 1 f = 56
X6i 1 0 0 1 f = 54
Crossover
X6i 1 0 0 1
0 1 0 0 X2i
X1i 1 1 0 0
0 1 1 1 X5i
X2i 0 1 0 0
0 1 1 1 X5i
Mutation
X6'i 1 0 0 0
X2'i 0 1 0 1
X1'i 1 1 1 1
1 0 1 1 X1"i
X5'i 0 1 0 0
X2i 0 1 0 0
0 1 1 0 X2"i
X5i 0 1 1 1
Generation (i + 1)
X1i+1 1 0 0 0 f = 56
X2i+1 0 1 0 1 f = 50
X3i+1 1 0 1 1 f = 44
X4i+1 0 1 0 0 f = 44
X5i+1 0 1 1 0 f = 54
X6i+1 0 1 1 1 f = 56

KODLAMA

Kodlama GA'nın çok önemli bir kısmını oluşturmaktadır.

Probleme GA uygulanmadan önce, verinin uygun şekilde kodlanması gerekmektedir.

Kurulan genetik modelin hızlı ve güvenilir çalışması için bu kodlamanın doğru yapılması gerekmektedir.

Kodlama Çeşitleri

- Permütasyon Kodlama
- Binary Kodlama
- Değer Kodlama
- Ağaç Kodlama

Permütasyon Kodlama

- **Düzenleme problemlerinde kullanılır.**
- **Burada her kromozom, sayıları bir sırada temsil etmektedir.**
- **Permütasyon kodlama, gezgin satıcı ve çizelgeleme problemleri için kullanışlıdır.**

- **Ör:**

Kromozom A	7 8 9 4 1
Kromozom B	8 7 9 1 4

Permütasyon Kodlama

- **Permütasyon kodlama, sıralama problemleri içinde yararlıdır.**

- **Bazı problemlerde bazı çaprazlama ve mutasyon türleri ve kromozomların tutarlılığı için (örneğin içerisinde gerçek sırayı tutan) düzeltmeler yapılması gerekmektedir.**

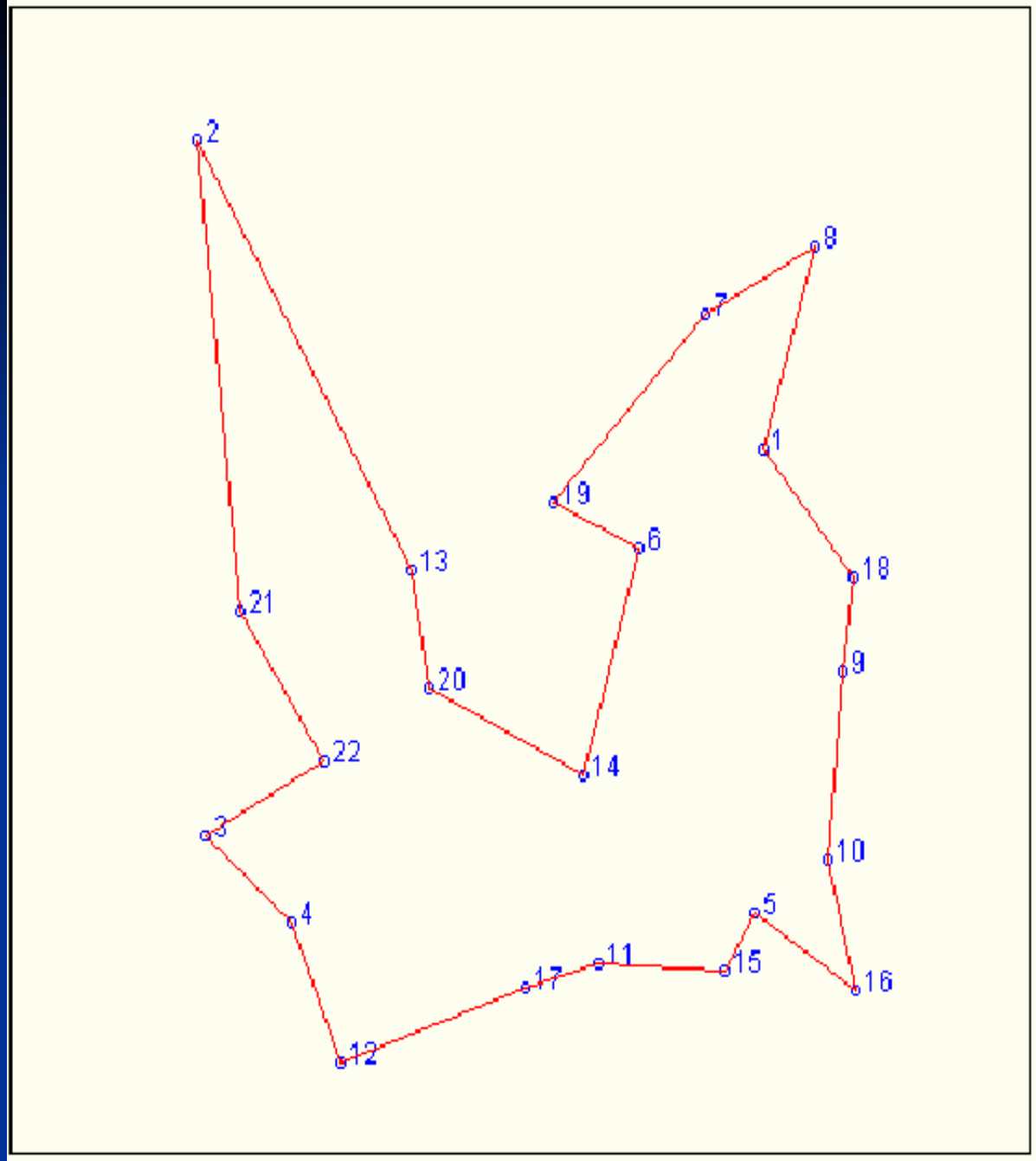
Permütasyon Kodlama

Örnek Problem: Gezgin satıcı problemi

- **Problem: Şehirler ve bu şehirler arasındaki uzaklıklar verilmektedir. Gezgin satıcı tüm bu şehirleri dolaşmak zorundadır. Fakat gereğinden fazla dolaşmamalıdır. En küçük dolaşma uzunluğunu verecek olan şehir dolaşma sırasıyla bulunmalıdır.**
- **Kodlama: Kromozom, gezgin satıcının dolaşacağı şehirlerin sırasını tutar.**

TSP Örnek:

22 şehir için izlenecek en kısa yol.

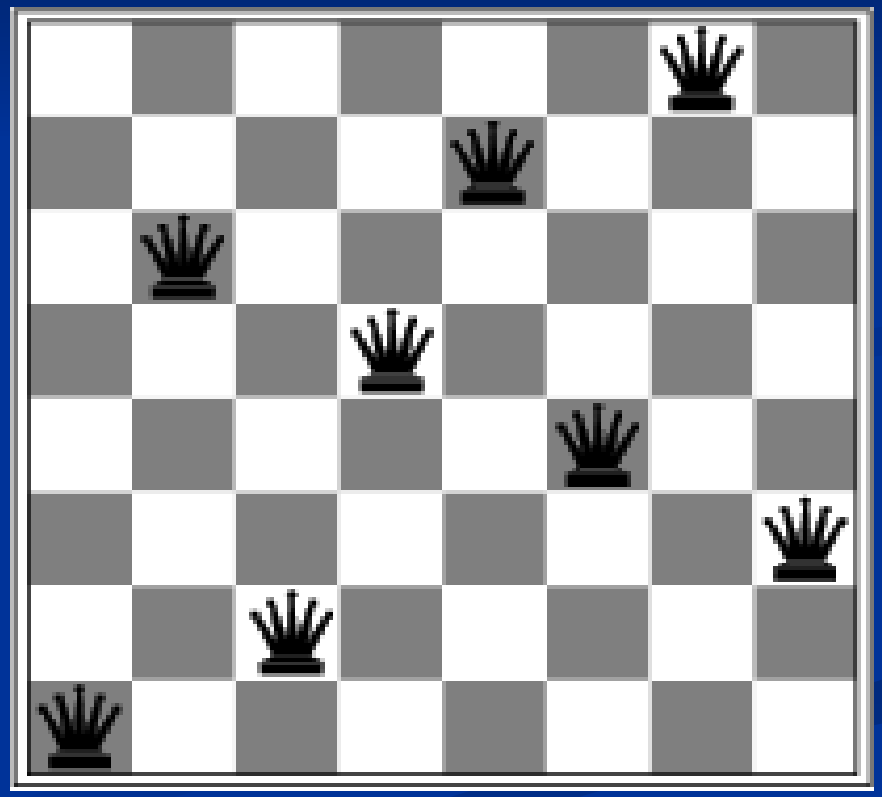


Permütasyon Kodlama

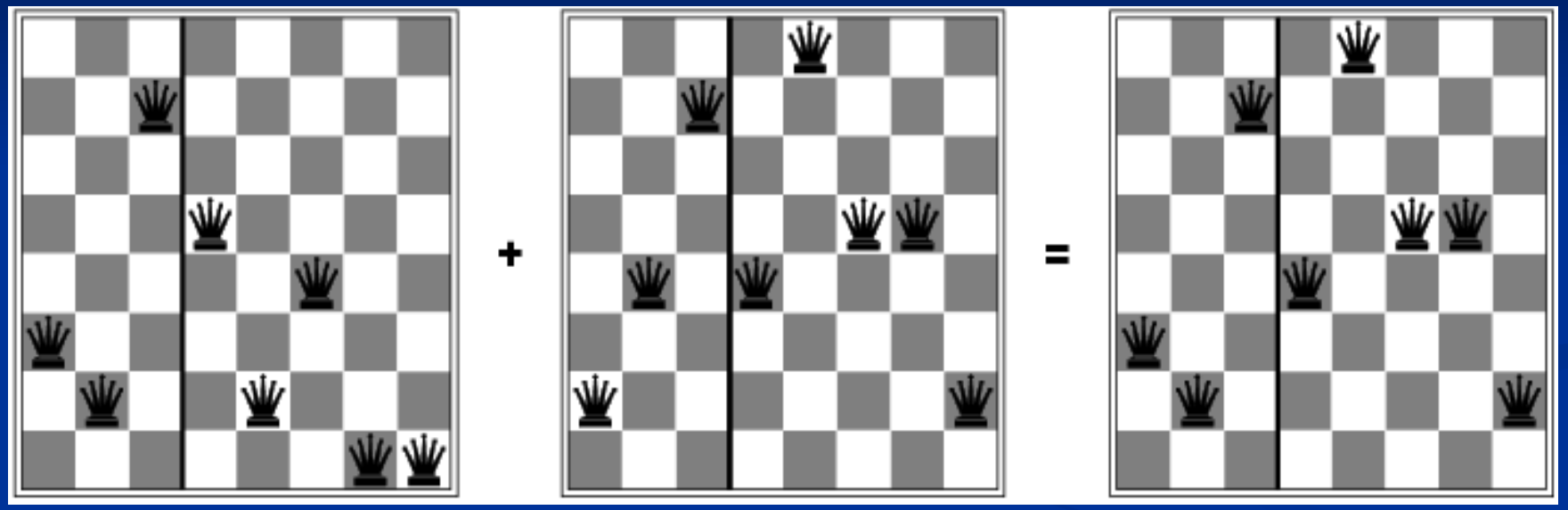
Örnek: 8-vezir problemi

- Kromozom her sütun için vezirin aşağıdan yukarıya kaçıncı karede yer aldığını gösteren 8 tamsayıdan oluşan bir dizidir.

kromozom = 16257483



8-vezir problemi



327 | 52411

247 | 48552

32748552

Binary (İkili) Kodlama

- Her kromozom ikili diziye sahiptir { 0, 1 }
- Bu dizideki her bit, çözümün belli karakteristiğini temsil eder veya tüm dizi bir sayıyı temsil eder.
- Kodlamada en sık kullanılan yöntemdir

Ör: Kromozom A = 10101001

Değer Kodlama

- **Gerçek sayılar gibi karmaşık değerlerin kullanıldığı problemlerde, ikili kodlama zor olduğu için doğrudan değer kodlanması kullanılabilir.**

- **Değer kodlamada, her kromozom bazı değerlere eşittir. Değerler, problemle ilgili herhangi bir şey olabilir. Gerçek sayılar, karakterler veya herhangi nesneler gibi.**

Kromozom A	1.2324 3.5354 4.6465 3.5556
Kromozom B	Doğu, Batı, Güney, Kuzey

- Değer kodlama bazı özel problemler için iyi bir seçimdir. Ancak, bu tip kodlamada probleme özgü yeni çaprazlama ve mutasyon yöntemleri geliştirmek gereklidir.

- *Örnek Problem:* Bir sinir ağı için ağırlıkları bulma

  Problem: Bir sinir ağı belirlenmiş mimariyle birlikte verilmektedir. Ağdan beklenen değeri almak için sinir ağındaki sinirler arasındaki ağırlıklar istenmektedir.

  Kodlama: Kromozomlardaki gerçek değerler sinir ağındaki ağırlıkları temsil eder.

  Çaprazlama yöntemi: İkili kodlamadaki tüm çaprazlamalar kullanılabilir.

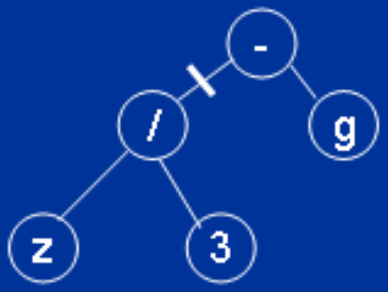
  Mutasyon Yöntemi (Küçük bir sayı ekleme -Gerçek sayı kodlama için-): Seçilen değerlere küçük bir sayı eklenir (veya çıkarılır).
- (1.29 5.68 2.86 4.11 5.55) => (1.29 5.68 2.73 4.22 5.55)

Ağaç Kodlama

- Bu yöntem gelişen, değişen programlar veya ifadeler için kullanılır.
- Ağaç kodlamada her kromozom, bazı nesnelerin (örneğin fonksiyonlar ya da programlama dilindeki komutlar gibi) arasındaki işlemleri içeren bir ağaç yapısından oluşmaktadır.

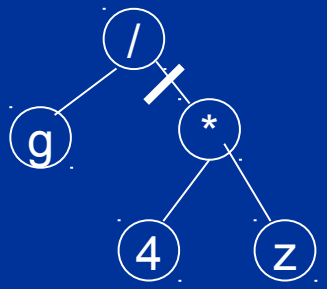
Ör:

Anne



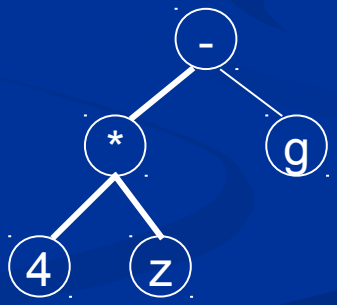
(z/3)-g

Baba



g/(4*z)

Çocuk



(4*z)-g

Elitizm (Seçkinlik)

- Çaprazlama ve Mutasyon yöntemleriyle yeni bir nesil oluştururken, en iyi kromozomları kaybetme olasılığımız vardır.
- Elitizm, en iyi kromozomların (ya da bir kısmının) ilk önce kopyalanıp sonra yeni nesle aktarıldığı yöntemin adıdır.
- Elitizm, GA'nın başarısını hızlı bir şekilde arttırabilir, çünkü bulunan en iyi çözümün kaybolmasını önler.

Genetik Algoritmalarda Parametre Seçimi

- Populasyon büyüklüğü
- Çaprazlama olasılığı
- Mutasyon olasılığı
- Kuşak aralığı
- Seçim stratejisi
- Fonksiyon ölçeklemesi

Örnek soru çözümü:

Amaç: $f(x)=x^2$, $x=[0,31]$ şeklinde verilen bir fonksiyonun, verilen aralıkta maksimizasyonu yapılması istenmektedir.

Adım 1: İlk olarak x sayısının kodlanması işlemi yapılmalıdır. x'in 0 ve 1'lerden oluşan 2 tabanındaki gösterilimi kullanılacaktır. Dolayısıyla x, 5 bit uzunluğunda bir kodla temsil edilecektir. Öyle ki 0: "00000" ve 31: "11111" olacaktır.

Adım 2: Toplumun birey sayısı n:4 olarak seçilmiştir. Toplumu oluşturan dört birey, her biri 5 bit uzunluğunda birer kromozomla temsil edildiği için toplam 20 kere yazı tura atmak suretiyle belirlenmiştir. Elde edilen birey kromozomlar:

Birey 1: 01101, $x = 13$, $x^2 = 169$
Birey 2: 11000, $x = 24$, $x^2 = 576$
Birey 3: 01000, $x = 8$, $x^2 = 64$
Birey 4: 10011, $x = 19$, $x^2 = 361$

Adım 3: Adım 2'de belirlenen bireyler için $f(x)=x^2$, bireylerin uygunluk değerlerini verir.

Dört bireyin toplam uygunluk değerleri "169+576+64+361=1170" dir. Dolayısıyla her bir bireyin rulet tekerleğinde kaplayacağı alan şu şekilde hesaplanır:

Birey 1: 169/1170=0.14 : %14

Birey 2: 576/1170=0.49 : %49

Birey 3: 64/1170=0.06 : %6

Birey 4: 361/1170=0.31 : %31

- Bu değerler, rulet tekerleğinin her çevrilişinde hangi olasılıkla hangi bireyin seçileceğini belirtir, yani 0.14 olasılıkla 1 numaralı birey seçilecektir.
- Rulet tekerleği ve bireylerin tekerlek üzerindeki dağılımları şekilde gösterilmiştir.

Rulet Tekerleği Dağılımı

Birey 1 : 1 kere
Birey 2 : 2 kere
Birey 3 : 0 kere
Birey 4 : 1 kere

Adım 4: Toplumda ki birey sayısının sabit kaldığı varsayıldığından dolayı, rulet tekerleği 4 kere çevrilerek çiftleşme havuzu oluşturulacaktır. Rulet tekerleği döndürülmüş ve bunun sonucunda elde edilen çiftleşme havuzu şu şekildedir:

- Aday 1 : 01101 (Birey 1)
- Aday 2 : 11000 (Birey 2)
- Aday 3 : 11000 (Birey 2)
- Aday 4 : 10011 (Birey 4)

Adım 5:

- Çiftleşme havuzu belirlendikten sonra iki aşamalı çaprazlama uygulanır.
- İlk aşamada adaylar çiftleşmek üzere rasgele olarak eşlenirler.
- Her ikili grup için bir kere zar atılarak çaprazlaşmanın oluşacağı nokta belirlenir. rasgele eşleştirme yapılmış ve bunun sonucunca (Aday 1, Aday 2) ve (Aday 3, Aday 4) ikili grupları oluşmuştur.
- Çaprazlaşma noktaları random olarak 1. Grup için k=4 ve 2. Grup içinde k=2 olarak belirlenmiştir. Bu aşamadan sonra çaprazlaşma gerçekleştirilmiş ve şu sonuçlar oluşmuştur: (çaprazlaşma noktaları “/” ile belirtilmiştir.)

- Çiftleşme grubu 1: (k=4)
  - Aday 1 : 0110/1 oluşan Birey 1 : 01100
  - Aday 2 : 1100/0 oluşan Birey 2 : 11001

- Çiftleşme grubu 2 : (k=2)
  - Aday 3 : 11/000 oluşan Birey 3 : 11011
  - Aday 4 : 10/011 oluşan Birey 4 : 10000

Adım 6:

- Son aşama olan mutasyon bitler düzeyinde uygulanır. Bu örnekte her bir bit için (toplam 20 bit var) mutasyon olma olasılığı 0.01 olarak seçilmiştir.
- Dolayısıyla her bir bit için ağırlıklı yazı/tura (mutasyon olasılığına göre) atılarak hangi bitlerin mutasyona uğrayacağı belirlenir.
- Bu işlem yapılmış ve sonuçta oluşan birey 3'ün 2 numaralı bitinde mutasyon olacağı ortaya çıkmıştır.

Oluşan Birey 3 : 11011

Mutasyon sonucu oluşan Birey 3 : 10011

- Bu adımın tamamlanmasıyla bir sonraki kuşağı oluşturacak toplumun bireyleri belirlenmiş olur. Yeni toplum şu şekildedir:

  - Birey 1 : 01100, $x=12$, $x^2=144$
  - Birey 2 : 11001, $x=25$, $x^2=625$
  - Birey 3 : 10011, $x=19$, $x^2=361$
  - Birey 4 : 10000, $x=16$, $x^2=256$

- Temel operatörden oluşan genetik algoritma her aşamada yeni oluşan kuşağa uygulanarak bir sonraki kuşak elde edilecektir.
- Bu örnekte tek bir iterasyon yapılmış ve başlangıç toplumundan bir sonraki kuşak oluşturulmuştur ancak genetik algoritmanın çalışmasının tam olarak gözlenebilmesi için tek bir iterasyon yeterli değildir.
- Bu örnekteki işlemlerde her şey çok fazla rasgele gibi görünse de, uygunluk değeri yüksek olan bireylerin seçilme ve çiftleşme olasılıkları yüksek olduğu için kuşaklar ilerledikçe toplumu oluşturan bireylerin uygunluk değerlerinin ortalamasının da arttığı gözlenecektir.(Bunun için tek bir iterasyon yeterli değildir.)

Diğer yöntemlerden farkı

- GA problemlerin çözümünü parametrelerin değerleriyle değil, kodlarıyla arar.
- GA, ne yaptığı konusunda bilgi içermez, nasıl yaptığını bilir.
- GA, aramaya tek bir noktadan değil, noktalar kümesinden başlar. Bu nedenle çoğunlukla yerel en iyi çözümde sıkışıp kalmazlar.

Diğer yöntemlerden farkı

- GA türev yerine uygunluk fonksiyonunun değerini kullanır. Bu değerin kullanılması ayrıca yardımcı bir bilginin kullanılmasını gerektirmez.

- GA deterministic kuralları değil olasılıksal kuralları kullanır.

Sonuçlar

- GA, parametre ve sistem tanılama, kontrol sistemleri, robot uygulamaları, görüntü ve ses tanıma, mühendislik tasarımları, planlama, yapay zeka uygulamaları, uzman sistemler, fonksiyon ve kombinasyonel eniyileme problemleri ağ tasarım problemleri, yol bulma problemleri, sosyal ve ekonomik planlama problemleri için diğer eniyileme yöntemlerinin yanında başarılı sonuçlar vermektedir.

A Fast TSP Solver Using GA on JAVA

(Hiroaki SENGOKU , Ikuo YOSHIHARA)

- Genetik algoritmalar geleneksel yöntemlerden (sezgisel arama yöntemleri gibi) herzaman daha yavaş olduğu için hibrid, melez yöntemler denenmiştir.

- Kullanılan bu melez yöntemde genetik algoritmalar 2opt yöntemi ile birleştirilerek kullanılmıştır.

- **Genetik algoritmaları tek başına kullanan Gezgin Satıcı Problemi çözücüleri , hibrid yöntemleri kullanan çözücülerden daha yavaştır.**

- **Bu iki yöntemi karşılaştırmanın en iyi yolu onları aynı makinede çalıştırmaktır.**

2opt Yöntemi

- **Kullanılan çözümde 2opt yöntemi mutasyonu sağlıyor, çaprazlama operatörü ise *yerel minimumun dışına çıkmayı* sağlıyor.**
- **Eğer 2opt yöntemi tek başına kullanılsaydı çözüm yöntemi başarısız olup, *global minimuma ulaşılamazdı.***
- **Bu yüzden çözümde ikisinin de kullanıldığı hibrid bir yönteme başvurulmuştur.**

ALGORİTMANIN ADIMLARI

- **Başlatmak (initialization) : M tane rasgele seçilmiş kişiden oluşan neslin oluşturulması (başlangıç nesil oluşumu)**

- **Doğal Seleksiyon : %Pe kadar kişinin populasyondan ayıklanması,nesilden çıkarılması durumudur.Böylece populasyon M x Pe/100 kadar azalmış olur.**

ALGORİTMANIN ADIMLARI

- **Çaprazlama : M x Pe/100 tane rasgele gen çifti seçilir, seçilen her gen çifti çaprazlanıp çocuk bireyler elde edilir.**

- **2opt Yöntemi ile Mutasyon : Populasyonun içerisinden %Pi kadar rasgele kişi seçilir ve 2opt yöntemi ile mutasyon uygulanarak bu genler iyileştirilir.**

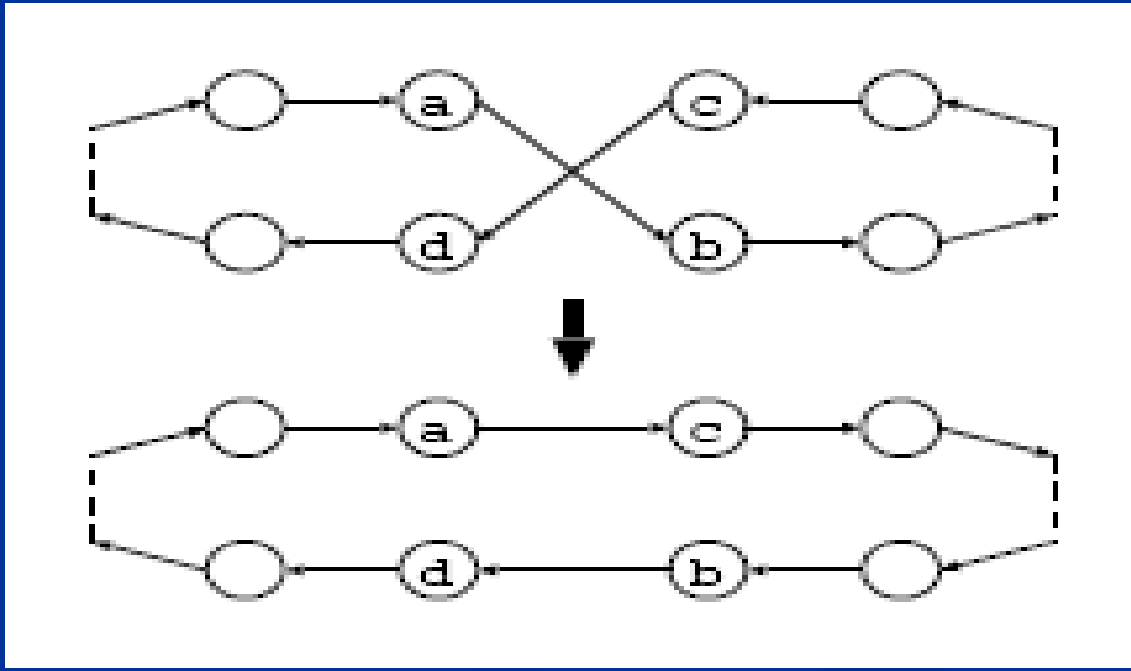
Elit kişi yada en yüksek uygunluk değerine sahip olan kişi her zaman seçilir.

Eğer gen mutasyona uğrayıp iyileştirildi ise hiçbir şey yapılmaz çünkü 2opt yöntemi eğer kombinasyonların en iyisi seçildi ise durur.

2Opt Yönteminin Özellikleri

- **TSP çözücü algoritmalar arasında en fazla bilinen yerel arama algoritmalardan biridir. Köşeden köşeye her bir turu iyileştirir ve alt düzeydeki turların sırasını tersine çevirir.**

- **Şekilde ab ve cd köşeleri kaldırılarak b ile c arasındaki alt düzey turların sırası da değiştirilip, tersine çevrilmiştir.**



- **Böylece yeniden düzenlenmiş turda gidilecek yol miktarı ilk durumdakinden daha azdır. Sebebi ise $ab + cd > ac + bd$ olmasıdır.**

- **Bu prosedür en iyi çözüm bulunana kadar devam eder. Bu süreç tekrar edilerek en üst düzey iyileştirme yapılır ve sonra durulur.**

- **Burada eğer $ac > ab$ ve $bd > cd$ olsa idi köşelere dokunulmaz hiçbir şey yapılmadan geçilirdi.**

Çoğalmak İçin Kullanılan Çaprazlama

- 2opt yöntemi bize yerel minimum değerleri verir. Örneğin çözdüğü herhangi 2 çözüm ayrı yerel minimum değerleri verebilir.

- Önemli olan global minimum a ulaşmaktır. Eğer alt düzey olan bu turların sonuçları birleştirilip kombine edilirse, çıkan bu en iyi değerlerin birleştirilmesiyle global minimum a yakınlaşmış oluruz.

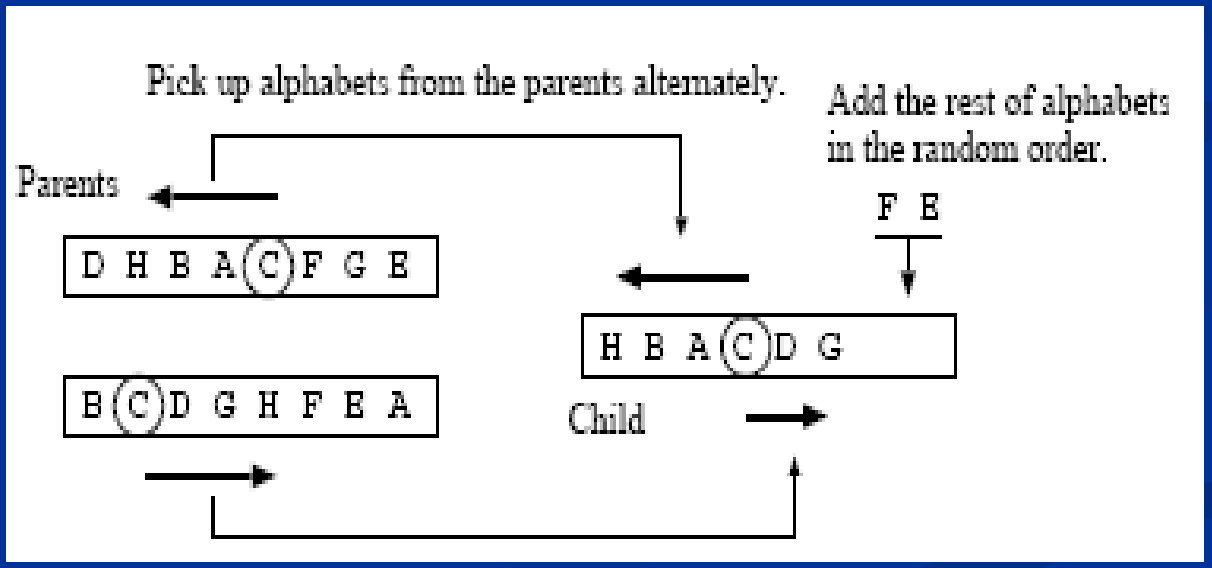
- **Önerdikleri yöntemde kullanılacak olan çaprazlama operatörü ebeveynlerin(parent) alt turlarının en uzun olası sıralamalarını verir.**

- **Bu tarz bir çaprazlamayı "Greedy Subtour Crossover (GSX) " adlandırmışlardır.**

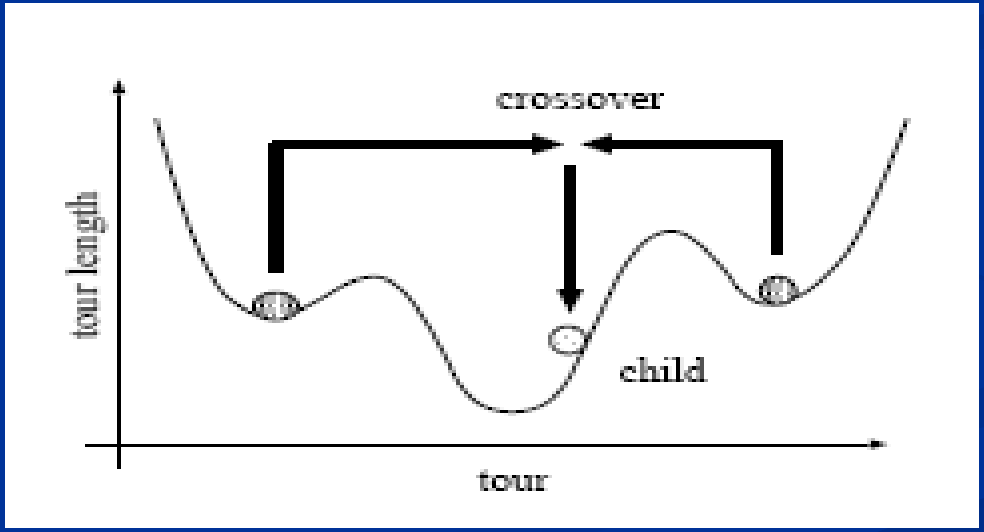
Algorithm: Greedy Subtour Crossover
Inputs: Chromosomes $g_a = (a_0, a_1, \ldots, a_{n-1})$ and $g_b = (b_0, b_1, \ldots, b_{n-1})$.
Outputs: The offspring chromosome g.

```
procedure crossover(g_a,g_b) {
    f_a ← true
    f_b ← true
    choose town t randomly
    choose x, where a_x = t
    choose y, where b_y = t
    g ← t
    do {
        x ← x − 1   (mod n),
        y ← y + 1   (mod n).
        if f_a = true then {
            if a_x ∉ g then {
                g ← a_x · g,
            } else {
                f_a ← false.
            }
        }
        if f_b = true then {
            if b_y ∉ g then {
                g ← g · b_y,
            } else {
                f_b ← false.
            }
        }
    } while f_a = true or f_b = true
    if |g| < |g_a| then {
        add the rest of towns
        to g in the random order
    }
    return g
}
```

- **Örneğin , g kromozomu = (D,H,B,A,C,F,G,E) olsun. Bunun anlamı gezgin satıcı D şehrinden başlayıp bu sırayı takip ederek diğer şehirleri gezip tekrar D şehrine döneceğidir.**



- **Deneyler sonucunda GSX kullanılarak bulunan sonuçların benzetilmiş tavlama (simulated annealing) yöntemi ile bulunan sonuçlara göre yerel minimumun dışına daha etkin bir şekilde çıktığı görülmüştür.**



Doğal Seleksiyon

- M x Pe/100 kadar populasyonda azalma olur. Genetik algoritmalarda hayatta kalım(survival) uygunluk değeri ile doğru orantılıdır. Bir diğer önemli faktör ise populasyondaki çeşitliliktir (diversity).

- Birbirleri ile benzer olan bireyler çeşitliliğin devamını sağlamak için ve genetik algoritmaların bilinen problemlerinden biri olan gelişmemiş yakınsamayı (immature convergence) önlemek için elenir.

Doğal Seleksiyon…

- İlk olarak bireyler uygunluk değerlerine göre sıralanır. Bitişik olan bireylerin uygunluk değerleri karşılaştırılır. Eğer aralarındaki fark ε' dan (küçük pozitif bir gerçel sayı) küçük ise elenen kişilerin sayısı R olana kadar bir önceki bireyler elenir.

- Sonra ise eğer r<R ise R-r kadar kişi en düşük uygunluk değeri sırası uygulanarak elenir.

JAVA APPLET

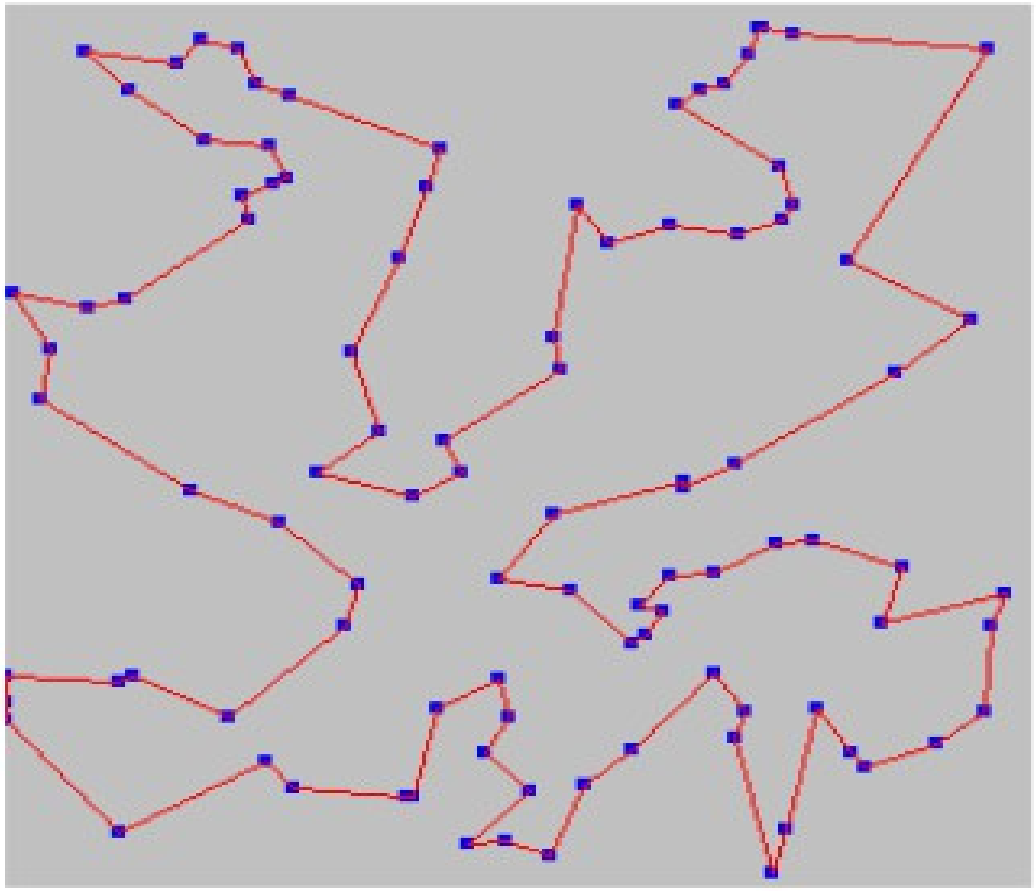
- Geliştirilen JAVA Applet

— http://www.hitachi.co.jp/Div/sdl/e-naiyo/e-senika22/TSP.html
sayfasındadır.

- Applet parametreleri :

✓ Population (M)
✓ Selection (Pe)
✓ 2opt (Pi)

A fast TSP solver using a genetic algorithm



Stop Clear

Generation: 366

Length of Route: 7.3900

# of Cities: 100

Population: 30

Selection [%]: 30

2 opt [%]: 20

Written by Hiroaki Sengoku

(C)1996-7 by Hitachi, Ltd.

Version 1.3

Beğenilen Noktalar

- Kolay ve eğlenceli olması
- Problem çözme sürecinin izlenebilir olması
- Genetik algoritmaların pratik problemlere de uygulanabilirliğinin görülmesi

Eksiklikler

- Grid olmadığından dolayı şehirlerin koordinatlara göre istenilen şekilde yerleştirilememesi
- Önceden tanımlı problemlerin denenememesi ve problemlerin saklanamaması
- Yakınsama grafiğinin gözükmemesi
- Problemin çözümü süresince geçen CPU süresinin gözükmemesi

Kaynaklar

[1]Prof.Dr.Çetin Elmas: "Yapay Zeka Uygulamaları Kitabı", Seçkin Yayın,2007,Ankara

[2]Yrd.Doç.Dr.Şule Gündüz Öğütücü: "Veri Madenciliği", "Farklı Sınıflandırma Yöntemleri",İstanbul Teknik Üniversitesi http://www3.itu.edu.tr/~sgunduz/courses/verimaden/

[3] http://tr.wikipedia.org/wiki/Genetik_algoritma

[4] http://www.yapay-zeka.org/modules/wiwimod/ index.php?page=GA

[5] Yrd.Doç.Dr.Rembiye KANDEMİR: "İleri Programlama Yöntemlerine Giriş-6",ppt.sunusu.

[6] Didem ÖKTEM: "Genetik Algoritmalar",ppt.sunusu.

[7] Göksel ÜÇER: "Genetik Algoritma/Programlama", 2007. http://gokselucer.blogspot.com (goksel.ucer@netsis.com.tr)

[8] Sengoku, H. , Yoshihara, I. : A Fast TSP Solver Using GA on JAVA