

# Genetik Algoritmalar

Mehmet Fatih AMASYALI Yapay Zeka Ders Notları

YILDIZ TEKNİK ÜNİVERSİTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

## Genetik Algoritmalar

- Genetik algoritmalar, evrim mekanizmasına (en iyinin yaşaması) dayanır.
- Amaç, uygunluk fonksiyonunun (fitness function) maksimizasyon / minimizasyon optimizasyonudur.
- Uygunluk fonksiyonu: Toplumdaki her kromozomun / birey / durum ne kadar iyi olduğu gösteren fonksiyondur. Büyük değer, iyi / uygun birey.
- K adet rasgele üretilmiş kromozom / durum / çözüm hipotezi ile başlar. Durumlar kümesine popülasyon denir.
- Kromozom / durumlar sıralı ya da sırasız karakter kümeleri ile temsil edilir. Genelde sayılar ya da 0 ve 1'lerden oluşur.

Mehmet Fatih AMASYALI Yapay Zeka Ders Notları

YILDIZ TEKNİK ÜNİVERSİTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

# Genetik Algoritmalar

Yeni kromozom / durumların üretilmesinde kullanılan operatörler;

- Seçme (selection) : Bir sonraki nesli üretecek kromozomların seçilmesidir.
- Yeniden kopyalama (reproduce) : Yeni çözümler üretmek için çaprazlama (crossover) işlemi yapılır.
- Değiştirme (mutation) : Kromozom'un bazı değerlerini rasgele değiştirir.

# Genetik Algoritma\*

```

function GENETIC_ALGORITHM( population, FITNESS-FN) return an individual
  input: population, a set of individuals
         FITNESS-FN, a function which determines the quality of the individual
  repeat
    new_population  $\leftarrow$  empty set
    loop for i from 1 to SIZE(population) do
      x  $\leftarrow$  RANDOM_SELECTION(population, FITNESS_FN)
      y  $\leftarrow$  RANDOM_SELECTION(population, FITNESS_FN)
      child  $\leftarrow$  REPRODUCE(x,y)
      if (small random probability) then child  $\leftarrow$  MUTATE(child)
      add child to new_population
    population  $\leftarrow$  new_population
  until some individual is fit enough or enough time has elapsed
  return the best individual
  
```

[\*] <https://github.com/aimacode/aima-pseudocode/blob/master/md/Genetic-Algorithm.md>

## Paralel Tepe Tırmanma - Local beam search

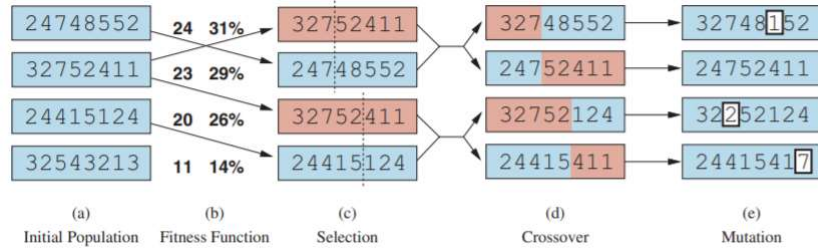
- Ana Fikir: Tek bir durumu izlemek yerine K taneyi izle
- K adet rasgele üretilmiş durumla başla
- Her bir iterasyonda k durumun hepsiden gidilebilecek durumları üret.
- Bu durumlardan biri hedefse dur. Değilse, en iyi k tanesini mevcut durumlar olarak ata ve bir önceki adıma dön.

GA vs. Paralel Tepe Tırmanma?

## Genetik Algoritma iş başında 8 vezir

- 8 vezirde kromozomların ifade edilmesi:
- Herbir sütunda bir vezir var.
- State = her bir vezirin sütündaki yeri  
=>  $8 \times \log_2(8) \text{ bits} = 24 \text{ bits}$  (ikili gösterimde)

## Genetik Algoritma iş başında 8 vezir\*



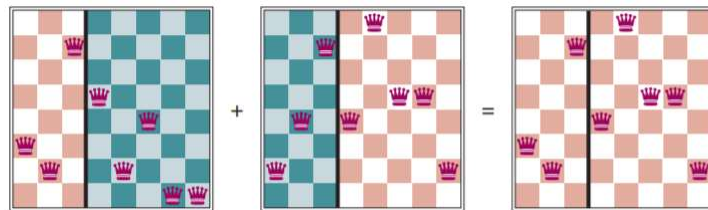
- Uygunluk fonksiyonu - Fitness function: Birbirini tehdit etmeyen vezir çiftlerinin sayısı (min = 0, max =  $8 \times 7/2 = 28$ )
- $24/(24+23+20+11) = 31\%$
- $23/(24+23+20+11) = 29\%$  vs.

[\*] <http://aima.cs.berkeley.edu/figures.pdf>

Mehmet Fatih AMASYALI Yapay Zeka Ders Notları

YILDIZ TEKNİK ÜNİVERSİTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

## 8 vezirde çaprazlama\*



[\*] <http://aima.cs.berkeley.edu/figures.pdf>

Mehmet Fatih AMASYALI Yapay Zeka Ders Notları

YILDIZ TEKNİK ÜNİVERSİTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

## Kromozom Seçim Türleri

1. Rulet tekeri seçimi
  - Bireylerin (kromozomların) uygunluk değerine göre seçim.
  - Fakat uygunluk değeri en büyük olanın seçileceği garanti değil, seçilme şansı fazla.
  - Her bir kromozomun seçilme şansı, kendi uygunluk değerinin tüm bireylerin uygunluk değerlerinin toplamına oranıdır.
2. Sıralama seçimi
  - Rulet tekeri seçimi, uygunluklar çok farklıysa problemlere yol açar. (Örneğin, en iyi kromozomun seçilme şansı %90 ise diğer kromozomlar çok az seçilme şansına sahip)
  - Sıralama seçimi önce popülasyonu uygunluk değerlerine göre küçükten büyüğe sıralar. Her kromozomun sıra değeri uygunluk değeri olarak atanır. En kötü 1, en iyi N. Seçilme şansı yine uygunluk değerinin toplam uygunluk değerine oranı ile bulunur.
  - Böylelikle bütün kromozomlara seçilme şansı verilir. Fakat bu yöntemde en iyi kromozomların seçilme şansı diğerlerinden çok farklı olmadığından çözüme yaklaşma yavaş olacaktır.

## Farklı Olasılıklarla Seçim?

- Seçilme şansları toplamı=100
- Bir aralığı 100 parçaya böl.
- Seçilme şanslarına göre parçaları kromozomlara ata.
- 1-100 arasında rasgele bir sayı üret.
- O sayının karşı geldiği aralığın atandığı kromozomu seç.
- *Seçilme şansları %44.34355 şeklinde olursa?*

## Gezgin Satıcı Problemi Traveling Salesman Problem

Şehirler arası bir rota bulmak

- Rota her şehirden geçmeli
- Toplam gidilen yol minimum olmalı

## Gezgin Satıcı Probleminde Kromozomların Gösterimi

Şehirlerin sıralı listesi. 8 şehir varsa:

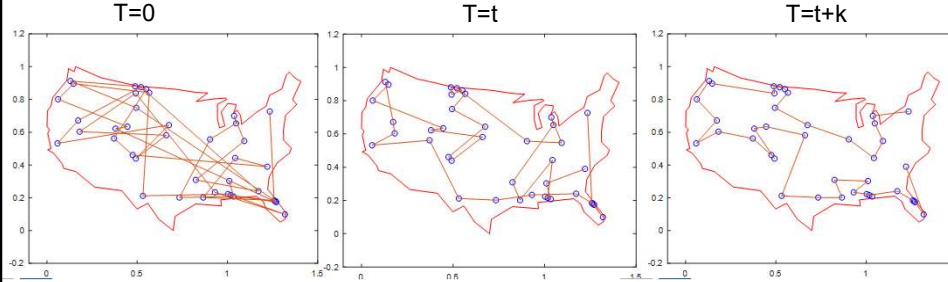
Olası iki kromozom / çözüm hipotezi

CityList1 (3 5 7 2 1 6 4 8)

CityList2 (2 5 7 6 8 1 3 4)

## Matlab TSP demo

40 rasgele nokta  
Temsil: 40 noktanın indisi  
 $T=0$



Mehmet Fatih AMASYALI Yapay Zeka Ders Notları

YILDIZ TEKNİK ÜNİVERSİTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

## Crossover'da Problem

Olası iki kromozom / çözüm hipotezi

CityList1 (3 5 7 | 2 1 6 4 8)

CityList2 (2 5 7 | 6 8 1 3 4)

Standart crossover uygularsak:

Yeni nesil:

CityList1 (**3** 5 7 6 8 1 **3** 4)

CityList2 (**2** 5 7 **2** 1 6 4 8)

Gezgin satıcı için bu problem.

Çözüm: ?

Mehmet Fatih AMASYALI Yapay Zeka Ders Notları

YILDIZ TEKNİK ÜNİVERSİTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

## Uniform Crossover

Kromozom boyutunda rastgele 1 ve 0'lardan oluşan template oluştur.

Template'deki 0'lar için P1 ve P2'nin yerlerini değiştir.  
1'ler için aynen al.

P <sub>1</sub>	A	B	C	D	E	F	G	H	I	J
P <sub>2</sub>	E	F	J	H	B	C	I	A	D	G
T	1	0	1	1	0	0	0	1	0	1

C <sub>1</sub>	A	F	C	D	B	C	I	H	D	J
C <sub>2</sub>	E	B	J	H	E	F	G	A	I	G

Gezgin satıcıya çözüm mü?

## Sıra Tabanlı Crossover Order-based Crossover

Kromozom boyutunda rastgele 1 ve 0'lardan oluşan template oluştur.

Child1 için:

Template'de 1 olan indislerin Parent1'deki değerlerini al. Child1'in aynı indisine koy. Child1'de bazı boşluklar kalır.

Template'de 0 olan indislerin Parent1'deki değerlerini al. (A D E G)

Bu listedekileri Parent2'deki sıralarına göre diz. (E D G A)

Elde edilen diziyle Child1'in boşluklarını bu sırayla doldur.

Child2 için:

Aynı prosedür (E C F A) → (A C E F)

Parent 1	A	B	C	D	E	F	G
Parent 2	E	B	D	C	F	G	A
Template	0	1	1	0	0	1	0

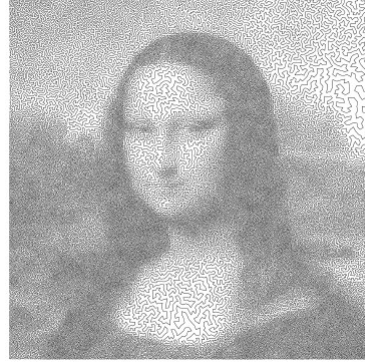
Child 1	E	B	C	D	G	F	A
Child 2	A	B	D	C	E	G	F



## Mona Lisa TSP\*

100 bin noktayı elini kaldırmadan birleştir  
En kısa yol?

Bir çözümün en iyi çözüm olduğunu  
bilmek mümkün mü?



[\*] <http://www.math.uwaterloo.ca/tsp/data/ml/mona-lisa100K.gif>

Mehmet Fatih AMASYALI Yapay Zeka Ders Notları

YILDIZ TEKNİK ÜNİVERSİTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

## GA ile fonksiyon optimizasyonu

**Problem:**  $f(x) = \cos(x/3) * \sin(x/4)$   $x \in [0, 127]$  şeklinde verilen bir fonksiyonun maksimum değerini bulmak

**Gösterim :** 128 farklı değer için 7 bit yeterli

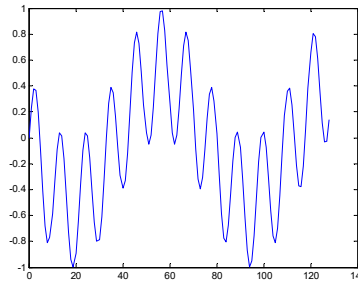
**Başlangıç :** a : 0000011

b : 1110111

c : 0010000

d : 0011010

**Uygunluk kriteri :**  $f(x)$



Fonksiyon iki parametrelili olursa?

$$F(x,y) = x^2 - y^3 + xy^2$$

Mehmet Fatih AMASYALI Yapay Zeka Ders Notları

YILDIZ TEKNİK ÜNİVERSİTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

## GA ile 8 Taş Çözüm yolunu da bulmak

4	3	6
2	1	8
7		5

$S_0$

1	2	3
4	5	6
7	8	

G

Operatörler:

L : Boş kare sola

R : Boş kare sağa

U : Boş kare yukarıya

D : Boş kare aşağıya

GA ile çözüm:

Kromozom gösterimi? (L,D,R,U,U,...)

Kromozom uzunluğu? (önceden belirlenmeli, iteratif artırılabilir)

Uygunluk fonksiyonu?

(operatörler sırayla uygulandığında oluşan durumun hedefe uzaklığı)

Çaprazlama ve mutasyon nasıl yapılmalı?

Mehmet Fatih AMASYALI Yapay Zeka Ders Notları

YILDIZ TEKNİK ÜNİVERSİTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

## GA'nın Performansını Etkileyen Faktörler

- Popülasyon büyüklüğü / Kromozom sayısı: Kromozom sayısını arttırmak çalışma zamanını artırırken, azaltmak da kromozom çeşitliliğini yok eder.
- Mutasyon Oranı: Kromozomlar birbirine benzemeye başladığında hala çözüm noktalarının uzağında bulunuyorsa mutasyon işlemi GA'nın sıkıştığı yerden (tüm kromozomlar aynı platoda) kurtulmak için tek yoldur. Ancak yüksek bir değer vermek GA'ın kararlı bir noktaya ulaşmasını engelleyecektir.
- Kaç Noktalı Çaprazlama Yapılacağı: Normal olarak çaprazlama tek noktada gerçekleştirilmekle beraber yapılan araştırmalar bazı problemlerde çok noktalı çaprazlamanın çok yararlı olduğunu göstermiştir.
- Çaprazlamanın sonucu elde edilen bireylerin nasıl değerlendirileceği: Elde edilen iki bireyin birden kullanılıp kullanılmayacağı.

Mehmet Fatih AMASYALI Yapay Zeka Ders Notları

YILDIZ TEKNİK ÜNİVERSİTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

## GA'nın Performansını Etkileyen Faktörler

- Durum kodlanmasının nasıl yapıldığı: Bir parametrenin doğrusal yada logaritmik kodlanması GA'nın performansında önemli bir farka yol açabilir.
- Başarı değerlendirmesinin nasıl yapıldığı: Akıllıca yazılmamış bir değerlendirme işlevi, çalışma zamanını uzatabileceği gibi çözüme hiçbir zaman ulaşmamasına da neden olabilir.

## GA - Değerlendirmeler

- Pozitif yönleri
  - Crossover ve mutasyon sayesinde lokal maksimum ve platolardan kurtulabilir.
  - Doğa temelli
- Negatif yönleri
  - Ayarlanabilir çok fazla parametreye sahip
    - Bir problemde gösterdiği başarıyı başka bir problemde tekrarlaması zor
  - Bazı problem türleri için iyi ama tepe tırmanmayı farklı başlangıçlarla tekrarlamaktan ya da paralel tepe tırmanmadan daha iyi olduğuna dair matematiksel bir kanıt yok.

## Uygulama: bir alanı tarama

- Amaç:  $N \times N$  lik bir alanda 8 yöne hareket,  $N \times N - 1$  adet hareket, minimum açıda dönüşle maksimum alanı gez
- Hareketler arası açı miktarı az olsun, yumuşak dönüşler yapsın.
- İyilik fonksiyonu 2 bileşene sahip:  $\min(\text{açı})$  ve  $\max(\text{alan})$
- Temsil:  $(N \times N - 1)$  adet 0-8 arası yönleri belirten sayılar

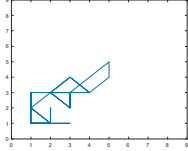
## Genetik Algoritma ile

**genetik\_tarama\_1robot.m**

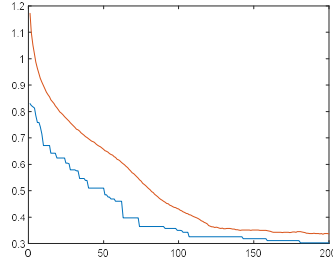
- Rasgele P adet çözümle başla
- G kez:
  - Çözümlerden çaprazlama ve mutasyonla yeni P adet yeni çözüm üret
    - Çaprazlama 2 noktadan, mutasyon %mu kadar
  - Üretilen çözümlerin iyilik değerlerini hesapla
    - Dönüş açılarını topla, gidilen farklı nokta sayısını topla
  - Yeni nesil: eski neslin bir kısmı + yeni neslin bir kısmı

## Bulunan çözümlerden biri

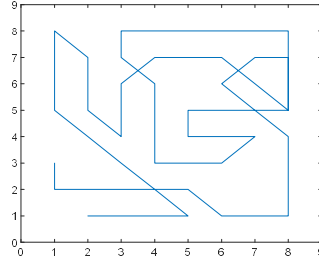
Rasgele  
başlangıçlardan biri



Sürecin ilerleyişi:  
Kırmızı popülasyonun ort.  
Mavi en iyi birey



Bulunan en iyi çözüm



P=5000; % popülasyon büyüklüğü  
mu=0.01; % mutasyon oranı  
G=200; % nesil sayısı  
cross=2; % 2 noktadan çaprazlama  
Yeni nesil: eski ve yeni neslin en iyi yarıları

7	7	7	7	2	2	2	2	1
1	1	6	5	5	6	1	1	8
7	7	6	6	6	7	1	1	1
3	3	3	3	3	5	6	5	5
5	7	7	8	3	3	1	7	7
7	1	1	3	4	6	6	5	5
5	3	3	2	3	3	3	3	1

## Uygulama: 2 nokta arası yol bulma

- Amaç:  $N \times N$  lik bir alanda verilen iki nokta arası yol bulmak
- İyilik fonksiyonu 3 bileşene sahip:
  - min(açı)
  - max(alan)
  - max(sonun hedefe yakınlığı)

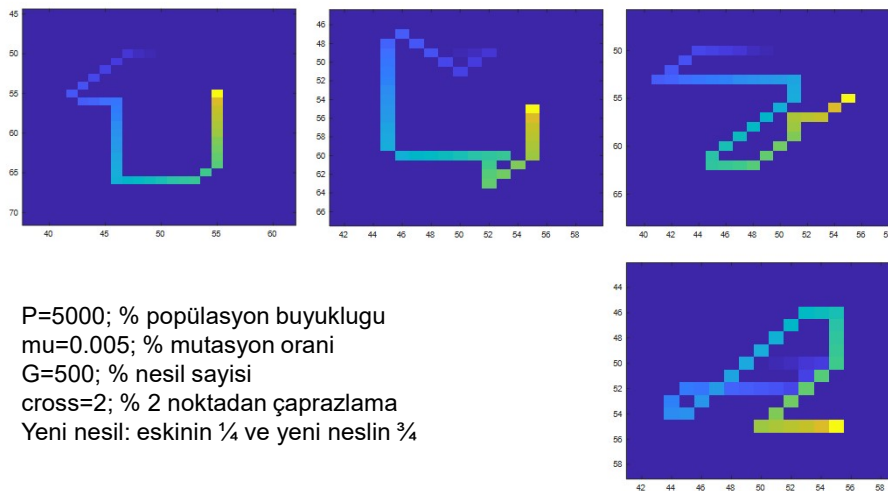
## Genetik Algoritma ile **genetik\_path\_bulma.m**

- Rasgele P adet u uzunluğunda çözümle başla
- G kez:
  - Çözümlerden çaprazlama ve mutasyonla yeni P adet yeni çözüm üret
    - Çaprazlama 2 noktadan, mutasyon %mu kadar
  - Üretilen çözümlerin iyilik değerlerini hesapla
    - Dönüş açılarını topla, gidilen farklı nokta sayısını topla, hedefe yakınlığı ölç
  - Yeni nesil: eski neslin bir kısmı + yeni neslin bir kısmı

Mehmet Fatih AMASYALI Yapay Zeka Ders Notları

YILDIZ TEKNİK ÜNİVERSİTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

## Bulunan çözümlere örnekler



Mehmet Fatih AMASYALI Yapay Zeka Ders Notları

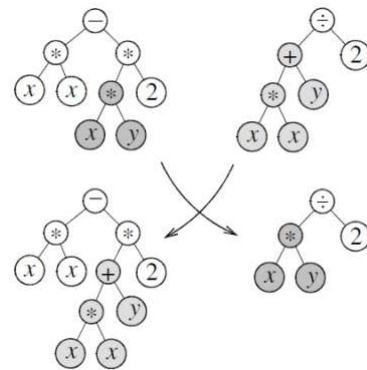
YILDIZ TEKNİK ÜNİVERSİTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

## İyilik fonksiyonları (objective / fitness functions) nasıl birleştirilir?

- Ölçekleri aynı değilse etkilerini farklılaştırır, normalize etmek gerekir
- Min  $\leftrightarrow$  Max dönüşümleri ?
- Önem dereceleri farklı ise ağırlıklı işlem gerekir
- Toplama
- Çarpma
- Maksimum alma
- Minimum alma

## Genetik Programlama

- Genetik algoritma değil ☺
- Programlama dillerindeki her geçerli program bir parse tree ile ifade edilebilir. Elimizde n adet program/ağaç olsa, bunları kullanarak yeni programlar üretebiliriz.
- Program program yazacak ☺
- En basiti, bir kelime bir işlemin işlem kısmı: Şu sayıları kullanarak şu sayıyı üretin
- Yine crossover, mutasyon ile yeni ağaçlar üretiliyor.



## Genetik Programlama

- Biraz zoru: 10 elemanlı herhangi bir diziyi sıralayacak kodu üretmek
- Biraz daha zoru: N elemanlı bir diziyi sıralayan kodu üretmek
- En zoru: herhangi bir görev tanımını koda dönüştürmek
- Soru(n)lar:
- Uygunluk fonksiyonu: K adet görevden (şu giriş için çıkış şu olmalı ikilileri) kaçını başardığı
- 10 görevin 7'sini başaran 9'unu başarana göre daha mı kötü?
- Küçük değişimlerin büyük etkileri:  $< \rightarrow >$  çok şey değiştirir
- Aynı işi yapan kodlar arası karşılaştırma?
- Gereksiz kod blokları

[\*] <http://math.gsu.edu.tr/hayral/>

Mehmet Fatih AMASYALI Yapay Zeka Ders Notları

YILDIZ TEKNİK ÜNİVERSİTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

## Kaynaklar

- <http://aima.cs.berkeley.edu/figures.pdf>
- <https://github.com/aimacode/aima-pseudocode/blob/master/algorithms.pdf>

Mehmet Fatih AMASYALI Yapay Zeka Ders Notları

YILDIZ TEKNİK ÜNİVERSİTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ