SELÇUK ÜNİVERSİTESİ
KONYA - 1975

YAPAY ZEKA

DR. ÖĞR. ÜYESİ TAHİR SAĞ

BİLGİSAYAR MÜHENDİSLİĞİ

KONYA, 2021

CROSSOVER OPERATORS ≡ ÇAPRAZLAMA OPERATÖRLERİ

Eşleme sürecinde, seçilen kromozomlardan bir ve birden fazla yeni nesil oluşturma olayına "çaprazlama" denir.

Kodlama türüne göre çok sayıda farklı çaprazlama operatörü önerilmiştir.

Binary Kodlamada en yaygın olarak kullanılan iki kromozomdan iki tane yeni nesil elde edilmesini sağlayan tek-noktalı-çaprazlama operatörüdür.

p_c çaprazlama oranı olmak üzere $p_c = 0.25$ alındığını varsayalım.

Bu oran; J popülasyon boyutunu göstermek üzere $J\ x\ p_c$ adedince aday çözümün çaprazlamaya tabi tutulacağını kontrol altında tutmayı sağlar.

Bu amaçla popülasyondaki her bir kromozom için [0, 1] aralığında random bir ρ sayısı üretilir.

Eğer ρ sayısı < p_c ise kromozom çaprazlama için seçilir.

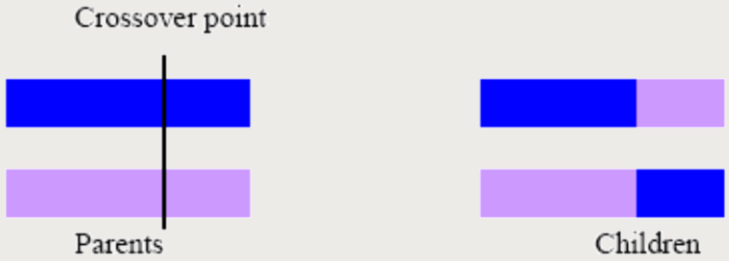
Ele alınan örnek için aşağıdaki ρ sayılarının random üretildiğini varsayalım:

$$\begin{bmatrix} 0.823 & 0.152 & 0.202 & 0.625 & 0.314 \\ 0.347 & 0.917 & 0.519 & 0.401 & 0.607 \end{bmatrix}$$

v_1' = [00100111010100111010111100]
v_2' = [10001110101100110111011100]
v_3' = [01100111101101110011101011]
v_4' = [11100110101110010101011110]
v_5' = [00100111010100111010111100]
v_6' = [10101110010110011001111010]
v_7' = [10001110101100110111011100]
v_8' = [01100111101101110011101011]
v_9' = [00001010100110111101111110]
v_{10}' =[01100111101101110011101011]

v_2' ve v_3' çaprazlama için seçilir.

3



Kromozom 25 bit olduğu için [1, 24] aralığında random bir çaprazlama noktası p_p seçilir.

$p_p = 2$ seçildiğini varsayarak;

$\mathbf{v}_2' = [\ 10|0011101011001101011100]$
$\mathbf{v}_3' = [\ 01|1001111011011001101011]$

$\mathbf{v}_2' = [\ 10|1001111011011001101011]$
$\mathbf{v}_3' = [\ 01|0011101011001101011100]$

Eğer $p_c = 0.5$ alınmış olsaydı, 5 kromozom çaprazlama için seçilecekti.

Bu durumda tek sayıda kromozom seçilmiş olacağı için 2 durum söz konusu olur:

(1) ya ekstra bir kromozom daha eklenir,

(2) ya da seçilen bir kromozom seçimden çıkarılır.

Bunun kararı da yine random bir şekilde verilir.

4

Gerçek kodlu basit çaprazlama

14	10	14	1
5	1	5	10

Aritmetik Parametre Noktalı Çaprazlama

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

Parents before crossover

	x_1	x_2	x_3	x_4	x_5
Parent 1	1,463	2,050	0,389	4,255	0,744
Parent 2	4,023	2,819	4,180	3,436	2,291

Offspring after crossover

	x_1	x_2	x_3	x_4	x_5
Offspring 1	1,463	2,050	0,389	3,746	0,744
Offspring 2	4,023	2,819	4,180	3,945	2,291

Randomly choosen crossover variable for a=0,379

MUTATION OPERATORS ≡ MUTASYON OPERATÖRLERİ

Lokal ekstremum noktalara takılmayı önlemek ve popülasyona ekstra çeşitlilik kazandırmak amacıyla kullanılır. (*exploration*)

Kromozomdaki bitlerin küçük bir yüzdesini değiştirir.

Mutasyon ile kromozomdaki bitler "1" ise "0", "0" ise "1" yapılır.

p_m mutasyon oranı olmak üzere $p_m = 0.01$ alındığını varsayalım.

Bu oran; bitlerin %1'inin *mutasyona* tabi tutulacağı anlamına gelir.

Ele aldığımız örnekte popülasyondaki toplam bit sayısı 250'dir.

Bu nedenle her jenerasyonda ortalama 2.5 adet mutasyon olması beklenir. Daha doğru bir ifadeyle 2 ya da 3 mutasyon.

Bu amaçla popülasyondaki her bir bit için [0, 1] aralığında random bir ϱ sayısı üretilir.

Eğer ϱ sayısı < p_m ise bit mutasyona uğratılır.

Mutasyon sonrası elde edilen yeni popülasyon yan tarafta görülmektedir.

Bit position	Random Number	Index of Chromosome	Number of bit
109	0.003	5	9
147	0.005	6	22
195	0.007	7	20

$v_1' = [\,0010011101010011101011100]$
$v_2' = [\,1010011110110111001101111]$
$v_3' = [\,0100111010110011011011000]$
$v_4' = [\,1110011010111001010101110]$
$v_5' = [\,0010011111010011101011100]$
$v_6' = [\,1010111001011001100110010]$
$v_7' = [\,0101110010110100000111010]$
$v_8' = [\,0110011110110111001101011]$
$v_9' = [\,0000101010011011110111110]$
$v_{10}' = [\,0110011110110111001101011]$

MUTATION OPERATORS ≡ MUTASYON OPERATÖRLERİ

$v_1' = [\,0010011101010011101011100]$
$v_2' = [\,1010011110110111001101111]$
$v_3' = [\,0100111010110011011011000]$
$v_4' = [\,1110011010111001010101110]$
$v_5' = [\,0010011111010011101011100]$
$v_6' = [\,1010111001011001100110010]$
$v_7' = [\,0101110010110100000111010]$
$v_8' = [\,0110011110110111001101011]$
$v_9' = [\,0000101010011011110111110]$
$v_{10}' = [\,0110011110110111001101011]$

$f(-0.885\ ,\ 1{,}150\ ) = 44.308$
$f(\ 1.121\ ,\ 2{,}256) = 41.692$
$f(-0.270\ ,\ 1{,}069) = 42.706$
$f(\ 2.105\ ,\ 2{,}919) = 39.250$
$f(-0.878\ ,\ 1{,}150\ ) = 44.294$
$f(\ 2.900\ ,\ 1{,}124) = 41.085$
$f(\ -0.051\ ,\ 1{,}285) = 41.940$
$f(\ 0.121\ ,\ 2{,}253) = 38.911$
$f(-1.335\ ,\ 3{,}710) = 31.677$
$f(\ 0.121\ ,\ 2{,}253) = 38.911$

Bu popülasyon için x1 ve x2 parametrelerinin reel değerleri ve her bir kromozomun uygunluk değeri verilmiştir.

Bu adım itibariyle GA'nın ilk jenerasyonu tamamlanır.

Aynı prosedür her yeni jenerasyon için sonlanma kriteri sağlanıncaya kadar tekrar eder.

- Sonlanma Kriterleri genel olarak 3 grupta toplanabilir:
  - Jenerasyon Sayısı
  - Maksimum fonksiyon değerlendirme sayısı (MaxFes)
  - Epsilon Değeri

Binary nonuniform mutation. This operator, also called dynamic mutation, was proposed by Janikow & Michalewicz (1991) and it is designed for fine-tuning capabilities for the real number representation of chromosomes. It can also be used for binary representation and in this case the probability of mutation of the gene is evaluated using the following formula:

$$R_m = p_m \cdot p(t,n)$$

where:

p_m - assumed probability of uniform mutation,

$p(t,n)$ – dynamic operator, which can be evaluated using different functions. One of the forms of this function is:

$$p(t,n) = a - \left(\frac{a}{1 + b \cdot e^{-c \cdot \left(t - \frac{n \cdot T}{N} \right)}} \right)$$

where:

a,b,c – assumed parameters,

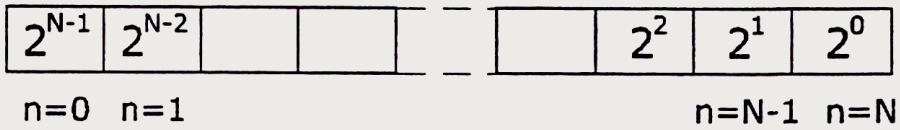
n - bit position in the gene,

t – generation number,

T- maximal number of generations,

N – number of all bits in the variable.

Graphical illustration of the bit position in the gene (variable) is presented below.



Real number uniform mutation. For the real number representation the mutation operators are different from the binary representation. A real number gene is mutated in the assumed range. Assuming that the chromosome $\mathbf{x} = [x_1, x_2, ..., x_N]^T$ is to be mutated we first generate a random number $k \in [1,N]$ and then produce the offspring $\mathbf{x}' = [x_1, x_2, ..., x_k', ..., x_N]^T$ where x_k' is evaluated from the formula:

$$x_k' = x_k^l + \rho_k \cdot (x_k^u - x_k^l) \tag{3.9}$$

where ρ_k is the random number from the domain [0,1] and x_k^l and x_k^u are typically lower and upper bounds on the variable x_k.

Michalewicz et al. (1994) proposed the so called *boundary mutation* in which the gene x_k' can be replaced by either x_k^l or x_k^u, each with equal probability.

Real number nonuniform mutation. In this type of mutation the gene x_k' is evaluated from the formula:

$$x_k' = \begin{cases} x_k + \Delta(t, x_k^u - x_k) & \text{for random number 0} \\ x_k - \Delta(t, x_k - x_k^l) & \text{for random number 1} \end{cases} \tag{3.10}$$

The function $\Delta(t,y)$ returns a value which is in the range $[0,y]$ and has the property that the probability of $\Delta(t,y)$ being close to zero increases as t increases, where t is the generation number. Quite often the function $\Delta(t,y)$ has the following form:

$$\Delta(t, y) = y * r * (1 - t/T)^b \tag{3.11}$$

where r is the random number from the domain [0,1], T is the maximal number of generations and b is the assumed parameter determining the degree of dependency on the generation number (we used $b = 2$).

KAPSAMLI BİR GA ÖRNEĞİ: Değerlendirme

9

$$\textit{Objective Function}$$

$$f(x_1, x_2) = 40 - \frac{9}{2}x_1 + 4x_2 - x_1^2 - 2x_2^2 + 2x_1x_2 - x_1^4 + 2x_1^2x_2$$

$$-1.5 \le x_1 \le 2.5$$

$$0 \le x_2 \le 5$$

Başlangıç Popülasyonunun toplam uygunluk değeri F=367.549

Birinci jenerasyon sonunda elde edilen yeni popülasyonun toplam uygunluk değeri F=404.774

GA'nın 200 jenerasyon sonunda elde ettiği değerler:

f(x1,x2)=44.512

x1=-1.037 ve x2=1.022

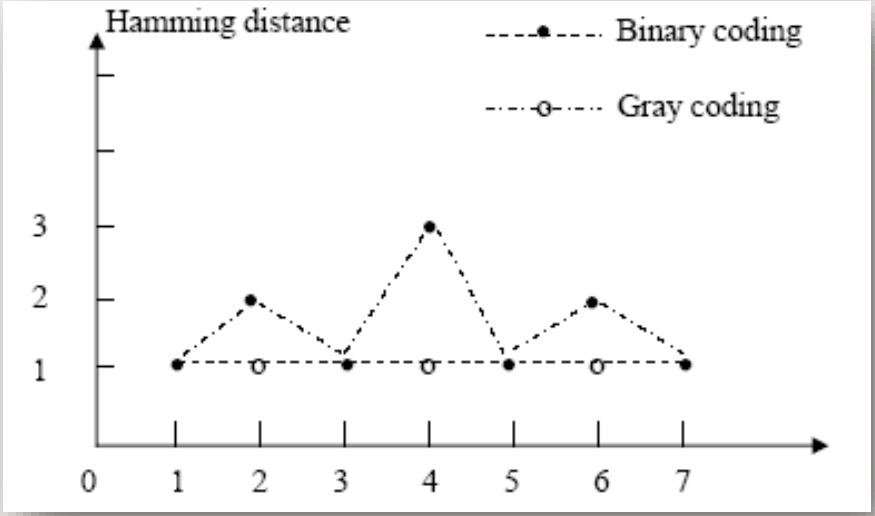
Farklı operatör değerleri ile 5 bağımsız çalıştırma sonunda elde edilen sonuçlar

Parameters→ Results↓	$J = 10$, $T = 200$ $p_c = 0.25$ $p_m = 0.01$	$J = 10$, $T = 200$ $p_c = 0.5$ $p_m = 0.01$	$J = 100$, $T = 20$ $p_c = 0.25$ $p_m = 0.01$	$J = 100$, $T = 20$ $p_c = 0.5$ $p_m = 0.01$	$J = 100$ $T = 20$ $p_c = 0.5$ $p_m = 0.02$
Best	44.512	44.504	44.505	44.513	44.513
Worst	43.639	44.463	44.410	44.494	44.449
Median	44.305	44.469	44.461	44.473	44.493

KROMOZOM KODLAMA BİÇİMLERİ

GA'da Kromozom kodlanma mekanizmalarını genellikle 4 başlık adı altında toplayabiliriz.

- Binary Coded Strings
- Gray Coding
- Real Coding
- Permutation Coding



Gray Coding ≡ Gri Kodlama

İkili kodlama sıklıkla kullanılırken, şekil'de gösterildiği gibi Hammimg 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.

Gray Coding ≡ Gri Kodlama

Değişkenlerdeki küçük bir değişimin uygunluktaki küçük bir değişiklikte sonuç vermesi gerektiği zaman, bir problemi ortaya koyar.

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.

Tabloda ikili ve Gray kodlamaların 3-bit için karşılaştırılması verilmiştir.

	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.

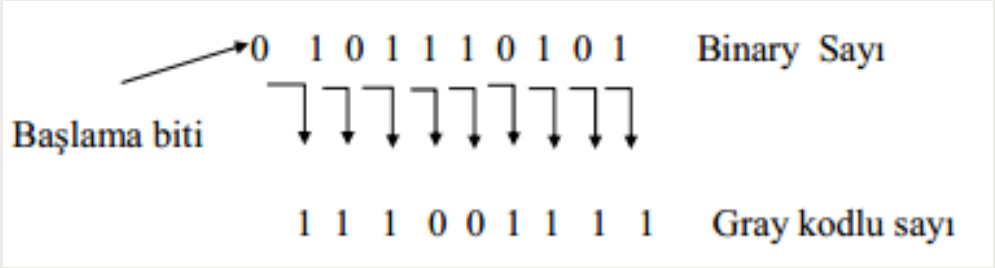
$$g_1 = b_1$$

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

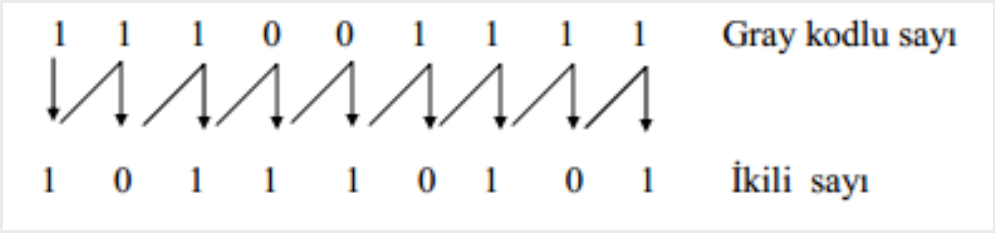
k : ikili sayıdaki bitin sırasıdır.
b1 : en anlamlı biti (yani sol en baştaki biti),
$\overline{b_k}$: b_k'nin değilini
+ : lojik OR lamayı
x : çarpımlar da lojik AND lemeyi gösterir.

Gray Coding ≡ Gri Kodlama

Diğer bir ifadeyle; ikili sistemdeki bir sayıyı Gray kodlu sayı şekline dönüştürmek için, en yüksek basamak değerine sahip bitin solunda ‘0’ olduğu kabul edilip, her bit solundaki bit ile toplanarak yazılır. Bu toplama işlemi XOR işlemidir. Bu işleme en düşük basamak değerlikli bite kadar devam edilir. Elde edilen sayı Gray kodlu sayıdır.



Gray kodlu bir sayıyı ikili sistemdeki sayı şekline dönüştürmek için, en soldaki bit olduğu gibi aşağıya indirilir ve indirilen sayıyla bir sonraki basamakta bulunan sayı toplanarak yazılır. Bulunan sayı ile bir sonraki basamaktaki sayı toplanır ve bu işleme en düşük değerlikli bite kadar devam edilir.



Real Coding ≡ Gerçek Sayı Kodlama

Gerçek Kodlama ≡ Gerçek Sayı Gösterimi ≡ Sürekli Gösterim ≡ Kayan Noktalı Gösterim

Bu kodlama biçiminde kromozomlar bir reel sayılar vektörü olarak tanımlanır.

İkili kodlamaya göre daha hassastır.

Daha kolay kodlanır ve kısıtların yönetilmesi daha kolaydır.

Permutation Encoding ≡ Permutasyon Kodlama

Gezgin satıcı problemi veya görev sıralama gibi sıralama problemlerinde kullanılabilir.
Her kromozom sırada konum belirten numara karakter dizisinden oluşur.

Kromozom A: 1 5 3 2 6 4 7 9 8
Kromozom B: 8 5 6 7 2 3 1 4 9

Bazı problemlerde bazı çaprazlama ve mutasyon türleri için kromozomların tutarlılığı için düzeltmeler yapılması gerekir.

Çaprazlama (Tek Noktalı Çaprazlama): Bir kesme noktası seçilir, kesme noktasına kadar ilk atadan, kesme noktasından sonraki kısımlar da ikinci atadan olmak üzere permütasyonlar kopyalanır. Aynı sayılar olmayan sayılarla değiştirilerek tutarlı yeni yavru elde edilir.

Mutasyon (Sıra Değiştirme): İki sayı seçilir ve yerleri değiştirilir.
(1 **2** 3 4 5 6 **8** 9 7) –> (1 **8** 3 4 5 6 **2** 9 7)

Seçim Mekanizmaları genel olarak 3 grupta toplanabilir:

- Proportional Selection (Rulet Tekerleği)
- Tournament Selection
- Ranking Selection

Schema 1.

Step 1. Set $j = 1$, where j is the number of a chromosome.

Step 2. Generate two random integer numbers a and b such that

$$1 \le a \le J \text{ and } 1 \le b \le J$$

where J is the population size.

Step 3. Compare chromosomes a and b from the t-th generation. If chromosome a is better than b put chromosome a in the place of the j–th chromosome in the $t + 1$ generation. Otherwise put chromosome b in this place.

Step 4. Set $j = j + 1$ and if $j \le J$ go to step 2. Otherwise go to the next generation.

Schema 2.

Step 1. Set $j = 1$, where j is the number of a chromosome.

Step 2. Generate a random integer number a such that

$$1 \le a \le J$$

Step 3. Compare the chromosomes j and a from the t-th generation. If the chromosome j is better than a put chromosome j in the place of the j–th chromosome in the $t + 1$-st generation. Otherwise put chromosome a in this place.

Step 4. Set $j = j + 1$ and if $j \le J$ go to step 2. Otherwise go to the next generation.

Tournament Selection ≡ Turnuva Seçimi

Schema-1 için üretilen 20 random sayı

6 8 4 5 3 7 6 10 1 2 5 8 9 3 10 7 1 4 2 9

$eval(v_1) = f(-1.335, 3{,}711) = 31.669$
$eval(v_2) = f(1.043, 1.803) = 41.429$
$eval(v_3) = f(-0.052, 3.143) = 32.737$
$eval(v_4) = f(0.729, 1.072) = 40.598$
$eval(v_5) = f(-0.885, 1.150) = 44.308$
$eval(v_6) = f(1.225, 3.005) = 41.076$
$eval(v_7) = f(-0.664, 4.812) = 18.894$
$eval(v_8) = f(0.121, 2.253) = 38.912$
$eval(v_9) = f(2.105, 2.919) = 39.255$
$eval(v_{10}) = f(1{,}177, 1.002) = 38.627$

v_1' = [10101110010110011001110 10] /v_6/
v_2' = [00100111010100111010111 00] /v_5/
v_3' = [01011100101101000001110 10] /v_3/
v_4' = [10101110010110011001110 10] /v_6/
v_5' = [10100010101101011100010 01] /v_2/
v_6' = [00100111010100111010111 00] /v_5/
v_7' = [11100110101110010101011 10] /v_9/
v_8' = [10101011010100110011010 10] /v_{10}/
v_9' = [10001110101100110111011 00] /v_4/
v_{10}'= [10100010101101011100010 01] /v_2/

Schema-2 için üretilen 10 random sayı

6 8 4 5 3 7 6 10 1 2

$eval(v_1) = f(-1.335, 3{,}711) = 31.669$
$eval(v_2) = f(1.043, 1.803) = 41.429$
$eval(v_3) = f(-0.052, 3.143) = 32.737$
$eval(v_4) = f(0.729, 1.072) = 40.598$
$eval(v_5) = f(-0.885, 1.150) = 44.308$
$eval(v_6) = f(1.225, 3.005) = 41.076$
$eval(v_7) = f(-0.664, 4.812) = 18.894$
$eval(v_8) = f(0.121, 2.253) = 38.912$
$eval(v_9) = f(2.105, 2.919) = 39.255$
$eval(v_{10}) = f(1{,}177, 1.002) = 38.627$

v_1' = [10101110010110011001110 10] /v_6/
v_2' = [10100010101101011100010 01] /v_2/
v_3' = [10001110101100110111011 00] /v_4/
v_4' = [00100111010100111010111 00] /v_5/
v_5' = [00100111010100111010111 00] /v_5/
v_6' = [10101110010110011001110 10] /v_6/
v_7' = [10101110010110011001110 10] /v_6/
v_8' = [01100111101101110011010 11] /v_8/
v_9' = [11100110101110010101011 10] /v_9/
v_{10}'= [10100010101101011100010 01] /v_2/

Ranking Selection ≡ Dereceli Seçimi

Popülasyon en iyiden en kötüye sıralanır.

Her bir kromozomun seçilme ihtimali uygunluk değerine değil derecesine bağlıdır.

Genellikle 2 metot kullanılır: *Linear Ranking* ve *Exponential Ranking*.

Linear ranking. In this case the ranking function has the following form:

$$p_j = q - (j-1) \times r$$

where p_j is the selection probability for the j-th chromosome in the ranking of the considered population and q is the probability of the best chromosome. Denoting by q_0 the probability for the worst chromosome we can determine the parameter r as follows:

$$r = \frac{q - q_0}{J - 1}$$

where J is the population size.

Intermediate chromosomes' ranks are decreased from q to q_0 proportionally to their rank. Setting $q_0 = 0$, the maximum selective pressure will be obtained.

Exponential ranking. Michalewicz, 1996 proposed the following exponential ranking function:

$$p_j = q(1-q)^{j-1}$$

In this case a larger value of q implies stronger selection pressure.

Hancock, 1994 proposed another exponential function. This function has the form:

$$p_j = q^{j-1}$$

where q is close to one, say 0.99. The best chromosome has the fitness equal to 1 and the worst chromosome has the fitness q^{J-1}.

Bu metotlar kabaca 4 kategoride sınıflandırılabilir:

- Rejecting Strategy
- Repairing Strategy
- Modifying Genetic Operator Strategy
- Penalty Function Strategy

Penaltı Fonksiyonu Stratejisi ile kısıtlı bir optimizasyon problemi, kısıtsız bir probleme dönüştürülmeye çalışılır.

Genel olarak matematiksel formu şu şekilde verilir:

$$\phi(\mathbf{x}, r) = f(\mathbf{x}) + r \sum_{m=1}^{M} [h_m(\mathbf{x})]^2 + r \sum_{k=1}^{K} G_k [g_k(\mathbf{x})]^2$$

Burada G_k Heaviside operatörüdür ve $G_k = \begin{cases} 0 & g_k(\mathbf{x}) \geq 0 \\ 1 & g_k(\mathbf{x}) < 0 \end{cases}$

r, ceza şartlarının büyüklüğünü kontrol eden pozitif bir çarpandır.

$g_k(\mathbf{x}) \geq 0$ şeklinde tanımlanan eşitsizlik kısıtlarını gösterir.

$h_m(\mathbf{x}) = 0$ şeklinde tanımlanan eşitlik kısıtlarını gösterir.

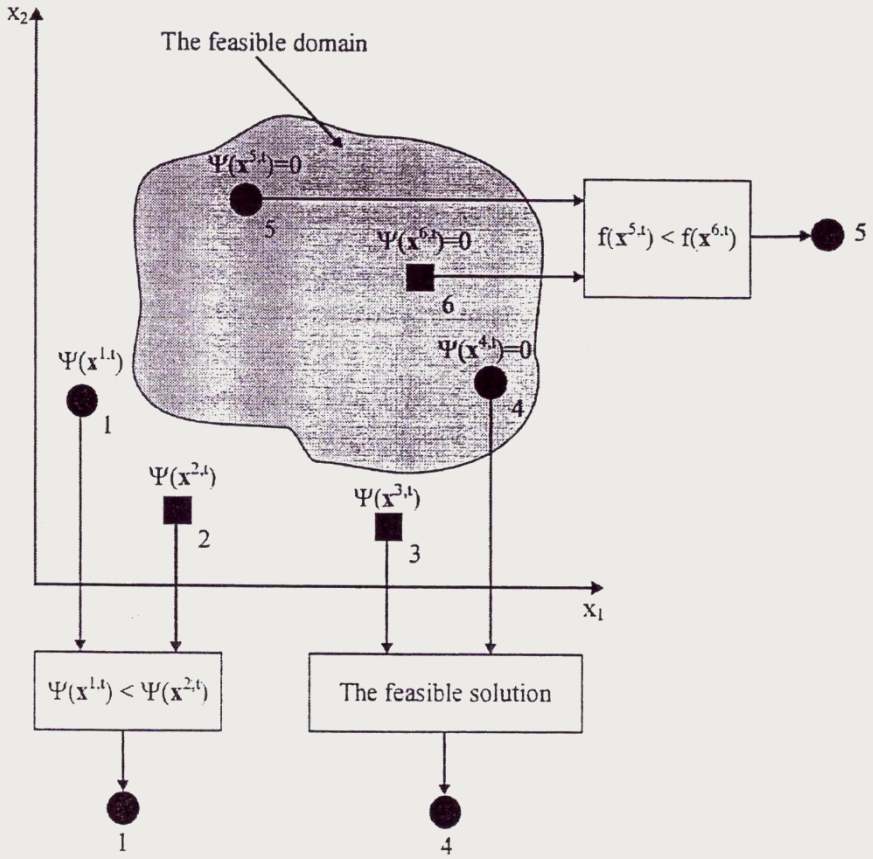
TURNUVA SEÇİMİNİN KISITLI PROBLEMLERDE UYGULANIŞI

Bu yöntemde iki kromozom arasındaki turnuva şu şekilde gerçekleştirilir:

Her iki kromozom da uygulanabilir bölgede değilse, uygun bölgeye daha yakın olan sonraki nesle alınır. Amaç fonksiyonunun değerleri her iki kromozom için de hesaplanmaz.

Bir kromozom uygulanabilir bölgede ve diğeri uygulanabilir bölgenin dışındaysa, uygulanabilir bölgedeki bir sonraki nesle alınır. Amaç işlevinin değerleri her iki kromozom için hesaplanmaz.

Her iki kromozom da uygulanabilir bölgede ise, hem kromozomlar için objektif fonksiyonun değerleri hesaplanır hem de objektif fonksiyonun daha iyi bir değerine sahip olan bir sonraki nesle alınır.



ÇALIŞMA ÖDEVİ

- Genetik Algoritmaların Csharp kodunu kendiniz yazmaya çalışın.
- Yakınsama grafiğini çizdirin.
- Sonlanma koşulu olarak MaxFes kullanın.
- Popülasyon boyutu, Çaprazlama ve Mutasyon Oranı gibi kullanıcı tanımlı parametreler dışardan girilebilir olsun.