

## GENETİK ALGORİTMALARDA DÜZENLİ POPULASYON VE DÜZENLİ OPERATÖR

Bilal ALATAŞ, Ali KARCI

Fırat Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği  
Bölümü, Elazığ.

### ÖZET

Doğal seçim ve genetik kurallara dayandırılmış bir arama işlemi olan genetik algoritmaların (GA) temel mantığında başlangıçta çözülecek problem için gelişigüzel olarak çözümler kümesinin üretilmesi ve daha sonra bu çözümlerin genetik bir süreçle iyileştirilmesi vardır. Bu çalışmada, genetik süreç içerisinde gelişigüzel oluşturulan başlangıç populasyonunun dezavantajlı durumunun giderilmesi ve genetik aramanın performansının artırılması için düzenli populasyon yönteminden esinlenerek geliştirilen yeni bir operatör sunulmuştur. Bu operatörle iki ebeveynden dört ya da probleme özgü olarak daha fazla, kaliteli çocuk birey oluşturulmakta; genetik çeşitlilik sağlanmakta ve erken yakınsama önlenmektedir. Bu operatör ve düzenli populasyonla, genetik aramanın sözü edilen problemlerin üstesinden geldiği ve global çözüme daha kısa sürede ulaştığı görülmüştür.

**Anahtar sözcükler:** Genetik Algoritmalar, Genetik Arama, Düzenli Populasyon, Düzenli Operatör, Genetik Algoritma Performansı

### UNIFORM POPULATION AND UNIFORM OPERATOR IN GENETIC ALGORITHMS

### ABSTRACT

The main idea of the Genetic Algorithms (GA) are based on natural selection and genetic rules which is first to generate randomly solution sets for the problem to solve and then to start the genetic process to improve these solutions. In this study, a new operator, developed by being inspired from the uniform population method, is proposed to eliminate the disadvantages of the randomly generated initial population in the genetic process and to improve the performance of

the genetic search. By this operator, based on the problem, four or more offsprings of high quality from two best individuals are generated; genetic diversity is ensured and early convergence is prevented. By using this operator with uniform population method, problems of genetic search are coped with and global solution is rapidly reached.

**Keywords:** Genetic Algorithms, Genetic Search, Uniform Population, Uniform Operator, Performance of Genetic Algorithm

## 1. GİRİŞ

GA'nın iki tane temel problemi vardır. Bunlar; genetik arama işleminin ıraksaması ve genetik aramanın yerel çözüme takılıp kalmasıdır. Bu problemleri gidermek veya GA'nın performansını arttırmak için çok sayıda araştırmacı çalışma yapmıştır. Bu çalışmalar beş gruba ayrılabilir: Yeni GA çeşidi tanımlayanlar [1], genetik operatörler üzerinde işlem yapanlar [2-3], şema teorisi üzerinde çalışma yapanlar [4], uygunluk fonksiyonu üzerinde çalışma yapanlar [5] ve başlangıç populasyonu üzerinde işlem yapanlar [6-12].

Bu çalışmada GA'nın yerel çözüme takılıp kalmasının ve arama işleminin ıraksamasının önlenmesi için düzenli populasyon ile beraber [8-12], etkili yeni bir operatörün kullanılması ile kaliteli bireylerin oluşturulması, hızlı bir şekilde global çözüme ulaşılması hedeflenmiştir.

Bu makalenin organizasyonu şu şekildedir: Bölüm 2'de GA'lar kısaca anlatılmış, 3. bölümde başlangıç populasyonunun dezavantajlarını ortadan kaldırmak için önerilmiş düzenli populasyon yöntemine ve çalışmasına değinilmiştir. GA'nın performansını arttırmak için önerilen yeni operatör 4. bölümde sunulmuş, deneysel sonuçlar 5. bölümde verilmiştir ve 6. bölümde makale sonuçlandırılmıştır.

## 2. GENETİK ALGORİTMALAR

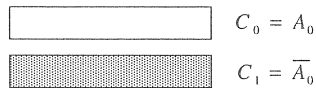
GA, doğadaki evrim mekanizmasını örnek alan bir arama yöntemidir ve bir veri grubundan özel bir veriyi bulmak için kullanılır. Amacı, problemler için doğada geçerli olan en iyinin yaşaması kuralına dayanarak sürekli iyileşen çözümler üretmektir. GA ile oluşturulan seçim, doğal topluluklara benzer bir şekilde bilgisayar hafızasına depo edilmiş kromozomlar üzerinde icra edilmektedir.

GA çalıştırılmadan önce bir problemin çözümü için değişkenlerin belirli bir düzende sıralanmasından oluşturulmuş kromozomların bir grubu başlangıç populasyonu olarak seçilir. Daha sonra bu populasyon, kodu çözülmüş biçimde değerlendirilir, çaprazlama ve mutasyondan oluşan evrim süreci başlar ve en uygun kromozomlar seçilir. Çaprazlama, iki kromozomun bir araya gelerek genetik bilgi değişimi yapmasıdır. Bunun yanında iki ebeveynden yeni kromozomlar üreten bir operatör olan çaprazlama, GA'ların temel bir özelliğidir. Mutasyon veya değişim ise bir kromozomun taşıdığı genetik bilgide bir nedene bağlı olmaksızın değişme olmasıdır. Mutasyon, aramada kısır döngüye girilmemesini sağlamak, toplulukta çözüm olmayan birbirine benzer bireylerden kurtulmak ve yeni alt optimum çözümler bulunmasını sağlamak için kullanılır. Bulunan en uygun kromozomlar yeni oluşturulan eşleşme havuzunda bu genetik işlemlerden geçirilerek bir alt populasyon, yani yeni nesil oluşturulur. Bu yeni nesil de benzer şekilde değerlendirilip, bilinen bir yerine geçme stratejisiyle eski neslin yerine geçer. Çevrim uygun birey veya bireyler bulunana kadar tekrarlanır.

### 3. DÜZENLİ POPULASYON

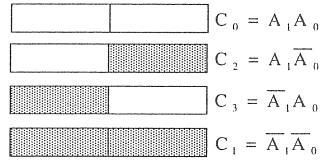
Herhangi bir optimizasyon problemi için kullanılan genetik çözümlerde, başlangıç populasyonu gelişigüzel oluşturulur. Fakat bu yöntemin bazı dezavantajları vardır. Başlangıç populasyonu uygun olmayan bölgede oluşturulmuş olabilir, populasyondaki tüm bireyler birbirlerine çok yakın bulunabilir ve çözümden çok uzakta olabilir veya lokal bir çözüme giderler ve bundan kurtulamazlar. Düzenli populasyon olarak adlandırılan yöntemde başlangıç populasyonu tamamen gelişigüzel değildir, düzgün dağıtılmıştır.

Bu yöntemin bir  $r$  parametresi vardır.  $r = 1$  durumunda bir kromozom gelişigüzel üretilir, bu kromozomun tümleyeni alınır ve bu tümleyen kromozom da populasyon içerisinde bir kromozom olarak yerini alır (Şekil-1). Bu durumda her kromozom bir parça olarak ele alınmıştır ve parça sayısını gösteren parametre  $r = 1$  olur.



**Şekil 1.**  $r = 1$  için gelişigüzel üretilmiş olan kromozomdan ikinci kromozomun elde edilmesi.

$r = 2$  olduğu durumda gelişigüzel üretilen kromozom iki parçaya bölünür. İlk olarak sağ yarının tümleyeni ile sol yarının birbirine eklenmesi ile bir kromozom elde edilir. Ondan sonra sol yarının tümleyeni ile sağ yarının birbirine eklenmesi ile başka bir kromozom elde edilir. Son olarak gelişigüzel üretilen kromozomun tümleyeni alınarak başka bir kromozom daha elde edilir. Kısaca gelişigüzel olarak üretilen bir kromozomdan 3 tane daha kromozom elde edilir ve bu durumun görsel şekli Şekil-2’de görülmektedir.

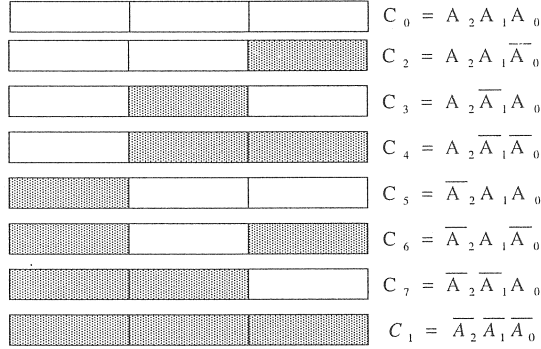


**Şekil 2.**  $r = 2$  için gelişigüzel bir kromozomdan üretilen kromozomlar.

Eğer  $n$  tane gelişigüzel üretilen kromozom varsa,  $r = 2$  için populasyonun boyutu  $n2^2 = 4n$  olur. Bu adımda elde edilen populasyon  $r = 1$  durumundaki populasyondan daha düzenli bir populasyondur. Çünkü,  $r = 2$  durumunda dört tane kromozom uzay içerisinde düzenli bir şekilde dağıtılmaya çalışılırken  $r = 1$  durumunda sadece iki kromozom uzayda düzenli bir şekilde dağıtılmaya çalışılır.

Populasyon daha düzenli hale getirilmek istendiğinde  $r$  değeri artırılır ve  $r = 3$  olmak üzere gelişigüzel bir kromozom üretilir ve bu kromozomun üç parçaya bölündüğü kabul edilir. Benzer şekilde hareket edilerek bir kromozomdan 7 tane kromozom elde edilir. Yani gelişigüzel üretilen kromozom sayısı  $n$  tane ise populasyon boyutu  $8n$  olur. Bundan dolayı populasyon biraz daha düzenli hale gelmiş olur. Şekil-3’te  $r = 3$  durumunun görsel şekli görülmektedir.

$r = m$  için gelişigüzel üretilen kromozom  $m$  tane parçaya bölünür ve aynı mantık ile diğer kromozomlar elde edilir. Bunun sonucunda bir tane gelişigüzel kromozomdan  $2^m - 1$  tane kromozom elde edilir.



**Şekil 3.**  $r = 3$  için gelişigüzel bir kromozomdan 7 tane kromozomun üretilmesi.

Eğer  $\phi$  tane gelişigüzel üretilen kromozom var ve  $r = m$  ise populasyonun boyutu (1) bağıntısında verilen formül ile hesaplanabilir.

$$(2^m - 1)\phi + \phi = \phi 2^m \quad (1)$$

Populasyonun biraz daha matematiksel olarak ifade edilmesi için  $C_{\phi 0}$  gelişigüzel üretilen  $\phi$ . kromozom olsun ve  $1 \leq \phi \leq (N/2^r)$  ve  $N$  populasyon boyutudur.  $I$  endeks kümesi ve  $I = \{i \mid i=0,1,\dots,2^r-1\}$ .  $C_{\phi i}$  kromozomu ise  $\phi$ . kromozomdan türetilen  $i$ . kromozomu temsil etsin. Düzenli populasyonun parametresi  $r$  verildiğinde gelişigüzel üretilen her kromozom  $r$  tane parçaya bölünür ve endeks kümesinin elemanları  $r$ -bit ile ifade edilebilirler.  $I$  endeks kümesinin her elemanın en sağdaki biti, en sağdaki parçayı; onun solundaki biti sağdan ikinci parçayı temsil eder ve bu şekilde devam edilerek en soldaki bit ise en soldaki parçayı temsil eder.  $i \in I$  için,  $i$ 'nin her biti temsil ettiği parçanın uzunluğu kadar genişletilir ve  $i_e$  ile gösterilir. Bu durumda gelişigüzel üretilen  $C_{\phi 0}$  kromozomundan türetilen kromozomlar

$$C_{\phi i} = C_{\phi 0} \oplus i_e, \quad 0 \leq i \leq 2^r - 1 \quad (2)$$

şeklinde olurlar. Başlangıç populasyonu  $P$  olmak üzere

$$P = \{C_{\phi i} \mid C_{\phi i} = C_{\phi 0} \oplus i_e, \quad 0 \leq i \leq 2^r - 1 \text{ ve } 1 \leq \phi \leq N/2^r\} \quad (3)$$

olur.

### 3.1 Olasılıksal Analiz

Düzenli populasyon yönteminin gerçekten düzenli populasyon ürettiğinin anlaşılması ve olasılıksal analizinin yapılabilmesi için bazı terimlerin verilmesi gerekir.

- $N$ : Populasyondaki birey sayısı
- $n$ : Bireyin gen sayısı
- $P(i)$ :  $i$ . bireyin oluşma olasılığı
- $\mu$ : Populasyonun başlangıç beklenen değeri
- $\sigma$ : Populasyonun standart sapma değeri

Olasılıksal hesaplamalara geçmeden önce düzenli populasyon yönteminde elde edilen başlangıç populasyonundaki bireylerden en az birinin çözümün olduğu bölgede olduğu düşünülecektir, çünkü gelişigüzel oluşturulan bireyin tümleyeni alındığında, bu iki bireyden en az biri çözüm bölgesinde olur. Olasılık analizi bir tane gelişigüzel üretilen kromozomdan türetilen kromozomlar üzerinde yapılacağından ve olasılık analizi bütün bireyler için yapılması veya bir gelişigüzel kromozomdan türetilen kromozomlar üzerinde yapılması sonucu değiştirmeyeceğinden bireylerin endekslenmesinde kullanılan çift endeks yerine tek endeks kullanılacaktır. Eğer gelişigüzel üretilen bir birey ( $C_0$  bireyi) dört parçaya bölünerek populasyon üretilecekse ( $r = 4$  için) populasyonda  $C_0$  bireyinden  $C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8, C_9, C_{10}, C_{11}, C_{12}, C_{13}, C_{14}, C_{15}$  bireyleri türetilir ve

$$C_0 + C_{15} = C_1 + C_{14} = C_2 + C_{13} = C_3 + C_{12} = C_4 + C_{11} = C_5 + C_{10} = C_6 + C_9 = C_7 + C_8 = 2^n - 1 \quad (4)$$

ilişkisi elde edilir. Bu bilgilerden sonra üretilcek olan populasyonun olasılık hesapları aşağıdaki gibi olur (düzenli dağılım özelliği):

$$P(C_0) = \frac{1}{2^n} \quad (5)$$

olur ve  $C_0$  bireyi gelişigüzel üretildikten sonra  $C_1, \dots, C_{15}$  bireylerinin ne oldukları kesindir ve

$$1 \leq i \leq 15 \text{ için } P(C_i | C_0) = 1 \quad (6)$$

olur. Buradan  $r$  parçaya bölme işlemi için  $2^r - 1$  tane birey gelişigüzel üretilen bir bireyden üretilir. Bu durumda  $1 \leq i \leq 15$  için  $P(C_i)$  olasılık değeri  $1/2^r$  olur. Çünkü  $C_0$  kromozomundan türetilen kromozom sayısı  $2^r$  olup; bunlardan birinin populasyon içerisinde olma olasılığı  $1/2^r$  olur.  $C = \{C_0, C_1, \dots, C_{15}\}$  olmak üzere  $C$  kümesinin ortalama değeri (beklenen değeri -  $\mu(C)$ ) hesaplanabilir.

$$\begin{aligned}
\mu(C) &= \sum_{i=0}^{15} C_i \cdot P(C_i) \\
&\geq \frac{1}{2^r} ((C_0 + C_{15}) + (C_1 + C_{14}) + (C_2 + C_{13}) + (C_3 + C_{12}) + \\
&\quad (C_4 + C_{11}) + (C_5 + C_{10}) + (C_6 + C_9) + (C_7 + C_8)) \\
&= \frac{1}{2^r} \cdot 2^{r-1} \cdot (2^n - 1) \\
&= \frac{2^n - 1}{2} \tag{7}
\end{aligned}$$

olur. Bu değer sadece bir tane gelişigüzel üretilen bir birey için bulunmuştur. Gelişigüzel oluşturulan birey sayısı  $N$  tane eleman içeren bir populasyon için  $N/2^r$  olur. Toplam populasyon için  $\mu(C)$  değeri  $N/2^r$  ile çarpılıp tekrar bu değere bölüneceği için aynı sonuç elde edilir. Buradan  $\mu = \mu(C)$  olduğu elde edilmiş olur.

Bu populasyonun varyansı ve standart sapmasını hesaplamak için  $C$  kümesinden faydalanılacaktır. İlk olarak  $C$  kümesinin varyansının-  $\sigma^2(C)$ - hesaplanması gerekir

$$\sigma^2(C) = \sum_{i=0}^{15} (C_i - \mu(C))^2 \tag{8}$$

$1 \leq i \leq 7$  ve  $8 \leq j \leq 15$  için her farklı  $(i, j)$  çifti için (her  $i$  ve  $j$  değeri sadece bir kez kullanılarak)  $(C_i - \mu(C)) = -(C_j - \mu(C))$  olduğundan

$$\sigma^2(C) = 2 \sum_{i=0}^7 (C_i - \mu(C))^2 \tag{9}$$

olur. Buradan farkların karesi alındığından dolayı elde edilecek olan sonucun karekökü alınıp standart sapma olan  $\sigma(C)$  değeri elde edildiğinde

$$\mu(C) > \sigma(C) \tag{10}$$

olur. Buradan değişim katsayısının

$$\frac{\sigma(C)}{\mu(C)} < 1 \tag{11}$$

olduğu görülür. Bunun anlamı hızlı yayılan bir topluluk değildir. Bu populasyonun düzgün olup olmadığını anlamak için 3. momente göz atılabilir.

$$\sigma^3(C) = \sum_{i=0}^{15} (C_i - \mu(C))^3 \quad (12)$$

$\sigma^3(C)$  değerini hesaplamak için  $1 \leq i \leq 7$  ve  $8 \leq j \leq 15$  ve her farklı (i,j) çifti için  $(C_i - \mu(C)) = -(C_j - \mu(C))$  olduğundan dolayı (her i ve j değeri sadece bir kez kullanılarak)

$$\sigma^3(C) = 0 \quad (13)$$

olur. Bu, elde edilen populasyonun beklenen değer etrafında düzgün bir şekilde dağılmış olduğu anlamına gelir.

Gelişigüzel başlangıç populasyonu üretme yönteminde hiper uzayda bu kadar düzgün dağılmış bir başlangıç populasyonu elde etme olasılığı sıfır değilse de çok küçük bir değerdir. Bundan dolayı genetik arama hem çok zaman alırken hem de daha önce bahsi geçen iki problemten (ıraksama ve yerel çözümde kalma) kaçınılamaz. Fakat düzenli populasyon yöntemi çok düzgün bir başlangıç populasyonu elde ettiğinden dolayı hem bu iki problemten kaçınılmış olur ve hem de daha kısa zamanda çözüme ulaşılır.

#### 4. DÜZENLİ OPERATÖR

Sunulan yöntemde genetik süreç devam ederken her iterasyonda farklı en iyi iki birey alınır ve bunlardan daha kaliteli dört ya da probleme bağlı olarak daha fazla sayıda çocuk birey elde edilir. Yöntemin sadece ikili kod kullanan GA için çalışması anlatılacaktır. Çocukların elde edilmesi şu şekilde gerçekleşir:

$X_1 = (x_{11}, x_{12}, \dots, x_{1n})$  ve  $X_2 = (x_{21}, x_{22}, \dots, x_{2n})$  populasyondaki en iyi iki birey ve  $x_{1i}, x_{2i} \in \{0,1\}$ ,  $1 \leq i \leq n$  olsun. Bu bireylerden farklılık gösteren bitlerin yeri tutulur ve farklı bit sayısı uzunluğunda bir dizi gelişigüzel oluşturulur. Daha sonra, oluşturulan bu diziden düzenli populasyon yöntemine benzer şekilde  $r=2$  durumu için dört tane farklı dizi oluşturulur ve bu dizilerdeki bit değerleri en iyi bireydeki farklılık gösteren bit pozisyonlarına sırayla dağıtılır. Örnek olarak Tablo-1'de 20 bitle ifade edilen bireylerden en iyi iki birey ve bunların farklı bit pozisyonları görülmektedir.



**Tablo 1.** En iyi iki birey ve farklılıklar

X <sub>1</sub>	1	0	0	1	1	0	0	1	1	1	0	1	0	1	0	0	0	1	1	1
X <sub>2</sub>	1	0	0	0	1	0	1	1	1	1	1	1	0	1	0	0	1	1	0	0
Y	1	0	0	*	1	0	*	1	1	1	*	1	0	1	0	0	*	1	*	*

Görüldüğü gibi 6 bit farklıdır ve gelişigüzel 6 bitlik bir dizi oluşturulur. Oluşturulan dizi 011001 olsun.  $r = 2$  için oluşturulan diğer diziler sırayla: 011110, 100001, 100110 olacaktır:

Gelişigüzel oluşturulan bu dizi ve bu diziden oluşturulan diğer 3 dizi, en iyi bireydeki farklı olan bit pozisyonlarına dağıtıldığında, oluşan yeni bireyler Tablo-2’de gösterilmiştir.

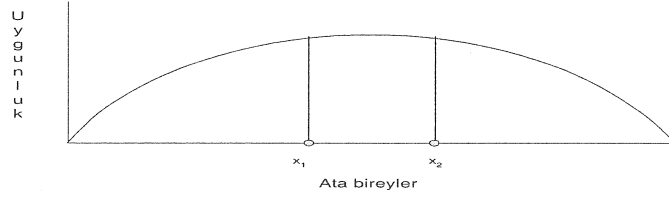
Elde edilen çocuk birey sayısı, probleme özgü olarak daha fazla da olabilir. Bu durumda  $r$ ’nin daha büyük değerleri için çocuk birey sayısı ayarlanabilir. Sonuçta bu genetik çeşitlilikle elde edilecek çocuk bireylerin, ana bireylerden daha kaliteli olması hedeflenmektedir. Bu durum, düzenli populasyonun yukarıdaki olasılık analizi incelenerek çok benzer bir yaklaşımla anlaşılabilir.

**Tablo 2.** Oluşturulan çocuk bireyler

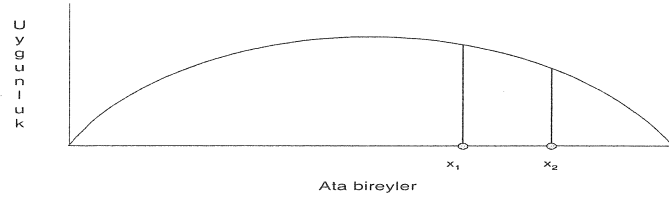
A	1	0	0	0	1	0	1	1	1	1	1	1	0	1	0	0	0	1	0	1
B	1	0	0	0	1	0	1	1	1	1	1	1	0	1	0	0	1	1	1	0
C	1	0	0	1	1	0	0	1	1	1	0	1	0	1	0	0	0	1	0	1
D	1	0	0	1	1	0	0	1	1	1	0	1	0	1	0	0	1	1	1	0

Yani herhangi bir maksimizasyon problemi için en iyi bireyler Şekil-4’teki  $x_1$ ,  $x_2$  ise, elde edilecek çocuk bireylerin  $x_1$ - $x_2$  aralığına düşürülmesi; Şekil-5’teki gibi ise  $x_1$ ’in soluna düşürülmesi amaçlanmıştır. Şekil-5’tekine benzer şekilde en iyi iki birey solda da bulunabilir ancak bu aynı durumdur ve bu kez de çocuk bireyler  $x_2$ ’nin soluna düşürülmeye çalışılır. Bu şekilde çözüme daha hızlı ulaşılır. Düzenli populasyon ile beraber kullanılarak genetik çeşitlilik sağlanmış, yerel çözümde kalma problemi giderilmiş ve çözümden iraksama önlenip global çözüme daha hızlı ulaşılmıştır.

Düzenli populasyon ve düzenli operatörün klasik GA’ya entegrasyonu ile geliştirilmiş GA’nın temel adımları Şekil-6’dadır.



Şekil 4. Global çözüme en yakın iki bireyin birinci durumu



Şekil 5. Global çözüme en yakın iki bireyin ikinci durumu

Buraya kadar sunulan kısımlar ikili kodlama için verilmiştir. Diğer kodlama yöntemlerinde de, düzenli populasyonda yapıldığı gibi, bu operatörün tanımlamaları yapılabilir. Bu operatörde tümlleme işlemi gerçekleştirildiğinden diğer kodlama yöntemleri için tümlleme, kısaca Şekil 6' daki gibidir.

Her genin alabileceği maksimum değer bellidir ve o genin halihazırdaki değeri bu maksimum değerden çıkarılır. Bu şekilde tümlenyeni alınmış olur veya 0 ile 1 arasında bir gerçel sayı gelişiğüzol olarak oluşturulur ve genin halihazırdaki değeri ile bu sayının çarpımı o genin tümlenyenini verir. Daha farklı tümlleme şekilleri tanımlanabilir, fakat bu kısımda en çok kullanılan iki tane tümlleme çeşidi üzerinde duruldu.

**BEGIN**

1. Başlangıç populasyonunu *düzenli populasyon yöntemine* göre üret

**2. DO**

2.1. Populasyonu değerlendir

2.2. Çaprazlama yap

2.3. En iyi iki birey için *düzenli operatör* ile yeni bireyler üret

2.4. Bunları çaprazlama ile oluşan populasyona ilave et

2.5. Mutasyon yap

2.6. Yeni populasyonu oluştur

**WHILE** (Bitim şartı sağlanmadı)

3. En iyi sonucu geri dönder

**END**

Şekil 6. Geliştirilmiş GA

## 5. SİMÜLASYON SONUÇLARI

Operatörün performansını test için MATLAB 6.0 ile bir program yapılmıştır. Programın giriş menüsü Şekil-7’de gösterilmiştir. GA için ikili kod kullanılmıştır.

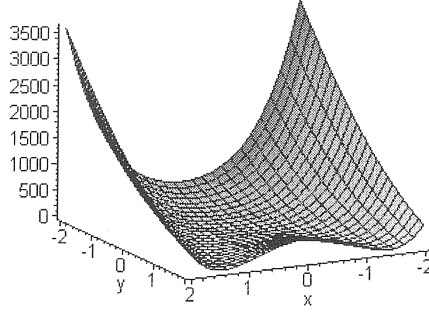
Test fonksiyonu için çok-modelli bir fonksiyon olan Rosenbrock fonksiyonu kullanılmıştır. Tanımı (2)’de verildiği gibidir ve iki değişkenli fonksiyon eğrisi Şekil 8’deki gibidir. Program, beş değişkenliye kadar çalıştırılabilmektedir.

$$f(x_1, x_2, \dots, x_n) = \sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2) + (1 - x_i)^2) \quad (14)$$

Bu çalışmada kullanılan Rosenbrock fonksiyonunun değişken tanım aralıkları istenildiği gibi ayarlanabilmekte; fonksiyon, maksimize ve minimize edilebilmektedir. Uygulamada minimize ve maksimize değerleri ve bu değeri veren değişkenlerin değeri bulunmaktadır.

Bu fonksiyon için düzenli operatörü kullanan ve kullanmayan GA için sonuçlar karşılaştırılmıştır. Düzenli populasyon ile beraber kullanılması ve kullanılmaması durumundaki sonuçların karşılaştırmaları da yapılmıştır.

Şekil 7. Programın giriş menüsü

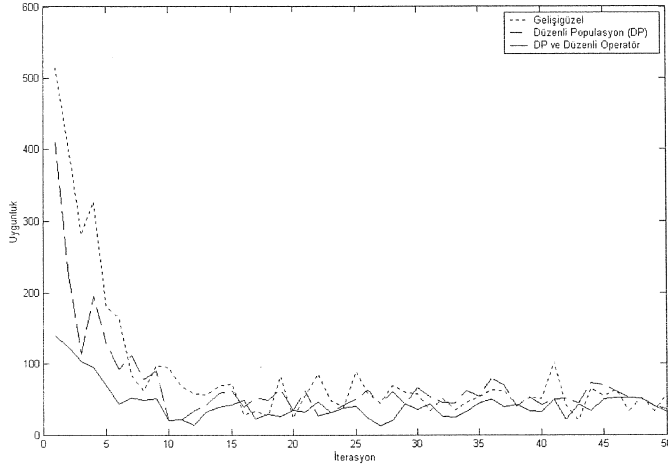


**Şekil 8.** İki değişkenli Rosenbrock fonksiyonu

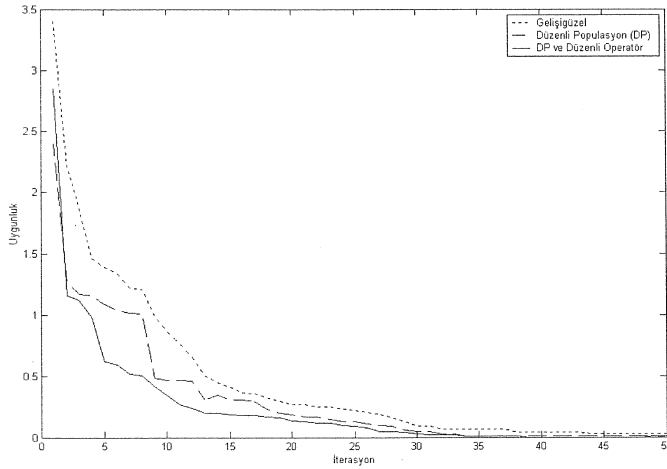
Şekil 9’da, 4 değişkenli Rosenbrock fonksiyonunun, 50 iterasyon boyunca gelişigüzel populasyon, düzenli populasyon, düzenli populasyon ve düzenli operatörü kullanan GA ile minimizasyonunda ortalama uygunluklar görülmektedir. Program 15 kez çalıştırılmış ve ortalamalar alınmıştır. Her değişken 40 bit ile gösterilmiş ( $-2 \leq x_1, x_2, x_3, x_4 \leq 2$ ), populasyon boyutu 32 olarak seçilmiş, rulet tekerleği seçimi, elitizm ve düzenli çaprazlama kullanılmıştır. Ancak, düzenli çaprazlamada ilk iterasyonda bölme noktalarının düzenli populasyon yöntemindeki bölme noktalarıyla çakışmaması sağlanmıştır, çünkü bu durumda çeşitlilik sağlanamaz. Mutasyon olasılığı 0.04, çaprazlama olasılığı ise 0.8’dir. Bu aralıklarda optimize edilmiş fonksiyonun gerçek değeri 0’dır ve bu değere değişkenlerin hepsi 1 değerini aldığı anda ulaşılmaktadır.

Programda girilen r değeri, düzenli populasyonda kullanılan değerdir; farklı r değeri seçilebilmekte ve buna bağlı olarak sonuçlar alınabilmektedir. Düzenli operatör için r değeri daima 2 seçilmiş, düzenli populasyon için ise bu uygulamada 5 seçilmiştir. Burada düzenli populasyon için r değeri çok küçük seçildiğinde çeşitlilik azalmakta çok büyük seçildiğinde ise genetik algoritmaların mantığına ters olarak tüm uzayın örneklenmesi ihtimali oluşmaktadır. Düzenli operatör için de bu durum geçerlidir ve büyük seçilmesi başlangıçta sabit seçilen populasyon boyutunu arttırmaktadır.

Burada hemen hemen tüm iterasyonlar boyunca düzenli populasyonla yeni operatörün beraber kullanılması durumundaki performans açıkça görülmektedir.



Şekil 9. Ortalama uygunluklar



Şekil 10. En iyi uygunluklar

Şekil 10'da ise uygulamanın 15 kez çalıştırılmasıyla elde edilen en iyi uygunluk değerlerinin ortalamasının iterasyon sayısına göre değişimi görülmektedir. Ortalama uygunluklardaki aynı değerler kullanılmıştır. En iyi değeri yakalamak düzenli operatörü kullanılarak az iterasyonla elde edilirken, bu operatör kullanılmadan sürekli kendi kendini tekrar eden bireylerle karşılaşmıştır. Yani değişim sağlanmamış ve böylece global çözüme uzak sonuç elde edilmiştir.

**Tablo 3.** İterasyon sayısı ve uygunluklar

Deney No	İterasyon			Uygunluk		
	GP	DP	DP+DO	GP	DP	DP+DO
1	7	8	19	0.04	0.02	0.01
2	5	18	25	0.07	0.02	0.00
3	13	9	27	0.14	0.01	0.00
4	7	7	39	0.08	0.05	0.01
5	11	9	47	0.03	0.09	0.01
6	3	3	16	0.03	0.05	0.00
7	4	4	21	0.06	0.04	0.01
8	13	12	41	0.03	0.04	0.00
9	15	8	35	0.03	0.06	0.00
10	16	11	16	0.02	0.02	0.01

Başka bir bitim şartı olarak da 7 iterasyon boyunca en iyi uygunluğun değişmediği anda iterasyonu durdurmak olarak seçilmiş ve uygunluk değerleri ile bu şarta ulaşmak için geçen iterasyon sayıları karşılaştırılmıştır. Yuvarlatılmış sonuçlar Tablo-3'te görülmektedir (GP: Gelişigüzel populasyon DP: Düzenli populasyon DP+DO: Düzenli populasyon ve düzenli operatör).

Tablo 3'te düzenli populasyon ve düzenli operatörün beraber kullanılması durumunda en iyi uygunluğun 7 iterasyon boyunca değişmediği andaki kesim şartında daha fazla iterasyonla bitime gittiği görülmektedir ancak sonuçlar çok daha uygundur. Düzenli operatörün kullanılmadığı diğer durumlarda gerçek global değere ulaşmak, çoğu zaman 50 iterasyonu aşmaktadır.

Yani bu bitim şartında gelişigüzel populasyon kullanan GA optimize sonuçlar elde edememektedir ve bazen global çözümden uzak sonuçlar vermektedir.

## 6. SONUÇ

Gelişigüzel başlangıç populasyonu üretme yönteminde hiperuzayda düzgün dağılmış bir başlangıç populasyonu elde etme olasılığı sıfıra yakındır. Bundan dolayı genetik arama çok zaman alırken, iraksama ve yerel çözümden kalma gibi problemler ortaya çıkar. Düzenli populasyon yöntemi ile çok düzgün bir başlangıç populasyonu elde edilir ve düzenli populasyon yönteminden esinlenerek önerilen düzenli operatör ile beraber kullanılması durumunda hem bu iki problemin ortadan kalkması sağlanır ve hem de daha kısa zamanda gerçek çözüme ulaşılır.

Ayrıca bu operatörle zaman içinde populasyonun çeşitliliğini kaybetmesi önlenir. Genetik süreç devam ederken, en iyi bireylerden farklı ve ata bireylerden daha iyi bireyler elde edilir. Bu şekilde yerel çözüme takılıp kalma önlenir ve daha az iterasyonla global çözüme ulaşılır.

DP kromozomları hiperuzayda çarpıklık katsayısı sıfır olan bir başlangıç populasyonu üretirken düzenli operatör ise, genetik süreçte en az bir kromozomun iyiye doğru gitmesini zorlamaktadır.

## **KAYNAKLAR**

1. Srivastava R. P., Goldberg D. E., Verification of the Theory of Genetic Algorithm Continuation, Proc. of the Genetic and Evolutionary Computation Conference, San Fransisco, California, 623-630, (2001).
2. Vasconcelos J. A., Ramirez J. A., Takahashi R. H. C., Saldanha R. R., Improvement in Genetic Algorithms, IEEE Trans. On Magnetics, 37: (3), 3414-3417, (2001).
3. Rowe J. E., PcHee N. F., The Effect of Crossover and Mutation Operators on Variable Length Linear Structures, Proc. of the Genetic and Evolutionary Computation Conference, San Fransisco, California, 535-542, (2001).
4. Schumacher C., Vose M. D. L., Whitley D., The No Free Lunch and Problem Description Length, Proc. of the Genetic and Evolutionary Computation Conference, San Fransisco, California, 565-570, (2001).
5. Branke J., Reducing the Sampling Variance When Searching for Robust Solutions, Proc. of the Genetic and Evolutionary Computation Conference, San Fransisco, California, 235-242, (2001).
6. Cao Y., Wu Q. H., Study of Initial Population in Evolutionary Programming, Proc. of the European Control Conference, (1997).
7. Hill R. R., A Monte Carlo Study of Genetic Algorithm Initial Population Generation Methods, Proc. of the 1999 Winter Simulation Conference, 543- 547, (1999).
8. Karcı A., Arslan A., Bidirectional Evolutionary Heuristic for the Minimum Vertex-Cover Problem, Journal of Computer and Electrical Engineering, Elsevier, 29, 111-120, (2003).

9. Karcı A., Çınar A., Comparison of Uniform Distributed Initial Population Method and Random Initial Population Method in Genetic Search, The 15<sup>th</sup> International Symposium on Computer and Information Sciences, 159-166, İstanbul, (2000).
10. Karcı A., Arslan A., Genetik Algoritmelerde Düzenli Populasyon, Gap IV. Mühendislik Kongresi, 601-607, (2002).
11. Karcı, A. ve Arslan, A., Uniform Population in Genetic Algorithms İ.Ü. Journal of Electrical & Electronics, 2: (2), 495-504, (2002).
12. Karcı A., Arslan A., Düzenli Populasyon (DP) Yönteminin Kaliteli Populasyon Üretme Yeteneğinin Olasılıksal Analizi, Politeknik Dergisi, Gazi Üniv., Teknik Eğt. Fak., Ankara, (Basımda).