

GENETİK SÜRECİN DÜZENLİLİK OPERATÖRÜYLE GLOBAL ÇÖZÜME DOĞRU HARETEKE ZORLANMASI

Bilal ALATAŞ, Ali KARCI

Bilgisayar Mühendisliği Bölümü

Mühendislik Fakültesi

Fırat Üniversitesi, 23119, Elazığ

e-posta: {balatas, akarci}@firat.edu.tr

Anahtar sözcükler: Genetik Algoritmalar, Genetik Arama, Düzenli Populasyon, Düzenlilik Operatörü, Performans

ÖZET

Doğal seçim ve genetik kurallara dayandırılmış bir arama işlem olan genetik algoritmaların (GA) temel mantığında başlangıçta çözülecek problem için gelişigüzel olarak çözümler kümesinin üretilmesi ve daha sonra bu çözümlerin iyileştirilmesi için genetik sürecin başlatılması vardır. Bu çalışmada, genetik süreç içerisinde başlangıç populasyonunun dezavantajlı durumunun giderilmesi ve genetik aramanın performansının artırılması için daha önce önerilen 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 ulaşıldığı görülmüştür.

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 bu konuda çalışma yapmıştır. Bu çalışmalar beş gruba ayrılabilir: Yeni GA çeşidi tanımlayanlar [1-3], genetik operatörler üzerinde işlem yapanlar [2, 4-6], şema teorisi üzerinde çalışma yapanlar [7], uygunluk fonksiyonu üzerinde çalışmalar yapanlar [8, 9] ve başlangıç populasyonu üzerinde işlem yapanlar [10-14].

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

Bu bildirinin 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 bildiri 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 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 çözümü 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

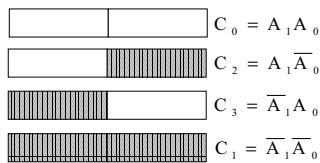
Herhangi bir optimizasyon problemi için kullanılan genetik çözümlerde, başlangıç popülasyonu gelişigüzel oluşturulur. Fakat bu tarz yöntemin bazı dezavantajları vardır. Başlangıç popülasyonu uygun olmayan bölgede oluşturulmuş olabilir, popülasyondaki tüm bireyler birbirlerine çok yakın bulunabilir ve çözümden çok uzakta olabilir veya lokal bir çözüme giderler ve bundan kurtulamazlar. Düzenli popülasyon olarak adlandırılan yöntemde başlangıç popülasyonu 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 popülasyon 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 [12-14].



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 elde edilir. Kısaca gelişigüzel olarak üretilen bir kromozomdan 3 tane daha kromozom elde edilir ve bu durumun görsel şekli Şekil-2'de görülmektedir.



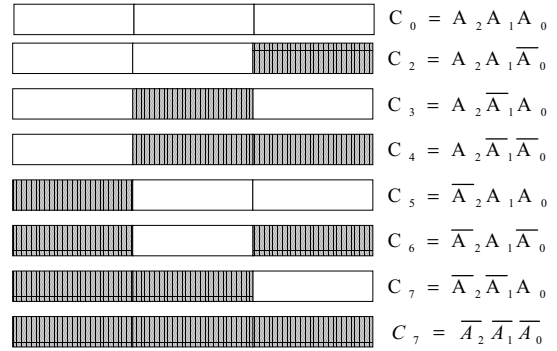
Şekil-2. $r=2$ için gelişigüzel bir kromozomdan üretilen kromozomlar.

$r=2$ için eğer n tane gelişigüzel üretilen kromozom varsa, popülasyonun boyutu $n2^2=4n$ olur. Bu adımda elde edilen popülasyon $r=1$ durumundaki popülasyondan daha düzenli bir popülasyondur. Çünkü dört tane kromozom uzay içerisinde düzenli bir şekilde dağıtılmaya çalışılırken $r=1$ durumunda sadece iki kromozom uzayda düzenli bir şekilde dağıtılmaya çalışılır.

Popülasyon 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 popülasyon boyutu $8n$ olur. Bundan dolayı popülasyon 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 ϕ tane gelişigüzel üretilen kromozom var ve $r=m$ ise popülasyonun boyutu (1) bağıntısında verilen formül ile hesaplanabilir.

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

Popülasyonun biraz daha matematiksel olarak ifade edilmesi için $C_{\phi 0}$ gelişigüzel üretilen ϕ . kromozom olsun ve $1 \leq \phi \leq (N/2^r)$. I endeks kümesi ve $I = \{i \mid i=0,1,\dots,2^r-1\}$. $C_{\phi i}$ kromozomu ise ϕ . kromozomdan türetilen i . kromozomu temsil etsin. Düzenli popülasyonun 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_c ile gösterilir. Bu durumda gelişigüzel üretilen $C_{\phi 0}$ kromozomundan türetilen kromozomlar

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

şeklinde olurlar. Başlangıç popülasyonu P olmak üzere

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

olur.

4. DÜZENLİLİK OPERATÖRÜ

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 $X_2 = (x_{21}, x_{22}, \dots, x_{2n})$ popülasyondaki 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 popülasyon 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 ve bunların farklı bit pozisyonları görülmektedir.

Tablo-1. En iyi iki birey ve farklılıklar

X_1	1	0	0	1	1	0	0	1	1	1	0	1	0	1	0	0	0	1	1	1
X_2	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.

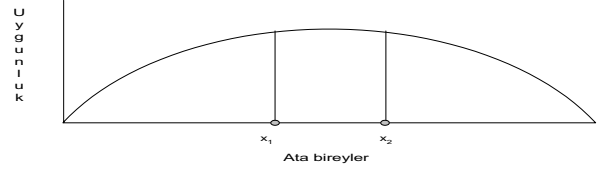
Tablo-2. Oluşturulan çocuk bireyler

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

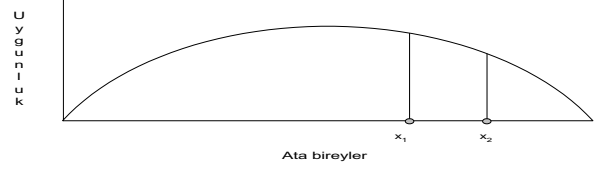
Elde edilen çocuk birey sayısı, probleme özgü olarak daha fazla da olabilir. Bu durumda r' nin daha büyük değerleri için çocuk birey sayısı ayarlanabilir. Sonuçta bu genetik çeşitlilikle elde edilecek çocuk bireylerin, ana bireylerden daha kaliteli olması hedeflenmektedir.

Yani herhangi bir maksimizasyon problemi için en iyi bireyler Şekil-6'daki x_1, x_2 ise, elde edilecek çocuk bireylerin x_1-x_2 aralığına düşürülmesi; Şekil-7'deki gibi ise x_1 'in soluna düşürülmesi amaçlanmıştır. Bu şekilde çözüme daha hızlı ulaşılır. Düzenli popülasyon ile beraber kullanılarak genetik çeşitlilik sağlanmış, yerel çözümde kalma problemi giderilmiş ve çözümden ıraksama önlenip global çözüme daha hızlı ulaşılmıştır.

Düzenli popülasyon ve düzenlilik operatörünün klasik GA'ya entegrasyonu ile geliştirilmiş GA'nın temel adımları Şekil-8'dedir.



Şekil-6



Şekil-7

BEGIN

1. Başlangıç popülasyonunu düzenli popülasyon yöntemine göre üret

2. DO

- 2.1. Popülasyonu değerlendir
- 2.2. Çaprazlama yap
- 2.3. En iyi iki birey için düzenlilik operatörüyle yeni bireyler üret
- 2.4. Bunları çaprazlama ile oluşan popülasyona ilave et
- 2.5. Mutasyon yap
- 2.6. Yeni popülasyonu oluştur

WHILE (Bitim şartı sağlanmadı)

3. En iyi sonucu geri döndür

END

Şekil-8. Geliştirilmiş GA

Buraya kadar sunulan kısımlar ikili kodlama için verilmiştir. Diğer kodlama yöntemlerinde de, düzenli popülasyonda yapıldığı gibi, bu operatörün tanımlamaları yapılabilir. Bu operatörde tümleme işlemi gerçekleştirildiğinden diğer kodlama yöntemleri için tümleme, kısaca şöyledir:

Gerçek sayı tabanlı kodlamada farklılık gösteren pozisyonların birleşimiyle oluşmuş dizi bir gerçek sayı ile ifade edilir. $R = (r_0 r_1 r_2 \dots r_{n-1})$ bir gerçek sayı olmak üzere bu sayı mantis ve üstel olmak üzere iki kısımdan oluşur. Gerçek sayıların normalize edildikleri kabul edilsin.

$$r_k = r_{mn} r_{m(n-1)} \dots r_{m1} \cdot r_{es} r_{e(s-1)} \dots r_{e1}$$

$$\bar{R} = \bar{r}_0 \bar{r}_1 \dots \bar{r}_{n-1}$$

$$\bar{r}_k = (9 - r_{mn}) \dots (9 - r_{m1}) \cdot (9 - r_{es}) \dots (9 - r_{e1})$$

olur. Bu yöntemde dizinin kendisi parçalara bölünerek tümleyen alma işlemi gerçekleştirilir.

Diğer bir kodlama yöntemi de karakter dizisinden oluşan dizilerin tanımlanmasıdır. Bu yöntemde ise bir karakterin en büyük ASCII değeri 255 olduğundan o karakterin değeri 255'ten çıkartılarak tümleyeni alınmış olur.

Farklılık gösteren pozisyonların birleşimiyle oluşan $R = c_n c_{n-1} \dots c_1$ karakter kodlamalı bir dizi olmak üzere

$\bar{R} = \bar{c}_n \bar{c}_{n-1} \dots \bar{c}_1$ ve $1 \leq k \leq \phi$ için.

$\text{ord}(\bar{c}_k) = \text{ord}(Z) - [\text{ord}(c_k) - \text{ord}(A)]$

olur ve $\text{ord}(\cdot)$ fonksiyonu kendisine parametre olarak gelen karakterin ASCII değerini hesaplar.

5. SİMÜLASYON SONUÇLARI

Operatörün performansını test için MATLAB 6.0 ile bir program yapılmıştır. Programın giriş menüsü Şekil-9'da gösterilmiştir. GA için ikili kod kullanılmıştır.

Şekil-9. Programın giriş menüsü

Test fonksiyonu için çok-modelli bir fonksiyon olan Rosenbrock fonksiyonu kullanılmıştır. Tanımı (2)'de verildiği gibidir ve programda beş değişkenliye kadar çalıştırılabilmektedir.

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

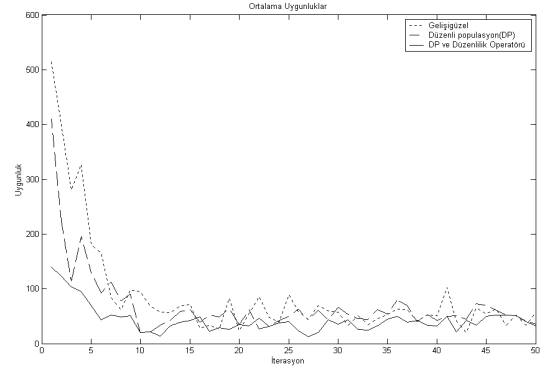
Bu çalışmada kullanılan Rosenbrock fonksiyonunun değişken tanım aralıkları istenildiği gibi ayarlanabilmekte; fonksiyon, maksimize ve minimize edilebilmektedir. Uygulamada minimize ve maksimize değerleri ve bu değeri veren değişkenlerin değeri bulunmaktadır.

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

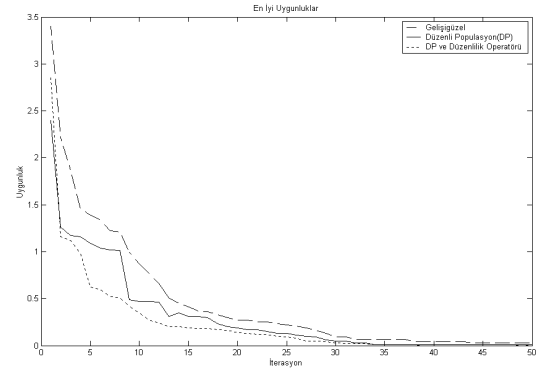
Şekil-10'da, 4 değişkenli Rosenbrock fonksiyonunun, 50 iterasyon boyunca gelişigüzel populasyon, düzenli populasyon, düzenli populasyon ve düzenlilik operatörü kullanan GA ile minimizasyonunda ortalama uygunluklar görülmektedir. Program 15 kez çalıştırılmış ve ortalamalar alınmıştır. Her değişken 40 bit ile gösterilmiş ($-2 \leq x_1, x_2, x_3, x_4 \leq 2$), populasyon boyutu 50 olarak seçilmiş, elitizm ve düzenli çaprazlama kullanılmıştır. 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 ulaşmaktadır.

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

Şekil-11'de ise 50 iterasyon boyunca uygulamanın 15 kez çalıştırılıp elde edilen en iyi uygunlukların ortalamasının alınmasıyla elde edilen grafik görülmektedir. Ortalama uygunluklardaki aynı değerler kullanılmıştır. En iyi değeri yakalamak düzenlilik 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.



Şekil-10. Ortalama uygunluklar



Şekil-11. En iyi uygunluklar

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 (GG: Gelişigüzel populasyon DP: Düzenli populasyon DP+DO: Düzenli populasyon ve düzenlilik operatörü).

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

Tablo-3. İterasyon sayısı ve uygunluklar

	İterasyon			Uygunluk		
	GG	DP	DP+DO	GG	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

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

6. SONUÇ

Gelişigüzel başlangıç popülasyonu üretme yönteminde hiperuzayda düzgün dağılmış bir başlangıç popülasyonu elde etme olasılığı sıfıra yakındır. Bundan dolayı genetik arama çok zaman alırken, ıraksama ve yerel çözümde kalma gibi problemler ortaya çıkar. Düzenli popülasyon yöntemi ile çok düzgün bir başlangıç popülasyonu elde edilir ve düzenli popülasyon yönteminden esinlenerek önerilen düzenlilik operatörü 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 popülasyonun 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.

KAYNAKLAR

- [1] Simoes A., Costa E., On Biologically Inspired Genetic Operators: Transformation in the Standart Genetic Algorithm, PROC. OF THE GENETIC AND EVOLUTIONARY COMPUTATION CONFERENCE, San Fransisco, California, pp. 584-591, 2001.
- [2] Smith J., Modelling GAs with Self Adaptive Mutation Rates, PROC. OF THE GENETIC AND EVOLUTIONARY COMPUTATION CONFERENCE, San Fransisco, California, pp. 599-606, 2001.
- [3] 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, pp. 623-630, 2001.
- [4] Poli R., Rowe J. E., McPhee N. F., Markov Chain Models for GP and Variable-Length GAs with Homologous Crossover, PROC.

OF THE GENETIC AND EVOLUTIONARY COMPUTATION CONFERENCE, San Fransisco, California, pp. 112-119, 2001.

- [5] Vasconcelos J. A., Ramirez J. A., Takahashi R. H. C., Saldanha R. R., Improvement in Genetic Algorithms, IEEE TRANS. ON MAGNETICS, Vol:37, No:3, pp. 3414-3417, 2001.
- [6] Rowe J. E., McPhee 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, pp. 535-542, 2001.
- [7] 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, pp. 565-570, 2001.
- [8] Branke J., Schmidt C., Schmeck H., Efficient Fitness Estimation in Noisy Environments, GENETIC AND EVOLUTIONARY COMPUTATION CONFERENCE, L. Spector et al. (eds.), Morgan Kaufmann, pp. 243-250, 2001.
- [9] Branke J., Reducing the Sampling Variance When Searching for Robust Solutions, PROC. OF THE GENETIC AND EVOLUTIONARY COMPUTATION CONFERENCE, San Fransisco, California, pp. 235-242, 2001.
- [10] Cao Y., Wu Q. H., Study of Initial Population in Evolutionary Programming, PROC. OF THE EUROPEAN CONTROL CONFERENCE, 1997.
- [11] Hill R. R., A Monte Carlo Study of Genetic Algorithm Initial Population Generation Methods, PROC. OF THE 1999 WINTER SIMULATION CONFERENCE, pp. 543-547, 1999.
- [12] Karcı A., Arslan A., Bidirectional Evolutionary Heuristic for the Minimum Vertex-Cover Problem, JOURNAL OF COMPUTER AND ELECTRICAL ENGINEERING, Elsevier (will be appeared).
- [13] Karcı A., Çınar A., Comparison of Uniform Distributed Initial Population Method and Random Initial Population Method in Genetic Search, THE 15th INTERNATIONAL SYMPOSIUM ON COMPUTER AND INFORMATION SCIENCES, pp. 159-166, Istanbul / Turkey, 2000.
- [14] Karcı A., Arslan A., Genetik Algoritmelerde Düzenli Popülasyon, GAP IV. MÜHENDİSLİK KONGRESİ, pp. 601-607, 2002.