DİFERANSİYEL GELİŞİM ALGORİTMASI İLE PROBLEM ÇÖZME ÜZERİNE BİR DEĞERLENDİRME

Serkan AYDIN[1], M. Sadrettin ZEYBEK[2]

ÖZET

GP ile yapılan çözümlerdeki en önemli eksiklik global minimuma ulaşma süresinin uzunluğudur. Bundan dolayı literatürdeki çalışmaların ortak hedefi genellikle yakınsama hızını ve doğruluğunu arttırmak üzerinedir. Bu çalışmada diferansiyel gelişim algoritması (DGA) kullanılarak literatürde sıkça kullanılan bazı problemlerin çözümünde yakınsama hız ve doğruluğunun arttırılması sağlanmıştır.

Anahtar Sözcükler: Gelişimsel programlama, genetik algoritma, diferansiyel gelişim.

ABSTRACT

The major shortcoming of evolutionary programming solutions is the length of time to reach the global minimum. Therefore, most of the studies in the literature focuses on the common goal; to increase the convergence speed and accuracy. In this study, differential evolution algorithm (DGA), is used to increase the convergence speed and accuracy in finding the solution of some problems in the literature.

Keywords: Evolutionary programming, genetic algorithm, differential evolution.

1. GİRİŞ

Bu çalışmada, literatürde sıkça ismi geçen bazı optimizasyon problemlerinin [1,2,3,4] ele alınarak bunların gelişimsel yöntemler ile çözüm teknikleri incelenerek geliştirilmiş ve bu geliştirilen çözüm yöntemiyle daha hızlı bir şekilde ve daha doğru sonuçlar elde edilebileceği hakkında çalışmalar yapılmıştır.

Genellikle bir optimizasyon tekniğinden, başlangıç sistem parametre değerlerine bağlı olmaksızın global minimumun bulunması istenir. Bunun yanında sonuca yakınsamanın hızlı olması ve kullanımının değişik problemlere kolayca uyarlanabilir olması için az sayıda kontrol parametresi içermelidir. Gelişimsel Programlar (GP) veya doğrudan arama metotları olarak da ifade edilen genetik algoritmalar, soğurma benzeşimi (simulated annealing), Nelder&Mead algoritması ve *Diferansiyel Gelişim Algoritması (DGA)* diğer nümerik çözüm metotlarına göre başlangıçtaki sistem parametre değerlerine bağlı olmadıklarından dolayı bir gelişimsel programlama metodu olan DGA kullanılmıştır. Doğrudan arama metotları arasında yer alan DGA, güçlü birer doğrudan arama optimizasyon teknikleri olan uyarlamalı soğurma benzeşimine ve Nelder&Mead yaklaşımlarına göre daha hızlı ve daha doğru sonuçlar verdiği Storn (1995,1999) [5,6] tarafından gösterilmiştir. Storn, bunun yanında, DGA'nın sadece birkaç kontrol parametresi gerektirdiğini kullanımının kolay olduğunu ve paralel hesaplamalar için oldukça elverişli olduğunu da belirtmiştir. Literatürde bu metodun bir çok probleme başarılı bir şekilde uygulandığı görülmektedir [7, 8, 9, 10, 11]. Lampinen [7] kısıtlamalı

[1] Doç. Dr., Celal Bayar Üniversitesi, HFT Teknoloji Fakültesi, Turgutlu – MANİSA
[2] Yrd. Doç. Dr., Celal Bayar Üniversitesi, Turgutlu MYO,Turgutlu – MANİSA

problemlerin çözümünde başarılı bir şekilde DGA’ nı kullanmıştır. Crutchley ve Zwolinski [8] DGA ile dijital filtre dizaynını gerçekleştirirken, Giannakoglou ve diğ. [11] ise paralel manipulator optimizasyonunun DGA ile gerçekleştirmişlerdir.

Bu anlatılan nedenlerden dolayı, bu çalışmada literatürdeki küre modeli ve Schwefel problemlerinin çözümlerinde DGA kullanılmıştır.

2. DİFERANSİYEL GELİŞİM ALGORİTMASI

Bir gelişimsel (evolutionary) optimizasyon algoritması olan *Diferansiyel Gelişim Algoritması* (DGA), kayan-nokta (floating-point) kodlanmış bir tür genetik algoritmadır [5]. Rasgele seçilmiş iki parametre vektörünün farkının rasgele seçilmiş 3. bir parametre vektörü ile toplanması esasına dayanmaktadır. Aşağıda, bu genel çalışma esası, amaç ölçütü, parametre vektörü ve populasyon gibi DGA’ nın tanım eşitlikleri anlatılarak, yeni populasyon üretimi ve seçim aşaması gibi çalışma teknikleri açıklanacaktır.

D boyutlu **V** parametre vektörüne bağlı amaç ölçütünü en genel şekilde

$$f(V):R^D \rightarrow R \qquad (1)$$

ifade edilebilir. Amaç ölçütü f(V)’ nin

$$V=(v_1,......,v_D) \qquad (2)$$

parametre uzayında minimize veya maksimize edilmesi problemin özünü teşkil etmektedir. Her bir parametre

$$v_i^{(L)} \leq v_i \leq v_i^{(U)} \quad i=1,....,D \qquad (3)$$

gibi alt ve üst sınır kısıtlamalarına sahip olabilir

Buna ek olarak amaç ölçütü f(V) üzerinde bazı lineer ve/veya lineer olmayan kısıtlamalar olabilir [7].

NP gerçel-değerli vektörden oluşan G’ ninci nesil populasyonu P_G:

$$P_G = (V_{1,G},......,V_{NP,G}) \quad G=0,...,G_{max} \qquad (4)$$

ve bu üye parametre vektörlerden bir tanesi:

$$V_{i,G} = (v_{1,i,G},.....v_{D,i,G})^T \quad i=1,...,NP \; G=0,....G_{max} \qquad (5)$$

şeklinde gösterilebilir. Burada G_{max} maksimum nesil sayısı ve D ise parameter vektörü boyutu olmaktadır. DE bir amaç ölçütünün optimizasyonunda, P_G’ yi oluşturan G’ ninci nesil populasyonunun üyeleri üzerinde işlem yapar. Bu üyelerin sayısı NP optimizasyon süresince sabittir.

Genellikle ilk nesil populasyon P_0’ ın parametreleri, eşitlik 3’ teki üst ve alt sınır kısıtlamalarına uygun olarak rastgele aşağıdaki gibi üretilir:

$$v_{j,i,0} = rand_j[0,1](v_j^{(U)}-v_j^{(L)}) + v_j^{(L)} \quad i=1,...NP, \; j = 1,.....,D \qquad (6).$$

Burada $rand_j[0,1]$, [0.0, 1.0] aralığında rastgele bir değerdir. Bu genel tanım eşitliklerinden sonra yeni populasyon üretilme tekniği ve bu esnada gerekli olan seçim aşaması Kısım 2.1’ de ayrıntılı olarak incelenecektir.

2.1. Yeni Populasyonun Üretilme Tekniği

P_{G+1} , G ninci populasyon P_G' nin rastgele alınan ve birleştirilen vektörlerinden oluşur. $P^A_{G+1}=(U_{1,G+1},....U_{NP,G+1})$ ile ifade edilen yeni nesile aday üye $\mathbf{U}_{i,G} = (u_{1,i,G},.....u_{D,i,G})$ 10 değişik tipte belirlenebilmektedir. İlk beş tip:

Tip 1- DE/rand/1/bin tipine göre :

$$u_{j,i,G+1} = \begin{cases} z_{j,i,G+1} = \begin{cases} y_{j,i,G+1} = v_{j,r3,G} + F(v_{j,r1,G} - v_{j,r2,G}) \text{ Eğer } v_j^{(L)} < y_{j,i,G+1} < v_j^{(U)} \\ \text{rand}_j[0,1](v_j^{(U)} - v_j^{(L)}) + v_j^{(L)} \text{ diğer durumlarda} \end{cases} \\ \text{Eğer} \quad \text{rand}_j[0,1) \leq CR \\ v_{j,i,G} \quad \text{diğer durumlarda} \end{cases} \tag{7}$$

Tip 2- DE/best /1/bin tipine göre :

$$u_{j,i,G+1} = \begin{cases} z_{j,i,G+1} = \begin{cases} y_{j,i,G+1} = v_{j,b,G} + F(v_{j,r1,G} - v_{j,r2,G}) \text{ Eğer } v_j^{(L)} < y_{j,i,G+1} < v_j^{(U)} \\ \text{rand}_j[0,1](v_j^{(U)} - v_j^{(L)}) + v_j^{(L)} \text{ diğer durumlarda} \end{cases} \\ \text{Eğer} \quad \text{rand}_j[0,1) \leq CR \\ v_{j,i,G} \quad \text{diğer durumlarda} \end{cases} \tag{8}$$

Tip 3- DE/rand to best /1/bin tipine göre :

$$u_{j,i,G+1} = \begin{cases} z_{j,i,G+1} = \begin{cases} y_{j,i,G+1} = v_{j,i,G} + F(v_{j,b,G} - v_{j,i,G}) + F(v_{j,r1,G} - v_{j,r2,G}) \text{ Eğer } v_j^{(L)} < y_{j,i,G+1} < v_j^{(U)} \\ \text{rand}_j[0,1](v_j^{(U)} - v_j^{(L)}) + v_j^{(L)} \text{ diğer durumlarda} \end{cases} \\ \text{Eğer} \quad \text{rand}_j[0,1) \leq CR \\ v_{j,i,G} \quad \text{diğer durumlarda} \end{cases} \tag{9}$$

Tip 4- DE/rand/2/bin tipine göre :

$$u_{j,i,G+1} = \begin{cases} z_{j,i,G+1} = \begin{cases} y_{j,i,G+1} = v_{j,r5,G} + F(v_{j,r1,G} - v_{j,r2,G} + v_{j,r3,G} - v_{j,r4,G}) \text{ Eğer } v_j^{(L)} < y_{j,i,G+1} < v_j^{(U)} \\ \text{rand}_j[0,1](v_j^{(U)} - v_j^{(L)}) + v_j^{(L)} \text{ diğer durumlarda} \end{cases} \\ \text{Eğer} \quad \text{rand}_j[0,1) \leq CR \\ v_{j,i,G} \quad \text{diğer durumlarda} \end{cases} \tag{10}$$

Tip 5- DE/best/2/bin tipine göre :

$$u_{j,i,G+1} = \begin{cases} z_{j,i,G+1} = \begin{cases} y_{j,i,G+1} = v_{j,b,G} + F(v_{j,r1,G} - v_{j,r2,G} + v_{j,r3,G} - v_{j,r4,G}) \text{ Eğer } v_j^{(L)} < y_{j,i,G+1} < v_j^{(U)} \\ \text{rand}_j[0,1](v_j^{(U)} - v_j^{(L)}) + v_j^{(L)} \text{ diğer durumlarda} \end{cases} \\ \text{Eğer} \quad \text{rand}_j[0,1) \leq CR \\ v_{j,i,G} \quad \text{diğer durumlarda} \end{cases} \tag{11}$$

eşitlikleriyle belirlenmektedir. Burada i = 1,....NP, j = 1,....,D değişmektedir. $r_1, r_2, r_3, r_4, r_5 \in \{1,...,NP\}$, i' den farklı rastgele olarak seçilmektedir. $CR \in [0,1]$, $F \in (0,1+]$ aralıklarındadır. Her i elemanı için yeni rastgele r_1, r_2, r_3, r_4 ve r_5 değerleri üretilir. b indisi ise G' ninci nesildeki en optimum sonucu veren parametre vektörünü ifade etmektedir. *Gerçel-değerli değişken, F ((0.0,1.0+]), çaprazlama(crossing) değişkeni, CR ([0,1]) ve populasyondaki üye sayısı, NP; DGA' nın ampirik olarak belirlenen kontrol parametreleridir.* Diğer 5 tipte ise yukarıdaki 5 tipin çaprazlama yapılacak elemanların seçiminde farklılık bulunmaktadır. Yukarıdaki beş eşitlikte de bulunan $\text{rand}_j[0,1) \leq CR$ j=1,….D , şartına ek olarak j=1,…D sıralaması da rasgele olarak değiştirildikten sonra çaprazlama yapılmaktadır. Tablo 1' de 3 elemanlı bir parametre vektörü ve 4 tane kısıtlamaya sahip bir problem için DGA algoritmasının Tip 1' e göre yeni bir üyeyi oluşturması ayrıntılı bir şekilde gösterilmiştir [12].

Tablo 1. DGA algoritması

	Üye 1	Üye 2	Üye 3	Üye 4	Üye 5
$f(V_t)$	14,41	14,16	10,46	14,08	13,64
v_{t1}	39,70	5,25	39,93	30,54	8,47
v_{t2}	21,45	34,56	32,63	36,20	34,04
v_{t3}	5,38	28,33	31,29	4,12	29,72
$c_1(V_t)$	9,47	84,52	8,85	33,28	79,18
$c_2(V_t)$	-13,87	17,95	-18,55	-7,11	15,33
$c_3(V_t)$	57,94	13,01	16,81	26,51	13,84
$c_4(V_t)$	74,59	46,24	47,60	61,81	46,42

Eski nesil P_G

F=0.8

CR=0.8

V_{t_max}=(40, 40, 40) üst sınırlar

V_{t_min}=(0, 0, 0) alt sınırlar

sınırlar dışında olanlar rasgele atanır

Mutasyon

Rasgele seçilmiş iki üyenin farkını F ile çarparak yine rasgele seçilmiş bir başka üye ile toplar.

Fark vektörü		Ağırlıklı Fark vektörü		Üye 1				Yeni Vektör
-	x F =	-	+	-	=	-		-
22,07		17,66		39,70		57,36		31,94
2,15		1,72		21,45		23,17		23,17
-25,6		-20,48		5,38		-15,1		7,84

Çaprazlama : Rasgele seçilen bir vektör ile mutasyon sonucu oluşan yeni vektörden ve CR' ye göre oluşan çaprazlama vektörüne göre Aday vektör U elde edilir.

Üye 1		Çaprazlama vektörü		Çaprazlama vektörü Tümleyeni		Yeni Vektör		Aday Vektör (U)
14,41	x	-	+	-	x	-	=	14.56
39,70		0		1		31,94		31,94
21,45		0		1		23,17		23,17
5,38		0		1		7,84		7,84

Seçim Aşaması

	Üye 1
$f(V_t)$	14,41
v_{t1}	39,70
v_{t2}	21,45
v_{t3}	5,38
$c_1(V_t)$	9,47
$c_2(V_t)$	-13,87
$c_3(V_t)$	57,94
$c_4(V_t)$	74,59

	Aday Vektör (U)
F(U)	14,56
u_1	31,94
u_2	23,17
u_3	7,84
$c_1(U)$	29,83
$c_2(U)$	5,59
$c_3(U)$	51,90
$c_4(U)$	70,74

	Üye 1	Üye 2	Üye 3	Üye 4	Üye 5
$f(V_t)$	14,56				
v_{t1}	31,94				
v_{t2}	23,17				
v_{t3}	7,84				
$c_1(V_t)$	29,83				
$c_2(V_t)$	5,59				
$c_3(V_t)$	51,90				
$c_4(V_t)$	70,74				

Yeni nesil P_{G+1}

2.2. Seçim Aşaması

Kısıtlamaların olmadığı problemlerde, P_{G+1}' in üyeleri $V_{i,G+1}$, şimdiki populasyon P_G' nin üyeleri $V_{i,G}$ ve aday populasyon, P^A_{G+1}' in üyeleri $U_{i,G+1}$' den takip eden *Seçim Kuralına* (SK) göre seçilir :

$$V_{i,G+1} = \begin{cases} U_{i,G+1} \text{ if } f(U_{i,G+1}) \le f(V_{i,G}) \quad veya \quad \left(\#c(V_{i,G}) \ge 0\right) < \left(\#c(U) \ge 0\right) \\ V_{i,G} \quad \text{diğer durumlarda} \end{cases} \tag{12}$$

Amaç ölçütü f(V)' yi minimize eden veya daha fazla sayıdaki kısıtı ($ci(V_t)$) sıfırdan yukarıya taşıyan (Şekil 1' de Üye 1 in $c2(V_t)$' si negatif iken c2(U) pozitif olduğundan dolayı F(U)=14,56 değerine yükselmesine rağmen Aday vektör U Üye 1 ile yer değiştirmiştir.) vektör bir sonraki nesilde yer alır.

3. UYGULAMA

Tablo 2' de verilen 7 problemin çözümünde DGA algoritması kullanılmıştır. Küre modeli, Schwefel problemi, genelleştirilmiş Rosenbrock, gürültülü Quartic fonksiyonları, Hiper-elipsoid ve Neumaier#3 problemlerinin çözümlerinde oldukça başarılı sonuçlar elde edilmiştir.

Tablo 2. Uygulamada kullandığımız fonksiyonlar ve literatürdeki isimleri

	Test Fonksiyonu	Fonksiyonun Literatürdeki İsmi
F1	$f_1(V) = \sum_{i=1}^{D} v_i^2$	Küre Modeli
F2	$f_2(V) = \sum_{i=1}^{D} \lvert v_i \rvert + \prod_{i=1}^{D} \lvert v_i \rvert$	Schwefel Problemi
F3	$f_3(V) = \sum_{i=1}^{D} \left[100(v_{i+1} - v_i^2)^2 + (v_i - 1)^2\right]$	Genelleştirilmiş Rosenbrock Fonksiyonu
F4	$f_4(V) = \sum_{i=1}^{D} i v_i^4 + random[0,1]$	Gürültülü Quartic Fonksiyonu
F5	$f_5(V) = \sum_{i=1}^{D} 2^i v_i^2$	Hiper-elipsoid
F6	$f_6(V) = \sum_{i=1}^{D} (v_i - 1)^2 - \sum_{i=1}^{D} (v_i v_{i-1}$	Neumaier#3
F7	$f_7(V) = \sum_{i=1}^{D} \left[\left(\ln(v_i - 2)\right)^2 + \left(\ln(10 - v_i)\right)^2\right] - \left(\prod_{i=1}^{D} v_i\right)^{0.2}$	Paviani Problemi

Maliyet fonksiyonlarının boyutları ve değişkenlerinin aralıkları diğer referanslardaki gibi seçilmiştir [1, 2, 3, 4]. Tablo 3' de belirtildiği gibi, maksimum nesil sayısı ile parametre vektörünün boyutu D adı geçen literatürdekilerdekiler ile aynı alınmıştır. Program için gerekli CR ve F parametrelerinin her ikisi de 0.5 olarak alınırken populasyon sayısı NP 300 olarak alınmıştır. Çözümde **Tip 1-** DE/rand/1/bin tipi kullanılmıştır.

Bu çalışmada bulunan değerler [1] referansındaki en iyi sonuç veren momentum katsayı gelişim programı (MCEP) algoritması ile kıyaslanarak Tablo 2' de sunulmuştur.

Tablo 3. 50 bağımsız çalışma sonunda elde edilen optimumların literatür ile kıyaslanması.

	DGA	MCEP	Global	D	Nesil (G_{max})	Limitler	F,CR,Strategy
F1	2.0367e-25	6.7e-13	0	30	1500	$[-100,100]^D$	0.5, 0.5,1
F2	3.7890e-12	3.5e-4	0	30	2000	$[-10,10]^D$	0.5, 0.5,1
F3	7.4062e-30	24	0	30	3000	$[-30,30]^D$	0.5, 0.5,1
F4	0.0027	0.001	0	30	3000	$[-1.28,1.28]^D$	0.5, 0.5,1
F5	1.3131e-8	4e-25	0	30	5000	$[-100,100]^D$	0.5, 0.5,1
F6	-4318.2	-157	-4930	30	3000	$[-900,900]^D$	0.5, 0.5,1
F7	-45.778470	-45.7788469	-45.778	10	1000	$[2,10]^D$	0.5, 0.5,1

4. SONUÇLAR VE ÖNERİLER

Bu yayında, literatürde sıkça ismi geçen Küre modeli, Schwefel problemi, genelleştirilmiş Rosenbrock, gürültülü Quartic fonksiyonları, Hiper-elipsoid ve Neumaier#3 problemlerinin çözümleri diferansiyel gelişim algoritması (DGA) ile gerçekleştirilmiştir. Elde edilen sonuçlar literatürdeki momentum katsayı gelişim programı (MCEP) ile karşılaştırılmıştır. Bu çalışmada kullanılan algoritma ile daha doğru sonuçlar elde edilebileceği gösterilmiştir.

Literatürde verilen diğer problemlerin çözümlerinin de DGA ile yapılarak elde edilen sonuçların karşılaştırılmasının yapılabileceği önerilmektedir.

KAYNAKLAR

[1] Yousef A., Javad P., Yagup A., Mohammed R. A., “Momentum coefficient for promoting accuracy and convergence speed of evolutionary programming”, Applied Soft Computing 12 (2012) 1765–1786

[2] Yao X., Liu Y., Lin G., (1999), “Evolutionary Programming Made Faster”, IEEE Trans. on Evolutionary Computation, vol. 3,no. 2.

[3] Wu H., He J., Yao X., (2005), “A online demo of evolutionary programming using mixed mutation strategy for solving function optimization”, UKCI, 264–271.

[4] Montaz M.A., khompatraporn C., Zabinsky Z. B., (2005), "A Numerical Evaluation of Several Stochastic Algorithms on Selected Continuous Global Optimization Test Problems", Journal of Global Optimization, Springer.

[5] Storn, R, 1999. System Design by Constraint Adaptation and Differential Evolution, IEEE Transactions on Evolutionary Computation, 3(1), 22–34.

[6] Storn, R. ve Price, K., 1995. Differential Evolution - a Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces, Technical Report TR-95-012, ICSI.

[7] Lampinen J., 2001. Solving Problems Subject to Multiple Nonlinear Constraints by The Differential Evolution, Proc. of MENDEL2001 7th Int Conf on Soft Computing, Brno Czech Republic, June 1-6.

[8] Crutchley D.A and Zwolinski M., 2002. Using Evolutionary and Hybrid Algorithms for DC Operating Point Analysis of Nonlinear Circuits, *Proceedings of the 2002 Congress on Evolutionary Computation, CEC'02*, Honolulu, Hawaii, May 12-17, Vol.1, 753–758.

[9] David C., Marco D. and Fred G., 1999. *New Ideas in Optimization*, McGraw-Hill, London (UK), 109–125

[10] Jan V. and Pravoslav M, 2001. New Approach to Analog Filters and Group Delay Equaliser Transfer Function Design, *The International Conference onElectronics, Circuits and Systems, ICECS 2001*, Malta, 2-5 September 2001, Vol.1, 157-160.

[11] Giannakoglou K.C., Tsahalis D.T., Periaux J., Papailiou K.D. and Fogarty T. 2002. *Evolutionary Methods for Design, Optimization and Control,* CIMNE, Barcelona (Spain), 223–228.

[12] Aydın S. “Mobil Robotlarda Evrimsel Metotlar İle Optimal Hareket Planlama”, doktora tezi, İTÜ, 2003.