

DİFERANSİYEL GELİŞİM ALGORİTMASI

Timur KESKİNTÜRK

İstanbul Üniversitesi, İşletme Fakültesi, Avcılar Kampusu, Avcılar, İstanbul
tkturk@istanbul.edu.tr

ÖZET

Diferansiyel gelişim algoritması (DGA), populasyon temelli sezgisel bir optimizasyon tekniğidir. Özellikle sürekli verilerin söz konusu olduğu problemlere yönelik olarak geliştirilmiştir. Temel olarak genetik algoritmaya (GA) dayanmaktadır. Aynı operatörlere sahip olmakla birlikte yapıları ve uygulamaları itibarıyla farklılıklar söz konusudur. Bu çalışmada öncelikle DGA algoritması incelenmektedir. Daha sonra klasik DGA üzerinde yapılan birtakım modifikasyonlara yer verilmektedir. Literatürden alınan bir problem üzerinde GA, DGA, daha önce geliştirilmiş olan iki DGA algoritması (HDE ve LUDE) ve çalışmamıza konu olan modifiye edilmiş diferansiyel gelişim algoritması (MEDGA) karşılaştırılmış, sonuçlar raporlanmıştır.

Anahtar Kelimeler: Diferansiyel gelişim algoritması, Modifiye edilmiş diferansiyel gelişim algoritması

1. GİRİŞ

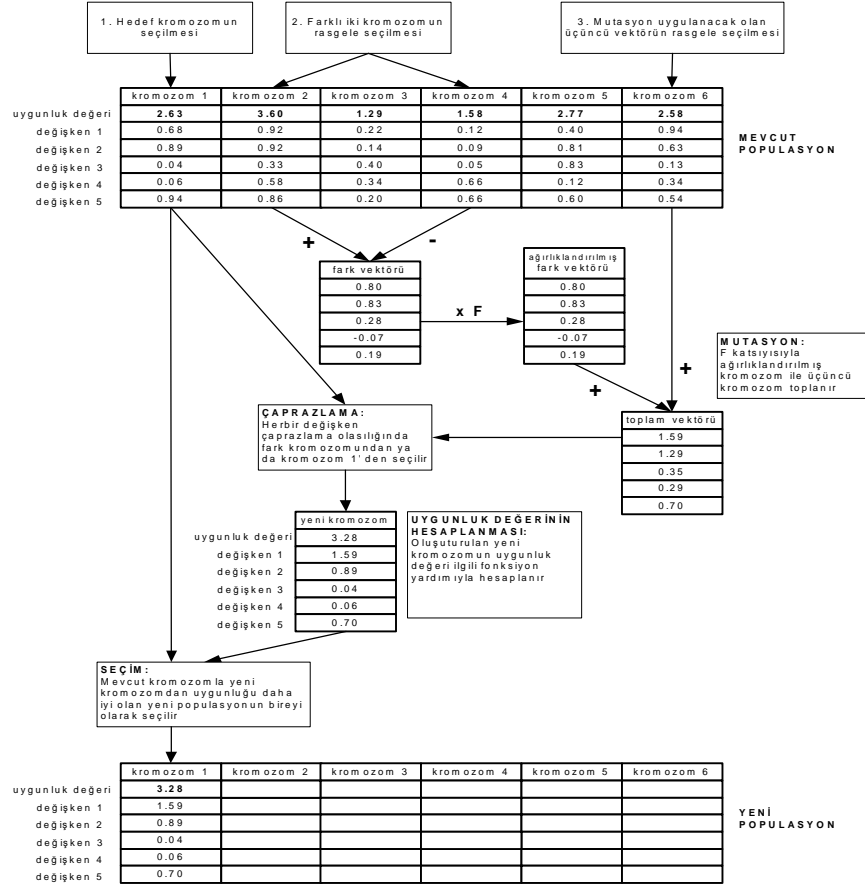
Doğrusal olmayan optimizasyon problemlerinin çözümünde sezgisellerin kullanımı oldukça yaygındır. Bunun başlıca nedenleri arasında modellenmesindeki kolaylık ve hızlı bir şekilde sonuç vermesidir. Deterministik yöntemlerle modelleme yapmak çok fazla matematiksel altyapı gerektirdiğinden ve problem boyutları arttığında yetersiz kaldığından sezgisellerin önemi her geçen gün artmaktadır. Diferansiyel gelişim Algoritması da optimizasyon problemlerinin çözümüne yönelik geliştirilmiş populasyon tabanlı sezgisellerden biridir. Bu çalışmada DGA tanıtılmış ve üzerinde yapılan birtakım modifikasyonlardan bahsedilmiştir. Uygulama bölümünde ise GA, klasik DGA, geliştirilen MEDGA, HDE ve LUDE örnek bir problem üzerinde karşılaştırılmıştır.

2. DİFERANSİYEL GELİŞİM ALGORİTMASI

Diferansiyel gelişim algoritması (DGA), Storn ve Price tarafından geliştirilmiştir populasyon temelli sezgisel optimizasyon tekniğidir (Storn ve Price, 1995; Karaboğa, 2004; Mayer ve diğerleri, 2005). GA'dan esinlenilerek özellikle sürekli parametrelili problemlerin çözümüne yönelik geliştirilmiştir. Populasyon bazlı çalışmamakta, tek tek kromozomlar operatörlere tabi tutulmakta ve böylelikle yeni bir birey elde edilmektedir. Yeni birey eskisinden daha iyi ise yeni birey, aksi takdirde eskisi gelecek jenerasyona aktarılmaktadır. DGA hızlı çalışmakta ve basitçe kodlanabilmektedir. Algoritmaya ait adımlar Şekil 1'de görülmektedir. GA'daki çaprazlama, mutasyon ve seçim burada da söz konusu olup farklı olarak tüm operatörler kromozomlara tek tek uygulanmakta ve yeni bireyin uygunluğuna göre seçim gerçekleştirilmektedir.

2.1 Parametreler

NP	: populasyon büyüklüğü $NP \geq 4$	$x_{j,i,G}$: G jenerasyonunda, i kromozomunun j geni
D	: değişken sayısı (gen sayısı)	$n_{j,i,G+1}$: mutasyon ve çaprazlamaya uğramış ara kromozom
CR	: çaprazlama oranı	$u_{j,i,G+1}$: $x_{j,i,G}$ den üretilen kromozom
G	: jenerasyon (1, 2, 3, ..., G_{max})	$r_{1,2,3}$: yeni kromozomun üretilmesinde kullanılacak kromozomlar $r_{1,2,3} \in \{1, 2, 3, \dots, NP\}$ $r_1 \neq r_2 \neq r_3 \neq i$
F	: ölçekleme faktörü	$x_j^{(l)}, x_j^{(u)}$: değişkenlerin alt ve üst sınır değerleri



Şekil 1 DGA'nın adımları: $f(x)=x_1+x_2+x_3+x_4+x_5$ (Schmidt ve Thierauf, 2005: 13)

2.2 Kodlama ve başlangıç popülasyonu

NP ücten büyük olmalıdır çünkü DGA da yeni kromozomların üretilmesi için mevcut kromozom dışında üç adet kromozom gerekmektedir. NP adet D boyutlu kromozomdan meydana gelen başlangıç popülasyonunun üretimi aşağıdaki gibidir (Karaboğa, 2004: 172).

$$\forall i \leq NP \wedge \forall j \leq D : x_{j,i,G=0} = x_j^{(1)} + \text{rand}_j [0,1] \cdot (x_j^{(u)} - x_j^{(1)}) \quad (1)$$

2.3 Mutasyon

Mutasyon, kromozomun genleri üzerinde rasgele değişiklikler yapmaktır. DGA'da mutasyona tabi tutulacak olan kromozom dışında ve birbirlerinden farklı olan üç kromozom seçilir ($r_{1,2,3}$). İlk ikisinin farkı alınır ve F parametresiyle çarpılır. F genellikle 0-2 arasında değerler almaktadır. Ağırlıklandırılmış fark kromozomu ile üçüncü kromozom toplanır (Denklem 5).

$$\forall j \leq D : n_{j,i,G+1} = x_{j,r_3,G} + F \cdot (x_{j,r_1,G} - x_{j,r_2,G}) \quad (2)$$

2.4 Çaprazlama

Elde edilen fark kromozomu ve $x_{i,G}$ kromozomu kullanılarak yeni deneme kromozomu ($u_{i,G+1}$) üretilir. Deneme kromozomuna genler CR olasılıkla fark kromozomundan 1-CR olasılıkla mevcut

kromozomdan seçilir. $j = j_{\text{rand}}$ koşulu, en az bir tane genin üretilen yeni kromozomdan alınmasını garanti etmek için kullanılmaktadır.

$$\forall j \leq D: x_{j,u,G+1} = \begin{cases} x_{j,n,G+1} & \text{eğer } \text{rand}[0,1] \leq RC \vee j=j_{\text{rand}} \\ x_{j,i,G} & \text{aksi durumda} \end{cases} \quad (3)$$

2.5 Uygunluk Fonksiyonu

Yeni bir kromozom (deneme kromozomu) elde edilmiştir. Yeni jenerasyona geçecek olan kromozomun belirlenmesinde kriter uygunluktur. Hedef kromozomun uygunluk değeri bilinmektedir. Hesaplanacak olan oluşturulan deneme kromozomuna ($u_{i,G+1}$) ait uygunluk değeridir.

2.6 Seçim

Kromozomlardan uygunluğu yüksek olanı yeni jenerasyona atanmaktadır. Seçim operatörüne ait işlem Denklem (7)'de görülmektedir.

$$\forall i \leq NP: x_{i,G+1} = \begin{cases} x_{u,G+1} & \text{eğer } f(x_{u,G+1}) \leq f(x_{i,G}) \\ x_{i,G} & \text{diğer durumlarda} \end{cases} \quad (4)$$

Döngü $G=G_{\text{max}}$ olana kadar devam ettirilmektedir. G_{max} olduğunda mevcut en iyi birey çözümdür.

3. MODİFİYE EDİLMİŞ DGA (MEDGA)

Diferansiyel gelişim algoritması ve benzeri sezgisellerde, arama sürecini hızlandırmak için değişik yollara başvurulur. DGA' da bu amaçla yapılan değişikliklerde ilki yeni jenerasyon oluşturulurken seçilen üç kromozomla ($r_{1,2,3}$) ilgilidir. Klasik DGA' da bu üç kromozom eski jenerasyon içerisinde seçilmektedir. MEDGA' da bundan farklı olarak $r_{1,2,3}$ operatörlere tabi tutulmamış eski kromozomlar ve eski kromozomların yerine geçmiş olan yeni kromozomlar arasından seçilir. Böylelikle yeni bireylerin oluşturulmasında daha elit bir havuz kullanılmış olur. İkinci değişiklik ise algoritma üzerinde önemli bir etkisi olan ölçekleme faktörünün (F) daha geniş bir aralıkta (0-3) rasgele olarak belirlenmesidir. Genelde bu değer daha küçük ve sabit bir değer olarak belirlenmektedir. Son değişiklik ise belli sayıda iterasyondan sonra kromozomların birbirlerine tamamen benzemesi sakıncasını ortadan kaldırmak amacıyla yapılan klasik GA mutasyonudur.

4. UYGULAMA

MEDGA, çalışmamızda Michalewicz (1992: 34)'in eserinden alınmış iki değişkenli kısıtsız doğrusal olmayan optimizasyon problemine uygulanmıştır (Denklemler 5). Michalewicz' in GA ile bulunduğu sonuçlar, Chiou ve Wang'ın (1999) DGA'ya belli eklentiler yaparak geliştirdikleri Hybrid Differential Evolution (HDE) ve Sarimveis ve Nikolakopoulos'un (2005) yine DGA'dan yola çıkarak geliştirdikleri line-up differential evolution (LUDE) algoritmasının sonuçları ile karşılaştırılmıştır.

$$f(x_1, x_2) = 21.5 + x_1 \sin(4 \pi x_1) + x_2 \sin(20 \pi x_2) \quad (5)$$

ve $-3.0 \leq x_1 \leq 12.1$
 $4.1 \leq x_2 \leq 5.8$

Michalewicz, ikili GA ile $NP=20$ ve $G_{\text{max}}=396$ iken elde ettiği en iyi değer $38.827553'$ tür. Chiou ve Wang'ın farklı mutasyonlar için $G_{\text{max}}=300$ iken DE (DGA) ve HDE sonuçları Tablo 1'de ve Sarimveis ve Nikolakopoulos' un farklı iterasyon sayıları için LUDE sonuçları Tablo 2' de yer almaktadır. Algoritma MATLAB programlama dilinde yazılmış ve Pentium 3,2 GHz, 512 MB RAM donanıma sahip bir bilgisayarda çalıştırılmıştır. Sonuçlar, 100 çalıştırma için maksimum, minimum, ortalama, standart sapma, $38.827553'$ ten daha iyi değer sayısı ve optimumu ($38.850294, x_1=11.62554$ ve $x_2=5.72504$) bulma sayısı olarak verilmiştir.

Tablo 1 DE ve HDE Sonuđları

Metot	Mutasyon Tipi	1	2	3	4	5
DE	Maksimum	37.6320	38.4167	38.4170	38.4328	38.6642
	Minimum	26.1486	29.6490	30.8265	27.5275	29.7856
	Ortalama	32.1692	35.5670	34.8150	32.8878	36.4652
	Standart sapma	3.5497	2.2435	2.6262	3.3589	2.3958
	>38.827553 =38.850294	0 Bilinmiyor	0 Bilinmiyor	0 Bilinmiyor	0 Bilinmiyor	0 Bilinmiyor
HDE	Maksimum	38.8503	38.8503	38.7328	38.8503	38.7328
	Minimum	38.5503	38.3328	36.2503	38.7328	38.6328
	Ortalama	38.8045	38.7217	38.2503	38.7763	38.7318
	Standart sapma	0.0847	0.1149	0.6411	0.0570	0.0100
	>38.827553 =38.850294	72 Bilinmiyor	15 Bilinmiyor	0 Bilinmiyor	37 Bilinmiyor	0 Bilinmiyor

Tablo 2 LUDE ve MEDGA Sonuđları

Metot		G=100	G=200	G=300	G=396
LUDE	Maksimum	38,8503	38,8503	38,8503	38,8503
	Minimum	38,1995	38,3199	38,4376	38,5873
	Ortalama	38,7447	38,7847	38,8167	38,8185
	Standart sapma	0,13964	0,10821	0,0692	0,07369
	>38.827553 =38.850294	38 Bilinmiyor	56 Bilinmiyor	70 Bilinmiyor	83 Bilinmiyor
MEDGA	Maksimum	38,85029448	38,85029448	38,85029448	38,85029448
	Minimum	38,71362215	38,73265928	38,73278708	38,73280596
	Ortalama	38,78809235	38,83142771	38,83950521	38,84744602
	Standart sapma	0,058055548	0,041006507	0,033277312	0,017116484
	>38.827553 =38.850294	47 19	81 64	90 80	97 94

Tablo 1 ve 2 incelendiđinde MEDGA' nın diđer algoritmalarla göre daha iyi sonuđlar ürettiđi söylenebilir. Optimumu bulma sayıları diđer algoritmalar için ilgili çalıřmalarda raporlanmamıřtır. MEDGA optimumu bulma sayıları bakımından da iyi bir performans sergilemiřtir. Yapılan deđiřikliklerin algoritmayı daha etkin hale getirdiđi rahatlıkla söylenebilir. Bu modifikasyonlar üzerinde yapılacak daha ayrıntılı çalıřmalarla etkinliđin daha üst seviyelere tařınabileceđi söylenebilir.

KAYNAKLAR

- Ali, M.M., Törn, A.,** (2004), "Population set-based global optimization algorithms: some modifications and numerical studies", Computer & Operations Research, 31, 1703-1725.
- Chiou, J.P., Wang, F.S.,** (1999), "Hybrid method of evolutionary algorithms for static and dynamic optimization problems with application to a fed-batch fermentation process", Computers & Chemical Engineering, 23, 1277-1291.
- Karabođa, D.,** (2004), Yapay Zeka Optimizasyonu Algoritmaları, İstanbul, Atlas Yayın Dađıtım.
- Mayer, D.G., Kinghorn, B.P., Archer, A.A.,** (2005), "Differential evolution – an easy and efficient evolutionary algorithm for model optimisation", Agricultural Systems, 83, 315-328.
- Michalewicz, Z.,** 1992, Genetic Algorithms + Data Structure = Evolution Programs, A.B.D., Springer & Verlag.
- Schmidt, H., Thierauf, G.,** (2005), "A combined heuristic optimization technique", Advances in Engineering Software, 36, 11-19.
- Storn, R., Price, K.,** 1995, "Differential evolution: a simple and efficient adaptive scheme for Global optimization over continuous spaces", Technical Report TR-95-012, International Computer Science Institute, Berkeley.