İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi Yıl: 5 Sayı: 9 Bahar 2006/1 s.85-99

DİFERANSİYEL GELİŞİM ALGORİTMASI

Timur KESKİNTÜRK*

ÖZET
Doğrusal olmayan problemlerin çözümüne yönelik olarak geliştirilmiş bir çok teknik söz konusudur. Özellikle değişken sayısına ve veri tiplerine bağlı olarak problemlerin zorluk dereceleri de artabilmektedir. Bu tip problemlerin deterministik yöntemlerle çözümü, problemin yapısına bağlı olarak hem modellemede hem de çözüm sürecinde zorluklar içerebilmektedir. Bunların üstesinden gelebilmek için sezgisel yöntemler geliştirilmiştir. Diferansiyel gelişim algoritması (DGA), özellikle sürekli verilerin söz konusu olduğu problemlerde etkin sonuçlar verebilen, işleyiş ve operatörleri itibariyle genetik algoritmaya dayanan populasyon temelli sezgisel optimizasyon tekniklerinden biridir. Bu çalışmada, diferansiyel gelişim algoritması tanıtılmış ve aşamaları anlatılmıştır. Çalışmanın sonunda, DGA literatürden alınmış bir probleme uygulanmış, sonuçlar genetik algoritma sonuçları ile karşılaştırılmıştır.

Anahtar Kelimeler: Diferansiyel Gelişim Algoritması, Genetik Algoritma, Mutasyon, Çaprazlama, Seçim.

DIFFERENTIAL EVOLUTION ALGORITHM

ABSTRACT
There are several techniques developed for solving nonlinear optimization problems. These problems become more difficult related to the number of variables and types of parameters. Solution of these problems with deterministic methods may include difficulties in both modeling and solving depending on the type of the problem. Heuristics are developed in order to overcome these difficulties. Differential evolutionary algorithm (DEA) related to genetic algorithm concerning process and operators, is an efficient population based heuristic optimization technique especially for problems of continuous variables. In this paper, DEA is presented and its operators are detailed. DEA is applied to a problem obtained from literature and results are compared with genetic algorithm.

Keywords: Differential Evolution, Genetic Algorithm, Mutation, Crossover, Selection.

* İstanbul Üniversitesi, İşletme Fakültesi, Sayısal Yöntemler ABD, İstanbul, tkturk@istanbul.edu.tr

1. GİRİŞ

Gerek fen bilimlerinde gerek sosyal bilimlerde ve bu bilim dallarının uygulama alanlarında karşılaşılan birçok problem, doğrusal veya doğrusal olmayan optimizasyon problemi olarak modellenebilmektedir. Uygulamadaki problemlerin büyük bir bölümü doğrusal olmayan bir yapıya sahiptir. Doğrusal olmayan problemlerin çözümüne yönelik olarak geliştirilmiş bir çok teknik söz konusudur. Özellikle değişken sayısına ve veri tiplerine bağlı olarak problemlerin zorluk dereceleri de artabilmektedir. Bu tip problemlerin deterministik yöntemlerle çözümü, hem problemin yapısına bağlı olarak modellemede hem de çözüm sürecinde zorluklar içermektedir. Ya istenilen sonuca ulaşılamamakta ya da kabul edilebilir sınırların dışında sürelerde ulaşılabilmektedir. Bunların üstesinden gelebilmek için sezgisel yöntemler geliştirilmiştir. Özellikle populasyon temelli sezgiseller çok noktalı arama prosedürleri sayesinde, hızlı bir şekilde sonuç verebilmektedirler. Bunlardan bazıları genetik algoritma (GA), bulanık mantık, karınca kolonisi algoritması, benzetilmiş tavlamadır.

Genetik algoritma (GA), şu ana kadar geliştirilmiş en popüler optimizasyon tekniklerindendir. GA konusunda en temel kaynaklar, prensiplerini ilk defa ortaya koyan Holland (1975), Goldberg (1989) ve Michalewicz (1992)' in eserleridir. GA farklı alanlarda başarıyla kullanılmaktadır. Kromozom olarak adlandırılan alternatif çözümler setine (populasyon) dayalı olarak çalışan bu algoritmada genellikle ikili kodlama kullanılmakta, değişkenler ikili sistemde temsil edilmektedir. İkili kodlamayla çalışan GA'lar gezgin satıcı benzeri kombinatoryal problemlerde oldukça etkili olarak çalışmaktadır. Bununla birlikte, uygulamada gerçek değer parametreli birçok bilimsel ve mühendislik problemiyle karşılaşılmaktadır. Gerçek parametreli problemlerin ikili kodlamalı standart genetik algoritmalarla çözümünde birtakım güçlükler söz konusudur. Sayısal parametrelerin söz konusu olduğu problemlerde, ikili tamsayılarla çalışan klasik genetik algoritmalar geniş dinamik sahayı tam olarak temsil edememekte, performansı yetersiz kalmaktadır. Bu güçlüklerin üstesinden gelebilmek için ikili klasik GA' da birtakım değişiklikler yapılmıştır (Hrstka ve Kucerova, 2004). Bunun yanında, gerçek parametreli problemlere yönelik olarak yeni GA' lar geliştirilmiştir (Michalewicz, 1992). Sürekli parametrelerin söz konusu olduğu problemlerin çözümüne yönelik geliştirilmiş algoritmalardan biri de Price ve Storn tarafından 1995 yılında geliştirilmiş olan DGA' dır (Storn ve Price, 1995; Price ve Storn, 1995). Populasyon tabanlı sezgisel bir algoritma olan DGA özellikle tamamen düzenlenmiş uzayda tanımlı ve gerçek değerli tasarım parametrelerini içeren fonksiyonları optimize etmek amacıyla kullanılan bir algoritmadır (Karaboğa, 2004). Geliştirildiğinden bu yana konuyla ilgili birçok çalışma yapılmıştır. Bu çalışmalardan bir kısmı algoritmanın iyileştirilmesi üzerinedir (Bergey ve Ragsdale, 2005; Hrstka ve Kucerova, 2004; Becerra ve Coello, 2005; Sun vd, 2005). Farklı problemlere yönelik, farklı yapılarda ve farklı parametrelerle çalışan birçok DGA geliştirilmiştir. GA' ya göre çok daha yeni sayılabilecek bir optimizasyon tekniği olan DGA geliştirilmeye devam etmektedir. Zaharie (2002)' nin kontrol parametrelerinin seçimi konusunda, Lin ve arkadaşlarının (2004) farklı değişken tipleriyle çalışabilen DGA'lara yönelik çalışmaları bulunmaktadır.

Çalışmanın ikinci bölümünde, DGA' nın temel prensipleri ve adımları ayrıntıları ile açıklanmıştır. Üçüncü bölümde ise literatürden alınan farklı problemlerin çözümleri DGA çözümleri ile karşılaştırılmıştır.

2. DİFERANSİYEL GELİŞİM ALGORİTMASI

DGA, Price ve Storn tarafından 1995 yılında geliştirilmiş, özellikle sürekli verilerin söz konusu olduğu problemlerde etkin sonuçlar verebilen, işleyiş ve operatörleri itibariyle genetik algoritmaya dayanan populasyon temelli sezgisel optimizasyon tekniğidir (Mayer vd, 2005; Storn ve Price, 1995; Storn, 2001). Temel olarak GA' ya dayanmaktadır. Populasyon tabanlıdır. Aynı anda birçok noktada araştırma yapmaktadır. İterasyonlar boyunca, operatörler yardımıyla problemin çözümü için daha iyi sonuçlar araştırılmaktadır. Klasik ikili GA' dan farklı olarak değişkenler gerçek değerleriyle temsil edilmektedir. GA' da da gerçek değerlerle kodlama kullanılmaktadır (Hrstka ve Kucerova, 2004, Michalewicz, 1992). Ancak Price ve Storn genetik operatörlerdeki birtakım değişikliklerle, gerçek değerlerle kodlamanın kullanıldığı problemlerin çözüm performansını arttırmaya çalışmışlardır. GA' daki çaprazlama, mutasyon ve seçim operatörleri DGA' da da kullanılmaktadır. Farklı olarak her bir operatör tüm populasyona sırayla uygulanmamaktadır. Kromozomlar tek tek ele alınmakta, rasgele seçilen diğer üç kromozomda kullanılarak yeni bir birey elde edilmektedir. Bu işlemler sırasında mutasyon ve çaprazlama operatörleri kullanılmış olmaktadır. Mevcut kromozomla elde edilen yeni kromozomun uygunlukları karşılaştırılarak uygunluğu daha iyi olan, yeni birey olarak bir sonraki populasyona aktarılmaktadır. Böylelikle seçim operatörü de kullanılmış olmaktadır. DGA' ya ait işlemler Şekil 1' de görülmektedir. Üretilen çözümlerin kalitesi, amaç fonksiyonuna ürettikleri değerle (uygunluk değeri) ölçülmektedir.

DGA' nın diğer sezgisellere önemli bir üstünlüğü de kolayca kodlanabilmesidir. Diğer algoritmalar için binlerle ifade edilen satırdan oluşan kodlar söz konusu iken DGA için yaklaşık 20 satırlık kod yeterli olmaktadır (Mayer vd, 2005).

2.1. Problem Ve Parametreler

NP	:	populasyon büyüklüğü (kromozom sayısı) NP ≥ 4 (1, 2, 3, …, i)
D	:	değişken sayısı (gen sayısı) (1, 2, 3, …, j)
CR	:	çaprazlama oranı [0.1,1.0]
G	:	jenerasyon (1, 2, 3, …, G_{max})
F	:	ölçekleme faktörü
$x_{j,i,G}$	:	G jenerasyonunda, i kromozomunun j parametresi (gen)
$n_{j,i,G+1}$	:	mutasyon ve çaprazlamaya tabi tutulmuş ara kromozom
$u_{j,i,G+1}$	:	$x_{j,i,G}$ den bir sonraki jenerasyon için üretilen kromozom (child-trial)
$r_{1,2,3}$	:	yeni kromozomun üretilmesinde kullanılacak rasgele seçilmiş kromozomlar $r_{1,2,3}$ ε {1, 2, 3, …, NP} $r_1 \neq r_2 \neq r_3 \neq$ i
$x_j^{(l)}$, $x_j^{(u)}$	:	değişkenlere ait alt ve üst sınır değerleri

Optimizasyon problemleri genel olarak aşağıdaki gibi ifade edilebilir (Shiakolas vd, 2005).

minimize $f(X)$ (1)

kısıtlar $g_k(X) \leq 0$ (2)

$$x_j^{(l)} \leq x_j \leq x_j^{(u)} \quad j = 1, ..., n \qquad (3)$$

$f(X)$ amaç fonksiyonu, $g_k(X)$ kısıtlar seti ve $X=\{x_1, x_2, x_3, ..., x_n\}$ gerçek değerli değişkenler setidir. $x_j^{(l)}$ ve $x_j^{(u)}$ sırasıyla değişkenlere ait alt ve üst sınır değerleridir

Amaç, problemin tüm kısıtlarını sağlayan en iyi çözümü araştırmaktır. DGA terminolojisinde amaç fonksiyonu, maliyet fonksiyonu (cost-function) olarak da adlandırılabilmektedir. DGA’ da genellikle sürekli değişkenlerle çalışılmakla birlikte, kesikli değişkenler ya da ikisinin kombinasyonuyla çalışanları da geliştirilmiştir (Shiakolas vd, 2005; Lin vd, 2004). Problemin amaç fonksiyonu DGA’ da uygunluk fonksiyonu olarak belirlenmekte ve her biri bir alternatif çözüm olan kromozomların değerini temsil etmektedir. DGA da genetik algoritma gibi kısıtlarla çalışamamaktadır. Kısıtların bir şekilde amaç fonksiyonu içerisinde yer almaları gerekmektedir. Bunu gerçekleştirmek için kısıtlardan uzaklaşmalar, ceza katsayılarıyla uygunluk fonksiyonunun değerinin düşürmekte ve böylece uygun çözüm alanından uzaklaşmalar cezalandırılmaktadır. Genlerle temsil edilen problem değişkenlerinin DGA operatörleri ile sınır dışında belirlenmeleri durumu sürekli kontrol edilmelidir. Aksi takdirde çözüm uygun olmayan alanlara kayacak ve bekli sonsuza gidecektir. Bunun engellenmesi için ise yapılabilecek iki düzeltme söz konusudur: Sınır dışındaki değerlerin sınıra çekilmesi, değişkenin alt ya da üst sınır değeri alması düzeltme alternatiflerinden birincisidir. İkincisi ise sınırların dışında değer almış değişkenlerin yeni değerlerinin alt ve üst sınır arasında rasgele belirlenmesidir. Problemin amaç fonksiyonu, değişkenler ve kısıtlar belirlendikten sonra aşağıdaki adımlar izlenerek DGA uygulanır.

2.2. Kodlama Ve Başlangıç Populasyonu

Probleme ait değişken sayısı her bir kromozoma ait gen (boyut) sayısını belirlemektedir (D). NP ise kullanıcı tarafından belirlenen kromozom sayısıdır. Her zaman üçten büyük olmalıdır. Çünkü DGA da yeni kromozomların üretilmesi için mevcut kromozom dışında üç adet kromozom gerekmektedir ($r_{1,2,3}$). Başlangıçta NP adet D boyutlu kromozomdan meydana gelen başlangıç populasyonu (P_0) aşağıdaki gibi üretilir (Karaboğa, 2004).

$$\forall i \le \text{NP} \wedge \forall j \le \text{D}: \; x_{j,i,\text{G}=0} = x_j^{(l)} + \text{rand}_j\left[0,1\right] \cdot \left(x_j^{(u)} - x_j^{(l)}\right) \tag{4}$$

Başlangıç populasyonu üretildikten sonra, aşağıda açıklanan operatörler G_{max} sayısınca uygulanarak algoritma tamamlanır. Son jenerasyondaki en iyi birey çözüm vektörüdür.

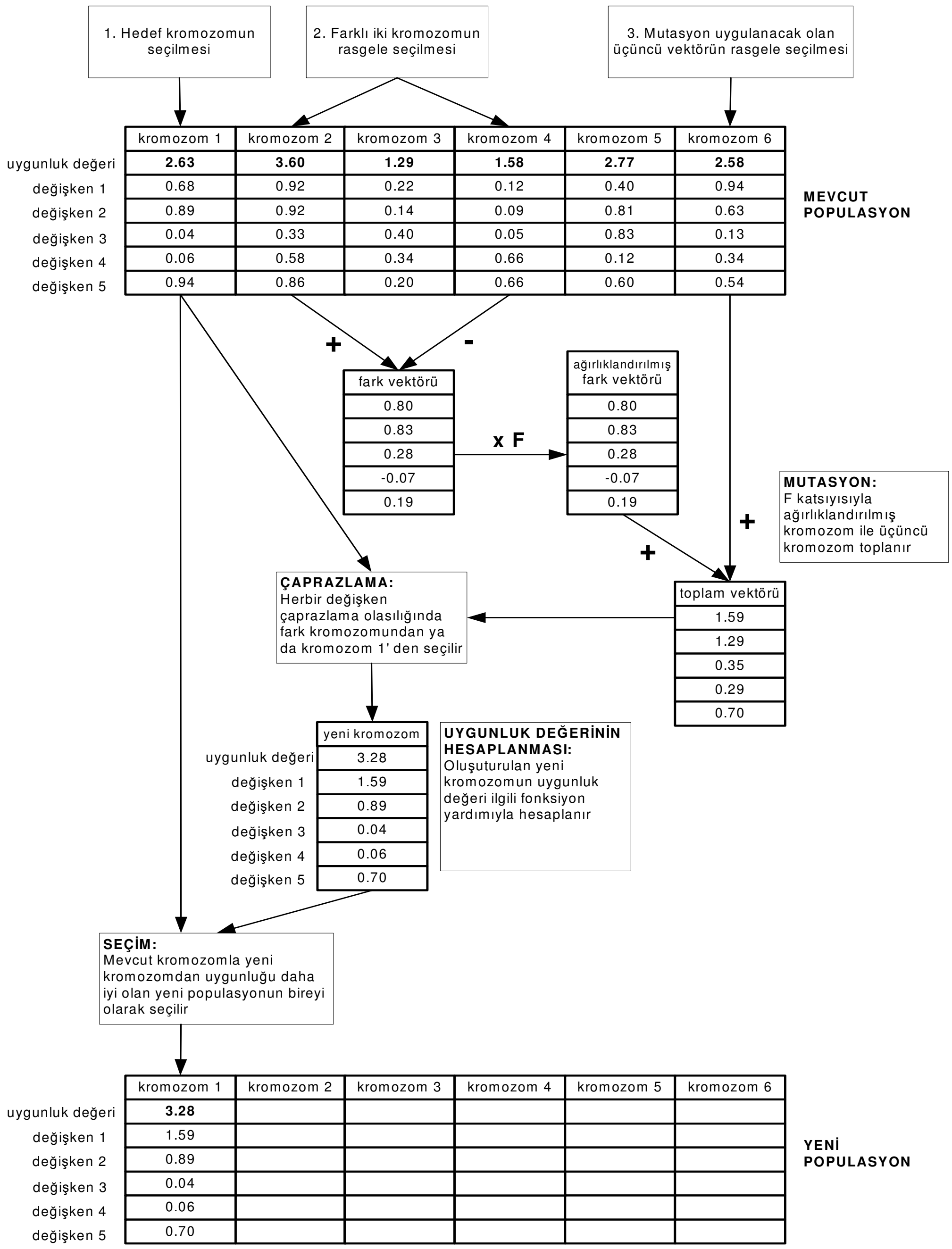
2.3. Mutasyon

Mutasyon, mevcut kromozomun bir kısım genleri üzerinde, rasgele belirlenmiş miktarlarda değişiklikler yapmaktır. Bu değişiklikler sayesinde kromozomunun temsil ettiği çözüm noktası, çözüm uzayında hareket etmektedir. Mutasyonun hedefine ulaşabilmesi için, doğru yönde doğru miktarda hareketi sağlayacak değişikliklerin belirlenmesi gerekmektedir.

Diferansiyel gelişim algoritmasında, mutasyon işlemine tabi tutulacak olan kromozom dışında ve birbirlerinden de farklı olan üç kromozom seçilir ($r_{1,2,3}$). Seçilen kromozomlardan ilk ikisinin farkı alınır. Daha sonra bu fark kromozomu F parametresiyle çarpılır. F parametresi genellikle 0-2 arasında değerler almaktadır. Elde edilen ağırlıklandırılmış fark kromozomu ile seçilen üçüncü kromozomu (r_3) ile toplanır. Böylece mutasyon sonucu çaprazlamada kullanılacak olan kromozom elde edilmiş olur ($n_{j,i,G+1}$).

$$\forall j \le \text{D}: n_{j,i,\text{G}+1} = x_{j,r_3,\text{G}} + \text{F} \cdot \left(x_{j,r_1,\text{G}} - x_{j,r_2,\text{G}}\right) \tag{5}$$

Farklı birçok mutasyon operatörü geliştirilmiştir (Storn, 2001).



Şekil 1. DGA' Nın Adımları. Kullanılan Fonksiyon: F(X)=X1+X2+X3+X4+X5 (Schmidt Ve Thierauf, 2005)

2.4. Çaprazlama

Çaprazlama yapılırken, mutasyon sonucu elde edilen fark kromozomu ve $x_{i,G}$ kromozomu kullanılarak yeni jenerasyona aday, deneme kromozomu ($u_{i,G+1}$) üretilir. Deneme kromozomuna ait her bir gen CR olasılıkla fark kromozomundan 1-CR olasılıkla mevcut kromozomdan seçilir. DGA' da kullanılan bu çaprazlama yöntemi, ikili GA' da kullanılan düzenli çaprazlamanın (uniform crossover), CR eklenmiş hali olarak da tanımlanabilir. Düzenli çaprazlamada her bir gen ayrı olarak değerlendirilip eşit olasılıkla iki ebeveyn kromozomundan birinden seçilmektedir. DGA' da ise eşit olasılık yerine CR olasılığı sözkonusudur. 0 ile 1 arasında üretilen rasgele sayı CR' den küçükse gen, $n_{j,i,\mathrm{G+1}}$' den aksi takdirde mevcut kromozomdan seçilir. Amaç belirlenen oranda genin yeni fark kromozomundan alınmasıdır. Buradaki $j = j_{\mathrm{rand}}$ koşulu, en az bir tane genin üretilen yeni kromozomdan alınmasını garanti etmek amacıyla konulmuştur. Rasgele seçilen j_{rand} noktasındaki gen CR' ye bakılmaksızın $n_{j,i,\mathrm{G+1}}$' den seçilir.

$$\forall j \le \mathrm{D}: \; x_{j,u,\mathrm{G+1}} = \begin{cases} x_{j,n,\mathrm{G+1}} & \text{eğer rand[0,1]} \le \mathrm{RC} \vee j{=}j_{\mathrm{rand}} \\ x_{j,i,\mathrm{G}} & \text{aksi durumda} \end{cases} \tag{6}$$

2.5. Uygunluk Fonksiyonu

Mutasyon ve çaprazlama operatörleri kullanılarak hedef kromozomla birlikte üç farklı kromozom kullanılarak yeni bir kromozom (deneme kromozomu) elde edilmiştir. Yeni jenerasyona (G=G+1) geçecek olan kromozomun belirlenmesinde kriter uygunluk değerleridir. Hedef kromozomun uygunluk değeri zaten bilinmektedir. Bu aşamada hesaplanacak olan $u_{i,G+1}$' e ait uygunluk değeridir. Problemin amaç fonksiyonuna $u_{i,G+1}$' e ait tüm u_j değerleri girilerek kromozomun değeri hesaplanır.

2.6. Seçim

Seçim operatörü ile mevcut jenerasyon ve üretilen yeni kromozomlar değerlendirilerek yeni jenerasyon oluşturulur. Kromozomların yeni jenerasyonda yer alma olasılıkları uygunluklarına bağlıdır. DGA' da karşılaştırma birebir yapıldığından seçim için karmaşık prosedürü olan seçim operatörlerine ihtiyaç duyulmamaktadır. Karşılaştırılan kromozomlardan uygunluğu yüksek olan kromozom yeni jenerasyonun bireyi olarak atanmaktadır. Seçim operatörüne ait işlem Denklem (7)' de görülmektedir.

$$\forall i \le \mathrm{NP}: \; x_{i,\mathrm{G+1}} = \begin{cases} x_{u,\mathrm{G+1}} & \text{eğer } \mathrm{f}(x_{u,\mathrm{G+1}}) \le \mathrm{f}(x_{i,\mathrm{G}}) \\ x_{i,\mathrm{G}} & \text{diğer durumlarda} \end{cases} \tag{7}$$

2.7. Algoritmanın durdurulması

Anlatılan operatörler kullanılarak yeni jenerasyonlar elde edilmektedir. Amaç sürekli daha iyi uygunluk değerine sahip kromozomlar elde etmek ve optimumu yakalamak ya da optimuma yaklaşmaktır. Bu döngü $G=G_{max}$ olana kadar devam ettirilmektedir. Algoritmanın durdurulması kriteri olarak, populasyondaki en iyi ve en kötü uygunluk değerleri arasındaki farkın çok küçük bir rakama ulaşması olarak da belirlenebilmektedir (Ali ve Törn, 2004).

$$f_{max} - f_{min} \leq \varepsilon \tag{8}$$

ε değeri çok küçük bir sayıdır, parametre olarak kullanıcı tarafından belirlenmektedir (10^{-6} gibi). Farklı bir durdurma kriteri olarak belli bir değerin altına düşmesi de belirlenebilir. Bu çalışmada algoritma belirlenen iterasyon sayısı kadar çalıştırıldıktan sonra durdurulmaktadır ($G=G_{max}$).

3. DGA' NIN OPTİMİZASYON PROBLEMİNE UYGULANMASI

Bu bölümde, DGA literatürden alınan doğrusal olmayan optimizasyon problemine uygulanmıştır. Karşılaştırma yapılırken, Problem 1 ve Problem 2 için Michalewicz'in (1992) elde etmiş olduğu basit ikili GA çözümleri kullanılmıştır. Problem 3 ve Problem 4'te ise Tang ve diğerlerinin (1998) elde etmiş oldukları üç farklı algoritmaya ait sonuçlar kullanılmıştır. Bunlar ceza fonksiyonu metodu (penalty function method (PFM)), geleneksel genetik algoritma (traditional genetic algorithm (GA)) ve melez genetik algoritmadır (hybrid genetic algorithm (HGA)). Problemler sırasıyla Denklem 9-12'de verilmiştir.

Problem 1 (Michalewicz, 1992):

$$\max\ f(x) = 21.5 + x_1 \sin(4\pi x_1) + x_2 \sin(20\pi x_2), \tag{9}$$

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

Problem 2 (Michalewicz, 1992):

$$\min\ f(x) = \sum_{i=1}^{5} i \cos\left[(i+1)x_1 + i\right] \sum_{i=1}^{5} i \cos\left[(i+1)x_2 + i\right], \tag{10}$$

ve $-10 \leq x_1 \leq 10$,
$-10 \leq x_2 \leq 10$.

Problem 3 (Bazaraa, 1985):

$$\max\ f\ (x) = -2\ x_1^2 + 2\ x_1\ x_2 - 2\ x_2^2 + 4\ x_1 + 6\ x_2, \tag{11}$$

ve $x_1 + x_2 \leq 2,$
$x_1 + 5x_2 \leq 5,$
$x_1, x_2 \geq 0.$

Problem 4 (Bazaraa, 1985):

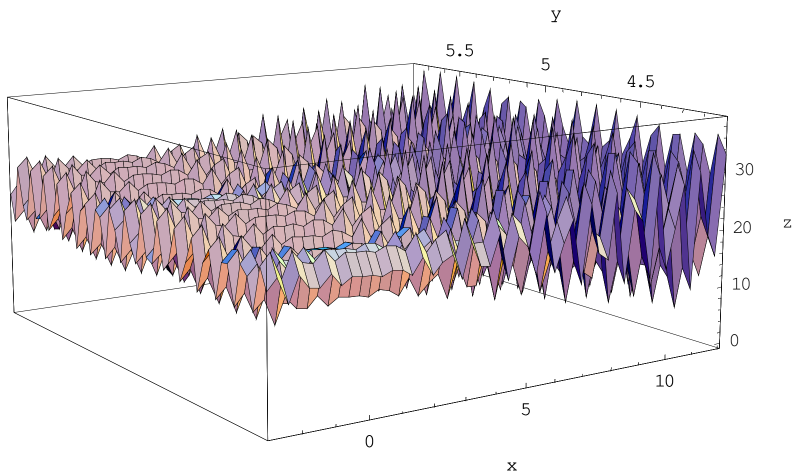
$$\max\ f\ (x) = -2\ x_1^2 + 2\ x_1\ x_2 - 2\ x_2^2 + 4\ x_1 + 6\ x_2, \tag{12}$$

ve $x_1^2 - x_2 \leq 0,$
$x_1 + 5x_2 \leq 5,$
$x_1, x_2 \geq 0.$

Diferansiyel gelişim algoritmasına ait kodlar MATLAB programlama dilinde yazılmış ve AMD 1.83 GHz, 256 MB RAM donanıma sahip bir bilgisayarda çalıştırılmıştır.

Michalewicz, problemleri (Problem 1 ve Problem 2) basit ikili genetik algoritma ile çözmüştür. Problem 1 için 20 kromozomdan oluşan populasyon büyüklüğü ve 396 iterasyon sonucu elde ettiği en iyi değer 38.827553 ve Problem 2 için raporlanan en iyi değer -186.5'tir (Michalewicz, 1992). Problem 1'e ait çözüm uzayı Şekil 2' de verilmişti.

Grafikten de anlaşılacağı üzere probleme ait çok sayıda lokal optimum nokta söz konusudur. DGA ile hedeflenen çok sayıdaki lokal optimumlara takılmadan global optimuma hızlı bir şekilde ulaşmaktır. Problem, DGA' da iki genden oluşan gerçek değerli kromozomlarla temsil edilmektedir. Kromozomların değeri, genlerdeki değişkenlerin Denklem (9)' a ürettikleri sonuçla ölçülmektedir. Amaç fonksiyonu ve problem değişkenleri DGA' da tanımlandıktan sonra 100, 200, 300 ve 396 jenerasyon için 100' er kere çalıştırılmıştır. Modelde populasyon büyüklüğü 20, CR 0.8 ve F=0.8 olarak belirlenmiştir. 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 Tablo 1' de verilmiştir.



Şekil 2. Fonksiyona Ait Çözüm Uzayı

Tablo 1. Problem 1'e Ait Sonuçlar (F=0.8)

	G=100	G=200	G=300	G=396
Maksimum	38.85029	38.8503	38.8503	38.8503
Minimum	38.71147	38.731	38.7326	38.7328
Ortalama	38.76656	38.8083	38.8264	38.8375
Standart sapma	0.04614	0.05121	0.04407	0.03576
Değişim katsayısı (%)	0.11902	0.13196	0.11351	0.09208
>38.827553	16	56	76	88
=38.850294	3	37	65	82

Yapılan denemeler sonucu F katsayısının değişken olmasının performans üzerinde oldukça etkili olduğu görülmüştür. Bu nedenle problemlerin çözümünde F değeri üç farklı şekilde kullanılmıştır. İlk olarak sabit bir değer olarak F 0.8 olarak alınmıştır. İkinci ve üçüncü çalıştırmalarda ise rasgele olarak (0,1) ve (0,3) aralıklarında belirlenmiştir (Tablo 2 ve Tablo 3).

Tablo 2. Problem 1'e Ait Sonuçlar (F=Rasgele(0,+1))

	G=100	G=200	G=300	G=396
Maksimum	38.8503	38.8503	38.8503	38.85029
Minimum	38.6847	38.7328	38.7328	38.7229
Ortalama	38.7794	38.8074	38.8197	38.83613
Standart sapma	0.05633	0.05496	0.05026	0.037397
Değişim katsayısı (%)	0.14526	0.14162	0.12947	0.096294
>38.827553	38	61	73	87
=38.850294	25	51	66	82

Tablo 3. Problem 1'e Ait Sonuçlar (F=Rasgele(0,+3))

	G=100	G=200	G=300	G=396
Maksimum	38.8503	38.8503	38.8503	38.85029
Minimum	38.7283	38.6503	38.7328	38.73281
Ortalama	38.7867	38.8189	38.8256	38.84165
Standart sapma	0.05738	0.05313	0.04772	0.030157
Değişim katsayısı (%)	0.14794	0.13687	0.12291	0.077641
>38.827553	45	73	79	92
=38.850294	31	68	77	88

Problem 1'e ait sonuçlar incelendiğinde, belirlenen problem için DGA' nın ikili GA' ya göre çok daha iyi sonuçlar ürettiği görülmüştür. İkili GA ile çözümde, 396 iterasyon sonucunda optimuma ulaşılamazken, DGA 100 iterasyon sonucunda dahi 45 kere (F=rasgele(0,+3)) optimum sonucu bulmuştur. İterasyon sayısı GA ile aynı (396) belirlendiğinde ise DGA 92 kez ikili GA' ya göre daha iyi sonuç bulmuştur. Optimumu bulma sayısı ise 88 olarak belirlenmiştir. F parametresi açısından bakıldığında, değişken olarak belirlendiği durumlarda daha iyi sonuçlar verdiği anlaşılmaktadır. Değişim katsayıları incelendiğinde tüm F parametreleri için ve tüm G değerlerinde algoritmanın birbirine çok yakın değerler ürettiği söylenebilir. Ortalamanın da optimuma yakın olması, tüm çalıştırmalarda optimum ya da optimuma yakın değerler elde edilmektedir kanaatine varılabilir.

Problem 2' de uygunluk fonksiyonu Denklem (10)'dur. Algoritma 100 jenerasyon için 100' er kere çalıştırılmıştır. Çünkü yapılan denemeler sonucu optimum ya da optimuma yakın değerlere DGA ile çok hızlı bir şekilde ulaşıldığı görülmüştür. Sonuçlar farklı F parametreleri ve 100 çalıştırma için maksimum, minimum, ortalama, standart sapma, -186.5'ten daha iyi değer sayısı ve optimumu (-186.7309088) bulma sayısı olarak düzenlenmiştir (Tablo 4).

Tablo 4. Problem 2'ye Ait Sonuçlar

	F=0.8	F=rasgele(0,+1)	F=rasgele(0,+3)
Maksimum	-186.7309088	-186.7309	-186.7309088
Minimum	-186.7306362	-186.7309	-123.5767709
Ortalama	-186.7309061	-186.7309	186.0993651
Standart sapma	2.86083E-05	9.45E-06	6.315413564
>38.827553	100	100	99
=38.850294	98	100	94

Problem 2 de Problem 1 gibi kısıtsız doğrusal olmayan bir problemdir. Sonuçlara bakıldığında DGA ile 100 iterasyonda hızlı bir şekilde optimum ya da optimuma çok yakın değerler bulunmaktadır. Önceki problemden farklı olarak F parametresinin değişim aralığı büyüdüğünde (0,+3) sonuçlar iyileşmemiş, değişim aralığının 0-1 olduğu duruma göre daha kötü sonuç vermiştir. Buradan anlaşılmaktadır ki F parametresi belirlenirken denemeler yapılmalı, probleme özgün F parametresi belirlenmelidir. Ancak her iki durumda da değişken olduğunda sabit olduğu duruma göre daha iyi sonuçlar vermiştir.

Problem 3 ve Problem 4 kısıtlı doğrusal olmayan optimizasyon problemleridir. Tang ve diğerleri (1998) her iki problemi PFM, GA ve kendi geliştirmiş oldukları HGA ile çözmüşlerdir. HGA için populasyon büyüklüğü 50, iterasyon sayısı 50 ve algoritma 50 kere çalıştırılmıştır. Sonuç olarak ilgili kaynakta maksimum değer, minimum yüzde sapma, ortalama yüzde sapma ve en kötü yüzde sapma raporlanmıştır. Her iki problem için elde edilen bu sonuçlarla DGA sonuçları Tablo 5'te verilmiştir. DGA sonuçlarında hata yüzdeleri DGA'nın bulduğu 7.03680 değeri kullanılarak hesaplanmıştır.

Tablo 5. Problem 3 Ve Problem 4'e Ait Sonuçlar (F=0.15)

Problem	Algoritma	Bazaraa sonuçları	Maksimum	Minimum hata (%)	Ortalama hata (%)	Maksimum hata (%)
Problem 3	PFM	7.160	6.995	2.30	8.25	10.3
	GA		7.10	0.84	15.94	21.4
	HGA		7.16085	0.000	0.15	0.261
	DGA		7.16129	0.000	0.017	0.186
Problem 4	PFM	6.613086	6.145	3.04	10.45	12.5
	GA		6.387	3.45	14.48	19.68
	HGA		6.61305	0.001	0.088	0.153
	DGA		7.03680	0.000	0.007	0.147

Tablo 5 incelendiğinde DGA'ın iki problem içinde çok daha iyi sonuçlar ürettiği görülmektedir. Problem 3 için değişkenlerin değerleri sırası ile 1.129032 ve 0.774194 ve uygunluk değeri 7.16129 olarak bulunmuştur. Bu değer her üç algoritmadan ve problemin orijinal kaynağında bulunan sonuçtan daha iyidir. Yüzde hatalara bakıldığında da DGA'nın her çalıştırmada birbirine benzer ve iyi sonuçlar ürettiği söylenebilir. Problem 4'te değişkenler 0.904988 ve 0.819002, fonksiyon değeri ise 7.03680 bulunmuştur. Problem 4'te de ilgili kaynaktaki çözümlerden daha iyi sonuçlar bulunduğu söylenebilir. Yine ortalama hataya bakıldığında her çalışmada oldukça iyi sonuçlar ürettiği söylenebilir.

Son iki problem için F parametresi belirlenirken daha küçük değerlerin daha iyi sonuçlar verdiği görülmüştür. Bunun sebebinin, değişkenlerin daha küçük değerlerde iyi sonuçlar vermesi, buna bağlı olarak değişimlerin küçük olmasının çözüm uzayındaki armada daha etkili sonuçlar vermesi olduğu düşünülmektedir. F' in değeri seçilirken, arama sürecindeki adımların nasıl olması gerektiği, değişkenlerin yapısı ve çözüm uzayı göz önünde bulundurulmalıdır.

4. SONUÇ

Literatürde DGA ile ilgili yapılmış olan çalışmalar incelendiğinde, özellikle sürekli parametrelerin söz konusu olduğu problemlerde oldukça başarılı sonuçlar verdiği görülmüştür. Çalışmamızda da, aşamaları ve çalışma prensipleri anlatılan DGA literatürden alınan farklı doğrusal olmayan optimizasyon problemlerine uygulanmış ve farklı genetik algoritma tiplerine göre çok daha iyi sonuçlar ürettiği görülmüştür. Özellikle parametrelerin doğru seçilmesiyle performansın oldukça iyi bir seviyeye gelebileceği anlaşılmaktadır. Parametre seçiminde literatürden özellikle uygulama alanına yönelik çalışmalardan faydalanmak, ayrıca test problemleri üzerinde denemeler yaparak karar vermek gerekmektedir.Karşılaştırma yapılırken ilgili kaynaklardaki performans kriterleri kullanılmıştır. Böylelikle farklı parametreler açısından bir karşılaştırma yapılmasına imkan verildiği düşünülmektedir.

Bu çalışmada, dilimizde hakkında oldukça sınırlı çalışmanın olduğu DGA tanıtılmaya, temel prensipleri açıklanmaya çalışılmıştır. DGA' yı teorik ya da pratik sahada kullanmayı düşünenlerin bu çalışmada verilen kaynaklarla sınırlı kalmayıp, uygulama alanına yönelik çalışmalar üzerinde yoğunlaşmalarında fayda olacaktır. GA gibi basit bir yapıya sahip olan DGA, optimizasyon problemlerine kolaylıkla uygulanabilir. Operatörlerin her bir birey için birlikte kullanılması, özel bir seçim yöntemi gerektirmemesi gibi nedenlerle GA' dan daha basit bir yapıya sahip olduğu söylenebilir. Esnek yapısı sayesinde farklı problemlere adaptasyonu da kolay olmaktadır. Başlangıç çözümünden bağımsızlık, paralel arama ve bunun sonucu olarak hızlı bir şekilde çözüme ulaşma DGA nın GA ile ortak avantajlarındandır. DGA ayrıca sürekli değerlerin kullanıldığı optimizasyon problemlerine GA' dan daha iyi sonuçlar üretmektedir. Yapılmış ve halen devam etmekte olan çalışmalarda DGA' nın farklı tipteki verilere (kesikli, sıralı sayı sistemi gibi) de uygulanabileceğini ve başarılı olabileceğini göstermektedir. Parametre seçimi ve kısıtlı problemlerin kısıtsıza dönüştürülmesinde zorluklar içermesine rağmen DGA doğrusal olmayan optimizasyon problemlerinde oldukça başarılıdır. Performansları her geçen gün artan bilgisayarlar, iterasyonların rahatlıkla arttırılmasına; böylelikle elde edilebilecek en iyi sonuçlara hızlı bir şekilde ulaşmaya imkan tanımaktadır.

5. KAYNAKÇA

Ali, M. M., Törn, A., (2004), "Population Set-Based Global Optimization Algorithms: Some Modifications and Numerical Studies", Computer & Operations Research, 31, 1703-1725.

Bazaraa, M.S., Shetty, L.M., (1985), "Non-Linear Programming: Theory and Algorithms", A.B.D., John Wiley & Sons.

Becerra, R.L., Coello, C.A.C., (2005), "Cultured Differential Evolution for Constrained Optimization", Comput. Methods Appl. Mech. Engnrg., Baskıda.

Bergey, P.K., Ragsdale, C., (2005), "Modified Differential Evolution: A Greedy Random Strategy for Genetic Recombination", Omega, 33, 255-265.

Goldberg, D.E., (1989), "Genetic Algoritms in Search Optimization and Machine Learning", A.B.D., Addison Wesley Publishing Company.

Holland, J. H., (1975), "Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence", University of Michigan Press.

Hrstka, O., Kucerova, A., (2004), "Improvemenets of Real Coded Genetic Algorithms Based on Differential Operators Preventing Premature Convergence", Advances in Engineering Software, 35, 237-246.

Storn, (2001), http://www.icsi.berkeley.edu/~storn/code.html, [02.02.2006].

Karaboğa, D, (2004), "Yapay Zeka Optimizasyonu Algoritmaları", İstanbul, Atlas Yayın Dağıtım.

Lin, Y.C., Hwang, K.S., Wang, F.S., (2004), "A Mixed-Coding Scheme of Evolutionary Algorithms to Solve Mixed-Integer Nonlinear Programming Problems", Computers and Mathematics with Applications, 47, 1295-1307.

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.

Shiakolas, P. S., Koladiye, D., Kebrle, J., (2005), "On The Optimum Synthesis of Six-Bar Linkages Using Differential Evolution and The Geometric Centroid of Precision Positions Technique", Mechanism and Machine Theory, 40, 319-335.

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.

Sun, J., Zhang, Q., P.K. Tsang, E., (2005), "DE/EDA: A New Evolutionary Algorithm for Global Optimization", Information Sciences, 169, 249-262.

Tang, J.F., Wang, D., (1998), "A Hybrid Genetic Algorithm for A Type of Nonlinear Programming Problem", Computers Math. Applic., 36, 11–21.

Zaharie, D., (2002), "Critical Values for Control Parameters of Differential Evolution Algorithms", Proceedings of Mendel 2002, 8th International Conference on Soft Computing.