GERÇEK ZAMANLI OPTİMİZASYON İÇİN GELİŞİME DAYALI HIZLI BİR ALGORİTMA

Selçuk ÖKDEM[1] **Derviş KARABOĞA**[2]

[1,2]Bilgisayar Mühendisliği Bölümü
Mühendislik Fakültesi
Erciyes Üniversitesi, 38039, Kayseri
[1]e-posta: okdem@erciyes.edu.tr [2] e-posta: karaboga@erciyes.edu.tr

Anahtar sözcükler: Diferansiyele dayalı gelişim, genetik algoritma, gerçek zamanlı optimizasyon, optimizasyon

ÖZET

Bu bildiri, diferansiyele dayalı gelişim (Differential Evolution, DE) algoritmasının, iyi bilinen bir optimizasyon algoritması olan Genetik Algoritmaya (GA) göre performans analizini sunmaktadır. DE algoritması oldukça yeni bir sezgisel yaklaşımdır ve temel avantajları; küresel minimumu başlangıç parametre değerlerinden bağımsız olarak bulabilmesi, bölgesel minimuma hızlı yakınsaması ve az sayıda kontrol parametresi kullanmasıdır. Bu çalışmada, DE algoritmasının performansı bazı popüler test fonksiyonları kullanılarak analiz edilmiştir.

1. GİRİŞ

Optimizasyon işlemlerinde uygun bir çözüm alanını elde etmek için çok çeşitli arama teknikleri kullanılmaktadır. Bunların çoğu üç temel sınıfta gruplandırabilir. Bunlardan birincisi hesaplama tabanlı tekniktir. İkincisi, çözüm uzayında her noktayı ele aldığımız ve bunları değerlendirdiğimiz tekniktir. Hesaplama yoğunluğu nedeniyle bu türden bir yaklaşım, özellikle basit problemler haricinde sınırlı fayda sağlamaktadırlar. Üçüncüsü ise yönlendirme yaparak rasgele arama yapabileceğimiz bir tekniktir. Bu tekniği kullanırken, arama işlemlerinde daha iyiye gitmek için sezgisellikten faydalanılmaktadır. Gelişime Dayalı (GD) algoritmalar bu gruba dahil edilen algoritma türleri olarak karşımıza çıkmaktadır [1].

GD algoritması direk arama algoritmalarının bir türüdür. Bir geleneksel direk arama metodu, dizayn parametre vektörlerinin değişimlerini üreten bir strateji kullanmaktadır. Bir değişim üretildiği zaman yeni parametre vektörü olarak kabul edilir veya reddedilir. Kabul edilme, minimizasyon probleminde objektif fonksiyon değerinin azaldığı durumda gerçekleşir. Bu metotla hızlı bir yakınsama sağlanır ancak yerel minimuma takılma ihtimali söz konusudur. Bu sorun, farklı vektörlerin aynı anda çalıştırılmalarıyla giderilebilir. Bu yönde bir yaklaşım DE algoritmasının ana fikrini oluşturmaktadır.

GA, çok popüler bir GD algoritmasıdır. Birçok genetik algoritma türünün geliştirilmesine rağmen çaprazlama, seleksiyon ve mutasyon işlemleri ayrı ayrı gerçekleştirildiği için uzun zamana ihtiyaç duyulmaktadır. Bu dezavantajın giderilmesi için DE olarak isimlendirilen GD stratejisi Storn tarafından önerilmiştir [2,3].

DE' yi self-adaptif yapan mutasyon işlemi ve seleksiyon işlemi, GA ile DE algoritması arasındaki temel farklılığı oluşturmaktadır. DE' de bütün çözümler uygunluk değerlerine bakılmaksızın ebeveyn olarak aynı seçilebilme şansına sahiptirler. GA' lardan daha iyi bir yakınsama performansı sağlayacak şekilde, yeni çözümler ve ebeveynleri yarışmayı kazanmaktadırlar.

DE algoritması, kısıtlamaları göz önünde bulundururken, problem objektiflerini modelleyerek objektif fonksiyonunu minimuma götüren bir stokastik optimizasyon metodudur. Algoritma temel olarak üç avantaja sahiptir. Bunlar; başlangıç parametrelerine bağlı olmaksızın gerçek global minimumun bulunması, hızlı yakınsaması ve az sayıda kontrol parametresi içermesidir [2].

Gerçek global minimum değerinin bulunabilmesi GD algoritmalarının performansını belirleyen temel kriterlerden biridir. DE' nin De Jong test fonksiyonlarındaki performans karşılaştırması iyi bilinen GA türleri olan PGA, Eshelman ve Grefensstette algoritmaları için yapılmış ve mühendislik alanında optimizasyon problemleri için gelecek vadeden bir GD algoritması olarak önerilmiştir [4]. Bu çalışmada DE' nin bu anlamdaki performansı, GA' ların popüler bir türü olan Düzenli Durumlu Genetik Algoritma (Steady State Genetic Algorithm, SSGA) ile karşılaştırılmıştır. İkinci bölümde DE algoritmasının çalışma prensipleri, üçüncü bölümde genetik algoritmalar ve SSGA incelenecektir. Dördüncü bölümde kullanılan test fonksiyonları açıklanacak ve son bölümde yorum ve karşılaştırmalar yapılacaktır.

2. DİFERANSİYEL GELİŞİME DAYALI ALGORİTMA

DE algoritması çaprazlama, mutasyon ve seleksiyon gibi GA' larda bulunan benzer operatörleri kullanan populasyon tabanlı bir algoritmadır. Daha iyi çözümler üretme işleminde DE ve GA arasında görülen temel fark çaprazlama ve mutasyon operasyonlarında yer almaktadır. Algoritma mutasyon operatörünü bir arama mekanizması olarak kullanmaktadır. Seleksiyon operatörü ise arama uzayında daha iyi bölgelere yönlendirilmek amacıyla kullanılmaktadır.

D parametresinden oluşan bir optimizasyon işlemi *D* boyutlu vektör olarak temsil edilir. Başlangıçta *NP* (populasyon sayısı) populasyonlu vektörler rasgele olarak oluşturulurlar. Populasyon; mutasyon, çaprazlama ve seleksiyon operatörlerinin uygulanmasıyla başarılı bir şekilde geliştirilir.

Mutasyon : Her bir $x_{i,G}$ için bir mutasyon vektörü aşağıdaki gibi üretilir.

$$v_{i,G+1} = x_{i,G} + K \cdot (x_{r1,G} - x_{i,G}) + F \cdot (x_{r2,G} - x_{r3,G}) \qquad (1)$$

Burada $i, r_1, r_2, r_3 \in \{1,2,..., NP\}$ rasgele seçilmiştir ve birbirinden farklı değerdedir. F, $(x_{r2,G} - x_{r3,G})$ fark vektörünün bir oranlama faktörüdür.

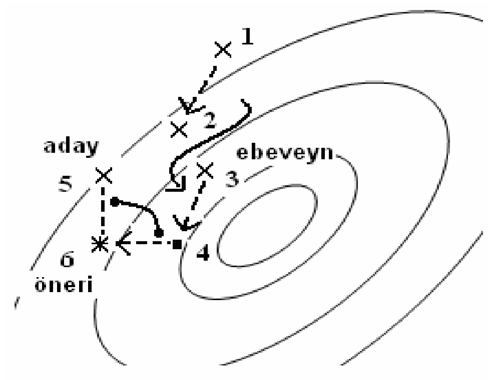
Çaprazlama : Ebeveyn vektör, mutasyona uğramış vektör ile birleşir ve bir deneme vektörü ($u_{ji,G+1}$) oluşturulur.

$$u_{ji,G+1} = \begin{cases} v_{ji,G+1} & \text{if } (rnd_j \leq \text{CR}) \text{ or } \quad j = rn_i \\ x_{ji,G} & \text{if } (rnd_j > \text{CR}) \text{ and } \quad j \neq rn_i \end{cases} \qquad (2)$$

Burada j =1,2,…..,D ve $r_j \in [0,1]$ aralığında rasgele bir değerdir. CR $\in [0,1]$ aralığında çaprazlama oranı ve $rn_i \in (1,2,\ldots,D)$ rasgele seçilen indeksdir.

Seleksiyon : Populasyon içerisinde bütün çözümler uygunluk değerlerine bakılmaksızın ebeveyn olarak aynı seçilebilme şansına sahiptirler. Yeni vektör mutasyon ve çaprazlama operasyonları değerlendirildikten sonra üretilmektedir. Sonra, yeni vektör ile ebeveyn vektörünün performansı mukayese edilerek daha iyi olan seçilmektedir. Eğer ebeveyn hala daha iyi durumda ise populasyon içerisinde tutulmaya devam edilmektedir.

DE işlevi Şekil 1'de detaylı olarak gösterilmektedir: iki populasyon üyesi (1,2) arasındaki fark üçüncü populasyon üyesine (3) eklenmekte, sonuç (4) aday (5) ile çaprazlamaya tabi tutulmakta ve bir çözüm önerisi (6) elde edilmektedir. Öneri değerlendirilmekte ve daha iyi bulunduğu takdirde aday yerini almaktadır [4].



Şekil-1. DE' de yeni bir çözümün elde edilmesi.

3. SİMÜLASYONDA KULLANILAN TEST FONKSİYONLARI

Tablo 1, yaygın olarak kullanılan test fonksiyonlarını içermektedir. Bu fonksiyonlardan ilk beş tanesi De Jong tarafından önerilmiştir [5]. De Jong' dan sonra da bu fonksiyonlar GA ve diğer algoritma araştırıcıları tarafından yaygın olarak kullanılmıştır. İlk dört fonksiyon (F1-F4) unimodal (tek optimum nokta içeren) tipinde, diğer fonksiyonlar multimodal (bir çok yerel minimum ancak bir genel minimum içeren) tipte fonksiyonlardır. Sphere[F1] fonksiyonu düzgün, unimodal ve ileri derecede konveks bir fonksiyondur. Rosenbrock[F2] fonksiyonu birçok dar tepecik içerdiğinden dolayı zor bir fonksiyon olarak düşünülmektedir. Tepe noktaları çok keskindir. Algoritmalar bu problemde ilerlenebilecek iyi noktaları tayin edememektedirler. Step[F3] fonksiyonu düzgün yüzey problemlerini temsil eden parçalı sürekli adım fonksiyonudur. Düz yüzeyler çözüme ilerleme yönü hakkında bir bilgi vermediği için optimizasyon algoritmaları için engel teşkil eden bölgelerdir. Quartic[F4] fonksiyon gürültü eklenmiş basit bir unimodal fonksiyondur. Gaussian türünde gürültü ile aynı noktada hiç bir zaman aynı değerin elde edilmemesi sağlanmaktadır. Bu fonksiyonda başarılı olamayan algoritmalar için gürültülü verilerde iyi sonuç veremeyeceği yorumu çıkarılabilmektedir. Foxholes[F5] birçok yerel minimum içeren bir fonksiyondur. Birçok standart optimizasyon algoritması bu fonksiyonda ilk karşılaştığı tepe noktasına takılmaktadır.

Schwefel, Rastrigin ve Griewank [F6-F8] fonksiyonları doğrusal olmayan multimodal fonksiyon türlerinin tipik örnekleridir. Rastrigin[F7] fonksiyonu geniş arama uzayı ve birçok yerel minimuma sahip olduğu için zor bir problem türüdür. Fonksiyon ileri derecede multimodal' dır. Schwefel fonksiyonu Rastrigin fonksiyonundan daha kolay olan ve genel minimumun çok uzağında ikinci en iyi minimuma sahip olan bir fonksiyondur. Griewank fonksiyonunda, toplama terimi fonksiyona parabollük özelliği kazandırmaktadır. Bu fonksiyonda yerel minimum derecesi parabollük derecesinden daha üst seviyededir. Çarpım terimi baz alınarak arama uzayının boyutları artırılmakta ve yerel minimumlar azaltılmaktadır.

Tablo 1' de fonksiyon tanımlamaları, matematiksel ifadeler ve çözüm arama işleminde sınır bölgeleri olarak kullanılan limit değerleri verilmiştir [6,7].

4. GA ve SSGA

GA' lar, GD algoritmaların en yaygın olarak bilinen türleridir. Bunlar doğal seleksiyon ve genetik prensiplerini taklit ederek çalışan, arama ve optimizasyon problemlerinde kullanılan adaptif metotlardır. GA' ların genel formu 1965 yılında John Holland tarafından geliştirilmiştir [8]. Bilinen birçok GA uygulamalarının başlama noktası olan bu çalışma ve onun doktora öğrencilerinin çalışmaları 1970' lere kadar bu alanda çalışan araştırmacılara rehberlik eden sınırlı kaynaklardandır [9]. Holland' ın önerdiği GA yapısı, çaprazlama, mutasyon ve seleksiyon olmak üzere üç operatöre sahiptir. Bu üç operatör gelişim işleminin, yeni alternatif bölgelere gitmesine veya o an üzerinde çalışılan bölgede devam edilmesini karar veren önemli bir işleve sahiptir. Basitliği, esnekliği ve işlev kolaylığı GA' ların birçok mühendislik ve fen bilimleri alanında başarılı bir şekilde kullanılmasını sağlamıştır.

Düzenli Durumlu Genetik Algoritma (SSGA)' nın temel özelliği kendisine özgü populasyon kullanımıyla karşımıza çıkmaktadır. SSGA, verilen bir başlangıç populasyon büyüklüğüyle başlar. Algoritma, temel populasyondan seçilen belirli bireylerin üretilmesinden sonra elde edilen çocuk bireyleri içeren yardımcı bir populasyon oluşturulmaktadır. Yeni üretilen çocuk bireyler değerlendirilmeye tabi tutularak temel populasyona eklenmektedir. Sonuç populasyonundaki her bir birey bir penaltı fonksiyonuna tabi tutularak uygunluk değerlerine göre sıralanmaktadır. Daha sonra sırayla kötü bireyler çıkartılarak populasyonun orjinal büyüklüğü korunmaktadır. Böylece iyi bireyler populasyonda tutulmuş olmaktadır. Her yeni populasyon oluşturulduğunda algoritma, sonlandırma kriterinin sağlanıp sağlanmadığına bakmaktadır. Sonlanma kriteri sağlanmamışsa yukarıda belirtilen işlemlere devam edilmektedir, aksi durumda algoritma durmaktadır.

5. SİMULASYON SONUÇLARI

Simulasyonda kullanılan DE algoritması populasyon sayısı (*NP*), oranlama faktörü (*F*), kombinasyon faktörü (*K*) ve çaprazlama oranı (*CR*) olmak üzere üç kontrol parametresine sahiptir. DE algoritmasının probleme bağlı parametreleri ise sonlandırılma kriteri olan çözüm değerinde tekrar sayısı (*R*) ve problem boyutunu belirleyen *D*' dir. Bu iki parametrenin değerleri optimize edilecek probleme bağlıdır. *R* değeri bu çalışmada 100 olarak alınmıştır. Bu durumda algoritmanın sonlanma kriteri elde edilen en iyi bireyin 100 iterasyon geçmesine rağmen aynı değeri (±0.001) muhafaza etmesi anlamına gelmektedir. Karşılaştırmada kullanılan SSGA için de aynı sonlandırma kriteri kullanılmıştır. DE algoritmasında kullanılan parametre değerleri *F* için 0.8, *K* için 0.5 ve *CR* için 0.8 seçilmiştir. Tablo 2'de yer alan DE sonuçları 50 ayrı koşma için alınan değerlerin ortalamasını ifade etmektedir. Her bir koşmada rastgele üretilen farklı başlangıç populasyonları kullanılmıştır.

Bu çalışmada DE' nin performansı SSGA algoritmasının performansı ile karşılaştırılmıştır. Tablo 2' de belirtilen populasyon sayıları için SSGA ve DE' nin ürettiği fonksiyon sonuçları yer almaktadır.

DE algoritmasının ürettiği fonksiyon sonuçları genel olarak değerlendirildiğinde, hemen hemen her fonksiyon için SSGA' ya oranla başarılı bir gelişim gösterdiği görülmektedir. Burada, her fonksiyon için aynı kontrol parametre değerlerinin kullanıldığını belirtmek gerekmektedir. Parametre değerlerinin fonksiyon türüne uygun olarak ayarlanabilme olanağı söz konusu olursa, DE' nin performansında artış gözlenecektir. Tablo 2' de DE' nin F7 fonksiyonu için zorlandığı görülmektedir. Bu durumu, fonksiyonun geniş arama uzayına sahip olmasının, alınan kontrol parametre değerleri için sorun teşkil ettiği anlamına gelmesi şeklinde yorumlayabiliriz.

6. DEĞERLENDİRME

DE algoritması mühendislik alanındaki optimizasyon problemlerinde kullanabileceğimiz yeni bir sezgisel yaklaşımdır. Bu çalışmada, DE' nin çözüme ilerleme başarısı SSGA ile karşılaştırılmıştır. Sonuçlardan DE' nin hızlı bir gelişim sağladığı görülmüştür. Gerçek zamanlı uygulamalarda hız ve zamanın önemli birer parametre olmasından dolayı DE' yi birçok gerçek zamanlı optimizasyon problemlerinde başarılı bir şekilde kullanılabileceğimiz bir GD algoritması olarak görebiliriz.

KAYNAKLAR

[1] Bergey P.K. and Ragsdale C., Modified differential evolution: a greedy random strategy for genetic recombination, OMEGA, Vol 33, Iss 3, 2, pp 255-265, 2005.

[2] Storn, R., Differential Evolution, A Simple and Efficient Heuristic Strategy for Global Optimization over Continuous Spaces, JOURNAL OF GLOBAL OPTIMIZATION, Vol 11, pp 341-359, 1997.

[3] http://www.icsi.berkeley.edu/~storn/code.html

[4] Karaboğa D. and Ökdem S., A Simple And Global Optimization Algorithm For Engineering Problems: Differential Evolution Algorithm, ELEKTRIK, TURKISH

JOURNAL OF ELECTRICAL & COMPUTER SCIENCES, Vol 12, Num 1, ISSN 1300-0632, pp. 53-60, 2004.

[5] De Jong K.A., An Analysis of the Behavior of a class of Genetic Adaptive Systems. Phd thesis, University of Michigan, Dissertation Abstracts International 36(10), 5140B. (University Microfilms No. 76-9381), 1975.

[6] Salomon R., Reevaluating Genetic Algorithm Performance under coordinate rotation of Benchmark Functions, BIOSYSTEM, 39, Elsevier Science, pp 263-278, 1995.

[7] Digalakis J.G. and Margaritis K.G., On Benchmarking Functions for Genetic Algorithms, Intern. J. COMPUTER MATH, vol 00, pp 1-27, 1997.

[8] Holland J., Adaptation in Natural and Artificial Systems, AnnArbor: University of Michigan Press, 1975

[9] Martorell S., Carlos S., Sánchez A. and Serradell V., Constrained optimization of test intervals using a steady-state genetic algorithm, RELIABILITY ENGINEERING & SYSTEM SAFETY, Vol 67, Iss 3, pp 215-232, 2000.

Tablo-1. Unimodal ve Multimodal Fonksiyonlar

Fonksiyon	Matematiksel İfade	Limit Aralığı
F1	$f_1 = \sum_{i=1}^{3} x_i^2$	$-5.12 \le x_i \le 5.12$
F2	$f_2 = 100(x_1^2 - x_2)^2 + (1 - x_1)^2$	$-2.048 \le x_i \le 2.048$
F3	$f_3 = \sum_{i=1}^{5} \mathrm{int}(x_i)$	$-5.12 \le x_i \le 5.12$
F4	$f_4 = \sum_{i=1}^{30} i x_i^4 + Gauss(0,1)$	$-1.28 \le x_i \le 1.28$
F5	$f_5 = 0.002 + \sum_{j=1}^{25}\left(\frac{1}{j} + \sum_{i=1}^{2}(x_i - a_{ij})^6\right)$	$-65.536 \le x_i \le 65.536$
F6	$f_6 = 10V + \sum_{i=1}^{10}\left(-x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)\right)$, $10V = 4189.829101$	$-500 \le x_i \le 500$
F7	$f_7 = 20A + \sum_{i=1}^{20}\left(x_i^2 - 10\cos(2\pi x_i)\right)$, $A = 10$	$-5.12 \le x_i \le 5.12$
F8	$f_8 = 1 + \sum_{i=1}^{10}\left(\frac{x_i^2}{4000}\right) - \prod_{i=1}^{10}\left(\cos\left(\frac{x_i}{\sqrt{i}}\right)\right)$	$-600 \le x_i \le 600$

Tablo-2. Farklı populasyonlarda SSGA ile DE' nin ürettiği fonksiyon değerleri

NP		**50**	**100**	**150**	**200**	**250**	**300**	**350**	**400**
F1	SSGA	5.25	3.85	1.81	0.6	0.8	0.7	0.75	0.68
	DE	0	0	0	0	0	0	0	0
F2	SSGA	4.8	3.74	1.61	0.4	0.5	0.4	0.45	0.48
	DE	0	0	0	0	0	0	0	0
F3	SSGA	4	3.8	1.6	0.6	1.74	1.76	1.78	1.8
	DE	0	0	0	0	0	0	0	0
F4	SSGA	12.35	6.21	2	1.2	1.95	1.96	1.97	1.98
	DE	0.001	0.055	0.018	0.011	0.028	0.088	0.171	0.031
F5	SSGA	12	5	1.8	1	1.86	1.87	1.88	1.89
	DE	0.036	0	0	0	0	0	0	0
F6	SSGA	6900	5700	5380	4400	4316.25	4318.25	4320.25	4322.25
	DE	1723.614	1617.065	1596.368	1571.86	1536.25	1523.38	1480.34	1470.375
F7	SSGA	65	33.6	10	12.2	10.06	10.07	10.35	10.07
	DE	138.199	138.413	137.566	135.145	133.64	136.775	130.555	130.106
F8	SSGA	9.8	4.2	1.9	2	1.96	1.97	1.98	1.99
	DE	0.515	0.479	0.459	0.453	0.432	0.419	0.418	0.419