

GEZGİN SATICI PROBLEMİNİN DİFERANSİYEL GELİŞİM ALGORİTMASI İLE ÇÖZÜMÜ

Timur KESKİNTÜRK
İstanbul Üniversitesi

ÖZET

Gezgin satıcı problemi (GSP) çözümü zor optimizasyon problemlerindedir. Düğüm sayısı arttıkça zorluk derecesi üssel olarak artmaktadır. Çözümüne yönelik bir çok yöntem geliştirilmiştir. Özellikle sezgisellerle daha iyi sonuçlar alınmaya çalışılmıştır. Genetik algorithmadan esinlenilerek geliştirilmiş olan diferansiyel gelişim algoritması (DGA), sürekli değerlerden oluşan problemlerin çözümüne yöneliktir. Ancak sürekli verilerin kesikli verilere dönüştürülmesi ile GSP benzeri kesikli değerlerin söz konusu olduğu, permütasyon kodlama ile çalışan problemlere de uygulanabilmektedir. Bu çalışmada DGA, dönüştürme operatörü kullanılarak GSP'ye uygulanmıştır. Sonuçlar göstermiştir ki klasik DGA algoritması ve kullanılan dönüştürme operatörü düğüm sayısı artan problemler için uygun değildir. Birtakım modifikasyonlar ve farklı dönüştürme operatörleri denenerek algoritma geliştirilebilir.

Anahtar Kelimeler: Diferansiyel Gelişim Algoritması, Gezgin Satıcı Problemi.

1. GİRİŞ

Endüstriyel problemlerin 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 arttıkça zaman yetersiz kaldığından sezgisellerin cazibesi her geçen gün artmıştır. Diferansiyel gelişim algoritması (DGA) da optimizasyon problemlerinin çözümüne yönelik geliştirilmiş populasyon tabanlı sezgisellerdendir. İlk olarak sürekli parametrelerden oluşan problemlere yönelik olarak geliştirilmiştir. Performansından dolayı birtakım dönüştürme operatörleriyle kesikli problemlere uygulanabilecek yapıya dönüştürülmüştür. Sürekli veriler kesikli verilere dönüştürülmüş ve GSP benzeri problemlerin çözümünde kullanılabilir hale getirilmiştir. Daha önce iş çizelgeleme (Onwubolu ve diğerleri, 2006) ve LLDP (loop layout design problem) problemlerinde (Nearchou, Baskıda) kullanılan DGA bu çalışmada GSP için kullanılmıştır.

2. GEZGİN SATICI PROBLEMİ

Gezgin satıcı problemi, doğrusal olmayan, kombinatoriyel bir optimizasyon problemidir. Bir satış elemanının, ya da bir aracın belli bir noktadan başlayıp, tüm noktalara ya da

merkezlere uğradıktan sonra yine aynı noktaya, toplam mesafeyi en az kılacak şekilde dönmesi şeklinde tanımlanabilir. Düğüm olarak da adlandırılan noktaların sayısı (n) problemin boyutunu, diğer bir ifadeyle değişken sayısını göstermektedir. Probleme ait tüm mümkün çözümlerin sayısı (n-1)!'dir. Konuyla ilgili olarak Taha (2000) ve Timor (2001)'un eserlerine bakılabilir. Problemin tamsayı programlama modeli aşağıdaki gibidir:

Girdiler:

n = probleme ait düğüm sayısı,

c_{ij} = i düğümünden j düğümüne gidiş maliyeti (mesafe vb.)

Karar değişkenleri:

$$x_{ij} = \begin{cases} 1, & \text{eğer } i' \text{ den } j' \text{ ye gidiliyorsa} \\ 0, & \text{aksi durumda} \end{cases}$$

Minimize:

$$\sum_{i=1}^n \sum_{j=1}^n x_{ij} c_{ij} \quad (1)$$

Kısıtlar:

$$\sum_{i=1}^n x_i - 1 \quad (2)$$

$$\sum_{j=1}^n x_j - 1 \quad (3)$$

Düğüm sayısı arttıkça çözüm alternatifi sayısı üssel olarak artmakta ve problem, deterministik yöntemlerle daha çözümü zor hale gelmektedir. Bu durumda devreye sezgisel yöntemler girmektedir. Geliştirilmiş birçok sezgisel GSP problemlerinin çözümünde kullanılmıştır. Bunlara örnek olarak genetik algoritma (Potvin, 1996), yapay sinir ağları (Potvin, 1993), benzetilmiş tavlama (Aarts ve diğerleri, 1997), karınca kolonisi algoritması (Dorigo ve Gambardella, 1997) verilebilir. Bu çalışmada populasyon temelli gelişim algoritmalarından olan diferansiyel gelişim algoritması GSP probleminin çözümünde kullanılmıştır.

3. DİFERANSİYEL GELİŞİM ALGORİTMASI

Diferansiyel gelişim algoritması, Storn ve Price tarafından 1995 yılında geliştirilmiştir (Storn ve diğerleri, 1995). Sürekli verilerden oluşan problemlerde etkin sonuçlar verebilen, genetik algoritmaya dayanan populasyon temelli sezgisel optimizasyon tekniğidir (Mayer ve diğerleri, 2005; Storn ve diğerleri, 1995; Storn, 2001). GA'daki tüm operatörler farklı bir şekilde uygulanmaktadır. Populasyon bazlı çalışmamakta, tek tek kromozomlar

operatörlere tabi tutulmakta ve yeni bir birey elde edilmektedir. Bu yönüyle DGA hızlı çalışmakta ve basitçe kodlanabilmektedir. Diğer algoritmalar için binlerle ifade edilen satırdan oluşan kodlar söz konusu iken DGA için yaklaşık 20 satırlık kod yeterli olmaktadır (Mayer ve diğerleri, 2005: 317). GA'daki çaprazlama, mutasyon ve seçim operatörleri burada da söz konusudur. Operatörler kromozomlara uygulanmakta ve elde edilen bireyin uygunluk değerine göre seçim işlemi gerçekleştirilmektedir. Yerine üretilen bireyden daha iyi bir uygunluğa sahipse yeni birey değilse mevcut birey yeni jenerasyona geçmektedir.

3.1 Parametreler

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

Problemin amaç fonksiyonu ve değişkenler belirlendikten sonra aşağıdaki adımlar izlenerek DGA uygulanır.

3.2 Kodlama ve başlangıç popülasyonu

NP kromozom sayısıdır. NP adet D değişkenli kromozomdan meydana gelen başlangıç popülasyonu aşağıdaki gibi üretilir (Karaboğa, 2004: 172).

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

Aşağıda açıklanan operatörler G_{max}'a ulaşıncaya kadar uygulanır.

3.3 Mutasyon

DGA'da mutasyona uygulanacak kromozom dışında farklı olan üç kromozom seçilir ($r_{1,2,3}$). İlk ikisinin farkı alınır ve F parametresiyle çarpılır. F parametresi genellikle 0-2 arasında değerler almaktadır. Elde edilen kromozom ile üçüncü kromozom toplanır (Denklem 5).

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

3.4 Çaprazlama

Mutasyon sonucu elde edilen kromozom ve $x_{i,G}$ kromozomu kullanılarak deneme kromozomu ($u_{i,G+1}$) üretilir. Genler CR olasılıkla yeni kromozomdan 1-CR olasılıkla mevcut kromozomdan seçilir. $j = j_{\text{rand}}$ koşulu, en az bir genin yeni kromozomdan alınmasını garanti etmek için kullanılmaktadır.

$$\forall j \in D : x_{j,u,G+1} = \begin{cases} x_{j,n,G+1} & \text{eğer } \text{rand}[0,1] \leq \text{RC} \quad \forall j = j_{\text{rand}} \\ x_{j,i,G} & \text{aksi durumda} \end{cases} \quad (6)$$

3.5 Uygunluk Fonksiyonu

Yeni jenerasyona geçecek olan kromozomun belirlenmesinde kriter uygunluk değeridir. Mevcut kromozomun uygunluk değeri zaten bilinmektedir. $u_{j,i,G+1}$ 'e ait uygunluk değeri hesaplanır. Problemimizde uygunluk değeri toplam tur uzunluğudur. Her bir kromozom bir turu temsil etmektedir. Amaç bu değerinin minimum olmasıdır.

3.6 Seçim

Uygunluğu yüksek olan kromozom yeni jenerasyonun bireyi olarak belirlenir. Seçim operatörüne ait işlem Denklem (7)'de görülmektedir.

$$\forall i \in \text{NP} : x_{i,G+1} = \begin{cases} x_{u,G+1} & \text{eğer } f(x_{u,G+1}) \leq f(x_{i,G}) \\ x_{i,G} & \text{diğer durumlarda} \end{cases} \quad (7)$$

Seçim ikili karşılaştırmalar sonucu gerçekleştirilmektedir. GA'daki gibi populasyon tabanlı bir seçim söz konusu değildir.

3.7 Algoritmanın durdurulması

Operatörler kullanılarak yeni jenerasyonlar elde edilmektedir. Bu işlemler $G = G_{\text{max}}$ olana kadar devam etmektedir. $G = G_{\text{max}}$ olduğunda jenerasyondaki en iyi birey problemin çözümüdür.

4. DGA'NIN GSP'YE UYGULANMASI

DGA daha önce de belirtildiği gibi sürekli verilerin söz konusu olduğu problemlere yönelik olarak geliştirilmiştir. Ancak bir dönüştürme operasyonu ile algoritma kesikli değerler içeren problemler için de kullanılabilir hale getirilmiştir. Sürekli parametrelerin kesikli parametrelere dönüştürülmesi işlemi Michalewicz tarafından GA için gerçekleştirilmiştir (1992). Buna göre GSP'de ki düğüm sayısı kadar genden oluşan kromozomdaki rakamlar, büyüklüklerine göre sıralanmakta, değerlerin sıra numaralarına göre tur belirlenmektedir.

Aşağıdaki örnekte dokuz düğümden oluşan bir GSP problemine üretilen sonuç görülmektedir:

$$v = (2.34, -1.09, 1.91, 0.87, -0.12, 0.99, 2.13, 1.23, 0.55)$$

Buna göre GSP turu,

$$2 - 5 - 9 - 4 - 6 - 8 - 3 - 7 - 1,$$

şeklinde gerçekleşecektir. Sürekli parametrelerle çalışan GA'lar da operatörler uygulandıktan sonra probleme ait uygunluk değeri ve sonuçlar hesaplanmak istendiğinde bu dönüştürme operatörü uygulanır.

DGA'da da Michalewicz'in dönüştürme operatörü kolaylıkla uygulanabilmektedir. Bu çalışmada Michalewicz'in dönüştürme operatörü kullanılmaktadır.

5. UYGULAMA

Diferansiyel gelişim algoritması bu çalışmada düğüm sayısı 5, 10, 20, 40 ve 50 olan ve rasgele üretilmiş beş ayrı probleme uygulanmıştır (tsp5, tsp10, tsp20, tsp40 ve tsp50). Üretilen problemlere ait mesafeler tablosu ve optimum turlara <http://www.isletme.istanbul.edu.tr/ogretim/timur/tsps.htm> adresinden ulaşılabilir. Problemler üretilirken mesafeler 300-1500 arası rasgele belirlenmiştir. Her bir problem için algoritma 100 kere çalıştırılmış ve ortalama tur uzunlukları ile ortalama yüzde sapmalar hesaplanmıştır. Sonuçlar her bir problem için Tablo 2'de verilmiştir. Algoritmaya ait parametreler ise Tablo 1'de verilmiştir. Parametrelerle ilgili ayrı bir çalışma yapılmamış olup literatürdeki değerler dikkate alınarak ve belli denemeler sonucu sezgisel olarak belirlenmiştir.

Tablo 1: DGA Parametreleri

Problemler	Populasyon büyüklüğü	İterasyon	Çaprazlama oranı (CR)	Değişim katsayısı (F)
tsp5	40	100	0.7	2
tsp10	40	500	0.7	2
tsp20	40	1000	0.7	2
tsp40	40	2000	0.7	2
tsp50	40	2000	0.7	2

Tablo 2: DGA'nın GSP Performansı ve Optimum İle Karşılaştırma.

Problemler	Düğüm sayısı	DGA (Ortalama)	Optimum	Ortalama hata (%)
tsp5	5	2662.47	2662.47	0
tsp10	10	5328.90	5379.21	0.88
tsp20	20	8806.99	7768.54	13.37
tsp40	40	19863.62	14237.16	39.52
tsp50	50	29228.17	17529.85	66.73

Tablo 2'de de görüldüğü gibi DGA algoritması mevcut haliyle küçük boyuttaki problemlerde etkin olmasına karşın düğüm sayısı arttıkça ortalama sapmalar oldukça

büyümüştür. 5 ve 10 düğümünden oluşan tsp5 ve tsp10 problemleri için iyi sonuçlar bulmasına rağmen düğüm sayısı arttıkça kabul edilemez sapmalar olduğu görülmektedir.

6. SONUÇ

DGA'nın orijinal haliyle ve Michalewicz'in dönüştürme fonksiyonuyla algoritma GSP problemleri için bir alternatif olmaktan çok uzaktır. DGA'nın GSP problemlerine alternatif bir çözüm metodu olabilmesi için yapılabilecek değişiklikler ve eklentileri üç ayrı başlık altında toplayabiliriz:

1. DGA algoritmasının yapısında, arama sürecini etkinleştirici birtakım değişiklikler yapmak. Bu konuda Keskintürk'ün (2006) ve Lin ve diğerlerinin (2004) çalışmasına bakılabilir. Her iki çalışmada da yapılan birtakım değişikliklerle arama sürecinin etkinliği artırılmaya çalışılmıştır.
2. Parametrelerin seçiminde bir optimizasyon çalışması yapmak. Sürekli verilerin çözümünde etkin olan parametreler dönüştürme fonksiyonunun kullanıldığı durumlarda etkin olmayabilir. GSP problemlerinde kullanılan DGA'lar için ayrı bir parametre optimizasyon çalışması yapılmalıdır.
3. Dönüştürme operatörü için farklı metotlar denemek. Bu konuda Onwubolu ve Davendra (2006) ve Nearchou (Baskıda)'nın geliştirmiş oldukları dönüştürme operatörleri incelenebilir.

DGA'nın GSP problemleri için bu çalışmadaki haliyle ve Michalewicz'in dönüştürme operatörüyle etkin olmadığı söylenebilir. Ancak bahsedilen değişiklikler deneyerek daha gerçekçi yorumlar yapılabilir.

KAYNAKLAR

- AARTS, E.H.L., KORST, J.H.M., LAARHOVEN, P.J.M., "A Quantitative Analysis of the Simulated Annealing Algorithm: A Case Study for the Traveling Salesman Problem", *J. Stats. Phys*, 50, 189-206, 1988.
- DORIGO, M., GAMBARDELLA, L.M., "Ant Colonies For The Travelling Salesman Problem", *Biosystems*, 43 (2), 73-81, 1997.
- KARABOĞA, D., *Yapay Zeka Optimizasyonu Algoritmaları*, İstanbul, Atlas Yayın Dağıtım, 2004.
- KESKİNTÜRK, T., "Diferansiyel Gelişim Algoritması", **Yöneylem Araştırması ve Endüstri Mühendisliği 26. Ulusal Kongresi**, Kocaeli Üniversitesi, 2006.
- LIN, Y.C., HWANG, K.S., WANG, F.S., "A mixed-coding of evolutionary algorithms to solve mixed-integer nonlinear programming problems", *Computers and Mathematics with Applications*, 47, 1295-1307, 2004.
- 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., **Genetic Algorithms + Data Structure = Evolution Programs**, A.B.D., Springer & Verlag, 1992.
- NEARCHOU, A. C., "Meta-heuristics from nature for the loop layout design problem", *Int. J. Production Economics*, Baskıda.

- ONWUBOLU, G., DAVENDRA, D., "Scheduling flow shops using differential evolution algorithm", **European Journal of Operational Research**, 171, 647-692, 2006.
- POTVIN, J.V., "The Traveling Salesman Problem: A Neural Network Perspective", **INFORMS Journal on Computing**, 5, 328-348, 1993.
- POTVIN, J.V., "Genetic Algorithms for the Traveling Salesman Problem", **Annals of Operations Research**, 63, 339-370, 1996.
- STORN, R., PRICE, K., "Differential evolution: a simple and efficient adaptive scheme for Global optimization over continuous spaces", **Technical Report TR-95-012**, International Computer Science Institute, Berkeley, 1995.
- STORN, R., Çevrimiçi: <http://www.icsi.berkeley.edu/~storn/code.html>, [02.02.2006], 2001.
- TAHA, H.A., Yöneyem Araştırması, Literatür Yayıncılık, İstanbul, 2000.
- TİMOR, Mehpere, Yöneyem Araştırması ve İşletmecilik Uygulamaları, İstanbul Üniversitesi İşletme Fakültesi Yayını, İstanbul, s.23-36, 2001.