Dokuz Eylül Üniversitesi
Sosyal Bilimler Enstitüsü Dergisi
Cilt: 10, Sayı:3, 2008

Çarpımsal Ceza Modeli İle Tamsayılı Programlama

Sabri Erdem[♣]

Özet

Doğrusal olmayan optimizasyon problemlerinin çözüm yöntemlerinden birisi, kısıtların sağlanmama durumlarında amaç fonksiyonunu olumsuz yönde değiştirecek bir ceza uygulanmasıdır. Çarpımsal ceza modeli, son dönem çalışmalarında henüz yer almakla birlikte, literatürde farklı yaklaşımlara sahip ceza teknikleri de yer almaktadır. Tamsayılı programlama problemlerinin çözümünde kullanılan yöntemler arasında kesme düzlemi yaklaşımları, dal sınır yöntemleri ve evrimsel optimizasyon uygulamaları sayılabilir.

Bu çalışma kapsamında ilk kez tamsayılı optimizasyon problemlerinin çözümünde çarpımsal ceza temelli kısıt sağlama yöntemi uygulanmıştır. Yöntem doğrusal olmayan ortak test problemlerinden Himmelblau üzerinde test edilmiş. Yöntemin, problemin kompleksliği ve büyüklüğü karşısındaki davranışı gözlemlenmiştir ve performansı da, diğer yaklaşımlarla karşılaştırmalı biçimde analiz edilmiştir.

Anahtar Sözcükler: *Çarpımsal Ceza, Tamsayılı Programlama, Evrimsel Optimizasyon*

[♣] Dokuz Eylül Üniversitesi, İşletme Fakültesi, sabri.erdem@deu.edu.tr

Abstract

One of the solution methods for the nonlinear optimization problems is to apply penalty that changes the value of objective function as contrary to optimization direction. Multiplicative penalty method, has been already taken place in the recent studies whereas there can be found other approaches in the literature. Cutting plane approaches, branch and bound methods and evolutionary optimization applications can be regarded as essential solutions to integer programming problems.

In scope of this study it is first time that the multiplicative penalty approach is used for integer programming as a new constraint handling method. The method is tested on the Himmelblau's problem which is one of the common test problems in the nonlinear optimization area. Method was also tested against complexity and size of the problem in terms of its performance as compared to existing methods' results.

Keywords: *Multiplicative Penalty, Integer Programming, Evolutionary Optimization*

1 GİRİŞ

Tamsayılı programlama özellikle ikinci dünya savaşı sonrası geliştirilen yöntemler ile son dönem optimizasyon çalışmalarında geniş yer tutmaktadır. Gerçek hayat problemlerinin tamsayılı programlama ile ifade edilmesi özellikle atama, ulaştırma, gezgin satıcı problemleri ve maksimum akış – en kısa yol problemlerine en iyi çözüm arayışlarını gündeme getirmektedir. Bu tarz problemler bilindiği üzere çoğunlukla NP-tam (NP-Complete) çözüm kompleksliğine sahiptirler. Dolayısıyla matematiksel programlama sınıfında bulunan kesme düzlemi, iç nokta (interior point) yaklaşımı, dal-sınır algoritmaları gibi yöntemler bu tarz problemlerinin çözümünde yetersiz kalmaktadır.

NP-tam türü gerçek hayat problemlerine son yıllarda evrimsel optimizasyon sınıfındaki yöntemlerle çözümler getirilmeye çalışılmaktadır. Bunlar arasında sıklıkla kullanılan yöntem genetik algoritmalardır. Genetik algoritmalar özellikle kısıtsız problemler için etkili çözümler sunmaktadır. Bununla birlikte kısıtlı problemlerin genetik algoritma gibi evrimsel hesaplama yöntemleriyle çözümünde farklı yaklaşımlar bulunmaktadır.

Kısıt sağlama yöntemi olarak ceza yaklaşımları, onarım algoritmaları, amaç fonksiyonu ve kısıtların birlikte alındığı ortak-evrimleşme (co-evolutionary) ve çok amaçlı optimizasyon yaklaşımları, melez (hibrid) yöntemler ve diğer gösterim şekilleri ve operatörler sayılabilir.

Evrimsel hesaplama yöntemlerinin sıklıkla kullandığı yöntemlerden birisi ceza fonksiyonu yöntemidir. Ceza yöntemi de farklı şekillerde uygulanmaktadır. Bunlar arasında statik ceza yaklaşımı, dinamik ceza yaklaşımı, ölü ceza, tavlama yaklaşımlı ceza, adaptif ceza ve ortak-evrimleşme (co-evolutionary) ceza yaklaşımı sayılabilir. Genel olarak ceza temelli evrimsel hesaplama yöntemlerinde kullanılan ceza fonksiyonu Denklem 1'deki gibi verilmektedir (Coello, 2001)

$$F(\vec{x}) = f(\vec{x}) \pm \left[\sum_{i=1}^{m} r_i G_i + \sum_{j=1}^{p} c_j L_j \right] \tag{1}$$

Burada $F(\vec{x})$ genişletilmiş amaç fonksiyonu, G_i ve L_j problem kısıtlar $g(\vec{x})$ ve $h(\vec{x})$'e ilişkin fonksiyonlar, r_i ve c_j ceza faktörü olarak nitelendirilen sabitlerdir. G_i ve L_j fonksiyonlarının en yaygın formu Denklem 2 ve 3'teki gibidir.

$$G_i = \max\left[0, g_i(\vec{x})\right]^{\beta} \tag{2}$$

$$L_j = \left| h_j(\vec{x}) \right|^{\gamma} \tag{3}$$

Burada β ve γ genellikle 1 veya 2 olarak seçilirler.

Ceza temelli yöntemler Denklem 1'den görüleceği üzere sağlanamayan kısıtların olduğu durumlar için toplamsal biçimde, ceza fonksiyonu tarafından üretilen ceza değerlerinin amaç fonksiyonuna, fonksiyonun değerini olumsuz yönde değiştirecek biçimde eklenmektedir. Burada sunulan çalışma ise ceza fonksiyonuna yeni bir yaklaşım getirerek çarpımsal olarak ele almaktadır. Yöntem ilk olarak Erdem (2007) tarafından tanıtılmıştır ancak tamsayılı programlamaya ilk defa bu çalışma dahilinde uygulanmaktadır.

2 MATERYAL VE YÖNTEM

1.1 Doğrusal Olmayan Kısıtlı Optimizasyon Problemleri

Çalışmanın bu bölümünde bir önceki optimizasyon problemlerine ilave olarak model karar değişkenlerinin oluşturduğu kısıtların da içinde bulunduğu doğrusal olmayan optimizasyon modeli ele alınmaktadır. Optimizasyon modeli, Model (4)'te tanıtılmaktadır.

$$
\begin{aligned}
&\min f(\vec{x}) \\
&g_i(\vec{x}) \geq b_i \qquad i = 1,2,\dots,p \\
&h_j(\vec{x}) = 0 \qquad j = 1,2,\dots,l \\
&f{:}\, \Re^n \rightarrow \Re \\
&\vec{x} \in E,\ E \subseteq R^n
\end{aligned}
\tag{4}
$$

Burada çözüm vektörü $\vec{x} = (x_1, x_2, \dots, x_n)$, *p:* eşitsizlik içeren kısıt sayısı, *l:* eşitlik içeren kısıt sayısıdır.

1.2 Yöntemin Yapısal Analizi

Burada önerilen kısıt işleme modeli, en küçüklemeye dayalı tek amaç fonksiyonlu problemleri ele almaktadır. Model, diğer modeller gibi, var olan kısıtları ceza temelli olarak amaç fonksiyonuna dahil etmektedir. Ceza temelli modellerde kısıtlar sağlanmadıkları takdirde amaç fonksiyonunu da içine alan uygunluk fonksiyonunu olumsuz yönde değiştirmektedirler. Çalışmadaki çarpımsal model de benzer şekilde bir *H(x)* uygunluk fonksiyonunu olumsuz yönde değiştirmeye çalışmaktadır. Ancak geleneksel modeller gibi uygunluk fonksiyonunu toplamsal olarak değil, bir çarpan olarak etkilemektedir.

Çarpımsal model daha önceden, Gen ve Cheng (1996) tarafından yapılan çalışmada nakledildiği üzere, Yokota v.d. (1995) tarafından Model (5)'te verilen optimizasyon modeli için, Denklem (6-a), (6-b) ve (7)'de görüldüğü biçimde önerilmiştir. Yokota v.d.'nin bu modeli Smith, Tate ve Coit'in yaklaşımı olan (Smith ve Tate, 1993) ceza fonksiyonunun çarpımsal model olarak ifade edilmesidir. Smith v.d. çalıştığı, Model (5)'te verilen doğrusal olmayan optimizasyon modelinde eşitlik kısıtına yer vermemektedirler (Gen ve Cheng, 1996).

$$\begin{aligned} &\max f(\vec{x}) \\ &g_i(\vec{x}) \le b_i \qquad i = 1,2,\dots,p \end{aligned} \tag{5}$$

Denklem (6-a)'daki $fitness(\vec{x})$ uygunluk fonksiyonudur. *P(x)* Denklem (6-b)'deki gibi ifade edilebilir:

$$fitness(\vec{x}) = f(\vec{x}) \times P(\vec{x}) \tag{6-a}$$

$$P(\vec{x}) = 1 - \frac{1}{m}\sum_{i=1}^{m}\left(\frac{\Delta b_i(\vec{x})}{b_i}\right) \tag{6-b}$$

Burada m: toplam kısıt sayısı ve,

$$\Delta b_i(\vec{x}) = \max[0, g_i(\vec{x}) - b_i] \quad (7)$$

Burada $\Delta b_i(\vec{x})$, i kısıtının sağlanmama miktarına karşılık gelmektedir.

Daha önceki benzer çalışmalardan ayrı olarak burada tanıtılan yöntem, bahsedildiği gibi, sadece kısıtların sayıca sağlanma oranına ya da sadece kısıt sağlama oranına bakmaz, kısıtlardaki bu her iki durumu da çarpımsal olarak dikkate alır. Eğer amaç fonksiyonu birleştirilmiş ve en küçüklenmeye çalışılan bir sezgisel model gibi düşünülürse, önerilen kısıt işleme modeli Denklem (8)'deki gibi olacaktır.

$$H(x) = 1.Parça \times 2.Parça \times 3.Parça, \quad (8)$$

Önerilen çarpımsal modelde çarpan terimler aşağıdaki gibi gruplanmıştır:

1. Parça: $f(\vec{x})$ (en küçüklenecek olan amaç fonksiyonu)
2. Parça: Kısıtlardan sapmaların miktarına bağlı fonksiyon
3. Parça: Bütün kısıtların sağlanmasına ilişkin fonksiyon

2. ve 3. Parçalar iki olursuz çözüm üzerinde bir kıyaslamaya da olanak tanımaktadır. 2. Parça çözüm noktalarındaki kısıtların sağlanamamasının göreceli bir ölçüsünü temsil etmektedir. Diğer taraftan 3. Parça kısıtların sayıca sağlanma oranıyla ilgilenmektedir. Bu açıklamalardan yola çıkılırsa, bir tarafta "bir kısıtın sağlanmasında büyük sapmaya sahip ancak diğerlerinin hepsini sağlamış bir çözüm noktası", diğer tarafta ise "bütün kısıtları çok küçük sapmalarla sağlayan çözüm noktası" durmaktadır. Kısıtlar sağlanmadığında, diğer olası bütün durumlar bu iki taraf arasında yer almaktadır.

Çarpımsal olarak elde edilen uygunluk fonksiyonu *H(x)*'in yeni durumdaki yapısı ise Denklem (9)'da görülmektedir.

$$H(\vec{x}) = H\left(f(\vec{x}), r(\vec{x}), t(\vec{x})\right) \tag{9}$$

Burada,

$f(\vec{x})$*:* En küçüklenecek olan amaç fonksiyonu

$r(\vec{x})$*:* kısıtlardaki toplam sapma miktarına bağlı fonksiyon

$t(\vec{x})$: kısıtların sayıca sağlanma oranına bağlı fonksiyon

1. Parça: Amaç fonksiyonunu içeren parça:

$$\left(f(x) + \varepsilon\right)^{W_1} \tag{10}$$

Burada ε sıfıra yakın pozitif bir reel sayıyı temsil etmektedir. Amaç fonksiyonunun değerinin *H(x)* fonksiyonu içerisinde en uygun biçimde yer alabilmesi W_1 değerine bağlıdır. ε ise amaç fonksiyon değerinin sıfır olması durumunda da *H(x)* değerinin hesaplanmasını sağlamaktadır.

Gerçek hayat problemlerinden olan tasarım uygulamalarında, kar enbüyüklemesi ya da maliyet enküçüklemesi temelli problemlerle çalışılacağı için amaç fonksiyonu değeri her zaman pozitif değerler alacaktır. Ancak matematiksel modellerde amaç fonksiyonun negatif değerler aldığı durumda 1. Parça $-\left|f(x) + \varepsilon\right|^{W_1}$ biçiminde alınarak *H(x)* değeri hesaplanmalıdır.

2. Parça: Sağlanmayan her bir kısıt için sapma miktarı:

$$sapma_i = \begin{cases} -g_i(\vec{x}) + b_i & kısıt\ "\leq"\ eşitsizliği\ içeriyorsa \\ -\left|h_i(\vec{x}) - b_i\right| & kısıt\ "="\ içeriyorsa \end{cases} \tag{11}$$

Buna bağlı olarak göreceli sapma miktarı,

$$d_i = \begin{cases} sapma_i / b_i & b_i \neq 0 \\ sapma_i & aksi\ halde \end{cases} \tag{12}$$

Göreceli toplam sapma miktarı fonksiyonu,

$$r(\vec{x}) = 1 / Exp(W_2 \sum_i d_i) \tag{13}$$

Kısıtların sağlanma durumunda sapma sıfır olarak alınmalıdır. Diğer bir deyişle pozitif farklar sapma olarak kabul edilmemelidir. Çünkü buradaki amaç çözüm noktalarının yaratabileceği kısıt ihlallerini dikkate almaktır. Kısıt ihlali yaratmayan farklar sapma olarak kabul edilmemelidir. Dolayısıyla Denklem (11) ve (12)'deki ifadelere göre sağlanmayan bir kısıtın sapma değeri negatif ve bunların toplamından oluşan d değeri de yine negatif değerler alacaktır. 2. Parça $e^{-W_2 \sum d_i}$ olduğuna göre pozitif bir W_2 için, 2. Parça bütün kısıtların sağlanması durumunda 1 değerini alacaktır. Kısıtların sağlanmama durumlarında ise 1'den büyük değerler alarak *H(x)* değerini büyütmeye çalışacaktır.

3. Parça: Sağlanan kısıtların sayıca oranı fonksiyonu:

$$t(\vec{x}) = u^{W_3} \tag{14}$$

$$u = \begin{cases} (m / m') & f(\vec{x}) > 0 \\ 1/(m/m') & aksi\, halde \end{cases} \tag{15}$$

m : toplam kısıt sayısı

m' : sağlanan kısıt sayısı

Bütün kısıtların sağlandığı durumda 3. Parça 1 değerini alacaktır. En az bir tane kısıtın sağlanmadığı durumda ise 3. Parça 1'den daha büyük bir değer alacaktır (W_3>0). Yani *f(x)*'in küçültülmeye çalışıldığı bir durumda, eğer en az bir kısıt sağlanmıyorsa 2. ve 3. Parça değeri 1'den büyük olacaktır. Dolayısıyla bütün kısıtların sağlandığı iki çözüm vektörü kıyaslanırken, 2. Parça ve 3. Parçanın değeri 1 olacağından, karar kriteri sadece *f(x)* fonksiyon değerleri olacaktır. Aksi durumlarda karar kriteri her zaman *H(x)* fonksiyon değeri olacaktır. Aslında, buradaki 3 parçanın birbirine göre ağırlıklandırılmasında, kıyaslamalar ve ikame ilişkileri de dikkate alınmalıdır.

$$H(\vec{x}) = \left(f(x) + \varepsilon\right)^{W_1} e^{-W_2 \sum_i d_i} u^{W_3} \tag{16}$$

Önerilen yöntemin birleştirilmiş ve bütünleşik yapısı ise Denklem (16)'da görülmektedir. Burada $W_1, W_2, W_3 \in \Re^+$ bahsedildiği üzere her bir bileşenin birbirine göre göreceli ağırlığını göstermektedir ve bu değerlerin değiştirilmesi seçilen iki farklı çözüm vektörü arasında bir kıyaslama ve ikame olanağı vermektedir.

2.1 Örnek Deneysel Problem

Burada verilen problem ilk olarak Himmelblau (1972) tarafından ortaya atılmıştır ve aynı isimle anılmaktadır. Problem diğer genetik algoritma benzeri ceza yöntemi kullanan yaklaşımlar için bir kıyaslama problemi haline gelmiştir (Gen ve Cheng, 1997). Problem 5 karar değişkeni (x1, x2, x3, x4 ve x5), doğrusal olmayan kısıt fonksiyonları ve sınır değerlerini içermektedir. Bu çalışmada ilave olarak bütün karar değişkenleri tamsayılı olarak ele alınmıştır.

Enküçükle:

$$f(x) = 5.3578547 x_3^2 + 0.8356891 x_1 x_5 + 37.293239 x_1 - 40{,}792.141 \quad (17)$$

$$g_1(x) = 85.334407 + 0.0056858 x_2 x_5 + 0.00026 x_1 x_4 - 0.0022053 x_3 x_5 \quad (18)$$

$$g_2(x) = 80.51249 + 0.0071317 x_2 x_5 + 0.0029955 x_1 x_2 + 0.0021813 x_3^2 \quad (19)$$

$$\mathrm{g}_3(\mathrm{x}) = 9.300961 + 0.0047026 x_3 x_5 + 0.0012547 x_1 x_3 + 0.0019085 x_3 x_4 \quad (20)$$

$$0 \le g_1(x) \le 92 \quad (21)$$

$$90 \le g_2(x) \le 110 \quad (22)$$

$$20 \le g_3(x) \le 25 \quad (23)$$

$$78 \le x_1 \le 102 \quad (24)$$

$$33 \le x_2 \le 45 \quad (25)$$

$$27 \le x_3 \le 45 \quad (26)$$

$$27 \le x_4 \le 45 \quad (27)$$

$$27 \le x_5 \le 45 \quad (28)$$

ve x_1, x_2, x_3, x_4 ve x_5 tamsayı

Tablo 1 ve 2, Himmelblau probleminin tamsayılı olmayan halindeki mevcut ceza yöntemleriyle çözülmüş sonuçlarını istatistiksel olarak ortaya koymaktadır.

Tablo 1 Ceza temelli bazı kısıt sağlama yöntemlerinin Himmelblau problemi performansları

Yöntem	**En İyi**	**Ortalama**	**En Kötü**	**Std.**
MGA	-31005.797	-30862.874	-30721.042	73.24
Gen	-30183.576	N/A	N/A	N/A
Static Penalty	-30790.272	-30446.462	-29834.385	226.342
(Himmelblau, 1972)	-30373.949	N/A	N/A	N/A
Coello	-31020.859	-30984.241	-30792.408	73.6335
Dynamic	-30903.877	-30539.916	-30106.25	200.035
Annealing	-30829.201	-30442.126	-29773.085	244.619
Adaptive	-30903.877	-30448.007	-29926.154	249.485
Death Penalty	-30790.271	-30429.371	-29834.385	234.555

Kaynak: Coello (2001)

3 BULGULAR

Çözüm için başlangıç parametreleri olarak W_1=0.2, W_2=0.4 ve W_3=0.4 alınarak öncelikle kısıtların sağlanmasına ağırlık verilmiştir. Başlangıç popülasyonu olarak 30, 60 ve 100 bireyli üç ayrı standart genetik algoritma çözümü 100'er nesil için denenmiştir. Her popülasyon için bulunan değerler istatistiksel olara Tablo 2'de verilmiştir. Her nesil ortalaması da ayrıca Şekil 1'de, nesil sayısına karşılık amaç fonksiyon hesaplama sayısındaki değişim Şekil 2'de verilmiştir.

Buradaki tablolarla karşılaştırılmak üzere çarpımsal ceza yaklaşımı ile tamsayılı olmayan çözüm Tablo2'de ve tamsayı kısıtının da bulunduğu durumdaki çözüm Tablo 3'de verilmektedir.

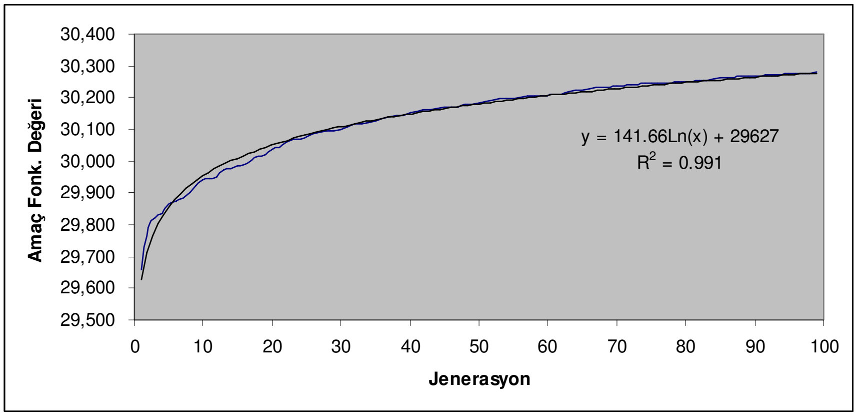
Tablo 2. Çarpımsal ceza yönteminin Himmelblau problemi üzerindeki performansı

Başlangıç Çözüm Noktası	20	50	100	200
Ortalama	-30992.21	-31023.90	-31025.36	-31025.59
Standart Sapma	5.114532	1.18736	0.230375	0.012063
En Kötü	-30895.80	-30969.10	-31014.08	-31025.13
En İyi	-31025.63	-31025.63	-31025.63	-31025.63

Kaynak: Erdem (2007)

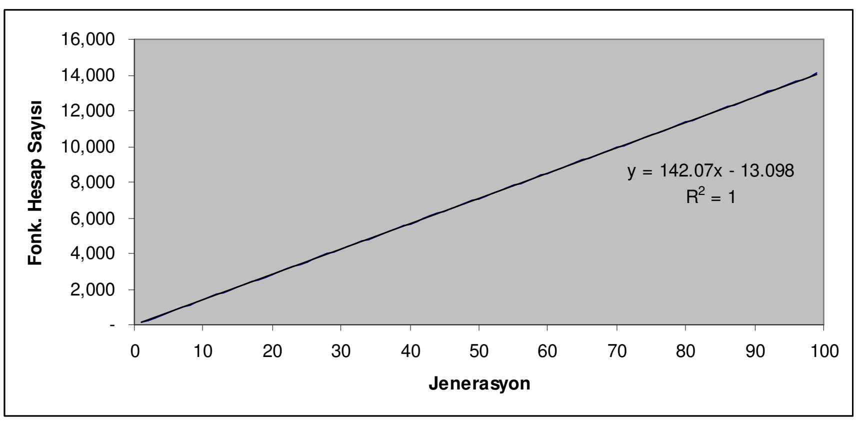
Tablo 3. Çarpımsal ceza yönteminin Himmelblau tamsayılı problemi üzerindeki performansı

Başlangıç Çözüm Noktası	30	60	100
Ortalama	-30355.54	-30323.22	-30294.78
Standart Hata	19.57	23.65	20.50
En Kötü	-29990.77	-29682.37	-29848.73
En İyi	-30699.24	-30633.22	-30592.43



Şekil 1. Nesillerde bulunan ortalama amaç fonksiyon değeri

Amaç fonksiyonu hesaplama sayısı, geliştirilen yöntemlerin performans karşılaştırmalarında verilen ölçütlerden birisidir ve literatürde de arzu edilen değerler 100 000 ve daha altındaki değerler olarak görülebilir. Yöntemin performansı Şekil 2'deki gibidir.



Şekil 2. Nesillerde elde edilen ortalama amaç fonksiyonu hesaplama sayısı

4 SONUÇLAR

Sonuçlara bakıldığında ortalama değerlerin Coello (2001)'de verilen yöntemlerden, MGA ve Coello tarafından kullanılan yöntemlerin dışındakilerden daha iyi olduğu söylenebilir. Buradaki çarpımsal yöntemde ilave olarak, bir de tamsayı kısıtı bulunduğu dikkate alınırsa yöntemin diğer problemler üzerinde de kullanılabileceği söylenebilir.

Şekil 1 dikkatle incelendiğinde amaç fonksiyonu değeri nesiller ilerledikçe logaritmik olarak iyileşmektedir. Buna karşılık nesillerle amaç fonksiyon hesaplama sayısı arasında tam bir doğrusal ilişki bulunduğu da Şekil 2'de görülmektedir. Nesil sayısının 100'e yaklaştığı durumda amaç fonksiyon hesaplama sayısı 30 binler seviyesine gelmektedir ve bu da yöntemin performansı açısından tatmin edicidir.

Önerilen yöntem, özellikle kesikli fonksiyonlara sahip, türevli çözümlerin uygulanamadığı ve klasik çözüm yöntemlerinin etkisiz kalabildiği durumlarda işletme ve iktisadi alanlardaki doğrusal olmayan kısıtlı problemlerin çözümünde rahatlıkla uygulanabilecek bir yapıdadır. Çözüm için önemli olan ceza fonksiyonunu ve uygun çalışma parametrelerini seçebilmektir.

Bundan sonraki çalışmalarda W_1, W_2 ve W_3'ün farklı değerlerle çalıştırılması ve bunun problem çözümüne etkisi araştırılabilir. Aynı zamanda yöntem atama, ulaştırma ve diğer grup tamsayılı problemler üzerinde denenip sonuçlar analiz edilerek yöntemin bir sonraki ilerleme yolu araştırılabilir.

Kaynaklar

Coello Carlos A. (2001). “Theoretical and Numerical Constraint Handling Techniques used With Evolutionary Algorithms: A Survey of State of the Art”, *Civil Engineering and Environmental Systems*

Coello, Carlos. A. ve Montes, E. M. (2002), “Constraint-handling in genetic algorithms through the use of dominance-based tournament selection”, *Advanced Engineering Informatics* 16, pp:193-203.

Erdem Sabri, (2007). *Evolutionary Algorithms for the Nonlinear Optimization*, Yayınlanmamış Doktora Tezi, Dokuz Eylül Üniv. Fen Bilimleri Enstitüsü

Gen Mitsuo ve Cheng Runwei. (1996). “A survey of Penalty Techniques in Genetic Algorithms”. In ToshioFukuda and Takeshi Furuhashi, editors, *Proceedings of the 1996 International Conference on Evolutionary Computation*, pp 804-809, Nagoya, Japan

Gen Mitsuo ve Cheng Runwei. (1997). *Genetic Algorithms & Engineering Design*, John Wiley and Sons, Inc, New York

Himmelblau D. M. (1972*). Applied Nonlinear Programming* McGraw-Hill

Smith A.E. ve Tate D.M. (1993). “Genetic Optimization Using a Penalty Function”. In Stephanies Forrest editor, *Proceedings of the 5. International Conference on Genetic Algorithms*, 499-503, San Mateo California

Yokota, T., Gen, M., Ida, K. ve Taguchi, T. (1995). “Optimal Design of System Reliability by an Improved Genetic Algorithm”. *Transactions of Institute of Electronics, Information and Computer Engineering*, J78-A(6):702-709, (Çalışma aslen Japonca olup Gen ve Cheng (1996)’dan nakledilmiştir).