

Kromozom Seçim Metriğinin Yinelemeli İki-Aşamalı Evrimsel Programlamada Performansa Katkısı

Turgut Önal^{1,*} ve Ulaş Beldek²

¹Çankaya Üniversitesi, Elektronik ve Haberleşme Mühendisliği Bölümü, 06810, Yenimahalle, Ankara, Türkiye

²Çankaya Üniversitesi, Mekatronik Mühendisliği Bölümü, 06810, Yenimahalle, Ankara, Türkiye

*Corresponding author: onalturgut@gmail.com

Özet. Bu çalışmada yeni bir evrimsel eniyileştirme algoritması olan yinelemeli iki-aşamalı evrimsel programlama algoritmasında kromozom seçmek için kullanılan metriğin farklılaştırılmasının eniyileştirme problemlerinin çözümlemedeki katkısı incelenmiştir. Gerçekleştirilen metrik değişikliği, önceden uygulanan metriğe göre test edilen eniyileştirme problemlerinde karşılaştırılabilecek sonuçlar sunmaktadır. Elde edilen sonuçlar bu çalışmada gösterilmektedir.

Anahtar Kelimeler. Genetik algoritma, evrimsel programlama, mutasyon, meta sezgisel metotlar.

Abstract. In this study, the contribution of changing the metric designated for selecting the chromosomes in solution of optimization problems in a new evolutionary optimization algorithm which is called as the Recursive Two-Stage Evolutionary Programming is investigated. The performed metric change puts forward comparable results in the tested optimization problems with respect to the metric applied before. The obtained results are presented in this study.

Keywords. Genetic algorithm, evolutionary programming, mutation, meta-heuristic methods.

1. Giriş

Meta sezgisel eniyileme (optimization) metotları herhangi bir eniyileştirme problemini türevsel bilgiler kullanmadan olasılıksal bazı yöntemlerle tasarlanmış bir kalite

Received August 17, 2012; accepted April 5, 2013.

Bu makale, 26-27 Nisan 2012 tarihlerinde Çankaya Üniversitesi'nin Ankara Merkez yerleşkesinde yapılmış olan 5. Çankaya Üniversitesi Mühendislik ve Teknoloji Sempozyumu'nda sunulan ve sadece geniş bildiri özeti bölümü hakem sürecinden geçerek bu sempozyum kitapçığında yayımlanan bir makalenin revize edilmiş şekli olup Sempozyum Değerlendirme Komitesi tarafından yayımlanmak üzere Çankaya University Journal of Science and Engineering dergisine gönderilmesi önerilmiş ve derginin bağımsız hakem değerlendirmeleri sonucunda yayıma kabul edilmiştir.

kriterine göre aday çözümleri geliştirerek çözmeye çalışan algoritmaların toplandığı kümeye verilen genel isimdir. Evrimsel algoritmalar (evolutionary algorithms) [1-3] ve tavlama benzetimi (simulated annealing) [4] bu algoritmalarından ve metotlardan en çok kullanılanlarına örnek olarak verilebilirler.

Evrimsel algoritmaların çeşitlerinden olan genetik algoritma [1-2] ve evrimsel programlama [3] bu algoritmaların en tanınmışlarından ve bu algoritmaların değişik varyasyonları yaygın olarak farklı çeşitteki eniyileme problemlerinde sıklıkla kullanılmaktadırlar. Bu algoritmalar eniyileme problemlerine birden fazla aday ilkel çözümden (kromozom) oluşan bir küme (nüfus, popülasyon) sunar. Bu nüfusun genetik operasyonlara benzeyen işlemlerle yapıları değiştirilir ve belli bir olasılıksal altyapı içinde ve ilerlemeci bir mantık dahilinde seçilmeleri ve dolayısı ile kromozomların değişimleri sağlanarak daha iyi çözümlerin elde edilmesine olanak verilir ve yeni nüfuslar elde edilir. Kromozomların uyumu (fitness) genelde eniyileme problemi çözmek amaçlı bir hedef fonksiyonunun azaltılması (minimization) ile belirlenir ve o kromozomun problemi çözmeye yatkınlığı hakkında bir fikir verir. Evrimsel algoritmalar, belli bir nesil boyunca koşturulduktan sonra ya da o ana kadar elde ettiğimiz en uyumlu kromozomun bizim açımızdan o problemi çözmekte yeterli derecede etkin olduğunu varsaydığımızda sonlandırılır ve de elde edilen sonuçlar (en uyumlu kromozom, algoritmanın kaç nesilde sonlandığı gibi bulgular) koşturduğumuz meta sezgisel algoritmanın sonucu olarak ilan edilir.

Bir nesildeki kromozomları değiştirmek ve daha iyi çözümler elde etmek amacıyla gerçekleştirilen genetik operasyonlara benzeyen işlemleri, bu işlemler çok farklı biçimlerde gerçekleştiriliyor olsalar bile, dört ana kulvarda toplayabiliriz. Bu işlemler çaprazlama (crossover), mutasyon (mutation), seçme veya doğal ayıklama (selection, natural selection) ve yeniden üreme (reproduction) işlemleridir.

Hangi genetik işlemlerden faydalanılıyor olursak olalım bu genetik işlemler uygulamalarına göre eniyilemesi gerçekleştirilecek olan hedef problemin yerel araştırma uzayını iki biçimde araştırmaya yardımcı olur. Bu araştırma biçimlerinden ilki keşfetme (exploration) işlevi diğeri de sömürme (exploitation) işlevi olarak adlandırılır. Keşfetme genel olarak problemle ilgili araştırma uzayının çok farklı ve göreceli olarak az incelenmiş bölgelerinin keşfedilmesini ve incelenmesini sağlarken sömürme eniyilemesi gerçekleştirilen problemin minimumlarının bulunma ihtimalinin olduğu

daha küçük yerel (local) bölgelerde özelliklerinin belirlenmesine yarar. Bir algoritmanın keşfetme ve sömürme gücü genetik operasyonların algoritma içindeki oranlarıyla ya da biçimleriyle düzenlenebilir. Eniyileme problemlerinin verimli bir biçimde çözümlenebilmeleri için bu iki temel özelliğin yeterli oranlarda algoritmanın içinde pay edilmesi çok önemli bir husustur. Genetik algoritmalarda keşfetme ve sömürme işlevleri çaprazlama ve mutasyon işlemleri ile sağlanırken evrimsel programlamada aynı işlevler sadece mutasyonlar vasıtasıyla gerçekleştirilir. Evrimsel programlamada mutasyon işlemi gerçekleştirilirken keşfetme ve sömürme karakterlerinin algoritmaya kazandırılması için farklı türde mutasyon işlemleri [5-7] de alternatif olarak kullanılabilir. Bu mutasyonların ortak amacı kromozom çeşitliliğini arttırarak farklı yeni çözümler (yavrular) oluşturup bu vasıta ile geniş çözüm uzaylarına algoritmayı yaymak ve bu çözüm uzaylarının farklı bölgelerini incelemektir.

Meta sezgisel metotların geliştirilme aşamasında en fazla gözlemlenen uygulamalardan biri bu uygulamaları melezleştirme [8], uygulama ile ilgili parametrelerin ve prosedürlerin değiştirilmesi ya da bu metotlara yerel sömürme kapasitelerini arttırmak amacıyla yerel araştırma yapan birimlerin eklenmesidir.

Bu çalışmada ele aldığımız meta sezgisel araştırma metodu evrimsel programlamanın uygun bir biçimde değiştirilmesiyle elde edilen yinelemeli iki aşamalı evrimsel programlamadır [9] (YİEP) (recursive two-stage evolutionary programming RTEP). Henüz yeni bir algoritma olarak ortaya çıkan bu meta sezgisel metot iki aşama kullanarak global keşfetme ile yerel sömürme işlevlerini orantılı olarak algoritma içinde uygulayan bir yapıya sahiptir. Algoritma global keşfetme ve yerel sömürme işlevlerinin başarıya ulaşması için mutasyon çeşitlerinin ve doğal seçim stratejilerinin olanaklarını kullanır. Özetleyecek olursak YİEP algoritmasının en belirgin özellikleri global keşfetme ve yerel sömürme işlevlerinin tekrarlanan bir yapıda gerçekleştirilmesi (algoritma yerel bir optimalin içine sıkıştığı durumlarda ya da umut verici geniş bir araştırma bölgesi bulunduğu), bu işlevler için sadece belirlenmiş bir mutasyon operasyonu kullanması ve mutasyon operasyonunun seçilen bireyler arasındaki uzaklığa dayanmasıdır.

Biz bu çalışmamızda kullanılan mutasyon işleminin gerçekleştirileceği bireyleri seçerken aralarındaki uzaklık tanımının değiştirilmesinin YİEP [9] algoritmasının çeşitli eniyileme problemlerindeki performansına katkısını inceledik. Algoritma mutasyon işlemini gerçekleştirirken, bireyleri belirlemede uzaklığa dayalı bir ölçüm kullandığından dolayı, uzaklığı belirleyen prosedürün değişmesinin algoritma üstündeki

etkisi çok belirgindir ve bu çalışmamızı gerçekleştirmemizde esas motive edici unsur olmuştur. Makalemizin geri kalanı orijinal YİEP'in anlatılması [9] (Bölüm 2), YİEP'in belli kısımlarının yeniden ele alınarak tarafımızdan değiştirilmiş halinin (YİEP₁) farklı eniyileme problemlerine uygulanması ve elde edilen karşılaştırmalı sonuçların sunulması (Bölüm 3) ve yorumlar ile ileride yapacağımız çalışmalara bu çalışmamızın katkısı (Bölüm 4) şeklinde organize edilmiştir.

2. Materyal ve Metot

YİEP algoritması orijinal olarak aşağıdaki adımlardan oluşmaktadır [9].

Adım 1. İçinde μ tane kromozom olan bir ilk nüfus (initial population) oluşturalım ve bu nüfus içindeki her bir kromozom x_i , $i = 1, 2, \dots, \mu$, n tane reel değerli bileşeni (n gene sahip kromozom) olan bir vektörle ifade ediliyor olsun: Bu durumda kromozom $x_i = \{x_i(1), x_i(2), \dots, x_i(n)\}$ şeklinde ifade edilecektir. $x_i(1), x_i(2), \dots, x_i(n)$ bileşenleri ise kromozomunun genlerini oluşturacaktır.

Adım 2. Yinelenmeden (bir işlevden diğerine geçmeden) önce ardışık kaç nesil boyunca keşfetme ve sömürme işlevlerinin uygulanacağını gösteren K_1 ve K_2 parametrelerini belirleyelim.

Adım 3. Her kromozomun uyum değerini eniyileme problemini tanımlayan bir amaç fonksiyonuna göre belirleyelim. Eğer nüfustaki en iyi uyum değeri kabul edilebilir bir değerse algoritmayı durduralım ve sonuçları açıklayalım. Yoksa algoritmayı devam ettirelim.

Adım 4. Adım 5 ile Adım 8 arasında anlatılan kısımları K_1 nesil boyunca uygulayalım (bu kısımlar tek bir nesil için gerçekleştirilen keşfetme işlevini belirtmektedir).

Adım 5. Nüfustaki her kromozom için: n -boyutlu araştırma uzayında ele alınan kromozoma *Öklid* uzaklığına göre en uzak M kromozomu belirleyelim. Belirlediğimiz M kromozom x_i 'nin 'yabancıları' olarak adlandırılacaktır. x_i için belirlediğimiz M yabancı kromozomun oluşturduğu kümeden bir tanesini x_i ile mutasyon işlemine tabi tutmak üzere rastgele seçelim, bu kromozom $y_i = \{y_i(1), y_i(2), \dots, y_i(n)\}$ olsun.

Adım 6. Nüfustaki her kromozom için: mutasyon operasyonu gerçekleştirilelim ve mutasyon sonucu μ tane yeni yavru kromozom oluşturalım. Mutasyon operasyonunu tek bir ebeveyn kromozom için şu şekilde tanımlayabiliriz: her kromozomun n tane geni bulunmaktadır. $1 \leq k \leq n$ olmak üzere rastgele bir (tam sayı olan) k değeri

belirleyelim; k değeri, mutasyona uğrayacak gen sayısını vermektedir. Kromozomdaki n genden yine k tanesini rastgele seçelim. Seçtiğimiz genler r_t ile gösterilsin. Dolayısı ile r_t , $t = 1, 2, \dots, k$ kromozomun mutasyona uğratılacak genleridir. Bu durumda yeni oluşacak yavru kromozom x'_i ile gösterilsin ve bu yeni yavru kromozomu genleriyle birlikte $x'_i = \{x'_i(1), x'_i(2), \dots, x'_i(n)\}$ şeklinde ifade edilsin. Bu durumda yavru kromozomun mutasyon sonucu oluşan genleri,

$$t = 1 : k, \quad x'_i(r_t) = x_i(r_t) + \sigma_i(r_t) \times N_t(0, 1), \quad (1)$$

şeklinde ifade edilecektir. (1)'de verilen ifadede $x'_i(r_t)$ yavru kromozomun r_t geninin değerini, $x_i(r_t)$ ebeveyn kromozomun r_t geninin değerini, $N_t(0, 1)$ ortalaması 0'a ve standart sapması 1'e eşit olan normal (Gaussian) dağılımına göre her t değeri için yeniden belirlenmiş rastgele sayıyı ve $\sigma_i(r_t)$ ise ebeveyn kromozomun r_t genini mutasyona uğratmak için kullanılan standart sapma değeridir. $\sigma_i(r_t)$ algoritmada x_i 'nin ve y_i 'nin r_t geni boyunca oluşan *Öklid* uzaklığı olarak belirlenmiştir. $\sigma_i(r_t)$ tek bir gen boyunca oluşan *Öklid* uzaklığı olarak tanımlandığı için tek bir bileşeni vardır ve de *Öklid* uzaklığı *mutlak* uzaklığa dönüşmektedir. Dolayısı ile,

$$\sigma_i(r_t) = \sqrt{(x_i(r_t) - y_i(r_t))^2} = |x_i(r_t) - y_i(r_t)| \quad (2)$$

şeklinde ifade edilebilir.

Adım 7. Yeni elde edilen yavru kromozomların x'_i , $i = 1, 2, \dots, \mu$, uyum değerlerini (fitness) hesaplayalım. Her ebeveyn kromozom x_i ve ona karşılık gelen yavru kromozom x'_i için: eğer ebeveyn kromozomun uyum değeri yavrusu kromozomun uyum değerinden büyükse ebeveyn kromozomu bir sonraki nesle aktaralım aksi takdirde yavru kromozomu gelecek nesle aktaralım.

Adım 8. Eğer neslin en iyi uyuma sahip kromozomunun uyumu yeterli derecede yüksekse, ya da maksimum nesil sayısına ulaşıldıysa algoritma sonlandırılır, yoksa algoritma devam eder.

Adım 9. Adım 10-13 arasında anlatılan kısımları K_2 nesil boyunca uygulayalım (bu kısımlar tek bir nesil için gerçekleştirilen sömürme işlevini belirtmektedir).

Adım 10. Nüfustaki her kromozom için: n -boyutlu araştırma uzayında ele alınan kromozoma *Öklid* uzaklığına göre en yakın M kromozomu belirleyelim. Bu belirlediğimiz M kromozom x_i 'nin 'komşuları' olarak adlandırılacaktır. x_i için belirlediğimiz M komşu kromozomun oluşturduğu kümeden bir tanesini x_i ile mutasyon işlemine tabi tutmak üzere rastgele seçelim, bu kromozom $y_i = \{y_i(1), y_i(2), \dots, y_i(n)\}$ olsun.

Adım 11. Nüfustaki her kromozom için: Adım 6'da anlatıldığı biçimde ve (1) kullanılarak μ tane yeni yavru kromozom oluşturalım. Adım 10'da y_i, x_i 'ye en yakın kromozomlar arasından seçilmiştir. Dolayısı ile $\sigma_i(r_t)$ algoritmada bu birbirlerine yakın iki kromozomun r_t genleri boyunca oluşan *Öklid* uzaklığıdır. Uzaklık Adım 6'daki gibi tek bir gen boyunca tanımlandığı için tek bir bileşeni vardır ve de *Öklid* uzaklığı *mutlak* uzaklığa (2)'de belirtildiği gibi dönüşmektedir.

Adım 12. Yeni elde edilen yavru kromozomların $x'_i, i = 1, 2, \dots, \mu$, için uyum değerlerini (fitness) hesaplayalım. Her ebeveyn kromozom x_i ve ona karşılık gelen yavru kromozom x'_i için: eğer ebeveyn kromozomun uyum değeri yavru kromozomun uyum değerinden büyük veya eşitse ebeveyn kromozomu bir sonraki nesle aktaralım aksi takdirde yavru kromozomu gelecek nesle aktaralım.

Adım 13. Eğer neslin en iyi uyuma sahip kromozomunun uyumu yeterli derecede yüksekse, ya da maksimum nesil sayısına ulaşıldıysa algoritma sonlandırılır, yoksa Adım 4'e dönülerek bir sonraki keşfetme ve sömürme işlevleri gerçekleştirilir.

Görüldüğü üzere YİEP [9] üç farklı parçanın (keşfetme işlevi, sömürme işlevi ve de bu işlevlerin yinelenen bir yapıda uygulanması) algoritmada kullanılmasıyla oluşturulmuş bir yapıya sahiptir.

YİEP'i incelediğimizde algoritmanın en kritik kısımlarından ikisinin Adım 5 ve Adım 10'da bahsedilen yabancılar ve komşular kümesine dahil kromozomların belirlenmesi olduğu görülmektedir. Bu prosedürler n -boyutlu araştırma uzayında *Öklid* uzaklığına göre tanımlanmıştır. Buna karşın bu prosedürler çok daha farklı şekilde de oluşturulabilir. Yaptığımız bu çalışmada bu uzaklığın n -boyutlu araştırma uzayında *Öklid* uzaklığına göre değil *mutlak* uzaklığa göre tanımlanmasının etkisini sınamak istedik. Bu değişikliği gerçekleştirmekteki temel motivasyonumuz Adım 6 ve Adım 11'de gözümüze çarpan bir ayrıntıdan kaynaklanmaktadır. Adım 6 ve Adım 11'de gerçekleştirilen mutasyon işleminde mutasyon işlemi için tanımlanan $\sigma_i(r_t)$ değeri x_i ve onun için seçilen y_i 'nin r_t geni boyunca oluşan *Öklid* uzaklığı olarak belirlenmiştir. Buna karşın tek bir gen boyunca tanımlanan bu parametre doğal olarak bize r_t geni boyunca oluşan *mutlak* uzaklığı vermektedir (2). Bu sebepten dolayı Adım 5 ve Adım 10'da yabancıları ve komşuları belirlemek üzere *mutlak* uzaklığın kullanılması planlanmıştır. Bu değişiklikler sonucu oluşan Adım 5 ve Adım 10 aşağıdaki biçimlerde yenilenmiştir:

Adım 5'. Nüfustaki her kromozom için: n -boyutlu araştırma uzayında ele alınan kromozoma *mutlak* uzaklığa göre en uzak M kromozomu belirleyelim. Bu belirlediğimiz M kromozom x_i 'nin 'yabancıları' olarak adlandırılacaktır. x_i için belirlediğimiz M yabancı kromozomun oluşturduğu kümeden bir tanesini x_i ile mutasyon işlemine tabi tutmak üzere rastgele seçelim, bu kromozom $y_i = \{y_i(1), y_i(2), \dots, y_i(n)\}$ olsun.

Adım 10'. Nüfustaki her kromozom için: n -boyutlu araştırma uzayında ele alınan kromozoma *mutlak* uzaklığa göre en yakın M kromozomu belirleyelim. Bu belirlediğimiz M kromozom x_i 'nin 'komşuları' olarak adlandırılacaktır. x_i için belirlediğimiz M komşu kromozomun oluşturduğu kümeden bir tanesini x_i ile mutasyon işlemine tabi tutmak üzere rastgele seçelim, bu kromozom $y_i = \{y_i(1), y_i(2), \dots, y_i(n)\}$ olsun.

Bu değişiklikler sonucu oluşan yeni algoritmaya YİEP₁ adını verdik. Bu aşamadan sonra YİEP ve YİEP₁ algoritmalarını karşılaştırmak üzere belli test fonksiyonları belirledik. Bu test fonksiyonları [9]'da kullanılan fonksiyonlardan seçilmiş ve türetilmiştir:

$$f_1(z) = \sum_{i=1}^D z_i^2, \quad (3)$$

$$f_2(z) = \sum_{i=1}^D [z_i^2 - 10 \cos(2\pi z_i) + 10], \quad (4)$$

$$f_3(z) = 4z_1^2 - 2.1z_1^4 + \frac{1}{3}z_1^6 + z_1z_2 - 4z_2^4 + 1.03. \quad (5)$$

Tablo 1'de bu test fonksiyonlarının özellikleri sunulmuştur.

TABLO 1. Test fonksiyonlarının özellikleri.

Test fonksiyonu $f(z) = f(z_1, z_2, \dots, z_D)$	Boyut (D)	Değişken değer aralığı	Minimum m değeri
$f_1(z)$	30	$[-100, 100]^{30}$	$f_{1,\min} = 0$
$f_2(z)$	30	$[-5.12, 5.12]^{30}$	$f_{2,\min} = 0$
$f_3(z)$	2	$[-5, 5]^2$	$f_{3,\min} = 0$

Bu fonksiyonlardan $f_1(z)$ tek bir tane minimum değeri bulunan $f_2(z)$ birçok yerel (local) minimum değeri olmakla birlikte sadece bir global minimumu bulunan ve $f_3(z)$ az sayıda minimumu bulunmakla birlikte boyutu küçük olan fonksiyonlardır.

3. Bulgular

YİEP ve YİEP₁'i karşılaştırmak amacıyla algoritma ile ilgili ilk uygulamada parametreler şu şekilde belirlenmiştir (bu parametreler [9]'da gerçekleştirilen benzetimlerle bağıntılı olarak seçilmiştir):

$K_1 = 1$ (ardışık olarak gerçekleştirilen keşfetme işlevi sayısı).

$K_2 = 1$ (ardışık olarak gerçekleştirilen sömürme işlevi sayısı).

$M = 10$ (mutasyon uygularken seçilen komşu ve yabancı kromozomların sayısı).

$n = D$ (problemdeki toplam değişken sayısı ya da kromozomlardaki toplam gen sayısı Tablo 1'de gösterilen test fonksiyonlarının boyutuna eşit olmak zorundadır).

$\mu = 50$ (nüfusun boyutu, her nesilde nüfus içinde 50 kromozom yer almaktadır).

Algoritma boyunca gerçekleştirilen uyum hesaplama işlemi: $f_1(z)$ ve $f_2(z)$ için 150000'dir (toplam 3000 nesle karşılık gelmektedir) ve $f_3(z)$ için bu değer 10000'dir (toplam 200 nesle karşılık gelmektedir).

Benzetimlerde kullanılan test fonksiyonu $f_k(z)$, $k = 1, 2, 3$ için herhangi bir x kromozomunun uyum değerini bulmak üzere belirlenen amaç fonksiyonu (maliyet fonksiyonu)

$$\text{Maliyet}(x) = (f_k(x) - f_{k,\min}), \quad (6)$$

ve kromozomun uyum değeri

$$\text{Uyum}(x) = \frac{1}{\text{Maliyet}(x)} \quad (7)$$

olarak belirlenmiştir.

Bu parametrelere ve prosedürlere göre her biri birbirinden bağımsız olarak koşutlanan 20 farklı eniyileştirme benzetiminin sonuçlarının ortalaması alınarak YİEP ve YİEP₁'in performansları (3), (4) ve (5)'de gösterilen test fonksiyonları için karşılaştırılmıştır. Karşılaştırma algoritmalar sonucu son nesilde elde edilen en iyi uyum değerlerinin ortalamasına ve son nesildeki ortalama nesil uyum değerlerinin ortalamasına bakılarak gerçekleştirilmiştir. Elde edilen sonuçlar Tablo 2 ve Tablo 3'te sunulmuştur.

Gerçekleştirilen ikinci uygulamada ise, K_1 ve K_2 değerlerini değiştirmenin algoritmaya katkısı sınanmak istenmiştir. Diğer parametreler ve prosedürler ilk uygulamadakiyle aynı tutulmak şartıyla yalnız $K_1 = 4$ ve $K_2 = 8$ alınarak YİEP ve YİEP₁ algoritmalarının performansları test fonksiyonlarında karşılaştırılmıştır. Elde edilen sonuçlar Tablo 4 ve Tablo 5'te sunulmuştur.

TABLO 2. Benzetim sonuçları: son nesildeki en iyi durum değerlerinin 20 bağımsız benzetimde ortalaması ($K_1 = 1$, $K_2 = 1$).

Algoritmalar	$f_1(z)$	$f_2(z)$	$f_3(z)$
YİEP	373.1376	0.0171	4.6842×10^8
YİEP ₁	404.8232	0.0177	9.0953×10^7

TABLO 3. Benzetim sonuçları: son nesildeki uyum değerlerinin ortalamasının 20 bağımsız benzetimde ortalaması ($K_1 = 1$, $K_2 = 1$).

Algoritmalar	$f_1(z)$	$f_2(z)$	$f_3(z)$
YİEP	155.9273	0.0110	6.1864×10^4
YİEP ₁	173.7142	0.0110	7.6992×10^4

TABLO 4. Benzetim sonuçları: son nesildeki en iyi durum değerlerinin 20 bağımsız benzetimde ortalaması ($K_1 = 4$, $K_2 = 8$).

Algoritmalar	$f_1(z)$	$f_2(z)$	$f_3(z)$
YİEP	2291.7	0.0195	2.2566×10^9
YİEP ₁	3059.4	0.0184	2.1673×10^{12}

TABLO 5. Benzetim sonuçları: son nesildeki uyum değerlerinin ortalamasının 20 bağımsız benzetimde ortalaması ($K_1 = 4$, $K_2 = 8$).

Algoritmalar	$f_1(z)$	$f_2(z)$	$f_3(z)$
YİEP	912.2135	0.0115	1.2471×10^5
YİEP ₁	1.3367×10^3	0.0115	1.2454×10^5

Tablo 2 ve Tablo 3'ten anlaşılacağı gibi ilk uygulamada YİEP₁ çok boyutlu ama tek bir minimuma sahip $f_1(z)$ için YİEP'e göre daha iyi sonuçlar vermektedir. Çok boyutlu ve çok fazla yerel minimuma sahip olan $f_2(z)$ fonksiyonu için YİEP ve YİEP₁ aralarında çok ufak farklar olsa da birbirine benzer sonuçlar sunmaktadır. Küçük boyutlu ve az miktarda yerel minimum barındıran $f_3(z)$ için ise YİEP optimal sonuca daha fazla yaklaşmış buna karşın YİEP₁ son nesildeki kromozomların ortalama uyum değerini daha çok arttırmıştır.

Tablo 4 ve Tablo 5 karşılaştırıldığında ikinci uygulamada da YİEP₁ çok boyutlu ama tek bir minimuma sahip $f_1(z)$ için YİEP'e göre daha iyi sonuçlar vermektedir. Çok boyutlu ve çok fazla yerel minimuma sahip olan $f_2(z)$ fonksiyonu için YİEP YİEP₁'e göre son nesildeki en iyi uyum değerlerinin ortalaması bakımından biraz daha iyi sonuç verse bile son nesildeki ortalama uyum değerlerinin ortalaması aynı çıkmıştır. Buna karşın küçük boyutlu ve az miktarda yerel minimum barındıran

$f_3(z)$ için ise $YİEP_1$ optimal sonuca daha fazla yaklaşmış buna karşın $YİEP$ son nesildeki kromozomların ortalama uyum değerini daha çok arttırmıştır.

4. Tartışma ve Sonuç

Bu çalışmada yeni tasarlanmış bir Evrimsel algoritma olan $YİEP$ [9], genel yapısı bozulmadan bazı yöntemsel değişimlerle ($YİEP_1$) farklı test fonksiyonlarının eniyilemesinde sınanmıştır. Elde edilen sonuçlar algoritmanın ilerde daha etkin bir şekilde nasıl geliştirileceğine dair ipuçları sunmaktadır. Bu çalışma kromozom seçimi ve kromozomlardaki genlerin mutasyon işlevi için seçimi gibi raslantısal prosedürlerin de bulunduğu test uygulamaları içeren bir tür deneysel meta-sezgisel algoritma çalışmadır. İçinde rastlantısal prosedürlerin olması dolayısıyla işlevlerin etkinlikleri deneysel çıkarımlar vasıtasıyla daha rahat gösterilebilir. Kromozom seçiminde kullanılan uzaklık metriğinin ilk uygulamada $f_1(z)$ ve $f_2(z)$ fonksiyonları için az olmakla birlikte daha iyi bir sonuç sunmasının *Öklid* ve *mutlak* metriklerin özelliklerine dayandığını düşünmekteyiz. *Öklid* uzaklığına göre sabit bir noktaya eşit veya yakın mesafede olan farklı birkaç noktanın, *mutlak* uzaklığa göre sıralamasını yapalım. Bu durumda, sabit noktaya göre daha dik bir bileşene (bileşenlere) sahip olan nokta sabit noktaya daha yakındır. $YİEP_1$ algoritmasında sömürme işlevi yürütülürken, bu biçimde yerleşmiş kromozomlardan birisi ana kromozomla (sabit nokta) mutasyon işlemine tabi tutulmak üzere seçilirse bu kromozom *mutlak* uzaklığa göre daha yakın olan (daha dik bileşenli) kromozom olacaktır. Sömürme işlevinde, ana kromozom ve seçilen kromozomun rastgele olarak dik bileşenli genlerinin mutasyon için seçilmesi dik bileşenli genlerin arasındaki mesafenin (2)'den dolayı göreceli olarak büyük olmasına sebep verecektir ve bu da algoritmanın yapısı gereği daha geniş bir araştırma uzayı parçasının taranmasına olanak verecektir. Bu özellik boyutu görece olarak daha büyük olan fonksiyonlarda ($f_1(z)$ ve $f_2(z)$ gibi) araştırma uzayının sömürme işleminde bile bir tür keşfetme özelliği katılması anlamına gelecektir. Özellikle sömürme ve keşfetme işlevleri birbirinin ardı sıra birer nesil boyunca sürdürülüyorsa ($K_1 = 1$ ve $K_2 = 1$) bu ilk nesillerde keşfetme işlevinin $YİEP_1$ 'de daha yoğun yapılmasını sağlar. Çok boyutlu fonksiyonlarda eniyileme işleminin yakınsaması düşük boyutlu fonksiyonlara göre daha uzun süreceğinden *mutlak* uzaklığın kullanılması algoritmanın ilk nesillerinde az da olsa bir avantaja neden olabilir. Ama ilerleyen nesillerde algoritma artık minimum değerlere yakınsamaya başladığında hem *mutlak* hem de *Öklid* uzaklığının ortalama yakın sonuçlar vereceğini unutmamak lazımdır. İlk uygulamada $f_1(z)$ ve $f_2(z)$ fonksiyonlarında elde ettiğimiz az

miktarda $YIEP_1$ lehine olan birbirlerine yakın sonuçlar bu duruma örnek olarak verilebilir.

İkinci uygulamada K_1 ve K_2 değerinin arttırılmasının en önemli sonucu test fonksiyonunda daha da etkinleşen ekstra sömürme ve keşfetme kapasitesidir. Bu sayede üç test fonksiyonunda da daha iyi sonuçlar elde edilmiştir. [9]'da belirtildiği gibi ilk uygulamaya göre daha yüksek K_1 ve K_2 değerlerinin kullanılması birçok test fonksiyonunda daha başarılı sonuçlar elde edilmesine yardımcı olacaktır. Sömürme kapasitesi minimum değerlere daha yaklaşıldıkça daha önemli bir husus olarak ortaya çıkar. $YIEP_1$ bu açıdan $f_1(z)$ gibi tek bir minimumu olan fonksiyonda daha başarılı olmuştur. İlk nesillerde keşfetme kapasitesiyle tek minimuma sahip fonksiyonun minimum noktasının olduğu bölge genel olarak belirlendikten sonra yoğun sömürme ($K_2 = 8$) işlevi sayesinde bu noktanın bulunduğu çukurun derinlemesine incelenmesi mümkün olmuştur. Buna karşın $f_2(z)$ gibi çok fazla minimum içeren ve çok boyutu olan bir fonksiyonda $YIEP$ az bir farkla öne geçmektedir. K_1 değerinin arttırılması sonucu oluşan ekstra keşfetme kapasitesi çok boyutlu ve çok fazla yerel minimum içeren $f_2(z)$ fonksiyonunda $YIEP$ 'in çok az farkla daha etkin olmasını sağlamıştır. Artan ardışık miktardaki sömürme işlevi *mutlak* metriğin özelliğinden dolayı $f_3(z)$ gibi az sayıda minimum içeren ve boyutu küçük olan fonksiyonlarda sömürme işlevini daha uygun hale getirmiştir ve $YIEP_1$ $f_3(z)$ fonksiyonunda biraz daha ön plana çıkmıştır.

Yapılan uygulamalar sonucunda, farklı fonksiyon tiplerinde ve farklı sömürme ve keşfetme kapasitesine sahip $YIEP$ algoritmalarında *Öklid* ya da *mutlak* uzaklığın kullanılması göreceli olarak ufak kazanımlar sağlanmasına katkıda bulunmuşlardır.

Adım 6'da ve Adım 11'de gerçekleştirilen mutasyon işleminin çoklu noktada gerçekleştirilen bir mutasyon işlemidir. Çoklu mutasyon algoritmanın keşfetme kapasitesini arttırırken sömürme kapasitesini daha çok azaltan bir husustur. Tek minimum noktası olan bir fonksiyonda daha çok sömürme işleviyle minimum noktasına olabildiğince yaklaşılması önemli bir husustur. Bundan sonraki çalışmalarda çoklu mutasyon yerine tek noktada mutasyonun çözüme etkisi incelenecektir.

Kaynaklar

- [1] J. H. Holland, *Adaptation in Natural and Artificial Systems*, The University of Michigan Press, MI: Ann Arbor 1975.
- [2] J. H. Holland, Genetic Algorithms, *Scientific American* (July 1992), 66–72.

- [3] D. B. Fogel, *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*, IEEE Press, New York 1995.
- [4] K. K. Lai and J.W.M. Chan, Developing a simulated annealing algorithm for the cutting stock problem, *Computers and Industrial Engineering* **32** (1997), 115–127.
- [5] M. Iwamatsu, Generalized evolutionary programming with Levy-type mutation, *Computer Physics Communications* **147** (2002), 729–732.
- [6] Q. Wu, Cauchy mutation for decision-making variable of Gaussian particle swarm optimization applied to parameters selection of SVM, *Expert Systems with Applications* **38** (2011), 4929–4934.
- [7] Q. Wu, Power load forecasting based on hybrid PSO with Gaussian and adaptive mutation and Ws-SVM, *Expert Systems with Applications* **37** (2010), 194–201.
- [8] R. Banos, C. Gil, J. Reca and F. G. Montoya, A memetic algorithm applied to the design of water distribution networks, *Applied Soft Computing* **10** (2010), 261–266.
- [9] M. S. Alam, M. M. Islam, X. Yao and K. Murase, Recurring two-stage evolutionary programming: a novel approach for numerical optimization, *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* **41** (2011), 1352–1365.