

AKÜ FEMÜBİD 17 (2017) 035102 (922-937)

AKU J. Sci. Eng. 17 (2017) 035102 (922-937)

DOI : 10.5578/fmbd.66295

Nümerik Fonksiyonların Optimizasyonu için Karşıt Tabanlı Yeni Bir Meta-Sezgisel Algoritma

Hasan TEMURTAŞ¹, Celal YAŞAR², Serdar ÖZYÖN^{3,*}¹Dumlupınar Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Kütahya.^{2,3}Dumlupınar Üniversitesi, Mühendislik Fakültesi, Elektrik Elektronik Mühendisliği Bölümü, Kütahya.e-posta: ¹hasan.temurtas@dpu.edu.tr, ²celal.yasar@dpu.edu.tr, ^{3,*}serdar.ozyon@dpu.edu.tr (*İletişim Yazarı)

Geliş Tarihi: 05.06.2017 ; Kabul Tarihi: 19.12.2017

Özet

Anahtar kelimeler

Optimizasyon, Meta-sezgisel algoritmalar, Yerçekimsel arama algoritması, Zıt konumlu öğrenme, Test fonksiyonları.

Bu çalışmada literatürde meta-sezgisel algoritmaların performanslarını artırmaya yönelik yaklaşımlardan biri olan zıt konumlu öğrenme kavramı (OBL), yerçekimsel arama algoritmasına (GSA) iki farklı şekilde uygulanmıştır. Birinci yaklaşım da (ObGSA-1), ilk popülasyonun oluşturulmasında ajanların yarısı rastgele atanırken, diğer yarısı bu ajanların simetrisine konumlandırılmıştır. İkinci yaklaşımda (ObGSA-2) ise ilk popülasyonda, rastgele olarak oluşturulan bütün ajanların zıt konumları belirlenmiş ve uygunluk değeri daha yüksek olan ajanlarla ilk popülasyon oluşturulmuştur. Bu yaklaşımlarla performans ve kararlılık açısından algoritma iyileştirilmiştir. Ortaya çıkan bu yeni algoritmaya zıt konumlu yerçekimsel arama algoritması (Opposite Based Gravitational Search Algorithm-ObGSA) adı verilmiştir. Performans analizi için ObGSA üç farklı yapıdaki test fonksiyonlarına uygulanmıştır. Bu sonuçlara geliştirilen her iki yaklaşımda (ObGSA-1, ObGSA-2), GSA'ya göre daha iyi sonuçlar vermiştir. İki yaklaşım kendi aralarında değerlendirildiğinde ise ObSA-2 yaklaşımının, ObGSA-1 yaklaşımına göre daha iyi değerler yakaladığı ve daha kararlı bir yapı olduğu sonucuna varılmıştır.

A Novel Opposite-Based Meta-Heuristic Algorithm for Numerical Function Optimization

Abstract

Keywords

Optimization, Meta-heuristic algorithms, Gravitational search algorithm, Opposition-based learning, Benchmark functions.

In this study, Opposite Based Learning concept (OBL) which is one of the approaches to increase the performance of meta-heuristic algorithms, has been applied to Gravitational Search Algorithm (GSA). This new algorithm that came out has been called Opposite Based Gravitational Search Algorithm (ObGSA). In the study OBL has been applied to GSA in two different ways and these were called as ObGSA-1 and 2 respectively. In ObGSA-1 while in the first population formation of GSA half of the agent have been assigned randomly, the other half has been located according to the symmetry of these agents. Whereas in ObGSA-2 in the first population the opposite locations of all the agents that were formed randomly have been defined and the first population has been formed with the agents whose compliance value were higher. ObGSA-1 and 2 have been applied to three test functions with different structures successfully for stability and performance analysis. Compared with GSA, ObGSA-1 and 2 have caught better results in shorter time. When the approaches have been evaluated among themselves, the result that has been reached is that ObSA-2 has a better and more stable structure than ObSA-1.

1. Giriş

Büyük arama uzaylarına sahip çok boyutlu problemlerin sayısal yöntemlerle çözümlerinin literatürde uzun süreler aldığı görülmüştür. Bu nedenle günümüzde karmaşık ve sayısal yöntemlerle çözümü zor ya da imkânsız olan problemlerin çözümünde sıklıkla meta-sezgisel algoritmalar kullanılmaktadır (Cura, 2008).

Son yıllarda karmaşık mühendislik problemlerinin çözümü için birçok algoritma geliştirilmiştir. Bu algoritmalarından bazıları Yerçekimsel arama algoritması (Gravitational Search Algorithm-GSA) (Rashedi vd. 2009, Rashedi vd. 2010), genetik algoritma (GA) (Goldberg, 1989), diferansiyel gelişim (DE) (Storn ve Price, 1997), parçacık sürü optimizasyonu (PSO) (Kennedy ve Eberhart, 1997), armoni arama (HS) (Geem vd. 2001), yapay arı kolonisi (ABC) (Karaboğa ve Baştürk, 2007), yüklü sistem arama (CSS) (Kaveh ve Talahatari, 2010), su dalgası optimizasyonu (WWO) (Zheng, 2015), balina optimizasyonu (WOA) (Mirjalili ve Lewis, 2016), güvercin optimizasyonu (CSA) (Rajabioun, 2011), güve-alevi optimizasyonu (MFO) (Mirjalili, 2015), karga arama (CSA) (Askarzadeh, 2016), sinüs kosinüs (SCA) (Mirjalili, 2016), optik tabanlı optimizasyon (OIO) (Kashan, 2015), çoklu dize optimizasyonu (MVO) (Mirjalili vd. 2016), radyal hareket optimizasyonu (RMO) (Rahmani ve Yusof, 2014), bozkurt optimizasyonu (GWO) (Mirjalili vd. 2014), simbiyotik organizmalar arama (SOS) (Cheng ve Prayogo, 2014) ve girdap arama algoritmaları (VS) (Doğan ve Ölmez, 2015) şeklinde belirtilebilir.

Genel olarak meta-sezgisel algoritmalar çözüm kümesini rastgele oluşturulan birey kümesi ile aramaya başlarlar. Bu nedenle ilk oluşturulan popülasyondaki bireylerin arama uzayındaki yerleşimleri en iyi sonucu elde edebilmek için oldukça önemlidir. İlk popülasyondaki bireylerin yerel minimumlar yakınında konumlanması durumunda çözümün en iyi sonuca yakınsaması sağlanamamaktadır. Bu durum aşılması gereken bir eksikliktir (Cura, 2008). Bu nedenle literatürde yer

alan meta-sezgisel algoritmalar değişik yöntemlerle bu eksikliği gidermek için iyileştirilmek veya geliştirilmek durumundadır. Meta-sezgisel algoritmaların performanslarının daha da iyileştirilebilmesi için literatürde farklı farklı yöntemler kullanılmaktadır. Bu yöntemlerden birisi de zıt konumlu öğrenme yapısıdır (Tizhoosh vd. 2005).

Literatürde birçok meta-sezgisel algoritmaya zıt konumlu öğrenme Opposite Based Learning-OBL yapısı eklenerek geliştirilmeye çalışılmıştır. Bunlardan bazıları, zıt konumlu diferansiyel gelişim (Rahnamayan ve Tizhoosh, 2008), zıt konumlu öğrenme tabanlı parçacık sürü optimizasyonu (Omran, 2009), zıt konumlu armoni arama (Singh ve diğerleri, 2013) ve zıt konumlu biyocoğrafya tabanlı optimizasyon algoritması (Ergezer vd. 2009) şeklindeki çalışmalardır.

Bu çalışmada, literatürde yerçekimsel arama algoritması (GSA) adıyla anılan meta-sezgisel algoritma geliştirilmek üzere seçilmiştir. Algoritma 2009 yılında Rashedi ve arkadaşları tarafından Newton'un yerçekimi ve hareket kanunlarından esinlenilerek geliştirilmiştir. Literatürde GSA ve geliştirilen versiyonlarının farklı mühendislik problemine başarıyla uygulandığı çalışmalara rastlanmıştır. Bunlardan bazıları, filtre modellemesi (Rashedi vd. 2011), veri madenciliği (Zahiri, 2012), hidrolik türbin yönetme sisteminin parametrelerinin tanımlanması (Li ve Zhou, 2011), optimal ekonomik güç dağıtımı (Swain vd. 2012) ve türbin ısı oranının tahmini problemleri (Zhang vd. 2013) şeklinde belirtilebilir.

Bu çalışmada GSA'ya zıt konumlu öğrenme kavramı iki farklı şekilde uygulanmıştır. Bunlardan birisi, ilk popülasyondaki bireylerin yarısı rastgele oluşturulurken, bireylerin diğer yarısı ise bunların her birisi için zıt konumunda yer alırlar. Çözüm kümesi bu bireylerden oluşturulan ilk popülasyonla aramaya başlanır. Diğer yöntem ise çözüm kümesi için bir birey oluşturulur ve hemen

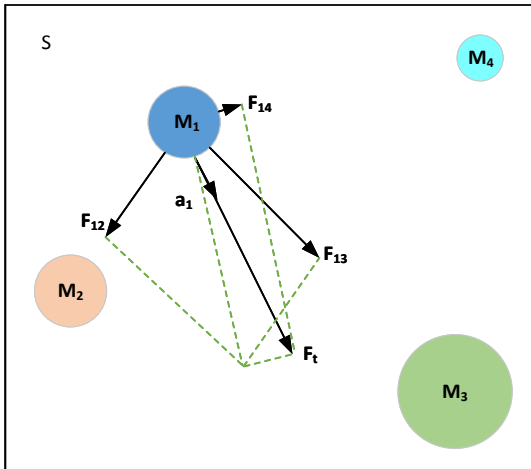
onun zıt konumunda başka bir birey daha oluşturulur. Bu iki birey karşılaştırılarak uygunluğu daha iyi olan birey çözüm kümesindeki popülasyona katılır, diğeri ise atılır. İşlem bu şekilde bireylerin sayısı tamamlanincaya kadar devam eder. Bu sayede daha hızlı bir öğrenme amaçlanmaktadır.

Geliştirilen yeni algoritmalar, üç farklı yapıdaki test fonksiyonlarına uygulanmış ve elde edilen sonuçları değerlendirilmiştir.

2. Yerçekimsel Arama Algoritması (GSA)

Fizikte, kütlesi olan nesnelere birbirlerine doğru hızlanma eğilimi göstermektedirler. Newton'un evrensel çekim kanununa göre, her bir noktasal kütle diğer noktasal kütleyle, ikisini birleştiren bir çizgi doğrultusundaki bir kuvvet ile çeker. Bu kuvvet bu iki kütlelerin çarpımıyla doğru orantılı, aralarındaki mesafenin karesi ile ters orantılıdır. Bu çekimin etkisiyle küçük olan kütle, büyük olan kütleyle doğru ivmelenir (Rashedi vd. 2009, Rashedi vd. 2010).

Algoritmanın temelinde, kütle olarak adlandırılan ajanlar dizini Newton'un hareket ve yerçekimi kanunlarının simülasyonu ile optimum çözümü bulmak üzere konumlandırılır. S arama uzayında konumlandırılan ajanlar (M) ve bir ajana (M_1) etkileyen kuvvetlerin gösterimi Şekil 1'de verilmiştir.



Şekil 1. S arama uzayında konumlandırılan bir ajana etkileyen kuvvetler

Şekilde yer alan her bir kütlelerin bulunduğu konum değerleri, mevcut problemin birer çözüm adayıdır. Kütlelerin büyüklüğü çözüme ne kadar yakın olduklarının göstergesidir.

GSA, i . kümenin konumunun denklem (1)'de verildiği gibi tanımlandığı, N ajanlı bir yapıyla aramaya başlar. Bu ajanlar GSA'da ilk popülasyon için S arama uzayında rastgele konumlandırılırlar.

$$X_i = (x_i^1, \dots, x_i^d, \dots, x_i^n), \quad i = 1, 2, \dots, N \quad (1)$$

Burada x_i^d , d . boyuttaki i . kümenin konumudur, n ise arama alanının boyutudur.

Her bir ajanın kütlesi, uygunluğu (çözüme yakınlığı) ile temsil edilir ve popülasyondaki diğer bireylerin uygunluğuna bağlı olarak denklem (2) ve (3)'e göre hesaplanır.

$$q_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)} \quad (2)$$

$$M_i(t) = \frac{q_i(t)}{\sum_{j=1}^s q_j(t)} \quad (3)$$

Bu denklemlerde $M_i(t)$ ve $fit_i(t)$ sırasıyla kütle ve t anındaki i . ajanın uygunluk değerini göstermektedir.

Bir minimizasyon problemi için $best(t)$ ve $worst(t)$ denklem (4) ve (5)'e göre tanımlanır.

$$best(t) = \min_{j \in \{1, \dots, s\}} fit_j(t) \quad (4)$$

$$worst(t) = \max_{j \in \{1, \dots, s\}} fit_j(t) \quad (5)$$

Problem maksimizasyon problemi ise $best(t)$ ve $worst(t)$ yer değiştirilerek aşağıdaki gibi tanımlanır.

$$best(t) = \max_{j \in \{1, \dots, s\}} fit_j(t) \quad (6)$$

$$worst(t) = \min_{j \in \{1, \dots, s\}} fit_j(t) \quad (7)$$

Algoritmada yer alan herhangi bir ajanın ivmesinin hesaplanması için, ilk olarak diğer ajanlar tarafından o ajanın üzerine uygulanan toplam 3 kuvvet hesaplanır. Bir ajana etkiyen toplam kuvvet denklem (8)'de verilmiştir. Daha sonra hareket kanunları kullanılarak denklem (9)'dan o ajanın ivmesi hesaplanır. Hesaplanan ivme değeri, ajanın mevcut hızına eklenerek yeni hız vektörü elde edilir. Son olarak, bir sonraki popülasyon için ajanın yer alacağı konum, denklem (11)'e göre belirlenir.

$$F_i^d(t) = \sum_{j \in k_{best}, j \neq i} rand_j G(t) \frac{M_j(t) M_i(t)}{R_{ij}(t) + \varepsilon} (x_j^d(t) - x_i^d(t)) \quad (8)$$

$$a_i^d(t) = \frac{F_i^d(t)}{M_i(t)} = \sum_{j \in k_{best}, j \neq i} rand_j G(t) \frac{M_j(t)}{R_{ij}(t) + \varepsilon} (x_j^d(t) - x_i^d(t)) \quad (9)$$

$$v_i^d(t+1) = rand_i \times v_i^d(t) + a_i^d(t) \quad (10)$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \quad (11)$$

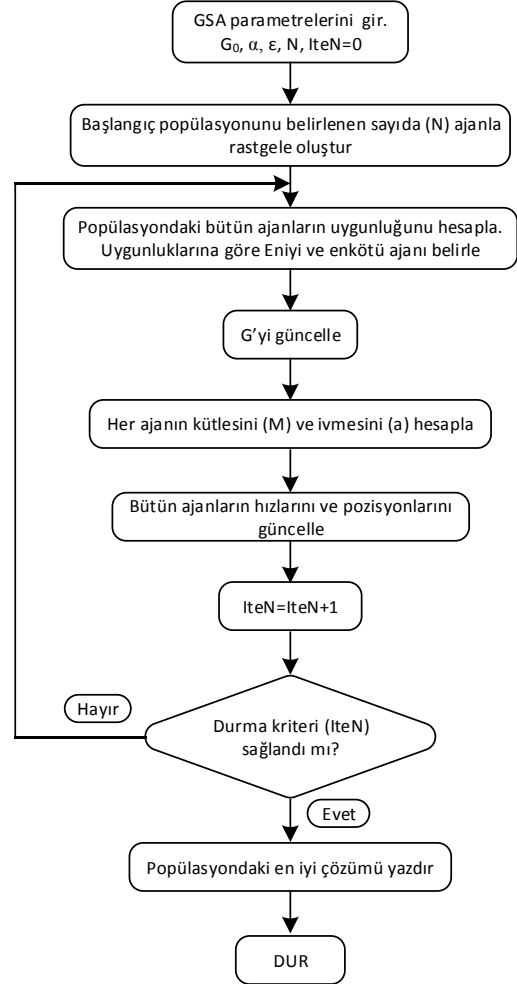
Burada $rand_i$ ve $rand_j$ [0,1] aralığında rastgele iki sayıdır. ε ise matematiksel olarak tanımsızlığı ortadan kaldırmak için kullanılan küçük bir değerdir. $R_{ij}(t)$, i ve j ajanları arasındaki öklid mesafesidir ve $R_{ij}(t) = \|X_i(t), X_j(t)\|_2$ olarak tanımlanır. k_{best} , en iyi uygunluk değerli dolayısıyla en büyük kütleyle sahip K ajanlarının bir dizisidir. Bu dizi başlangıçta K_0 'da başlatılan ve zamanla azaltılan bir fonksiyondur. Burada K_0 , toplam ajan sayısı (N) kümesidir ve doğrusal olarak 1'e düşürülür. Bunun anlamı başlangıçta bütün ajanlar birbirlerine kuvvet uygularken, zaman geçtikçe kuvvet uygulayan ajan sayısı azalacak ve sonunda sistemde diğer kütlelere kuvvet uygulayan tek bir ajan kalacaktır.

Kuvvet ve ivme denklemlerinde yer alan yerçekimsel sabit değeri $G(t)$, başlangıç değeri G_0 , α sabit katsayı, t iterasyon sayısı, T ise bitiş iterasyon sayısını göstermek üzere denklem (12)'de verilmiştir.

$$G(t) = G_0 e^{-\alpha(t/T)} \quad (12)$$

G_0 ve α 'nın algoritmanın yakınsama hızı ve en iyi çözümü bulması üzerine etkileri büyüktür.

Dolayısıyla her problemin özelliğine göre bu değerlerin yeniden belirlenmesi gerekmektedir. Yukarıda çözüm aşamaları anlatılan GSA algoritmasının akış diyagramı Şekil 2'de verilmiştir.



Şekil 2. GSA akış diyagramı

3. Zıt Konumluluk Kavramı (OBL-Opposition-Based Learning)

Meta-sezgisel algoritmalar, en iyi çözüme ulaşmak üzere hesaplamaya bir başlangıç popülasyonu ile başlarlar. İlk popülasyon tanımlanırken, ajanlar çözüm uzayında rastgele konumlarda oluşturulurlar. Fakat rastgele oluşturulan konumlar yerine, uygunluk değeri daha iyi olan bireylere sahip bir ilk popülasyon ile hesaplamaya başlamak, popülasyonun en iyi çözüme ulaşırken geçirdiği gelişim işlemini hızlandırabilir. Bu nedenle, 2005 yılında Tizhoosh tarafından zıt konumlu öğrenme kavramı ortaya atılmıştır. Tizhoosh

tarafından tanımı yapılan bu yaklaşım şu şekildedir, herhangi rastgele bir sayının zıt konumlu durumu çözüme rastgele sayıdan büyük ihtimalle daha yakındır (Tizhoosh vd. 2005). Bundan dolayı, bir sayının zıt konumlu değeri ile birlikte oluşturulan başlangıç popülasyonunun, en iyi çözüme yakınsamak için daha küçük bir arama uzayına ihtiyaç duyacağı söylenebilir. Bu işlem yakınsamayı hızlandırabilir. Zıt konumlu öğrenme kavramı; zıt konumlu sayı, nokta ve optimizasyon olarak aşağıda alt bölümler halinde açıklanmıştır.

3.1. Zıt konumlu sayı

x sayısı $[a, b]$ aralığında tanımlı gerçek bir sayı ise bu sayının zıttı (\check{x}), zıt konum teoremine uygun olarak denklem (13)'e göre tanımlanır (Tizhoosh vd. 2005).

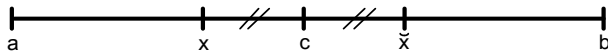
$$\check{x} = a + b - x \quad (13)$$

3.2. Zıt konumlu nokta

Denklem (13)'de yer alan ifade çok boyutlu diziler için genelleştirilebilir. Bunun için d-boyutlu uzayda bir $P = (x_1, x_2, \dots, x_d)$ noktası tanımlansın. Burada $x_1, x_2, \dots, x_d \in R$ ve $x_i \in [a_i, b_i] \forall i \in \{1, 2, \dots, d\}$ olsun. Bu noktanın zıt konumunun $\check{P} = (\check{x}_1, \check{x}_2, \dots, \check{x}_d)$ bileşenleri aşağıdaki denklem ile tanımlanır.

$$\check{x}_i = a_i + b_i - x_i \quad (14)$$

Tek boyutlu uzay için $[a, b]$ aralığında tanımlı bir x noktası ve bu noktanın zıt konumu \check{x} aşağıdaki şekilde gösterilmiştir.



Şekil 3. Tek boyutlu uzay için bir nokta ve bu noktanın zıt konumunun gösterimi

Şekil 3'te $[a, b]$ düzlem sınırlarını, c ise düzlem merkezini temsil eder. Tanımlanan nokta ve bu

noktanın zıt konumu düzlem merkezine eşit uzaklıktadır (Tizhoosh vd. 2005).

3.3. Zıt konumlu optimizasyon

Bir optimizasyon probleminin çözümünde $P = (x_1, x_2, \dots, x_d)$ gibi d-boyutlu arama uzayında tanımlanmış bir nokta, popülasyonda yer alan her bir aday çözüm için benzetilebilir. Zıt konumlu nokta tanımına göre bu noktanın zıt konumu $\check{P} = (\check{x}_1, \check{x}_2, \dots, \check{x}_d)$ şeklinde gösterilir. Bu durumda yeni tanımlanan zıt konumlu nokta mevcut problemin çözüm adaylarından biridir. O zaman amaç fonksiyonuna göre değerlendirildiğinde her iki aday çözümün uygunluk fonksiyonları sırasıyla $f(P)$ ve $f(\check{P})$ olacaktır. Daha iyi çözüm için eğer $f(\check{P}) > f(P)$ ise, uygunluk değeri daha iyi olan \check{P} bireyi P bireyinin yerini alır (Tizhoosh vd. 2005).

4. Yerçekimsel arama algoritmasının geliştirilmesi

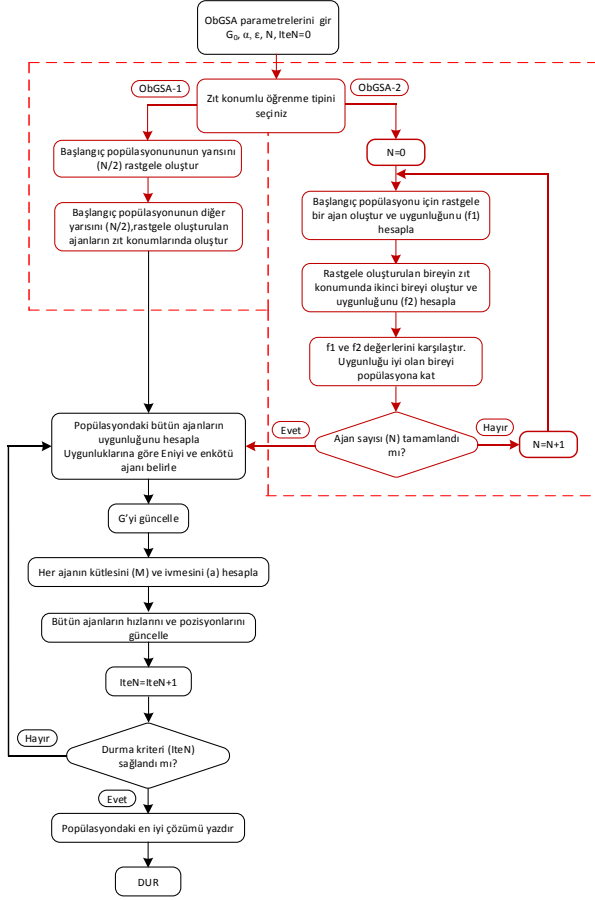
Bu çalışmada GSA'nın performansının iyileştirilmesi için iki farklı zıt konumlu öğrenme kavramı türetilmiştir. Bu iki farklı zıt konumlu öğrenme kavramı GSA'ya ayrı ayrı entegre edilerek, zıt konumlu yer çekimsel arama algoritması (Opposite Based Gravitational Search Algorithm-ObGSA) denilen iki farklı algoritma elde edilmiştir. Bunlara ObGSA-1 ve 2 algoritmaları adı verilmiş ve iki farklı durum alt başlığında incelenmiştir.

Çalışmada önerilen ObGSA-1 ve 2 algoritmaları için akış diyagramı Şekil 4'te verilmiştir. Verilen diyagramda GSA algoritmasına göre farklılıklar kırmızı renkte gösterilmiştir.

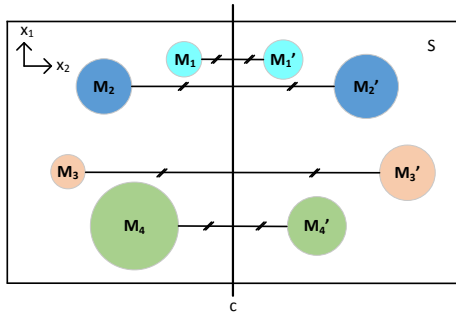
4.1. Durum 1

Birinci yaklaşım (ObGSA-1), GSA'nın ilk popülasyonunun oluşturulmasında ajanların yarısı rastgele atanırken, diğer yarısının bu ajanların simetrisine konumlandırılması temeline dayanır. Şekil 5'te gösterildiği gibi ajanın rastgele oluşturulan ilk konumu ve zıt konumunun kütlesi farklı olabilir.

Bunun nedeni arama uzayında farklı noktalara yerleştirilen ajanların uygunluğunun farklı olmasıdır.



Şekil 4. ObGSA-1 ve ObGSA-2 için akış diyagramı

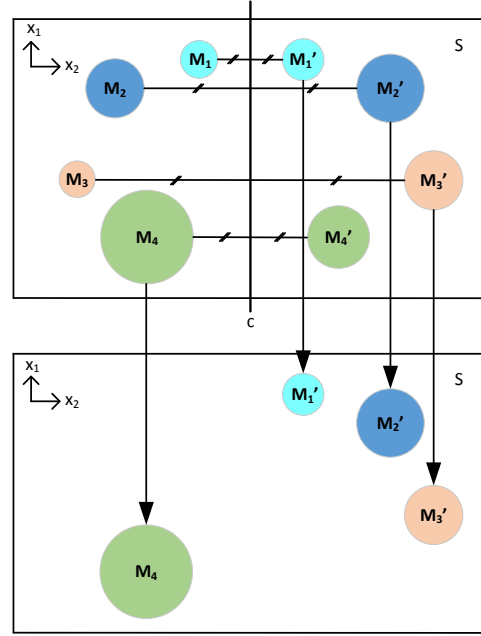


Şekil 5. S arama uzayına sahip, iki boyutlu bir problem için ObGSA-1 algoritması ilk popülasyon yapısı

4.2. Durum 2

İkinci yaklaşımda (ObGSA-2) ise ilk popülasyonda, rastgele olarak oluşturulan bütün ajanların zıt konumları belirlenmiş ve uygunluk değeri daha yüksek olan ajanlarla ilk popülasyon oluşturulmuştur. Bu işlem ile, problemin çözümüne

daha yakın yani uygunluk değeri daha yüksek olan bireyler ile arama işlemine başlanarak GSA'nın yakınsama hızının artırılması amaçlanmıştır. Algoritmanın ilk popülasyon yapısının oluşumu Şekil 6'da gösterilmiştir.



Şekil 6. S arama uzayına sahip, iki boyutlu bir problem için ObGSA-2 algoritması ilk popülasyon yapısı

5. Test Fonksiyonları

Önerilen algoritmaların (ObGSA-1 ve 2), performanslarının değerlendirilebilmesi için literatürde farklı araştırmacılar tarafından daha önce GSA ile çözümü yapılmış olan 23 adet test fonksiyonu seçilmiştir. Bu fonksiyonlar üç grup halinde Çizelge 1, 2 ve 3'te verilmiştir. Çizelgelerde (D) değeri fonksiyonun boyutunu, (S) arama uzayını, f_{min} ise fonksiyonun minimum değerini göstermektedir. Çizelge 1'deki f_{1-7} fonksiyonları tek bir optimum noktaya sahip (unimodal), Çizelge 2'deki f_{8-13} fonksiyonları ise birçok lokal minimum noktaları olan (multimodal) fonksiyonlardır. f_{1-13} fonksiyonları yüksek boyutlu veya geniş arama uzayına sahip fonksiyonlardır. Çizelge 3'te yer alan, f_{14-23} fonksiyonları ise düşük ve sabit boyutlu olup az sayıda lokal minimum noktaları olan multimodal fonksiyonlardır (Rashedi vd. 2009, Rashedi vd. 2010). Çizelge 3'teki fonksiyonlardan $f_{14,15}$ ve f_{19-23}

'e ait $a_i, b_i, c_i, a_{ij}, b_{ij}$ ve P_{ij} değerleri Ekler bölümündeki Ek Çizelge 1-5'te verilmiştir.

Çizelge 1. Unimodal test fonksiyonları

Formülü	Fonksiyon Adı	D	Aralık (S)	f_{min}
$f_1(x) = \sum_{i=1}^n x_i^2$	Sphere	30	$[-100,100]^n$	0
$f_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	Schwefel's No: 2.22	30	$[-10,10]^n$	0
$f_3(x) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2$	Schwefel's No: 1.2	30	$[-100,100]^n$	0
$f_4(x) = \max\{ x_i , 1 \leq i \leq n\}$	Schwefel's No: 2.21	30	$[-100,100]^n$	0
$f_5(x) = \sum_{i=1}^{n-1} \left[100(x_{i+1} - x_i)^2 + (x_i - 1)^2 \right]$	Rosenbrock	30	$[-30,30]^n$	0
$f_6(x) = \sum_{i=1}^n (x_i + 0.5)^2$	Step	30	$[-100,100]^n$	0
$f_7(x) = \sum_{i=1}^n ix_i^4 + \text{random}[0,1]$	Quartic	30	$[-1.28, 1.28]^n$	0

Çizelge 2. Multimodal test fonksiyonları

Formülü	Fonksiyon Adı	D	Aralık (S)	f_{min}
$f_8(x) = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$	Schwefel's No:2.26	30	$[-500,500]^n$	$(-418.9829) \times n$
$f_9(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	Rastrigin	30	$[-5.12, 5.12]^n$	0
$f_{10}(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$	Ackley	30	$[-32, 32]^n$	0
$f_{11}(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	Griewank	30	$[-600, 600]^n$	0
$f_{12}(x) = \frac{\pi}{n} \left\{ 10 \sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] \right\} + \sum_{i=1}^{n-1} u(x_i, a, k, m)$ $y_i = 1 + \frac{x_i + 1}{4}, a = 10, k = 100, m = 4, u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	Penalized No: 1	30	$[-50, 50]^n$	0
$f_{13}(x) = 0.1 \left\{ \sin^2(3\pi x_1) + \sum_{i=1}^n (x_i - 1)^2 [1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2 [1 + \sin^2(2\pi x_n)] \right\} \\ \dots + \sum_{i=1}^n u(x_i, a, k, m), a = 5, k = 100, m = 4$	Penalized No: 2	30	$[-50, 50]^n$	0

Çizelge 3. Sabit boyutlu Multimodal test fonksiyonları

Formülü	Fonksiyon Adı	D	Aralık (S)	f_{min}
$f_{14}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^j (x_i - a_{ij})^6} \right)^{-1}$	Shekel's Foxholes	2	$[-65.53, 65.53]^2$	0.998
$f_{15}(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_i (b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right]$	Kowalik	4	$[-5, 5]^4$	0.0003
$f_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	Six-Hump Camel-Back	2	$[-5, 5]^2$	-1.0316
$f_{17}(x) = \left(x_2 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 - 6 \right)^2 + 10 \left(1 - \frac{1}{8\pi} \right) \cos x_1 + 10$	Branin	2	$[-5, 10] \times [0, 15]$	0.397887
$f_{18}(x) = [1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times \\ \dots [30 + (2x_1 - 3x_2)^2 (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	Goldstein-Price	2	$[-2, 2]^2$	3
$f_{19}(x) = -\sum_{i=1}^4 c_i \exp\left(-\sum_{j=1}^3 a_j (x_j - p_{ij})^2\right)$	Hartman3	3	$[0, 1]^3$	-3.86278
$f_{20}(x) = -\sum_{i=1}^4 c_i \exp\left(-\sum_{j=1}^6 a_j (x_j - p_{ij})^2\right)$	Hartman6	6	$[0, 1]^6$	-3.32237
$f_{21}(x) = -\sum_{i=1}^5 \sum_{j=1}^4 [(x_j - a_{ij})(x_j - a_{ij})^r + c_i]^{-1}$	Shekel5	4	$[0, 10]^4$	-10.1532
$f_{22}(x) = -\sum_{i=1}^7 \sum_{j=1}^4 [(x_j - a_{ij})(x_j - a_{ij})^r + c_i]^{-1}$	Shekel7	4	$[0, 10]^4$	-10.4029

$$f_{23}(x) = -\sum_{i=1}^{10} \sum_{j=1}^4 [(x_j - a_{ij})(x_j - a_{ij})^T + c_i]^{-1}$$

Shekel10 4 [0,10]⁴ -10.5364

6. Sayısal Sonuçlar

Bu çalışmada yer alan test fonksiyonlarının çözümünde kullanılan parametre değerleri, literatürde GSA algoritmasıyla aynı değerler olarak alınmıştır. Bunun nedeni algoritmalar arasındaki karşılaştırmanın doğru ve sağlıklı yapılabilmesi

içindir. Böylelikle geliştirilen metotlar performans, yakınsama, kararlılık ve hız açısından daha doğru değerlendirilmiş olacaktır. Buna göre ObGSA-1 ve ObGSA-2’de kullanılan parametre değerleri Çizelge 4’te verildiği gibi alınmıştır.

Tablo 4. ObGSA parametre değerleri

Fonksiyonlar	İterasyon Sayısı (N)	Ajan Sayısı (S)	Fonksiyon Çağırımı (FCall)	Yerçekimsel Sabit Başlangıç Değeri (G ₀)	Sabit Sayı (α)	Tanımsızlık Sabiti (ε)
f ₁₋₁₃	1000	50	50000	100	20	10 ⁻⁶
f ₁₄₋₂₃	500	50	25000	100	20	10 ⁻⁶

Test fonksiyonlarının çözümü için MATLAB R2015b’de geliştirilen program Intel Xeon E5-2637 v4 3.50 GHz işlemcili ve 128 GB RAM bellekli iş istasyonunda Çizelge 1 ve 2’deki fonksiyonlar için 1000 iterasyon (50000 fonksiyon çağırımı, FCall) ve Çizelge 3’teki fonksiyonlar için 500 iterasyon (25000 fonksiyon çağırımı, FCall) çalıştırılmıştır.

Çizelge 1 ve 2’de tanımlanan unimodal ve multimodal fonksiyonlar için 30-D için elde edilen sonuç değerleri Çizelge 5 ve 6’da verilmiştir. Çizelge 3’te tanımlanan sabit boyutlu multimodal fonksiyonlar için elde edilen çözüm değerleri ise Çizelge 7’de verilmiştir.

Çizelge 5. 30-D için elde edilen veriler (Çizelge 1, 30 çalışma - 1000 iterasyon)

	ObGSA-1	ObGSA-2		ObGSA-1	ObGSA-2
f ₁	Worst	1.066443e+01	f ₅	Worst	8.656350e+01
	Average	1.380808e+00		Average	2.706349e+01
	Best	1.365652e-12		Best	2.074149e+01
	StdDev	2.411636e+00		StdDev	1.186378e+01
	Time	2.16501		Time	2.09321
f ₂	Worst	4.716947e-05	f ₆	Worst	7.659818e+00
	Average	1.671869e-06		Average	1.358154e+00
	Best	4.009520e-08		Best	7.598752e-14
	StdDev	8.448853e-06		StdDev	1.846968e+00
	Time	2.26315		Time	2.11734
f ₃	Worst	2.057761e+01	f ₇	Worst	4.617968e-02
	Average	4.646430e+00		Average	1.672162e-02
	Best	3.028937e-03		Best	4.692711e-03
	StdDev	4.993821e+00		StdDev	9.379363e-03
	Time	2.38989		Time	2.0266

	Worst	2.603601e+00	2.349976e+00
	Average	1.336551e+00	1.851467e-01
f_4	Best	1.468560e-01	9.738700e-08
	StdDev	5.089365e-01	4.745116e-01
	Time	2.45213	1.69728

Çizelge 6. 30-D için elde edilen veriler (Çizelge 2 - 30 çalışma - 1000 iterasyon)

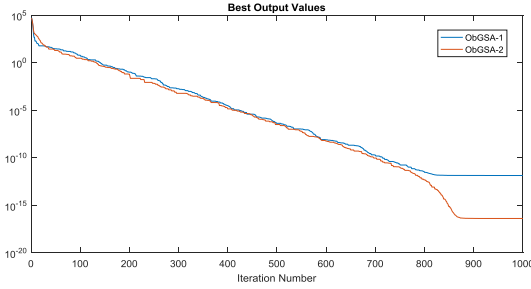
		ObGSA-1	ObGSA-2			ObGSA-1	ObGSA-2
f_8	Worst	-1.730552e+03	-2.178011e+03	f_{11}	Worst	5.880381e-02	3.247729e+00
	Average	-2.608178e+03	-2.943003e+03		Average	6.638065e-03	1.361523e+00
	Best	-3.548445e+03	-4.140124e+03		Best	0.000000e+00	0.000000e+00
	StdDev	4.741298e+02	4.755051e+02		StdDev	1.289413e-02	8.342248e-01
	Time	1.92572	2.01641		Time	2.47056	2.11075
f_9	Worst	2.055825e+02	1.981292e+02	f_{12}	Worst	4.146722e-01	1.036692e-01
	Average	1.688443e+02	1.614077e+02		Average	2.418951e-02	6.911476e-03
	Best	5.401875e+01	2.102973e+01		Best	1.176842e-07	9.172671e-08
	StdDev	3.320461e+01	3.969982e+01		StdDev	7.887565e-02	2.585960e-02
	Time	2.05912	1.90469		Time	2.21105	2.02591
f_{10}	Worst	3.346689e+00	1.340421e+00	f_{13}	Worst	1.104330e+01	1.150080e+01
	Average	3.015097e-01	2.073597e-01		Average	3.925010e-01	5.573520e-01
	Best	2.290611e-08	2.085781e-08		Best	1.938549e-17	1.656850e-17
	StdDev	7.102950e-01	4.207322e-01		StdDev	1.979782e+00	2.231900e+00
	Time	2.41483	1.85494		Time	2.17271	2.03207

Çizelge 7. Sabit boyutlu fonksiyonlar için elde edilen veriler (Çizelge 3 - 30 çalışma - 500 iterasyon)

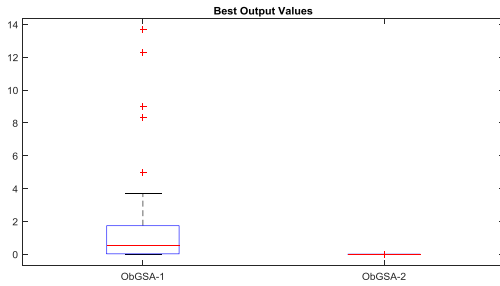
		ObGSA-1	ObGSA-2			ObGSA-1	ObGSA-2
f_{14}	Worst	2.066390e+00	3.603355e+00	f_{19}	Worst	-3.770885e+00	-3.774014e+00
	Average	1.061234e+00	1.282673e+00		Average	-3.857785e+00	-3.859678e+00
	Best	9.980039e-01	9.980038e-01		Best	-3.862782e+00	-3.862782e+00
	StdDev	1.945656e-01	6.802949e-01		StdDev	1.876127e-02	1.592661e-02
	Time	1.13278	1.07075		Time	0.729532	0.705061
f_{15}	Worst	2.071401e-01	2.071401e-01	f_{20}	Worst	-3.176203e+00	-3.155381e+00
	Average	1.838029e-01	1.682407e-01		Average	-3.237117e+00	-3.248387e+00
	Best	5.539650e-02	2.299353e-02		Best	-3.321996e+00	-3.321996e+00
	StdDev	4.573953e-02	5.516680e-02		StdDev	5.577679e-02	6.068528e-02
	Time	1.01336	0.968433		Time	0.872941	0.849832
f_{16}	Worst	-1.031080e+00	-1.031628e+00	f_{21}	Worst	-2.630472e+00	-2.630472e+00
	Average	-1.031595e+00	-1.031628e+00		Average	-6.707436e+00	-6.611098e+00
	Best	-1.031628e+00	-1.031628e+00		Best	-1.015320e+01	-1.015320e+01
	StdDev	1.107588e-04	1.152429e-12		StdDev	3.132268e+00	3.091600e+00
	Time	0.85123	0.75055		Time	0.97736	0.84177
f_{17}	Worst	3.978874e-01	3.978874e-01	f_{22}	Worst	-3.724300e+00	-2.751934e+00
	Average	3.978874e-01	3.978874e-01		Average	-9.410883e+00	-9.501208e+00
	Best	3.978874e-01	3.978874e-01		Best	-1.040294e+01	-1.040294e+01
	StdDev	0.000000e+00	3.188873e-16		StdDev	2.232526e+00	2.308527e+00
	Time	0.665883	0.57112		Time	1.24842	1.08095
	Worst	3.000002e+00	3.000000e+00		Worst	-2.871143e+00	-3.835427e+00

	Average	3.000000e+00	3.000000e+00		Average	-1.028090e+01	-9.866312e+00
f_{18}	Best	3.000000e+00	3.000000e+00	f_{23}	Best	-1.053641e+01	-1.053641e+01
	StdDev	3.362972e-07	5.916844e-09		StdDev	1.375958e+00	2.010295e+00
	Time	0.600939	0.681809		Time	1.18703	0.923511

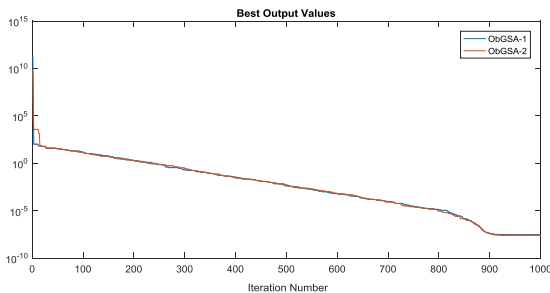
Çizelge 5’de yer alan unimodal f_1 ve f_2 fonksiyonları için 30 çalışmada elde edilen en iyi çözümlere ait, iterasyon sayısına göre yakınsamayı gösteren grafikler Şekil 7 ve 9’da ve en iyi değerlerin yayılımını gösteren kutu grafikleri ise Şekil 8 ve 10’da verilmiştir.



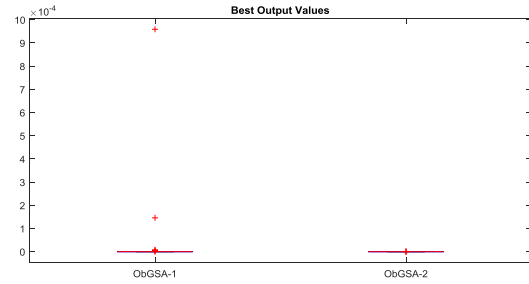
Şekil 7. f_1 için en iyi sonuçların elde edildiği yakınsama eğrileri (30-D)



Şekil 8. f_1 için kutu grafikleri (30-D)



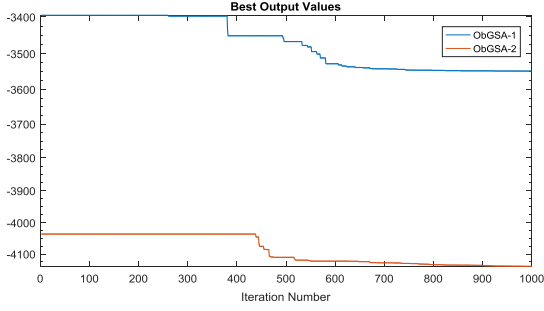
Şekil 9. f_2 için en iyi sonuçların elde edildiği yakınsama eğrileri (30-D)



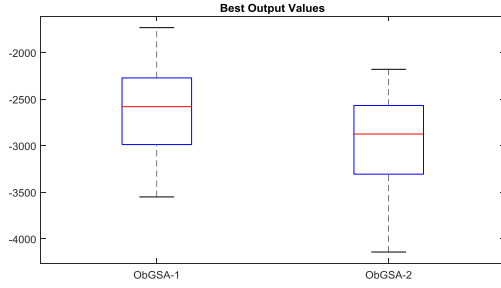
Şekil 10. f_2 için kutu grafikleri (30-D)

Çizelge 5’te bulunan fonksiyonlar için örnek olarak alınan fonksiyonlara ait grafikler incelendiğinde, yakınsama eğrilerinden ObGSA-2 yaklaşımının ObGSA-1 yaklaşımına göre daha iyi değerlere yakınsadığı görülmüştür. Fonksiyonların 30 kez çözümünden elde edilen kutu grafiklerine bakıldığında ise ObGSA-1 yaklaşımında f_1 fonksiyonu için 5 adet değer, f_2 fonksiyonu için ise 3 adet değer sapma göstermiştir, buna karşın her iki fonksiyon içinde ObGSA-2 yaklaşımı 30 çalışmada da minimum değeri kararlılıkla yakalamıştır.

Çizelge 6’da yer alan multimodal f_8 ve f_9 fonksiyonları için 30 çalışmada elde edilen en iyi çözümlere ait, iterasyon sayısına göre yakınsamayı gösteren grafikler ve en iyi değerlerin yayılımını gösteren kutu grafikleri Şekil 11, 12, 13 ve 14’te gösterilmiştir. Literatürde rastlandığı gibi benzer grafikler çizelgedeki bütün fonksiyonlar için elde edilmiş, fakat çok fazla yer kaplaması nedeniyle çalışmada gösterilmemiştir. Ama sonuç ve değerlendirmeler için çalışmada verilmeyen grafiklerde dikkate alınmıştır.

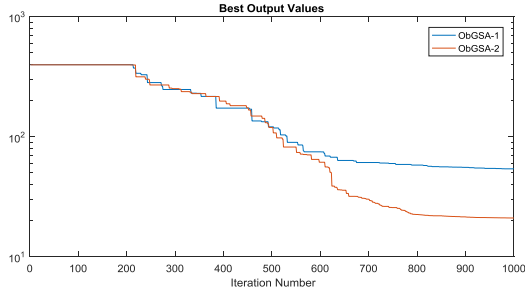


Şekil 11. f_8 için en iyi sonuçların elde edildiği yakınsama eğrileri (30-D)

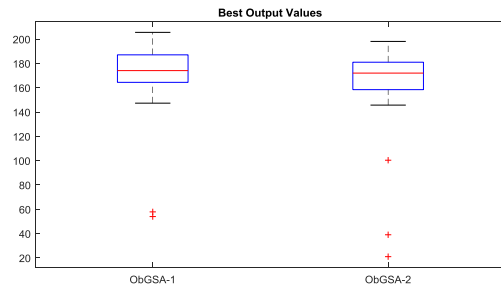


Şekil 12. f_8 için kutu grafikleri (30-D)

10



Şekil 13. f_9 için en iyi sonuçların elde edildiği yakınsama eğrileri (30-D)

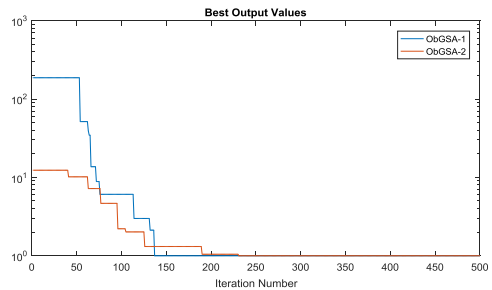


Şekil 14. f_9 için kutu grafikleri (30-D)

yaklaşımına göre yakınsama yönünden üstünlüğü göze çarpmaktadır. Fonksiyonların çözümünden elde edilen kutu grafiklerine bakıldığında ise f_8 fonksiyonu için ObGSA-2 yaklaşımının minimuma daha yakın bir ortanca değer yakaladığı, f_9 fonksiyonu için ise ObGSA-1 yaklaşımının ortanca değerinin yüksek olmasına rağmen daha az sayıda sapma gösteren değere sahip olduğu görülmüştür.

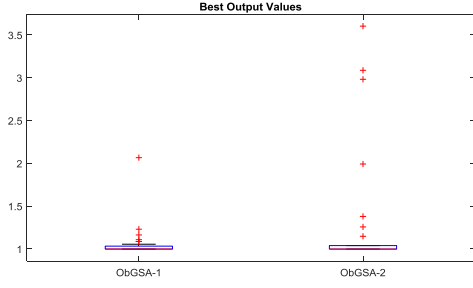
Çizelge 7'de yer alan sabit boyutlu f_{14} ve f_{15} fonksiyonları için 30 çalışma ile elde edilen en iyi çözümlere ait, iterasyon sayısına göre yakınsamayı gösteren grafikler Şekil 15 ve 17'de ve aynı fonksiyonlara ait 30 çalışma için en iyi değerlerin yayılımını gösteren kutu grafikleri Şekil 16 ve 18'de verilmiştir.

Çizelge 7'den seçilen sabit boyutlu f_{14} ve f_{15} fonksiyonları için çizdirilen grafiklerde, f_{14} için ObGSA-1 yaklaşımının, f_{15} için ise ObGSA-2 yaklaşımının yakınsama bakımından üstünlüğü görülmektedir. Çözümlerden elde edilen kutu grafiklerine bakıldığında ise f_{14} fonksiyonu için yakınsamalarda olduğu gibi ObGSA-1 yaklaşımının hem minimuma daha yakın bir ortanca değer yakaladığı hem de daha az sapma gösteren değere sahip olduğu anlaşılmaktadır. f_{15} fonksiyonu için ise ObGSA-1 yaklaşımdaki 6 adet sapma gösteren değere karşın ObGSA-2 yaklaşımda sapma gösteren değer bulunmamaktadır.

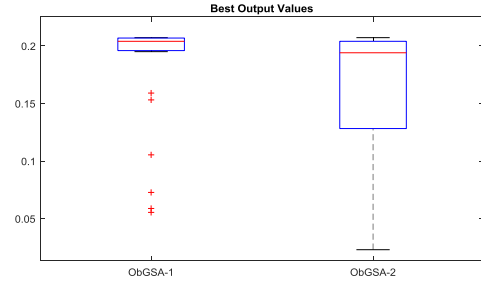


Şekil 15. f_{14} için en iyi sonuçların elde edildiği yakınsama eğrileri (2-D)

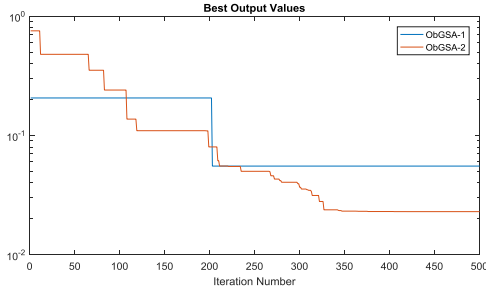
Çizelge 6'dan örnek olarak alınan fonksiyonlara ait grafiklerde ObGSA-2 yaklaşımının ObGSA-1



Şekil 16. f_{14} için kutu grafikleri (2-D)



Şekil 18. f_{15} için kutu grafikleri (4-D)



Şekil 17. f_{15} için en iyi sonuçların elde edildiği yakınsama eğrileri (4-D)

Çalışmada bütün fonksiyonlar için elde edilen en iyi sonuçlar (Rashedi vd. 2009)'da yer alan 30-D için verilen en iyi sonuçlarla Çizelge 8'de karşılaştırılmıştır. Literatürden alınan sayısal değerler, karşılaştırmaların daha kolay anlaşılabilmesi için çalışmadaki formatla aynı şekilde yazılmıştır. Çalışmada geliştirilen ObGSA-1 ve 2 algoritmaları kendi aralarında karşılaştırıldığında, en iyi sonuç değeri çizelgede koyu renkle gösterilmiştir.

Çizelge 8. 30-D için literatür karşılaştırması

Best Outputs	f_{min}	GSA [1]	ObGSA-1	ObGSA-2
f_1	0	7.300000e-11	1.365652e-12	4.104103e-17
f_2	0	4.030000e-05	4.009520e-08	2.986695e-08
f_3	0	0.160000e+03	7.710099e-06	9.866712e-07
f_4	0	3.700000e-06	1.468560e-01	9.738700e-08
f_5	0	2.516000e+01	2.074149e+01	2.020163e+01
f_6	0	8.300000e-11	7.598752e-14	6.291316e-17
f_7	0	0.018000e+00	4.692711e-03	2.347075e-03
f_8	-418.9829 x n	-2.800000e+03	-3.548445e+03	-4.140124e+03
f_9	0	1.532000e+01	5.401875e+01	2.102973e+01
f_{10}	0	6.900000e-06	2.290611e-08	2.085781e-08
f_{11}	0	0.290000e+00	0.000000e+00	0.000000e+00
f_{12}	0	0.010000e+00	1.176842e-07	9.172671e-08
f_{13}	0	3.200000e-32	1.938549e-17	1.656850e-17
f_{14}	1	3.700000e+00	9.980039e-01	9.980038e-01
f_{15}	0.0003	8.000000e-03	5.539650e-02	2.299353e-02
f_{16}	-1.0316	-1.031600e+00	-1.031628e+00	-1.031628e+00
f_{17}	0.398	0.397900e+00	3.978874e-01	3.978874e-01
f_{18}	3	3.000000e+00	3.000000e+00	3.000000e+00
f_{19}	-3.86	-3.735700e+00	-3.862782e+00	-3.862782e+00
f_{20}	-3.32	-2.056900e+00	-3.321996e+00	-3.321996e+00
f_{21}	-10.1532	-6.074800e+00	-1.015320e+01	-1.015320e+01
f_{22}	-10.4028	-9.339900e+00	-1.040294e+01	-1.040294e+01
f_{23}	-10.5363	-9.454800e+00	-1.053641e+01	-1.053641e+01

Çizelge 8'de verilen değerler incelendiğinde, ObGSA-1 ve 2 ile elde edilen sonuçların, GSA'dan oldukça iyi oldukları görülmektedir. ObGSA-1, 15 fonksiyonda GSA'dan daha iyi sonuçlar bulmasına karşın, ObGSA-2, 20 fonksiyonda daha iyi sonuçlara

yakınsamıştır. ObGSA-1 ve 2 kendi aralarında karşılaştırıldığında ise, ObGSA-2'nin yaklaşımının 1'e göre daha üstün olduğu söylenebilir. Aynı parametre değerleriyle yapılan çözümlerde ObGSA-

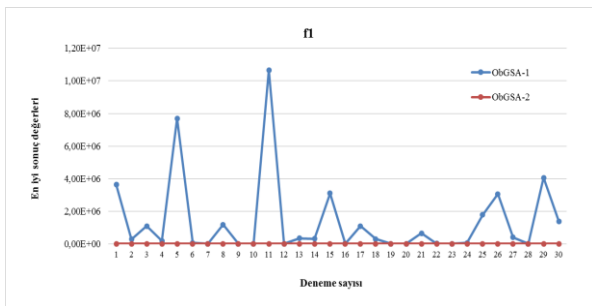
2 yaklaşımı, birçok fonksiyonda daha iyi sonuçlar yakalamıştır.

7. Performans Değerlendirilmesi

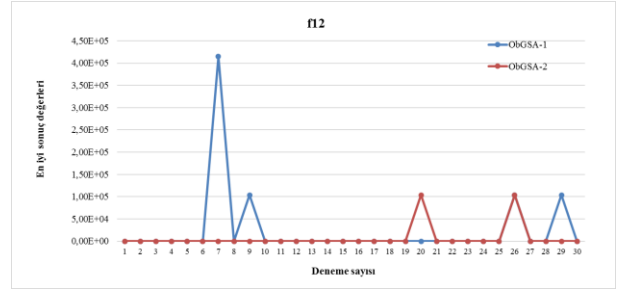
Literatürde bu tür çalışmaları istatistiki olarak karşılaştırabilmek için elde edilen sonuçlara eşleştirilmiş t-testi (Paired sample T-Test) ve Wilcoxon testi uygulanmaktadır (Garcia vd. 2009). Veri sayısının az olması durumunda parametrik olan eşleştirilmiş t-testi tam doğru sonuç veremediğinden, parametrik olmayan Wilcoxon testi uygulanarak daha hassas ve doğru sonuç elde edilmektedir. Bu nedenle Çizelge 8’de yer alan 23 adet test fonksiyonuna ait GSA, ObGSA-1 ve 2 algoritmalarına ait 30 çalışmanın en iyi değerleri, eşleştirilmiş t-testi ve Wilcoxon testi gibi istatistiki değerlendirme testlerine tabi tutulmuştur. Elde edilen istatistiki test sonuçları Çizelge 9’da verilmiştir. Verilerin analizinde anlamlılık düzeyi literatürdeki gibi 0.05 olarak alınmıştır.

ObGSA-1 ve 2, 23 adet test fonksiyonu için 30 kez çalıştırılmıştır. Elde edilen en iyi değerlerin 30 çalışma için kararlılıklarını gösteren grafikler, seçilen f_1 , f_{12} ve f_{14} fonksiyonları için Şekil 19, 20 ve 21’de gösterilmiştir.

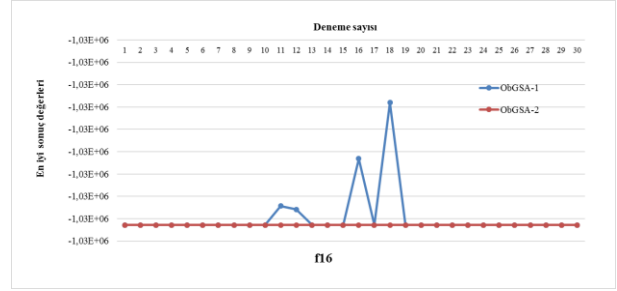
12



Şekil 19. f_1 için 30 çalışma en iyi sonuçları (30-D)



Şekil 20. f_{12} için 30 çalışma en iyi sonuçları (30-D)



Şekil 21. f_{16} için 30 çalışma en iyi sonuçları (2-D)

ObGSA-2 algoritması, aynı parametrelerle 30 çalışma değeri için ObGSA-1 algoritmasına göre daha fazla sayıda minimum sonuca yakınsamıştır. Bu nedenle ObGSA-2’nin, ObGSA-1’den daha kararlı bir yapıya sahip olduğu sonucuna varılmıştır.

GSA ve ObGSA algoritmalarıyla 23 adet test fonksiyonu için elde edilen optimum değerler arasında anlamlı bir farklılığın bulunduğu görülmektedir. Veriler dikkatle incelendiğinde, bu anlamlı farklılığın ObGSA-2 lehine daha fazla olduğu görülmektedir.

Çizelge 9. f_{1-23} fonksiyonlarına ait 30 çalışma ve 30-D için elde edilen en iyi değerler için istatistiki test sonuçları

	T-Test (p-değeri)	Wilcoxon (p-değeri)
f_{1-7}		
GSA vs ObGSA-1	0.3423	0.1563
GSA vs ObGSA-2	0.3402	0.0156
ObGSA-1 vs 2	0.2451	0.0156
f_{8-13}		
GSA vs ObGSA-1	0.3914	0.4375
GSA vs ObGSA-2	0.3654	0.4375
ObGSA-1 vs 2	0.3352	0.0625

f_{14-23}		
GSA vs ObGSA-1	0.0439	0.0195
GSA vs ObGSA-2	0.0430	0.0195
ObGSA-1 vs 2	0.3434	0.5000
f_{1-23}		
GSA vs ObGSA-1	0.2572	0.0067
GSA vs ObGSA-2	0.2726	0.0019
ObGSA-1 vs 2	0.3015	0.00012207

f_{1-7} fonksiyonları için GSA'nın, ObGSA-1 ile arasında anlamlı bir farkın olmadığı ($p > 0.05$), fakat ObGSA-2 ile aralarında anlamlı bir farklılık olduğu ($p < 0.05$) görülmüştür. f_{8-13} fonksiyonları için GSA ile her iki yaklaşım arasında da anlamlı bir fark olmasına rağmen ($p < 0.05$), f_{14-23} fonksiyonları için anlamlı bir farklılığın olmadığı ($p > 0.05$) görülmüştür. Çözümü yapılan fonksiyonların tamamına f_{14-23} bakıldığında ise, GSA algoritması ile her iki yaklaşım arasında da yapılan çalışmanın amacını doğrular nitelikte anlamlı farklılıklar olduğu ($p < 0.05$) görülmüştür. Veriler dikkate alındığında, bu anlamlı farklılığın ObGSA-2 yaklaşımı lehine daha fazla olduğu görülmektedir.

Ekler

Ek Çizelge 1: $f_{14}(x)$ fonksiyonundaki a_{ij} katsayıları

j	1	2	3	4	5	...	21	22	23	24	25
$a_{ij} \ i=1,2$	-32	-16	0	16	32	...	-32	-16	0	16	32
	-32	-32	-32	-32	-32	...	32	32	32	32	32

Ek Çizelge 2: $f_{15}(x)$ fonksiyonundaki a_i ve b_i katsayıları

i	1	2	3	4	5	6	7	8	9	10	11
a_i	0.1957	0.1947	0.1735	0.1600	0.0844	0.0627	0.0456	0.0342	0.0323	0.0323	0.0246
b_i^{-1}	0.25	0.5	1	2	4	6	8	10	12	14	16

Ek Çizelge 3: $f_{19}(x)$ fonksiyonundaki a_{ij} , c_i ve P_{ij} katsayıları

i	$a_{ij}, j=1,2,3$			c_i	$P_{ij}, j=1,2,3$		
1	3	10	30	1	0.3689	0.1170	0.2673
2	0.1	10	35	1.2	0.4699	0.4387	0.7470
3	3	10	30	3	0.1091	0.8732	0.5547
4	0.1	10	30	3.2	0.03815	0.5743	0.8828

Ek Çizelge 4: $f_{20}(x)$ fonksiyonundaki a_{ij} , c_i ve P_{ij} katsayıları

i	$a_{ij}, j=1,2,3,4,5,6$	c_i	$P_{ij}, j=1,2,3,4,5,6$
-----	-------------------------	-------	-------------------------

Teşekkür

Bu çalışma Dumlupınar Üniversitesi, Bilimsel Araştırma Projeleri Komisyonu (BAP) tarafından, 2016-65 nolu proje kapsamında desteklenmiştir.

8. Sonuç

Çalışmada zıt konumlu öğrenme yapısı GSA'ya iki farklı yaklaşımla entegre edilerek ObGSA-1 ve 2 algoritmaları geliştirilmiştir. Her iki yaklaşımda 23 adet test fonksiyonu için 30'ar kez çözülmüştür. Bu çözümlerin tamamında ObGSA-1 ve 2 algoritmaları GSA'dan daha iyi sonuçlar vermiştir. Bu nedenle optimizasyon problemlerinin çözümünde daha iyi sonuçları daha kısa sürelerde elde etmek için bu çalışmada geliştirilen yaklaşımlar tercih edilebilir. Yaklaşımlar arasında bir karşılaştırma yapıldığında ise ObGSA-2'nin 1'den uygunluk, kararlılık ve süre bakımından daha iyi olduğu görülmüştür.

1	10	3	17	3.5	1.7	8	1	0.131	0.169	0.556	0.012	0.828	0.588
2	0.05	10	17	0.1	8	14	1.2	0.232	0.413	0.830	0.373	0.100	0.999
3	3	3.5	1.7	10	17	8	3	0.234	0.141	0.352	0.288	0.304	0.665
4	17	8	0.05	10	0.1	14	3.2	0.404	0.882	0.873	0.574	0.109	0.038

Ek Çizelge 5: $f_{21}(x)$, $f_{22}(x)$ ve $f_{23}(x)$ fonksiyonlarındaki a_{ij} ve c_i katsayıları

i	$a_{ij}, j=1,2,3,4$				c_i
1	4	4	4	4	0.1
2	1	1	1	1	0.2
3	8	8	8	8	0.2
4	6	6	6	6	0.4
5	3	7	3	7	0.4
6	2	9	2	9	0.6
7	5	5	3	3	0.3
8	8	1	8	1	0.7
9	6	2	6	2	0.5
10	7	3.6	7	3.6	0.5

Kaynaklar

Askarzadeh, A., 2016. A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm, *Computers and Structures*, **169**, 1-12.

Cheng, M.Y., Prayogo, D., 2014. Symbiotic organisms search: A new metaheuristic optimization algorithm, *Computers and Structures*, **139**, 98-112.

Cura, Tunçhan, 2008. Modern sezgisel teknikler ve uygulamaları. Papatya Yayıncılık, 14-15.

Doğan, B., Ölmez, T., 2015. A new metaheuristic for numerical function optimization: Vortex search algorithm, *Information Sciences*, **293**, 125-145.

Ergezer, M., Simon, D., and Du, D., 2009. Oppositional biogeography-based optimization, in: Proceedings of IEEE International Conference on Systems, Man, and Cybernetics (ICSMC'09), pp. 1009-1014.

García, S., Molina, D., Lozano, M., and Herrera, F., 2009. A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 Special Session on Real Parameter Optimization, *Journal of Heuristics*, **15**, 617-44.

Geem, Z.W., Kim, J.H., and Loganathan, G.V., 2001. A new heuristic optimization algorithm: Harmony search, *Simulation*, **76 (2)**, 60-68.

Goldberg, D.E., Genetic Algorithms in Search, Optimization, and Machine Learning, 1989. Addison-Wesley Publishing Company, Inc.

Karaboğa, D., Baştürk, B., 2007. A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm, *Journal of Global Optimization*, **39 (3)**, 459-471.

Kashan, A.H., 2015. A new metaheuristic for optimization: Optics inspired optimization (OIO), *Computers and Operations Research*, **55**, 99-125.

Kaveh, A., Talahatari, S., 2010. A novel heuristic optimization method: Charged system search, *Acta Mechanica*, **213 (3-4)**, 267-289.

Kennedy, J., Eberhart, R., 1995. Particle Swarm Optimization, *Proceedings of IEEE International Conference on Neural Networks*, **5**. 1942-1948.

Li, C., Zhou, J., 2011. Parameters identification of hydraulic turbine governing system using improved gravitational search algorithm, *Energy Conversion and Management*, **52 (1)**, 374-381.

Mirjalili, S., 2015. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm, *Knowledge-Based Systems*, **89**, 228-249.

- Mirjalili, S., 2016. SCA: A sine cosine algorithm for solving optimization problems, *Knowledge-Based Systems*, **96**, 120-133.
- Mirjalili, S., Lewis, A., 2016. The whale optimization algorithm, *Advances in Engineering Software*, **95**, 51-67.
- Mirjalili, S., Mirjalili, S.M., and Hatamlou, A., 2016. Multi-verse optimizer: A nature -inspired algorithm for global optimization, *Neural Computing and Applications*, **27**, 495-513.
- Mirjalili, S., Mirjalili, S.M., and Lewis, A., 2014. Grey wolf optimizer, *Advances in Engineering Software*, **69**, 46-61.
- Omran, M.G.H., 2009. Using opposition-based learning with particle swarm optimization and barebones differential evolution, Particle Swarm Optimization, Aleksandar Lazinica (Ed.), InTech, 373-384.
- Rahmani, R., Yusof, R., 2014. A new simple, fast and efficient algorithm for global optimization over continuous search-space problems: Radial movement optimization, *Applied Mathematics and Computation*, **248**, 287-300.
- Rahnamayan, S., Tizhoosh, H.R., and Salama, M.M.A., 2008. Opposition-based differential evolution, *IEEE Transactions on Evolutionary Computation*, **12 (1)**, 64-79.
- Rajabioun, R., 2011. Cuckoo optimization algorithm, *Applied Soft Computing*, **11**, 5508-5518.
- Rashedi, E., Nezamabadi-pour, H. and Saryazdi, S., 2009. GSA: A gravitational search algorithm, *Information Sciences*, **179 (13)**, 2232-2248.
- Rashedi, E., Nezamabadi-pour, H. and Saryazdi, S., 2010. BGSA: Binary gravitational search algorithm, *Natural Computing*, **9 (3)**, 727-745.
- Rashedi, E., Nezamabadi-pour, H., and Saryazdi, S., 2011. Filter modeling using gravitational search algorithm, *Engineering Applications of Artificial Intelligence*, **24**, 117-122.
- Singh, R.P., Mykherje, V., and Ghoshal, S.P., 2013. The opposition-based harmony search algorithm, *Journal of the Institution of Engineers (India): Series B*, **94 (4)**, 247-256.
- Storn, R., Price, K., 1997. Differential evolution - A simple and efficient heuristic for global optimization over continuous spaces, *Journal of Global Optimization*, **11**, 341-359.
- Swain, R.K., Sahu, N.C., and Hota, P.K., 2012. Gravitational search algorithm for optimal economic dispatch, *Procedia Technology*, **6**, 411-419.
- Tizhoosh, H.R., 2005. Opposition-based learning: A new scheme for machine intelligence, in: Proceedings of International Conference on Computational Intelligence for Modelling, Control and Automation, and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'05), vol. I, pp. 695-701.
- Zahiri, S.H., 2012. Fuzzy gravitational search algorithm an approach for data miming, *Iranian Journal of Fuzzy Systems*, **9 (1)**, 21-37.
- Zhang, W., Niu, P., Li, G., and Li, P., 2013. Forecasting of turbine heat rate with online least squares support vector machine based on gravitational search algorithm, *Knowledge-Based Systems*, **39**, 34-44.
- Zheng, Y.J., 2015. Water wave optimization: A new nature-inspired metaheuristic, *Computers and Operations Research*, **55**, 1-11.