



Artımsal Popülasyonlu Karga Arama Algoritması

Burhanettin Durmuş^{1*}, Ayhan Gün², Hasan Temurtaş³

^{1*} Kütahya Dumlupınar Üniversitesi, Mühendislik Fakültesi, Elektrik-Elektronik Bölümü, Kütahya, Türkiye, (ORCID: 0000-0002-8225-3313),

burhanettin.durmus@dpu.edu.tr

² Kütahya Dumlupınar Üniversitesi, Mühendislik Fakültesi, Elektrik-Elektronik Bölümü, Kütahya, Türkiye (ORCID: 0000-0002-4223-2518), ayhan.gun@dpu.edu.tr

³ Kütahya Dumlupınar Üniversitesi, Mühendislik Fakültesi, Bilgisayar Bölümü, Kütahya, Türkiye (ORCID: 0000-0001-6738-3024), hasan.temurtas@dpu.edu.tr

(İlk Geliş Tarihi 4 Mayıs 2021 ve Kabul Tarihi 5 Ağustos 2021)

(DOI: 10.31590/ejosat.932389)

ATIF/REFERENCE: Durmuş, B., Gün, A. & Temurtaş, H. (2021). Artımsal Popülasyonlu Karga Arama Algoritması. *Avrupa Bilim ve Teknoloji Dergisi*, (25), 615-622.

Öz

Karga arama algoritması (CSA) kargaların zekâ davranışlarına temellendirilen yeni meta-sezgisellerden biridir. Basit yapısı ve az sayıda parametreye ihtiyaç duyması ona avantaj sağlamasına rağmen, erken yakınsama problemi ve yerel optimuma kolayca düşmesi özellikle çokmodlu (MM) problem çözümlerinde performansını düşürmektedir. Bu çalışmada, CSA'nın bu zayıflığını güçlendirmek ve etkinliğini arttırmak için artımsal popülasyon (IPOP) temelli CSA (ICSA) algoritmaları geliştirilmektedir. Genişleyen bir popülasyonu temel alan IPOP stratejisi ile hesaplama boyunca çözüm çeşitliliğinin sağlanması hedeflenmektedir. Geliştirilen dört adet CSA versiyonu 100-boyutlu test fonksiyonlarına uygulanarak performansları gözlemlenmiştir. Elde edilen sonuçlar, önerilen metodların temel CSA'nın performansını iyileştirdiğini göstermektedir.

Anahtar Kelimeler: Meta-sezgisel, Optimizasyon, Karga arama algoritması.

Crow Search Algorithm with Incremental Population

Abstract

The crow search algorithm (CSA) is one of the new metaheuristics based on the intelligence behavior of crows. Although its simple structure and the need for few parametric adjustments give it an advantage, the problem of early convergence and easily falling to the local optimum decreases its performance in multimodal problems. In this paper, incremental population (IPOP) based CSA (IPOP-CSA) algorithm has been developed to strengthen this weakness of CSA and increase its efficiency. Using the IPOP strategy based on an expanding population, it is aimed to maintain the diversity of solutions throughout evolution. The four CSA versions developed are applied to 100-dimensional test functions to monitor their performance. The results obtained show that the proposed methods improve the performance of the basic CSA.

Keywords: Metaheuristic, Optimization, Crow search algorithm.

* Sorumlu Yazar: burhanettin.durmus@dpu.edu.tr

1. Giriş

Doğadaki birçok canlının davranış özellikleri içgüdüsel ve rastgele görünüyorsa olmasına rağmen, belirli bir akıl ve öğrenme kurallarına dayanır. Özellikle sürü halinde yaşayan canlıların ortak akılla hareketleri, onlara yiyecek arama, hayatta kalma ve avlanma yeteneklerinde önemli derecede katkı sağlamaktadırlar. Canlıların sahip olduğu içgüdüsel ve rastlantısal davranışlar, optimizasyon temelli çözüm metodlarının gelişimine de ilham olmaktadır. Gerek tasarım gerekse üretim safhalarında karşılaşılan problemlerin çözümlenmelerinde en uygun çözümü bulma hedefi, çeşitli canlıların hareket stratejilerini taklit ederek optimizasyon hesaplamaları temelinde yeni metodların üretilmesini sağlamaktadırlar. ABC (Karaboğa ve Baştürk, 2007), ACO (Dorigo ve Di Caro, 1999), COA (Rajabioun, 2011), PSO (Kennedy ve Eberhart, 1995), GWO (Mirjalili ve ark., 2014) ve WOA (Mirjalili ve Lewis, 2016) gibi algoritmalar bunların en iyi bilinenleridir ve literatürde yaygın bir şekilde kabul görmüşlerdir.

Öteyandan bu meta-sezgisellerin problem çözümlenmelerindeki performans kabiliyetlerini ve etkinliklerini arttırmak için kaos, zıt konumluluk ve hibritleştirme gibi çeşitli stratejiler bir araya getirilmektedir. Bu iyileştirmeleri konu alan çalışmaların bazıları şu şekildedir; kaotik haritalama yaklaşımları algoritmanın ilkendirme aşamasında kullanılarak CS-PSO algoritması geliştirilmiştir (Xu ark., 2018). Doğrusal popülasyon boyutu olasılığı, global arama faktörü (Yu ve ark., 2018), zıt konumlu öğrenme (Gao ve Liu, 2011) ve algoritmik bileşen konfigürasyonu (Aydın ve ark., 2017) gibi araçları kullanan çok sayıda ABC versiyonu geliştirilmiştir. Öz-öğrenme ve dinamik bilgi paylaşımı temelinde dayandırılan geliştirilmiş ACO algoritmaları sunulmuştur (Yang ve Zhuang, 2010; Zhang ve Zhang, 2017). Benzer yaklaşımlar ile COA, GWO ve WOA 'nın gelişmiş versiyonlarının sunulduğu birçok makale rapor edilmektedir (Abdollahi ve ark., 2016; Li ve ark., 2018; Sahoo ve Chandra, 2017; Saidala ve Devarakonda, 2018; Kaur ve Arora, 2018). Sonuç olarak, optimizasyon problemlerinin çözümünde meta-sezgisel algoritmaların kabiliyetlerinin artırılması güncelliğini koruyan bir alan olmayı sürdürmektedir.

CSA, son dönemde önerilmiş doğa esinli sezgisel algoritmalarından biridir (Askarzadeh, 2016). Kargaların zeka davranışları temelinde geliştirilen bu algoritma, çeşitli optimizasyon problemlerinin çözümünde kayda değer başarımlar sunmaktadır. CSA 'nın basit yapısı ve az sayıdaki parametrik ayarlamaya ihtiyaç duyması onun çeşitli problemlerin çözümüne adaptasyonunu kolay kılmaktadır (Askarzadeh, 2016). Ancak CSA, arama stratejisinin efektif olmayışı sebebiyle MM problemlerinin çözümünde düşük bir performans sergilemektedir. Bundan dolayı bu çalışmada CSA 'nın kabiliyetini arttırmaya odaklanılmıştır. Bu bağlamda, CSA 'nın çözüm çeşitliliğini sürdürmek ve yakınsama performansını geliştirmek için ICSA algoritmaları geliştirilmiştir. IPOP yaklaşımının algoritmaların performansına önemli ölçüde katkı sağladığı literatürde raporlanmaktadır (Aydın ve Özyön, 2013; Özyön ve ark., 2018). Bu strateji temel olarak mevcut popülasyona belirli iterasyon adımlarında yeni ajan eklemeye dayanır. Bu sayede hesaplama sonuna kadar algoritmanın aday çözüm üretimindeki çeşitlilik sürdürülmektedir. Bu çalışmada, 4 farklı IPOP yaklaşımı kullanılarak geliştirilen CSA varyantları tekmodlu (UM: unimodal), çokmodlu (MM: multimodal) ve ayırık (S: separate) özelliklere sahip 24 fonksiyondan oluşan bir test setine

uygulanmıştır. Ayrıca klasik CSA makalesinde, 10-boyuta sahip çok az sayıda (sadece 5 adet) test fonksiyonu üzerinden algoritmanın performansı değerlendirilmiştir. Bu çalışmada ise 24 adet test fonksiyonu üzerinden 100-boyutlu problemlere hem klasik CSA hem de önerilen algoritmalar uygulanarak daha geniş bir değerlendirme gerçekleştirilmiştir.

2. Klasik CSA

Klasik CSA, kargaların zekâ davranışlarına dayanan yeni sezgisel algoritmalarından biridir. Kargaların farkındalık yaratmak, araçlar kullanmak, yüzleri tanımak gibi istihbarat davranışları onların zeki kuşlardan biri olarak kabul edilmesini sağlamaktadır (Askarzadeh, 2016). Kargaların yiyecek saklama ve kurtarma davranışlarının takliti üzerine geliştirilen CSA algoritması, diğer popülasyon tabanlı arama algoritmaları gibi arama uzayında bireylerin yani kargaların rastgele konumlandırılması ile hesapsal sürece başlar. N adet kargaya sahip ve d -boyutlu bir problem için ilkendirme aşağıdaki gibi tanımlanır:

$$x_{i,G} = [x_{i,G}^1, x_{i,G}^2, \dots, x_{i,G}^d] \quad (1)$$

$$i = 1, 2, \dots, N \quad \text{ve} \quad G = 1, 2, \dots, \text{maxIter}$$

Burada $x_{i,G}$, i . karganın G . nesildeki pozisyonunu temsil eder.

Optimizasyon problemine bağlı olarak amaç fonksiyonuna göre her bir karganın uygunluğu değerlendirilir. Daha sonra her bir karga için hatırlama kabiliyetlerini temsil eden, ziyaret edilen yerlerin konumları Denklem (2) 'deki gibi hafızaya alınır. Burada $M_{i,G}$ i . karganın mevcut iterasyona kadar ziyaret edilen eniyi konumunu temsil eder.

$$M_{i,G} = [m_{i,G}^1, m_{i,G}^2, \dots, m_{i,G}^d] \quad (2)$$

Kargalar yiyecek saklamak için çevrede hareket ederler. CSA 'da kargaların bu davranışı farkındalık olasılığı (AP) ve uçuş uzunluğu (fl) gibi parametreler ile manipüle edilmiş ve aşağıdaki denklemde bir araya getirilmiştir.

$$x_{i,G+1} = \begin{cases} x_{i,G} + r_i \cdot fl \cdot (M_{j,G} - x_{i,G}) & r_i > AP \\ random \quad aksihalde & \end{cases} \quad (3)$$

Burada fl , i . karganın uçuş uzunluğunu, $M_{j,G}$ j . karganın en iyi pozisyonunu, AP farkındalık olasılığını ve r_i ise rastgele sayıyı temsil eder (Askarzadeh, 2016).

Yeni pozisyonun sınır değerleri kontrol edildikten sonra amaç fonksiyonuna göre değerlendirilir. Eğer yeni pozisyon öncekinden daha iyi ise karganın hafızası güncellenir. Gelişim süreci durdurma kriterine ulaşıncaya kadar iterasyonlar boyunca sürdürülür. Final iterasyona ulaşıncaya en iyi sonuç geri döndürülür.

3. Artımsal CSA

Klasik CSA 'da sabit bir popülasyon havuzu kullanılmakta iken bu çalışmada artan bir popülasyon temelinde arama yapan 4 adet ICSA algoritmaları türetilmiştir. Artımsal popülasyon boyutuna sahip bu algoritmalar 4 farklı yeni birey ekleme stratejisini kullanırlar. Türetilen ICSA 'lar klasik CSA 'dan bu noktada farklılık göstermekte olup akış şeması Şekil 1'de gösterilmiştir.

Tablo 1. Popülasyona Yeni Birey Ekleme Kuralları

Durumlar	Türetilen Algoritmalar	Yeni Birey Ekleme Kuralı
Durum-1	ICSA-1	$x_{yeni,G} = x_{eniyei,G}$
Durum-2	ICSA-2	$x_{yeni,G} = x_{rand}$
Durum-3	ICSA-3	$x_{yeni,G} = x_{eniyei,G} + rand * (x_{i,G} - x_{eniyei,G})$
Durum-4	ICSA-4	$x_{yeni,G} = x_{i,G} + rand * (x_{eniyei,G} - x_{i,G})$

Genişleyen bir ajan popülasyonu aslında sosyal öğrenme temeline dayandırılır (Montes De Oca ve Stützle, 2008). Bu stratejiye göre popülasyon tabanlı sezgisel algoritmaların arama boyunca sabit bir popülasyon boyutu yerine adım uzunluğu (SL : step length) gibi belirli aralıklarla popülasyona yeni bireylerin eklenmesi ile hesaplama devam ettirilir. Böylelikle popülasyon boyutu dinamik bir hal alır.

Yeni birey eklenmesi popülasyon içerisinde hızlı bir yakınsamaya ve de çözüm çeşitliliğinin sürdürülebilirliğine olanak sağlamaktadır. Burada dikkat edilecek husus artan popülasyon boyutu ile sabit popülasyon boyutuna sahip metodların aynı şartlarda karşılaştırmalarıdır. Doğru (adil) bir karşılaştırma adına durdurma kriteri olarak iterasyon sayısı yerine fonksiyon çağırım sayısı (FE 's: function evaluation numbers) daha doğru bir yaklaşım olacaktır (Yavuz, 2021; Özyön 2020). Bu çalışmada da durdurma kriteri olarak FE 's kullanılmıştır.

algoritmasına uygulanmaktadır. Yeni birey ekleme işlemi için dört durum belirlenmiştir. Bu dört duruma göre türetilen CSA varyasyonları ve yeni birey ekleme kuralları Tablo 1'de sunulmuştur. Burada, $x_{yeni,G}$ G . jenerasyona eklenecek yeni bireyi (kargayı), $x_{eniyei,G}$ uygunluğu en iyi bireyi, x_{rand} rastgele bireyi, $x_{i,G}$ G . Jenerasyondan rastgele seçilen hedef bireyi ve $rand$ ise $[0,1]$ aralığında rastgele sayıyı temsil eder.

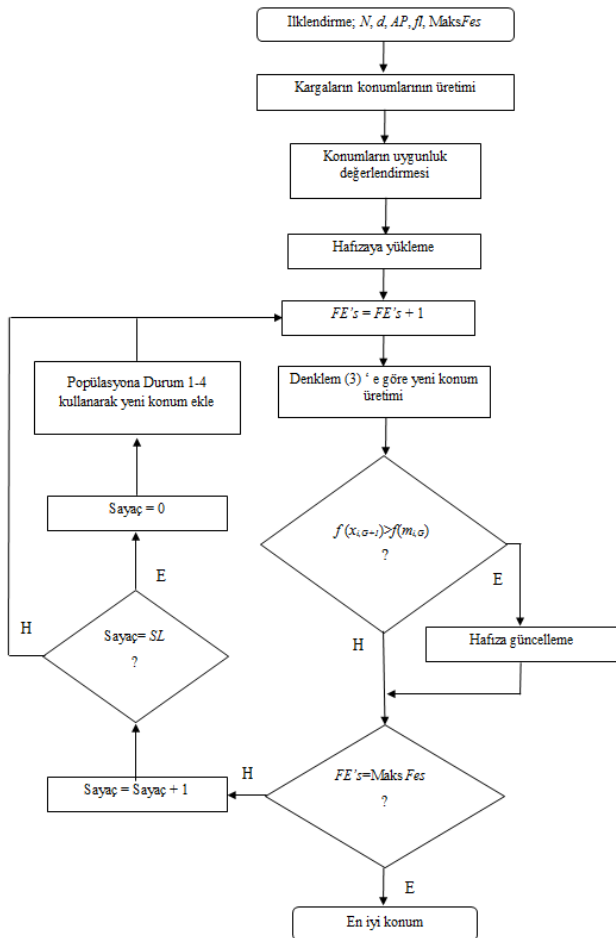
Yeni pozisyonun sınır değerleri kontrol edildikten sonra amaç fonksiyonuna göre değerlendirilir. Eğer yeni pozisyon öncekinden daha iyi ise karganın hafızası güncellenir. Gelişim süreci durdurma kriterine ulaşıncaya kadar iterasyonlar boyunca sürdürülür. Final iterasyona ulaşıncaya en iyi sonuç geri döndürülür.

4. Simülasyon Sonuçları

Önerilen ICSA algoritmalarının performanslarını değerlendirmek için Tablo 2 'de verilen 24 adet test fonksiyonu kullanılmıştır. Bu fonksiyonlar UM, MM ve S olmak üzere farklı karakterlere sahip fonksiyonlardır. Orijinal CSA makalesinde bu fonksiyonlardan sadece f_1 , f_5 , f_{10} , f_{13} ve f_{15} fonksiyonları kullanılmıştır ve boyut olarak sadece 10-boyut seçilmiştir (Askarzadeh, 2016). Bundan dolayı ilk olarak 10-boyutlu bu beş fonksiyona ait orijinal CSA ve önerilen ICSA sonuçları Tablo 3 'de verilmiştir. Öte yandan doğru bir karşılaştırma yapmak için CSA 'nın kullandığı parametre değerleri orijinal makaledeki değerlere bağlı kalınarak $AP=0.1$ ve $fl=0.5$ olarak seçilmiştir. Ayrıca önerilen ICSA algoritmaları dinamik bir popülasyon boyutuna sahip olduğundan önerilen algoritmaları eşit şartlarda yarıştırmaya adına, durdurma kriteri olarak FE 's temelli bir sınırlama getirilmiştir. Orijinal makalede olduğu gibi maksimum FE 's (MaksFes) 40000 olarak belirlenmiştir. Yeni birey ekleme adım uzunluğu $SL=5$ olarak seçilmiştir. Simülasyon çalışmaları Core i7 işlemciye sahip 8-GB RAM donanımlı bir PC 'de yapılmıştır. Her bir fonksiyon için 30 bağımsız çalışma gerçekleştirilerek elde edilen en iyi, ortalama ve standart sapma (S.S.) değerleri tablolarda sunulmuştur.

Tablo 3'deki değerler incelendiğinde önerilen ICSA 'ların klasik CSA 'yı geride bıraktığı görülmektedir. Özellikle MM fonksiyonlar olan f_{13} ve f_{15} 'de klasik CSA lokal optimumlara takılmıştır. ICSA 'lar f_{13} 'de global optimuma ulaşmayı başarmıştır. f_{15} 'de ise ICSA-2 ve ICSA-4 global optimuma ulaşmışlar, diğerleri ise klasik CSA 'ya göre daha düşük hata değerleri üretmişlerdir.

Otuz bağımsız denemeden elde edilen 100-boyutlu test fonksiyonlarına ait hesaplama sonuçları sırasıyla Tablo 4 ve 5 'de sunulmuştur. Elde edilen en iyi sonuçlar koyu renkte vurgulanmıştır. Tablo 4 ve 5 'e göre ICSA-1 algoritması 3 fonksiyonda, ICSA-2 algoritması 4 fonksiyonda, ICSA-3 algoritması 9 fonksiyonda ve ICSA-4 algoritması 4 fonksiyonda diğerlerine göre başarılı olmuşlardır. Diğer fonksiyonlarda ise eşitlik söz konusudur. Önerilen algoritmaların tüm durumlarda klasik CSA 'ya göre genel olarak daha iyi sonuçlar ürettiği görülmektedir.



Şekil 1. ICSA Akış Şematiği

Literatürde mevcut popülasyonun zenginleştirilmesi için farklı yaklaşımlar rapor edilmiştir (Özyön ve ark., 2018). Benzer yaklaşımlar bu çalışmada da kullanılarak CSA

Tablo 2. Test Fonksiyonları

f	UM	MM	S	Fonksiyon Tanımı
f_1	√	-	√	$f_1 = \sum_{i=1}^n x_i^2$
f_2	√	-	-	$f_2 = \sum_{i=1}^n (10^6)^{(i-1)/(n-1)} x_i^2$
f_3	√	-	√	$f_3 = \sum_{i=1}^n ix_i^2$
f_4	-	√	√	$f_4 = \sum_{i=1}^n x_i ^{(i+1)}$
f_5	√	-	-	$f_5 = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $
f_6	√	-	-	$f_6 = \max \{ x_i , 1 \leq i \leq n \}$
f_7	√	-	√	$f_7 = \sum_{i=1}^n (x_i + 0.5)^2$
f_8	√	-	√	$f_8 = \sum_{i=1}^n ix_i^4$
f_9	√	-	√	$f_9 = \sum_{i=1}^n ix_i^4 + random[0,1)$
f_{10}	√	-	-	$f_{10} = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$
f_{11}	-	√	√	$f_{11} = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$
f_{12}	-	√	√	$f_{12} = \sum_{i=1}^n [y_i^2 - 10 \cos(2\pi y_i) + 10]$ $y_i = \begin{cases} x_i & x_i < \frac{1}{2} \\ \frac{round(2x_i)}{2} & x_i \geq \frac{1}{2} \end{cases}$
f_{13}	-	√	-	$f_{13} = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$
f_{14}	√	-	-	$f_{14} = 418.98 * n - \sum_{i=1}^n x_i \sin(\sqrt{ x_i })$
f_{15}	-	√	-	$f_{15} = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos 2\pi x_i\right) + 20 + e$
f_{16}	-	√	-	$f_{16} = \frac{\pi}{n} \left\{ 10 \sin^2(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_n - 1)^2 \right\} + \sum_{i=1}^n u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{1}{4} (x_i + 1) u_{x_i, a, k, m} \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a \leq x_i \leq a \\ k(x_i - a)^m & x_i < -a \end{cases}$
f_{17}	-	√	-	$f_{17} = \frac{1}{10} \left\{ \sin^2(\pi x_1) + \sum_{i=1}^{n-1} (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})] + (x_n - 1)^2 [1 + \sin^2(2\pi x_{i+1})] \right\} + \sum_{i=1}^n u(x_i, 5, 100, 4)$
f_{18}	-	√	√	$f_{18} = \sum_{i=1}^n x_i \sin(x_i) + 0.1x_i $
f_{19}	-	√	-	$f_{19} = \sum_{i=1}^{n-1} (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})] + \sin^2(3\pi x_1) + x_n - 1 [1 + \sin^2(3\pi x_n)]$
f_{20}	-	√	-	$f_{20} = \sum_{i=1}^D \left(\sum_{k=0}^{k_{\max}} [a^k \cos(2\pi b^k (x_i + 0.5))] \right) - D \sum_{k=0}^{k_{\max}} [a^k \cos(2\pi b^k 0.5)] \quad a = 0.5, b = 3, k_{\max} = 20$
f_{21}	-	√	-	$f_{21} = 0.5 + \frac{\sin^2\left(\sqrt{\sum_{i=1}^n x_i^2}\right) - 0.5}{\left(1 + 0.001 * \left[\sum_{i=1}^n x_i^2\right]\right)^2}$
f_{22}	-	√	√	$f_{22} = \frac{1}{n} \sum_{i=1}^n (x_i^4 - 16x_i^2 + 5x_i)$
f_{23}	-	√	√	$f_{23} = -\sum_{i=1}^n \sin(x_i) \sin^{20}\left(\frac{i * x_i^2}{\pi}\right)$
f_{24}	√	-	-	$f_{24} = (x_i - 1)^2 \sum_{i=2}^n i(2x_i^2 - x_{i-1})^2$

Tablo 3. 10-Boyutlu Test Fonksiyonlarına ait Deneysel Sonuçlar

Fonksiyon	İndeks	CSA	ICSA-1	ICSA-2	ICSA-3	ICSA-4
f_1	En iyi	9.54E-13	7.28E-21	1.25E-18	4.09E-11	2.65E-29
	Ortalama	4.09E-11	1.24E-07	1.34E-03	1.28E-06	9.63E-09
	S.S.	6.17E-11	2.48E-07	2.68E-03	1.45E-06	1.92E-08
f_5	En iyi	9.37E-06	9.36E-16	7.97E-14	1.41E-16	6.57E-17
	Ortalama	6.27E-03	1.73E-03	2.40E-03	5.74E-03	1.98E-03
	S.S.	1.99E-02	3.74E-03	4.80E-03	1.04E-02	3.41E-03
f_{10}	En iyi	1.52	0.00E+00	5.95E-26	1.75E-22	4.45E-29
	Ortalama	10.86	2.30E-04	3.81E-04	2.21E-04	4.66E-05
	S.S.	22.76	7.76E-04	1.52E-03	1.00E-03	9.63E-05
f_{13}	En iyi	0.0099	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Ortalama	0.21	2.02E-02	5.37E-02	1.68E-03	1.95E-02
	S.S.	0.12	1.03E-01	1.97E-01	7.37E-03	1.02E-01
f_{15}	En iyi	1.02E-06	6.05E-12	0.00E+00	9.10E-14	0.00E+00
	Ortalama	1.90	3.29E-03	3.29E-03	2.30E-03	4.00E-03
	S.S.	0.79	9.27E-03	9.48E-03	7.08E-03	8.49E-03

Tablo 4. 100-Boyutlu f_1 - f_{10} Test Fonksiyonlarına ait Deneysel Sonuçlar

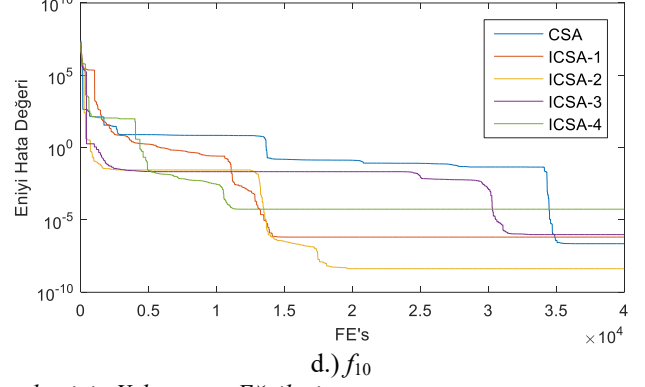
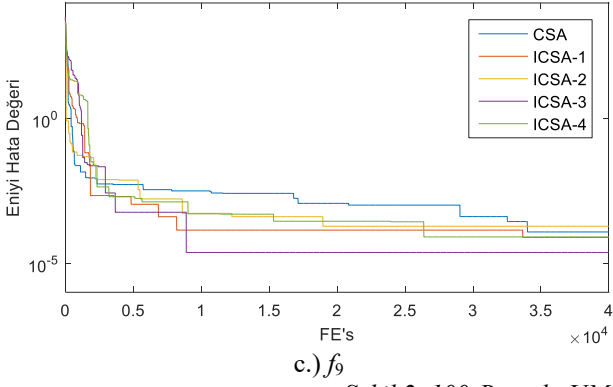
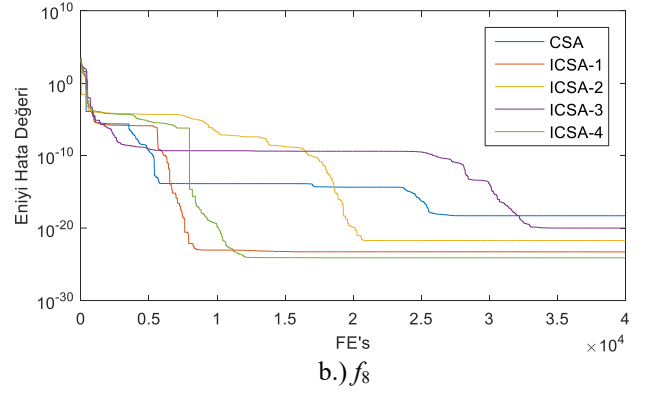
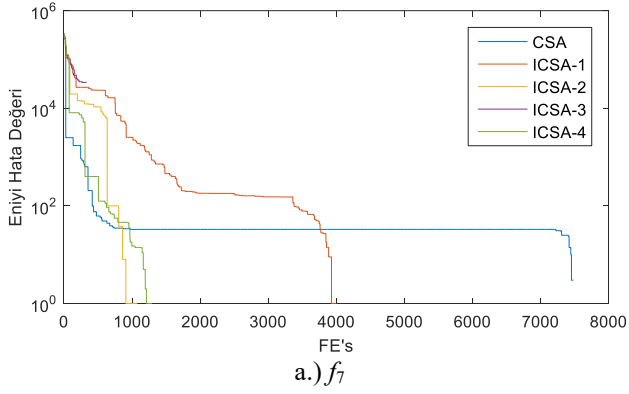
Fonksiyon	İndeks	CSA	ICSA-1	ICSA-2	ICSA-3	ICSA-4
f_1	En iyi	8.62E-05	2.43E-08	1.18E-09	3.68E-11	2.33E-09
	Ortalama	9.43E-03	6.38E-04	1.12E-02	7.17E-03	7.49E-03
	S.S.	1.48E-02	1.24E-03	2.05E-02	9.75E-03	2.05E-02
f_2	En iyi	3.48E-03	1.12E-02	3.48E-10	3.22E-04	4.21E-04
	Ortalama	3.62E+02	5.67E+01	2.25E+01	2.72E+01	6.12E+02
	S.S.	4.85E+02	1.11E+02	3.18E+01	3.54E+01	5.86E+02
f_3	En iyi	8.76E-06	6.60E-08	3.68E-08	9.14E-12	1.42E-11
	Ortalama	1.12E-03	2.15E-03	8.32E-06	2.26E-05	2.48E-06
	S.S.	1.31E-03	2.81E-03	1.23E-05	4.02E-05	3.72E-06
f_4	En iyi	1.27E-03	1.27E-03	1.27E-03	1.27E-03	1.27E-03
	Ortalama	6.65E-02	2.60E-02	1.30E-02	7.59E-03	2.43E-02
	S.S.	1.36E-01	6.08E-02	3.31E-02	1.40E-02	3.54E-02
f_5	En iyi	2.22E-04	1.89E-06	6.00E-03	2.10E-03	3.41E-05
	Ortalama	4.50E-03	1.37E-02	6.56E-02	9.51E-03	2.72E-02
	S.S.	4.94E-03	1.76E-02	6.01E-02	1.31E-02	2.61E-02
f_6	En iyi	1.30E-06	6.07E-04	1.70E-03	1.97E-04	4.72E-07
	Ortalama	6.08E-04	8.29E-03	2.83E-02	2.78E-02	6.21E-03
	S.S.	7.99E-04	6.16E-03	2.98E-02	4.24E-02	6.62E-03
f_7	En iyi	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Ortalama	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	S.S.	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_8	En iyi	5.02E-19	5.12E-24	1.89E-22	9.90E-21	7.93E-25
	Ortalama	6.97E-14	1.98E-11	7.26E-13	1.20E-15	6.41E-11
	S.S.	1.36E-13	3.68E-11	1.36E-12	2.14E-15	1.28E-10
f_9	En iyi	1.21E-04	7.85E-05	1.91E-04	2.38E-05	8.20E-05
	Ortalama	3.87E-04	1.81E-03	5.08E-04	3.19E-04	1.94E-03
	S.S.	2.65E-04	1.66E-03	3.63E-04	2.61E-04	1.51E-03
f_{10}	En iyi	2.28E-07	6.53E-07	4.27E-09	9.71E-07	5.54E-05
	Ortalama	2.68E-02	2.80E-04	1.14E-03	5.98E-03	1.57E-03
	S.S.	5.31E-02	4.10E-04	1.72E-03	1.14E-02	1.42E-03

Tablo 5. 100-Boyutlu f_{11} - f_{24} Test Fonksiyonlarına ait Deneysel Sonuçlar

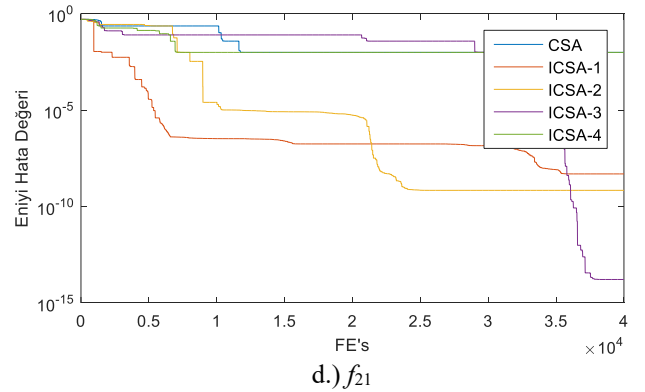
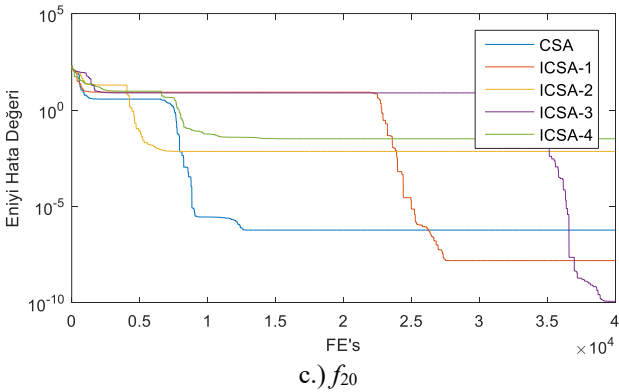
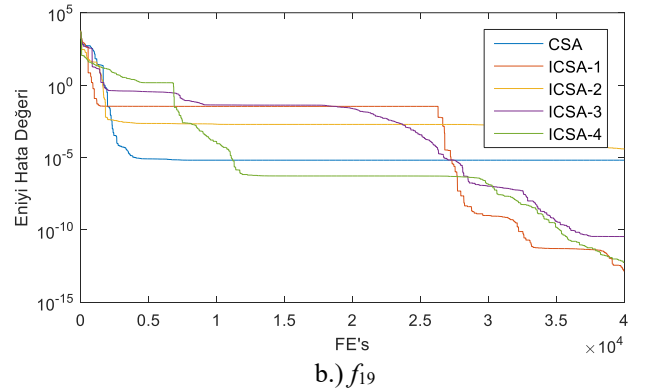
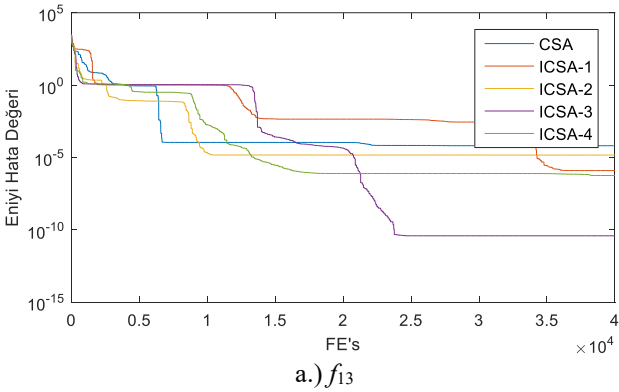
Fonksiyon	İndeks	CSA	ICSA-1	ICSA-2	ICSA-3	ICSA-4
f_{11}	En iyi	1.74E-06	5.98E-07	4.32E-07	1.33E-08	2.53E-06
	Ortalama	7.13E-04	1.97E-03	1.01E-03	5.14E-03	8.45E-04
	S.S.	1.36E-03	2.45E-03	1.27E-03	8.68E-03	8.23E-04
f_{12}	En iyi	1.78E-06	2.66E-04	1.07E-06	3.78E-06	3.16E-08
	Ortalama	2.56E-03	1.37E-02	1.62E-03	4.02E-05	2.10E-04
	S.S.	3.00E-03	2.26E-02	2.94E-03	1.85E-05	3.44E-04
f_{13}	En iyi	6.42E-05	1.22E-06	1.45E-05	3.80E-11	5.58E-07
	Ortalama	1.14E-02	3.73E-04	1.68E-02	2.28E-04	6.21E-04
	S.S.	1.30E-02	7.16E-04	3.31E-02	4.44E-04	7.63E-04
f_{14}	En iyi	1.31E-15	1.23E-13	2.77E-16	3.01E-15	1.51E-15
	Ortalama	1.26E-12	4.91E-12	1.67E-13	1.66E-11	3.53E-10
	S.S.	2.48E-12	7.94E-12	1.48E-13	3.20E-11	7.00E-10
f_{15}	En iyi	1.40E-04	4.51E-06	3.28E-06	1.38E-05	3.21E-04
	Ortalama	1.08E-02	7.51E-03	4.75E-03	4.25E-03	1.34E-02
	S.S.	1.65E-02	1.29E-02	5.32E-03	4.56E-03	1.97E-02
f_{16}	En iyi	4.47E-09	3.93E-11	4.49E-12	4.52E-12	4.77E-13
	Ortalama	4.71E-06	3.55E-06	7.51E-07	1.53E-05	1.65E-05
	S.S.	7.60E-06	5.66E-06	1.46E-06	3.29E-05	3.71E-05
f_{17}	En iyi	3.77E-07	2.38E-12	2.06E-08	2.41E-08	1.70E-10
	Ortalama	4.06E-04	1.70E-05	1.61E-03	4.06E-04	8.80E-04
	S.S.	7.92E-04	2.27E-05	2.21E-03	5.56E-04	1.17E-03
f_{18}	En iyi	1.49E-05	3.96E-07	6.39E-05	4.41E-10	4.82E-05
	Ortalama	3.29E-03	2.97E-04	2.04E-03	1.81E-04	1.33E-03
	S.S.	4.31E-03	2.68E-04	1.07E-03	2.08E-04	1.09E-03
f_{19}	En iyi	6.42E-06	1.41E-13	3.76E-05	3.39E-11	5.46E-13
	Ortalama	1.44E-03	1.11E-03	4.11E-03	5.18E-04	1.62E-04
	S.S.	2.84E-03	9.69E-04	4.86E-03	9.52E-04	3.01E-04
f_{20}	En iyi	5.93E-07	1.56E-08	7.11E-03	1.20E-10	3.22E-02
	Ortalama	6.80E-01	9.79E-01	5.48E-01	1.03E+00	2.35E+00
	S.S.	8.68E-01	1.21E+00	6.54E-01	2.06E+00	2.61E+00
f_{21}	En iyi	9.71E-03	4.87E-09	6.84E-10	4.63E-14	9.71E-03
	Ortalama	3.79E-02	1.88E-02	2.65E-02	1.55E-02	2.62E-02
	S.S.	3.21E-02	1.53E-02	2.22E-02	2.10E-02	1.34E-02
f_{22}	En iyi	-7.83E+01	-7.83E+01	-7.83E+01	-7.83E+01	-7.83E+01
	Ortalama	-7.83E+01	-7.83E+01	-7.83E+01	-7.83E+01	-7.83E+01
	S.S.	3.11E-06	1.05E-06	9.55E-07	7.45E-07	3.71E-05
f_{23}	En iyi	6.91E-16	6.13E-16	7.74E-16	0.00E+00	8.07E-16
	Ortalama	1.06E-15	1.00E-15	1.05E-15	9.01E-16	1.04E-15
	S.S.	2.21E-16	2.18E-16	2.00E-16	4.97E-16	2.67E-16
f_{24}	En iyi	2.49E-01	2.49E-01	2.49E-01	2.49E-01	2.49E-01
	Ortalama	2.90E-01	2.62E-01	2.49E-01	2.50E-01	2.50E-01
	S.S.	8.04E-02	2.58E-02	2.66E-05	2.33E-04	3.13E-04

UM fonksiyonlarda, algoritmanın yakınsama hızı onun global arama kabiliyetini gösterdiği için önemlidir (Yao, Liu ve Lin, 1999). Bu durumu incelemek üzere f_7 - f_{10} fonksiyonlarının yakınsama eğrileri Şekil 2 'de gösterilmiştir. Bu şekil incelendiğinde önerilen ICESA algoritmalarının daha düşük FE 's değerlerinde global optimum komşuluklarına yakınsadığı görülmektedir. Bu sonuç, geliştirilen algoritmaların hızlı bir yakınsama oranına sahip olduğu şeklinde yorumlanabilir. Öte yandan, MM fonksiyonlarda boyutun artması lokal optimum sayısını arttıracığından, algoritmaların optimuma takılma

tehlikesi kaçınılmazdır. Bu durumu gözlemlemek için örnek olarak f_{13} , f_{19} , f_{20} ve f_{21} fonksiyonlarına ait yakınsama eğrileri Şekil 3 'de verilmiştir. Bu şekle göre f_{13} fonksiyonunda klasik CSA 'nın lokal optimuma takıldığı ve hata değerinin 6.42E-5 altına düşemediği görülürken, önerilen ICESA-3 'nın 3.80E-11 seviyelerine ulaştığı izlenmektedir. Benzer durum diğer MM fonksiyonlar için de geçerli olup, genel olarak önerilen ICESA 'ların daha düşük değerlere ulaştığı gözlemlenmektedir.



Şekil 2. 100-Boyutlu UM fonksiyonlar için Yakınsama Eğrileri



Şekil 3. 100-Boyutlu MM Fonksiyonlar için Yakınsama Eğrileri

Algoritmaların performans karşılaştırmasında kullanılan diğer bir araç ise istatistiksel karşılaştırmalardır. Algoritmaların birbirlerine göre üstünlüklerini ya da benzerliklerini belirleyen bu yöntemlerden en yaygın olanlarında biride Wilcoxon testidir. Bu teste göre iki algoritma arasındaki anlamlılık derecesi (significance level) bağımsız parçalara göre belirlenmeye

çalışılır. Buna göre; p -değeri iki metot arasındaki önem derecesinin göstergesidir (Derrac ve ark., 2011).

Bu çalışmada, önerilen ICSA 'lar ve klasik CSA arasındaki karşılaştırmalar için Wilcoxon Signrank testi kullanılmıştır. Signrank testi için 100-boyutlu fonksiyonlardan elde edilen en iyi sonuç değerleri kullanılmıştır. İstatistiksel karşılaştırma

sonuçları Tablo 6 'da verilmiştir. MM fonksiyonlarda, ICSA-1 ve ICSA-3 algoritmaları $\alpha = 0.05$ anlamlılık seviyesine göre klasik CSA 'da önemli bir gelişme sağlamışlardır.

Tablo 6. Wilcoxon Testi p-değer Sonuçları

Karşılaştırma	$f_1 - f_{10}$	$f_{11} - f_{24}$
CSA ile ICSA-1	0.9453	0.0425
CSA ile ICSA-2	0.9800	0.3804
CSA ile ICSA-3	0.7422	0.0161
CSA ile ICSA-4	0.0781	0.7646

5. Sonuçlar ve Tartışma

Bu çalışmada, yüksek boyutlu problemlerin çözümü için ICSA versiyonları türetilmiştir. Temel CSA 'nın aksine IPOP stratejisi kullanan bu CSA versiyonlarında, popülasyona yeni bireyin eklenmesi hesapsal süreçte genetik çeşitliliğin sürdürülmesine yardımcı olmaktadır. Böylelikle birden fazla lokal optimum içeren MM problemlerin çözümünde daha iyi sonuçlar elde edilebilir. Geliştirilen metotların performansı yüksek boyutlu test fonksiyonlarında izlenmiştir. Elde edilen sonuçlar önerilen metotların CSA 'nın erken yakınsama problemini iyileştirdiğini göstermektedir. Ayrıca geliştirilen ICSA 'ların MM fonksiyonlarda genetik çeşitliliği zenginleştirdiği ve bunun sonucunda da çoğu durumda lokal optimumlardan kaçabildiği görülmüştür. Gelecek çalışmalarda geliştirilen metotların gerçek-dünya problemlerine uygulanması ve IPOP stratejisinin diğer meta-sezgisellere uyarlanması hedeflenmektedir.

Kaynakça

Abdollahi, M., Bouyer, A. & Abdollahi, D. (2016). Improved cuckoo optimization algorithm for solving systems of nonlinear equations. *The Journal of Supercomputing*, 72, 1246-1269.

Askarzadeh, A (2016). A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm. *Computer & Structures*, 169, 1-12.

Aydın, D. & Özyön, S. (2013). Incremental artificial bee colony with local search to economic dispatch problem with ramp rate limits and prohibited operating zones. *Energy Conversion and Management*, 65, 397-407.

Aydın, D., Yavuz, G. & Stützle, T. (2017). ABC-X: a generalized, automatically configurable artificial bee colony framework. *Swarm Intelligence*, 11(1), 1-38.

Derrac, J., García, S., Molina, D., & Herrera, F. (2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, 1(1), 3-18.

Dorigo, M. & Di Caro, G. (1999). *The ant colony optimization metaheuristic, new ideas in optimization*. McGraw-Hill, New York, pp 11-32.

Gao, W. & Liu, S. (2011). Improved artificial bee colony algorithm for global optimization. *Information Processing Letters*, 111(17), 871-882.

Karaboğa, D. & Baştürk, B. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee

colony (ABC) algorithm. *Journal Global Optimization*, 39(3), 459-471.

Kaur, G. & Arora, S. (2018). Chaotic whale optimization algorithm. *Journal of Computational Design and Engineering*, 5(3), 275-284.

Kennedy, J. & Eberhart, R. (1995, November). Particle swarm optimization. In 1995 IEEE International Conference on Neural Networks, 4, 1942-1948.

Li, S. Y., Wang, S. M., Wang, P. F., Su, X. L., Zhang, X. S. & Dong, Z. H. (2018). An improved grey wolf optimizer algorithm for the inversion of geoelectrical data. *Acta Geophysica*, 66, 607-621.

Mirjalili, S., Mirjalili, S. M. & Lewis, A. (2014). Grey wolf optimizer. *Advance Engineering Software*, 69, 46-61.

Mirjalili, S. & Lewis, A. (2016). The whale optimization algorithm. *Advance Engineering Software*, 95, 51-67.

Montes de Oca, M. A. & Stützle, T. (2008, July). Towards incremental social learning in optimization and multiagent systems. In 10th Annual Conference Companion on Genetic and Evolutionary Computation (GECCO'08), New York, 1939-1944.

Özyön, S., Yaşar, C. & Temurtaş, H. (2019). Incremental gravitational search algorithm for high-dimensional benchmark functions. *Neural Computing and Applications*, 31, 3779-3803.

Özyön, S. (2020). Yenilenebilir enerji üretim birimleri içeren çevresel-ekonomik güç dağıtım probleminin yüklü sistem arama algoritması ile çözümü. *Avrupa Bilim ve Teknoloji Dergisi*, 18, 81-90.

Rajabioun, R. (2011). Cuckoo optimization algorithm. *Applied Soft Computing*, 11, 5508-5518.

Sahoo, A. & Chandra, S. (2017). Multi-objective grey wolf optimizer for improved cervix lesion classification. *Applied Soft Computing*, 52, 64-80.

Saidala, R. K. & Devarakonda, N. (2018). Improved whale optimization algorithm case study: clinical data of anaemic pregnant woman. *Data Engineering and Intelligent Computing*, 542, 271-281.

Xu, X., Rong, H. & Trovati, M. (2018). CS-PSO: chaotic particle swarm optimization algorithm for solving combinatorial optimization problems. *Soft Computing*, 22(3), 783-795.

Yang, J. & Zhuang, Y. (2010). An improved ant colony optimization algorithm for solving a complex combinatorial optimization problem. *Applied Soft Computing*, 10(2), 653-660.

Yao, X., Liu, Y. & Lin, G. (1999). Evolutionary programming made faster. *IEEE Transaction on Evolutionary Computation*, 3(2), 82-102.

Yavuz, G. (2021). 100 Basamak probleminin JADE algoritması ile çözümü. *Avrupa Bilim ve Teknoloji Dergisi*, 21, 493-500.

Yu, W., Li, X., Cai, H., Zeng, Z. & Li, X. (2018). An improved artificial bee colony algorithm based on factor library and dynamic search balance. *Mathematical Problems in Engineering*, 3102628, 1-16.

Zhang, Q. & Zhang, C. (2018). An improved ant colony optimization algorithm with strengthened pheromone updating mechanism for constraint satisfaction problem. *Neural Computing and Applications*, 30, 3209-3220.