


L-shade Algoritmasının Otomatik Parametre Yapılandırma Yöntemi ile İyileştirilmesi*

Araştırma Makalesi/Research Article

 Gürcan YAVUZ*

Bilgisayar Mühendisliği, Kütahya Dumlupınar Üniversitesi, Kütahya, Türkiye

gurcan.yavuz@dpu.edu.tr

(Geliş/Received:09.12.2021; Kabul/Accepted:16.03.2022)

DOI: 10.17671/gazibtd.1034921

Özet— Son dönemin önemli metasezgisellerinden olan L-shade algoritması uyarlanabilir bir DE varyantıdır. Az sayıda kontrol parametresine sahiptir ve algoritmanın çalışmasını etkilemektedir. Bunların doğru şekilde belirlenmesi algoritma performansında kritik role sahiptir. Bu çalışmada, L-shade algoritmasına ait kontrol parametreleri bir otomatik parametre yapılandırma aracı olan irace kullanılarak belirlenmiştir. Yapılan işlemin etkinliği CEC 2014 ölçüt seti kullanılarak test edilmiştir. Elde edilen sonuçlar, L-shade'nin ön tanımlı parametreleri ve bir başka parametre belirleme aracı olan paramils'nin elde ettiği parametreler ile karşılaştırılmıştır. Deney sonuçları göstermiştir ki kullanılan yapılandırma aracı ile elde edilen parametre değerleri ile daha iyi sonuçlar elde edildiği görülmüştür.

Anahtar Kelimeler— l-shade, optimizasyon, otomatik parametre yapılandırma

Improvement of L-shade Algorithm with Automatic Parameter Configuration Method *

Abstract— The L-shade algorithm, one of the important meta-heuristics of the last decade, is an adaptive DE variant. It has few control parameters and affects the execution of the algorithm. Determining these correctly has a critical role in algorithm performance. In this study, the control parameters of the L-shade algorithm are determined using irace, an automatic parameter configuration tool. The efficiency of the operation was tested using the CEC 2014 criteria set. The results obtained are compared with the predefined parameters of the L-shade and the parameters obtained by paramils, another parameter determination tool. The results of the experiment showed that better results were obtained with the parameter values obtained with the configuration tool.

Keywords— automatic parameter configuration, l-shade, optimization

1. GİRİŞ (INTRODUCTION)

Mühendislik ve bilgisayar bilimlerine ait birçok problem sürekli optimizasyon problemleri kategorisine girmektedir. Bu problemlerin çözümünde son yıllarda metasezgisel algoritmalar yoğun bir şekilde kullanılmaktadır[1-4]. Bunların başında Diferansiyel Gelişim (Differential Evolution, DE) algoritması ve bu algoritmanın varyantları gelmektedir [1].

DE algoritması, 1995 yılında Storn ve Price tarafından geliştirilmiş popülasyon tabanlı bir algoritmadır [5]. Bu

algoritmanın kolay kodlanabilir oluşu ve gösterdiği üstün performans araştırmacıların ilgi odağı haline gelmesini sağlamıştır. “No-Free-Lunch (NFL)” teoremine göre tüm optimizasyon problemleri için tek bir algoritma olmayacağından algoritma geliştiricileri farklı problem türlerini çözmek için DE üzerinde çeşitli iyileştirmeler gerçekleştirmişler [6-8]. Bu iyileştirmelerin başında çeşitli popülasyon tabanlı algoritmalarda da kullanılan kontrol parametrelerinin uyarlanabilir olarak belirleme yöntemi gelmektedir [9,10]. DE varyantları içinde buna en güzel örnek ise Doğrusal Popülasyon Azaltma Stratejisini Kullanan Başarı Geçmişine Dayalı

Uyarlanabilir DE (Success-History based Adaptive DE using Linear Population Size Reduction, L-shade) algoritmasıdır [11].

IEEE Evrimsel Hesaplama konferansının (CEC) gerçek parametre optimizasyonu için düzenlenen özel oturumuna ait ölçüt setleri, sürekli optimizasyon algoritmalarının karşılaştırılmasında bir standart haline almıştır. Her yıl düzenlenen bu yarışmada katılan algoritmalar arasında en iyi olan algoritmalar belirlenmektedir. Son yıllarda bu yarışmaya L-shade damgasını vurmuştur. L-shade ve bu algoritmanın varyantları CEC yarışmalarını ilk sıralarda tamamlamışlardır [12,13].

Optimizasyon algoritmaları dışarıdan ya da algoritma içerisinde değerleri belirlenen kontrol parametrelerine sahiptir ve bunların algoritma performansı üzerinde etkileri oldukça fazladır [14]. Literatürde optimizasyon algoritmalarının geliştirilme aşamasında belirlenen kontrol parametrelerinin kullanılmasıyla algoritmanın sahip olduğu performansı göstermediği görülmüştür. Bu parametrelerin ince ayarlar yapılarak daha da iyileştirildiği ve algoritmaların daha iyi performans verdikleri görülmektedir. Elhosseini vd., Fil Sürü Optimizasyon (elephant herding optimization, EHO) algoritmasının parametrelerini çeşitli yöntemler ile belirleyerek EHO'nun performansını daha da yukarıya taşımışlardır [15]. Liao vd. Artan Popülasyon Boyutlu CMA-ES (CMA-ES with increasing population size, iCMA-ES) algoritmasını otomatik parametre yöntemlerini kullanarak belirlediği parametreler ile algoritmanın performansını geliştirmişlerdir [16]. Liao vd. otomatik parametre yapılandırmanın, optimizasyon algoritmalarının performansını artırdığını yedi farklı algoritmayı deneylerde kullanarak test etmişler ve algoritmaların performansını geliştirdiği sonucuna varmışlar [17]. Aydın vd., dokuz farklı Yapay Arı Kolonisi (artificial bee colony, ABC) varyantının kontrol parametrelerini otomatik yapılandırıcılar kullanılarak belirlemişlerdir [18]. Belirlenen yeni parametreler ile çalıştırılan ABC varyantlarının sonuçları, ön tanımlı parametreler kullanılarak çalıştırılan varyantların sonuçlarından daha iyi olduğunu göstermiştir.

L-shade algoritması, başarımını etkileyen dört adet kontrol parametresine sahiptir. Bunlara verilen değerler algoritmanın geliştirildiği ölçüt seti CEC 2014 kullanılarak belirlenmiştir. Bu çalışmada, L-shade algoritmasının iç yapısında herhangi bir değişiklik yapılmadan sadece otomatik parametre belirleme aracı irace ile L-shade parametreleri belirlenerek CEC 2014 ölçüt setindeki performansının iyileştirilmesi gerçekleştirilmiştir.

2. L-SHADE ALGORİTMASI (L-SHADE ALGORITHM)

L-shade, Tanabe ve Fukunaga tarafından önerilmiş SHADE algoritmasının yine aynı araştırmacıların popülasyon azaltma stratejisi kazandırılmış yeni halidir. Bu algoritma, 2014 yılında düzenlenen Evrimsel Hesaplama Kongresindeki Gerçek Parametre

Optimizasyonun özel oturumunda birinciliği elde etmiştir [19]. Devamında gerçekleştirilen Evrimsel Hesaplama kongrelerinde varyantları yine ilk sıralarda yer almıştır [12,13].

L-shade, uyarlanabilir bir DE varyantıdır. Orijinal DE algoritmasının arama ve problem çözme yeteneklerini geliştirme için yeni stratejiler önermiştir. DE algoritmasında popülasyon boyutu NP, mutasyon ölçeklendirme faktörü F ve çaprazlama oranı CR algoritma çalışması süresince değerleri sabit olarak kalmaktadır. Bu da DE algoritmasının yerel optimuma takılmasına sebep olmaktadır. Bu yetersizliği iyileştirmek için L-shade algoritmasında parametre güncelleme stratejileri önerilmiştir. Buna göre popülasyondaki her birey için ayrı F ve CR atanmakta ve bunların değerleri algoritma çalışması süresince güncellenmektedir. Bu güncelleme işlemi, algoritmanın başlangıcından itibaren başarılı kabul edilen bireylere ait F ve CR değerleri tanımlanmış kümelerle kaydedilmektedir. Daha sonra mutasyon ve çaprazlama adımında kullanılacak olan CR ve F parametreleri bu kümedeki değerler kullanılarak üretilmektedir.

Orijinal DE algoritmanın performansına etki eden faktörlerden olan bir diğer durum ise algoritma çalışma süresince popülasyon boyutunun sabit kalmasıdır. Algoritma iyi bir çözüm bulamasa da aynı popülasyon boyutu ile çalışmaya devam etmektedir. Bunun sebebi arama uzayı büyük kalmakta ve algoritma iyi çözümlere ulaşamamaktadır. L-shade, bu yetersizliği gidermek için doğrusal olarak azalan popülasyon boyut stratejisini (Linear population size reduction, LPSR) önermiştir. Algoritma isminin başında yer alan "L" ifadesi buna atf yapmaktadır. Algoritma çok sayıda birey içeren büyük bir popülasyon ile optimizasyon sürecine başlar. Bunun sayesinde başlangıçta algoritma arama uzayının her tarafında birey oluşturarak iyi çözümleri aramaktadır. İterasyonlar ilerledikçe algoritma daha iyi çözümler ile çalışacağı için bu çözümlerden daha fazla faydalanabilmek adına popülasyon boyutu azaltılmaktadır. Böylelikle algoritmanın iyi çözümlere ulaşması sağlanır. Bu işlem, az sayıda çözüm kalıncaya kadar devam eder ve popülasyon boyutu azaltılır. Algoritma, NP = 18 x D kadarlık bir popülasyon boyutu ile çalışmaya başlar. D burada problem boyutunu göstermektedir. İterasyonlar ilerledikçe popülasyonu boyutu değerine kadar azaltılmaktadır. Şekil 1'de L-Shade algoritmasının sözde koduna yer verilmiştir [11,20]. Algoritma detayları şöyledir:

Algoritma, beş adet küme içermektedir. Bunlar; A, MF, MCR, SF ve SCR kümeleridir. Bunlardan ilki olan A kümesi, popülasyon çeşitliliğini sağlamak için bireyleri tutmaktadır. Başlangıçta herhangi bir eleman içermeyen bu kümeye ekleme işlemi, algoritmanın seçme adımında yapılmaktadır. Seçme adımında mevcut birey vektör, deneme vektöründen daha kötü ise mevcut vektörün bilgilerinin kaybolmaması için A kümesine dahil edilir. A kümesinin boyutu en fazla $|A| = r^{arc} \times NP_g$ (r^{arc} , L-

shade algoritmasına ait bir kontrol parametresidir.) kadar olabilmektedir. SF ve SCR kümelerinde popülasyondaki her birey vektörü için ayrı ayrı F ve CR değerleri saklanmaktadır. Her iterasyonda bu kümeler boşaltılmaktadır. MF ve MCR kümeleri, SCR ve SF kümelerindeki elemanlar kullanılarak ağırlıklı Lehmer ortalaması ile doldurulmaktadır [11]. Kümedeki elemanların başlangıçtaki ilk değerleri 0.5'tir.

```

1:  $G = 1, N_G = NP_{init}$ 
2: A, MF, MCR, SF, SCR kümelerini oluştur
3:  $A = \{\}, NP, H, p$  ve  $r^{arc}$  ilk değerlerini ata
4:  $P_g$  popülasyonunu ( $P_g = x_{1,G}, \dots, x_{N,G}$ ) rastgele üret.
5:
6: MF ve MCR bütün değerleri 0.5 olarak ata.
7: while sonlanma kriterleri sağlanmadığı sürece do
8:   for  $i = 1: Np$ 
9:     Denklem 2 ve 3 kullanarak  $CR_i$  ve  $F_i$  üret
10:    mutasyon işlemi uygula
11:    parametre uyarlama stratejisini uygula
12:    çaprazlama işlemi uygula
13:    seçme işlemi uygula
14:    if  $f(u_i) \leq f(x_i)$ 
15:       $A \leftarrow x_i^G, SCR \leftarrow CR_i, SF \leftarrow F_i$ 
16:    doğrusal popülasyon azaltma stratejisi uygula
17:    MF ve MCR değerlerini ağırlıklı lehman ortalama ile hesapla
18:  $G = G + 1$ 

```

Şekil 1. L-shade algoritmasının sözde kodu
(Pseudocode of the l-shade algorithm)

L-shade, çalışmaya ilklendirme adımı ile başlamaktadır. İlklendirme adımında problemin sınır değerleri dikkate alınarak bir P popülasyonu rastgele üretilir. Popülasyondaki birey vektörleri denklem (1) ile üretilir:

$$x_{i,j} = x_{\min,j} + \text{rand}_{i,j}[0,1] \times (x_{\max,j} - x_{\min,j}) \quad (1)$$

Denklemden yer alan $\text{rand}_{i,j}$, [0,1] aralıktan uniform dağılım kullanılarak rastgele üretilen bir sayıdır. $x_{\min,j}$ ve $x_{\max,j}$ sırasıyla belirlenmeye çalışılan parametrelerin alt ve üst sınırlarını göstermektedir.

İlklendirme adımından sonra algoritma, belirlenen maksimum değerlendirme sayısı tamamlanmaya kadar sırasıyla birbirini takip eden mutasyon, çaprazlama, seçme adımı ve doğrusal popülasyon azaltma adımı icra eder. Bu adımların detayları aşağıda verilecektir.

İterasyonlara başlamadan önce algoritma her birey için F_i ve CR_i değerlerini üretmektedir. Bu iki parametrenin değeri Denklem (2) ve (3) belirlenmektedir

$$F_i = \text{randc}(MF_{ri}, 0.1) \quad (2)$$

$$CR_i = \text{randn}(MCR_{ri}, 0.1) \quad (3)$$

Denklem (2) ve (3)'de yer alan $\text{randc}()$ ve $\text{randn}()$ Cauchy ve Normal dağılımı kullanarak rastgele üretilen değerleri göstermektedir. 0.1, Cauchy dağılımın ölçeklendirme parametresi ve Normal dağılımın varyans değeridir. Burada yer alana r_i değeri [1, H] aralığından üretilen rastgele sayıyı göstermektedir. H, MF ve MCR kümelerinin boyutunu göstermektedir ve algoritma başında tanımlanır

Mutasyon adımında, popülasyondaki her bir bireyin bilgileri kullanılarak bir mutant vektörü üretilir. Bu üretim işlemi, L-shade algoritmasında 'current-to-pbest/1' olarak bilinen mutasyon stratejisi kullanılarak gerçekleştirilir. Bu stratejinin mutant vektör üretim denklemi Denklem (4)'de verilmiştir:

$$v_i = x_i + F_i \times (x_{pbest} - x_i) + F_i \times (x_{r1} - x_{r2}) \quad (4)$$

v_i , iterasyondaki i. bireyin mutant vektörünü göstermektedir. $i \neq r1 \neq r2$ olmak üzere; x_{r1} , P popülasyonundan rastgele seçilmiş bir vektörü göstermektedir. x_{r2} ise, A kümesi ve mevcut popülasyon P'nin birleşiminin oluşturduğu kümeden ($A \cup P$) rastgele seçilmiş olan bireyi göstermektedir. x_{pbest} , popülasyondaki en iyi %p birey içerisinde rastgele seçilmiş olan bireyi göstermektedir.

Çaprazlama adımında, orijinal DE algoritmasında olduğu gibi mevcut birey ile mutant vektörün bilgileri kullanılarak bir deneme vektörü $u_{i,j}$ oluşturulmaktadır. Orijinal DE algoritmasından farklı olarak L-shade her birey için ayrı bir CR_i değeri kullanmaktadır. Bu değer de daha önce anlatıldığı gibi denklem (3) kullanılarak üretilmektedir.

$$u_{i,j} = \begin{cases} v_{i,j} & \text{if } j = j_{rand} \text{ or } \text{rand}_{i,j}[0,1] \leq CR_i \\ x_{i,j} & \text{otherwise} \end{cases} \quad (5)$$

Burada j_{rand} , [0, d] aralığından rastgele üretilmiş bir sayıdır.

Seçme adımında algoritma, üretmiş olduğu deneme vektörünün iyi olup olmadığını kontrol etmektedir. Bunun için deneme vektörünün uygunluk değeri ile mevcut bireyin uygunluk değeri karşılaştırılır. Mutant vektörünün uygunluk değeri daha iyi ise mevcut birey mutant vektör ile değiştirilir. Aksi durumda yani mevcut bireyin uygunluk değeri daha iyi ise algoritma mevcut birey ile yoluna devam eder.

$$x_{i,j} = \begin{cases} u_{i,j} & \text{if } f(u_i) \leq f(x_i) \\ x_{i,j} & \text{otherwise} \end{cases} \quad (6)$$

L-shade algoritmasında yer alan son adım ise doğrusal popülasyon boyut azaltılma adımıdır. Orijinal DE

algoritmasından farklı olarak L-shade algoritmasının popülasyon boyutu algoritma çalışması süresince aynı kalmamaktadır. L-shade, büyük bir popülasyon ile çalışmaya başlar ve popülasyonun boyutu iterasyonlar boyunca doğrusal olarak azaltılmaktadır. Her iterasyonda popülasyon boyutu Denklem (7) ile belirlenir. Popülasyondan, yeni popülasyon boyutu ile eski popülasyon boyutu farkı kadar kötü olarak tanımlanan bireyler çıkartılır. Bu sayede algoritmanın hep iyi bireyler ile çalışması sağlanır.

$$Np(G+1) = \text{round}\left[\left(\frac{Np_{\min} - Np_{\text{init}}}{NFE_{\max}}\right)\right] \times NFE + Np_{\text{init}} \quad (7)$$

NP_{\min} , en küçük popülasyon boyutunu, NP_{init} başlangıç popülasyonun boyutunu göstermektedir ve $NP_{\min} = 4$ 'dür. NFE ve NFE_{\max} , sırasıyla mevcut iterasyondaki fonksiyon çağrım sayısını ve en büyük fonksiyon çağrım sayısını temsil etmektedir.

3. OTOMATİK PARAMETRE YAPILANDIRMA (AUTOMATIC PARAMETER CONFIGURATION)

Literatürde yer alan araştırmalara göre, optimizasyon algoritmalarının performansları sahip oldukları kontrol parametrelerine bağlıdır [18]. Bu parametreler için uygun değerlerin seçilmesi büyük önem arz etmektedir. Araştırmacılar bu değerleri bulmak için geleneksel olarak parametrelerin teker teker denemesi yolunu tercih etmektedir [21]. Bu yaklaşım deneme yanılma ile olmaktadır ve hataya duyarlı zaman alıcı bir işittir. Ayrıca değerleri belirleyen kişiye de duyarlı bir süreçtir. Bu işlemi kolaylaştırmak için otomatik parametre yapılandırma yöntemleri önerilmiştir. Bunlardan bazıları; Parametre Yinelemeli Yerel Arama (Parameter Iterated Local Search, paramils) [22], Cinsiyete Dayalı Genetik Algoritma (Gender based Genetic Algorithm, GGA) [23], Sıralı Model Tabanlı Algoritma yapılandırması (Sequential model-based algorithm configuration, SMAC)[24], iterated F-Race [25].

L-shade kontrol parametrelerinin yapılandırılması için iterated F-Race yöntemi kullanılmıştır [25]. Iterated F-Race, F-Race yöntemini tekrarlı şekilde uygulayan bir yöntemdir. F-Race, aday algoritma yapılandırmaları arasında en iyi algoritma yapılandırmasını seçmede kullanılan bir yarışma metodudur [17]. Yarışma iteratif olarak uygulanmaktadır. F-Race, çalışmaya oluşturduğu aday başlangıç yapılandırmaları ile başlamaktadır. Bu yapılandırmalar, F-Race yöntemine girdi olarak verilmiş olan problem setine uygulanır. Yapılandırmaların elde ettiği sonuçlara bir takım istatistiksel testler uygulanarak iyi ve kötü yapılandırmalar belirlenir. İyi olan aday yapılandırma ayrılır ve kötü olan aday yapılandırmalar elenir. Böylelikle bir yarışmanın bir turu tamamlanmış olur. Sonraki yarışma turu için mevcut turun en iyi aday yapılandırma bilgisi kullanılarak bir sonraki iterasyonda yarışmaya katılacak olan yeni aday yapılandırmalar üretilir. Bu işlem, F-Race yönteminde tanımlanan çalışma bütçesine kadar devam eder. Bütçe tamamlandığında

yarışmanın galibi olan en iyi aday yapılandırma, çıktı olarak verilir. Deneylerde iterated F-Race yöntemini bir R paketi olarak kullanıcılara sağlayan irace kütüphanesinden faydalanılmıştır [26].

4. DENEYSEL SONUÇLAR (EXPERIMENTAL RESULTS)

Deneylerde, L-shade algoritma yazarlarının sağlamış oldukları c++ kodundan faydalanılmıştır [11]. Elde edilen algoritmanın kodu dışarıdan parametre alacak şekilde yapılandırılmıştır.

4.1. Deney Ortamı (Test Environment)

Deneyler için 30 adet probleme sahip olan CEC 2014 ölçüt seti seçilmiştir. Ölçüt setine ait fonksiyon isimleri ve fonksiyonlara ait bilgiler Tablo 1'de verilmiştir. Bu ölçüt seti; Unimodal, Multimodal, Hibrit ve Kompozit olmak üzere dört farklı türde fonksiyona sahiptir. Bu ölçüt setinin çözümünde Liang vd. belirtmiş oldukları kriterler dikkate alınmıştır [19]. Bu kriterler kısaca şunlardır: test fonksiyonları 10, 30, 50 ve 100 boyut için çözülmektedir..

Tablo 1. Cec 2014 fonksiyonlar ve özellikleri (Functions and features in the cec 2014 benchmark set)

| Tür | Id | Fonksiyonlar |
|------------|--------|--|
| Unimodal | f1 | Rotated high conditioned elliptic function |
| | f2 | Rotated bent cigar function |
| | f3 | Rotated discus function |
| Multimodal | f4 | Shifted and rotated Rosenbrock function |
| | f5 | Shifted and rotated Ackley's function |
| | f6 | Shifted and rotated Weierstrass function |
| | f7 | Shifted and rotated Griewank's function |
| | f8 | Shifted Rastrigin function |
| | f9 | Shifted and rotated Rastrigin's function |
| | f10 | Shifted Schwefel function |
| | f11 | Shifted and rotated Schwefel's function |
| | f12 | Shifted and rotated Katsuura function |
| | f13 | Shifted and rotated HappyCat function |
| | f14 | Shifted and rotated HGBat function |
| | f15 | Shifted and rotated Expanded Griewank's plus |
| | f16 | Shifted and rotated Expanded Scaffer's F6 |
| | Hibrit | f17 |
| f18 | | Hybrid function 2 (f2,f12,f8,f2,f12,f8) |
| f19 | | Hybrid function 3 (f7,f6,f4,f14,f7,f6,f4,f14) |
| f20 | | Hybrid function 4 (f12,f3,f13,f8,f12,f3,f13,f8) |
| f21 | | Hybrid function 5 |
| f22 | | Hybrid function 6 |
| Kompozit | f23 | Composition function 1 |
| | f24 | Composition function 2 (f10,f9,f14,f10,f9,f14) |
| | f25 | Composition function 3 (f11,f9,f1,f11,f9,f1) |
| | f26 | Composition function 4 |
| | f27 | Composition function 5 |
| | f28 | Composition function 6 |
| | f29 | Composition function 7 (f17,f18,f19,f17,f18,f19) |
| | f30 | Composition function 8 (f20,f21,f22,f20,f21,f22) |
| | | Arama aralığı: [-100, 100] ^D |

Çalıştırma işlemi, D problemin boyutunu göstermek üzere D x 10000 fonksiyon çağrım sayısına kadar devam ettirilir. Bütün algoritmalar, her bir fonksiyon için 51 defa

bağımsız olarak çalıştırılır. Algoritmaların elde ettiği hata değerleri $f(x) - f(x^*)$ şeklinde hesaplanır. Burada; x bir aday çözümü, x^* optimum çözümü göstermektedir. Hata değeri 10^{-8} 'den daha küçük olan değerler "0" olarak kabul edilir. Deney başlangıcında ölçüt seti, eğitim ve test seti olmak üzere iki kısma ayrılmıştır. Eğitim seti için, CEC 2014 10 boyut problemler tercih edilmiştir. 30, 50 ve 100 boyut ise test seti olarak belirlenmiştir.

Otomatik parametre yapılandırma işlemi için irace aracı kullanılmıştır. Deneylerde irace, ön tanımlı ayarları kullanılarak çalıştırılmıştır. L-shade'nin parametreleri yapılandırılırken eğitim seti olarak kullanılan 10 boyut problemlerin çalıştırma bütçesi 10×5000 fonksiyon çağırımı sayısı olarak belirlenmiştir. Algoritmanın kontrol parametreleri ve aralık değerleri Tablo 2'de verilmiştir. Ayrıca tabloda irace ve paramils kullanılarak elde edilen parametre değerleri de listelenmiştir. Paramils ve ön tanımlı (default) değerler L-shade'nin orijinal çalışmasından elde edilmiştir [11,20].

Tablo 2. L-shade algoritmasının kontrol parametreleri ve aralık değerleri listelenmektedir. Paramils ve irace ile elde edilene ve default parametreler

(The control parameters and range values of the l-shade algorithm are listed. What is obtained with paramils and irace and default parameters)

| Parametreler | Aralıklar | L-shade-default | L-shade-paramils | L-shade-irace |
|--------------|-----------------------------|-----------------|------------------|---------------|
| r^{Ninit} | 15, 16, ..., 24, 25 | 20 | 18 | 13 |
| r^{arc} | 1.0, 1.1, ..., 2.9, 3.0 | 2.0 | 2.6 | 1.6 |
| P | 0.05, 0.06, ..., 0.14, 0.15 | 0.1 | 0.11 | 0.12 |
| H | 2, 3, ..., 9, 10 | 5 | 6 | 10 |

4.2. Cec 2014 Deney Sonuçları (Cec 2014 Test Results)

L-shade'nin ön tanımlı parametreler kullanılarak çalıştırılan hali L-shade-default, irace ile parametreleri yapılandırılmış hali L-shade-irace ve paramils (Hutter vd., 2009) ile parametrelerinin belirlenmiş hali L-shade-paramils'nin 30, 50 ve 100 boyut sonuçları sırasıyla Tablo 3, Tablo 4 ve Tablo 5'de verilmiştir. Tabloların alt kısmında yer alan rank satırı, ölçüt setindeki 30 problemde algoritmaların elde ettiği sıralama değerinin ortalamasını göstermektedir. Win, lost ve draw satırları ise, L-shade-irace'nin karşılaştırıldığı algoritmalarla göre aldığı galibiyet, mağlubiyet ve beraberlik sayılarını göstermektedir. Tablolarda verilen sonuçların yanında L-shade-irace, L-shade-default ve L-shade-paramils 30, 50 ve 100 boyutta bazı fonksiyonlar için yakınsama grafikleri elde edilmiştir ve Şekil 2'de listelenmiştir.

Deneylerde, L-shade-irace ile L-shade-default ve L-shade-paramils'nin elde ettikleri sonuçların istatistiksel

olarak anlamlı olduğunu araştırmak için her problemde $\alpha = 0.05$ 'de Wilcoxon işaretli sıra testi gerçekleştirilmiştir. Tablolarda L-shade-default ve L-shade-paramils sütunlarının önünde yer alan "+" işareti, L-shade-irace'in o problemde elde ettiği sonuçların önemli ölçüde iyi olduğunu, "-" işareti önemli ölçüde kötü olduğunu ve " \approx " işareti ise sonuçların istatistiksel olarak bir farkının olmadığını göstermektedir.

Tablo 3. Cec 2014 30 boyut ortalama hata değerleri. Koyu renkli olarak verilen sonuçlar en iyi değerleri göstermektedir

(Cec 2014 30 dimension mean error values. Results in bold indicate the best values)

| Fonk. | L-shade-irace | L-shade-default | L-shade-paramils |
|---------------|------------------|----------------------------|----------------------------|
| f1 | 0.000E+00 | + 0.000E+00 | + 0.000E+00 |
| f2 | 0.000E+00 | \approx 0.000E+00 | \approx 0.000E+00 |
| f3 | 0.000E+00 | \approx 0.000E+00 | \approx 0.000E+00 |
| f4 | 0.000E+00 | \approx 0.000E+00 | \approx 0.000E+00 |
| f5 | 2.010E+01 | + 2.013E+01 | + 2.016E+01 |
| f6 | 3.292E-02 | - 0.000E+00 | - 0.000E+00 |
| f7 | 0.000E+00 | \approx 0.000E+00 | \approx 0.000E+00 |
| f8 | 0.000E+00 | + 0.000E+00 | + 0.000E+00 |
| f9 | 7.999E+00 | - 6.042E+00 | - 6.096E+00 |
| f10 | 2.082E-02 | + 2.082E-02 | - 0.000E+00 |
| f11 | 1.020E+03 | + 1.090E+03 | + 1.145E+03 |
| f12 | 1.715E-01 | + 2.183E-01 | - 1.262E-01 |
| f13 | 1.401E-01 | - 1.118E-01 | - 8.399E-02 |
| f14 | 1.970E-01 | + 2.316E-01 | + 2.563E-01 |
| f15 | 3.289E+00 | + 9.173E+00 | + 3.788E+00 |
| f16 | 8.204E+00 | + 8.711E+00 | + 8.846E+00 |
| f17 | 2.994E+02 | - 2.069E+02 | + 4.521E+02 |
| f18 | 4.530E+00 | + 5.355E+00 | + 8.338E+00 |
| f19 | 3.299E+00 | + 3.369E+00 | + 4.393E+00 |
| f20 | 3.906E+00 | + 5.755E+00 | + 4.089E+00 |
| f21 | 1.416E+02 | + 1.470E+02 | + 1.704E+02 |
| f22 | 2.687E+01 | - 2.481E+01 | - 2.527E+01 |
| f23 | 3.152E+02 | \approx 3.152E+02 | \approx 3.152E+02 |
| f24 | 2.233E+02 | + 2.235E+02 | - 2.231E+02 |
| f25 | 2.026E+02 | \approx 2.026E+02 | - 2.026E+02 |
| f26 | 1.001E+02 | - 1.001E+02 | + 1.001E+02 |
| f27 | 3.000E+02 | \approx 3.000E+02 | \approx 3.000E+02 |
| f28 | 8.379E+02 | - 8.334E+02 | + 8.382E+02 |
| f29 | 7.156E+02 | - 7.148E+02 | + 7.164E+02 |
| f30 | 1.862E+03 | - 1.289E+03 | + 1.468E+03 |
| rank | 1.63 | 1.67 | 1.87 |
| Win/lost/draw | | 13 / 9 / 8 | 12 / 10 / 8 |

Tablo 3 incelendiğinde, üç algoritma da unimodal fonksiyonlarda (f1-f3) optimum değerini bulmuşlardır. Multimodal fonksiyonların sonuçlarına bakıldığında f4, f7, f8 fonksiyonlarında üç algoritmanın da "0" değerini elde ettiği görülmektedir. L-shade-irace, f5, f11, f14, f15

ve f16 fonksiyonlarında birinciliği elde etmiştir. F9’da L-shade-default, f12 ve f13’de ise L-shade-paramils ilk sırada yer almıştır. Hibrit fonksiyonlarda ise L-shade-irace’in 6 fonksiyonun 4’ünde birinciliği elde ederek diğer iki algoritmaya göre üstünlüğü olduğu görülmektedir. Onu hibrit fonksiyonlarda iki birincilik değeri ile L-shade-default takip etmektedir. Paramils ile edilen parametrelerin hibrit fonksiyonlarda iyi sonuçlar vermediği görülmektedir. Kompozit fonksiyonlarda ise 6 birincilik elde eden öntanımlı parametreleri kullanan L-shade-default ilk sırada yer almıştır. Onu, parametreleri paramils ile yapılandırılmış L-shade-paramils izlemektedir. L-shade-irace ise iki birincilik elde etmiştir. 30 boyut kompozit fonksiyonlarında L-shade-irace, diğer iki yapılandırmaya göre yeterli başarıyı gösterememiştir. Tablonun alt kısmında bütün fonksiyonların özet bir değerlendirilmesi yer almaktadır. Buna göre, L-shade-irace 30 boyutta 1.63’lük ortalama rank değeri ile diğer algoritmaların önünde yer aldığı görülmektedir.

Tablo 4. Cec 2014 50 boyut ortalama hata değerleri. (Cec 2014 50 dimension mean error values. Results in bold indicate the best values)

| Fonk. | L-shade-irace | L-shade-default | L-shade-paramils |
|---------------|------------------|--------------------|--------------------|
| f1 | 6.507E+03 | - 3.975E+02 | - 1.074E+03 |
| f2 | 0.000E+00 | + 0.000E+00 | + 0.000E+00 |
| f3 | 0.000E+00 | + 0.000E+00 | + 0.000E+00 |
| f4 | 9.810E+01 | + 9.840E+01 | ≈ 9.810E+01 |
| f5 | 2.020E+01 | + 2.023E+01 | - 2.020E+01 |
| f6 | 2.330E-01 | + 1.731E+00 | - 0.000E+00 |
| f7 | 0.000E+00 | ≈ 0.000E+00 | ≈ 0.000E+00 |
| f8 | 0.000E+00 | + 1.208E-08 | + 9.864E-08 |
| f9 | 8.489E+00 | + 8.768E+00 | + 1.238E+01 |
| f10 | 1.249E-02 | + 1.811E-01 | + 8.384E-02 |
| f11 | 2.444E+03 | + 3.306E+03 | + 2.883E+03 |
| f12 | 1.597E-01 | + 2.611E-01 | + 1.876E-01 |
| f13 | 2.155E-01 | - 1.672E-01 | - 1.563E-01 |
| f14 | 2.928E-01 | + 3.001E-01 | + 2.986E-01 |
| f15 | 4.918E+03 | + 4.919E+03 | + 4.919E+03 |
| f16 | 1.680E+01 | + 1.772E+01 | + 1.719E+01 |
| f17 | 1.834E+03 | - 1.572E+03 | - 1.129E+03 |
| f18 | 1.088E+02 | - 7.398E+01 | - 7.791E+01 |
| f19 | 5.674E+00 | + 5.956E+00 | + 1.119E+01 |
| f20 | 1.308E+01 | + 3.299E+01 | + 2.263E+01 |
| f21 | 5.747E+02 | - 4.990E+02 | - 5.012E+02 |
| f22 | 5.476E+01 | + 8.907E+01 | + 1.654E+02 |
| f23 | 3.440E+02 | ≈ 3.440E+02 | ≈ 3.440E+02 |
| f24 | 2.754E+02 | - 2.746E+02 | - 2.743E+02 |
| f25 | 2.049E+02 | + 2.059E+02 | + 2.057E+02 |
| f26 | 1.002E+02 | + 1.002E+02 | - 1.002E+02 |
| f27 | 3.000E+02 | + 3.501E+02 | + 3.473E+02 |
| f28 | 1.102E+03 | - 1.089E+03 | + 1.126E+03 |
| f29 | 7.907E+02 | - 7.885E+02 | + 7.956E+02 |
| f30 | 7.014E+03 | - 6.307E+03 | - 6.472E+03 |
| rank | 1.67 | 2.03 | 1.87 |
| Win/lost/draw | | 16 / 10 / 4 | 15 / 10 / 5 |

L-shade deneylerinin ikinci kısmı 50 boyut için gerçekleştirilmiş ve sonuçları ise Tablo 4’te listelenmiştir. F1-f3 unimodal fonksiyonlarda, f1 haricinde üç algoritma da aynı sonuca ulaşmıştır. Multimodal fonksiyonlarda, 30 boyutta olduğu gibi L-shade-irace’in 10 birincilik ile diğer algoritmalara karşı büyük üstünlüğü olduğu söylenebilir. Hibrit fonksiyonlarda, üç birincilik L-shade-irace ve L-shade-default, tek birinciliği de L-shade-paramils elde etmiştir. Son olarak, kompozit fonksiyonlarda parametre yapılandırma yöntemlerini kullanan L-shade-paramils ve L-shade-irace ön tanımlı parametreleri kullanan L-shade-default’un gerisinde kalmıştır. L-shade-default 4 birincilik, L-shade-paramils ve L-shade-irace ise üçer birincilik elde etmişlerdir. Genel toplama bakıldığında, L-shade-irace 1.67, L-shade-paramils ise 1.87 ortalama rank değeri elde ederek parametre yapılandırma yöntemlerini kullanan L-shade’ler 2.03’lük ortalama rank değerine sahip ön tanımlı parametre yöntemi L-shade-default’un önünde yer almıştır.

Tablo 5. Cec 2014 100 boyut ortalama hata değerleri. (Cec 2014 100 dimension mean error values. Results in bold indicate the best values)

| Fonk. | L-shade-irace | L-shade-default | L-shade-paramils |
|--------------|------------------|--------------------|--------------------|
| f1 | 1.323E+05 | + 1.615E+05 | - 9.979E+04 |
| f2 | 0.000E+00 | + 0.000E+00 | + 0.000E+00 |
| f3 | 0.000E+00 | + 0.000E+00 | + 0.000E+00 |
| f4 | 1.435E+02 | + 1.520E+02 | + 2.125E+02 |
| f5 | 2.052E+01 | + 2.060E+01 | + 2.058E+01 |
| f6 | 1.129E+01 | - 5.824E+00 | - 6.875E+00 |
| f7 | 0.000E+00 | ≈ 0.000E+00 | ≈ 0.000E+00 |
| f8 | 3.291E-03 | + 6.290E-02 | - 3.118E-03 |
| f9 | 3.274E+01 | + 3.792E+01 | - 2.284E+01 |
| f10 | 5.521E+00 | + 4.087E+01 | + 2.031E+01 |
| f11 | 9.970E+03 | + 1.132E+04 | + 1.087E+04 |
| f12 | 3.880E-01 | + 4.749E-01 | + 4.265E-01 |
| f13 | 2.578E-01 | - 2.066E-01 | - 1.861E-01 |
| f14 | 3.295E-01 | + 3.571E-01 | + 3.358E-01 |
| f15 | 2.114E+01 | + 2.231E+01 | + 2.256E+01 |
| f16 | 3.904E+01 | - 3.902E+01 | + 3.954E+01 |
| f17 | 2.822E+03 | + 5.113E+03 | + 4.120E+03 |
| f18 | 2.531E+02 | - 2.184E+02 | - 2.254E+02 |
| f19 | 9.716E+01 | - 9.215E+01 | + 9.758E+01 |
| f20 | 2.830E+02 | - 1.857E+02 | - 1.837E+02 |
| f21 | 2.447E+03 | - 2.114E+03 | - 2.140E+03 |
| f22 | 1.091E+03 | - 1.016E+03 | + 1.150E+03 |
| f23 | 3.482E+02 | ≈ 3.482E+02 | ≈ 3.482E+02 |
| f24 | 3.913E+02 | + 3.983E+02 | + 3.928E+02 |
| f25 | 2.000E+02 | ≈ 2.000E+02 | ≈ 2.000E+02 |
| f26 | 2.000E+02 | ≈ 2.000E+02 | ≈ 2.000E+02 |
| f27 | 3.488E+02 | - 3.250E+02 | - 3.237E+02 |
| f28 | 2.165E+03 | + 2.289E+03 | + 2.339E+03 |
| f29 | 7.427E+02 | + 7.764E+02 | + 7.638E+02 |
| f30 | 8.668E+03 | - 8.334E+03 | + 9.506E+03 |
| rank | 1.63 | 1.93 | 1.83 |
| Win / lost / | | 14 / 10 / 6 | 15 / 9 / 6 |

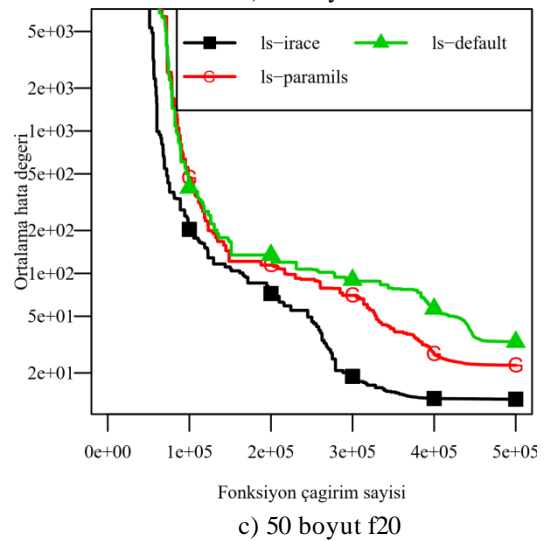
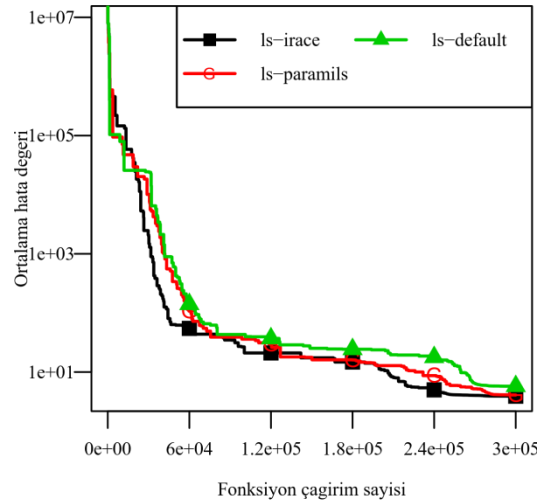
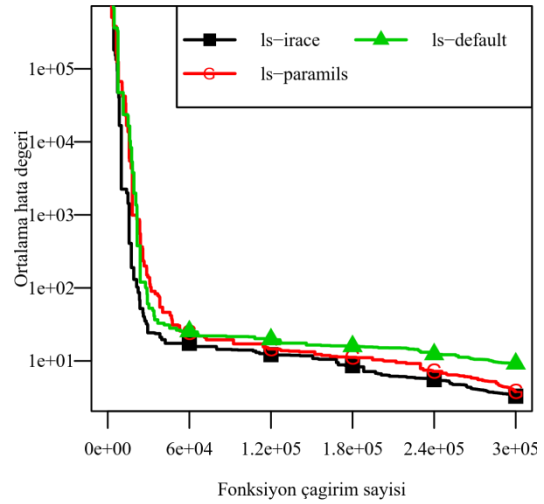
Yapılan deneylerin son kısmı ise CEC 2014 100 boyut deneyleridir. Tablo 5 incelendiğinde L-shade-irace elde ettiği 1.63 ortalama rank değerine göre problemin boyutu arttıkça performansı aynı şekilde devam etmiştir. Unimodal fonksiyonlarda f1 fonksiyonu hariç bütün parametre yöntemleri aynı sonucu elde etmiştir. Multimodal fonksiyonlarda L-shade-irace'in birincilik sayısı 50 boyuttakine göre azalsa da 8 birincilik diğer iki yöntemin önünde yer almıştır. Hibrit fonksiyonlar incelendiğinde ise L-shade-default 4 birincilik almış diğer iki yöntem tek birincilikte kalmıştır. Buradan elde edilen sonuç göstermektedir ki boyut arttıkça hibrit fonksiyonlarda iki parametre yapılandırma yönteminin algoritmayı iyileştirme etkisinin az olduğu görülür. Son olarak, ölçüt setinin en zor problem türleri olan kompozit fonksiyonlarda L-shade-irace'in 6 birincilik, L-shade-paramils'in ve L-shade-default'un 4 birincilik elde ettiği söylenebilir.

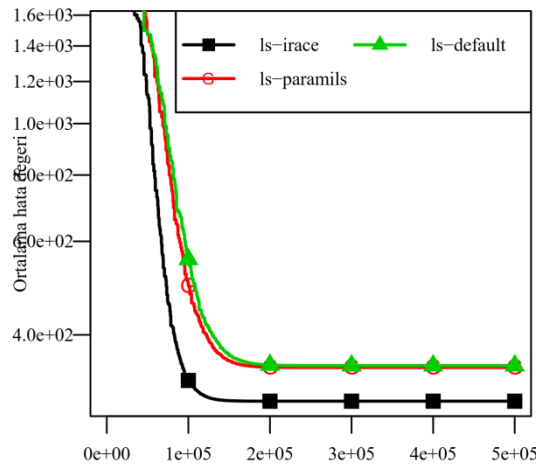
Deneylerden elde edilen bulgulara göre otomatik parametre yapılandırma araçları algoritmanın performansına önemli katkılar sağlamaktadır. Bütüne bakacak olursak otomatik parametre yöntemleri irace ve paramils'nin belirlediği parametreler ön tanımlı parametrelerden daha iyi sonuçlar elde etmiştir. Ölçüt setinin büyük bir kısmını oluşturan multimodal fonksiyonlarda irace aracı daha iyi parametreler belirlemiş bunun sonucu olarak L-shade-irace yapılandırması bu fonksiyonlarda birincilikler elde etmiştir. Ayrıca yapıları daha karmaşık ve çözülmesi daha zor olan hibrit ve kompozit fonksiyonlarda irace ve paramils araçlarının L-shade algoritmasının üstünde olumlu etkileri olduğu görülmüştür. Bütün bu olumlu etkilerin aksine yapıları daha basit olan unimodal fonksiyonlarda otomatik parametre yapılandırma yöntemleri ayırt edici bir fark oluşturmamıştır.

5. SONUÇ (CONCLUSION)

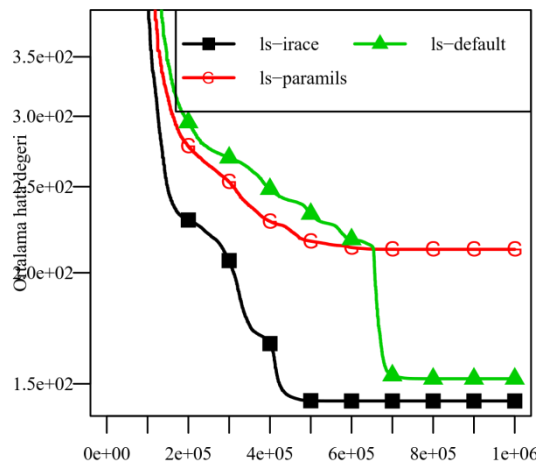
Bu çalışmada, CEC evrimsel algoritma yarışmasının son yıllarına damga vurmuş bir DE varyantı olan L-shade algoritması iyileştirilmeye çalışıldı. L-shade algoritması üzerinde herhangi bir değişiklik yapılmadan algoritmanın performansı daha yukarıya çıkarılabileceği düşüncesi ile yola çıkılmıştır. Algoritmanın kontrol parametreleri irace aracı yardımıyla belirlendi. Deneyler için CEC 2014 ölçüt seti kullanıldı. İrace ile elde edilen parametre değerleri, ön tanımlı parametre değeri ile ve paramils ile elde edilen parametre değerleri ile karşılaştırıldı. Elde edilen sonuçlara göre algoritmanın kontrol parametrelerinin algoritma üzerinde oldukça etkisinin olduğu görüldü. Bir parametre aracının kullanılması hiçbir parametre aracının kullanılmaması durumundan daha iyi olduğu söylenebilir. Son olarak, irace ile belirlenen parametreleri kullanan l-shade algoritmasının elde ettiği sonuçların oldukça rekabetçi olduğu ifade edilebilir.

İlerleyen çalışmalarda farklı parametre yapılandırma araçları kullanılarak bir karşılaştırma yapılabilir. Ayrıca fonksiyon türlerine özel parametreler belirlenerek kontrol parametrelerinin algoritma üzerindeki etkileri araştırılabilir.

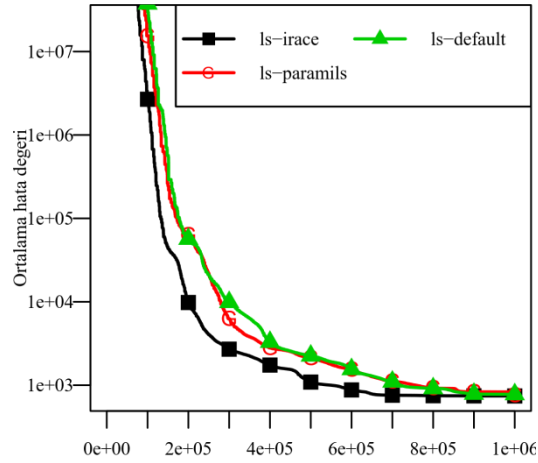




d) 50 boyut f27



e) 100 boyut f4



f) 100 boyut f29

Şekil 2. 30 boyut (f15, f20), 50 boyut (f20, f27) ve 100 boyut (f4, f29) için bazı fonksiyonların yakınsama grafikleri
(Convergence plots of some functions for 30 dimensions (f15, f20), 50 dimensions (f20, f27) and 100 dimensions (f4, f29))

KAYNAKLAR (REFERENCES)

- [1] B. M. Pant, H. Zaheer, L. Garcia-Hernandez, A. Abraham, "Differential Evolution: A review of more than two decades of research", *Eng. Appl. Artif. Intell.*, 90, February, 103479, 2020.
- [2] T. Dokeroglu, E. Sevinc, T. Kucukyilmaz, A. Cosar, "A survey on new generation metaheuristic algorithms", *Comput. Ind. Eng.*, 137, August, 106040, 2019.
- [3] S. A. UYMAZ, "Evaluation of the Most Valuable Player Algorithm for Solving Real-World Constrained Optimization Problems", *Bilişim Teknol. Derg.*, 14, 4, 345–353, 2021.
- [4] G. Yavuz, "Diversified Position Update Equation-Based SSA with Refreshing-Gap Strategy for global optimization", *J. Comput. Sci.*, 60, 101597, 2022 doi: 10.1016/j.jocs.2022.101597.
- [5] R. Storn, K. Price, "Differential Evolution - A simple and efficient adaptive scheme for global optimization over continuous spaces", *Tech. report, Int. Comput. Sci. Inst.*, 11, 1–15, 1995, doi: 10.1023/A:1008202821328.
- [6] D. H. Wolpert, W. G. Macready, "No free lunch theorems for optimization", *IEEE Trans. Evol. Comput.*, 1, 1, 67–82, Apr. 1997, doi: 10.1109/4235.585893.
- [7] I. Boussaid, J. Lepagnot, P. Siarry, "A survey on optimization metaheuristics", *Inf. Sci.*, 237, 82–117, 2013, doi: 10.1016/j.ins.2013.02.041.
- [8] K. Hussain, M. N. Mohd Salleh, S. Cheng, Y. Shi, "Metaheuristic research: a comprehensive survey", *Artif. Intell. Rev.*, 52, 4, 2191–2233, 2019, doi: 10.1007/s10462-017-9605-z.
- [9] S. Aslan, "Time-Based Dance Scheduling for Artificial Bee Colony Algorithm and Its Variants", *Int. J. Comput. Intell. Syst.*, 12, 2, 597, 2019, doi: 10.2991/ijcis.d.190425.001.
- [10] S. Aslan, D. Karaboga, H. Badem, "A new artificial bee colony algorithm employing intelligent forager forwarding strategies", *Appl. Soft Comput.*, 96, 106656, 2020, doi: 10.1016/j.asoc.2020.106656.
- [11] R. Tanabe, A. S. Fukunaga, "Improving the search performance of SHADE using linear population size reduction", **Proc. 2014 IEEE Congr. Evol. Comput. CEC 2014**, 1658–1665, 2014, doi: 10.1109/CEC.2014.6900380.
- [12] A. P. Piotrowski, J. J. Napiorkowski, "Step-by-step improvement of JADE and SHADE-based algorithms: Success or failure?", *Swarm Evol. Comput.*, 1–21, 2018, doi: 10.1016/J.SWEVO.2018.03.007.
- [13] G. Yavuz, "Metasezgisel Algoritmaların Karşılaştırılmasında Kullanılan Ölçüt Setleri", **Dijital Mühendislik**, Iksad Publications, TR, 29–59, 2020.
- [14] R. Tanabe, A. Fukunaga, "Reviewing and Benchmarking Parameter Control Methods in Differential Evolution," *IEEE Trans. Cybern.*, 50, 3, 1170–1184, 2020, doi: 10.1109/TCYB.2019.2892735.
- [15] M. A. Elhosseini, R. A. El Sehiemy, Y. I. Rashwan, X. Z. Gao, "On the performance improvement of elephant herding optimization algorithm", *Knowledge-Based Syst.*, 166, 58–70, 2019, doi: 10.1016/j.knosys.2018.12.012.

- [16] T. Liao, M. A. M. de Oca, T. Stützle, “Computational results for an automatically tuned CMA-ES with increasing population size on the CEC’05 benchmark set”, *Soft Comput.*, 17, 6, 1031–1046, 2013, doi: 10.1007/s00500-012-0946-x.
- [17] T. Liao, D. Molina, T. Stützle, “Performance evaluation of automatically tuned continuous optimizers on different benchmark sets”, *Appl. Soft Comput. J.*, 27, 490–503, 2015, doi: 10.1016/j.asoc.2014.11.006.
- [18] D. Aydın, G. Yavuz, T. Stützle, “ABC-X: a generalized, automatically configurable artificial bee colony framework”, *Swarm Intell.*, 11, 1, 1–38, 2017, doi: 10.1007/s11721-017-0131-z.
- [19] J. J. Liang, B. Y. Qu, P. N. Suganthan, “Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization”, *Comput. Intell. Lab.*, 2013.
- [20] A. P. Piotrowski, “L-SHADE optimization algorithms with population-wide inertia”, *Inf. Sci.*, 468, 117–141, 2018, doi: 10.1016/j.ins.2018.08.030.
- [21] C. Huang, Y. Li, X. Yao, “A Survey of Automatic Parameter Tuning Methods for Metaheuristics”, *IEEE Trans. Evol. Comput.*, 24, 2, 201–216, 2020, doi: 10.1109/TEVC.2019.2921598.
- [22] F. Hutter, H. H. Hoos, K. Leyton-Brown, T. Stützle, “ParamILS: An automatic algorithm configuration framework”, *J. Artif. Intell. Res.*, 36, 1, 267–306, 2009, doi: 10.1613/jair.2808.
- [23] C. Ansótegui, M. Sellmann, K. Tierney, “A Gender-Based Genetic Algorithm for the Automatic Configuration of Algorithms”, **International Conference on Principles and Practice of Constraint Programming**, Berlin, 2009.
- [24] F. Hutter, H. H. Hoos, K. Leyton-Brown, “Sequential Model-Based Optimization for General Algorithm Configuration”, *Lion-5*, 507–523, 2011.
- [25] M. Birattari, Z. Yuan, P. Balaprakash, T. Stützle, “F-Race and Iterated F-Race: An Overview”, **Experimental Methods for the Analysis of Optimization Algorithms**, Springer, Berlin, 311–336, 2010.
- [26] M. López-Ibáñez, J. Dubois-Lacoste, L. Pérez Cáceres, M. Birattari, T. Stützle, “The irace package: Iterated racing for automatic algorithm configuration”, *Oper. Res. Perspect.*, 3, 43–58, 2016, doi: 10.1016/j.orp.2016.09.002.