

# Kutulama Problemi için Geliştirilmiş Karınca Aslanı Optimizasyonu Algoritması

## Improved Antlion Optimization Algorithm for Bin Packing Problem

Haydar KILIÇ  
Bilecik Şeyh Edebali Üniversitesi  
Mühendislik Fakültesi  
Bilgisayar Mühendisliği Bölümü  
Bilecik, TÜRKİYE  
haydar.kilic@bilecik.edu.tr

Uğur YÜZGEÇ  
Bilecik Şeyh Edebali Üniversitesi  
Mühendislik Fakültesi  
Bilgisayar Mühendisliği Bölümü  
Bilecik, TÜRKİYE  
ugur.yuzgec@bilecik.edu.tr

### Öz

Bu çalışmada kutulama problemi için bir geliştirilmiş karınca aslanı optimizasyon algoritması (GKAO) önerilmiştir. Karınca aslanı optimizasyon algoritması (KAO) temel olarak karınca aslanlarının avlanma stratejilerini taklit eden bir meta-sezgisel optimizasyon algoritmasıdır. KAO algoritmasının büyük handikaplarından birisi uzun çalışma süresidir. KAO yapısında yer alan rastgele karınca yürüyüşü modeline seçim yönteminde yapılan iyileştirmelerle ortaya çıkarılan GKAO bu handikapı ortadan kaldırmıştır. Önerilen GKAO algoritması kutulama problemi olarak adlandırılan optimizasyon problemine uyarlanarak test edilmiştir. Önerilen algoritma parçacık sürüsü optimizasyon algoritması (PSO), ateş böceği algoritması (FA), istilacı yabani ot optimizasyon algoritması (IWO) ve karınca aslanı optimizasyon algoritması (KAO) ile karşılaştırılmıştır. Sonuçlar önerilen GKAO algoritma performansının kullanılan meta-sezgisel algoritma performanslarından daha başarılı olduğunu göstermiştir.

**Anahtar Sözcükler—** Karınca Aslanı Algoritması, Optimizasyon, Kutulama Problemi

### Abstract

In this study, an improved antlion optimization algorithm (IALO) was proposed for bin packing problem. Antlion optimization algorithm is a meta-heuristic optimization algorithm that basically imitates the hunting mechanism of antlions. The biggest disadvantage of antlion algorithm is its long running time.

**Gönderme ve Kabul tarihi:** 21.04.2018-26.10.2018

**Makale türü:** Araştırma

By the improvements on the ant random walking model and selection method in ALO algorithm, IALO algorithm eliminated this deficiency. The proposed IALO algorithm was tested by adapting to the optimization problem known as bin packing problem. The proposed IALO algorithm was compared with particle swarm optimization algorithm (PSO), firefly algorithm (FA), invasive weed optimization algorithm (IWO) and antlion optimization algorithm (ALO). The results show that the performance of IALO algorithm is more successful than the performances of used meta-heuristic algorithms.

**Keywords—** Antlion Algorithm, Optimization, Bin Packing Problem

### 1. Giriş

Günümüzün mühendislik bilimlerinde, meta-sezgisel algoritmalar, optimizasyon problemlerini çözmeye büyük avantajlara sahiptir. Son yıllarda farklı optimizasyon problemlerinin çözümünde pek çok meta-sezgisel algoritmaların geliştirildiği görülmektedir. Bu algoritmalar evrimsel tabanlı, fiziksel tabanlı, sürü zekasına dayanan ve biyolojik ilhamlı algoritmalar olarak sınıflandırılabilirler [1-3].

Evrimsel tabanlı meta-sezgisel algoritmalar en bilinenleri, genetik algoritma (GA) [4][5] vefarsal gelişim (DE) algoritmasıdır [6][7]. Bu algoritma sınıfında, rastgele bir popülasyon ile çözüme başlanır ve bu popülasyon çarpazlama ve mutasyon gibi çeşitli evrimsel mekanizmalarla güncellenir. Benzetilmiş tavlama algoritması (SA) [8], tabu arama algoritması (TS) [9], harmoni arama algoritması (HSA) [10][11] ve kurbağa sıçraması algoritması (SFLA) [12][13] fiziksel tabanlı meta-sezgisel algoritmaların en popüler olanlarıdır. Sürü zekasına dayanan algoritmalar kuş sürüleri, karınca kolonileri, balık sürüleri gibi kolektif zekayı taklit eden meta-sezgisel algoritmalar

grubundandır [1]. Bu algoritmalarından bazıları parçacık sürüsü optimizasyon algoritması (PSO) [14][15], yapay arı kolonisi algoritması (ABC) [16][17], karınca koloni algoritması (ACO) [18][19] olarak sayılabilir. Yapay bağımsız algoritması (AI) [20][21] ve bakteriyel yiyecek arama algoritması [22][23] biyolojik ilhamlı meta-sezgisel algoritmalar önemli örneklerdendir.

Bu çalışmada 2015 yılında Mirjalili [24] tarafından sunulan karınca aslanı optimizasyon (KAO) algoritması ele alınmıştır. Bu algoritma, karınca aslanı larvalarının kendilerine özgü avlanma tekniklerinden esinlenerek geliştirilmiş bir meta-sezgisel optimizasyon algoritmasıdır. Literatürde kontrolcü tasarımı [25][26], yük sevkiyat problemi [27][28], rota planlaması [29], esnek süreç planlaması [30], üretim çizelgeleme [31], optimizasyon tabanlı regülatör [32], optimal topluluk tespiti [33], optimal filtre tasarımı [34] ve güç sistemleri optimizasyon problemleri [35] gibi mühendislik alanlarında KAO algoritmasına ait uygulamalara rastlanmaktadır.

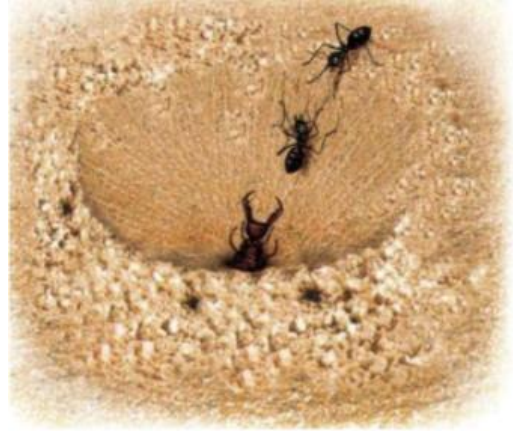
Kutulama problemi lojistik ve üretim gibi alanlarda karşımıza çıkan bir optimizasyon problemidir. Bu problem bir kutunun içerisinde en az boş alan kalacak şekilde eşyaları yerleştirmeyi hedeflemektedir. Kutulama problemi boyutlarına göre bir, iki ve üç boyutlu kutulama problemleri olarak üçe ayrılmaktadır. Kamyon yükleme, konteynır yükleme ve şerit paketleme problemleri bu problemin özel halleridir [36][37].

Bu çalışma kapsamında karınca aslanı optimizasyon algoritması geliştirilerek, GKAO algoritması kutulama problemine uyarlanmıştır. Önerilen GKAO algoritmasının performansına test etmek için parçacık sürüsü optimizasyon algoritması (PSO), ateş böceği algoritması (FA), istilacı yabani ot optimizasyon algoritması (IWO) ve karınca aslanı optimizasyon algoritması (KAO) kullanılmıştır. Elde edilen sonuçlar önerilen GKAO algoritmasının diğer meta-sezgisel algoritmalarla alternatif olabileceğini göstermektedir.

### 3. Karınca Aslanı Algoritması

Myrmeleontidae ailesinden olan karınca aslanları larva evresindeki son derece ilginç beslenme davranışlarından ismini alan yırtıcı bir böcek türüdür. Karınca aslanları karıncaların bulunduğu bölgelere tuzaklarını dairesel bir yol çizerek bir koni şeklinde oluştururlar ve bu tuzagın dibine yani koninin sivri ucuna kendilerini gömerek tuzaga düşecek karıncaları beklerler. Karıncalar tuzaga girdiğinde tuzaktan çıkmasını engellemek ve tuzagın dibine kaydırmak

amacıyla karınca aslanları kum fırlatmaya başlarlar. Sonunda tuzagın dibine kaydırdıkları karıncaları büyük çeneleri ile yutarlar. Bu şekilde gelişen her bir avlanma işinden sonra karınca aslanlar tuzaklarını yeni bir av için hazır hale getirirler. Bu avlanma mekanizması Şekil 1’de gösterilmektedir.



Şekil 1. Karınca aslanı avlanma stratejisi.

Bu ilginç avlanma mekanizmasına ait matematiksel model rastgele yürüyüşlerle başlar:

$$X(t) = \begin{bmatrix} 0 \\ \text{cumsum}(2r(t_1) - 1) \\ \text{cumsum}(2r(t_2) - 1) \\ \vdots \\ \text{cumsum}(2r(t_n) - 1) \end{bmatrix} \quad (1)$$

Burada  $n$  maksimum iterasyon sayısı,  $trastgele$  yürüyüş adımları,  $\text{cumsum}$  kümülatif toplam ve aşağıda tanımlanan  $r(t)$  bir stokastik fonksiyondur:

$$r(t) = \begin{cases} 1, & \text{if } rand > 0.5 \\ 0, & \text{if } rand \leq 0.5 \end{cases} \quad (2)$$

Rastgele yürüyüşe başlayan karıncaları arama uzayında tutmak için aşağıdaki formülle bu yürüyüşleri normalize etmek gerekmektedir:

$$X_i^t = (X_i^t - a_i)(d_i^t - c_i^t)(b_i - a_i)^{-1} + c_i^t \quad (3)$$

Buradaki  $i$  değişken sayısını,  $t$  iterasyon sayısını,  $a$  minimum rastgele yürüyüşünü,  $b$  maksimum rastgele yürüyüşünü,  $c$  ve  $d$  her bir iterasyonda güncellenen karınca aslanı pozisyonlarının sırasıyla minimum ve maksimum değerlerini göstermektedir.

Karıncaların yürüyüşleri karınca aslanlarından doğal olarak etkilenmektedir. Karınca tuzaga girdiğinde,

karınca aslanı onları tuzağın dibine çekmek için kum fırlatmaya başlar. Bu işleme ait matematik modellemesi olmak üzere aşağıdaki gibidir:

$$c_i^t = Antlion_i^t + c^t \quad (4)$$

$$d_i^t = Antlion_i^t + d^t \quad (5)$$

$$c^t = c^t \cdot I^{-1} \quad (6)$$

$$d^t = d^t \cdot I^{-1} \quad (7)$$

Burada  $I$  kaydırma oranını göstermektedir ve optimizasyon sırasında belirli oranlarda artırılır. Bu mekanizmanın ayrıntıları Mirjalili'nin çalışmasında [24] bulunabilir. Eşitlik (3) yardımıyla karıncalar rulet tekerleği ile seçilen karınca aslanı ve elit karınca aslanı etrafında dolaşırlar. Bu şekilde karıncaların yeni pozisyonları aşağıdaki gibi güncellenir:

$$Ant_i^t = 0.5(R_A^t + R_E^t) \quad (8)$$

Burada  $Ant_i^t$  iterasyondaki  $i$ . karıncayı ifade etmektedir. Karınca aslanı tuzağın dibine kaydırıldığı karıncaları yediğinde kendi pozisyonunu Eşitlik (9)'a göre günceller:

$$if f(Ant_i^t) < f(Antlion_i^t), Antlion_i^t = Ant_i^t \quad (9)$$

Burada  $Antlion_i^t$  iterasyondaki  $i$ . karınca aslanını,  $Ant_i^t$  iterasyondaki  $i$ . karıncayı ifade eder.

#### 4. Geliştirilmiş Karınca Aslanı Optimizasyon Algoritması (GKAO)

Karınca aslanı optimizasyon algoritması (KAO) Mirjalili [24] çalışmasında farklı özelliklere sahip optimizasyon test fonksiyonları için başarılı sonuçlar vermesine rağmen literatürde bu algoritma ile ilgili herhangi bir çalışma süresi analizine rastlanamamıştır. KAO algoritmasının en büyük dezavantajı çalışma süresinin uzun olmasıdır. Bunun nedeni karınca yürüyüş modeli için kullanılan rastgele yürüyüş modeli uzunluğudur. Bu nedenle geliştirilen KAO algoritmasında öncelikler rastgele yürüyüşlerde güncelleme yapılarak, var olan algoritmanın koşma süresi bakımından iyileştirilmesi sağlanmıştır. Buradarastgele yürüyüş modelinde maksimum iterasyon sayısı yerine bu sayının yüzde yirmisi alınarak daha kısa rastgele yürüyüşlerle daha optimal bir sonuç elde edilmeye çalışılmıştır.

$$X(t) = [0, \dots, cumsum(2r(t_n - 1))], \quad n: [1, Max\_iter/5] \quad (10)$$

Meta-sezgisel optimizasyon algoritmalarında bir sonraki jenerasyon için birey seçiminde bir çok yöntem kullanılmaktadır. Bu yöntemlerden birisi de rulet tekerleği yöntemidir. Mirjalili [24] çalışmasında rastgele yürüyüş modelinde karıncaların seçilen bir karınca aslanı etrafında yürümesini formüle etmiş ve bu seçim işlemini de rulet tekerleği yöntemi ile yapmıştır. Ancak kullanılan optimizasyon problemine ait arama uzayı negatif değerler içeriyorsa, bu yöntem seçilen karınca aslanını sürekli ilk indeksten almaktadır. Bu hata rulet tekerleğine gönderilen değer in mutlak değeri alınarak Eşitlik (11)'deki gibi çözülmüştür.

$$\frac{|f(Antlion_i^{-1})|}{\sum_{j=1}^n |f(Antlion_j^{-1})|}, i = 1, 2, \dots, n \quad (11)$$

KAO algoritmasında, karıncalar  $I$  kaydırma oranında kaydırılarak karınca aslanına yem olmaktadır. Daha sonra karınca aslanı pozisyonu güncellenmekteydi. Önerilen GKAO yapısında bu güncelleme işlemi rastgele değişen bir parametreye bağlanarak yeni bir model oluşturuldu. Bu model karıncaların tuzak içerisindeki durumlarına göre karınca aslanının pozisyonunu güncellemektedir.

$$\left. \begin{aligned} c_i^t &= Antlion_i^t + c^t \\ d_i^t &= Antlion_i^t + d^t \end{aligned} \right\} if 0.75 < opt < 1 \quad (12)$$

$$\left. \begin{aligned} c_i^t &= Antlion_i^t - c^t \\ d_i^t &= Antlion_i^t - d^t \end{aligned} \right\} if 0.5 < opt < 0.75 \quad (13)$$

$$\left. \begin{aligned} c_i^t &= -Antlion_i^t + c^t \\ d_i^t &= -Antlion_i^t + d^t \end{aligned} \right\} if 0.25 < opt < 0.5 \quad (14)$$

$$\left. \begin{aligned} c_i^t &= -Antlion_i^t - c^t \\ d_i^t &= -Antlion_i^t - d^t \end{aligned} \right\} if opt < 0.25 \quad (15)$$

Elit karınca aslanı ve rulet tekerleği yöntemi ile seçilen karınca aslanı etrafında yürüyen karıncaların normalize edilmiş yürüyüş modelleri karıncaların güncel pozisyonlarının hesaplanmasında aşağıdaki gibi kullanılmaktadır.

$$Ant_i^t = \frac{R_A^{r(t_n)} + R_E^{r(t_n)}}{2}, n = 1, 2, \dots, \frac{Max\_iter}{5} \quad (16)$$

Burada  $r(t_n)[0, t_n]$  aralığında değişen rastgele sayıyı göstermektedir. Karıncaların güncelleme sonrasında arama uzayı sınırları dışına çıkmasını engellemek için rastgele bir şekilde tekrar arama uzayında bir pozisyon belirlenir. Eşitlik (17)'de bu mekanizma verilmektedir.

$$\left. \begin{aligned} Ant_i^t &= b_{low} + rand \times (b_{up} - b_{low}) \\ &if (Ant_i^t > b_{up}) or (Ant_i^t < b_{low}) \end{aligned} \right\} \quad (17)$$

Burada  $rand(0-1)$  arasında rastgele bir sayıyı,  $b_{low}$  alt sınırı,  $b_{up}$  üst sınırı göstermektedir. Orijinal KAO algoritmasında, her iterasyon sonunda karıncalar ve karınca aslanları birleştirilerek, maliyet değerlerine göre sıralanmakta ve popülasyon boyu kadar olan ilk kısımların karınca aslanı olarak kabul edilmektedir. Önerilen algoritmada ise birleştirme ve sıralama işlemleri yerine iterasyon sonunda  $i$ . karınca ile  $i$ . karınca aslanı maliyetleri karşılaştırılarak, daha iyi olan bir sonraki iterasyondaki karınca aslanı pozisyonu olarak alınmıştır. Önerilen GKAO algoritmasına ait sözde kod yapısı Algoritma 1'de verilmektedir.

#### Algoritma 1. Kutulama Problemi için GKAO algoritmasının sözde kodu.

```

Karınca aslanları başlangıç pozisyonu belirle
Karınca aslanları maliyet değerleri hesapla
Elit karınca aslanı ve pozisyonu sakla
while (iterasyon < maksimum iterasyon)
for (her bir karınca aslanı)
    Karınca aslanı seçimi
    Rastgele yürüyen karıncaların bir tuzağa doğru kayması (Eşitlik 12-15)
    Elit karınca aslanı etrafında karıncanın rastgele yürüyüş modelinin çıkarılması
    Seçilen karınca aslanı etrafında karıncanın rastgele yürüyüş modelinin çıkarılması
    Rastgele yürüyüş modelinin normalize edilmesi
    Karıncanın pozisyonun belirleme (Eşitlik 8)
    if Karınca arama uzayı dışında ise,
        Rastgele arama uzayı içerisine at
    end if
end for
Karınca maliyet değerleri hesapla
for (her bir karınca aslanı)
    if karıncanın maliyeti daha iyi ise,
        karınca aslanı karıncayı yer ve pozisyonunu karıncanıniki ile günceller.
    end if
end for
Elit karınca aslanı güncelle
end while

```

## 5. Kutulama Problemi

Kutulama problemi üzerinde çok çalışılan klasik ve zor bir optimizasyon problemidir. Bu problemde temel olarak bir kutunun içerisinde en az boş alan bırakılarak, eşyaların en optimum şekilde nasıl yerleştirileceğine çözüm bulmaya çalışılmaktadır. Kutulama problemi boyutuna ve paketlenen nesneye göre sınıflandırılabilir [36][37].

$s_p$  boyutunda  $P$  adet ürün ve her birinin kapasitesi  $C$  olan  $B = \{1, \dots, P\}$  aday kutular kümesi için kutulama

optimizasyon probleminin matematiksel tanımı aşağıda verilmiştir.

$$\min \sum_{b \in B} y_b \quad (18)$$

$$s. t. \sum_{b \in B} x_{pb} = 1, p \in P$$

$$\min \sum_{p \in P} s_p x_{pb} \leq C y_b, b \in B$$

$$x_{pb} \in \{0,1\}, p \in P, b \in B$$

$$y_b \in \{0,1\}, b \in B$$

Burada kullanılan kısıtlarda her bir ürünün tam olarak bir kutuya atanması sağlanmaktadır.  $y_b$  ve  $x_{pb}$  ikili değişkenlerdir. Eğer  $x_{pb} = 1$  ise,  $p$  ürününün  $b$  kutusuna atıldığını göstermektedir. Aynı şekilde eğer  $y_b = 1$  ise,  $b$  kutusunun kullanıldığı anlaşılmaktadır.

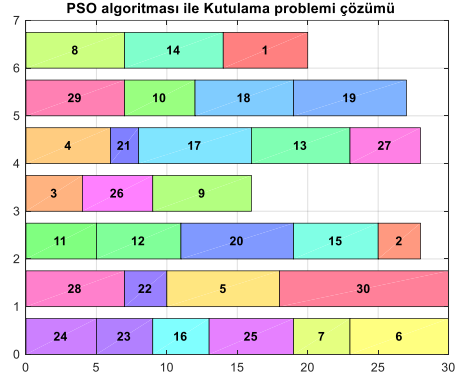
## 6. Deneysel Sonuçlar

Bu çalışma kapsamında önerilen GKAO algoritması kutulama problemi için uyarlanarak, bu problemin çözümü için elde edilen algoritma sonuçları literatürden alınan parçacık sürüsü optimizasyon algoritması (PSO), ateş böceği algoritması (FA), istilacı yabani ot optimizasyon algoritması (IWO) ve karınca aslanı optimizasyon algoritması (KAO) ile karşılaştırılmıştır. Kutulama problemi örneği [www.yarpiz.com](http://www.yarpiz.com) web sitesinden alınmıştır [38]. Bu problemde, farklı ebatlarda 30 ürünün maksimum kapasitesi 30 olan kutulardan en az kaç tanesine minimum maliyetle yerleştirilebileceği bulunmaya çalışılmıştır. Çözümlerde kutu kapasitesini aşma durumlarında maliyet fonksiyonuna ihlal cezası ( $\alpha \cdot \sqrt{viol}$ ) ilave edilmiştir. Burada  $\sqrt{viol}$  ortalama ihlali göstermektedir ve maliyet hesaplamalarında  $\alpha = 60$  olarak kullanılmıştır. Bir boyutlu kutulama probleminde her bir ürün boyu  $v[1 \times 30]$  isimli bir vektörde tutulmaktadır.

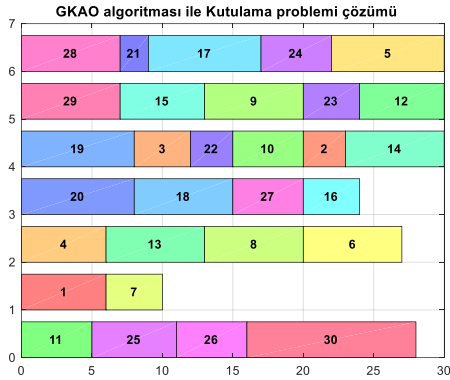
GKAO ve diğer meta-sezgisel algoritmalara ait kodlar, Intel(R) Core(TM) i5-3230M CPU@2.60GHz RAM/8 bir bilgisayarda oluşturulmuştur. Her bir algoritma için popülasyon boyu 20 ve maksimum iterasyon sayısı ise 1000 olarak kullanılmıştır. Karşılaştırma için bu çalışmada kullanılan meta-sezgisel algoritmaların parametreleri Tablo 1'de sunulmaktadır. Şekil 2-6'da bu çalışmada önerilen GKAO ve diğer meta-sezgisel algoritmalar ile bulunan kutulama problemi çözümleri gösterilmektedir.

Tablo 1.Kutulama probleminde kullanılan metasezgisel algoritmaların parametre değerleri.

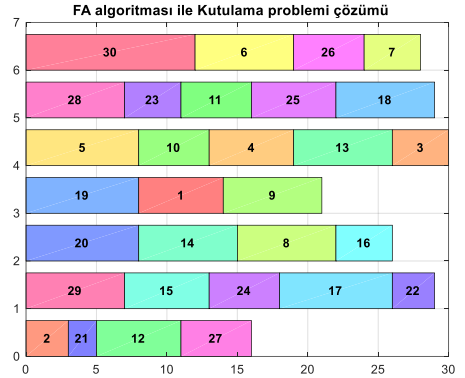
Algoritma	Parametreler
PSO	Atalet ağırlığı : 1.0 Atalet ağırlık sönümleme oranı : 0.99 Bireysel öğrenme katsayısı : 1.5 Global öğrenme katsayısı: 2.0
FA	Işık emme katsayısı: 1.0 Başlangıç çekim katsayısı: 2.0 Mutasyon katsayısı : 0.2 Mutasyon sönümleme oranı: 0.98
IWO	Minimum tohum sayısı : 0 Maksimum tohum sayısı : 5.0 Varyans azaltma bileşeni : 2.0 Standart sapma başlangıç değeri : 1.0 Standart sapma son değeri : 0.001
KAO	Karınca aslanı sayısı : 20
GKAO	Karınca aslanı sayısı : 20



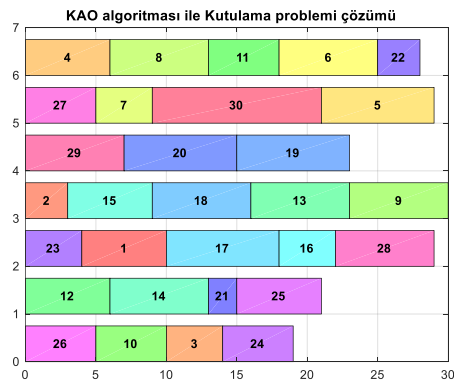
Şekil 4. PSO algoritması ile bulunan kutulama problemi çözümü.



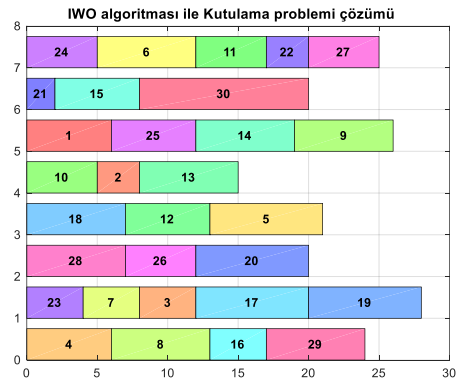
Şekil 2. GKAO algoritması ile bulunan kutulama problemi çözümü.



Şekil 5. FA algoritması ile bulunan kutulama problemi çözümü.



Şekil 3. KAO algoritması ile bulunan kutulama problemi çözümü.

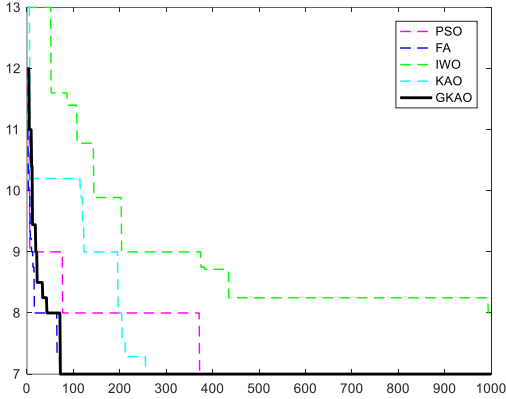


Şekil 6. IWO algoritması ile bulunan kutulama problemi çözümü.

Şekil 6'da gösterilen IWO algoritması ile bulunan kutulama problemi çözümü haricinde önerilen GKAO ve diğer algoritmaların çözümlerinde 1000 iterasyon



sonunda maliyet değeri 7 kutu olarak bulunmuştur. Algoritma çözümleri arasındaki fark ürünlerin eşleştirildiği kutu farklılıkları olarak öne çıkmaktadır. Şekil 7'de önerilen GKAO ve diğer meta-sezgisel algoritmalarının kutulama optimizasyon problemi çözümünde elde edilen yakınsama eğrileri bir arada gösterilmiştir. Şekilden de görüleceği gibi, GKAO ve FA algoritmaları performans olarak yaklaşık 80. iterasyonda en iyi maliyet değerini yakalamışlardır.



Şekil 7. Kutulama problemi için yakınsama eğrileri (PSO, FA, IWO, KAO, GKAO)

## 7. Sonuçlar ve Tartışma

Bu çalışmada meta-sezgisel algoritmalarından birisi olan ve karınca aslanı avlanma stratejisine dayanan karınca aslanı optimizasyon (KAO) algoritması ele alınmıştır. İlk olarak mevcut orijinal KAO algoritması geliştirilerek, algoritma hızlandırılmış ve önerilen GKAO algoritması zor optimizasyon problemlerinden birisi olan kutulama problemine uyarlanmıştır. GKAO algoritmasının kutulama problemi çözümü performansını değerlendirmek için literatürden PSO, FA, IWO ve orijinal KAO algoritması alınarak, bu algoritmaların karşılaştırılması yapılmıştır. Elde edilen sonuçlardan GKAO algoritmasının FA algoritması ile birlikte en iyi yakınsama eğrisine sahip olduğu görülmektedir. İleriki çalışmalarda GKAO algoritmasını farklı mekanizmalar eklemek suretiyle algoritmanın performansının daha da geliştirilmesi ve GKAO algoritmasının farklı zor optimizasyon problemlerine uygulanması hedeflenmektedir.

## Kaynakça

[1] S. J. Nanda and G. Panda, "A survey on nature inspired metaheuristic algorithms for partitional clustering," *Swarm and Evolutionary Computation*, vol. 16. pp. 1–18, 2014.

[2] Z. Beheshti and S. M. H. Shamsuddin, "A review of population-based meta-heuristic algorithm," *International Journal of Advances in Soft Computing and its Applications*, vol. 5, no. 1, pp. 1-35, 2013.

[3] M. H. N. Tayarani, X. Yao, and H. Xu, "Meta-Heuristic Algorithms in Car Engine Design: A Literature Survey," *IEEE Trans. Evol. Comput.*, vol. 19, no. 5, pp. 609–629, 2015.

[4] J. H. Holland, "Adaptation in Natural and Artificial Systems," *Ann Arbor MI Univ. Michigan Press*, vol. Ann Arbor, p. 183, 1975.

[5] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, vol. Addison-Wesley, 1989.

[6] R. Storn and K. Price, "Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces," *J. Glob. Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[7] K. V Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*, vol. 28. 2005.

[8] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by Simulated Annealing," *Science (80-)*, vol. 220, no. 4598, pp. 671–680, 1983.

[9] F. Glover, "Future paths for integer programming and links to artificial intelligence," *Comput. Oper. Res.*, vol. 13, no. 5, pp. 533–549, 1986.

[10] Zong Woo Geem, Joong Hoon Kim, and G. V. Loganathan, "A New Heuristic Optimization Algorithm: Harmony Search," *Simulation*, vol. 76, no. 2, pp. 60–68, 2001.

[11] X. S. Yang, "Harmony search as a metaheuristic algorithm," *Studies in Computational Intelligence*, vol. 191. pp. 1–14, 2009.

[12] M. Eusuff, K. Lansey, and F. Pasha, "Shuffled frog-leaping algorithm: A memetic meta-heuristic for discrete optimization," *Eng. Optim.*, vol. 38, no. 2, pp. 129–154, 2006.

[13] K. K. Bhattacharjee and S. P. Sarmah, "Shuffled frog leaping algorithm and its application to 0/1 knapsack problem," *Appl. Soft Comput. J.*, vol. 19, pp. 252–263, 2014.

[14] J. Kennedy and R. Eberhart, "Particle swarm optimization," *Neural Networks, 1995. Proceedings., IEEE Int. Conf.*, vol. 4, pp. 1942–1948 vol.4, 1995.

[15] R. Eberhart, J. Kennedy, "A new optimizer using particle swarm theory", in: *Sixth International Symposium on Micro Machine and Human Science, MHS, 1995*, pp. 39–43.

[16] D. Karaboga and C. Ozturk, "A novel clustering approach: Artificial Bee Colony (ABC) algorithm," *Appl. Soft Comput. J.*, vol. 11, no. 1, pp. 652–657, 2011.

[17] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," *Technical Report-TR06*, Erciyes University, Engineering Faculty, Computer Engineering Department, Tech. Rep., 2005.

[18] M. Dorigo, V. Maniezzo, and A. Colomi, "Ant system: Optimization by a colony of cooperating agents," *IEEE*

- Trans. Syst. Man, Cybern. Part B Cybern., vol. 26, no. 1, pp. 29–41, 1996.
- [19] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Comput. Intell. Mag.*, vol. 1, no. 4, pp. 28–39, 2006.
- [20] D. Dasgupta, "Artificial Immune Systems and their Applications", Springer-Verlag, 1999, ISBN3540643907.
- [21] L.N. de Charsto, J. Timmis, "An Introduction to Artificial Immune Systems: A New Computational Intelligence Paradigm", Springer-Verlag, 2002.
- [22] K. Passino, "Biomimicry of bacterial foraging for distributed optimization and control", *IEEE Control Syst. Mag.*, vol. 22, no.3, pp.52–67, 2002.
- [23] S. Mishra, "A hybrid least square-fuzzy bacterial foraging strategy for harmonic estimation", *IEEE Trans. Evol. Comput.*, vol. 9, no.1, pp.61–73, 2005.
- [24] S. Mirjalili, "The ant lion optimizer," *Adv. Eng. Softw.*, vol. 83, pp. 80–98, 2015.
- [25] More Raju, Lalit Chandra Saikia, Nidul Sinha, "Automatic generation control of a multi-area system using ant lion optimizer algorithm based PID plus second order derivative controller", *International Journal of Electrical Power and Energy Systems*, Volume 80, September 2016, Pages 52-63, ISSN 0142-0615
- [26] Satheeshkumar, R., Shivakumar, R. "Ant Lion Optimization Approach for Load Frequency Control of Multi-Area Interconnected Power Systems", *Circuits and Systems*, 7(09), 2357,2016.
- [27] Kamboj, V. K., Bhadoria, A., Bath, S. K., "Solution of non-convex economic load dispatch problem for small-scale power systems using ant lion optimizer", *Neural Computing and Applications*, 1-12, 2016.
- [28] Nischal, M. M., Mehta, S., "Optimal load dispatch using ant lion optimization", *Int J Eng Res Appl*, 5(8), 10-19, 2015.
- [29] Yao, P., Wang, H., "Dynamic Adaptive Ant Lion Optimizer applied to route planning for unmanned aerial vehicle", *Soft Computing*, 1-14, 2016.
- [30] Petrovic, M., Petronijevic, J., Mitic, M., Vukovic, N., Plemic, A., Miljkovic, Z., Babic, B., "The ant lion optimization algorithm for flexible process planning. *JPE*, 18(2), 65-68, 2015.
- [31] N. Chopra and S. Mehta, "Multi-objective optimum generation scheduling using Ant Lion Optimization," 2015 Annual IEEE India Conference (INDICON), New Delhi, 2015, pp. 1-6.
- [32] Gupta, E., Saxena, A., "Performance Evaluation of Antlion Optimizer Based Regulator in Automatic Generation Control of Interconnected Power System", *Journal of Engineering*, 2016.
- [33] Babers, R., Ghali, N. I., Hassanien, A. E., Madbouly, N. M., "Optimal community detection approach based on Ant Lion Optimization", In *Computer Engineering Conference (ICENCO)*, 2015, 11th International (pp. 284-289). IEEE.
- [34] Nair, S. S., Rana, K. P. S., Kumar, V., Chawla, A., "Efficient Modeling of Linear Discrete Filters Using Ant Lion Optimizer", *Circuits, Systems, and Signal Processing*, 1-34,2016.
- [35] Rebecca, N., Shin, M., MH, S., Zuriani, M., "Ant Lion Optimizer for Optimal Reactive Power Dispatch Solution", *Journal of Electrical Systems*, (3), 67-74,2015.
- [36] Martinez-Sykora, A., Alvarez-Valdes, R., Bennell, J. A., Ruiz, R. and Tamarit, J. M., "Metaheuristics for the irregular bin packing problem with free rotations," *Eur. J. Oper. Res.*, vol. 258, no. 2, pp. 440–455, 2017.
- [37] Christensen, H. I., Khan, A., Pokutta, S. and Tetali, P., "Approximation and online algorithms for multidimensional bin packing: A survey," *Computer Science Review*, vol. 24. pp. 63–79, 2017.
- [38] Yarpiz, Bin Packing Problem using GA, PSO, FA, and IWO. <http://yarpiz.com/363/ypap105-bin-packing-problem>, Accessed 20 April 2018.