



## Hybrid flower pollination algorithm approach for the two-dimensional bin packing problem

Harun Gezici<sup>1\*</sup>, Haydar Livatyalı<sup>2</sup>

<sup>1</sup>Electronics and Automation Department, Kırklareli University, Kırklareli, 39100, Turkey

<sup>2</sup>Mechatronics Engineering Department, Yıldız Technical University, İstanbul, 34349, Turkey

### Highlights:

- Study of two-dimensional bin packaging problem
- Suggesting a hybrid meta heuristic optimization algorithm
- Modification of flower pollination algorithm

### Keywords:

- Bin packing problem
- Flower pollination algorithm
- Genetic algorithm
- Combinatorial optimization
- Hybridization

### Article Info:

Research Article  
Received: 06.07.2020  
Accepted: 16.10.2021

### DOI:

10.17341/gazimmfd.764853

### Correspondence:

Author: Harun Gezici  
e-mail:  
harun.gezici@klu.edu.tr  
phone: +90 288 214 1845

### Graphical/Tabular Abstract

Meta-heuristic algorithms are frequently used in the solution of the two-dimensional bin packaging problem (2DBPP). This is because meta heuristic algorithms reach acceptable solutions in reasonable time in cases where there are many examples. In this study, a new hybrid meta heuristic algorithm is proposed for the solution of 2DBPP (2DHFPGA). The proposed algorithm was created by hybridizing the flower pollination algorithm (FPA) and genetic algorithm (GA). In 2DBPP, n small rectangular objects are placed inside larger rectangles (containers). The aim is to minimize the number of containers required for packaging all small objects. There are two rules for classifying 2DBPP. The first one is orientation and the second one is the cut type. This study focuses on the 2DBPPRF problem where there is no guillotine cut and objects can be rotated.

Table A. Comparison of the results.

Algorithm	Class (10)	Subgroups (50)	Friedman test values
BRKGA	1 (class 2)	27	2.7
SVC2BPRF	1 (class 7)	20	4.2
CS	2 (class 5, 9)	23	3.75
BA	1 (class 8)	20	4.35
MA	1 (class 2)	20	4.4
RSMP	0	16	6
2DHFPGA	6 (class 1-4,6,10)	33	2.6

**Purpose:** Minimum container numbers have not been reached in 2DBPP with meta heuristic optimization algorithms yet. The aim of this study is to get better results than the previous studies.

### Theory and Methods:

The global search procedure of the proposed algorithm is carried out with FPA. The local search operator in the structure of FPA has been modified. The variable in this operator representing the local pollination distance was removed from the equation and the Levy distribution was added instead. With this change, FPA's global search capability has been increased. The local search procedure of the proposed algorithm is carried out with 4 different mutation operators of GA. These are swap, scramble, reversion and displacement.

### Results:

The proposed algorithm has been tested with a data set with 10 classes and 50 subgroups. The obtained results were compared with 6 meta-heuristic algorithms which have previously published. Average container numbers of each class and subgroup were used as comparison parameters. In addition, Friedman test was performed to compare the performances of the algorithms. In the first column of Table A, shows the names of the algorithms. the number of classes and subgroups that they have successfully solved can be seen in the second and third columns and the average rank values obtained from the Friedman test are given in the last column.

### Conclusion:

2DFPGA has been the most successful algorithm by achieving the best results in the 6 classes of the data set. According to these data, the proposed algorithm achieved the best results in 60% of the data set classes and achieved 3 times more success than its closest competitor. In addition, the proposed algorithm was the most successful algorithm in 33 of the 50 subgroups by achieving the best results. Considering these data, the proposed algorithm successfully resolved 66% of the subgroups and achieved 12% more successful results than its closest competitor. According to the average rank values obtained from the Friedman test, the proposed algorithm was the most successful algorithm with 2.6 scores.



## İki boyutlu kutu paketleme probleminin çözüm için hibrit çiçek tozlaşma algoritması yaklaşımı

Harun Gezici<sup>1\*</sup>, Haydar Livatyalı<sup>2</sup>

<sup>1</sup>Kırklareli Üniversitesi, Teknik Bilimler Meslek Yüksek Okulu, Elektronik ve Otomasyon Bölümü, 39100 Merkez Kırklareli, Türkiye

<sup>2</sup>Yıldız Teknik Üniversitesi, Makine Fakültesi, Mekatronik Mühendisliği Bölümü, 34349, Beşiktaş İstanbul, Türkiye

### Ö N E Ç İ K A N L A R

- İki boyutlu kutu paketleme problemi ele alınmıştır
- Melez meta-sezgisel optimizasyon algoritması önerilmektedir
- Çiçek tozlaşma algoritması modifiye edilmiştir

#### Makale Bilgileri

Araştırma Makalesi  
Geliş: 06.07.2020  
Kabul: 16.10.2021

#### DOI:

10.17341/gazimmfd.764853

#### Anahtar Kelimeler:

Kutu paketleme problemi,  
çiçek tozlaşma algoritması,  
genetik algoritma,  
kombinatoriyal optimizasyon,  
hibritleme

#### ÖZ

İki boyutlu kutu paketleme problemi (2BKPP), kesme ve paketleme problemlerinin (KPP) bir alt dalıdır. Araştırmacılar, 2BKPP'nin çözümünde meta-sezgisel algoritmaları sıklıkla kullanmaktadırlar. Bunun nedeni meta-sezgisel algoritmaların çok sayıda örneğin bulunduğu vakalarda kabul edilebilir çözümlere makul sürelerde ulaşmasıdır. Bu çalışmada, 2BKPP'nin çözümü için yeni bir melez meta-sezgisel algoritma önerilmektedir. Önerilen algoritma çiçek tozlaşma (ÇTA) ile genetik algoritmanın (GA) hibritlenmesiyle oluşturulmuştur. ÇTA'nın global arama kabiliyetini geliştirmek için yerel arama operatöründe değişiklik yapılmıştır. Bu çalışma kapsamında önerilen algoritma, son yıllarda yayımlanan altı farklı meta-sezgisel algoritma ile karşılaştırılmıştır. Karşılaştırma için 10 sınıf, 50 alt grup ve 500 örneğin bulunduğu bir veri seti kullanılmıştır. Her bir sınıfın ve her bir alt grubun, ortalama konteyner değerleri karşılaştırma parametresi olarak kullanılmıştır. Ayrıca, algoritmaların birbirlerine göre performanslarını karşılaştırmak için Friedman testi uygulanmıştır. Önerilen algoritma veri setinin 10 sınıfının 6'sında ve 50 alt grubunun 33'ünde en başarılı sonuçları elde etmiş, Friedman testinde ise 2,6 skor ile en başarılı algoritma olmuştur. Elde edilen sonuçlar önerilen meta-sezgisel algoritmanın geçerliliğini teyit etmektedir.

## Hybrid flower pollination algorithm approach for the two-dimensional bin packing problem

### H I G H L I G H T S

- Study of two dimensional bin packaging problem
- Suggesting a hybrid meta-heuristic optimization algorithm
- Modification of flower pollination algorithm

#### Article Info

Research Article  
Received: 06.07.2020  
Accepted: 16.10.2021

#### DOI:

10.17341/gazimmfd.764853

#### Keywords:

Bin packing problem,  
flower pollination algorithm,  
genetic algorithm,  
combinatorial optimization,  
hybridization

#### ABSTRACT

Two-dimensional bin packing problem (2DBPP) is a sub-branch of cutting and packaging problems (CPP). Researchers often use meta-heuristic algorithms in the solution of 2DBPP, because in cases where a large number of samples exist, meta-heuristic algorithms reach acceptable solutions in reasonable time. In this article, a novel hybrid meta-heuristic algorithm is proposed for the solution of 2DBPP. The proposed algorithm combines flower pollination algorithm (FPA) and genetic algorithm (GA). In order to improve FPA's global search capability, FPA's local search operator is modified. The proposed algorithm has been compared to six recently published meta-heuristic algorithms. For comparison, a data set containing 10 classes, 50 subgroups and 500 samples was used. Average container values of each class and of each subgroup were used as comparison parameters. In addition, Friedman test was used to evaluate the relative performance of the algorithms. The proposed algorithm achieved the most successful results in 6 of the 10 classes of the data set and 33 of the 50 subgroups. In the Friedman test, it was the most successful algorithm with a score of 2.6. The results confirm the validity of the proposed meta-heuristic algorithm.

## 1. GİRİŞ (INTRODUCTION)

Kesme ve paketleme problemleri (KPP), endüstri mühendisliği, bilgisayar mühendisliği ve yöneylem araştırmaları vb. disiplinlerde gerçek dünya problemlerine uyarlanabilir [1]. KPP, kombinatorial optimizasyon (eniyeleştirme) problemlerindedir [2-4]. İki boyutlu kutu paketleme problemi (2BKPP), np-hard problem sınıfındadır [5, 6]. Küçük objelerin büyük konteynerlere yerleştirme işlemi aynı zamanda metal, cam, ahşap gibi büyük levhalardan küçük objelerin kesilmesi işlemine de uyarlanabilir [7]. Cam, ahşap, kağıt, tekstil, otomotiv vb. sanayi dallarında ham maddenin optimum kullanımına yönelik çalışmalarda 2BKPP'den faydalanılabilir [1, 8, 9]. 2BKPP, çevrim içi ve çevrim dışı olmak üzere iki kısma ayrılmaktadır. Problemin çevrim-dışı türü için paketlenen objelerin genişliği, yüksekliği ve sırası gibi bilgiler bilinmektedir. Bu bilgilere göre sezgisel yöntemler ile yerleşim sıralaması manipüle edilebilir. Problemin çevrim-içi türünde ise paketlenen bir sonraki obje hakkında bilgi bulunmamaktadır. Bir sonraki kutu beklenmeden mevcut kutunun uygun bir yere yerleştirilmesi gerekir [10].

2BKPP'de paketlenen objelerin döndürülerek (90°) veya döndürülmeden yerleştirilmesi de belirleyici bir süreç adıdır. Genellikle objelerin döndürülmesi daha iyi sonuçlar vermektedir [10]. Döndürme kararı probleme özgü olarak tanımlanmaktadır. Örneğin bir sac levha üzerinden kesim yapılırken döndürmenin bir sakıncası yoktur. Ancak desenli bir kumaşın kesilmesi işleminde döndürme istenmeyebilir.

2BKPP sınıflandırmak için iki kural bulunmaktadır. Bunlardan birincisi yönlendirir. Yerleştirilecek olan objenin 90 derece döndürülüp döndürülmeyeceğini ifade eder. İkincisi ise kesme türüdür. Kesme işleminde giyotin kesimin gerekli olup olmadığını ifade eder. Bu iki kuralın kombinasyonu ile 2BKPP dört farklı grupta sınıflandırılabilir [8, 11]. Bu gruplar şöyledir.

- 2BKPP|Y||G|: Ögeler yönlendirilir (Y) ve giyotin kesim gereklidir (G).
- 2BKPP |D||G|: Ögeler 90 derece döndürülebilir (D) ve giyotin kesim gereklidir (G).
- 2BKPP |Y||GY|: Ögeler yönlendirilir (Y) ve giyotin kesim gerekli değildir (GY).
- 2BKPP |D||GY|: Ögeler 90 derece (D) döndürülebilir ve giyotin kesim gerekli değildir (GY). [8]

Bu çalışmada giyotin kesimin olmadığı, yerleştirilecek ürünlerin döndürülebildiği 2BKPPDGY problemi üzerinde durulmaktadır. Bu problem, sabit konteyner ölçülü iki boyutlu kutu paketleme problemi (2-D single bin size bin packing problem) sınıfına girmektedir [12, 13]. 2BKPP araştırmacılar tarafından kesin, sezgisel ve meta-sezgisel yöntemlerle çözülmeye çalışılmıştır. Kesin yöntemler ile diğer yöntemlere göre daha iyi sonuçlar elde edilse de obje sayısının fazla olduğu vakalarda çözüm süreleri çok

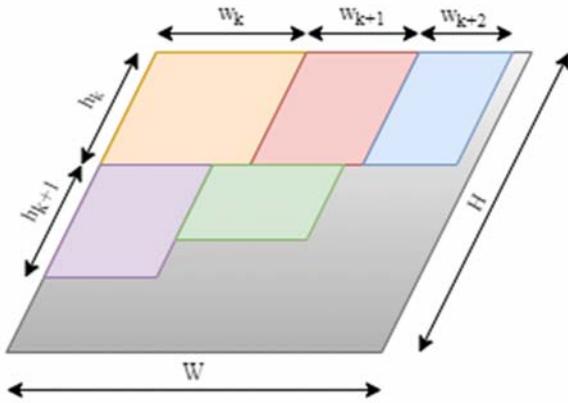
uzundur. Sezgisel ve meta-sezgisel yöntemler ise en iyi sonucu garanti etmezler. Ancak obje sayısının fazla olduğu vakalarda kabul edilebilir çözümleri daha kısa sürelerde bulabilirler [14-16]. Son zamanlarda yapılan çalışmalarda, sezgisel ve meta-sezgisel algoritmaların verimliliklerinin arttığı görülmektedir. 2BKPP'nin çözümü için kullanılan erken dönem sezgisel ve meta-sezgisel algoritmalar üzerine kapsamlı bir inceleme Hopper vd. [17] tarafından sunulmuştur. Daha yakın zamanlarda yapılan çalışmalar ise şöyledir. Thomas ve Chaudhari [18] genetik algoritma (GA) ile yeni bir uygunluk fonksiyonunu birleştirerek algoritmanın yakınsama hızını artırmışlardır. Chen vd. [19] 2BKPP'nin çözümü için GA ile parçacık sürüsü optimizasyonu (PSO) algoritmalarını hibritlemişlerdir. GA ve PSO parametreleri bazı örnekler üzerinde deney yapılarak belirlenmiştir. Hong vd. [20] tavlama benzetimi (TB) ile en uygun algoritmasını birleştirerek melez bir yapı ortaya çıkarmışlardır. Yazarlar elde edilen sonuçların mevcut tekniklerden elde edilen sonuçlardan daha iyi olduğunu belirtmişlerdir. Gomez ve Terashima-Marin [21] son zamanlarda araştırmacıların odak noktası haline gelen hiper-sezgisel bir algoritma önermişlerdir. Önerilen bu algoritma vakanın durumuna göre algoritma seçimi yapmaktadır. Önerilen hiper-sezgisel algoritma GA, Pareto Evrimsel Algoritması ve Diferansiyel Evrim Algoritmasından oluşmaktadır. Yazarlar önerilen algoritmanın, bir dizi sezgisel algoritmadan daha iyi sonuçlar elde ettiğini bildirmişlerdir. Zheng vd. [22] 3BKPP'yi çözmek için rasgele anahtarlamalı GA yöntemini kullanmışlardır. Ayrıca yazarlar algoritmalarına bulanık mantık denetleyicisini de adapte etmişlerdir. Beyaz vd. [10] bir başka hiper-sezgisel algoritma önermişlerdir. Bu algoritma GA ve memetik algoritma (MA) temeline dayanmaktadır. Yazarlar önerdikleri algoritmaya ayrıca sonlu sonraki sığdır, sonlu ilk sığdır ve en uygun azalan yükseklik gibi sezgisel algoritmaları da dâhil etmişlerdir. Sim vd. [23] öğrenebilen bir hiper-sezgisel algoritma sunmuşlardır. Önerilen algoritma her vakadan sonuçlar çıkararak yeni vakalar için farklı sezgisel yöntemler geliştirmektedir. Virk ve Singh [1] doğa tabanlı meta-sezgisel algoritmalarından guguk kuşu arama algoritması (GKA), yarasa algoritması (YA) ve çiçek tozlaşma algoritması (ÇTA) kullanmıştır. Son zamanlarda popüler hale gelen bu algoritmaların KPP üzerindeki performanslarını incelemişlerdir. Bennell vd. [24] yenilikçi bir çaprazlama operatörü kullanan bir GA sunmuşlardır. Önerilen algoritmada yerleştirme yönergesi olarak en uygun sezgisel yöntemi kullanılmıştır. Thapatsuwan vd. [25] yapay bağışıklık sistemi (YBS), PSO ve GA yönergelerini içeren bir algoritma sunmuşlardır. Yazarlar YBS'nin daha iyi sonuçlar vermiş olmasına rağmen GA'ya göre süre dezavantajına sahip olduğunu bildirmişlerdir. Gonçalves ve Resende [26] 2BKPP ve 3BKPP'nin çözümü için rasgele anahtarlamalı bir GA sunmuşlardır. López-Camacho vd. [27] bir dizi sezgisel algoritmayı problemin türüne göre seçen hiper-sezgisel bir algoritma önermişlerdir. 2BKPP alanının da yayınlanmış bu çalışmalarda henüz minimum konteyner sayılarına ulaşamamıştır. Son yıllardaki araştırmalar özellikle GA, tabu arama, PSO ve TB gibi meta-

sezgisel algoritmalarla yoğunlaşmıştır [1, 5]. Bu çalışmada son yıllarda keşfedilen ve farklı alanlarda kullanımı yaygınlaşmakta olan ÇTA ile 2BKPP çözümü gerçekleştirilmiştir. ÇTA'nın performansını artırmak için GA ile hibritlenmiştir. Önerilen algoritmaya, iki boyutlu hibrit çiçek tozlaşma genetik algoritması (2BHÇTGA) adı verilmiştir. Yerel arama yönergesi GA'nın dört farklı mutasyon operatörü ile gerçekleştirilmektedir. Global arama yönergesi ise ÇTA ile gerçekleştirilmektedir. Global arama kabiliyetinin iyileştirilmesi için ÇTA'nın yerel arama yönergesi modifiye edilmiştir. Önerilen algoritma geniş bir veri seti ile test edilerek altı meta-sezgisel algoritma ile kıyaslanmıştır. Karşılaştırma parametresi olarak ortalama konteyner sayısı ve Friedman testinden elde edilmiş ortalama rank değeri kullanılmıştır. Alınan sonuçlar önerilen algoritmanın başarısını ortaya koymaktadır.

Bölüm 2'de 2BKPP tanımlanmaktadır. Bölüm 3'te geliştirilen algoritma hakkında bilgi verilmektedir. Bölüm 4'te veri setine ait bilgiler açıklanmakta ve karşılaştırmalı sonuçlar verilmektedir. Bölüm 5'te sonuçlar tartışılmaktadır.

## 2. İKİ BOYUTLU KUTU PAKETLEME PROBLEMİ (TWO DIMENSIONAL BIN PACKING PROBLEM)

2BKPP'de,  $n$  adet küçük dikdörtgen obje vardır ve bu objelerin genişlik  $w_k$  ve yükseklik  $h_k$  ölçüleri bilinmektedir ( $k \in \{1, \dots, n\}$ ). Bu küçük objeler, genişliği  $W$  yüksekliği  $H$  olan, daha büyük bir dikdörtgen objenin (konteyner) içerisine yerleştirilir (Şekil 1).



Şekil 1. 2BKPP konteyner yerleşimi  
(2DBPP container placement)

Yerleştirme işlemi ortogonal şekilde olmalıdır. Burada amaç, objeler üst üste gelmeyecek şekilde konteynerlere yerleştirmek ve tüm objelerin paketlenmesi için gerekli olan konteyner sayısını minimum yapmaktır [10,28–30]. 2BKPP ile ilgili tanımlayıcı denklemler Beyaz vd. [10] çalışmasından alınmıştır. Denklemlerde kullanılan kısaltmalar şu şekildedir.  $C$  toplam konteyner sayısıdır.  $w_i, h_i$ ;  $i$ . kutunun genişliği ve yüksekliğidir.  $W_j, H_j$ ;  $j$ . konteynerin genişliği ve yüksekliğidir,  $x_i, y_i$ ;  $i$ . kutunun sol alt köşesinin koordinatıdır.  $w_i^x, w_i^y$ ;  $i$ . kutunun X eksenine paralel genişliğidir,  $h_i^x, h_i^y$ ;  $i$ . kutunun Y eksenine paralel

yüksekliğidir.  $le_{ik}$ ;  $i$ . kutu,  $k$ . kutunun sol kenarına yerleştirilir.  $ri_{ik}$ ;  $i$ . kutu,  $k$ . kutunun sağ tarafına yerleştirilir.  $un_{ik}$ ;  $i$ . kutu,  $k$ . kutunun alt tarafına yerleştirilir.  $ab_{ik}$ ;  $i$ . kutu,  $k$ . kutunun üst tarafına yerleştirilir.  $p_{ij}$ ;  $i$ . kutu,  $j$ . konteynera yerleştirildi ise  $p_{ij} 1$ , değilse 0'dır.  $c_j$ ;  $j$ . konteyner kullanıldı ise  $c_j 1$  değilse 0'dır. Toplam konteyner sayısını Eş. 1 ile ifade edilir.

$$C = \text{Min} \sum_{j=1}^n c_j \quad (1)$$

Eş. 2-Eş. 6, konteynerlere yerleştirilecek olan kutuların üst üste gelmemesini ifade eder.

$$x_i + (w_i w_i^x) + (h_i h_i^x) \leq x_k + (1 - le_{ik}), \forall i, k \ i < k \quad (2)$$

$$x_k + (w_k w_k^x) + (h_k h_k^x) \leq x_i + (1 - ri_{ik}), \forall i, k \ i < k \quad (3)$$

$$y_i + (w_i w_i^y) + (h_i h_i^y) \leq y_k + (1 - un_{ik}), \forall i, k \ i < k \quad (4)$$

$$y_k + (w_k w_k^y) + (h_k h_k^y) \leq y_i + (1 - ab_{ik}), \forall i, k \ i < k \quad (5)$$

$$le_{ik} + ri_{ik} + un_{ik} + ab_{ik} \leq p_{ij} + p_{kj} - 1, \forall i, k \ i < k \quad (6)$$

Eş. 7, her bir kutunun sadece bir konteynera paketlenmesini ifade eder.

$$\sum_{j=1}^c p_{ij} = 1, \forall i \quad (7)$$

Eş. 8, İçerisine bir kutu yerleştirildiğinde o konteynerin kullanıldığını ifade eder.

$$\sum_{i=1}^B p_{ij} \leq M c_j, \forall j \quad (8)$$

Eş. 9, Eş. 10, konteynerdeki tüm kutuların konteynerin boyutunu aşmamasını ifade eder.

$$x_i + (w_i w_i^x) + (h_i h_i^x) \leq W_j + (1 - p_{ij})M, \forall i, j \quad (9)$$

$$y_i + (w_i w_i^y) + (h_i h_i^y) \leq H_j + (1 - p_{ij})M, \forall i, j \quad (10)$$

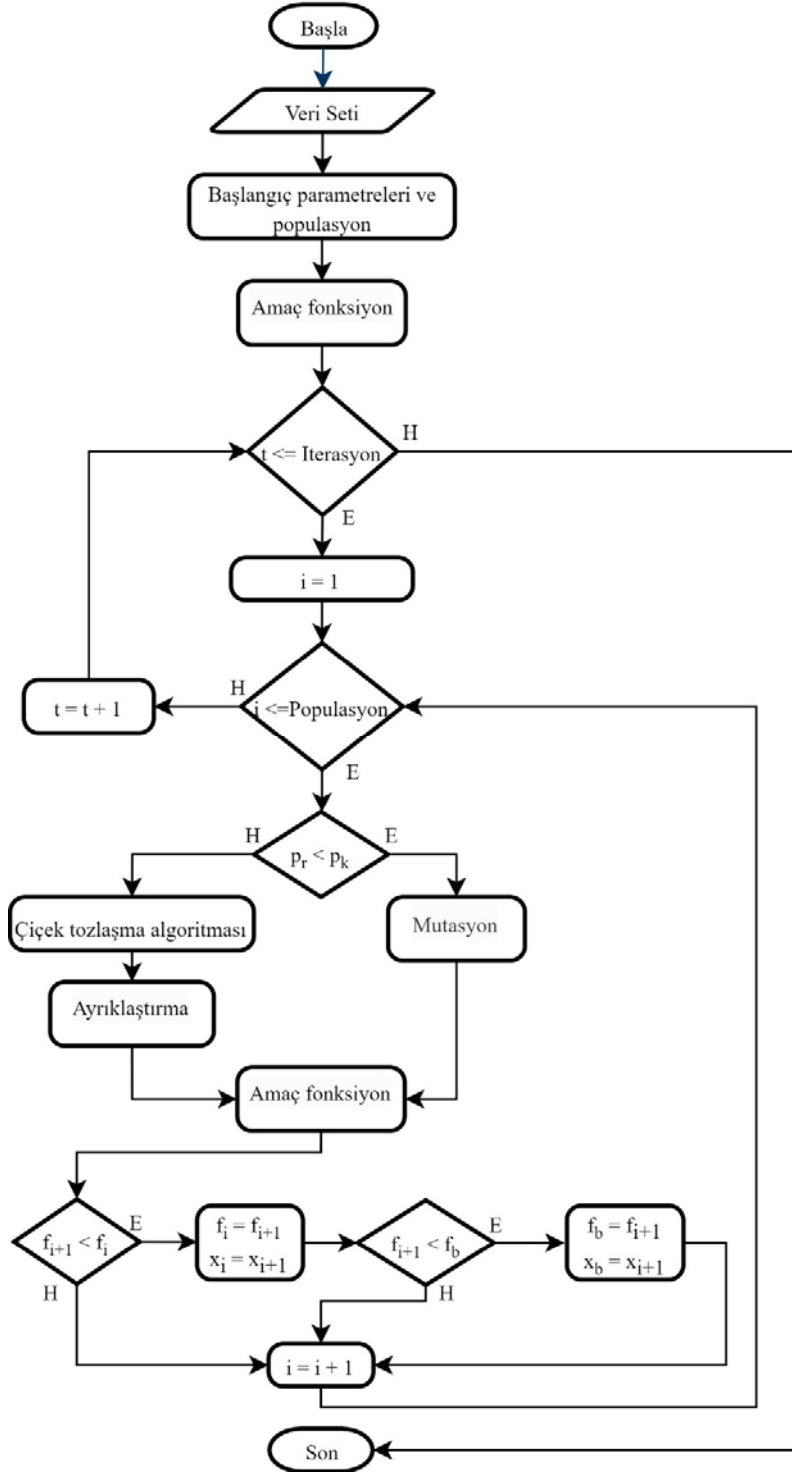
Eş. 11, Eş. 12, değişkenlerin değer aralıklarını açıklamaktadır.

$$w_i^x, w_i^y, h_i^x, h_i^y, le_{ik}, ri_{ik}, un_{ik}, ab_{ik}, p_{ij}, c_j \in 0,1, \forall i, k \ i < k \quad (11)$$

$$x_i, y_i \geq 0, \forall i \quad (12)$$

## 3. ÖNERİLEN ALGORİTMANIN İKİ BOYUTLU KUTU PAKETLEME PROBLEMİNE UYGULANMASI (IMPLEMENTATION OF THE PROPOSED ALGORITHM TO THE 2-D BIN PACKING PROBLEM)

Önerilen algoritma, ÇTA, mutasyon, uygunluk fonksiyonu ve yerleştirme stratejisi olmak üzere 4 ana aşamadan oluşmaktadır. Önerilen algoritmanın akış şeması Şekil 2'de



Şekil 2. 2BHÇTGA akış diyagramı (2DHFPGA flowchart)

verilmiştir. Önerilen algoritma veri setinin okunmasıyla başlar. Ardından popülasyon büyüklüğü ve olasılık değeri ( $p_k$ ) gibi başlangıç parametreleri belirlenir. Olasılık değeri yapılan bir dizi denemeden sonra 0,7 olarak bulunmuştur. Her iterasyonda 0 ile 1 arasında rastgele geçici olasılık değeri belirlenir ve  $p_r$  ile sembolize edilir.  $p_r$ ,  $p_k$ 'dan büyük ise global arama, küçük ise yerel arama yapılır. Önerilen

algoritmada global arama işlemi ÇTA ile gerçekleştirilir. Yerel arama ise GA'nın dört farklı mutasyon operatörleri ile gerçekleştirilmektedir. Başlangıç parametreleri belirlendikten sonra başlangıç popülasyonu rasgele oluşturulur. Başlangıç popülasyonunun uygunluk değerleri amaç fonksiyon ile hesaplanır. En iyi uygunluk değerine sahip olan birey elit birey olarak seçilir. Daha sonra her bir

birey için olasılık değerine göre global arama veya yerel arama prosedürleri uygulanır. Yeni türetilen popülasyonun tekrar uygunluk değerleri hesaplanır. Her iterasyonda türetilen yeni bireylerin uygunluk değerleri bir önceki bireyden daha iyi ise yeni birey eski bireyin yerini alır. Elit bireyden, daha iyi uygunluk değerine sahip bir birey türetilmiş ise o birey elit birey olarak belirlenir. Popülasyondaki tüm bireyler için bu işlemler uygulandıktan sonra iterasyon sayısı bir artırılır ve her iterasyonda bu işlemler tekrar edilir. Programın durma koşulu maksimum iterasyondur. Akış diyagramında, yerleştirme yönergesi ile amaç fonksiyonu bir bütün olarak kabul edilmiş ve amaç fonksiyonu olarak belirtilmiştir. Akış diyagramında kullanılan bazı kısaltmalar şu şekildedir; t iterasyon,  $x_i$ ; mevcut birey,  $x_{i+1}$ ;  $x_i$ 'den türetilen birey (yeni birey),  $f_i$ ;  $x_i$ 'nin uygunluk değeri,  $f_{i+1}$ ;  $x_{i+1}$  bireyin uygunluk değeri,  $x_b$ ; en iyi uygunluk değerine sahip birey (elit birey),  $f_b$ ;  $x_b$ 'in uygunluk değeridir.

### 3.1. Çiçek Tozlaşma Algoritması (Flower Pollination Algorithm)

ÇTA, birçok meta-sezgisel optimizasyon algoritması [31] gibi doğadan ilham alınarak tasarlanmıştır. Çiçeklerin tozlaşabilmesi için polen transferi gerçekleştirmeleri gerekmektedir. Bu polen transferinin matematiksel modeli ÇTA olarak tanımlanır. Birçok meta-sezgisel algoritmada olduğu gibi ÇTA'da global arama ve yerel arama yönergelerini içerir. Bu arama yönergeleri de çiçeklerin tozlaşma şekillerinin 2 ana gruba indirilmesi ile modellenmiştir. Çiçekler, tozlaşma yönergelerini gerçekleştirebilmek için tozlayıcılara ihtiyaç duyarlar. İki tür tozlayıcı bulunmaktadır. Bunlardan birincisi yarası, sinek, arı gibi biyotik tozlayıcılardır. Bu tozlayıcılar çiçeklerin polenlerini uzak mesafelere taşıyabilir. Ayrıca bu tozlayıcıların uçuş rotaları Levy dağılımı ile modellenmektedir. İkinci tip tozlayıcı ise su ve rüzgâr gibi abiyotik tozlayıcılardır. Bu tozlaşma şeklinin kısa mesafelerde gerçekleştiği kabul edilir. Tozlaşma işlemi ise kendi kendine tozlaşma ve çapraz tozlaşma olmak üzere iki gruba ayrılarak modellenmiştir. Kendi kendine tozlaşma bir bitkinin aynı çiçeklerinin ya da farklı çiçeklerinin birbirlerini tozlaması olarak tanımlanır. Çapraz tozlaşma ise farklı bitkilerin çiçeklerinin birbirlerini tozlaması olarak kabul edilir. Kendi kendine tozlaşma ÇTA'da yerel arama yönergesi olarak kullanılırken, biyotik çapraz tozlaşma global arama yönergesi olarak tanımlanmıştır. Kendi kendine tozlaşma sürecinde, çiçeklerin birbirlerine yakınlıkları ve rüzgâr gibi bazı etmenler vardır. Bu etmenler nedeniyle yerel arama süreci genel arama süreci içerisinde daha fazla ağırlığa sahiptir. Daha ayrıntılı bilgi için şu çalışmalara bakılabilir [32-34]. ÇTA'nın global arama yönergesinin matematiksel ifadesi Eş. 13, Eş. 14'teki gibidir.

$$x_i^{t+1} = x_i^t + L(x_i^t - x_{best}) \quad (13)$$

$$L \sim \frac{\lambda \Gamma(\lambda) \sin\left(\frac{\pi\lambda}{2}\right)}{\pi} \frac{1}{s_1^{1+\lambda}} \quad s \gg s_0 > 0 \quad (14)$$

Formüldeki t iterasyon sayısını, i polen demeti veya çözüm vektörü indisini,  $x_i$  çözüm vektörünü,  $x_{best}$ , mevcut jenerasyondaki tüm çözüm vektörlerinin en iyisini temsil etmektedir. L, tozlayıcıların uçuş rotalarının matematiksel modeli olan Levy uçuşudur.  $\Gamma(\lambda)$  standart gama fonksiyonu, s ise levy adımlarının birbirlerinden mesafe olarak büyük farklılıklarının olduğunu göstermektedir. Yerel tozlaşma ise Eş. 15'teki gibi formüle edilebilir.

$$x_i^{t+1} = x_i^t + \varepsilon(x_j^t - x_k^t) \quad (15)$$

Denklemden  $x_j^t$  ve  $x_k^t$  aynı bitkinin farklı çiçeklerinin polen demetlerini yani çözüm kümesinin farklı çözümlerini temsil etmektedir.  $\varepsilon$  ise rastgele bir yerel tozlaşma mesafesini temsil eder ve 0-1 arasında normal bir dağılıma sahiptir. Yerel tozlaşmanın mı yoksa küresel tozlaşmanın mı yapılacağı anahtarlama olasılığı  $p \in [0, 1]$  ile kontrol edilir.

Bu çalışmada önerilen algoritmanın yerel arama yönergesi GA'nın mutasyon operatörleri ile gerçekleştirilmektedir. Global arama ise ÇTA ile gerçekleştirilmektedir. Yapılan bir kısım ön denemelerden edinilen tecrübe ile önerilen algoritmanın performansını artırmak için ÇTA'nın yerel arama yönergesinde değişikliğe gidilmiştir. İlgili denklemde; rastgele belirlenen yerel tozlaşma mesafesini temsil eden  $\varepsilon$  yerine Levy dağılımı kullanılmıştır. Oluşturulan bu yeni yerel arama operatörüne melez yerel arama operatörü adı verilmiştir. Bu yapılan değişiklikler önerilen algoritmanın global arama kabiliyeti artırılmıştır. Yapılan değişiklikler sonucunda ortaya çıkan yeni ifade Eş. 16'da verilmiştir.

$$x_i^{t+1} = x_i^t + L(x_j^t - x_k^t) \quad (16)$$

### 3.2. Ayırıklaştırma (Discretization)

ÇTA, içerisindeki Levy dağılımı çarpımı nedeniyle sürekli formda bir meta-sezgisel algoritmadır. Bu algoritmanın kombinatoriyal optimizasyon problemi olan KPP'de kullanılabilmesi için ayrık forma dönüştürülmesi gerekmektedir. Literatürde önerilen ayrık forma dönüştürme yöntemlerinden 4 tanesi seçilmiştir. Bu yöntemler, En Büyük Dereceli Değer (EBDD - The Largest Ranked Value) [35], En Küçük Pozisyon Değeri (EKPD - The Smallest Position Value) [36], En Büyük Sıra Değeri (EBSD - The Largest Order Value) [37], Sıralama Değeri (SD - The Rank Order Value) [18]. Şekil 3'te 5 elemanlı sürekli formdaki bir dizinin ayrık forma dönüştürülmesi örneği verilmiştir. SD yönteminde dizinin en küçük elemanına 1 en büyük elemanına 5 değeri verilmektedir. SD yöntemi, sürekli formdaki kümeyi ayrık forma dönüştürürken çözüm vektörünün dizilimine diğer yöntemler kadar müdahale etmemektedir. Yapılan ön denemeler sonucunda SD yöntemi, önerilen algoritma ile kullanıldığında, ortalama konteyner sayısı parametresinde daha iyi sonuçlar verdiği tespit edilmiştir. Bu nedenle ayırıklaştırma işleminde bu yöntem kullanılmıştır.

### 3.3. Uygunluk Fonksiyonu (Fitness Function)

Amaç fonksiyonunun hesaplanabilmesi için önerilen algoritma ile sırası belirlenen kutuların konteynerlere yerleştirilmesi gerekmektedir. Bu yerleştirme işlemi iki aşamada yapılmaktadır. İlk olarak en az, sıradaki kutunun alanı kadar boş alana sahip konteynerler belirlenir. Daha sonra bu konteynerler boş alanlarına göre küçükten büyüğe doğru sıralanır. İkinci aşamada ise sıradaki kutu yerleşebileceği ilk uygun konteynerin ilk uygun konumuna yerleştirilir. KPP'nin çözümünde, uygunluk fonksiyonunun çıktısı olarak konteyner sayısı kullanılır ise bu, algoritma durgunluğu adı verilen bir soruna yol açabilir. Bunun nedeni birden fazla dizilimin aynı konteyner sayısını verebilecek olmasıdır. Bu nedenle algoritmanın verimliliğini artırmak için konteyner doluluk oranının uygunluk fonksiyonu çıktısı olarak kullanılması daha avantajlı olacaktır. Konteyner doluluk oranı Eş. 17 ve Eş. 18 ile hesaplanır.

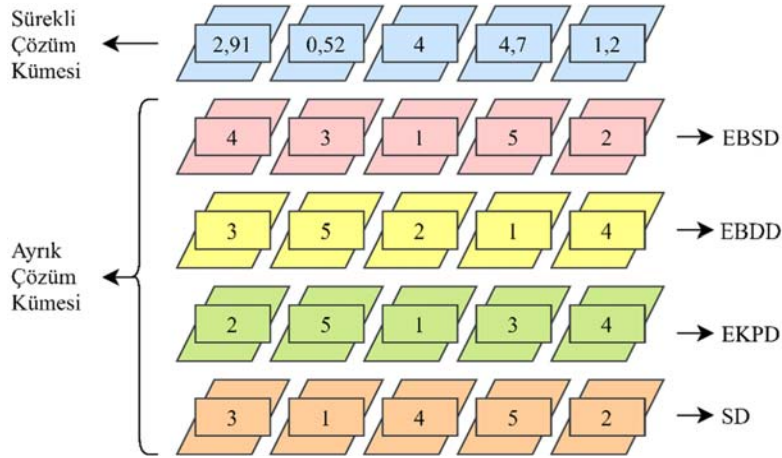
$$DO_i = \frac{\sum_{j=1}^n w_j h_j}{WH} \quad (17)$$

$$ekF = 1 - \left( \frac{\sum_{i=1}^m (DO_i)^2}{m} \right) \quad (18)$$

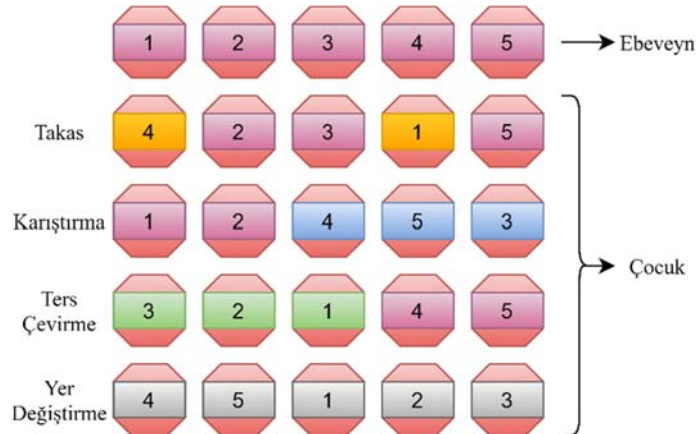
Burada  $n$  dikdörtgen kutuların sayısını,  $w_j$   $j$ . dikdörtgen kutunun genişliğini,  $h_j$   $j$ . dikdörtgen kutunun yüksekliğini ( $j \in \{1, \dots, n\}$ ),  $W$  konteynerin genişliğini,  $H$  konteynerin yüksekliğini,  $DO_i$   $i$ . konteynerin doluluk oranını,  $m$  kullanılan toplam konteyner sayısını,  $ekF$  ise uygunluk fonksiyonunu göstermektedir.

### 3.4. Mutasyon Aşaması (Mutation Phase)

Önerilen algoritmada mutasyon işlemi 4 farklı yöntemle gerçekleştirilmektedir. Bu mutasyon yöntemleri takas, karıştırma, ters çevirme ve yer değiştirmedir. Takas yönteminde, rastgele belirlenen iki kutunun yeri değiştirilir [38]. Karıştırma yönteminde, rastgele bir alt dizi seçilir ve seçilen dizideki kutuların yerleri rastgele değiştirilir [39]. Ters çevirme yönteminde, rastgele bir alt dizi seçilir ve bu alt dizideki kutuların yerleri ters çevrilir [40]. Yer değiştirme yönteminde, ise rastgele belirlenen bir alt dizi rastgele belirlenen başka bir yere yerleştirilir [41]. Her iterasyonda yukarıda belirtilen yöntemlerden bir tanesi seçilerek mutasyon işlemi gerçekleştirilir. Seçim işlemi  $p_m \in [0, 1]$  anahtarlama olasılığı ile gerçekleştirilmektedir. Şekil 4'te mutasyon yöntemlerine ait örnekler verilmiştir.



Şekil 3. Ayrıklaştırma (Discretization).



Şekil 4. Mutasyon operatörleri (Mutation operators)

#### 4. DENEYSEL SONUÇLAR (EXPERIMENTAL RESULTS)

Bu bölümde önerilen algoritmanın deneysel sonuçları sunulmaktadır. Önerilen algoritma Python dilinde kodlanmıştır. Hesaplamalar 8 GB RAM, 2,30 GHz Intel Core i7 işlemcili, Linux Mint işletim sistemi ile çalışan bir bilgisayarda yapılmıştır. Önerilen algoritmada popülasyon büyüklüğü 50 ve iterasyon sayısı 20 olarak seçilmiştir. Önerilen algoritma 10 kere çalıştırılmış ve en iyi sonuçlar karşılaştırma yapabilmek için kaydedilmiştir. Deneyler için literatürde bulunan 2 adet veri seti kullanılmıştır. Birinci veri seti Berkey-Wang veri setidir [42]. Berkey-Wang veri seti 6 sınıftan oluşmaktadır (sınıf 1-6). İkinci veri seti Martello-Vigo veri setidir[43]. Martello-Vigo veri seti 4 sınıftan oluşmaktadır (sınıf 7-10). Martello-Vigo veri setinde 4 farklı tipte kutu bulunmaktadır ve kutuların tipleriyle ilgili bilgiler tablo 1 'de verilmiştir. Tablo 1'de  $w$ , kutu genişliği,  $h$ , kutu yüksekliği,  $W$ , konteyner genişliği,  $H$ , konteyner yüksekliğidir.

**Tablo 1.** Sınıf 7-10 kutu tipleri (Class 7-10 box type)

Tip	Kutu Ölçüsü	
1	$w \in [(2/3)W, W]$	$h \in [1, (1/2)H]$
2	$w \in [1, (1/2)W]$	$h \in [(2/3)H, H]$
3	$w \in [(1/2)W, W]$	$h \in [(1/2)H, H]$
4	$w \in [1, (1/2)W]$	$h \in [1, (1/2)H]$

Tablo 2'de deneylerde kullanılan veri setiyle ilgili bilgiler verilmektedir. Kutu boyutu etiketli sütunlarda kutuların hangi kurallara göre türetildiği belirtilmektedir.

Deneylerde kullanılan veri setinde, 10 sınıf ve her sınıfın 5 alt grubu vardır. Bu alt gruplar kutu sayısına göre belirlenmiştir. Tablo 3'te  $n$  ile gösterilen sütun her bir alt grubun kaç adet kutu içerdiğinin göstermektedir. Her bir alt grup, aynı sayıda kutuya sahip 10 adet örnek içermektedir ve veri seti toplamda 500 adet örnekten oluşmaktadır. Diğer algoritmalar ile karşılaştırmak için, kıyaslama parametresi olarak veri setinin 50 alt grubunun ve 10 sınıfının ortalama konteyner sayıları kullanılmıştır. Önerilen algoritma, ön tanımlı rasgele-anahtarlı genetik algoritma (ÖTRAGA - biased random key genetic algorithm) [26], rotasyonsuz

ardışık sıra değeri düzeltmeli iki boyutlu kutu paketleme (RASDD2BKPP - sequential value correction two bin packing rotational free) [8], GKA, YA [5], MA [30] ve kalan alanı maksimize eden paketleme (KAMP - residual space maximized packing) [11] algoritmaları ile karşılaştırılmıştır. Tablo 3'te önerilen algoritmanın ve karşılaştırmak için seçilen algoritmanın sonuçları verilmektedir. Tablo 3'te birinci sütunda veri setinin sınıf bilgisi, ikinci sütunda her bir örnekteki kutu sayısı, üçüncü ve sekizinci sütunlar arasında karşılaştırma için seçilen algoritmaların sonuçları, dokuzuncu sütunda önerilen algoritmanın sonuçları ve son sütunda önerilen algoritmanın çözüm süreleri milisaniye cinsinden verilmiştir.

Tablo 3 incelendiğinde önerilen algoritma, 50 alt grubun 33'ünde en iyi çözüme ulaşırken, bu sayı ÖTRAGA için 27, RASDD2BKPP için 20, GKA için 23, YA için 20, MA için 20 ve KAMP için 16'dır. Önerilen algoritma, 10 adet sınıfın 6'sında en iyi çözüme ulaşırken (sınıf 1-4,6,10), bu sayı ÖTRAGA için 1 (sınıf 2), RASDD2BKPP için 1 (sınıf 7), GKA için 2 (sınıf 5, 9), YA için 1 (sınıf 8), MA için 1 (sınıf 2) ve KAMP için 0'dır. Şekil 5'te bazı örneklerin konteyner yerleşimleri verilmiştir.

Algoritmaların genel performans değerlendirmesini yapabilmek için parametrik olmayan Friedman testi uygulanmıştır [40]. Friedman testi yapılırken sınıfların ortalama konteyner değerleri kullanılmıştır. Friedman testi ikiden fazla örneğin bulunduğu durumlarda uygulanır ve tek yönlü ANOVA'nın parametrik olmayan türüdür. Friedman testi bağımlı gözlemlerde uygulanan parametrik olmayan bir testtir. Şekil 6'da Friedman testinden elde edilmiş ortalama rank değerleri verilmiştir. Şekil 6 incelendiğinde önerilen algoritmanın en iyi sonucu elde ettiği görülmektedir.

#### 5. SONUÇLAR VE TARTIŞMALAR (RESULTS AND DISCUSSIONS)

Bu çalışmada 2BKPP'nin çözümü için 2BHÇTGA algoritması önerilmiştir. Önerilen algoritma meta-sezgisel algoritma olan ÇTA ile GA'nın hibritlenmesi ile oluşturulmuştur ve literatürde son yıllarda yayınlanan altı meta-sezgisel algoritma ile karşılaştırılmıştır. Karşılaştırma yapmak için 10 sınıf, 50 alt-grup ve 500 örnekten oluşan Berkey-Wang ve Martello-Vigo veri setleri kullanılmıştır.

**Tablo 2.** Veri seti bilgileri (Dataset information)

Sınıf	Konteyner Boyutu	Kutu Boyutu ( $w, l$ )	Sınıf	Konteyner Boyutu	Kutu Boyutu ( $w, l$ )
1	10x10	Kesikli düzgün dağılım [1,10]	6	100x100	Kesikli düzgün dağılım [1,10]
2	30x30	Kesikli düzgün dağılım [1,10]	7	100x100	%70 tip 1 %10 tip 2,3,4
3	40x40	Kesikli düzgün dağılım [1,10]	8	100x100	%70 tip 2 %10 tip 1,3,4
4	100x100	Kesikli düzgün dağılım [1,10]	9	100x100	%70 tip 3 %10 tip 1,2,4
5	100x100	Kesikli düzgün dağılım [1,10]	10	100x100	%70 tip 4 %10 tip 1,2,3

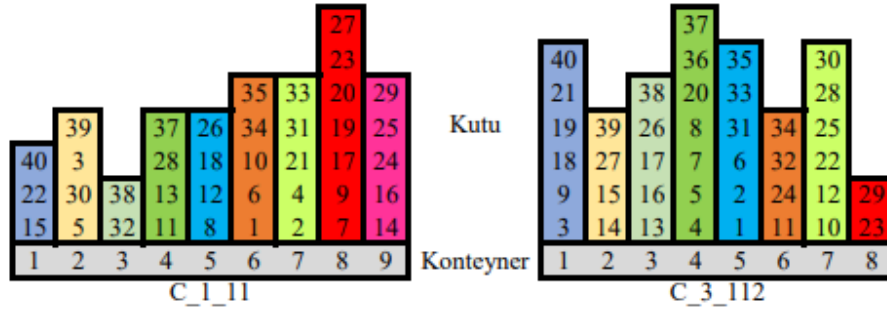


**Tablo 3.** Ortalama konteyner sayıları (Average bin numbers)

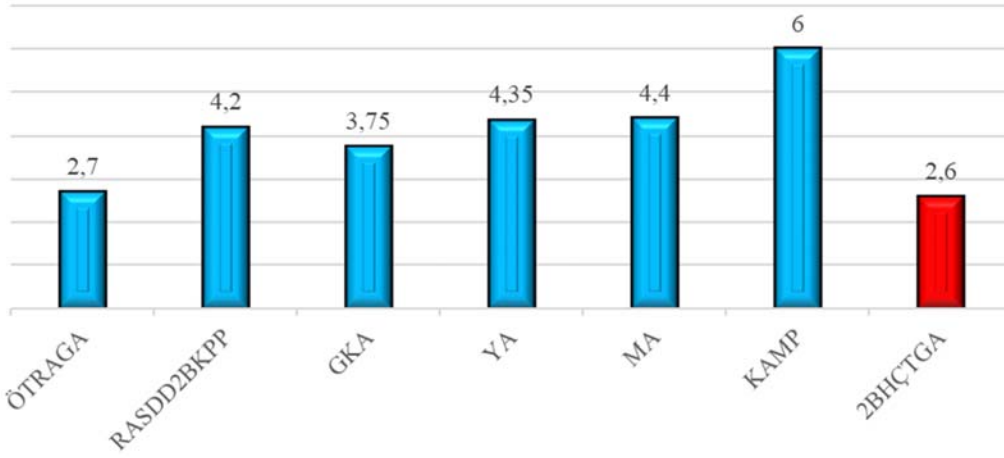
Sınıf	n	ÖTRAGA	RASDD2DBPP	GKA	YA	MA	KAMP	2BHÇTGA	Süre
1	20	6,6	6,6	6,6	6,6	6,6	6,7	6,6	1,352
	40	12,8	12,8	12,4	12,9	12,8	13,1	12,8	4,432
	60	19,5	19,5	19,4	19,5	19,5	19,6	19,2	9,061
	80	27	27	27	27	26,7	27,1	26,8	14,990
	100	31,3	31,3	31,4	31,4	31,3	31,4	31,3	24,984
	Ort.	19,44	19,44	19,36	19,48	19,38	19,58	19,34	10,964
2	20	1	1	1	1	1	1	1	3,884
	40	1,9	2	2	2	1,9	2	1,9	20,510
	60	2,5	2,5	2,5	2,5	2,5	2,5	2,5	49,342
	80	3,1	3,2	3,1	3,2	3,1	3,1	3,1	87,147
	100	3,9	4	3,9	4	3,9	3,9	3,9	140,662
	Ort.	2,48	2,54	2,5	2,54	2,48	2,5	2,48	60,309
3	20	4,7	4,7	4,7	4,7	4,7	4,7	4,7	1,692
	40	9,2	9,2	9	9,2	9,2	9,5	9	5,467
	60	13,4	13,4	13,5	13,5	13,4	13,6	13,3	10,805
	80	18,2	18,4	18,2	18,6	18,4	18,6	18,1	17,476
	100	22	22	21,9	22	22,1	22,2	22	27,621
	Ort.	13,5	13,54	13,46	13,6	13,56	13,72	13,42	12,612
4	20	1	1	1	1	1	1	1	3,868
	40	1,9	1,9	1,9	1,9	1,9	1,9	1,9	19,344
	60	2,3	2,5	2,5	2,5	2,5	2,5	2,3	46,376
	80	3,1	3,2	3	3,3	3,1	3,2	3	80,447
	100	3,7	3,9	3,8	3,9	3,8	3,7	3,7	131,245
	Ort.	2,4	2,5	2,44	2,52	2,46	2,46	2,38	56,256
5	20	5,9	5,9	5,9	5,9	5,9	5,9	5,9	1,531
	40	11,4	11,4	11	11,5	11,4	11,5	11,2	4,769
	60	17,2	17,3	17,5	17,5	17,4	17,5	17,2	9,605
	80	23,9	23,9	24	23,9	23,9	24	23,7	15,581
	100	27,7	27,7	27	27	27,9	28,3	27,5	24,799
	Ort.	17,22	17,24	17,08	17,16	17,3	17,44	17,1	11,257
6	20	1	1	1	1	1	1	1	3,858
	40	1,6	1,7	1,8	1,8	1,9	1,8	1,5	20,184
	60	2,1	2,1	2,1	2,1	2,1	2,1	2,1	49,500
	80	3	3	3	3	3	3	3	86,981
	100	3,2	3,4	3,5	3,4	3,4	3,3	3,2	142,265
	Ort.	2,18	2,24	2,28	2,26	2,28	2,24	2,16	60,558
7	20	5,2	5,2	5,1	5,2	5,2	5,4	5	1,754
	40	10,2	10,4	10,1	10,5	10,2	10,7	10,2	5,657
	60	14,6	14,5	14,9	14,6	14,6	15,3	14,6	12,173
	80	20,8	20,8	21,2	20,9	21,1	21,9	21,2	19,473
	100	25	25	25,4	25,2	25,3	26,2	25,6	30,230
	Ort.	15,16	15,14	15,34	15,28	15,28	15,9	15,32	13,857
8	20	5,3	5,3	5,4	5,3	5,3	5,4	5,3	1,759
	40	10,3	10,3	10,4	10,1	10,4	10,7	10,2	5,457
	60	14,7	14,7	15,2	14,8	14,8	15,5	14,7	11,847
	80	20,4	20,5	20,6	20,3	20,8	21,4	21,1	19,230
	100	25,2	25,3	26	24,9	25,7	26,4	25,9	29,864
	Ort.	15,18	15,22	15,52	15,08	15,4	15,88	15,44	13,631
9	20	14,3	14,3	14,3	14,3	14,3	14,3	14,3	1,153
	40	27,5	27,5	26,8	27,3	27,5	27,5	27,7	3,487
	60	43,5	43,5	43,6	43,5	43,5	43,5	43,6	7,206
	80	57,3	57,3	57,5	57,3	57,3	57,3	57,5	12,086
	100	69,3	69,3	69,3	69,3	69,3	69,3	69,4	18,350
	Ort.	42,38	42,38	42,3	42,34	42,38	42,38	42,5	8,456
10	20	4,1	4,1	4	4,1	4,1	4,2	4	2,070
	40	7,2	7,2	7,4	7	7,3	7,3	7,1	6,846
	60	9,9	9,9	10	9,9	9,9	10	9,7	15,732
	80	12,5	12,6	12,6	12,7	12,8	13	12,3	27,024
	100	15,4	15,5	15,8	16	15,7	16,1	15,4	39,682
	Ort.	9,82	9,86	9,96	9,94	9,96	10,12	9,7	18,271

Karşılaştırma parametresi olarak sınıfların ve alt-grupların ortalama konteyner sayıları kullanılmıştır. 2BHÇTGA veri

setinin 6 sınıfında en iyi sonuçları elde etmiştir. İkinci en iyi algoritma ise GKA algoritmasıdır ve iki sınıfta daha iyi



Şekil 5. Konteyner yerleşimleri (Container placement)



Şekil 6. Friedman test değerleri (Friedman test values)

sonuçlar sağlamıştır. Bu veriler dikkate alındığında önerilen algoritma, veri seti sınıflarının %60'ında en iyi sonuçları elde etmiş ve en yakın rakip algoritmadan 3 kat fazla sınıfta başarı sağlamıştır. Ayrıca, 50 alt-grubun 33'ünde en iyi sonuçları elde ederek en başarılı algoritma olmuştur. Alt-gruplarda ikinci en iyi algoritma ÖTRAGA olup, 27 alt-grubu başarılı bir şekilde çözmüştür. Bu veriler dikkate alındığında önerilen algoritma tüm alt grupların %66'sını başarı ile çözmüş ve en yakın rakibinden %12 daha başarılı sonuçlar elde etmiştir. Diğer yandan, rakip algoritmalar, sınıf 7, 8 ve 9'da önerilen algoritmadan daha başarılı olmuştur. Bu sınıflarda bulunan kutuların büyüklükleri konteyner boyutlarına yakındır. Bu durum önerilen algoritmanın görece küçük boyutlu kutuların bulunduğu veri setlerinde daha başarılı olduğunu göstermektedir.

Algoritmaların birbirlerine göre performanslarını değerlendirmek için Friedman testi uygulanmıştır. Test sonuçlarına göre önerilen algoritma 2,6 puan ile en başarılı algoritma olmuştur. Tüm bu sonuçlar önerilen hibrit meta-sezgisel algoritmanın geçerliliğini doğrulamıştır. Bu çalışma meta-sezgisel algoritmaların KPP'nin çözümlerini giderek daha da iyileştirebileceğini göstermektedir. Ayrıca yeni keşfedilen meta-sezgisel optimizasyon algoritmalarının paketleme problemlerinin sonuçlarını olumlu yönde geliştirdiğini göstermektedir. Yeni keşfedilen meta-sezgisel

algoritmaların daha önce keşfedilen algoritmalar ile hibritlenmesi, KPP'nin çözüm kalitesini artırmaktadır.

ÇTA'nın performansını artırmak için global arama ve yerel arama yönergelerinin yeniden düzenlenmesi düşünülebilir. Bu düzenleme gerçekleştirilirken meta-sezgisel algoritmaların genelinde bulunan rastgele değişkenler ÇTA'ya eklenebilir. Meta-sezgisel algoritmaların verimliliği için global ve yerel arama yönergeleri arasındaki denge önemlidir. ÇTA'da bu denge olasılık anahtarı P ile sağlanmaktadır ve P değeri sabittir. Ancak optimizasyon problemleri birbirlerinden farklı özellikleri sahiptir. Bazı optimizasyon problemleri meta-sezgisel algoritmaların global arama yöngesini test ederken bazıları da yerel arama yöngesini test eder. Bu nedenle ÇTA'da bulunan P değerinin dinamik olarak belirlenmesi algoritmanın performansını artırabilir. P değerinin belirlenmesinde kaotik harita yöntemlerinin kullanılması ve en verimli kaotik haritanın belirlenmesi önemli sonuçlar ortaya koyabilir.

#### KAYNAKLAR (REFERENCES)

1. Virk A.K., Singh K., Application of Nature Inspired Algorithms to Optimize Multi-objective Two-Dimensional Rectangle Packing Problem, Journal of Industrial Integration and Management, 04 (04), 1950010, 2019.

2. Stoyan Y., Pankratov A., Romanova T., Cutting and packing problems for irregular objects with continuous rotations: mathematical modelling and non-linear optimization, *Journal of the Operational Research Society*, 67 (5), 786–800, 2017.
3. Cui Y., Song X., Chen Y., Cui Y.-P., New model and heuristic solution approach for one-dimensional cutting stock problem with usable leftovers, *Journal of the Operational Research Society*, 68 (3), 269–280, 2017.
4. Cui Y.-P., Yao Y., Zhang D., Applying triple-block patterns in solving the two-dimensional bin packing problem, *Journal of the Operational Research Society*, 69 (3), 402–415, 2018.
5. Virk A.K., Singh K., Solving Two-Dimensional Rectangle Packing Problem Using Nature-Inspired Metaheuristic Algorithms, *Journal of Industrial Integration and Management*, 03 (02), 1850009, 2018.
6. Grandcolas S., Pinto C., A New Search Procedure for the Two-dimensional Orthogonal Packing Problem, *Journal of Mathematical Modelling and Algorithms in Operations Research*, 14 (3), 343–361, 2015.
7. Lodi A., Monaci M., Pietrobuoni E., Partial enumeration algorithms for Two-Dimensional Bin Packing Problem with guillotine constraints, *Discrete Applied Mathematics*, 217, 40–47, 2017.
8. Cui Y.P., Cui Y., Tang T., Sequential heuristic for the two-dimensional bin-packing problem, *European Journal of Operational Research*, 240 (1), 43–53, 2015.
9. Dahmani N., Krichen S., Ghazouani D., A variable neighborhood descent approach for the two-dimensional bin packing problem, *Electronic Notes in Discrete Mathematics*, 47, 117–124, 2015.
10. Beyaz M., Dokeroglu T., Cosar A., Robust hyperheuristic algorithms for the offline oriented/non-oriented 2D bin packing problems, *Applied Soft Computing Journal*, 36, 236–245, 2015.
11. Wang Y., Chen L., Two-dimensional residual-space-maximized packing, *Expert Systems with Applications*, 42 (7), 3297–3305, 2015.
12. Dyckhoff H., A typology of cutting and packing problems, *EJOR*, 44 (2), 145–159, 1990.
13. Wäscher G., Haußner H., Schumann H., An improved typology of cutting and packing problems, *European Journal of Operational Research*, 183 (3), 1109–1130, 2007.
14. Oliveira J.F., Neuenfeldt A., Silva E., Carravilla M.A., A survey on heuristics for the two-dimensional rectangular strip packing problem, *Pesquisa Operacional*, 36 (2), 197–226, 2016.
15. Abdel-Basset M., El-Shahat D., El-Henawy I., Solving 0–1 knapsack problem by binary flower pollination algorithm, *Neural Computing and Applications*, 31 (9), 5477–5495, 2019.
16. Akyol S., Alataş B., Automatic mining of accurate and comprehensible numerical classification rules with cat swarm optimization algorithm, *Journal of the Faculty of Engineering and Architecture of Gazi University*, 31 (4), 840–857, 2016.
17. Hopper E., Turton B.C.H., An Empirical investigation of meta-heuristic and heuristic algorithms for a 2D packing problem, *European Journal of Operational Research*, 128 (1), 34–57, 2001.
18. Thomas J., Chaudhari N.S., A new metaheuristic genetic-based placement algorithm for 2D strip packing, *Journal of Industrial Engineering International*, 10 (1), 1–16, 2014.
19. Chen W., Zhai P., Zhu H., Zhang Y., Hybrid algorithm for the two-dimensional rectangular layer-packing problem, *Journal of the Operational Research Society*, 65 (7), 1068–1077, 2014.
20. Hong S., Zhang D., Lau H.C., Zeng X., Si Y.-W., A hybrid heuristic algorithm for the 2D variable-sized bin packing problem, *European Journal of Operational Research*, 238 (1), 95–103, 2014.
21. Gomez J.C., Terashima-Marín H., Evolutionary hyperheuristics for tackling bi-objective 2D bin packing problems, *Genetic Programming and Evolvable Machines*, 19 (1–2), 151–181, 2018.
22. Zheng J.N., Chien C.F., Gen M., Multi-objective multi-population biased random-key genetic algorithm for the 3-D container loading problem, *Computers and Industrial Engineering*, 89, 80–87, 2015.
23. Sim K., Hart E., Paechter B., A lifelong learning hyperheuristic method for bin packing, *Evolutionary Computation*, 23 (1), 37–67, 2015.
24. Bennell J.A., Soon Lee L., Potts C.N., A genetic algorithm for two-dimensional bin packing with due dates, *International Journal of Production Economics*, 145 (2), 547–560, 2013.
25. Thapatsuwan P., Pongcharoen P., Hicks C., Chainate W., Development of a stochastic optimisation tool for solving the multiple container packing problems, *International Journal of Production Economics*, 140 (2), 737–748, 2012.
26. Gonçalves J.F., Resende M.G.C., A biased random key genetic algorithm for 2D and 3D bin packing problems, *International Journal of Production Economics*, 145 (2), 500–510, 2013.
27. López-Camacho E., Terashima-Marín H., Ochoa G., Conant-Pablos S.E., Understanding the structure of bin packing problems through principal component analysis, *International Journal of Production Economics [Internet]*, 145 (2), 488–499, 2013.
28. Dokeroglu T., Bayir M.A., Cosar A., Robust heuristic algorithms for exploiting the common tasks of relational cloud database queries, *Applied Soft Computing Journal*, 30, 72–82, 2015.
29. Dokeroglu T., Cosar A., Optimization of one-dimensional Bin Packing Problem with island parallel grouping genetic algorithms, *Computers and Industrial Engineering*, 75 (1), 176–186, 2014.
30. Fernández A., Gil C., Baños R., Montoya M.G., A parallel multi-objective algorithm for two-dimensional bin packing with rotations and load balancing, *Expert Systems with Applications*, 40 (13), 5169–5180, 2013.
31. Aydılek İ.B., Multi-class classification with modified firefly optimization algorithm, *Journal of the Faculty of Engineering and Architecture of Gazi University*, 32 (4), 1097–1108, 2017.

32. Yang X.-S., Karamanoglu M., He X., Flower pollination algorithm: A novel approach for multiobjective optimization, *Engineering Optimization*, 46 (9), 1222–1237, 2014.
33. Ram J.P., Babu T.S., Dragicevic T., Rajasekar N., A new hybrid bee pollinator flower pollination algorithm for solar PV parameter estimation, *Energy Conversion and Management* [Internet], 135, 463–476, 2017.
34. Salgotra R., Singh U., Application of mutation operators to flower pollination algorithm, *Expert Systems with Applications* [Internet], 79, 112–129, 2017.
35. Liang J.J., Pan Q.K., Tiejun C., Wang L., Solving the blocking flow shop scheduling problem by a dynamic multi-swarm particle swarm optimizer, *International Journal of Advanced Manufacturing Technology* [Internet], 55 (5), 755–762, 2011.
36. Tasgetiren M.F., Liang Y.C., Sevkli M., Gencyilmaz G., Particle swarm optimization and differential evolution for the single machine total weighted tardiness problem, *International Journal of Production Research*, 44 (22), 4737–4754, 2006.
37. Qian B., Wang L., Hu R., Wang W.L., Huang D.X., Wang X., A hybrid differential evolution method for permutation flow-shop scheduling, *International Journal of Advanced Manufacturing Technology*, 38, 757–777, 2008.
38. Deep K., Mebrahtu H., Nagar A.K., Novel GA for metropolitan stations of Indian railways when modelled as a TSP, *International Journal of Systems Assurance Engineering and Management*, 9 (3), 639–645, 2018.
39. Adrian A.M., Utamima A., Wang K.J., A comparative study of GA, PSO and ACO for solving construction site layout optimization, *KSCE Journal of Civil Engineering*, 19 (3), 520–527, 2015.
40. Abdel-Basset M., Manogaran G., Abdel-Fatah L., Mirjalili S., An improved nature inspired meta-heuristic algorithm for 1-D bin packing problems, *Personal and Ubiquitous Computing*, 22 (5–6), 1117–1132, 2018.
41. Durmaz E.D., Şahin R., NSGA-II and goal programming approach for the multi-objective single row facility layout problem, *Journal of the Faculty of Engineering and Architecture of Gazi University*, 32 (3), 941–955, 2017.
42. Berkey J.O., Wang P.Y., Two-Dimensional Finite Bin-Packing Algorithms, *The Journal of the Operational Research Society*, 38 (5), 423–429, 1987.
43. Martello S., Vigo D., Exact solution of the two-dimensional finite bin packing problem, *Management Science*, 44 (3), 388–399, 1998.