VERİ BİLİMİ DERGİSİ
www.dergipark.gov.tr/veri

Hibrit Karga-Genetik Algoritmasını Kullanarak 3 Boyutlu Kutu Paketleme Problemi Çözme

Hazem Khudeer[1]*, Hasan Erbay[2]

[1]*Kırıkkale Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği, Kırıkkale*
[2] *Türk Hava Kurumu Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği, Ankara*

Makale Bilgisi

Başvuru:
17/07/2020
Kabul:
27/10/2020

Özet

Kutu Paketleme Problemi'ni çözmek için çeşitli yaklaşımlar ve çözümler geliştirilmiştir. Bu çalışmada ise Üç Boyutlu Kutu Paketleme Problemi için Hibrit Meta Sezgisel Algoritma'lar geliştirilmiştir. Bu hibrit algoritma, Karga Arama Algoritması ve Genetik Algoritma'nın bir kombinasyonudur. Üç Boyutlu Küboid Kutu Paketleme Problemi ilk kez Karga Algoritma ve Hibrit Karga-Genetik Algoritma kullanılarak çözülmüştür. Üç Boyutlu KPP'lerin özellikleri ; 90 derece dönebilen küboid kutular, sabit genişlik ve uzunluk ve yüksekliği değişebilen konteynerdir. Bu problemi çözmek için daha önce elde edilmiş karşılaştırma veri setimiz olmadığı için bu veri seti rastgele oluşturulmuştur. Yaptığımız çalışmada Karga Algoritma, Genetik Algoritma, Hibrit Karga-Genetik Algoritma kullanılarak test edilmiş ve sonuçlar gözlemlenmiştir. Bu sonuçlara bakılarak görülüyor ki Hibrit Karga-Genetik Algoritma'nın daha iyi sonuçlar verdiği görülmüştür.

Anahtar Kelimeler: *Kombinatoryal Problemleri. Kutu Paketleme Problemi, Sezgisel, Meta-Sezgisel, Karga Arama Algoritma, Genetik algoritma, Optimizasyon, Tedarik zinciri.*

Solving 3d Bin Packing Problem Using Hybrid Crow-Genetic Algorithm

Abstract

Various approaches and solutions have been developed to solve the Bin Packaging Problem (BPP). In this study, Hybrid Meta-Heuristic Algorithms have been developed for the Bin Packaging Problem. It is a hybrid algorithm, a combination of the Crow Search Algorithm and Genetic Algorithm. 3D Bin Packaging Problem was solved for the first time by using Crow Algorithm and Hybrid Crow-Genetic Algorithm. Properties of Three-dimensional BPPs; Cuboid boxes that can rotate 90 degrees, and containers with fixed-width and length, but changin[1]g height. Since there wasn't a benchmark dataset available to test the solution to the problem, the data was generated

[1] * İletişim e-posta: hazemkhd@hotmail.com
** Bu çalışmanın bir kısmı III. International Conference on Data Science and Applications 2020'de sözlü olarak sunulmuştur.

randomly. There were different algorithms developed throughout the study, based on the Crow Algorithm, Genetic Algorithm, Hybrid Crow-Genetic Algorithm. The results stated that the Hybrid Crow-Genetic Algorithm performed the best.

Keywords: Supply Chain, Bin Packing Problem (BPP) 3D, Combinatorial Optimization, Hybrid Meta-Heuristic, Crow Search algorithm, Genetic Algorithm.

1 Giriş

Kombinatoryal problemler yapay zekanın önemli bir parçasıdır. Kutu Paketleme Problemi (KPP) bu problemlerden biridir. KPP problemi kısaca, bir çok kutu bir konteyner içine en az boşluk bırakarak yerleştirmek ile ilgilenir. KPP'nin birçok çeşidi vardır. Ancak en anlamlı ve zorlayıcı olanı, farklı boyutlardaki küboid şekilli öğelerin dörtgensel biçimde konteynerlere paketlendiği 3 boyutlu olanıdır. Meta-sezgisel algoritmaları kullanarak kombinatoryal problemlerde daha verimli sonuçlar elde etmek [1], bu algoritmaların tercih edilmesinin sebeplerinden biridir. Ötedenberi, KPP birçok farklı yaklaşımla çözülmeye çalışılmaktadır [2,3], ancak bu çalışma kapsamında kullanılan Karga Algoritma ve Hibrit Karga Genetik Algoritma'larının kullanılması literatüre farklı yaklaşımlar sunmaktadır. Karganın yiyecek arayışındaki yüksek zekası bu yöntemi seçmenin ana nedenidir.

KPP, Tedarik Zinciri sektöründe önemli bir yer almaktadır ve Araç Yönlendirme Sorunu ile yakından ilişkilidir. Kargo şirketleri tarafından da kullanılır. KPP, nakliye maliyetlerini düşürme etkisi nedeniyle taşıma sektöründe birinci önceliktedir.

Bu sorunun karmaşık ve çok geniş çözüm alanı vardır. Bu tip sorunlara NP-Zor denir [4]. 3 boyutlu KPP, sıradan yöntemler veya Kaba Kuvvet (Brute-Force) algoritması kullanarak özellikle girdi boyutunun büyük olduğu durumlarda pratikte çözülememektedir. Bunun temel nedeni küboid şekilli dönebilen n adet kutunun bir konteynera

(n! * 6 ^ n) farklı biçimde yerleştirilebilir olmasıdır. Öte yandan verilerin temsili ise başka bir zorluktur. Ayrıca, algoritmanın cevaplama süresi kritiktir. Bazı durumlarda birkaç saniye içinde bir karar vermek gerekir. Çünkü gecikme nakliye sürecinde aşırı maliyetlidir. Örneğin, nakliye uçağının 1 dakika gecikmesi yaklaşık 60$'dır. Bu çalışmanın katkısı, 3 boyutlu KPP çözümünde ilk kez Karga Algoritmasının [5] kullanılmasıdır. Bu çalışmada, 3boyutlu KPP Genetik Algoritma , Karga Algoritma, Hibrid Genetik-Karga Algoritma ve Karga-Genetik Algoritma gibi farklı algoritmalar kullanılarak çözülmüştür. Kullanılan yöntemlerde hem rastgele örnekler hem de optimum çözümü bilinen bir örnek üzerinden karşılaştırma yapılmıştır. Bu çalışmanın odaklandığı yaklaşım, yüksekliği değişken olan bir konteynırın hacmini mümkün olduğunca en aza indirgemektir. Değerlendirme ölçütümüz kullanılan konteynırın hacmi ile alakalıdır. Dolayısıyla bir konteynerde ne kadar az hacim kullanılarak yerleşim yapılırsa o kadar iyi sonuç elde edilmiş olur.

Çalışmanın geriye kalan kısmında aşağıdaki gibi organize edilmiştir. Çalışmanın ikinci bölümünde, KPP'nin tanımı, çeşitleri, kriterleri ve çözüm yöntemleriden bahsedilmiştir. Kullanılan algoritmalar ve veri temsili hakkında bilgi verilmiştir. Ayrıca literatür taraması da yapılmıştır. Çalışmanın üçüncü bölümünde, KPP'nin bu çalışmada kullanılan tibi ve kriterlerinden bahsedilmiştir. Buna ek olarak bahsedilen algoritmaların KPP üzerine pratik olarak nasıl uygulanabileceği gösterilmiştir. Çalışmanın son bölümünde ise elde edilen sonuçlar sunulmuş ve analiz edilmiştir.

2 Materyal ve Metod

2.1 Kutu Paketleme Problemi

KPP basitçe, bir yada birden fazla konteyner içine kutuların en iyi şekilde nasıl yerleştirilmesi gerektiği ile ilgilenir. Bu şekilde konteyner içinde minimum miktarda boş alan bırakarak, kullanılan konteyner sayısının mümkün olduğunca azalmasını sağlamak temel amaçtır. Kutu Paketleme Problemi klasik ve popüler bir optimizasyon problemidir. 1970'lerden bu yana KPP birçok araştırmacının ilgisini çekmiştir ve bu konu ile alakalı bazı başarılar elde edilmiştir [2-4].

2.1.1 Kutu Paketleme Problemi Tipleri

KPP bir çok çeşidi vardır. Problemin boyutlarına göre

3 çeşide ayırılbilir. Birincisi bir boyutlu KPP, örneğin

bir zinciri en az atık ile istenilen uzunluğa göre bölme problemi. İkincisi ise 2 boyutlu KPP, iki boyutlu öğelerle ilgilenir, örneğin en az atıkla ahşap yada cam kesme problemi düşünülebilir.

Üçüncü çeşidi ise üç boyutlu KPP problemlerinin en kamaşık tipi olarak tanımlanır. Bu çalışma üç boyutlu KPP ile ilgilenir.

2.1.2 KPP kriterleri

KPP kriterleri, kutuları yerlerşirirken göz önünde bulundurulması gereken kısıtlardır. KPP kriterleri uygulama alanına ve yüklenecek öğelere göre değişir. Bu kriterler oryantasyon, çevrimiçi/çevrimdışı ve ağırlıktır. Oryantasyon, kutuların kaç farklı şekilde döndürebilindiğini belirtir. Çevrimiçi/çevrimdışı ise yerleştirilecek kutular daha önce bilinip bilinmediğini belirtir. Kutular ağırlığı ve konteynerin taşıyabildiği ağırlık çok önemli bir kriterdir.

2.2 Sezgisel algoritmalar

KPP'lerin hesaplamalı olarak zor olması, makul bir sürede kabul edilebilir sonuçlar elde etmek için çok sayıda sezgisel yaklaşıma yol açmıştır [3, 6]. Sezgisel algoritmalarda bulunan çözümler çoğu zaman kesin olarak en uygun çözümü vermemektedir ve bulunan çözümün en uygun çözüm olup olmadığına da emin olunanamamaktadır. Ayrıca, optimum çözüme yakınlığı hesaplanamamakta veya garanti edilememektedir. Bununla birlikte, düşük hesaplama süreleri göz önüne alındığınıp, kesin yöntemlerle karşılaştırıldığında büyük kombinatoryal optimizasyon problemlerinin büyük örnekleri için etkili bir çözümleme stratejisi olarak kabul edilmektedir. Bu yöntemlere Yaklaşım Algoritmaları da denebilir. KPP için Yaklaşım Algoritmaları iki kategoride sınıflandırılabilir. Belirli bir sırayla öğeleri dikkate alan ve bunları kutuların içine tek tek yerleştiren yönteme çevrimiçi yöntem denir. Diğer sınıf çevrimdışı algoritmaları içerir. Öğeleri boyuta göre sıralayarak verilen öğe listesini değiştirir. Bu algoritmalar artık bu sorunun çevrimiçi değişkeni için geçerli değildir. Sezgisel yaklaşımlarla ilgili literatür, KPP'de kesin yaklaşımlar kullananlardan çok daha fazladır. Ayrıca çoğunlukla meta-sezgisel yaklaşımlarla birlikte kullanılırlar. Sezgisel algoritmalar, örneğin En Derin Alt-Sol Doldur Algoritma, Sol Alt sezgisel taramasının üç boyutlu olana göre bir uzantısıdır. Bu konteynerdeki en derin konumları dikkate alarak kutuyu alta ve sola mümkün olan en derin seviyede yerleştirir [7]. Bu tür yöntemlerin, elde edilen hacim yoğunluğu açısından performansı aşağıdakileri içeren bir dizi faktöre bağlıdır; Kutu sayısı, işlendikleri dizi ve tüm kutu yönelimleri [8]. En Derin Sol- Alt Algoritma'nın karmaşıklığı yüksek olduğundan, bu algoritmanın bilgisayar tarafından verimli bir şekilde uygulanması gerekmektedir.

```
Tekrar et.
    Sıradaki kutuyu al.
    Tekrar et.
        Sıradaki konteynerde yeterince boş hacım olup olmadığını kontrol et.
        Yeteri kadar hacım var ise.
            Sıradaki kutuyu en derin en aşağıda en sol müsait yere yerleştir.
    Sıradaki kutu yerleştirilene kadar yada konteynerler bitene kadar devam et.
    Sıradaki kutu yerleşmedi ise.
        Yeni boş bir konteyner ekle.
        Sıradaki kuyuyu yeni eklenen konteynerin ilk koordinatörüne yerleştir.
Kutular bitene kadar devam et.
```

Şekil 1. En Derin Alt-Sol Doldur Algoritma Sözde Kodu

2.3 Meta-Sezgisel algoritmalar

Farklı sezgisel çözümlerin herhangi biri tatmin edici sonuçlar üretmediği için Meta-Sezgisel algoritmalar incelenmiştir. Meta-sezgisel, sezgisel optimizasyon algoritmalarını geliştirmek için bir dizi yönerge veya strateji sağlayan, üst düzey problemden bağımsız algoritmik bir çerçevedir [9]. Bu algoritmalar gelişmiş ülkelerde sanayi devrimine neden olmuştur. Meta-Sezgisel Algoritma'ların dikkate değer örnekleri arasında Genetik/Evrimsel Algoritmalar, Tabu Arama, Benzetilmiş Tavlama, Değişken Komşuluk Arama, Karga Arama-Algoritması, Büyük Komşuluk Arama ve Karınca Kolonisi Optimizasyonu bulunur. Meta-Sezgisel bir çerçevede ifade edilen yönergelere göre, sezgisel bir optimizasyon algoritmasının probleme özgü bir uygulaması da Meta-Sezgisel olarak adlandırılır. Terim Glover [10] tarafından icat edilmiş ve Yunanca ön ek meta- (metá, ötesinde üst düzey anlamında) sezgisel (Yunan heuriskein veya euriskein'den aramak için) ile birleştirilmiştir. Meta-Sezgiler, bilim camiası tarafından, çoğu zaman dal, sınır ve dinamik programlama gibi daha geleneksel tam karma-tamsayılı optimizasyon yöntemlerinin yerine uygulanabilir üstün bir alternatif olarak gösterilmiştir. Özellikle karmaşık problemler veya büyük problem vakaları için, Meta-Sezgisel çözüm genellikle çözüm kalitesi ve hesaplama süresi arasında daha iyi bir denge sunar. Üstelik Meta-Sezgisel yöntemler kesin yöntemlerden iki önemli yönüyle daha esnektir. Birincisi, Meta-Sezgisel çerçeveler genel terimlerle

tanımlandığından Meta-Sezgisel Algoritmalar, farklı problemler ve farklı durumlar arasında büyük ölçüde değişebilen, beklenen çözüm kalitesi ve izin verilen hesaplama süresi açısından çoğu gerçek yaşam optimizasyon probleminin ihtiyaçlarına uyacak şekilde uyarlanabilir. İkincisi, Meta-Sezgiselleştirme optimizasyon probleminin formülasyonuna ilişkin herhangi bir talepte bulunmaz. Bununla birlikte, bu esneklik iyi bir performans elde etmek için probleme özel adaptasyon gerektirir ve bu da ciddi maliyete sebep olur.

2.3.1 Genetik Algoritma(GA)

Genetik Algoritmalar, evrimsel algoritmaların büyük kısmına ait olan, uyarlanabilir sezgisel arama algoritmalarıdır. Genetik Algoritmalar doğal seleksiyon ve genetik fikirlerine dayanmaktadır. Bunlar, aramayı çözüm alanında daha iyi performans bölgesine yönlendirmek için geçmiş verilerle sağlanan rastgele aramanın akıllıca kullanılmasıdır. Genellikle optimizasyon sorunları ve arama sorunları için yüksek kaliteli çözümler üretmek için kullanılırlar. Genetik Algoritmalar ilk kez 1975'te Holland tarafından tanıtılmış [11] ve birçok araştırmacı tarafından incelenmiştir. Genetik algoritmalar doğal seleksiyon sürecini simüle eder. Bu da çevrelerindeki değişikliklere uyum sağlayabilen türlerin hayatta kalabilmesini, üremesini ve gelecek nesillere geçmesini sağlar. Basit bir deyişle, bir problemi çözmek için ardışık nesil bireyler arasında "en uygun olanın hayatta kalmasını" simüle eder. Her nesil bir birey popülasyonundan oluşur ve her birey arama alanında bir noktayı ve olası çözümü temsil eder. Her birey bir karakter yada bit dizisi olarak temsil edilir. Bu dizi kromozoma benzer. Her birey, belirli bir sorun için arama alanında bir çözümü temsil eder. Her birey, bileşenlerin sonlu bir uzunluk vektörü olarak kodlanır. Bu değişken bileşenler genlere benzer. Şekil 2'de GA sözde kodu verilmştir.

```
İlk nesil çözümleri rastgele olarak yarat

Nesli fitness fonksiyonuna göre değerlendirir

Kromozomları fitness skoruna göre sırala

Nesil sayısı maksimum nesil sayısından küçüktür ise devam et(While)

    Elitizm kromozomlarını önceki nesillerden al ve Pi'ye koy

    Önceki nesil çapraz kromozomlar yapmak ve Pi'ye koymak

    Önceki nesil popülasyondan mutasyon kromozomları yapmak ve Pi'ye koy

    (Pi)'yi değerlendir

    (Pi)'yi sırala

    Durma koşulu doğruysa

        Çık

While döngüsü bitişi
```

Şekil 2. Genetik Algoritma Sözde Kodu

2.3.2 Karga-Arama Algoritma

Karga Arama Algoritması, kargaların akıllı grup davranışlarına dayanan yeni bir meta-sezgisel yöntemdir. Karga Arama Algoritması 2016'da A. Askarzadeh tarafından önerilmiştir. Karga Arama Algoritması, kargaların ek beslenmesini özel olarak depolaması ve ona ihtiyacı olduğunda geri almasından esinlenmektedir [5].

Kargaların diğer kuşları izledikleri, diğer kuşların yiyeceklerini nerede sakladıklarını gözlemledikleri ve sahibi ayrıldıktan sonra çaldığı bilinmektedir.

Eğer bir karga hırsızlık yapmışsa, saklanan yerleri gelecekteki bir kurban olmaktan kaçınmak için taşımak gibi ekstra önlemler alacaktır. Aslında bir hırsızın davranışını tahmin etmek için bir hırsız olma deneyimlerini kullanırlar ve çalınmaya karşı korumak için önbelleklerini kullanarak en güvenli rotayı belirleyebilirler [12]. Özet olarak kargalar sürü şeklinde yaşar ve sakladıkları yemeğin yerlerini ezberlerler. Kargalar hırsızlık yapmak için birbirlerini takip ederler ve önbelleklerini olası bir hırsızlığa karşı korurlar. "d" boyutlu bir ortam olduğu varsayıldığında, karga sayısına N, arama alanındaki karga i'nin t tekrarlamasında konumu bir vektörle belirtilir. $k^{i,t}$ (i = 1, 2, ...,N; t = 1, 2, ... t_{maks}) $k^{i,t}$= [$k_1^{i,t}, k_2^{i,t}, \ldots, k_d^{i,t}$] ve t_{maks} maksimum tekrarlama sayısıdır. Her karga, bir hafızaya

sahiptir. Bu hafiza sakladığı yemeğin yerini içermektedir. Karga i'nin hafızası $h^{i,t}$ ile gösterilir. $h^{i,t}$, şimdiye kadar elde edilen en iyi saklanma yeridir. Her karganın hafızasına en iyi deneyiminin konumu ezberlenmiştir. Kargalar çevrede hareket eder ve daha iyi yiyecek kaynakları arar. j karganın yemek sakladığı yeri ($h^{j,t}$) ziyaret etmek istediği varsayılır ise bu tekrarlamada karga i, karga j'nin yemek sakladığı yere yaklaşmak için karga j'yi takip etmeye karar verir.

Bu halde, iki durum yaşanabilir [5, 13]:

➞ Durum 1:

Karga j, karga i'nin onu takip ettiğini bilmemektedir. Matematiksel olarak $r_j \geq AP^{j,t}$. Sonuç olarak, karga i karga j'nin yemek sakladığı yere yaklaşacak ve bu durumda, karga i'nin yeni pozisyonu aşağıdaki gibi elde edilir:

$$K^{i,t+1} = K^{i,t} + r_i * fl^{i,t} * \left(h^{j,t} - K^{i,t}\right) \quad (1)$$

Burada r_i, 0 ile 1 arasında eşit dağılımlı rasgele bir sayıdır; $fl^{i,t}$, t tekrarında karga i'nin uçuş uzunluğunu belirtir. Küçük fl değerleri yerel aramaya yol açar ($k^{i,t}$ çevresi) ve büyük değerler global arama ile sonuçlanır ($k^{i,t}$'den uzak). $AP^{j,t}$, karga j'nin tekrarlamadaki farkındalık olasılığını gösterir.

➞ Durum 2:

Karga j, karga i'nin takip ettiğini bilmektedir. Sonuç olarak karga j önbelleğini çalınmanın önlenmesi için arama alanından başka bir konuma giderek karga i'yi kandırır

$$K^{i,t+1} = Rastgele\ Konum \quad (2)$$

Karga Arama Algoritması'nda yoğunlaşma ve çeşitlendirme, esas olarak Farkındalık Olasılığı parametresi tarafından kontrol edilir. Sonuç olarak, küçük Farkındalık Olasılığı değerlerinin kullanılması yoğunlaşmayı artırır, büyük Farkındalık Olasılığı değerlerinin kullanılması çeşitliliği artırır. Şekil 3'de Karga Arama Algoritma sözde kodu verilmiştir.

Arama alanında N Karga sürüsünün konumunu rastgele yarat.

Kargaların konumunu değerlendir.

Her karganın hafızasını ilk konumu olarak ayarla.

t t_{maks} 'tan daha küçüktür ise tekrarla(While döngüsü):

(for döngüsü) i = 1: N (sürünün tüm N kargaları)

Rastgele takip edilecek kargalardan birini seç (örneğin j)

Bir farkındalık olasılığı(AP) tanımla

$r_j \geq AP^{j,t}$ ise

$k^{i,t+1} = k^{i,t} + r_i \times fl^{i,t} \times (h^{j,t} - k^{i,t})$

Yok ise

$k^{i,t+1}$ = arama alanında rastgele konum

(for sona eriyor)

Yeni pozisyonların fizibilitesini kontrol et

Kargaların yeni pozisyonunu değerlendir

Kargaların hafızasını güncelle

Tekrarlama (While döngüsü sona eriyor)

Şekil 3. Karga Arama Algoritma Sözde Kodu.

2.4 Hibrit Çözümler

Hibrit algoritma, aynı sorunu çözen iki veya daha fazla algoritmayı birleştiren bir algoritmadır. Ya birini seçer (verilere bağlı olarak) ya da algoritma boyunca bunlar arasında geçiş yapar. Bu genellikle her birinin istenen özelliklerini birleştirmek için yapılır. Böylece genel algoritma tekli bileşenlerden daha iyidir.

Hibrit algoritmalar, algoritmaların arama yeteneğinin geliştirilmesinde önemli bir rol oynar. Hibridizasyon, her algoritmanın avantajlarını birleştirerek hibrit bir algoritma oluşturmayı ve eşzamanlı olarak önemli dezavantajları en aza indirmeyi amaçlamaktadır. Genel olarak, hibridizasyonun sonucunda hesaplama hızı veya doğruluk açısından bazı iyileştirmeler yapabilir.

Hibrit, algoritmalar birleşimi olabilir veya yöntemler birleşimi de olabilir. Bu çalışmada iki tane Meta-Sezgisel algoritmanın karışımı ve Sezgisel Algoritma kullanılmıştır.

2.5 Literatör Taraması

Kutu Paketleme Problemi akademik camiada çok çalışılmıştır. Ancak karmaşıklığı, varyasyonları, kısıtlamaları ve endüstrinin birçok sektörünün odak noktası olduğu için akademisyenler tarafından yeni yöntemler ve algoritmalar kullanılarak sonuçları iyileştirilmeye devam edilmektedir. Meta-Sezgisel yöntemler, Kombinatoryal Problemleri çözmekle ünlü olduğu gibi KPP [14, 15], GSP [16, 17] ve Araç Yönlendirme Problemi [18, 19] gibi diğer sorunları çözmek için de kullanılmaktadır. Gerçek olarak Meta-Sezgisel yöntemler Kombinatoryal Problemlerin çözümünde başarılı olmuştur. Abdel-Basset vd. 1 boyutlu KPP için doğadan ilham alan Meta-Sezgisel Algoritmayı geliştirdiler [20]. Lodia vd. Kısmi Numaralandırma Algoritmaları kullanarak giyotin kısıtlamaları ile 2 boyutlu KPP'yi çözdüler [21]. En zorlu 3 boyutlu KPP tipi için son zamanlarda birçok çözüm önerilmiştir. Bunlardan biri olan Çok Seviyeli Yerel Arama Sezgisel yöntemi Alessio Trivella ve David Pisinger tarafından sunulmuştur [22]. Başka bir çözüm de, Kang vd. tarafından yeni bir paketleme stratejisine [2] sahip Hibrit, bir Genetik Algoritmadır. Sanal Makineler yönetiminde örtüşen öğeler, önemli bir KPP sayılmaktadır. Sanal Makineler Kutu Paketleme Problemi, Sezgisel, Açgözlü(Greedy) , Açgözlü olmayan ve Genetik Algoritmaları kullanarak zaman kısıtlamaları ve kalite gereksinimlerine göre Grange vd. tarafından bir çözüm sunulmuştur [23]. Takviye Öğrenme Yöntemleri kullanılarak yakın zamanda umut verici sonuçlar ortaya çıkmıştır. Hu vd. Derin Takviye Öğrenme tabanlı yöntemi kullanarak 3 boyutlu KPP için bir başarı elde etmeyi başarmışlardır [4]. Laterre vd. Sıralı Ödül'ü kullanarak genel Monte Carlo ağacı arama, sezgisel algoritmalar ve tamsayı programlama çözücülerinden daha iyi performans göstermiştir [24]. Duan vd., Seçilmiş Öğrenme Yaklaşımını kullanarak iyi tasarlanmış açgözlü algoritmadan ciddi oranda maliyet azalması elde edebilmişlerdir [25].

3 Araştırma Bulguları ve Tartışma

Bu bölümde KPP'nin kullanılan türü ve kriterlerinden bahsedilecektir. Ayrıca bahsedilen algoritmaların KPP üzerine pratik olarak nasıl uygulanabileceği gösterilecek ve çeşitli denemelerden bahsedilecektir.

3.1 Problemin kullanılan kriterleri

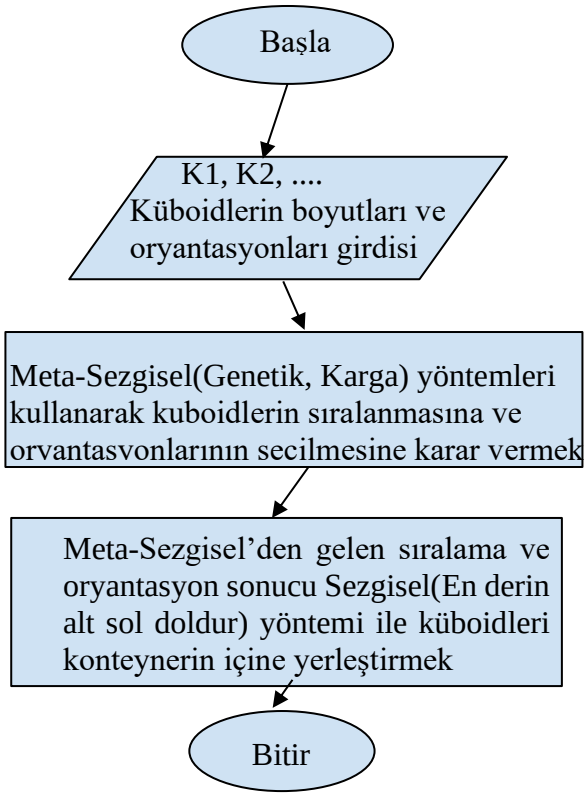
KPP'nin bir çok çeşidi bulunmaktadır. Bu yüzden problemin tipini ve kriterlerini belirtmek büyük önem taşımaktadır. Bu çalışmanın konusu detaylı olarak şu şekilde tanımlanabilir; küboid şeklinde ve 90 derecede döndürülebilen(altı farklı yön) N adet kutuyu, sabit uzunluk, sabit genişlik ve serbest (uzatılabilir) yükseklik ile üç boyutlu tek bir konteynere yerleştirmek istenmektedir. Ancak konteynerin yüksekliği en aza indirilerek yerleştirme yapılmalıdır. Bu şekilde bütün kutuları sığabilecek en minimum hacimde bir konteyner bulunmaya çalışılır.

3.2 Uygulama

Meta-Sezgisel algoritmalar güçlüdür ve diğer yöntemlerle çözülmesi zor olan çok çeşitli sorun alanlarıyla başa çıkabilir. Ancak bu algoritmalar genel yöntemler olduğundan dolayı probleme yönelik spesifik bilgi içermez. Bu şekilde problem bilgileriyle hibridizasyon, arama alanını daha akıllıca keşfederek çözümün kalitesini ve Meta-Sezgisel algoritmanın performansını önemli ölçüde arttırabilir. Bu çalışmada kullanılan yöntem bir konteynera kutular paketlenerek tek tek yerleştirilir. Bu yöntemin dezavantajı, bir kutunun nereye yerleştirileceğine ilişkin kararların yalnızca yerel ölçütlerle verilebilmesidir. Böylece algoritma zayıf bir paketleme ile sonuçlanabilir. Bu sorunun üstesinden gelmek ve farklı paketleme popülasyonunu yaratmak için hem Meta-Sezgisel olarak Genetik ve Karga Algoritmaları, hem Sezgisel olarak En Derin Sol- Alt Algoritma kullanılmıştır.

3.2.1 Metod Diyagramı

Bu çalışmada KPP çözümü 3 aşamaya bölündü. Birincisi küboidlerin en iyi şekilde sıralanması. İkincisi ise küboidlerin en iyi oryantasyonda koyulması. Son aşaması küboidlerin en iyi yerleştirme şekli uygulanması. Şekil 4 diyagram metodun akışını göstermektedir.



Şekil 4. Metod Diyagramı

3.2.2 Genetik algoritma uygulanması

Sıradan Genetik Algoritma'nın bir probleme uygulanması zor değildir ama iyice tasarlanmış bir Genetik Algoritmaya ulaşmak ciddi zorluktur. Genetik algoritması ne kadar iyi tasarlanırsa o kadar optimal çözümlere ulaşılabilir.

● Kromozom Temsili

Kromozom yapısının temsil edilmesi Genetik Algoritma'da büyük bir rol oynamaktadır. Bu Algoritma uygulanırken kromozom temsili ne kadar basit ve minimize olursa o kadar iyi olur.
KPP'de kromozom, bir küboidlerin sırası ve oryantasyonlarından oluşmaktadır. Problemin girdisi küboidlerin uzunluk, genişlik ve yüksekliğinden oluşmaktadır. Bu şekilde bu problemin girdisi tek bir dizi olup, bütün kromozomlar aynı girdiyi paylaşmaktadır. Başka bir deyişle bu kromozom iki pozitif tamsayı dizisinden oluşmaktadır. Birisi küboidlerin sırası, öbürü küboidlerin oryantasyonlarıdır. Altı küboid yönü aşağıdaki gibi sembolize edilmiştir:

1- (uzunluk, genişlik, yükseklik)
2- (uzunluk, yükseklik, genişlik)
3- (genişlik, uzunluk, yükseklik)
4- (genişlik, yükseklik, uzunluk)
5- (yükseklik, genişlik, uzunluk)
6- (yükseklik, uzunluk, genişlik)

Bu şekilde boyutları işlerken oryantasyon koduna göre hangisi uzunluk, hangisi genişlik, hangisi yükseklik bilinmektedir örneğin verilen 5 kutunun boyutları aşağıdaki gibidir:
1- (7, 6, 2) 2- (1, 2, 3) 3- (11, 10, 12)
4- (5, 12, 11) 5- (7, 7, 7)
İlk varsayılan oryantasyon 1 kodu olan(uzunluk, genişlik, yükseklik) olandır. Demek ki bu örnek kromozomu temsil edilmesi aşağıda gibidir:

Çizelge 1

Küboidlerin Sıralanması	1	2	3	4	5
Küboidlerin oryantasyonları	1	1	1	1	1

Bu kromozom üzerinde birkaç değişiklik aşağıdaki çizelge gibi yapıldığında

Çizelge 2

Küboidlerin Sıralanması	5	1	2	4	3
Küboidlerin oryantasyonları	2	1	3	5	6

Boyutları aşağıda gibi olur:

1- (7, 7, 7) 2- (7, 6, 2) 3- (2, 1, 3)
4- (11, 12, 5) 5- (12, 11, 10)
Yerleştirme, algoritmaya girdiyi verirken boyutların (uzunluk, genişlik, yükseklik) olarak geldiğini varsayılıyor.

● İlk Nesil

İlk nesil rastgele olarak yaratılmaktadır. Girdilerin oryantasyon kodu aksi belirtilmemiş ise varsayılan olarak 1. oryantasyon olarak kabul edilir. Girdilerin sıralamasıve oryantasyonları rastgele bir işleme sokularak yeni bir sıralama ve oryantasyon oluşturulur. Bu işlem bir neslin kromozom sayısı kadar tekrarlandığında ilk nesil oluşturulur.

● Uygunluk Hesaplama

Bu hesaplama çözümlerin değerlendirilmesi anlamına gelir. Çözümün değerlendirmesi, kutuları konteynerin içine En Derin Alt-Sol Doldur algoritmasını kullanarak yerleştirdikten sonra

hesaplanır. Konteyner küboid şeklinde olduğu için aşağıdaki gibi hacmi hesaplanabilir.

Konteyner Dolan Hacmi =
(Uzunluk × Genişlik × Ulaşılan Yükseklik) (3)

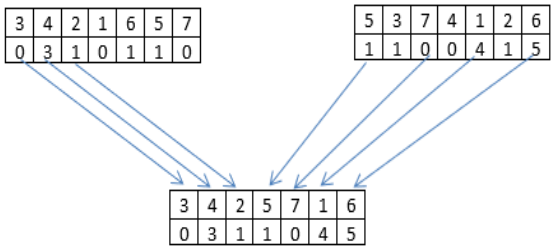
Konteynerin uzunluğu ve genişliği sabit olduğu için sadece yükseklik değerinin ne kadar değiştiği hesaplanır. Buradaki amacın, yüksekliği en aza indirmek olduğu unutulmamalıdır.

● Seçim

Bir nesilden başka nesile geçerken oransal olarak en iyi kromozomlar taşınır ve bu seçilen kromozomlar üzerine çaprazlama ve mutasyon uygulanıp yeni nesil bireyler oluşturulmaktadır. Seçim oranı bu çalışmada %50 olarak belirlenmiştir.

● Çaprazlama

Çaprazlama, farklı kromozomların çiftleşerek yeni bir kromozom ortaya çıkartmasına denir. Bu iki kromozom en iyi seçilen kromozomlardan rastgele olarak seçilir. Bu işlem çaprazlama oranına göre tekrarlanır. Bu çalışmada çaprazlama oranı %25 olarak belirlenmiştir. Örneğin Şekil 5'de iki kromozom arasında çaprazlama işlemi uygulanmıştır.



Şekil 5. Çaprazlama işlemi

● Mutasyon

Mutasyon işlemi bir kromozom üzerine uygulanır. Seçilen en iyi kromozomlardan rastgele olarak bir kromozom seçilir ve onun üzerine mutasyon uygulanır. Kromozomdan herhangi bir kutu seçilir ve bu kutunun oryantasyonu değiştirilerek mutasyon uygulanır. Mutasyon seçim ve çaprazlama gibi belli oranda uygulanır. Bu çalışmada mutasyon oranı %25 belirlenmiştir.

Örnek:

Şekil 6. Çaprazlama işlemi

● Durdurma Koşulu

Bu algoritma tekrarlanması, nesil maksimum sayısına gelene kadar devam eder ve ulaşılan en iyi çözüm verilir. Bu çalışmada nesil maksimum sayısı değişken olarak belirlenir.

3.2.3 Karga-Arama Algoritma

● Algoritma Değişkenlerin Ayarlanması

Bu aşamada sürü boyutu, maksimum yineleme (yemek saklandığı yeri değiştirme) sayısı, uçuş uzunluğu ve farkındalık olasılığı gibi değişkenlerin değeri verilir. Bu çalışmada verilen değerler aşağıdaki gibidir:
1- Sürü boyutu: 50 Karga,
2- Maksimum yineleme sayısı: 50 kere,
3- Uçuş uzunluğu: Bu parametre, karga algoritması gerçek hayatta uygulanırken bir şeyi karşılamadığı için ihmal edilmiştir,
4- Farkındalık olasılığı: Bütün kargalara 0 ve 1 arasında bir değer rastgele olarak verilmiştir.

● İlk Konum ve Hafıza Ayarlanması:

Kargaların yemeğinin konumları rastgele olarak verilmiştir. Hafızalarında da ilk başta karşılaştırılacak pozisyon olmadığı için aynı pozisyon değeri verilmiştir.

● Uygunluk Değerlendirme

Genetik algoritması gibi kutuları konteynerin içine En Derin Alt-Sol Doldur algoritmasını kullanarak yerleştirdikten sonra hesaplanabilir. Değerlendirme formülü olarak eşitlik 3 kullanılmıştır.

● Kargaların Takibi Başlatılması ve Yeni Pozisyona Geçirmesi

Sırayla her karga rastgele olarak başka bir karga seçip, onu takip edip yiyeceğini nerede sakladığını öğrenmeye çalışır. Örneğin karga i karga j'yi saklanan yiyeceklerinin konumunu keşfetmek için takip eder. Karga j'nin Farkındalık olasılığı $AP^{j,t}$ r_j'dan daha büyük ise (karga j'nin dikkatli olup takip

edildiğini fark edip bilerek uzak bir yere gitmesi anlamına gelir) rastgele yeni bir pozisyon yaratılır ve karga i yemeğini oraya taşır. Bu durumda karga algoritması keşif yapılması anlamındadır. Karga j'nin Farkındalık olasılığı $AP^{j,t}$ r_j'dan daha küçük yada eşit ise (karga j'nin yeteri kadar dikkatli olmadığı anlamına gelir) aşağıdaki formülü kullanarak yeni pozisyon/konum hesaplayıp ona geçer. Bu durumda ise karga algoritması sömürü(exploitation) yapması anlamındadır. Burada 1 ve 2 eşitlikleri pratik olarak nasıl uygulanacağı anlatılacaktır**:**

($h^{j,t}$ - $k^{i,t}$): karga j'nin hafızası ve karga i'nin yiyeceğinin eski pozisyonu arasındaki fark/mesafe anlamına gelir. Bunu hesaplamak için pratik olarak bir sürü yöntem vardır. Bu çalışmada Hamming Mesafesi yöntemi kullanılmıştır. Karga (çözüm) temsilinde iki değişken olduğu için iki Hamming Mesafesi oluşmaktadır. Birisi küboidlerin sıralaması Hamming Mesafesi, diğeri oryantasyon Hamming Mesafesi. Bu problemde Hamming mesafesini hesaplamak için iki çözüm arasındaki farkın çıkartılması lazım. İki çözüm arasındaki farkın çıkartılması için sırayla kutulara bakılır. İki çözümde de birinci sırada aynı kutunun olup olmadığına bakılır. Aynı ise fark dizisinin birinci öğesi 1 değilse 0 olur. Oryantasyon için de aynı şey uygulanır. Daha sonra ortaya çıkan dizi sütunlarının toplamı alınıp, iki öğeli bir dizi ortaya çıkarılır. Aşağıdaki örnek Hamming Mesafesi'nin nasıl hesaplanacağını göstermektedir.

($h^{j,t}$ -$k^{i,t}$) = Hamming Mesafesi

Örnek: İki tamsayı dizisi arasındaki Hamming Mesafesi hesaplanmaktadır.

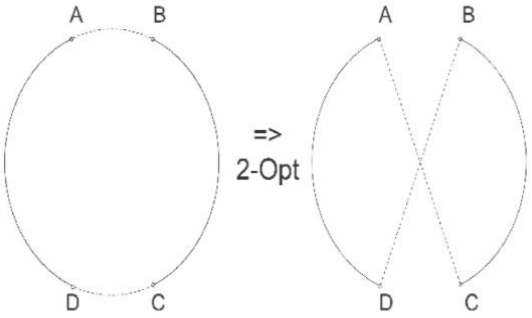
$$\text{Hamming Mesafesi} = \begin{bmatrix}5&1\\4&2\\3&3\\1&3\\2&6\end{bmatrix} - \begin{bmatrix}4&1\\5&1\\2&5\\1&4\\3&6\end{bmatrix} = \begin{bmatrix}1&0\\1&1\\1&1\\0&1\\1&0\end{bmatrix} = [4\ \ 3]$$

Şekil 6. Hamming Mesafesi Hesaplanması.

$k^{i,t+1}$Hamming Mesafesi: Bu ifade Karga Algoritma'sında karganın hareketi anlamındadır. Bu hareketi pratik olarak ifade edebilmek için literatürde iki çeşit vardır:

➜ 2-opt

Bu işlev 1965 yılında Lin tarafından tanımlanmıştır ve o zamandan beri yönlendirme sorununu çözmek için yaygın olarak kullanılmaktadır [26, 27]. KPP yönlendirme sorununa benzemektedir. İkisinin de çözümü bir sıralamadan oluşmaktadır. 2-opt, çözüm dizilerini daire gibi yaptıktan sonra, rastgele iki küboidi ortadan seçip arasındaki diziyi alıp terslemektedir. Aşağıdaki örnek 2-opt nasıl yapıldığını göstermektedir, örneğin dört küboidden oluşan bir dizi verilsin. Dizi sıra ile A, B, C, D dir. 2-opt uyguladıktan sonra A, C, B, D olarak çevrilmiştir.



Şekil 7. 2 OPT Temsili.

➜ 3-opt

Lin tarafından da önerilen 3-opt işlemi 2-opt'e benzer. Bu durumda küboidler 3'e çıkarılmıştır. Bu yöntemi kullanmanın karmaşıklığı 2-opt'den daha fazladır.

Özet olarak bu çalışmada 2-opt yöntemi kullanılmıştır. Bu 2-opt yöntemi Hamming mesafesi kadar uygulanır. Bu şekilde karga hareketini gerçekleştirir. Algoritmanın karmaşıklığını basitleştirmek için r_i $vefl^{i,t}$ parametreleri dikkate alınmamıştır.

3.2.4 Hibrit Karga-Genetik Algoritma

Genetik Algoritma'da kullandığımız ilk nesli rastgele yaratmaktansa Karga Algoritma'sı kullanarak oluşturduğumuz sonuçları Genetik Algoritma ya ilk nesil olarak verilirse bulunan çözüm optimum çözüme daha yakın bir çözüm olur. Yani yeni bir problem çözülürken daha önce bu problemle alakalı Karga Algoritma'sı tarafından bulunan çözümlerden faydalanılarak çözüm uzayı daraltılmaya çalışılır. Bu şekilde probleme Hibrit Karga-Genetik Algoritma uygulanmıştır.

3.2.5 Hibrit Genetik-Karga Algoritma

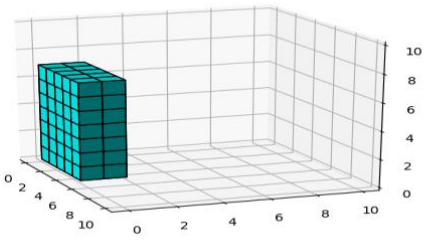
Karga Algoritma'sı kullanılırken yiyeceklerin ilk konumlarını rastgele yaratmaktansa Genetik Algoritmadan çıkan çözümler Karga Algoritma'sına yiyeceklerinin ilk konumları olarak verilir. Bu şekilde probleme Hibrit Genetik-Karga Algoritma uygulanmıştır.

3.2.6 En Derin Alt-Sol Doldur Algoritma

Bu algoritma, metodun sezgisel kısmıdır. En Derin Alt-Sol Doldur, en iyi yerleştirme algoritmalarından biridir. Önceki oluşan boşlukları doldurma özelliğinden dolayı konteynerin hacminin azaltılmasına yardımcı olur. Pratikte nasıl uygulanacağı anlatmak için bir örnek üzerinden gidilmelidir.

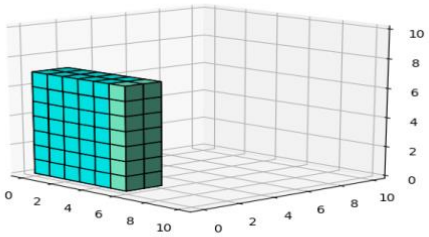
Örnek: [(5, 2, 7), (1,2,7), (4, 2, 3), (4, 2, 5), (4, 2, 1)] En Derin Alt-Sol Doldur Algoritma ile yukarıdaki örneğin kutularını(küboidlerini) yerleştirmek üzere aşağıdaki şekillerde (8, 9, 10, 11, 12) yerleştirme işlemi aşama aşama gösterilmektedir.

1. Aşama: (5, 2, 7) Küboidi konteynere yerleştirildi.



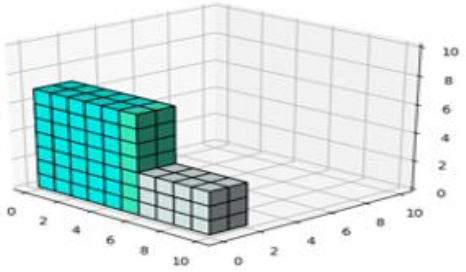
Şekil 8. En Derin Alt-Sol Doldur Algoritma Uygulanması 1. Aşama

2. Aşama: (5,2,7) Küboidi konteynere yerleştirildikten sonra, (1,2,7) Küboidi konteynere yerleştirildi.



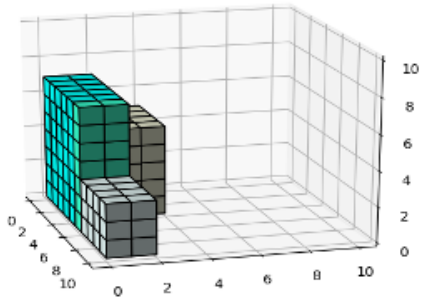
Şekil 9. En Derin Alt-Sol Doldur Algoritma Uygulanması 2. Aşama

3. Aşama: (1,2,7) Küboidi konteynere yerleştirildikten sonra, (4, 2, 3) Küboidi konteynere yerleştirildi.



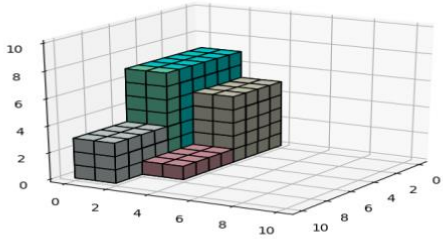
Şekil 10. En Derin Alt-Sol Doldur Algoritma Uygulanması 3. Aşama

4. Aşama: (4, 2, 3) Küboidi konteynere yerleştirildikten sonra konteynerin en derin alt sol tarafı dolduğu için (4, 2, 5) Küboidi konteynerin müsait olan en derin alt sol tarafına yerleştirildi.



Şekil 11. En Derin Alt-Sol Doldur Algoritma Uygulanması 4. Aşama

5. Aşama: (4, 2, 5) Küboidi konteynere yerleştikten sonra, (4, 2, 1) Küboidi de konteynere yerleştirildi.



Şekil 12. En Derin Alt-Sol Doldur Algoritma Uygulanması 5. Aşama

3.2.7 Test Verilerinin Üretimi

Bu çalışmada Kutu Paketleme Problemi varsayılan kriterlerle birlikte Benchmark veri seti olmadığı için rastgele veri seti yaratılmıştır. Yaratılan küboid veri seti boyutları 1 ve 12 arasındadır. Konteyner ise 14 genişlikli, 14 uzunluklu ve serbest yüksekliklidir. Bu rakamlar da ölçü birimi bulunmamakta ve değiştirilebilirdir. Gerçek hayatta bu rakamın ölçü birimi santimetre, milimetre yada herhangi bir metreye ait bir ölçü birimi olabilir. Bu ölçü birimi problemin gerçek hayatta uygulamasına göre belirlenebilir.

3.3. Test Verilerinin Üretimi

Bu çalışma, piton programlama dili kullanılarak geliştirilmiştir. Çalışmada numpy, matplotlib vb. kütüphaneleri kullanılmıştır. Bütün kullanılan algoritmalar da kod olarak hazır bir kütüphane kullanılmayıp, yazar tarafından geliştirilmiştir. Geliştirilen yöntemin uygulandığı bilgisayar özellikleri aşağıda verilmiştir.

➜ İşletim Sistemi: Windows 10 (64 bit)

➜ İşlemci: Intel Core i7

➜ 2.6 GHz. RAM: 16GB

➜ HDD: 500GB

Deneme olarak 4 farklı test yapılmıştır. Metodları değerlendirmek için farklı küboid sayısında örnekler yaratılmıştır. Sonuçları ayrı ayrı analiz edilmiştir. Denemeler Karga Sürü boyutu 50 Karga, Karga maksimum yineleme sayısı 50, Kromozom sayısı 50, Genetik nesil sayısı 50, Seçim oranı %50, Çaprazlama oranı %25 ve Mutasyon oranı %25 parmetreleri ile gerçekleştirilmiştir. Algoritmalarda ilk aşamada rastgele çözümler yaratıldığı için bir kere denemenin adil olmadığı düşünülmüştür. Her algoritma, aynı örnek üzerinde 10 kere çalıştırılmıştır ve aşağıdaki her test sonucu bu 10 testin ortalamasıdır. Sonuç tablolarında Genetik, Karga-Arama, Hibrit Karga-Genetik ve Hibrit Genetik-Karga Algoritmalarına sıra ile GA, KAA, HKG ve HGK kısaltmaları olarak kullanılmıştır.

● ***15 Küboidli Test***

Tablo 1'de 20 tane 15'li küboid örneği testinin metodlarının karşılaştırılmasını göstermektedir:

Tablo 1. Metodların Performansı

Metod 1	Metod 2	Daha iyi Performans gösteren	Oransal Başarısı
GA	KAA	GA	1.44
KAA	HGK	HGK	2.7
KAA	HKG	HKG	3.7
GA	HGK	HGK	1.2
GA	HKG	HKG	2.2
HGK	HKG	HKG	1

Tablo 1'de metodların karşılaştırılmasına bakıldığında metodlar arasında en iyi performans gösteren Hibrit Karga-Genetik'dir.

● ***20 Küboidli Test***

Tablo 2'de 20 tane 20'li küboid örneği testinin metodlarının karşılaştırılmasını göstermektedir:

Tablo 2. 20'li küboid örneği testinin metodlarının karşılaştırılmasını

Metod 1	Metod 2	Daha iyi Performans gösteren	Oransal Başarısı
GA	KAA	GA	2

KAA	HGK	HGK	2.8
KAA	HKG	HKG	4.1
GA	HGK	HGK	0.79
GA	HKG	HKG	2.13
HGK	HKG	HKG	1.35

Metodların karşılaştırılması tablosunda bakıldığında metodlar arasında en iyi performans gösteren Hibrit Karga-Genetik'dir.

● ***25 Küboidli Test***

Tablo 3'de 20 tane 25'li küboid örneği testinin metodlarının karşılaştırılmasını göstermektedir:

Tablo 3. 25'li küboid örneği testinin metodlarının karşılaştırılması

Metod 1	Metod 2	Daha iyi Performans gösteren	Oransal Başarısı
GA	KAA	GA	2.7
KAA	HGK	HGK	3.2
KAA	HKG	HKG	4.5
GA	HGK	HGK	0.59
GA	HKG	HKG	1.88
HGK	HKG	HKG	1.29

Metodların karşılaştırılması tablosunda bakıldığında metodlar arasında en iyi performans gösteren Hibrit Karga-Genetik'dir.

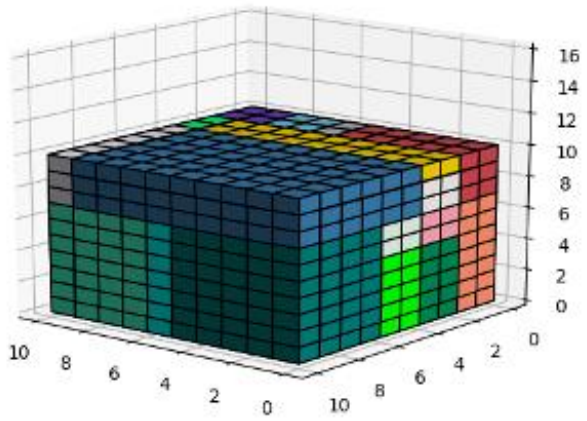
● ***Optimum Çözümü Bilinen Bir Örnek***

Genelde KPP'lerde ulaşılan bir çözüm küboidlerin toplam hacmine bakarak değerlendirilebilir. Başka bir deyişle ulaşılan çözüm de, konteynerin hacmi küboidlerin toplam hacmine ne kadar yakın duruyor ise bu çözümün ne kadar iyi olduğuna karar verilebilir. Ancak her örnek bu şekilde değildir. Bazı örneklerde optimum çözüm, küboidlerin toplam hacmi ya da ona yakın bir rakam olmayabilir. Bu yüzden optimum çözümü bilinen bir örnek hazırlamak gerektirmiştir. Bu örnek üzerinden değerlendirmek daha doğrudur. Örnek aşağıdaki özelliklerine sahiptir.

➜ Küboidlerin boyutları şu şekilde:(uzunluk, genişlik, yükseklik) verilmiştir.

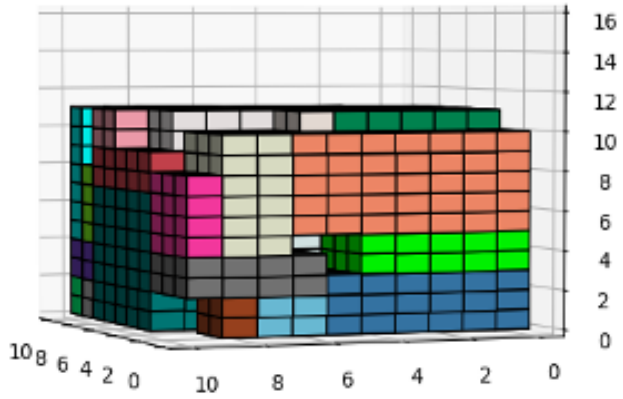
[(5, 2, 7),(1,2,7), (4, 2, 3), (4, 2, 5), (4, 2, 1), (2,2,1) ,(3,2,5), (3, 2, 1), (4,2,3) ,(5,4,7),(1,4,7),(4,4,7) ,(6,2,4) ,(3,2,6),(10,2,2),(1,2,4) ,(3,2,4),(1,1,4) ,(4,1,4),(3,2,2),(3,1,4),(5,2,3),(1,2,3),(2,2,3), (2,2,3), (9,2,2), (1,2,3),(9,6,3),(1,6,3),(9,2,1)]

➜ Küboidler yukarıdaki sırada (uzunluk, genişlik, yükseklik) olarak (10, 10, 10) boyutlu bir konteynerin içine boşluksuz olarak yerleşebilir. Bu yüzden yukarıdaki verilen sıralama ve oryantasyon bu örneğin en optimum çözümdür. Aşağıdaki resimlerde yerleşmesi iki yönden görülebilir.



Şekil 313. Verilen örneğin optimum çözümü

Dolayısıla Optimum ulaşılabilecek hacım küboidlerin toplam hacmıdır=1000 br^3.
Bu örnek çözülürken farklı hiperparametre ayarları denenmiştir ve en iyi ulaşılan çözüm 1100 br^3 hacimli bir konteynerdir. En iyi hiperparametre ayarları Karga Sürü boyutu 200 Karga, Karga maksimum yineleme sayısı 50, Kromozom sayısı 200, Genetik nesil sayısı 50, Seçim oranı %50, Çaprazlama oranı %25 ve Mutasyon oranı %25 olarak belirlenmiştir. Şekil 314 ulaşılan en iyi çözüm temsilini göstermektedir. Tablo 4 ise en iyi ulaşılan hiperparametre ayarları sonuçlarını göstermektedir.



Şekil 314. Verilen örnek için en iyi ulaşılan çözüm

Tablo 4. Hiper parameter sonuçları

Metod	Ort.	Maks.	Min.	Optimum Hacim	Süre/sa
KAA	1200	1200	1200	1000	181.43
HKG	1160	1200	1100	1000	374.50
GA	1200	1200	1200	1000	186.96
HGK	1200	1200	1200	1000	365.21

4 Sonuçlar

Bu çalışmada KPP'nin bir çeşidinin üzerinde çalışılmıştır. Bu çalışmadaki temel amaç, 3 boyutlu kutuları bir konteynerin içine optimum bir şekilde yerleştirilmektir. Yöntem olarak Sezgisel ve Meta-Sezgisel yöntemler birlikte kullanılmıştır. Sezgisel olarak En Derin Alt-Sol Doldur Algoritma'sı, Meta-Sezgisel olarak dört farklı algoritma kullanılmıştır. Bunlar, Karga Arama Algoritma, Genetik Algoritma, Hibrit Genetik-Karga ve Hibrit Karga-Genetik Algoritma'dır. Sayısal sonuçlar Hibrit Karga-Genetik'in en iyi yöntem olduğunu gösteriyor. Bu çalışmanın katkısı, 3 boyutlu KPP'de Karga Arama, Hibrit Genetik-Karga ve Hibrit Karga-Genetik Algoritma'larının ilk defa uygulanmasıdır. Gelecek çalışmada kademeli çözümler ve takviyeli derin öğrenmeyi dahil ederek çözümler üretmektir.

Kaynaklar

[1] Osaba, E., Yang, X.-S., Diaz, F., Lopez-Garcia, P. and Carballedo, R. (2016). An improved discrete bat algorithm for symmetric and asymmetric Traveling Salesman Problems. Engineering Applications of Artificial Intelligence, 48, pp.59–71.

[2] Kang, K., Moon, I. and Wang, H. (2012). A hybrid genetic algorithm with a new packing strategy for the three-dimensional bin packing problem. Applied

Mathematics and Computation, 219(3), pp.1287–1299.

[3] Lodi, A., Martello, S. and Vigo, D. (2002). Heuristic algorithms for the three-dimensional bin packing problem. European Journal of Operational Research, 141(2), pp.410–420.

[4] Hu, H., Zhang, X., Yan, X., Wang, L. and Xu, Y. (2017). Solving a New 3D Bin Packing Problem with Deep Reinforcement Learning Method. arXiv:1708.05930 [cs]. [online] Available at: https://arxiv.org/abs/1708.05930 [Accessed 4 Jun. 2020].

[5] Askarzadeh, A. (2016). A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm. Computers & Structures, 169, pp.1–12.

[6] J. Egeblad. Heuristics for Multidimensional Packing Problems.PhD thesis, University of Copenhagen, Department of Computer Science, 2008.

[7] Art, Richard Carl. 1966. "An Approach to the Two Dimensional Irregular Cutting Stock Problem." PhD diss., Massachusetts Institute of Technology.

[8] Bennell, J. A., and J. F. Oliveira. 2009. "A Tutorial in Irregular Shape Packing Problems." Journal of the Operational Research Society 60: S93–S105.

[9] K. Sörensen and F. Glover. Metaheuristics. In S.I. Gass and M. Fu, editors, Encyclopedia of Operations Research and Management Science, pages 960–970. Springer, New York, 2013.

[10] F. Glover. Future paths for integer programming and links to artificial intelligence. Computers and Operations Research, 13:533–549, 1986.

[11] Holland, J.H.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor (1975).

[12] Clayton N, Emery N. Corvide cognition. Curr Biol 2005;15:R80–1.

[13] Díaz, P., Pérez-Cisneros, M., Cuevas, E., Avalos, O., Gálvez, J., Hinojosa, S. and Zaldivar, D. (2018). An Improved Crow Search Algorithm Applied to Energy Problems. Energies, [online] 11(3), p.571. Available at: https://www.mdpi.com/1996-1073/11/3/571 [Accessed 4 Jun. 2020].

[14] Laabadi, S., Naimi, M., Amri, H.E. and Achchab, B. (2020). A Binary Crow Search Algorithm for Solving Two-dimensional Bin Packing Problem with Fixed Orientation. Procedia Computer Science, [online] 167, pp.809–818. Available at: https://www.sciencedirect.com/science/article/pii/S1877050920308863 [Accessed 11 Jun. 2020].

[15] Fu, Y. and Banerjee, A. (2020). Heuristic/meta-heuristic methods for restricted bin packing problem. Journal of Heuristics.

[16] Liu, Y. and Cao, B. (2020). A Novel Ant Colony Optimization Algorithm With Levy Flight. IEEE Access, [online] 8, pp.67205–67213. Available at: https://ieeexplore.ieee.org/abstract/document/9056538 [Accessed 11 Jun. 2020].

[17] Ashraf, U., Liang, J., Akhtar, A., Yu, K., Hu, Y., Yue, C., Masood, A.M. and Kashif, M. (2020). Meta-heuristic Hybrid Algorithmic Approach for Solving Combinatorial Optimization Problem (TSP). Communications in Computer and Information Science, pp.622–633.

[18] Pitakaso, R., Sethanan, K. and Jamrus, T. (2020). Hybrid PSO and ALNS algorithm for software and mobile application for transportation in ice manufacturing industry 3.5. Computers & Industrial Engineering, [online] 144, p.106461. Available at: https://www.sciencedirect.com/science/article/abs/pii/S0360835220301959 [Accessed 11 Jun. 2020].

[19] Kuhn, H., Schubert, D. and Holzapfel, A. (2020). Integrated order batching and vehicle routing operations in grocery retail – A General Adaptive Large Neighborhood Search algorithm. European Journal of Operational Research. [online] Available at: https://www.sciencedirect.com/science/article/abs/pii/S0377221720303088 [Accessed 11 Jun. 2020].

[20] Abdel-Basset, M., Manogaran, G., Abdel-Fatah, L. and Mirjalili, S. (2018). An improved nature inspired meta-heuristic algorithm for 1-D bin packing problems. Personal and Ubiquitous Computing, 22(5–6), pp.1117–1132.

[21] Lodi, A., Monaci, M. and Pietrobuoni, E. (2017). Partial enumeration algorithms for Two-Dimensional Bin Packing Problem with guillotine constraints. Discrete Applied Mathematics, 217, pp.40–47.

[22] Trivella, A. and Pisinger, D. (2016). The load-balanced multi-dimensional bin-packing problem. Computers & Operations Research, [online] 74, pp.152–164. Available at: https://www.sciencedirect.com/science/article/abs/pii/S0305054816300909 [Accessed 11 Jun. 2020].

[23] Grange, A., Kacem, I. and Martin, S. (2018). Algorithms for the bin packing problem with overlapping items. Computers & Industrial Engineering, [online] 115, pp.331–341. Available at: https://www.sciencedirect.com/science/article/abs/pii/S0360835217305004 [Accessed 11 Jun. 2020].

[24] Laterre, A., Fu, Y., Jabri, M.K., Cohen, A.-S., Kas, D., Hajjar, K., Dahl, T.S., Kerkeni, A. and Beguir, K. (2018). Ranked Reward: Enabling Self-Play Reinforcement Learning for Combinatorial Optimization. arXiv:1807.01672 [cs, stat]. [online] Available at: https://arxiv.org/abs/1807.01672 [Accessed 4 Jun. 2020].

[25] Duan, L., Hu, H., Qian, Y., Gong, Y., Zhang, X., Xu, Y. and Wei, J. (2019). A Multi-task Selected Learning Approach for Solving 3D Flexible Bin Packing Problem. arXiv:1804.06896 [cs, stat]. [online]

Available at: https://arxiv.org/abs/1804.06896 [Accessed 4 Jun. 2020].

[26] Bianchessi, N., & Righini, G. (2007). Heuristic algorithms for the vehicle routing problem with simultaneous pick-up and delivery. Computers & Operations Research, 34, 578–594.

[27] Tarantilis, C., & Kiranoudis, C. (2007). A flexible adaptive memory-based algorithm for real-life transportation operations: Two case studies from dairy and construction sector. European Journal of Operational Research, 179, 806– 822.