T.C.
GEBZE TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

KISIT PROGRAMLAMA YÖNTEMİYLE KUTU PAKETLEME PROBLEMLERİNİN ÇÖZÜMÜ

AHMET KARAKAŞ
YÜKSEK LİSANS TEZİ
ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI

GEBZE
2023

T.C.
GEBZE TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

KISIT PROGRAMLAMA YÖNTEMİYLE KUTU PAKETLEME PROBLEMLERİNİN ÇÖZÜMÜ

AHMET KARAKAŞ
YÜKSEK LİSANS TEZİ
ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI

DANIŞMANI
DR. ÖĞR. ÜYESİ ÖZGÜR ÜNSAL

GEBZE
2023

T.R.
GEBZE TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

SOLUTION OF BIN PACKING PROBLEMS WITH CONSTRAINT PROGRAMMING METHOD

AHMET KARAKAŞ

A THESIS SUBMITTED FOR THE DEGREE OF MASTER OF SCIENCE

DEPARTMENT OF INDUSTRIAL ENGINEERING

THESIS SUPERVISOR
ASST. PROF. DR. ÖZGÜR ÜNSAL

GEBZE
2023

GEBZE TEKNİK ÜNİVERSİTESİ	**YÜKSEK LİSANS JÜRİ ONAY FORMU**

GTÜ Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 09/02/2023 tarih ve 2023/13 sayılı kararıyla oluşturulan jüri tarafından 07/03/2023 tarihinde tez savunma sınavı yapılan Ahmet KARAKAŞ'ın tez çalışması Endüstri Mühendisliği Anabilim Dalında YÜKSEK LİSANS tezi olarak kabul edilmiştir.

JÜRİ

ÜYE
(TEZ DANIŞMANI) : Dr. Öğr. Üyesi Celal Özgür ÜNSAL

ÜYE : Doç. Dr. Tülay KORKUSUZ POLAT

ÜYE : Dr. Öğr. Üyesi Burak PAÇ

ONAY

Gebze Teknik Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun
…..…/…..…/..….… tarih ve ………/……... sayılı kararı.

İMZA/MÜHÜR

ÖZET

Bu tezde özellikle perakende sektöründeki depolama, sevkiyat ve mağaza yerleşim konularında karşımıza çıkan pek çok sorunun arka planında bulunan kutu paketleme problemi ele alınmaktadır.

Bu genel problemin tek boyutlu, iki boyutlu ve üç boyutlu versiyonları için geliştirilen mevcut çözüm yöntemlerinin ortaya konmasının ardından, genel problemin iki farklı türü ele alınarak tek boyutlu, iki boyutlu ve üç boyutlu versiyonlar için kısıt programlama (KP) ve karma tamsayılı programlama (KTP) modelleri geliştirilmiştir. Modeller literatürdeki veri setleri ile kıyaslanmış ve geliştirilen kısıt programlama modellerinin optimum veya optimuma yakın çözümleri karma tam sayılı programlamaya göre çok kısa sürede üretebildiği ortaya konmuştur.

Bu tez, üç boyutlu kutu paketleme probleminin farklı varyasyonları için ilk kez kısıt programlama modelleri sunması ve farklı boyutlu problemler için KP ile KTP yöntemlerinin kıyaslamasını yapması ile literatüre katkıda bulunmaktadır.

Anahtar Kelimeler: Kutu Paketleme Problemleri, Kısıt Programlama, Karma Tamsayılı Programlama.

SUMMARY

In this thesis, the bin packaging problem, which is in the background of many problems that we encounter especially in the field of storage, shipment and store layout in the retail sector, is discussed.

After presenting the existing solution methods developed for one-dimensional, two-dimensional and three-dimensional versions of this general problem, two different types of the general problem are discussed and constraint programming (KP) and mixed integer programming (KTP) for one-dimensional, two-dimensional and three-dimensional versions. models have been developed. The models were compared with the data sets in the literature and it was revealed that the developed constraint programming models could produce optimum or near-optimal solutions in a very short time compared to mixed integer programming.

This thesis contributes to the literature by presenting for the first time constraint programming models for different variations of the three-dimensional bin packing problem and comparing KP and KTP methods for different dimensional problems.

Key Words: Bin Packaging Problems, Constraint Programming, Mixed Integer Programming.

TEŞEKKÜR

Başta, yüksek lisans eğitimimde ve akademik hayatımda desteğini ve yardımlarını hiçbir zaman esirgemeyip bilgisi ile bu çalışmanın oluşmasının yolunu açan danışmanım Dr. Öğr. Üyesi Özgür ÜNSAL'a,

İhtiyaç duyduğum zaman desteğini hiç esirgemeyen ve her zaman yardımcı olan Dr. Öğr. Üyesi Tülay KORKUSUZ POLAT'a,

Bütün çalışmam boyunca yanımda olan, desteklerini esirgemeyen ailem ve dostlarıma en içten teşekkürlerimi sunarım.

İÇİNDEKİLER

SİMGELER VE KISALTMALAR DİZİNİ

Simgeler ve Kısaltmalar		**Açıklamalar**
KTP	:	Karma Tamsayılı Programlama
KP	:	Kısıt Programlama
TKDPP	:	Tek Boyutlu Dikey Paketleme Problemleri
İKDPP	:	İki Boyutlu Dikey Paketleme Problemleri
ÜKDPP	:	Üç Boyutlu Dikey Paketleme Problemleri
TKBPP	:	Tek Boyutlu Kutu Paketleme Problemleri
İKBPP	:	İki Boyutlu Kutu Paketleme Problemleri
ÜKBPP	:	Üç Boyutlu Kutu Paketleme Problemleri
AEORS	:	Avrupa Yöneylem Araştırması Dernekleri Birliği
cm	:	Santimetre
dk	:	Dakika
sn	:	Saniye

ŞEKİLLER DİZİNİ

TABLOLAR DİZİNİ

1. GİRİŞ

Kutu paketleme problemleri, amacın bir dizi öğeyi minimum sayıda konteynere veya sabit kapasiteli kutulara paketlemek olduğu bir optimizasyon problemleri sınıfını ifade eder. Bu problemler, yöneylem araştırması ve bilgisayar bilimi alanlarında geniş çapta incelenir ve depo yönetimi, lojistik ve kaynak tahsisi dahil olmak üzere çok çeşitli pratik uygulamalara sahiptir.

Kutu paketleme probleminin en temel versiyonlarından biri, değişen boyutlardaki bir dizi öğenin sabit bir kapasiteye sahip sabit sayıda kutuya paketlenmesini içeren tek boyutlu kutu paketleme problemidir. Amaç, paketleme sürecinde kullanılan kutu sayısını en aza indirmektir. Bu problem NP-zordur, yani onu kesin olarak çözmek için bilinen bir polinom zamanlı algoritma yoktur. Bununla birlikte, optimuma yakın çözümler bulmak için kullanılabilecek birkaç yaklaşık algoritma vardır.

Kutu paketleme probleminin bir başka yaygın versiyonu, paketlenecek öğelerin dikdörtgen olduğu ve kutuların da dikdörtgen olduğu iki boyutlu kutu paketleme problemidir. Amaç, kullanılan kutu sayısını en aza indirirken aynı zamanda boşa harcanan alan miktarını da en aza indirmektir. Bu problem aynı zamanda NP-zordur ve bir dizi sezgisel ve yaklaşık algoritma kullanılarak ele alınabilir.

Kutu paketleme probleminin başka bir versiyonu, öğelerin farklı şekil ve boyutlarda olduğu ve amacın onları değişen şekil ve boyuttaki minimum sayıda konteynere paketlemek olduğu üç boyutlu kutu paketleme problemidir. Bu aynı zamanda NP-zor bir problemdir ve ancak küçük ölçekli problemler kesin çözüm yöntemleri ile optimum çözümler elde edilebilmektedir.

Bu genel problemin gerçek hayat uygulamalarından doğan pek çok farklı varyasyonu vardır. Bunlara örnek olarak, şerit paketleme problemleri, tek boyutlu kutu paketleme problemine benzeyen ancak en fazla öğenin tek bir kutuya yerleştirilmeye çalışıldığı sırt çantası problemi ve dikey (ortogonal) paketleme problemleri gösterilebilir.

Bu çalışmamızda iki farklı varyasyonu ele almaktayız:

- Literatürde kutu paketleme olarak bilinen elimizdeki tüm öğeleri en az sayıda kutuya paketlemek paketleme problemi (KPP)
- Literatürde dikey (ortogonal) paketleme olarak bilinen tek bir kutuya en fazla sayıda (ya da değerde) öğeyi sığdırma problemi (DPP)

Farkedilebileceği üzere, iki problem birbirleriyle oldukça ilişkili olmalarına rağmen farklı amaç fonksiyonu ve kısıtlara sahiptirler. Her iki varyasyonun gerçek dünya uygulamalarında küçük ölçekli problem örnekleri ile nadiren karşılaşılmakta, büyük ölçekli kutu paketleme problemi örneklerinin çözümü için genellikle sezgisel veya yaklaşık algoritmalar kullanılmaktadır. Bu algoritmaların tipik olarak nispeten kısa bir süre içinde optimuma yakın çözümler sunabiliyor olmaları onları gerçek dünya uygulamaları için kesin algoritmalara kıyasla çok uygun hale getirir.

Kullanım alanı ve değişen amaçar için kullanılan kutu paketleme problemleri çözüm yöntemleri şunlardır:

- Kesin algoritmalar: Bu algoritmalar, olası tüm çözümleri göz önünde bulundurarak ve en uygun olanı seçerek kutu paketleme sorununa kesin bir çözüm sunar. Bununla birlikte, kesin algoritmalar genellikle hesaplama açısından zor ve özellikle büyük ölçekli problemler için bir çözüm bulmak uzun zaman alabilir.

    - Doğrusal programlama ve Simpleks algoritması
    - Dal ve sınır algoritmaları
    - Dinamik programlama

- Yaklaşık algoritmalar: Bu algoritmalar, kutu paketleme problemine hızlı bir şekilde optimuma belli bir ölçüde yakınlık garantisi veren çözümler üretmek için kullanılır.
- Metasezgisel algoritmalar: Bu genel algoritmalar, kutu paketleme sorununa optimuma genellikle çok yakın çözümleri hızlı bir şekilde bulmak için kullanılır. Yaklaşık algoritmalardan farkı ise optimuma yakınlık garantisi sunmamasıdır.

- Yapay zeka temelli algoritmalar: Bu yöntemler, kutu paketleme sorunlarını kesin ya da optimuma yakın olarak çözmek için Sinir Ağları, Makine Öğrenimi ve Derin Öğrenme gibi teknikleri kullanır.

  - Kısıt programlama: Yapay zeka sınıfında gösterilen yayılma algoritmalarını kullanan kısıt programlama hem kesin ve hem de sezgisel çözüm yöntemi olarak kullanılabilmektedir.

Üstte kısaca belirtilen çözüm yöntemleri tezin ikinci bölümünde açıklanacaktır. Kullanılacak yöntemin seçimi ise çalışılan probleme, uygulama alanına ve çözüm kalitesi ile hesaplama süresi arasında istenen dengeye bağlıdır. Bu ödünleşimin göz ardı edilmesi önemli sonuçlar doğuracaktır. Öyle ki, gün içinde defalarca çözülmesi gereken büyük ölçekli bir problem için kesin yöntemlerin kullanılmasının mümkün olmayacağı gibi, sadece bir kez çözülecek ve geniş zamana sahip olunan bir uygulamada yaklaşık algoritmalar ile çözüm üretmek büyük kayıplara sebep olabilecektir.

Kutu paketleme problemleri birçok alanda ve gerçek dünya süreçlerinde ortaya çıkmaktadır. Kutu paketleme problemlerinin bazı uygulama alanları şunlardır.

- Depo yönetimi: Kutu paketleme algoritmaları, belirli bir öğe grubunu depolamak için gereken kutu veya konteyner sayısını en aza indirerek, malların depolarda depolanmasını optimize etmek için kullanılabilir.
- Lojistik: Kutu paketleme algoritmaları, belirli bir öğe grubunu taşımak için gereken konteyner sayısını en aza indirerek konteynerlerin gemilere, kamyonlara ve trenlere yüklenmesini optimize etmek için kullanılabilir.
- Kaynak tahsisi: Kutu paketleme algoritmaları, bant genişliği veya disk alanı gibi kaynakları, israfı en aza indirecek ve genel verimliliği en üst düzeye çıkaracak şekilde farklı görevlere tahsis etmek için kullanılabilir.
- Stok kesme problemi: Kutu paketleme algoritmaları, israfı en aza indirmek için büyük bir malzeme parçasının daha küçük parçalara kesileceği stok kesme probleminde yaygın olarak kullanılır.

- Üretim: Kutu paketleme algoritmaları, bir üretim montaj hattındaki parçaların yerleşimini optimize etmek veya israfı en aza indirmek ve verimliliği artırmak için malzeme tabakalarının kesilmesini optimize etmek için kullanılabilir.
- Çizelgeleme: Kutu paketleme algoritmaları, makineler veya çalışanlar gibi kaynakların kullanımını verimliliği en üst düzeye çıkaracak ve boşta kalma süresini en aza indirecek şekilde planlamak için kullanılabilir.
- Düzensiz şekillerin paketlenmesi: Kutu paketleme algoritmaları, düzensiz şekilli ürünlerin bir kutu veya palet içinde paketlenmesi gibi, düzensiz şekilli öğeleri kaplara paketlemek için kullanılabilir.

1.1. Tezin Amacı, Katkısı ve İçeriği

Kutu paketleme problemleri gerçek dünyada çokça karşılaşılan ve eniyi çözümü için yüksek hesaplama eforu gerektiren karmaşık optimizasyon problemlerindendir. Gerçek dünyada oluşabilecek kutu paketleme problemlerine karşı bir önceki bölümde kısaca değinildiği üzere pek çok çözüm yöntemi geliştirilmiştir. Ancak gerçek dünya problemlerinde ele alınan kutu boyutları, hacimleri ve ağırlıkları çeşitlilik gösterdiğinden ve problemin çözümü için istenilen hedefler neticesinde ele alınan problemlerin çözümü bilgisayarda çok fazla zaman ihtiyacına neden olmaktadır.

Bu tez ile tek boyutlu, iki boyutlu ve üç boyutlu KPP ve DPP versiyonları ele alınmış, bu versiyonlar kısıt programlama ve karma tam sayılı programlama çözüm yöntemleri kullanılarak modellenip çözülmüş ve kullanılan çözüm yöntemlerinin performansları değerlendirilmiştir. Böylece geliştirilen kısıt programlama modelleri ile kısa sürede kesin çözüm yöntemine yakın sonuçlar elde edilebileceği ortaya konmuştur. Ayrıca belirtilmelidir ki, bu tez ile üç boyutlu problemler (KPP ve DPP) için bir kısıt programlama modeli ilk kez ortaya konmakta ve yine bu üç problem için karma tam sayılı programlama ve kısıt programlama modelleri bir arada ilk kez kıyaslanmaktadır.

Tezin geri kalanının içeriği şu şekilde planlanmıştır: ikinci bölümde kutu paketleme problemlerinin tek boyutlu, iki boyutlu ve üç boyutlu varyasyonları karar değişkenleri, kısıtlar, amaç fonksiyonları ve gerçek dünya kullanım alanları, her bir varyant için kullanılan çözüm yöntemleri anlatılmıştır. Üçüncü bölümde ise tek boyutlu, iki boyutlu ve üç boyutlu KPP ve DPP problemlerinin matematiksel modellemeleri kısıt programlama ve karma tamsayılı programlama olacak şekilde iki farklı çözüm metodolojisi ile ele alınmıştır. Dördüncü bölümde, üçüncü bölümde belirtilen matematiksel modellerin Avrupa Yöneylem Araştırması Dernekleri Birliği tarafından paylaşılan gerçek dünya verileri üzerinde hesaplamalı deneyler gerçekleştirilmiştir. Son bölümde ise modellerin çalışma performansları ve vaka çalışmasının sonuçları ile gelecek çalışmalar değerlendirilmiştir.

2. KUTU PAKETLEME PROBLEMLERİ

Kutu paketleme problemi (KPP), kullanılan kutu sayısını en aza indirmek amacıyla, belirli boyutlardaki öğelerin sınırlı kapasiteye sahip kaplara veya kutulara yerleştirilmesini bir başka deyişle paketlenmesini ele alan bir kombinatoryal optimizasyon problemidir. Problem, uygulamaya bağlı olarak tek boyutlu, iki boyutlu ve üç boyutlu gibi çeşitli versiyonlara gruplandırılarak çözüm metodolojileri geliştirilmiştir [Lodi, 2002]. KPP'nin bir varyasyonu olarak tek bir kutuya en fazla sayıda (ya da değere sahip) öğenin paketlenmesini amaçlayan dikey paketleme problemi (DPP) de aynı şekilde 3 farklı boyuttaki versiyonda incelenebilir. Bu problemin tek boyutlu hali ise sırt çantası problemiyle eşdeğerdir.

Bu bölümde problemlerin tanımları açıkça verilmiş sonrasında ise bu problemler için literatürde kullanılan çözüm yöntemleri açıklanmıştır. Belirtilmelidir ki problemlerin yapısal benzerlikleri sebebiyle KPP için geliştirilen algoritmaların çoğu DPP'ye uyarlanabilmektedir. Bu sebeple problem tanımları her iki problem için de verilmiş ancak yöntemler ağırlıklı olarak KPP üzerinden açıklanmıştır.

Tek boyutlu paketleme problemleri, farklı boyutlarda sahip belirli bir ürün setini sınırlı kapasiteye sahip bir yükleme alanına doldurulması ile ilgilenir, örneğin, öğeleri bir kutuya veya bir torbaya paketlemek gibi.

İki boyutlu kutu paketleme, iki boyutlu bir dizi öğeyi sınırlı bir alana sığdırmakla ilgilenir, örneğin bir palet veya bir malzeme levhası üzerindeki öğeleri paketlemek gibi.

Üç boyutlu kutu paketleme, üç boyutlu öğelerin sınırlı kapasiteye sahip bir konteynere yerleştirilmesi ilgilenir; örneğin, öğeleri bir nakliye konteynerine veya bir depoya paketlemek gibi [Martello, 2000].

Sıradaki bölümde değinileceği üzere yaklaşık yöntemler, sezgisel yöntemler ve kesin algoritmalar dahil olmak üzere paketleme problemleri için çeşitli çözüm metodolojileri vardır. Kısıt programlama ve karma tamsayılı programlama, bu problemleri çözmek için kullanılan iki popüler ve etkili yaklaşımdır. Kısıt programlama, karmaşık kombinatoryal problemlerin verimli bir şekilde modellenmesine ve çözülmesine izin veren bir programlama paradigması iken, karma tamsayı programlama, doğrusal programlamayı tamsayılı değişkenlerle birleştiren bir matematiksel optimizasyon tekniğidir [Gonçalves, 2007].

2.1. Tek Boyutlu Paketleme Problemleri

Tek boyutlu kutu paketleme problemi TBKPP, kullanılan kutu sayısını en aza indirmek amacıyla, l_i boyutundaki ($i \in N = \{1,2,3, \dots, n\}$) n öğenin sınırlı kapasiteye (L) sahip kutulara veya şeritlere üst üste gelmeyecek şekilde yerleştirilmesini içeren bir kombinatoryal optimizasyon problemidir [López-Camacho, 2014]. Bir başka deyişle, problem çözüldüğünde eldeki *n* öğenin tamamı *L* kapasiteye sahip en az sayıda kutu kullanarak paketlenmiş olacaktır. Bu tanım, ele alınan problemin şerit kesme problemine eşdeğer olduğunu ortaya koymaktadır.

Tek boyutlu dikey (ortogonal) paketleme problemi TBDPP, L kapasitesine sahip tek bir kutunun kullanıldığı ve l_i boyutundaki toplam n öğe içinden en fazla sayıda (ya da değerde) öğeyi bu kutuya paketlemeyi amaçlayan bir kombinatoryal optimizasyon problemdir. Bir başka deyişle, problem çözüldüğünde eldeki *n* öğenin bir alt kümesi *L* kapasiteye sahip bir kutuya paketlenmiş olacaktır.

TBKPP ve TBDPP, öğelerin nakliye veya depolama için alanlara paketlenmesini eniyilemek için kullanıldığı lojistik ve envanter yönetimi gibi çeşitli endüstrilerde geniş bir uygulama yelpazesine sahiptir. Problemlerin etkili bir biçimde çözülmesi, bu endüstrilerde nakliye, depolama ve elleçleme ile ilgili maliyetlerin düşürülmesine yardımcı olmaktadır. Bunun yanı sıra, kamyonların ağırlık kapasitelerine göre yüklenmesinden televizyon reklamlarının yayınların aralarına yerleştirilmesi gibi uygulama alanları bulmuştur. Ayrıca şerit kesme problemleri (kablo, kereste, kağıt, gibi öğelerin tek boyut gözeterek kesilmesi) bu problemin uygulama alanlarındandır [Wei, 2020].

2.1.1. TBKPP Çözüm Yöntemleri

TBKPP özelinde spesifik olarak hazırlanmış yöntemler aşağıda anlatılacaktır. Diğer boyutlu kutulama problemleri ile ortak çözüm yöntemleri sonrasında bahsedilecektir.

TBKPP Spesifik Çözüm Yöntemleri:

- First Fit Algoritması (FFA)

FFA, TBKPP çözmek için basit bir sezgisel yöntemdir. Algoritmanın temel fikri, bir dizi öğeyi yinelemek ve her öğeyi, onu yerleştirmek için yeterli kapasiteye sahip olan ilk bölmeye yerleştirmektir. Böyle bir bölme yoksa yeni bir bölme açılır ve öğe bu bölmeye yerleştirilir. Algoritma, tüm öğeler kutulara yerleştirilene kadar bu şekilde ilerler [Gyárfás, 1988].

FFA'nın ana avantajlarından biri basitliğidir. Anlaşılması ve uygulanması kolaydır ve orta büyüklükteki problemler için hızlı çalışır [Gyárfás, 1988]. Bununla birlikte, algoritmanın optimal bir çözüm üretmesi garanti edilmez ve fazla sayıda bölmenin kullanılmasına yol açabilir [man Jr., 1996].

Genel olarak, FFA, asıl amaç optimal bir çözüm yerine hızlı ve anlaşılması kolay bir çözüm bulmak olduğunda TBKPP çözmek için kullanılmaktadır. Problemdeki öğe sayısı büyüdükçe bu yöntemin kullanımı problemin çözümü için yeterli olmayacaktır.

- First Fit Decreasing Algoritması (FFDA)

FFDA, TBKPP çözmek için yaygın olarak kullanılan FFA'nın bir çeşididir. İki algoritma arasındaki temel fark, paketleme işlemine başlamadan önce öğelerin boyutlarına göre büyükten küçüğe göre sıralanmasıdır [Yao, 1980]. Bu şekilde, en büyük öğeler önce yerleştirilir ve algoritma kutularda kalan boş alandan yararlanır.

FFDA, TBKPP çözmek için basit ve verimli bir yöntemdir ve genellikle diğer algoritmalarla karşılaştırma için bir kıyaslama noktası olarak kullanılır [Gyárfás, 1988]. Algoritma her zaman optimal bir çözüm bulamamaktadır, ancak nispeten az sayıda kutu ile iyi çözümler üretebilmektedir. FFDA, FFA'nın çalışma performansını ve çözüm kalitesini artırır.

FFDA maddelerin çok farklı boyutlarda olduğu problemlerin çözümü için iyi bir algoritmadır ve diğer algoritmalarla karşılaştırma yapmak için bir ölçüt olarak kullanılabilir. Ancak FFA'da olduğu gibi öğe sayısının fazla olduğu durumlarda yeterli değildir.

• Refined First Fit Algoritması (RFFA)

RFFA, TBKPP çözmek için yaygın olarak kullanılan FFA'in bir çeşididir. İki algoritma arasındaki temel fark, RFFA algoritmasının, her bir öğenin nereye yerleştirileceği konusunda daha doğru kararlar vermek için öğeler ve kutular hakkında ek bilgiler kullanmasıdır [Yao, 1980].

RFFA, FFDA benzer şekilde öğeleri azalan boyut sırasına göre sıralayarak başlar. Bununla birlikte, RFFA her kutuda kalan boş alanı da takip eder ve bu bilgiyi bir sonraki öğe için hangi bölmenin kullanılacağını belirlemek için kullanır [man Jr. (1996)].

RFFA, her kutuda kalan boş alanı hesaba katarak ve paketleme sürecini yönlendirmek için kullanarak FFDA'den daha kullanışlıdır. Bu, daha az kutu kullanılmasına ve mevcut alanın daha verimli kullanılmasına yardımcı olur.

RFFA, bazı TBKPP için en uygun çözümleri bulmaktadır ancak FFDA'den daha karmaşık ve zaman alıcıdır. Problemin kısıtlarına göre FFDA'den daha iyi ve kötü performans göstermektedir.

• Best Fit Algoritması (BFA) ve Best Fit Decreasing Algoritması (BFDA)

BFA, TBKPP çözmek için yaygın bir çözüm metodolojisidir. Algoritma, paketlenecek öğeleri yineleyerek ve mevcut öğenin en az kalan alanla sığacağı kutuyu bularak çalışır. Bu yaklaşım, kullanılan kutu sayısını ve dolayısıyla toplam atığı en aza indirmeyi amaçlar [Kenyon, 1996]. BFA, paketleme işlemine başlamadan önce öğeleri azalan boyut sırasına göre sıralayarak geliştirilebilir; bu, en uygun çözümü garanti eden veya en azından iyi bir yaklaşık çözüm üreten En Uygun Azalan (BFDA) olarak bilinmektedir. BFDA, TBKPP çözmek için tüm algoritmalar arasında en verimli algoritmalardan biri olarak kabul edilmektedir.

2.2. İki Boyutlu Paketleme Problemleri

İki boyutlu kutu paketleme problemi İBKPP, kullanılan kutu sayısını en aza indirmek amacıyla, (l_i, w_i) ölçülerindeki ($i \in N = \{1,2,3, \dots, n\}$) n dikdörtgen öğenin sınırlı kapasiteye (L, W) sahip dikdörten alanlara yerleştirilmesini içeren bir kombinatoryal optimizasyon problemidir. Bu problemde hiçbir öğe üst üste gelmemelidir [Gonçalves, 2013]. Bir başka deyişle, problem çözüldüğünde eldeki n öğenin tamamı (L,W) boyutlarına sahip en az sayıda kutu kullanarak paketlenmiş olacaktır.

İki boyutlu dikey (ortogonal) paketleme problemi İBDPP, (L, W) boyutlarına sahip tek bir dikdörtgen alanın (kutunun) kullanıldığı ve (l_i, w_i) ölçülerindeki toplam n dikdörtgen öğe içinden en fazla sayıda (ya da değerde) öğeyi üst üste gelmeyecek şekilde bu kutuya paketlemeyi amaçlayan bir kombinatoryal optimizasyon problemdir. Belirtilmelidir ki, ele aldığımız bu problemde kutuya yerleştirilen dikdörtlerin çevrilmeleri mümkün değildir. Bir başka deyişle, (l_i, w_i) ölçüleriyle verilen dikdörtgen bir öğe 90 derece çevrilerek (w_i, l_i) ölçülerindeki bir dikdörtgen olarak kutuya yerleştirilemez.

Bu tür problemlere, ürünlerin nakliye veya depolama için konteynırlara veya paletlere paketlenmesi gereken imalat ve lojistikte yaygın olarak rastlanır.

2.2.1. İBKPP Çözüm Yöntemleri

İBKPP çözüm yöntemleri sezgisel ve kesin çözümler olarak sınıflandırılmıştır. İBKPP özelinde spesifik olarak hazırlanmış yöntemler aşağıda anlatılacaktır. Diğer boyutlu kutulama problemleri ile ortak çözüm yöntemleri sonrasında bahsedilecektir.

İBKPP Spesifik Çözüm Yöntemleri

- Nesting Algoritması (NA)

NA, İBKPP çözmek için bir yöntemdir. Paketlenecek öğeleri iç içe bir şekilde düzenleyen, kompakt ve verimli bir paketleme oluşturan sezgisel bir yöntemdir. Algoritma, en büyük öğeyi kutuya yerleştirerek başlar ve ardından kalan öğeleri ilk öğe tarafından oluşturulan boş alanlara yerleştirir [Goodman, 1994]. Algoritma, tüm öğeler yerleştirilene veya daha fazla yer kalmayana kadar boş alanları daha küçük öğelerle doldurarak bu şekilde devam eder. NA basit, hızlı ve uygulaması kolay bir yöntemdir, ancak öğelerin belirli yönelimini veya boş alanların konumunu dikkate almadığından her zaman optimal bir çözüm üretememektedir.

- Slide Pack Algoritması (SPA)

SPA, İBKPP çözmek için kullanılan bir yöntemdir. Öğeleri bir kutuya paketlemek için kayan pencere tekniğini kullanan sezgisel bir yöntemdir. Algoritma, kutunun sol üst köşesine bir öğe yerleştirerek ve ardından bir sonraki öğe için bir sonraki en uygun konumu bulmak üzere kalan alanın üzerine bir pencere kaydırarak başlar. Pencere, bir sonraki öğe için olası tüm konumları kapsayacak şekilde sağa, aşağı ve ardından çapraz olarak hareket ettirilir. Geçerli pencereye en uygun öğe daha sonra o konuma yerleştirilir ve tüm öğeler yerleştirilene veya boş yer kalmayana kadar işlem tekrarlanır [Smith, 2014].

- Skyline Pack Algoritması (SKYPA)

SKYPA, İBKPP için sezgisel bir algoritmadır. Mevcut paketleme çözümünde en alçakta bulunan dikdörtgenin üst kenarını temsil eden bir çizgi olan ufuk çizgisi kavramına dayanmaktadır. Algoritma, ilk dikdörtgeni orijine yerleştirerek başlar ve ardından geçerli ufuk çizgisine dayalı olarak en uygun konumu bularak sonraki dikdörtgeni yinelemeli olarak kutuya yerleştirir [Wei, 2013].

2.3. Üç Boyutlu Kutu Paketleme Problemleri (ÜBKPP)

Üç boyutlu kutu paketleme problemi ÜBKPP, kullanılan kutu sayısını en aza indirmek amacıyla, (l_i, w_i, h_i) ölçülerindeki ($i \in N = \{1,2,3, \dots, n\}$) n dikdörtgen prizma şeklindeki öğenin sınırlı kapasiteye (L, W, H) sahip yine dikdörten prizmalara yerleştirilmesini içeren bir kombinatoryal optimizasyon problemidir. Bu problemde hiçbir öğe hiç bir düzlemde üst üste gelmemelidir [Hu, 2017]. Bir başka deyişle, problem çözüldüğünde eldeki n öğenin tamamı (L,W,H) boyutlarına sahip en az sayıda dikdörtgen prizma şeklinde kutu kullanarak paketlenmiş olacaktır.

Üç boyutlu dikey (ortogonal) paketleme problemi ÜBDPP, (L, W, H) boyutlarına sahip tek bir dikdörtgen prizmanın kullanıldığı ve (l_i, w_i, h_i) ölçülerindeki toplam n dikdörtgen prizma şeklinde öğe içinden en fazla sayıda (ya da değerde) öğeyi hiçbir düzlemde üst üste gelmeyecek şekilde bu kutuya paketlemeyi amaçlayan bir kombinatoryal optimizasyon problemdir. İBDPP’de olduğu gibi bu problemde de yerleştirilen dikdörtgen prizmaların çevrilmeleri mümkün değildir.

ÜBKPP ve ÜBDPP amacın üç boyutlu nesneleri sabit sayıda daha büyük üç boyutlu kutulara yada konteynırlara paketlemek; kutu sayısını, nakliye konteynırları ve depolama alanlarında kullanılmayan alanı en aza indirmek olduğu bir optimizasyon problemleri sınıfını oluşturur [Wang, 2010]. Bu problem lojistik, üretim ve depolama gibi çeşitli endüstrilerde kullanılmaktadır ve stok yönetimi, üretim planlama ve dağıtım gibi alanlara da uygulanabilir. Kesin yöntemler, sezgisel yöntemler ve metasezgisel yöntemler dahil olmak üzere ÜBKPP çözmek için çeşitli algoritmik yaklaşımlar vardır.

2.3.1. ÜBKPP Çözüm Yöntemleri

ÜBKPP çözüm yöntemleri sezgisel, metasezgisel ve kesin çözümler olarak sınıflandırılmıştır. ÜBKPP özelinde spesifik olarak hazırlanmış yöntemler aşağıda anlatılacaktır. Diğer boyutlu kutulama problemleri ile ortak çözüm yöntemleri sonrasında bahsedilecektir.

ÜBKPP Spesifik Çözüm Yöntemleri:

• The Simple and Efficient Lifting and Insertion Algoritması (SELIA)

SELIA, ÜBKPP çözmek için sezgisel bir yöntemdir. Farklı boyutlara ve yönlere sahip nesneleri sınırlı sayıda konteynıra veya sabit hacimli kutulara paketlemeyi içeren, iyi bilinen ÜBKPP bir çeşididir. SELIA, kullanılan kutu sayısını ve boşa harcanan alan miktarını en aza indirmeye çalışırken, nesneleri kutulara verimli bir şekilde paketlemek için kaldırma ve yerleştirme tekniklerinin bir kombinasyonunu kullanır [Lin, 1993].

• Smile Algoritması (SA)

SA, ÜBKPP çözmek için sezgisel bir yöntemdir. SELIA ile BFDA değiştirilmiş bir versiyonunun birleşimine dayanır. Algoritma, öğeleri azalan hacim sırasına göre sıralayarak başlar ve ardından kalan boş alanı takip ederken öğeleri bölmeye yerleştirmek için SELIA algoritmasını kullanır. Ardından, değiştirilmiş BFDA, kalan boş alanı hesaba katarak bir sonraki öğeyi yerleştirmek için en iyi bölmeyi bulmak için kullanılır [Sridhar, 2017].

• Wall Building Algoritması (WBA)

WBA, ÜBKPP çözmek için sezgisel bir yöntemdir. Algoritma, ilk öğeyi kutuya yerleştirerek başlar ve ardından öğeleri birer birer eklemeye devam eder. Algoritma, her öğe için, daha önce yerleştirilmiş öğelerin yanına yerleştirilip yerleştirilemeyeceğini kontrol ederek bir öğe "duvarı" oluşturur. Yerleştirilebilirse duvara eklenir, yoksa yeni bir duvar oluşturulur. Algoritma, tüm öğeler yerleştirilene veya mevcut duvarlara başka öğe eklenemeyene kadar devam eder. Algoritmanın amacı, bölmedeki duvar sayısını ve boşa harcanan alanı en aza indirmektir [Sridhar, 2017].

• Tree Generating Heuristic Algoritması (TGHA)

TGHA, ÜBKPP çözmek için bir yöntemdir. Algoritma, ağaçtaki her düğümün tek bir öğeyi temsil ettiği, bölmedeki öğelerin düzenini temsil etmek için bir ikili ağaç yapısı oluşturmayı içerir. TGHA, en büyük hacme sahip öğeyi seçip ağacın köküne yerleştirerek başlar. Algoritma daha sonra geri kalan öğeleri yinelemeli olarak iki gruba ayırır ve bunları kök düğümün sol ve sağ alt ağaçlarına yerleştirir. Algoritma, tüm öğeler ağaca yerleştirilene kadar bu şekilde devam eder [Burke, 2006].

• The Stability Method Algoritması (TSMA)

TSMA, ÜBKPP çözmek için sezgisel bir algoritmadır. Algoritma, önce öğeleri boyutlarına göre azalan sırayla sıralayarak ve ardından kararlılığa dayalı bir sezgisel yöntem kullanarak bunları birer birer kutuya yerleştirerek çalışır. Bir kutu konfigürasyonunun kararlılığı, kutudaki boş alanın hacminin kutunun toplam hacmine oranıyla belirlenir. Algoritma, her öğeyi bölmeye yerleştirmek için konumu dikkatlice seçerek kutu yapılandırmasının kararlılığını en üst düzeye çıkarmayı amaçlar [Chazelle, 1983].

• Evolutionary Algoritması (EA)

EA, doğal evrim ilkelerine dayanan ÜBKPP çözmek için bir yöntemdir. Aday çözümlerden oluşan bir popülasyon oluşturarak, uygunluklarını değerlendirerek ve ardından yeniden üretmek ve yeni nesiller için en iyi çözümleri seçerek doğal seçilim sürecini simüle eder. Algoritma, çözümlerin kalitesini artırmak için çaprazlama, mutasyon ve seçim gibi genetik operatörleri yinelemeli olarak uygular. EA'nın, geniş bir arama alanını keşfettiği ve optimuma yakın çözümler bulabildikleri için, karmaşık ve büyük ölçekli 3B kutu paketleme sorunlarını çözmek için etkili olduğu bulunmuştur [Levine, 2004]. Ancak, aynı zamanda hesaplama açısından zor ve çok sayıda işlev değerlendirmesi gerektirebilmektedir.

2.4. KPP ve DPP Genel Çözüm Yöntemleri

Kutu paketleme problemlerinin boyut bazında spesifik çözüm yöntemleri yukarıda belirtilmiştir. Bu kısımda KPP ve DPP versiyonları için ortak olarak kullanılabilecek genel çözüm yöntemleri anlatılmıştır.

2.4.1. Kesin Çözüm Yöntemleri

Paketleme problemleri için kesin çözüm yöntemleri, tüm olası çözümleri kapsamlı bir şekilde araştırarak soruna en uygun çözümü bulmayı içermektedir. Bu yöntemler tipik olarak küçük ölçekli problemler için kullanılmaktadır. Bu yöntemler kesin olarak kabul edilmektedir, çünkü en uygun çözümü bulmaları garanti edilmektedir. Ancak büyük ölçekli problemler için zaman alıcı ve hesaplama açısından zordur.

• Linear Programming (LP) ve Simpleks Yöntemi

LP, doğrusal optimizasyon problemlerini çözmek için kullanılan matematiksel bir optimizasyon yöntemidir. Bir dizi kısıtlamaya tabi olan doğrusal bir amaç fonksiyonunun en aza indirilmesini veya en üst düzeye çıkarılmasını içerir. Bu yöntem, verilen kutu kapasitesi kısıtlamalar dahilinde farklı boyutlardaki öğeleri paketlemek için kullanılan kutu sayısını en aza indirmenin amaçlandığı TBKPP, İBKPP ve ÜBKPP kutu paketleme problemlerine uygulanabilmektedir [Martello, 1998].

• Dynamic Programming (DP)

DP, TBKPP, İBKPP ve ÜBKPP alanındaki sorunları çözmek için kullanılabilen bir matematiksel optimizasyon yöntemidir. Problemi daha küçük alt problemlere ayırmayı ve bu alt problemleri yinelemeli bir şekilde çözmeyi içerir. Her bir alt problemin çözümü daha sonra kullanılmak üzere bir tabloda saklanır ve bu da gelecekteki alt problemler için verimli çözümlerin bulunmasına olanak tanır [Brandao, 2016]. Dinamik programlama, alt problemlerin üst üste geldiği problemler için özellikle yararlıdır, çünkü bu alt problemlerin çözümlerini depolayabilir ve yeniden kullanabilir, böylece daha verimli bir çözüm süreci ortaya çıkar. Bununla birlikte, bu yöntem, özellikle daha büyük problemler için hesaplama açısından zordur.

• Mixed Integer Programming (KTP)

KTP, LP ile tamsayılı programlamayı birleştiren bir matematiksel optimizasyon yöntemidir. Kutu paketleme problemlerinde, KTP, bazı değişkenlerin tamsayı değerleri alması gereken TBKPP, İBKPP ve ÜBKPP çözmek için kullanılabilir. Kutu paketleme bağlamında, bu, öğelerin belirli kutulara atanmasını veya kullanılacak kutu sayısının belirlenmesini içerebilir. KTP algoritmaları, bir dizi kısıtlamayı ve hedefi karşılaması gereken en uygun çözümü bulmak için LP teknikleri ve arama prosedürlerinin bir kombinasyonunu kullanır [Martello, 1998]. KTP algoritmaları tarafından üretilen çözüm, genellikle optimal çözüme iyi bir yaklaşımdır ve daha fazla optimizasyon için bir başlangıç noktası olarak kullanılabilir. Ancak, KTP hesaplama açısından zordur ve büyük ölçekli problemlerde düşük performanslıdır.

• Branch and Bound Algoritması (BBA)

BBA, kutu paketleme problemi gibi kombinatoryal optimizasyon problemlerini çözmek için bir yöntemdir. Algoritma, problemi daha küçük alt problemlere bölerek ve ardından her bir alt problemi ayrı ayrı çözerek çalışır. Her bir alt problem için en uygun çözümü tahmin etmek için bir sınırlayıcı fonksiyon kullanır ve ardından yalnızca mevcut en iyi çözümü iyileştirme potansiyeline sahip olan alt problemleri araştırır. Bu işlem, global bir optimum bulunana kadar tekrarlanır [Iori, 2021]. BBA'nın ana avantajlarından biri, arama alanını önemli ölçüde azaltmak için simetriler ve alt modülerlik gibi probleme özgü yapılardan yararlanabilmesidir. Ancak, özellikle büyük ölçekli problemler için hesaplama açısından zordur.

• Branch and Cut Algoritması (BCA)

BCA, BBA tekniklerini doğrusal programlama (LP) ile birleştiren, kutu paketleme problemlerini çözmek için kesin bir algoritmadır. Algoritma, problemin lineer gevşemesini çözerek başlar ve daha sonra, eğer çözüm uygun değilse, kesirli değişkenlere dallanarak yeni alt problemler üretir. Bu alt problemler daha sonra LP tekniği ile çözülür ve uygun bir çözüm bulunana kadar süreç tekrarlanır. Algoritma, uygun olmayan çözümleri ortadan kaldıran kısıtlamalar ekleyerek LP gevşemesini güçlendirmek için kesme düzlemlerini kullanır [Hifi, 2010].

• Branch and Price Algoritması (BPA)

BPA, kutu paketleme problemlerini çözmek için kesin bir çözüm yöntemidir. BBA algoritmasının bir uzantısıdır, burada karar değişkenlerinde dallanmaya ek olarak, algoritma ayrıca arama işlemi sırasında yeni değişkenler (sütunlar) üretir ve fiyatlandırır. Algoritma, kutu paketleme problemini bir karma tamsayılı programlama (KTP) olarak formüle ederek başlar ve ardından arama işlemi sırasında dinamik olarak değişkenler (sütunlar) oluşturmak için bir sütun oluşturma tekniği kullanır. Yeni değişkenler oluşturmak için kullanılan fiyatlandırma sorunu, tipik olarak bir LP algoritması kullanılarak çözülür. Algoritma, karar değişkenleri üzerinde dallanmaya ve optimal bir çözüm bulunana veya önceden tanımlanmış bir durdurma kriteri karşılanana kadar yeni değişkenler üretmeye devam eder [Iori, 2021]. Branch and Price, problemin yapısından yararlanabildiği ve değişkenler oluşturmak için probleme özgü bilgileri kullanabildiği için büyük ölçekli ve karmaşık kutu paketleme problemlerini çözmede özellikle etkilidir.

• Constraint Programming (KP)

KP, optimize etmek için açık bir amaç fonksiyonu yazmak yerine kısıtlamaları ve değişkenler arasındaki ilişkileri belirterek optimizasyon problemlerini çözen kısıtlama tabanlı bir optimizasyon yöntemidir. Kutu paketleme problemleri bağlamında, KP, öğe boyutlarının, kutu kapasitelerinin ve yerleştirme kurallarının kısıtlamalarını modellemek ve ardından tüm kısıtlamaları karşılayan bir çözüm bulmak için kullanılabilir [Delorme, 2016]. Bu yöntem, çok sayıda kısıtlama içeren ve arama sürecinde yüksek düzeyde esneklik gerektiren problemler için çok uygundur. Belirtilmelidir ki, KP hem sezgisel hem de kesin çözüm yöntemi olarak kullanılabilmektedir [Delorme, 2016].

2.4.2. Meta-Sezgisel Çözüm Yöntemleri

Meta-sezgisel yöntemler çeşitli doğal ve yapay süreçleri taklit etme fikrine dayanmaktadır ve nispeten kısa sürede iyi çözümler bulabilmektedir.

• Genetic Algorithm (GA)

GA, doğal evrim ilkelerinden ilham alan meta-sezgisel bir yöntemdir. Algoritma, potansiyel çözümleri, öğenin kutuya yerleştirilmesini temsil eden, kromozomlar olarak bilinen bir dizi ikili dizi olarak temsil ederek çalışır. Algoritma daha sonra, birden fazla nesil boyunca çözüm popülasyonunu geliştirmek için seçim, çaprazlama ve mutasyon gibi genetik operatörleri kullanır. Bu süreç, tatmin edici bir çözüm bulunana veya bir durdurma kriteri sağlanana kadar devam eder. GA'nın, özellikle parça sayısının veya kutu boyutunun büyük olduğu ve sorunun kesin yöntemlere uygun olmadığı durumlarda, kutu paketleme problemlerini çözmede etkili olduğu gösterilmiştir [Burke, 2006].

• Simulated Annealing (SA)

SA, optimizasyon problemlerini çözmek için kullanılan bir meta-sezgisel algoritmadır. Algoritma, kusurları azaltmak ve malzemenin stabilitesini artırmak için bir malzemenin ısıtıldığı ve ardından yavaşça soğutulduğu metalürjideki tavlama kavramına dayanmaktadır. Benzer şekilde SA algoritması, optimum çözümü bulmak için yüksek bir sıcaklıkla başlayıp zamanla kademeli olarak azaltarak bu süreci simüle eder [Tole, 2023].

SA algoritması bir ilk çözümle başlar ve ardından mevcut çözümde küçük rastgele değişiklikler yaparak yeni bir çözüm üretir. Bu değişiklikler, yeni ve mevcut çözümlerin kalitesindeki farka ve mevcut sıcaklığa bağlı olarak kabul edilir veya reddedilir. Kabul olasılığı, daha yüksek sıcaklıklarda daha yüksektir ve algoritmanın çözüm uzayını daha geniş bir şekilde keşfetmesine izin verir. Sıcaklık düştükçe kabul olasılığı da azalır ve algoritma yerel olarak optimal bir çözüme yakınsamaya başlar. Algoritmanın performansı, soğutma programının seçimine ve ilk çözüme bağlıdır [Tole, 2023].

- Ant Colony Optimization (ACO)

ACO, karınca kolonilerinin yiyecek arama davranışlarından ilham alan bir meta-sezgisel algoritmadır. Algoritma, yiyecek arayan karıncaların davranışlarını simüle ederek çalışır. Her karınca bir çözüm adayı olarak kabul edilir ve diğer karıncalar tarafından bırakılan feromon izlerine dayalı olarak kararlar alarak problem uzayını kat eder. Feromon izleri bir çözümün kalitesini temsil eder, daha güçlü izler daha iyi çözümlere işaret eder. Karıncalar problem uzayında hareket ederken buldukları çözümlerin üzerine feromonlar bırakırlar ve bu da diğer karıncaların davranışlarını etkiler. Zamanla, feromon izleri en uygun çözümde birleşir [Haouari, 2009].

- Particle Swarm Optimization (PSO)

PSO, kuş sürülerinin davranışlarından ilham alan bir meta-sezgisel algoritmadır. Kutu paketleme problemi de dahil olmak üzere optimizasyon problemlerini çözmek için kullanılır. PSO'da, parçacık adı verilen bir dizi aday çözüm, kendi en iyi çözümlerinin ve şimdiye kadar bulunan küresel en iyi çözümün rehberliğinde çözüm uzayında hareket eder. Parçacıkların hareketi, bir dizi kurala göre her yinelemede güncellenen bir dizi hız ve konum parametresi tarafından kontrol edilir. Dezavantajlarından biri, algoritmanın başlangıç koşullarına ve parametre ayarına duyarlı olmasıdır [Haouari, 2009].

- Tabu Search Algorithm (TSA)

TSA bir meta-sezgisel optimizasyon algoritmasıdır. Algoritma, yakın zamanda elde edilen çözümleri takip etmek için "tabu listesi" adı verilen bir bellek yapısı kullanır ve bu bilgiyi aramayı daha iyi çözümlere yönlendirmek için kullanır. Algoritma bir ilk çözümle başlar ve tabu listesindeki çözümlerden kaçınarak yinelemeli olarak çözümde küçük değişiklikler yapar [Delorme, 2016]. Keşif ve sömürünün bir kombinasyonunu kullanan algoritma, nispeten kısa bir süre içinde iyi veya optimuma yakın bir çözüm bulmayı amaçlar. Algoritma, eldeki kutu paketleme probleminin belirli özelliklerine uyarlanabilir. Diğer yöntemler daha popüler ve verimli olduğundan tabu aramanın kutu paketleme problemlerini çözmek için en çok kullanılan yöntemlerden biri değildir.

3. MATEMATİKSEL YÖNTEMLER

Tezin bu bölümünde TBKPP, İBKPP ve ÜBKPP için KTP ve KP modelleri ve parametreleri her bir problem türü için ayrı ayrı olacak şekilde sunulmuştur.

3.1. Matematiksel Yöntemlerde Kullanılan Notasyonlar

Tablo 3.1: Matematiksel yöntemlerde kullanılan notasyonlar ve açıklamaları.

Notasyon	Açıklama
x_i, y_i, z_i	Paketlenen kutu konumları karar değişkenleri.
l_i, w_i, h_i	Paketlenen kutuları sırasıyla boy, en ve yükseklik boyutları.
L, W, H	Paketleme yapılan alanın sırasıyla boy, en ve yükseklik boyutları.
n	Paketlenecek kutu sayısı.
m	Paketleme yapılabilecek kutu sayısı.
M	Çok büyük bir sayı.
XT_i	Paketlenme ikili değişkeni eğer $XT_i = 1$ ise *i* kutusu paketlenmiştir.
XT_{ik}	Paketlenme ikili değişkeni eğer $XT_{ik} = 1$ ise *i* kutusu *k alanına* paketlenmiştir.
YT_k	Paketlenme ikili değişkeni eğer $YT_k = 1$ ise kutu k alanına paketlenmiştir.
$a_{ij}, b_{ij}, c_{ij}, d_{ij}, e_{ij}, f_{ij}$	Paketlenen *i* kutusunun *j* kutusuna göre yönünü ifade etmektedir. $a_{ij} = 1$ ise i kutusu j kutusunun solunda konumlandırılmıştır. Sırasıyla $b_{ij}, c_{ij}, d_{ij}, e_{ij}, f_{ij}$ notasyonları; sağ, ön, arka, yukarı, aşağı yönleri ifade etmektedir.

3.2. Ortogonal Bin Packing Problemi

3.2.1. TBDPP için Matematiksel Yöntemler

TBDPP tek boyutta paketleme işlemlerinin yapıldığı problem türüdür. Problemin çözümü için gereken parametreler:

- Değişkenler: Matematiksel modelin çalışması için gereken değerlerdir. Karar değişkenleri, kesikli değişkenler ve sürekli değişkenler TBDPP için kullanılabilir.

- Kısıtlar: TBDPP için bu çalışmada kullanılan kısıt paketlemenin yapılacağı alanın boyutudur. Diğer bir değişle kutunun kapasitesidir. Diğer bir kısıt ise paketlenecek öğelerin birbiri ile çakışmama kısıtıdır.
- Amaç Fonksiyonu: TBDPP için bu çalışmada KTP modellerinde paketlenen kutu sayısının maksimizasyonu olarak alınmıştır.
- Çözücü: TBDPP için IBM ILOG CPLEX ve KP çözücüleri kullanılmıştır.

3.2.1.1. TBDPP için KP Modeli

$Subject\ to$

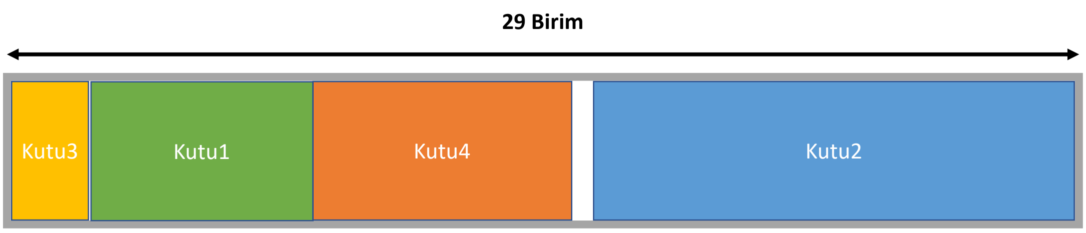
$$
\begin{aligned}
|x_j - x_i| &\geq l_j, \qquad \forall\, i, j, i < j, \\
x_i &\geq l_i, \forall\, i\,, \\
x_i &\leq L, \forall\, i, \\
x_i &\geq 0
\end{aligned}
\tag{3.1}
$$

İlk ve ikinci kısıtlar paketlenen kutuların çakışmamasını sağlayan kısıtlardır. Üçüncü kısıt ise paketleme işleminin yapıldığı alanın boyut yani kapasite kısıtıdır. Son kısıt ise kutuların paketleneceği konumların negatif olmamasını sağlayan kısıttır.

- KP ile Örnek TBDPP Çözümü

Yukarıda belirtilen modelin uzunlukları sırasıyla 6, 13, 2, 7 birim olan kutuların uzunluğu 29 birim olan yükleme alanına paketlenmesi probleminin KP çözümü aşağıdaki gibidir.

Çözüm sonucunda kutuların yükleme alanındaki konumları sırasıyla 8, 29, 2, 15 değerleridir.



Şekil 3.1: TBDPP KP çözümü.

3.2.1.2. TBDPP için KTP Modeli

$$Minimize$$

$$\sum_{i=1}^{n} XT_i$$

$$Subject\ to$$

$$
\begin{aligned}
x_i + l_i \ &\geq \ x_j + (1 - a_{ij}).M, \qquad \forall\ i,j,i < j, \\
x_j + l_j \ &\geq \ x_i + (1 - b_{ij}).M, \qquad \forall\ i,j,i < j, \\
a_{ij} + \ b_{ij} \ &\geq \ XT_i, \qquad \forall i,j,i < j, \\
x_i \ &\leq \ L + (1 - XT_i).M, \qquad \forall i,j, \\
&a_{ij}, b_{ij}, XT_i, binary \\
&x_i \geq 0
\end{aligned}
\tag{3.2}
$$

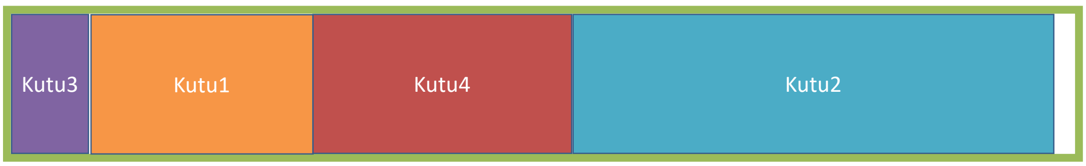
İlk ve ikinci kısıtlar paketlenen kutuların çakışmamasını sağlayan kısıtlardır. Üçüncü kısıt paketlemelerin birbirleriyle olan yön tayinlerinin yapıldığı kısıttır. Dördüncü kısıt ise paketleme işleminin yapıldığı alanın boyut yani kapasite kısıtıdır. Beşinci kısıt ise ikili değişkenleri ifade etmektedir. Son kısıt ise kutuların paketleneceği konumların negatif olmamasını sağlayan kısıttır. Amaç fonksiyonu ise belirlenen paketleme alanına en fazla kutunun paketlenmesini amaçlamaktadır.

• KTP ile Örnek TBDPP Çözümü

Yukarıda belirtilen modelin uzunlukları sırasıyla 6, 13, 2, 7 birim olan kutuların uzunluğu 29 birim olan yükleme alanına paketlenmesi probleminin KTP çözümü aşağıdaki gibidir.

Çözüm sonucunda kutuların yükleme alanındaki konumları sırasıyla 8, 28, 2, 15 değerleridir.

29 Birim



Şekil 3.2: TBDPP KTP çözümü.

3.2.2. İBDPP için Matematiksel Yöntemler

İBDPP iki boyutta paketleme işlemlerinin yapıldığı problem türüdür. Problemin çözümü için gereken parametreler:

- Değişkenler: Matematiksel modelin çalışması için gereken değerlerdir. Karar değişkenleri, kesikli değişkenler ve sürekli değişkenler İBDPP için kullanılabilir.
- Kısıtlar: İBDPP için bu çalışmada kullanılan kısıt paketlemenin yapılacağı alanın boy ve enidir. Diğer bir değişle kutunun kapasitesidir. Diğer bir kısıt ise paketlenecek öğelerin birbiri ile çakışmama kısıtıdır.
- Amaç Fonksiyonu: İBDPP için bu çalışmada KTP modellerinde paketlenen kutu sayısının maksimizasyonu olarak alınmıştır.
- Çözücü: İBDPP için IBM ILOG CPLEX ve KP çözücüleri kullanılmıştır.

3.2.2.1. İBDPP için KP Modeli

$Subject\ to$

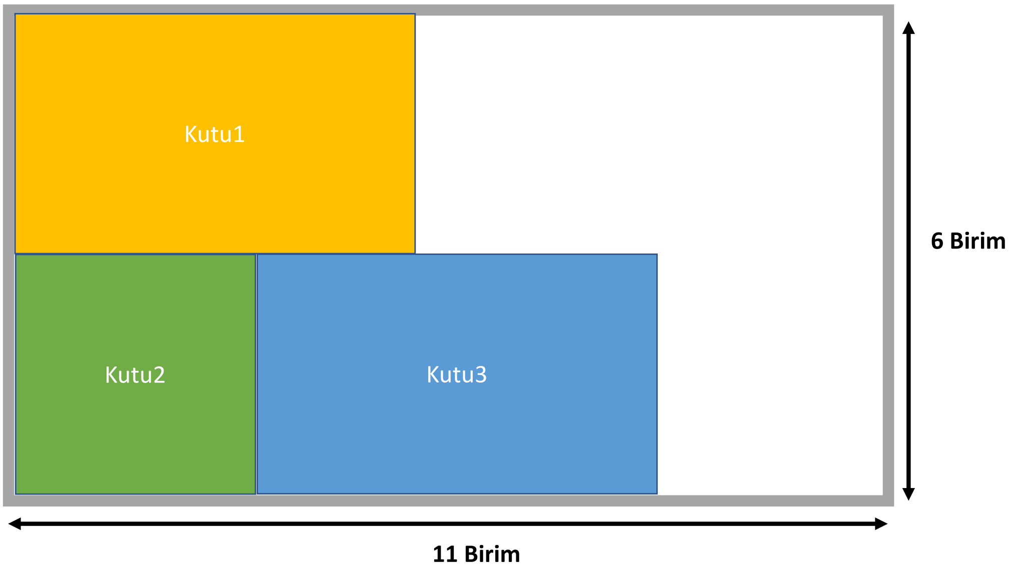
$$
\begin{aligned}
|x_j - x_i| &\geq l_j, \quad \forall\, i, j, i < j,\\
|y_j - y_i| &\geq w_j, \quad \forall\, i, j, i < j,\\
x_i &\geq l_i, \forall i\,,\\
y_i &\geq w_i, \forall i\,,\\
x_i &\leq L, \forall\, i,\\
y_i &\leq W, \forall\, i,\\
x_i, y_i &\geq 0
\end{aligned}
\tag{3.3}
$$

İlk dört kısıt paketlenen kutuların çakışmamasını sağlayan kısıtlardır. Beşinci ve Altıncı kısıt ise paketleme işleminin yapıldığı alanın boyut yani kapasite kısıtıdır. Son kısıt ise kutuların paketleneceği konumların negatif olmamasını sağlayan kısıttır.

- KP ile Örnek İBDPP Çözümü

Yukarıda belirtilen modelin boyutları ([boy, en]) sırasıyla [5, 3], [3, 3], [5, 3] birim olan kutuların uzunluğu [11,6] birim olan yükleme alanına paketlenmesi probleminin KP çözümü aşağıdaki gibidir.

Çözüm sonucunda kutuların yükleme alanındaki konumları sırasıyla [5, 6], [3, 3], [8,3] değerleridir.



Şekil 3.3 İBDPP KP çözümü.

3.2.2.2. İBDPP için KTP Modeli

Maksimize

$$\sum_{i=1}^{n} XT_i$$

Subject to

$$\begin{aligned}
x_i + l_i &\geq x_j + (1 - a_{ij}).M, && \forall\, i, j, i < j, \\
x_j + l_j &\geq x_i + (1 - b_{ij}).M, && \forall\, i, j, i < j, \\
y_i + w_i &\geq y_j + (1 - c_{ij}).M, && \forall\, i, j, i < j, \\
y_j + w_j &\geq y_i + (1 - d_{ij}).M, && \forall\, i, j, i < j, \\
a_{ij} + b_{ij} + c_{ij} + d_{ij} &\geq XT_i, && \forall i, j, i < j, \\
x_i &\leq L + (1 - XT_i).M, && \forall i, j, \\
y_i &\leq W + (1 - XT_i).M, && \forall i, j, \\
&a_{ij}, b_{ij}, c_{ij}, d_{ij}, XT_i, binary \\
&x_i, y_i \geq 0
\end{aligned} \tag{3.4}$$

İlk dört kısıt paketlenen kutuların çakışmamasını sağlayan kısıtlardır. Beşinci kısıt paketlemelerin birbirleriyle olan yön tayinlerinin yapıldığı kısıttır. Altıncı ve yedinci kısıt ise paketleme işleminin yapıldığı alanın boyut yani kapasite kısıtıdır. Sekizinci kısıt ise ikili değişkenleri ifade etmektedir. Son kısıt ise kutuların paketleneceği konumların negatif olmamasını sağlayan kısıttır. Amaç fonksiyonu ise belirlenen paketleme alanına en fazla kutunun paketlenmesini amaçlamaktadır.

• KTP ile Örnek İBDPP Çözümü

Yukarıda belirtilen modelin boyutları ([boy, en]) sırasıyla [5, 3], [3, 3], [5, 3] birim olan kutuların uzunluğu [11,6] birim olan yükleme alanına paketlenmesi probleminin KTP çözümü aşağıdaki gibidir.

Çözüm sonucunda kutuların yükleme alanındaki konumları sırasıyla [5, 6], [10,3], [5, 3] değerleridir.

Şekil 3.4 İBDPP KTP çözümü.

3.2.3. ÜBDPP için Matematiksel Yöntemler

ÜBDPP üç boyutta paketleme işlemlerinin yapıldığı problem türüdür. Problemin çözümü için gereken parametreler:

- Değişkenler: Matematiksel modelin çalışması için gereken değerlerdir. Karar değişkenleri, kesikli değişkenler ve sürekli değişkenler ÜBDPP için kullanılabilir.
- Kısıtlar: ÜBDPP için bu çalışmada kullanılan kısıt paketlemenin yapılacağı alanın boy, en ve yüksekliktir. Diğer bir değişle kutunun kapasitesidir. Diğer bir kısıt ise paketlenecek öğelerin birbiri ile çakışmama kısıtıdır.
- Amaç Fonksiyonu: ÜBDPP için bu çalışmada KTP modellerinde paketlenen kutu sayısının maksimizasyonu olarak alınmıştır.
- Çözücü: İBDPP için IBM ILOG CPLEX ve KP çözücüleri kullanılmıştır.

3.2.3.1. ÜBDPP için KP Modeli

$Subject\ to$

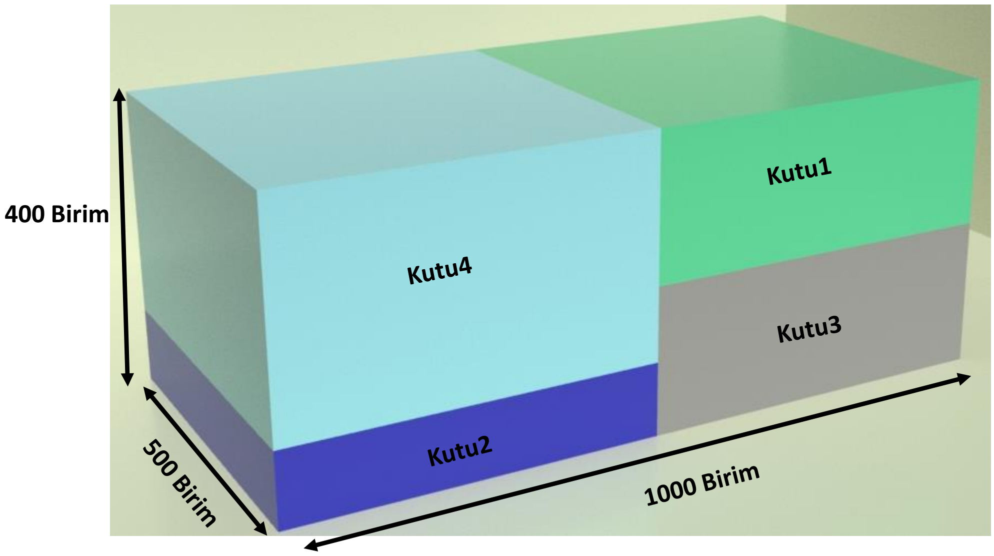
$$
\begin{aligned}
|x_j - x_i| &\geq l_j, \quad \forall\, i,j, i < j,\\
|y_j - y_i| &\geq w_j, \quad \forall\, i,j, i < j,\\
|z_j - z_i| &\geq h_j, \quad \forall\, i,j, i < j,\\
x_i &\geq l_i, \forall i\,,\\
y_i &\geq w_i, \forall i\,,\\
z_i &\geq h_i, \forall i\,,\\
x_i &\leq L, \forall\, i,\\
y_i &\leq W, \forall\, i,\\
z_i &\leq H, \forall\, i,\\
x_i, y_i, z_i &\geq 0
\end{aligned}
\tag{3.5}
$$

İlk altı kısıt paketlenen kutuların çakışmamasını sağlayan kısıtlardır. Yedinci, sekizinci ve dokuzuncu kısıt ise paketleme işleminin yapıldığı alanın boyut yani kapasite kısıtıdır. Son kısıt ise kutuların paketleneceği konumların negatif olmamasını sağlayan kısıttır.

- KP ile Örnek ÜBDPP Çözümü

Yukarıda belirtilen modelin boyutları ([boy, en,yükseklik]) sırasıyla [500, 500, 200], [500, 500, 100], [500, 500, 200], [500, 500, 300] birim olan kutuların uzunluğu [1000, 500, 400] birim olan yükleme alanına paketlenmesi probleminin CP çözümü aşağıdaki gibidir.

Çözüm sonucunda kutuların yükleme alanındaki konumları sırasıyla [1000, 500, 400], [500, 500, 100], [1000, 500, 200], [500, 500, 400] değerleridir.



Şekil 3.5 ÜBDPP KP çözüm.

3.2.3.2. ÜBDPP için KTP Modeli

$$Maksimize$$

$$\sum_{i=1}^{n} XT_i$$

$$Subject\ to$$

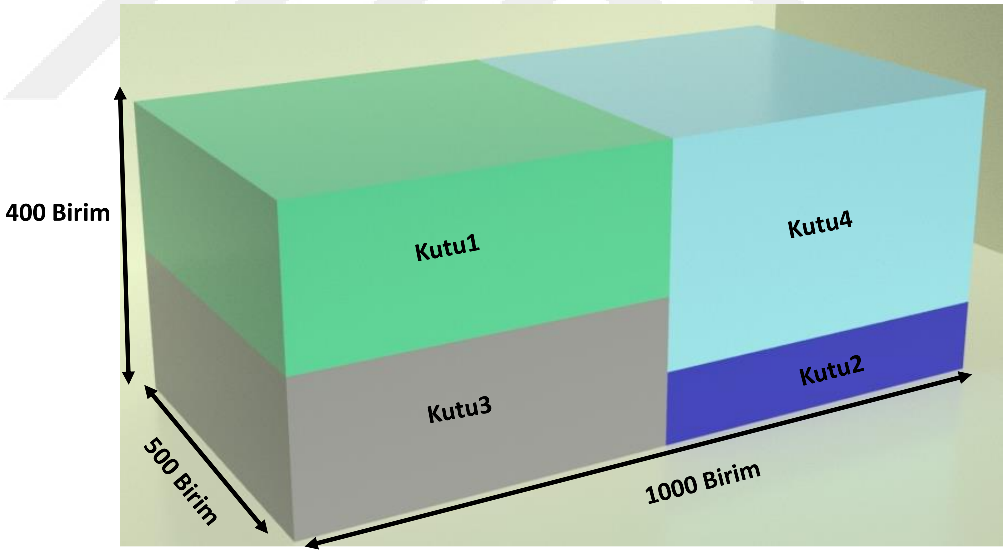
$$\begin{aligned}
x_i + l_i &\geq x_j + (1 - a_{ij}).M, && \forall\, i,j,i<j, \\
x_j + l_j &\geq x_i + (1 - b_{ij}).M, && \forall\, i,j,i<j, \\
y_i + w_i &\geq y_j + (1 - c_{ij}).M, && \forall\, i,j,i<j, \\
y_j + w_j &\geq y_i + (1 - d_{ij}).M, && \forall\, i,j,i<j, \\
z_i + h_i &\geq z_j + (1 - e_{ij}).M, && \forall\, i,j,i<j, \\
z_j + h_j &\geq z_i + (1 - f_{ij}).M, && \forall\, i,j,i<j, \\
a_{ij} + b_{ij} + c_{ij} + d_{ij} + e_{ij} + f_{ij} &\geq XT_i, && \forall i,j,i<j, \\
x_i &\leq L + (1 - XT_i).M, && \forall i,j, \\
y_i &\leq W + (1 - XT_i).M, && \forall i,j, \\
z_i &\leq H + (1 - XT_i).M, && \forall i,j, \\
&a_{ij}, b_{ij}, c_{ij}, d_{ij}, e_{ij}, f_{ij}, XT_i, binary \\
&x_i, y_i, z_i \geq 0
\end{aligned} \tag{3.6}$$

İlk altı kısıt paketlenen kutuların çakışmamasını sağlayan kısıtlardır. Yedinci kısıt paketlemelerin birbirleriyle olan yön tayinlerinin yapıldığı kısıttır. Sekizinci, dokuzuncu ve onuncu kısıt ise paketleme işleminin yapıldığı alanın boyut yani kapasite kısıtıdır. On birinci kısıt ise ikili değişkenleri ifade etmektedir. Son kısıt ise kutuların paketleneceği konumların negatif olmamasını sağlayan kısıttır. Amaç fonksiyonu ise belirlenen paketleme alanına en fazla kutunun paketlenmesini amaçlamaktadır.

- KTP ile Örnek ÜBDPP Çözümü

Yukarıda belirtilen modelin boyutları ([boy, en,yükseklik]) sırasıyla [500, 500, 200], [500, 500, 100], [500, 500, 200], [500, 500, 300] birim olan kutuların uzunluğu [1000, 500, 400] birim olan yükleme alanına paketlenmesi probleminin KTP çözümü aşağıdaki gibidir.

Çözüm sonucunda kutuların yükleme alanındaki konumları sırasıyla [500, 500, 400], [1000, 500, 100], [500, 500, 200], [1000, 500, 400] değerleridir.



Şekil 3.6 ÜBDPP KTP çözümü.

3.3. Bin Packing Problemi

3.3.1. TBKPP için Matematiksel Yöntemler

TBKPP tek boyutta paketleme işlemlerinin yapıldığı problem türüdür. Problemin çözümü için gereken parametreler:

- Değişkenler: Matematiksel modelin çalışması için gereken değerlerdir. Karar değişkenleri, kesikli değişkenler ve sürekli değişkenler TBKPP için kullanılabilir.
- Kısıtlar: TBKPP için bu çalışmada kullanılan kısıt paketlemenin yapılacağı alanın boyutu ve bu alandan kaç tane kullanılacağıdır. Diğer bir kısıt ise paketlenecek öğelerin birbiri ile çakışmama kısıtıdır.
- Amaç Fonksiyonu: TBKPP için bu çalışmada KP ve KTP modellerinde paketleme yapılacak alanın sayısının minimizasyonu olarak alınmıştır.
- Çözücü: TBKPP için IBM ILOG CPLEX ve KP çözücüleri kullanılmıştır.

3.3.1.1. TBKPP için KP Modeli

$Minimize$

$$\sum_{k=1}^{m} YT_k$$

$Subject\ to$

$$\begin{gathered} if\ (XT_{ik} + XT_{jk} = 2)\ then(\,|x_j - x_i| \ \geq\ l_j), \qquad \forall\, i, j, i < j, \\ x_i \ \geq\ l_i, \forall i\,, \\ \sum_{k=1}^{m} XT_{ik} = 1, \forall\, i, \\ x_i \leq\ L + (1 - XT_{ik}).M, \qquad \forall i, k \\ \sum_{i=1}^{n} XT_{ik} \leq M.YT_k, \forall\, j, \\ YT_k \ \geq\ YT_{k+1}, 1 \leq j < m \\ XT_{ik}, YT_k, binary \end{gathered} \tag{3.7}$$

$$x_i \geq 0$$

İlk kısıt eğer yüklenen öğeler aynı kutuya yerleştirilirse birbirleriyle çakışmama kısıtı ve ikinci kısıtta yine kutuların çakışmamasını sağlayan kısıttır. Üçüncü kısıt ise her bir öğenin bir kutuya yerleştirilmesini sağlayan kısıttır. Dördüncü kısıt ise paketleme işleminin yapıldığı alanın boyut yani kapasite kısıtıdır. Beşinci kısıt ise paketlenen öğeler ile paketleme alanı arasındaki bağlantının kurulduğu kısıttır bu kısıt sayesinde herhangi bir öğe bir kutuya paketlendiğinde o kutunun kullanıldığı modele bildirilmektedir. Altıncı kısıt çözüm uzayının büyümemesi için ilk indisli kutuların kullanılmasını sağlayan kısıttır. Yedinci kısıt ise kısıtta belirtilen değişkenlerin ikili değişken olduğunu ifade etmektedir. Son kısıt ise kutuların paketleneceği konumların negatif olmamasını sağlayan kısıttır.

- KP ile Örnek TBKPP Çözümü

Yukarıda belirtilen modelin uzunlukları sırasıyla 6, 13, 2, 7 birim olan kutuların uzunluğu 20 birim olan yükleme alanlarına paketlenmesi probleminin KP çözümü aşağıdaki gibidir.

Çözüm sonucunda 2 kutu paketleme için kullanılmıştır.

3.3.1.2. TBKPP için KTP Modeli

$Minimize$

$$\sum_{k=1}^{m} YT_k$$

$Subject\ to$

$$x_i + l_i \geq x_j + (1 - a_{ij}).M, \qquad \forall\, i, j, i < j,$$

$$x_j + l_j \geq x_i + (1 - b_{ij}).M, \qquad \forall\, i, j, i < j, \qquad (3.8)$$

$$a_{ij} + b_{ij} \geq XT_{ik} + XT_{jk} - 1, \qquad \forall i, j, k, i < j,$$

$$\sum_{k=1}^{m} XT_k = 1, \forall\, i,$$

$$x_i \leq L_k + (1 - XT_{ik}).M, \qquad \forall i, k$$

$$\sum_{i=1}^{n} XT_{ik} \leq M.YT_k, \forall j,$$

$$a_{ij}, b_{ij}, XT_{ik}, YT_k, binary$$

$$x_i \geq 0$$

İlk ve ikinci kısıtlar paketlenen kutuların çakışmamasını sağlayan kısıtlardır. Üçüncü kısıt paketlemelerin birbirleriyle olan yön tayinlerinin yapıldığı kısıttır. Dördüncü kısıt her bir öğenin bir kutuya yerleştirilmesini sağlayan kısıttır. Beşinci kısıt ise paketleme işleminin yapıldığı alanın boyut yani kapasite kısıtıdır. Altıncı kısıt paketlenen öğeler ile paketleme alanı arasındaki bağlantının kurulduğu kısıttır bu kısıt sayesinde herhangi bir öğe bir kutuya paketlendiğinde o kutunun kullanıldığı modele bildirilmektedir. Yedinci kısıt ise ikili değişkenleri ifade etmektedir. Son kısıt ise kutuların paketleneceği konumların negatif olmamasını sağlayan kısıttır. Amaç fonksiyonu ise girdi olarak verilen öğelerin paketlenmesi için kullanılan kutu sayısının minimizasyonudur.

• KTP ile Örnek TBKPP Çözümü

Yukarıda belirtilen modelin uzunlukları sırasıyla 325, 311, 254, 198, 198, 201, 316, 254 birim olan kutuların uzunluğu 1600 birim olan yükleme alanlarına paketlenmesi probleminin KTP çözümü aşağıdaki gibidir.

Çözüm sonucunda 2 kutu paketleme için kullanılmıştır.

3.3.2. İBKPP için Matematiksel Yöntemler

TBKPP iki boyutta paketleme işlemlerinin yapıldığı problem türüdür. Problemin çözümü için gereken parametreler:

- Değişkenler: Matematiksel modelin çalışması için gereken değerlerdir. Karar değişkenleri, kesikli değişkenler ve sürekli değişkenler İBKPP için kullanılabilir.
- Kısıtlar: İBKPP için bu çalışmada kullanılan kısıt paketlemenin yapılacağı alanın boyutu ve bu alandan kaç tane kullanılacağıdır. Diğer bir kısıt ise paketlenecek öğelerin birbiri ile çakışmama kısıtıdır.

- Amaç Fonksiyonu: TBKPP için bu çalışmada KP ve KTP modellerinde paketleme yapılacak alanın sayısının minimizasyonu olarak alınmıştır.
- Çözücü: TBKPP için IBM ILOG CPLEX ve KP çözücüleri kullanılmıştır.

3.3.2.1. İBKPP için KP Modeli

$Minimize$

$$\sum_{k=1}^{m} YT_k$$

$Subject\ to$

$$\begin{gathered}
if\ (XT_{ik} + XT_{jk} = 2)\ then(\ |x_j - x_i| \ \geq\ l_j), \qquad \forall\, i, j, i < j, \\
if\ (XT_{ik} + XT_{jk} = 2)\ then(\ |y_j - y_i| \ \geq\ w_j), \qquad \forall\, i, j, i < j, \\
x_i \ \geq\ l_i, \forall i\ , \\
y_i \ \geq\ w_i, \forall i\ , \\
\sum_{k=1}^{m} XT_{ik} = 1, \forall\ i, \\
x_i \leq\ L + (1 - XT_{ik}).M, \qquad \forall i, k \\
y_i \leq\ W + (1 - XT_{ik}).M, \qquad \forall i, k \\
\sum_{i=1}^{n} XT_{ik} \leq M.YT_k, \forall\ j, \\
YT_k \ \geq\ YT_{k+1}, 1 \leq j < m\ , \\
XT_{ik}, YT_k, binary \\
x_i, y_i \geq 0
\end{gathered} \tag{3.9}$$

İlk iki kısıt eğer yüklenen öğeler aynı kutuya yerleştirilirse birbirleriyle çakışmama kısıtı ve sonraki iki kısıtta yine kutuların çakışmamasını sağlayan kısıttır. Beşinci kısıt ise her bir öğenin bir kutuya yerleştirilmesini sağlayan kısıttır. Sonraki iki kısıt ise paketleme işleminin yapıldığı alanın boyut yani kapasite kısıtıdır. Sekizinci kısıt ise paketlenen öğeler ile paketleme alanı arasındaki bağlantının kurulduğu kısıttır bu kısıt sayesinde herhangi bir öğe bir kutuya paketlendiğinde o kutunun kullanıldığı modele bildirilmektedir. Dokuzuncu kısıt çözüm uzayının büyümemesi için ilk indisli kutuların kullanılmasını sağlayan kısıttır. Onuncu kısıt ise kısıtta belirtilen değişkenlerin ikili değişken olduğunu ifade etmektedir. Son kısıt ise kutuların paketleneceği konumların negatif olmamasını sağlayan kısıttır.

• KP ile Örnek İBKPP Çözümü

Yukarıda belirtilen modelin uzunlukları sırasıyla boy; [4, 4, 5, 3, 7, 1, 6] birim ve en; [4, 3, 4, 3, 2, 4, 2] birim olan kutular uzunluğu ([boy,en]) [8,5] birim olan yükleme alanlarına paketlenmesi probleminin KP çözümü aşağıdaki gibidir.

Çözüm sonucunda 3 kutu paketleme için kullanılmıştır.

3.3.2.2. İBKPP için KTP Modeli

$$
\begin{aligned}
&Minimize \\
&\qquad \sum_{k=1}^{m} YT_k \\
&Subject\ to \\
&x_i + l_i \geq x_j + (1 - a_{ij}).M, \qquad \forall\, i,j,i<j, \\
&x_j + l_j \geq x_i + (1 - b_{ij}).M, \qquad \forall\, i,j,i<j, \\
&y_i + w_i \geq y_j + (1 - c_{ij}).M, \qquad \forall\, i,j,i<j, \\
&y_j + w_j \geq y_i + (1 - d_{ij}).M, \qquad \forall\, i,j,i<j, \\
&a_{ij} + b_{ij} + c_{ij} + d_{ij} \geq XT_{ik} + XT_{jk} - 1, \qquad \forall i,j,k,i<j, \\
&\sum_{k=1}^{m} XT_k = 1, \forall\, i, \\
&x_i \leq L_k + (1 - XT_{ik}).M, \qquad \forall i,k \\
&y_i \leq W_k + (1 - XT_{ik}).M, \qquad \forall i,k \\
&\sum_{i=1}^{n} XT_{ik} \leq M.YT_k, \forall\, j, \\
&a_{ij}, b_{ij}, c_{ij}, d_{ij}, XT_{ik}, YT_k, binary \\
&x_i, y_i \geq 0
\end{aligned}
\tag{3.10}
$$

İlk dört kısıt paketlenen kutuların çakışmamasını sağlayan kısıtlardır. Beşinci kısıt paketlemelerin birbirleriyle olan yön tayinlerinin yapıldığı kısıttır. Altıncı kısıt her bir öğenin bir kutuya yerleştirilmesini sağlayan kısıttır. Sonraki iki kısıt ise paketleme işleminin yapıldığı alanın boyut yani kapasite kısıtıdır. Dokuzuncu kısıt paketlenen öğeler ile paketleme alanı arasındaki bağlantının kurulduğu kısıttır bu kısıt sayesinde herhangi bir öğe bir kutuya paketlendiğinde o kutunun kullanıldığı modele bildirilmektedir. Onuncu kısıt ise ikili değişkenleri ifade etmektedir. Son kısıt ise kutuların paketleneceği konumların negatif olmamasını sağlayan kısıttır. Amaç fonksiyonu ise girdi olarak verilen öğelerin paketlenmesi için kullanılan kutu sayısının minimizasyonudur.

- KTP ile Örnek İBKPP Çözümü

Yukarıda belirtilen modelin uzunlukları sırasıyla boy; [1600, 1600, 1600, 1600] birim ve en; [480, 480, 480, 480] birim olan kutular uzunluğu ([boy,en]) [1600, 480] birim olan yükleme alanlarına paketlenmesi probleminin KTP çözümü aşağıdaki gibidir.

Çözüm sonucunda 2 kutu paketleme için kullanılmıştır.

3.3.3. ÜBKPP için Matematiksel Yöntemler

ÜBKPP üç boyutta paketleme işlemlerinin yapıldığı problem türüdür. Problemin çözümü için gereken parametreler:

- Değişkenler: Matematiksel modelin çalışması için gereken değerlerdir. Karar değişkenleri, kesikli değişkenler ve sürekli değişkenler ÜBKPP için kullanılabilir.
- Kısıtlar: ÜBKPP için bu çalışmada kullanılan kısıt paketlemenin yapılacağı alanın boyutu ve bu alandan kaç tane kullanılacağıdır. Diğer bir kısıt ise paketlenecek öğelerin birbiri ile çakışmama kısıtıdır.
- Amaç Fonksiyonu: ÜBKPP için bu çalışmada KP ve KTP modellerinde paketleme yapılacak alanın sayısının minimizasyonu olarak alınmıştır.
- Çözücü: ÜBKPP için IBM ILOG CPLEX ve KP çözücüleri kullanılmıştır.

3.3.3.1. ÜBKPP için KP Model

$$
\begin{aligned}
&Minimize \\
&\sum_{k=1}^{m} YT_k \\
&Subject\ to \\
&if\ (XT_{ik} + XT_{jk} = 2)\ then(\ |x_j - x_i| \geq l_j), \quad \forall\, i, j, i < j, \\
&if\ (XT_{ik} + XT_{jk} = 2)\ then(\ |y_j - y_i| \geq w_j), \quad \forall\, i, j, i < j, \\
&if\ (XT_{ik} + XT_{jk} = 2)\ then(\ |z_j - z_i| \geq h_j), \quad \forall\, i, j, i < j, \\
&x_i \geq l_i, \forall i, \\
&y_i \geq w_i, \forall i, \\
&z_i \geq h_i, \forall i, \\
&\sum_{k=1}^{m} XT_{ik} = 1, \forall\, i, \\
&x_i \leq L + (1 - XT_{ik}).M, \quad \forall i, k \\
&y_i \leq W + (1 - XT_{ik}).M, \quad \forall i, k \\
&z_i \leq H + (1 - XT_{ik}).M, \quad \forall i, k \\
&\sum_{i=1}^{n} XT_{ik} \leq M.YT_k, \forall\, j, \\
&XT_{ik}, YT_k, binary \\
&x_i, y_i, z_i \geq 0
\end{aligned} \tag{3.11}
$$

İlk üç kısıt eğer yüklenen öğeler aynı kutuya yerleştirilirse birbirleriyle çakışmama kısıtı ve sonraki üç kısıtta yine kutuların çakışmamasını sağlayan kısıttır. Yedinci kısıt ise her bir öğenin bir kutuya yerleştirilmesini sağlayan kısıttır. Sonraki üç kısıt ise paketleme işleminin yapıldığı alanın boyut yani kapasite kısıtıdır. On birinci kısıt ise paketlenen öğeler ile paketleme alanı arasındaki bağlantının kurulduğu kısıttır bu kısıt sayesinde herhangi bir öğe bir kutuya paketlendiğinde o kutunun kullanıldığı modele bildirilmektedir. On ikinci kısıt çözüm uzayının büyümemesi için ilk indisli kutuların kullanılmasını sağlayan kısıttır. On üçüncü kısıt ise kısıtta belirtilen değişkenlerin ikili değişken olduğunu ifade etmektedir. Son kısıt ise kutuların paketleneceği konumların negatif olmamasını sağlayan kısıttır.

• KP ile Örnek ÜBKPP Çözümü

Yukarıda belirtilen modelin uzunlukları sırasıyla boy; [500, 500, 500, 500, 200] birim, en; [500, 500, 500, 500, 300] ve yükseklik; [200, 100, 200, 300, 400] birim olan kutular uzunluğu ([boy, en, yükseklik]) [1000, 500, 400] birim olan yükleme alanlarına paketlenmesi probleminin KP çözümü aşağıdaki gibidir.

Çözüm sonucunda 2 kutu paketleme için kullanılmıştır.

3.3.3.2. ÜBKPP için KTP Model

$$Minimize$$

$$\sum_{k=1}^{m} YT_k$$

$$Subject\ to$$

$$
\begin{aligned}
x_i + l_i \ &\geq \ x_j + \left(1 - a_{ij}\right).M, \qquad \forall\, i, j, i < j,\\
x_j + l_j \ &\geq \ x_i + \left(1 - b_{ij}\right).M, \qquad \forall\, i, j, i < j,\\
y_i + w_i \ &\geq \ y_j + \left(1 - c_{ij}\right).M, \qquad \forall\, i, j, i < j,\\
y_j + w_j \ &\geq \ y_i + \left(1 - d_{ij}\right).M, \qquad \forall\, i, j, i < j,\\
z_i + h_i \ &\geq \ z_j + \left(1 - e_{ij}\right).M, \qquad \forall\, i, j, i < j,\\
z_j + h_j \ &\geq \ z_i + \left(1 - f_{ij}\right).M, \qquad \forall\, i, j, i < j,
\end{aligned}
$$

$$a_{ij} + \ b_{ij} + \ c_{ij} + \ d_{ij} + \ e_{ij} + \ f_{ij} \ \geq \ XT_{ik} + \ XT_{jk} - 1, \qquad (3.12)$$

$$\forall i, j, k, i < j,$$

$$\sum_{k=1}^{m} XT_k = 1, \forall\, i,$$

$$
\begin{aligned}
x_i &\leq \ L_k + (1 - XT_{ik}).M, \qquad \forall i, k\\
y_i &\leq \ W_k + (1 - XT_{ik}).M, \qquad \forall i, k\\
z_i &\leq \ H_k + (1 - XT_{ik}).M, \qquad \forall i, k
\end{aligned}
$$

$$\sum_{i=1}^{n} XT_{ik} \leq M.YT_k, \forall\, j,$$

$$a_{ij}, b_{ij}, c_{ij}, d_{ij}, e_{ij}, f_{ij}, XT_{ik}, YT_k, binary$$

$$x_i, y_i, z_i \geq 0$$

İlk altı kısıt paketlenen kutuların çakışmamasını sağlayan kısıtlardır. Yedinci kısıt paketlemelerin birbirleriyle olan yön tayinlerinin yapıldığı kısıttır. Sekizinci kısıt her bir öğenin bir kutuya yerleştirilmesini sağlayan kısıttır. Sonraki üç kısıt ise paketleme işleminin yapıldığı alanın boyut yani kapasite kısıtıdır. On ikinci kısıt paketlenen öğeler ile paketleme alanı arasındaki bağlantının kurulduğu kısıttır bu kısıt sayesinde herhangi bir öğe bir kutuya paketlendiğinde o kutunun kullanıldığı modele bildirilmektedir. On üçüncü kısıt ise ikili değişkenleri ifade etmektedir. Son kısıt ise kutuların paketleneceği konumların negatif olmamasını sağlayan kısıttır. Amaç fonksiyonu ise girdi olarak verilen öğelerin paketlenmesi için kullanılan kutu sayısının minimizasyonudur.

• KTP ile Örnek İBKPP Çözümü

Yukarıda belirtilen modelin uzunlukları sırasıyla boy; [325, 311, 254, 198, 198, 201, 316, 254] birim, en; [227, 235, 186, 146, 146, 155, 213, 191] ve yükseklik; [124, 159, 171, 184, 184, 273, 135, 191] birim olan kutular uzunluğu ([boy, en, yükseklik]) [1600, 2400, 2591] birim olan yükleme alanlarına paketlenmesi probleminin KTP çözümü aşağıdaki gibidir.

Çözüm sonucunda 1 kutu paketleme için kullanılmıştır.

4. HESABA DAYALI ÇÖZÜMLER

Bu bölümde, yukarıda modelleri belirtilen; ortogonal bin packing ve bin packing modellerinin çözümünde kullanılan KP ve KTP'nin performansını değerlendirmek için yaptığımız hesaplamalı çalışmalar sunulmuştur. Uygulama için varsayılan parametre ayarlarıyla IBM ILOG CPLEX OPL (sürüm 20.1.0) kullanılıp ve hesaplama deneylerinde 1.6 GHz'de çalışan Intel Core i5-10210U CPU ve 8 GB RAM ile donatılmış 64 bit bir bilgisayar kullanıldı.

AEORS (AEORS, "Avrupa Yöneylem Araştırma Dernekleri Birliği’dir ve "Uluslararası Yöneylem Araştırma Dernekleri Federasyonu" IFORS bünyesinde bölgesel bir gruptur. Amacı Yöneylem Araştırmasını tüm Avrupa'da teşvik etmektir. EURO, İsviçre'de yerleşik kâr amacı gütmeyen bir kuruluştur. [Web 1, (2023)]) kurumu tarafından web sitelerinde paylaşılan hazır veri setleri kullanılmıştır. Veri setlerinin yazılım üzerinde döngüsel olarak değişen ve zamanların sonuç olarak yazdırıldığı programlama işlemleri modellerin devamında eklenerek işlem kolaylığı sağlanmıştır.

4.1. Ortogonal Bin Packing Çözümleri

Bu problem türü için AEORS veri setleri kullanılmıştır ve paketleme yapılacak alan sayısı bir tanedir. Problemin çözümü için çalışılan örneklem boyutları 10, 20, 30, 40, 50, 60, 70, 80, 90, 100’dür. Belirtilen örneklem boyutlarının her biri için 5 farklı veri seti kullanılarak elde edilen ortalama veri süreleri tablolarda belirtilmiştir. KP ve KTP çözüm süreleri örneklem boyutlarına göre belirtilmiştir. KP çözümleri için kullanılan süre sınırı 5 dk, KTP içinse 60 dk olarak belirlenmiştir. Modeller bu şartlar altında veri setleri ile çalıştırılmıştır.

4.1.1. TBDPP Çözümleri

Ortogonal bin packing çözümleri kısmında belirtilen şartlar altında hazır veri setleri çalışılmış ve aşağıda tabloda belirtilen sonuçlar elde edilmiştir.

Tablo 4.1: TBDPP çözüm sonuçları (sn).

Örneklem Sayısı	KP	KTP
10	0,022	0,067
20	0,021	0,075
30	0,031	0,197
40	0,054	-
50	0,051	-
60	0,067	-
70	0,079	-
80	0,091	-
90	0,109	-
100	0,127	-

KTP modeli için örneklem sayısı 30'un üzerine çıktığı takdirde kullanılan veri setine göre çözüm süreci Ortogonal bin packing çözümleri kısmında belirtilen süre şartının üzerinde kaldığından dolayı "-" ile gösterilmiştir.

4.1.2. İBDPP Çözümleri

Ortogonal bin packing çözümleri kısmında belirtilen şartlar altında hazır veri setleri çalışılmış ve aşağıda tabloda belirtilen sonuçlar elde edilmiştir.

Tablo 4.2: İBDPP çözüm sonuçları (sn).

Örneklem Sayısı	KP	KTP
10	0,032	0,072
20	0,041	0,173
30	0,076	0,499
40	0,110	1,940
50	0,197	4,486
60	0,308	13,610
70	0,433	28,227
80	0,428	58,982
90	0,512	105,822
100	0,653	833,537

4.1.3. ÜBDPP Çözümleri

Ortogonal bin packing çözümleri kısmında belirtilen şartlar altında hazır veri setleri çalışılmış ve aşağıda tabloda belirtilen sonuçlar elde edilmiştir.

Tablo 4.3: ÜBDPP çözüm sonuçları (sn).

Örneklem Sayısı	KP	KTP
10	0,039	0,065
20	0,055	0,186
30	0,113	0,476
40	0,154	3,226
50	0,322	8,322
60	0,504	18,834
70	0,613	37,463
80	0,713	82,635
90	0,849	121,548
100	1,005	128,521

4.2. Bin Packing Çözümleri

Bu problem türü için AEORS veri setleri kullanılmıştır ve paketleme yapılacak alan sayısı birden fazladır. Problemin çözümü için çalışılan örneklem boyutları 10, 20, 30, 40, 50, 60, 70, 80, 90, 100'dür. Belirtilen örneklem boyutlarının her biri için 5 farklı veri seti kullanılarak elde edilen ortalama veri süreleri tablolarda belirtilmiştir. KP ve KTP çözüm süreleri örneklem boyutlarına göre belirtilmiştir. KP çözümleri için kullanılan süre sınırı 5 dk, KTP içinse 60 dk olarak belirlenmiştir. Modeller bu şartlar altında veri setleri ile çalıştırılmıştır.

4.2.1. TBKPP Çözümleri

Bin packing çözümleri kısmında belirtilen şartlar altında hazır veri setleri çalışılmış ve aşağıda tabloda belirtilen sonuçlar elde edilmiştir.

Tablo 4.4: TBKPP çözüm sonuçları (sn).

Örneklem Sayısı	KP	KTP
10	0,080	0,262
20	0,060	-
30	0,048	-
40	0,049	-
50	0,087	-
60	0,160	-
70	0,089	-
80	0,092	-
90	0,092	-
100	0,069	-

KTP modeli ve TBKPP problemi için örneklem sayısı 10’un üzerine çıktığı takdirde kullanılan veri setine göre çözüm süreci bin packing çözümleri kısmında belirtilen süre şartının üzerinde kaldığından dolayı “-“ ile gösterilmiştir.

4.2.2. İBKPP Çözümleri

Bin packing çözümleri kısmında belirtilen şartlar altında hazır veri setleri çalışılmış ve aşağıda tabloda belirtilen sonuçlar elde edilmiştir.

Tablo 4.5: İBKPP çözüm sonuçları (sn).

Örneklem Sayısı	KP	KTP
10	0,030	0,046
20	0,040	0,095
30	0,058	0,164
40	0,063	4,026
50	0,079	10,608
60	0,091	-
70	0,096	-
80	0,106	-
90	0,097	-
100	0,100	-

KTP modeli ve İBKPP problemi için örneklem sayısı 50'nin üzerine çıktığı takdirde kullanılan veri setine göre çözüm süreci bin packing çözümleri kısmında belirtilen süre şartının üzerinde kaldığından dolayı "-" ile gösterilmiştir. Burada KTP modeli örneklem sayısı olarak TBKPP ile kıyaslandığında daha fazla sayıda işlem yapmıştır. Bunun nedeni TBKPP'nin çözüm uzayının İBKPP'ye göre tek boyutta kalmasından dolayıdır.

4.2.3. ÜBKPP Çözümleri

Bin packing çözümleri kısmında belirtilen şartlar altında hazır veri setleri çalışılmış ve aşağıda tabloda belirtilen sonuçlar elde edilmiştir.

Tablo 4.6: ÜBKPP çözüm sonuçları (sn).

Örneklem Sayısı	KP	KTP
10	0,036	0,172
20	0,051	0,681
30	0,059	3,303
40	0,056	30,853
50	0,081	155,818
60	0,089	-
70	0,106	-
80	0,100	-
90	0,101	-
100	0,093	-

KTP modeli ve İBKPP problemi için örneklem sayısı 50'nin üzerine çıktığı takdirde kullanılan veri setine göre çözüm süreci bin packing çözümleri kısmında belirtilen süre şartının üzerinde kaldığından dolayı "-" ile gösterilmiştir.

5. SONUÇLAR

Bir, iki ve üç boyutlu kutu paketleme ve ortogonal kutu paketleme konusundaki çalışmanın sonuçları, optimizasyon için KP ve KTP kullanmanın etkinliğini göstermektedir. Ortogonal bin packing problemlerinde, modeller bin packing problemlerine göre daha fazla örneklem üzerinde çalıştı ve çözümleri elde etti.

Ortogonal bin packing problemleri için, KP yaklaşımı, optimizasyon süresi nispeten yüksek olmasına rağmen, yüksek derecede uygulanabilirliğe sahip çözümlerle sonuçlandı. Öte yandan, KTP, daha yüksek bir hesaplama süresiyle daha optimize edilmiş çözümler sağladı.

Bin packing problemleri için, KP yaklaşımı, optimizasyon süresi diğer problem türüne göre nispeten düşüktür. Bu problem türünde de KP yüksek derecede uygulanabilirliğe sahiptir. Öte yandan, KTP, daha yüksek bir hesaplama süresi ve bazı büyük sayılı örneklemlerde istenilen hesaplama süresi düzeyinde kalamamasından dolayı bu problem türünde KP'ye göre düşük performans göstermiştir.

Genel olarak, sonuçlarımız, kutu paketleme problemleri için KP ile KTP'nin kıyaslamasını sunmaktadır. KP'nin kullanımı, azaltılmış sürede çalışma ile uygulanabilir çözümler sağlarken, KTP, daha kesin sonuçlarla ancak uzun işlem süreleriyle çözümler sağlamıştır. Bu bulguların, lojistik, üretim ve dağıtım da dahil olmak üzere çeşitli endüstrilerdeki kutu paketleme problemlerinin pratik uygulaması için önemli çıkarımları vardır.

Gelecek çalışmalarda KP ile bu çalışmada kullanılmayan kırılganlık, yön değiştirebilme vb. kısıtların çözümleri araştırılarak KP uygulamaları özellikle lojistik sektöründe arttırılacaktır.

KAYNAKLAR

Brandao F., Pedroso J. P., (2016), “Bin packing and related problems: General arc-flow formulation with graph compression”, Computers & Operations Research, 69, 56-67.

Burke E. K., Hyde M. R., Kendall G., (2006), "Evolving bin packing heuristics with genetic programming", Parallel Problem Solving from Nature-PPSN IX: 9th International Conference, 860-869, Reykjavik, Iceland, 9-13 September.

Chazelle B., (1983), "The bottomn-left bin-packing heuristic: An efficient implementation", IEEE Transactions on Computers, 32.08, 697-707.

Delorme M., Iori M., Martello S., (2016), “Bin packing and cutting stock problems: Mathematical models and exact algorithms”, European Journal of Operational Research, 255(1), 1-20.

Dósa G., (2007), "The tight bound of first fit decreasing bin-packing algorithm is FFD (I)≤ 11/9 OPT (I)+ 6/9", Combinatorics, Algorithms, Probabilistic and Experimental Methodologies: First International Symposium, 1-11, Hangzhou, China, 7-9 April.

Gonçalves J. F., Resende M. G. C., (2013), "A biased random key genetic algorithm for 2D and 3D bin packing problems", International Journal of Production Economics, 145.2, 500-510.

Gonçalves J. F., (2007), "A hybrid genetic algorithm-heuristic for a two-dimensional orthogonal packing problem", European Journal of Operational Research, 183.3, 1212-1229.

Goodman E. D., Tetelbaum A. Y., Kureichik V. M., (1994), "A genetic algorithm approach to compaction, bin packing, and nesting problems", Technical Report No: 940702, Case Center for Computer-aided Engineering and Manufacturing, Michigan State University, USA.

Gyárfás A., Lehel J., (1988), "On-line and first fit colorings of graphs", Journal of Graph theory, 12.2, 217-227.

Haouari M., Serairi M., (2009), "Heuristics for the variable sized bin-packing problem", Computers & Operations Research, 36.10, 2877-2884.

Hifi M., Kacem I., Négre S., Wu L., (2010), "A linear programming approach for the three-dimensional bin-packing problem", Electronic Notes in Discrete Mathematics, 36, 993-1000.

Hu H., Zhang X., Yan X., Wang L., Xu Y., (2017), "Solving a new 3d bin packing problem with deep reinforcement learning method", arXiv preprint, arXiv:1708.05930.

Iori M., De Lima V. L., Martello S., Miyazawa F. K., Monaci M., (2021), “Exact solution techniques for two-dimensional cutting and packing”, European Journal of Operational Research, 289(2), 399-415.

Kenyon C., (1996), "Best-Fit Bin-Packing with Random Order", Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, 359-364, Atlanta Georgia, USA, 28-30 January.

Levine J., Ducatelle F., (2004), "Ant colony optimization and local search for bin packing and cutting stock problems", Journal of the Operational Research Society, 55.7, 705-716.

Lin, J-L., Foote B., Pulat S., Chang C-H., Cheung J. Y., (1993), "Hybrid genetic algorithm for container packing in three dimensions", Proceedings of 9th IEEE Conference on Artificial Intelligence for Applications, 353-359, Orlando, FL, USA, 01-05 March.

Lodi A., Martello S., Vigo D., (2002), "Recent advances on two-dimensional bin packing problems", Discrete Applied Mathematics, 123.1-3, 379-396.

López-Camacho E., Terashima-Marin H., Ross P., Ochoa G., (2014), "A unified hyper-heuristic framework for solving bin packing problems", Expert Systems with Applications, 41.15, 6876-6889.

Coffman E. G. Jr., Garey M. R., Johnson D. S., (1996), Approximation algorithms for NP-hard problems, 1st Edition, Course Technology Inc.

Martello S., Vigo D., (1998), "Exact solution of the two-dimensional finite bin packing problem", Management science, 44.3, 388-399.

Martello S., Pisinger D., Vigo D., (2000), "The three-dimensional bin packing problem", Operations research, 48.2, 256-267.

Smith D., (2014), "Bin packing with adaptive search", Proceedings of the First International Conference on Genetic Algorithms and their Applications, 1st Edition, Psychology Press.

Sridhar R., Chandrasekaran M., Sriramya P., Page T., (2016), "Optimization of heterogeneous Bin packing using adaptive, genetic algorithm", IOP conference series: materials science and engineering, 196-208, Chennai, India, 20-21 October.

Tole K., Moqa R., Zheng J., He K., (2023), “A Simulated Annealing approach for the Circle Bin Packing Problem with Rectangular Items”, Computers & Industrial Engineering, 109004.

Wang H., Chen Y., (2010), "A hybrid genetic algorithm for 3D bin packing problems", 2010 IEEE Fifth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA), 703-707, Changsha, China, 23-26 September.

Web 1, (2023), https://www.euro-online.org/web/pages/1/home, (Erişim Tarihi: 09/02/2023).

Wei L., Oon W-C., Zhu W., Lim A., (2013), "A goal-driven approach to the 2D bin packing and variable-sized bin packing problems", European Journal of Operational Research, 224.1, 110-121.

Wei L., Luo Z., Baldacci R., Lim A., (2020), "A new branch-and-price-and-cut algorithm for one-dimensional bin-packing problems", INFORMS Journal on Computing, 32.2, 428-443.

Yao A. C-C., (1980), “New algorithms for bin packing”, Journal of the ACM (JACM), 27(2), 207-227.

ÖZGEÇMİŞ

Ahmet KARAKAŞ, 2015 yılında başladığı Sakarya Üniversitesi Mühendislik Fakültesi Endüstri Mühendisliği Bölümünü 2019 yılında başarıyla tamamlayarak aynı yıl yüksek lisans eğitimine Gebze Teknik Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalında başladı. 2015 yılından bu yana optimizasyon, veri bilimi ve yapay zekâ üzerine çalışmaktadır.