T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

# BÜYÜK BOYUTLU KUTULAMA PROBLEMLERİ İÇİN BİR ALGORİTMA ÖNERİSİ

SEZGİ TEKİL

YÜKSEK LİSANS TEZİ
ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI
ENDÜSTRİ MÜHENDİSLİĞİ PROGRAMI

DANIŞMAN
YRD. DOÇ. DR. VİLDAN ÇETİNSAYA ÖZKIR

İSTANBUL, 2017

**T.C.**

**YILDIZ TEKNİK ÜNİVERSİTESİ**

**FEN BİLİMLERİ ENSTİTÜSÜ**

# BÜYÜK BOYUTLU KUTULAMA PROBLEMLERİ İÇİN BİR ALGORİTMA ÖNERİSİ

Sezgi TEKİL tarafından hazırlanan tez çalışmasıtarihinde 09.05.2017 aşağıdaki jüri tarafından Yıldız Teknik Fen Bilimleri Endüstri Mühendisliği Anabilim Dalı'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

**Tez Danışmanı**

Yrd. Doç. Dr. Vildan ÖZKIR

Yıldız Teknik Üniversitesi

**Jüri Üyeleri**

Yrd. Doç. Dr. Vildan ÖZKIR

Yıldız Teknik Üniversitesi _____

Yrd. Doç. Dr. Tufan DEMİREL

Yıldız Teknik Üniversitesi _____

Prof. Dr. Şule ÖNSEL

Doğuş Üniversitesi _____

# ÖNSÖZ

Teknolojinin gelişmesiyle artan üretim hacmi, küreselleşmenin sonucu ile gelişen ticaret, lojistik operasyonlarının iyileştirilmesini zorunlu kılmıştır. Bu operasyonları yüksek hızda ve düşük maliyetle yürütmek için matematiksel optimizasyon modelleri kullanılmaktadır. Konteyner yükleme problemi (KYP), küresel lojistik ve tedarik zinciri için önemli endüstriyel ve ticari uygulamaya sahiptir. Optimum sonuçlara ulaşabilmek için kesin çözüm veren algoritmaların kullanılmasının yanı sıra, büyük boyutlu problemlerin çözümünde kesin algoritmalar yetersiz kalabilmektedir. Bu çalışmada, bir filtre fabrikasının lojistik operasyonları incelenerek, ticari araçların ve konteynerlerin karmaşık yük ve yükleme planlarının optimizasyonu amaçlanmaktadır. Kutulama algoritmaları, konteyner sayısının azaltılması ve yükleme hızının arttırılması amacıyla geliştirilerek önerdiğimiz İki Boyutlu Kombinatoryal Alan Maksimizasyon Algoritması-2D-CAMFA modeli ile uygun optimal kararların alınmasında kullanılacaktır. Bu tez çalışmamda beni her zaman destekleyen, akademik deneyim ve bilgileriyle tezimin bu aşamaya kadar gelmesini sağlayan başta danışman hocam Yrd. Doç. Dr. Vildan ÖZKIR'a, geliştirdiğimiz algoritmanın modelini kurma ve kodunu yazma konusunda aylarca çalıştığımız, modelin her aşamasında emeği olanAr. Gör. Şahika Koyun'a, destekleri içinAr. Gör. Ezgi Uluhan'a, tezimi uygulamak için lojistik operasyonlarını ve verilerini paylaşan Filmar Filtre LTD ŞTİ firmasına, beni her konuda destekleyen her zaman yanımda olan umarım uzun yıllar da yanımda olacak babam Celal Tekil'e, annem Mine Tekil'e, Burak Ergün'e ve aileme..

Çok Teşekkür Ederim…

Nisan, 2017

Sezgi TEKİL

# İÇİNDEKİLER

Sayfa

SİMGE LİSTESİ ..... vii

KISALTMA LİSTESİ ..... viii

ŞEKİL LİSTESİ ..... ix

ÇİZELGE LİSTESİ ..... x

ÖZET ..... xi

ABSTRACT ..... xiii

BÖLÜM 1

GİRİŞ ..... 1

1.1 Literatür Özeti ..... 2

1.2 Tezin Amacı ..... 4

1.3 Hipotez ..... 5

BÖLÜM 2

KUTULAMA PROBLEMİ ..... 6

2.1 Kutulama Problemlerinin Çeşitleri ..... 6

2.2 Bir boyutlu kutulama problemleri ..... 8

2.2.1 Bir Boyutlu Kutulama Modeli ..... 8

2.2.2 Bir Boyutlu Kutulama Problemleri Algoritmaları ..... 9

2.3 İki Boyutlu Kutulama Problemleri ..... 12

2.3.1 İki Boyutlu Kutulama Probleminin Yapısı ..... 12

2.3.2 İki Boyutlu Kutulama Probleminin Modeli ..... 12
2.3.3 İki Boyutlu Kutulama Problemi Algoritmaları ..... 14
2.4 Üç Boyutlu Kutulama Problemi ..... 21
2.4.1 Üç Boyutlu Kutulama Probleminin Yapısı ..... 21
2.4.2 Üç Boyutlu Kutulama Probleminin Modeli ..... 22
2.4.3 Üç Boyutlu Kutulama Problemi Algoritmaları ..... 25
BÖLÜM 3
KONTEYNER YÜKLEME PROBLEMİ ..... 26
3.1 Konteyner Yükleme Problemlerinin Çeşitleri ..... 29
3.2 Konteyner Yükleme Problemlerinin Kısıtları ..... 30
3.2.1 Konteyner Kaynaklı Kısıtlar ..... 30
3.2.2 Paket ile ilgili Kısıtlar ..... 32
3.2.3 Kargo Kaynaklı Kısıtlar ..... 33
3.2.4 Yükleme Kaynaklı Kısıtlar ..... 34
BÖLÜM 4
ÖNERİLEN YÖNTEM ..... 35
4.1 İki Boyutlu En Uygun Yerleştirme Raf Algoritması ..... 35
4.2 İki Boyutlu Kombinatoryal Alan Ençoklama Yerleştirme Algoritması ..... 38
4.2.1 İki Boyutlu Kombinatoryal Alan Maksimizasyon Yerleştirme Modeli ..... 40
4.3 İki Boyutlu Kombinatoryal Alan Maksimizasyon Yerleştirme Algoritmasının Performansının Değerlendirilmesi ..... 43
BÖLÜM 5
LOJİSTİK SEKTÖRÜNDE BİR UYGULAMA ..... 44
5.1 Problemin Tanımı ..... 45
5.2 Konteynerlerin Özellikleri ..... 46
5.3 Amaçlar ve Kısıtlar ..... 48
5.4 2D-CAMFA Uygulanması ve Sonuçları ..... 48
BÖLÜM 6
SONUÇLAR VE ÖNERİLER ..... 49
KAYNAKLAR ..... 51

EK-A

VERİLER.....60
ÖZGEÇMİŞ..... 73

## SİMGE LİSTESİ

$A_j$ j indisli bölmenin ikili satır matrisi

$a_m$ m adet nesnenin hacmi

B Kullanılan kutuların optimum değeri

H Konteyner yüksekliği

$h_i$ i ndisli nesnenin yüksekliği

m R listesindeki nesne sayısı

N Kullanılan kutunun adeti

$p_{ij}$ i indisli nesnenin j indisli bölmesine yerleşme durumu

V Kutunun hacmi

W Konteyner genişliği

$w_i$ i indisli nesnenin genişliği

$x_{ij}$ i indisli nesnenin j kutusuna yerleşme durumu

$y_i$ i indisli nesnenin yerleşme durumu

## KISALTMA LİSTESİ

| | |
|---|---|
| 1BKP | Bir Boyutlu Kutulama Problemleri |
| 2BKP | İki Boyutlu Kutulama Problemleri |
| 3BKP | Üç Boyutlu Kutulama Problemleri |
| AYA | Alternatif Yönergeler Algoritması |
| BF | En Uygun Yer Algoritması |
| BFD | Azalan En Uygun Yer Algoritması |
| FF | İlk Uygun Yer Algoritması |
| FFD | Azalan İlk Uygun Yer Algoritması |
| GA | Giyotin Kesme Algoritması |
| KP | Kutulama Problemleri |
| MAXRECT | En Büyük Dikdörtgensel Alan Algoritması |
| MAXRECTS-BL | Sol Alt Maksimum Diktörtgensel Alan Algoritması |
| NF | En Yakın Uygun Yer Algoritması |
| NFD | Azalan En Yakın Uygun Yer Algoritması |
| SA | Raf Yerleştirme Algoritması |
| S-BF | En Uygun Sığan Raf Yerleştirme Algortiması |
| S-BFD | Raf Yerleşimde En Uygun Sığanların Azalan Yükseklik Stratejisi |
| S-FFD | Raf Yerleşimde İlk Sığanların Azalan Yükseklik Stratejisi |
| S-NF | En Yakın Sığan Raf Yerleştirme Algoritması |
| S-NFD | Raf Yerleşimde Sonraki Sığanların Azalan Yükseklik Stratejisi |
| WF | En Kötü Uygun Yer Algoritması |
| WFD | Azalan En Kötü Uygun Yer Algoritması |

## ŞEKİL LİSTESİ

Sayfa

Şekil 2.1 Bir Boyutlu Kutulama Problemi ..... 7
Şekil 2.2 İki Boyutlu Kutulama Problemi ..... 7
Şekil 2.3 Giyontinsiz Örnek ve Giyotinli Örnek [25]. ..... 14
Şekil 2.4 Raf Yerleştirme Stratejileri[30] ..... 15
Şekil 2.5 Raf Yerleştirme Stratejileri [30] ..... 16
Şekil 2.6 Alternatif Yönergeler Algoritması [30] ..... 17
Şekil 2.7 Raf Yerleştirme ..... 18
Şekil 2.8 MAXRECT Veri Seti İçin Yerleşme Kuralı[30] ..... 20
Şekil 2.9 MAXRECTS-BL Algoritmasıyla Üretilen Maksimum Dikdörtgenler [30] ..... 21
Şekil 2.10 Üç Boyutlu Kutu Kordinatları[20] ..... 22
Şekil 2.11 Üç Boyutlu Kutuların Yerleşmesi ..... 25
Şekil 3.1 Konteyner Yükleme ..... 26
Şekil 3.2 Tır ve Konteyner Ölçüleri ..... 30
Şekil 3.3 Paket Kaynaklı Kısıtlar ..... 32
Şekil 3.4 Kargoların Müşterilere Teslimi ..... 33
Şekil 3.5 Konteynerin Taşınması ..... 34
Şekil 4.1 İki Boyutlu Nesne ve Konteyner Görünümü ..... 36
Şekil 4.2 İki Boyutlu BFD-Raf Algoritması Pseudo Kodu ..... 37
Şekil 4.3 Kutulara Sığabilecek Nesnelerin Kombinasyon Örneği ..... 39
Şekil 4.4 Nesnelerin Kutulara Yerleşmesi ..... 39
Şekil 5.1 İskenderun Lima'nından bir görüntü ..... 44
Şekil 5.2 Filtre Çeşitleri ..... 45
Şekil 5.3 Filtrelerin Çeşitleri, Filtre Boyutları Ve Koli Ölçüleri ..... 46
Şekil 5.4 Konteyner İçi Paket Yükleme ..... 46
Şekil 5.5 Sırasıyla 20' ve 40' Dry Van, Hig Cube Konteyner ..... 47

## ÇİZELGE LİSTESİ

Sayfa

Çizelge 2.1 Sezgisel Algoritmaların Karşılaştırılması [22] ..... 11
Çizelge 2.2 Şerit Yerleştirme Algoritmaları [6] ..... 16
Çizelge 3.1 Literatürdeki Kutulama ve Konteyner Yükleme Çalışmaları ..... 27
Çizelge 3.2 Literatürdeki Kutulama ve Konteyner Yükleme Çalışmaları(devam)..... 28
Çizelge 4.1 2D-CAMFA Sözde Kodu ..... 40
Çizelge 4.2 2D-CAMFA Sözde Kodu (devam) ..... 41

# ÖZET

## BÜYÜK BOYUTLU KUTULAMA PROBLEMLERİ İÇİN BİR ALGORİTMA ÖNERİSİ

Sezgi TEKİL

Endüstri Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

Tez Danışmanı: Yrd. Doç. Dr. Vildan ÇETİNSAYA ÖZKIR

Kutulama problemi, kullanılan kutu sayısını en aza indirmek amacıyla, çeşitli hacimlerdeki nesnelerin çeşitli hacimlerdeki kutulara yerleştirilmesi problemidir. Kutulama problemi NP-hard problemi olmasına rağmen, büyük boyuttaki problemler için ileri düzeydeki algoritmalarla optimal çözümler bulunabilmektedir.

Kutulama problemleri bir nevi kesim problemidir. Kutulama problemlerinin özel diğer bir hali de, her bir nesne hacmi ve değeri ile karakterize edildiğinde yalnızca bir kutunun azami değerde nesne ile doldurulması problemi sırt çantası problemi olarak bilinmektedir. Kutulama problemlerinin birçok varyasyonları vardır; 2B kutulama, lineer kutulama, ağırlıkça paketleme, maliyetçe yerleştirmeve benzerleri gibi. Ayrıca, kapları doldurma, ağırlık kapasite kısıtları ile kamyon yükleme, FPGA yarı iletken çip tasarımı,

medya ve teknoloji haritalama, dosya yedeklerini oluşturmak gibi birçok kutulama uygulamaları vardır.

Kutulama problemlerinin karşımıza çıktığı diğer bir alanda konteyner atama problemleridir. Günümüzde teknolojinin gelişmesi ve küreselleşme ile artan ticaret hacmi, lojistik operasyonlarının optimizasyonunu zorunlu hale getirmiştir. Bu operasyonları yüksek hızda, yüksek doğrulukta ve düşük maliyetle yürütmek için matematiksel optimizasyon araçları ve modelleri sıklıkla kullanılmaktadır. Konteyner yükleme problemleri, sık karşılaşılan lojistik optimizasyon problemlerinden biridir. Optimum sonuçlara ulaşabilmek için kesin çözüm veren algoritmaların kullanılmasının yanı sıra, büyük boyutlu problemlerin çözümünde kesin algoritmalar yetersiz kalabilmektedir. Bu çalışmada, bir filtre fabrikasının lojistik operasyonları incelenerek, ticari araçların ve konteynerlerin karmaşık yük ve yükleme planlarının optimizasyonu amaçlanmaktadır. Kutulama algoritmaları, konteyner sayısının azaltılması ve yükleme hızının arttırılması amacıyla geliştirilecek ve uygun optimal kararların alınması için önerdiğimiz 2B-CAMFA modeli uygulanacaktır.

**Anahtar Kelimeler:** Kutulama Problemleri, Sezgisel Algoritmalar, Konteyner Yükleme Problemleri,2B-CAMFA

## ABSTRACT


### AN ALGORITHM FOR LARGE SCALE BIN PACKING PROBLEM

Sezgi Tekil

Department of Industrial Engineering

MSc. Thesis

Advisor: Assist. Prof. Dr. Vildan ÇETİNSAYA ÖZKIR

In the bin packing problem, objects of different volumes must be packed into a finite number of bins or containers each of volume V in a way that minimizes the number of bins used. Despite the fact that the bin packing problem has an NP-hard computational complexity, optimal solutions to very large instances of the problem can be produced with sophisticated algorithms.

The bin packing problem can also be seen as a special case of the cutting stock problem. When the number of bins is restricted to 1 and each item is characterised by both a volume and a value, the problem of maximising the value of items that can fit in the bin is known as the knapsack problem. There are many variations of this problem, such as 2D packing, linear packing, packing by weight, packing by cost, and so on. They have many applications, such as filling up containers, loading trucks with weight capacity constraints, creating file backups in media and technology mapping in Field-programmable gate array semiconductor chip design.

Recently, the technology is improving day by day and the globalization results in increasing of international trade. Because of these improvements, logistic operation must be faster than the past. For keeping on these operations with low cost, higher

accuracy and rapidity, the mathematical optimization tools and models are highly used. Container loading problem is also one of the logistic operations. Due to get optimum results we must apply appropriate algorithms but in some cases these algorithms are not enough for solutions of three dimensional bin packing problems. In this study, we investigate the logistic operations of a filter factory and for the commercial vehicles and container's' complex load and loading plan. By improving the work load speed and decreasing the number of container in bin packing problems, we will implement the 2D Combinatorial Area Fitting Algorithm 2D-CAMFA model that we recommend to make appropriate optimum decisions. We also compare the solutions of Self-Best Fit Algorithm with 2D-CAMFA model.

**Keywords:** Bin packing, Heuristic Algorithms,2D-CAMFA

# BÖLÜM 1

## GİRİŞ

Global dünyada artan uluslararası ticaret, lojistik operasyonlarının optimizasyonunu zorunlu hale getirmiştir. Lojistik sistemlerin matematiksel optimizasyon ve istatistiksel karar verme yaklaşımları kullanılarak modellenmesi, çeşitli amaçlar gözetilerek uygun çözüm seçeneklerinin değerlendirilmesi bu konudaki çabaların bir sonucudur. Ağ tasarımı problemleri, envanter problemleri gibi ana problem başlıklarının yanı sıra, lojistik operasyonların en verimli şekilde gerçekleştirilebilmesi için spesifik lojistik problemleri günümüzde bilim adamlarının dikkatini çekmektedir. Konteyner yükleme ve yerleştirme problemleri de bu özel problemlerden biridir. Problemin temel amacı, optimal kapasite kullanımı ile kullanılan konteynerlerin enazlanması operasyonların verimliliğini arttırmaktadır. Konteynerler, dayanıklı ve uzun ömürlü olması, kolay yerleştirilebilir ve taşınabilir olması özellikleri nedeniyle, taşıma sürecinin verimli ve etkin olarak gerçekleştirilmesine olanak sağlamaktadır. Bu nedenle, toplam lojistik maliyetlerini enazlamak üzere, konteyner rotalama veyükleme problemlerininoptimizasyonu problemleri ilgi çekici hale gelmektedir.

Konteyner yükleme problemleri, yük ebatları ve konteynerlerin çeşitli kapasitelerinin değişkenliği nedeniyle, 'NP-zor' problemler sınıfına aittir. Bu durum problem boyutu büyüdükçe, problemin etkin bir şekilde çözülmesi için kullanılabilecek sezgisel algoritmaların önemini arttırmaktadır. Konteyner yükleme problemleri, literatürde kutulama problemleri olarak anılmaktadır.

Bu çalışma, kullanılan toplam kutu sayısını enazlamak ve kutu içi azami alan kullanımını teşvik etmek amacıyla, iki boyutlu kutulama problemlerinin çözümü için sezgisel bir algoritma

önermektedir. Geliştirilen İki Boyutlu Kombinatoryal Alan Yerleştirme Algoritması (2D-CAMFA), literatürde yer alan En Uygun Sığan Raf Yerleştirme Algoritması (S-BF) ile karşılaştırılarak, çalışma süresi ve çözüm kalitesi açısından daha iyi sonuçlar üretmektedir.

Tez çalışmasının birinci bölümünde problem hakkında genel bilgi verilmekte, tezin amacı ve hipotezi özetlenmektedir. İkinci bölümde, tek boyutlu, iki boyutlu ve üç boyutlu kutulama problemlerinin yapıları, matematiksel modelleri ve geliştirilen çözüm algoritmaları açıklanmaktadır. Üçüncü bölüm, konteyner yükleme problemlerinin çeşitleri, özellikleri ve özel kısıtlarını ayrıntılı olarak açıklamaktadır. Dördüncü bölümde, kutulama problemlerinin çözümünde kullanılması önerilen kutulama algoritması olan 2D-CAMFA anlatılmaktadır. Önerilen algoritmanın performansını göstermek üzere gerçekleştirilen deneysel çalışma da bu bölümde yer almaktadır. Beşinci bölümde, Türkiye sınırları içinde yer alan bir filtre fabrikasının lojistik operasyonlarının önemli bir bileşeni olan yükleme problemi ele alınmaktadır. Bu bölümde, firmanın konteyner yükleme probleminin çözümü için literatürde yer alan algoritmalar ve bu tez çalışması kapsamında önerilen 2D-CAMFA sonuçları karşılaştırmalı olarak verilmektedir. Son bölümde, kutulama problemlerinin çözümünde geliştirilen kutulama algoritmasının performansını deneysel ve gerçek veri üzerindeki sonuçları tartışılmaktadır. Bu karşılaştırılmalı tartışma ve algoritmanın varsayımlarından hareket ederek, gelecek çalışmalar için öneriler de bu bölümde sunulmaktadır.

## 1.1 Literatür Özeti

Kutulama problemleri bir çeşit kesim problemidir [1]. Kesim problemleri, belirli büyüklükteki nesneleri, genellikle fire miktarını enazlamak amacıyla, çeşitli büyüklük ve miktarda nesnelere kesilmesi problemidir. Özel bir kutulama problemi olan sırt çantası problemi ise, nesnelerin hacim ve değer bilgisinin kullanılması ile toplam azami değeri elde edecek bir sırt çantasının doldurulmasıdır.

Kutulama algoritmalarının geliştirilmesinde ilk katkı sağlayan George ve Robinson [2]'ın çalışmalarını takiben, Bischo ve Marriott [3] yeni sezgisel algoritmalar literatüre kazandırmıştır. Sırt çantası yükleme problemlerinin çözümüne yönelik araştırmaların öncüleri ise Gehring vd. [20] dir.

Üretim ve lojistik operasyonlarının yönetilmesi sırasında karşılaşılan en yaygın problem olan kutulama problemi, göreceli olarak küçük boyuttaki nesnelerin belirli boyutlardaki göreceli olarak daha büyük olan kutulara yerleştirilmesi için ebat, hacim ve çeşitli kapasite kısıtları göz önüne alınarak yerleştirilmesidir. Üçüncü bölümde ayrıntılı olarak açıklanan yerleştirme kısıtları, bu problemin çözümünü daha da zorlaştırmaktadır. Kutulama problemleri olarak karşılaşılan en önemli alanlar konteyner yükleme [3], kapasite kısıtlı tır yükleme [4], bulut teknolojisi [5], teknoloji haritalama çalışma alanlarıdır.

Kutulama problemlerinin matematiksel modellenmesi konusunda, ilk olarak 1970'lerin başında, Coffman vd. [6] problemin matematiksel temelleri üzerinde çalışmış ve yerleştirme problemlerinin çözümü için dört sezgisel algoritma tasarlamıştır

Wolke vd. [5], bulut veri merkezlerinde dinamik kaynak ayırma stratejilerini analiz etmiştir. Sanal makinelerini incelemek için tek boyutlu kutulama sezgisel yöntemlerini kullanmıştır.

Paul vd. [7], sanal makinelerin enerji tüketimini ve dağıtım maliyetini enazlamak amacıyla, sanal makine yerleştirme problemini çözmüştür. Bu çalışmada, çözüm için En İyi Sığan (BF), İlk Sığdırma (FF), Sonraki Sığdırma (NF), En Kötü Sığdırma (WF) ve Rasgele Ayırma (RF) algoritmaları kullanılmış, en iyi sığan algoritması diğerlerine göre daha iyi sonuçlar üretmiştir.

Wang [8], iki boyutlu kutulama problemlerinin çözümüne yönelik, alan verimliliğini esas alan Dip-Alt-Sol doldurma (Deepest-Bottom-Left-Fill) ve Azami Dokunan Alan (Maximum Touching Area) adında iki algoritma önermiştir. Önerilen sezgisel algoritmalar, 22 test probleminde Tabu Arama ve Karınca Kolonisi Eniyilemesi algoritmalarından çözüm kalitesi ve hız açılarından daha iyi sonuç vermiştir.

Geiger[9], kullanılan kutu sayısını enazlamak amacı ile birlikte kutuların heterojenliğini de aynı anda enazlamayı amaçlayan tek boyutlu kutulama probleminin çözümü için çok amaçlı En İyi Sığan (BF) algoritması geliştirmiştir.

Fernández vd.[10], nesne döndürülmesini izin verilen iki boyutlu kutulama probleminin çözümü için tek amaçlı ve çok amaçlı memetik algoritma önermiştir. Önerilen çok amaçlı algoritmanın performansını, Zitzler ve Thiele [11] tarafından performans değerlendirme metrikleri olan kapsam metriği ve hiper hacim metriği açısından çok amaçlı parçacık sürü

optimizasyonu algoritması ile karşılaştırılmış ve önerilen algoritmanın daha iyi sonuçlar elde ettiğini belirtmiştir.

Kang vd. [12], üç boyutlu bir kutulama probleminin çözümü için hibrit genetik algoritma önermiş ve algoritmanın performansını literatürde yer alan [13] çözümlerle karşılaştırmıştır.

Lim vd. [14], üç boyutlu kutulama probleminin çözümü için çok yüzlü yapılandırma tekniği geliştirmiş ve deneysel çalışmalar ile tekniğin performansını konteyner yükleme problemleri için önerilen sezgiselı [13] yöntemler ile kıyaslamak suretiyle göstermiştir.

Li vd. [15], üç boyutlu kutulama problemini öncelikle bir karma tamsayılı programlama modeli olarak sunmuş ve çözümü için yeni bir yerleştirme sezgiseli stratejisi ile diferansiyel evrim algoritmasını birleştirmiştir. Geliştirilen yöntem, 30 test probleminde Dip-Alt-Sol doldurma (Deepest-Bottom-Left-Fill) sezgiseline göre performans üstünlüğü göstermiştir.

Lia vd. [16] , üç boyutlu kutulama problemlerine yeni bir alt sınır geliştirmişlerdir. Üç boyutlu kutulama problemlerine kombinatoryal bir bakış açısından yeni bir alt sınır sunarak literatürdeki sonuçlara teorik olarak hakim olduklarını göstermektedir.

Thapatsuwan vd. [17], üç boyutlu kutulama probleminin çözümü için yapay bağışıklık sisteminin kullanımını önermiştir. Literatürde yer alan 21 test problemi için, yapay bağışıklık sisteminin genetik algoritma ve parçacık sürü optimizasyonu yöntemlerinden daha iyi sonuçlar ürettiğini göstermiştir.

Sheikhalishahi vd. [18], her bir iş için çok kaynaklı ve farklı kapasiteli çizelgeleme problemi için, kapasite dengesizliklerini gidermek amacıyla, kuyruk ve servisçi seçimi gerçekleştiren sezgisel kutulama algoritması geliştirmiştir.

## 1.2 Tezin Amacı

Kutulama problemi, farklı boyutlardaki birden çok nesnenin farklı boyutlardaki birden çok kutuya yerleştirilmesi problemidir. Konteyner yükleme problemlerinin çözümünde kutulama problemlerinin çözümü için önerilen algoritmalar kullanılmaktadır. Lojistik maliyetlerin enazlamasında, optimal konteyner yükleme planlarının oluşturulması yüksek öneme sahiptir.

Konteyner yükleme probleminin hesaplama açısından yüksek karmaşıklığa sahip olması, geliştirilen algoritmaların performansını sınırlamaktadır. Gerçek hayattaki problemlerin daha çok kısıt ve amaç içermesi, mevcut algoritmaların pratikte uygulanabilirliğini azaltmaktadır. Bu tez çalışması ile, iki boyutlu kutulama probleminin çözümü için, nesnelerin üçüncü bir özelliğinin de (derinlik, ürün çeşidi, öncelik, ağırlık vb.) değerlendirilmesi ve kutuda azami alan kullanımını sağlayan nesneyi öncelikli olarak yerleştirmek suretiyle toplam kullanılan kutu sayısını enazlamayı sağlayan sezgisel bir algoritma geliştirilmesi amaçlanmaktadır. Ayrıca, gerçek bir firmanın konteyner yükleme problemini ele alarak, algoritmanın etkin çözüm üretmesi sayesinde firmanın kullanılan konteyner sayısının enazlanması ve lojistik maliyetlerinin düşürülmesi sağlanmaktadır.

### 1.3 Hipotez

Konteyner yükleme problemleri kendi içinde bulundurduğu karmaşık yapısından dolayı NP-zor problemlerdir. Bu problemlerin çözümüne yönelik yapılan çalışmalar da kutulama problemlerinin algoritmalarına başvurulmuştur. Raf yerleştirme algoritması da bunlardan birisidir. Ancak, Raf Yerleştirme Algoritmalarının problemin çözümde yetersiz kaldığı sonucu çıkarılmış ve bu algoritmaya geliştirdiğimiz İki Boyutlu Kombinatoryal Alan Maksimizasyon Alan Algoritması-2D-CAMFA modeli önerilmiştir. Elde ettiğimiz sayısal sonuçlara göre 2D-CAMFA modelinin konteyner azlaması ve konteyner içi alan çoklamasından ötürü Raf Yerleştirme Algoritmasından daha iyi olduğu görülmüştür.

# BÖLÜM 2

# KUTULAMA PROBLEMİ

Kutulama problemi, farklı hacimlerdeki nesnelerin farklı hacimlerdeki birden çok kutuya yerleştirilmesi problemidir. Kutulama probleminin genel amacı, kullanılan kutu sayısını enazlamaktır. Literatürdekutulama problemi NP-hard problem olmasına rağmen, büyük boyuttaki problemler için geliştirilen algoritmalarla optimal çözümler bulunabilmektedir.

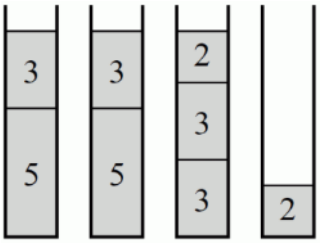
Literatürde incelenen kutulama problemleri, çoğunlukla konteyner yükleme problemi ve sırt çantası yükleme problemi başlıkları altında incelenmektedir. Konteyner yükleme problemi, mevcut konteyner hacimlerinden yüksek seviyede faydalanmak suretiyle, maliyet enazlanması hedefini amaçlamaktadır. Konteyner yükleme problemi, geometrik atama modeli olarak tanımlanmakta ve tüm dikdörtgensel nesnelerin dikdörtgenselkonteynerlere atanmasını sağlamayı amaçlamaktadır. Konteyner yükleme problemleri, karmaşıklık açısından NP-zor olduğundan, literatürde etkin çözümler üreten sezgisel algoritmalar geliştirilmektedir.

## 2.1 Kutulama Problemlerinin Çeşitleri

Kutulama problemleri, öncelikli olarak yerleştirme yapılırken göz önünde bulundurulan boyut sayısına göre sınıflandırılır. Bunlar,

- ***Bir boyutlu kutulama problemi*:** 1BKP'de farklı hacimdeki farkli miktarlarda bulunan nesnelerin belirli bir hacimdeki kutulara yerleştirilmesini içerir. Burada yerleştirme özelliği olarak ağırlık özelliği de kullanılabilir. Önemli olan durum, yerleştirme yapılan

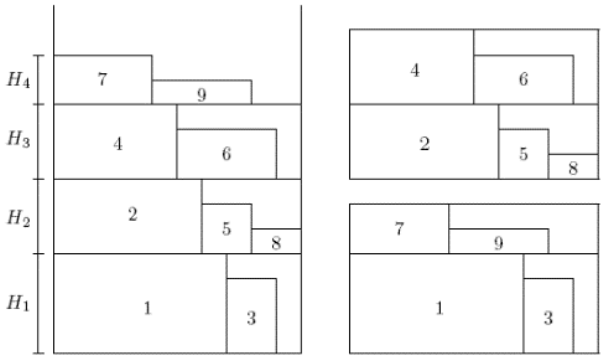
kutuların azami kapasitesini aşmamaktır. Problemin amacı, kullanılan toplam kutu sayısını enazlamak veya kullanılmayan kutu hacmini enazlamaktır.




Şekil 2.1 Bir Boyutlu Kutulama Problemi

Şekil 2.1'de tek boyut değerlendirilmek suretiyle nesnelerin kutulara yerleştirilmesini göstermektedir. [19].

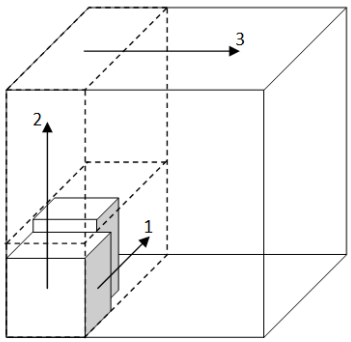
- ***İki boyutlu kutulama problemi*:** 2BKP'de genellikle nesnelerin yükseklik ve genişliği göz önünde bulundurularak iki boyutlu bir kutunun içini yerleştirmek amaçlanır.




Şekil 2.2 İki Boyutlu Kutulama Problemi

Şekil 2.2 'de görüldüğü gibi iki boyutlu nesneleriki boyutlu kutulara yerleştirilir[20].Kâğıt, cam, tel gibi iki boyutlu yerleştirme yapılabilecek ürünler için pratikte uygulamaları bulunmaktadır.

- ***Üç boyutlu kutulama problemi*:** 3BKP'de genişlik, yükseklik ve derinlik göz önüne alınarak, dikdörtgenprizma şekilli nesnelerin, üç boyutlu kutuların koordinatlarına yerleştirmesi ile gerçekleştirilir.Bu noktada amaç,asgari sayıda kutu kullanmak suretiyle, tüm dikdörtgen prizma şekilli nesnelerin kutulara yerleştirilmesidir[21].




Şekil 2.3 Üç Boyutlu Kutulama Problemi

Şekil 2.3 üç boyutlu nesnelerin üç boytlu kutulara yerleştirilmesini göstermektedir. Yerleştirme gerçekleştirilirken kutuların ve nesnelerin her bir eksendeki koordinatlarına yerleştirme yapılır.

## 2.2 Bir boyutlu kutulama problemleri

1BKP problemi, belirli bir hat veya çizgi içerisine belirli kapasitelerdeki daha küçük çizgilerin yerleştirilmesidir[6]. 1BKP pratikte bellek atama ve iş çizelgeleme problemlerinde uygulama alanı bulmaktadır.

### 2.2.1 Bir Boyutlu Kutulama Modeli

Bu bölümde, 1BKP probleminin genel matematiksel yapısını açıklanmaktadır:

1BKP problemi, $c$ her bir kutunun kapasitesini göstermek üzere, $n$ adet sırasıyla $\{w_1, w_2, \cdots, w_n\}$ özelliklerine sahip nesnenin kutulara atanmasını içerir. 1BKP modeli, kullanılan kutu sayısının enazlanması amacıyla, her bir kutu için $c$kapasitesini aşmama kısıtını göz önünde bulundurarak, her bir nesnenin bir kutuya atanmasını sağlar.

$$Min\ Z = \sum_{i=1}^{n} Y_i \tag{2.1}$$

$$şka. \quad \sum_{j=1}^{n} w_j X_{ij} \leq cY_i \qquad i \in N = \{1, \dots, n\}. \tag{2.2}$$

$$\sum_{i=1}^{n} X_{ij} = 1 \qquad j \in N. \tag{2.3}$$

$$Y_i \in \{0,1\} \qquad i \in N. \tag{2.4}$$

$$X_{ij} \in \{0,1\} \qquad j \in N, i \in N. \tag{2.5}$$

1BKP'nin matematiksel modelinde,$Y_i i$ kutusunun kullanılması durumunda 1 değerini, aksi halde 0 değerini alan ikili değişkenleri, $X_{ij} j$ nesnesinin $i$ kutusuna atanması durumunda1 değerini, aksi halde 0 değerini alan ikili değişkenleri temsil etmek üzere, denklem (2.1) kullanılan kutu sayısını enazlama amacını, denklem (2.2) bir kutuya atanan nesnelerin o kutunun hacmini aşmamasını, denklem (2.3) her bir nesnenin yalnızca bir kutuya atanabileceğini ifade etmektedir. Denklem (2.4) ve denklem (2.5) ilgili değişkenlerin (0,1) tamsayı değerleri alabileceğini göstermektedir.

### 2.2.2 Bir Boyutlu Kutulama Problemleri Algoritmaları

1BKP algoritmaları çevrimiçi ve çevrimdışı olmak üzere iki sınıfa ayrılır. Çevrimiçi algoritmalar, nesnelerin geliş sırasına göre paketlenir. Dolayısıyla, herhangi bir nesnenin bir kutuya ataması, tüm nesnelerin içinde bulunduğu bir $\mathcal{R}$ kümesi hakkında bilgi sahibi olmaksızın gerçekleştirilir. Çevrimiçi algoritmalar, paketlenecek nesnelerin sırayla geldiği ve ulaştığı anda yerleştirilmesi gerektiği durumlarda kullanılır. Çevrimdışı algoritmalar ise, yerleştirmeişlemi boyunca, tüm nesnelerin içinde bulunduğu bir $\mathcal{R}$ kümesi hakkında bilgi sahibi olunan durumlarda kullanılır[6]. Çevrimdışı algoritmalar, $\mathcal{R}$ kümesinin tamamıyla yerleştirmeişlemi gerçekleşmeden bilinmesi sebebiyle, çevrimiçi algoritmalara göre kullanılan kutu sayısı ve yerleştirme performansı açısından yüksek avantaj oluşturur.

Kutulama problemi NP-zor problem olmasına rağmen, büyük boyuttaki problemler için ileri düzeydeki algoritmalarla optimal çözümlere ulaşılabilmektedir[22][23]. Bu problemlerin çözümü için temelde kullanılan dört algoritma bulunmaktadır.

- ***İlk uygun yeralgoritması (First Fit -FF)*:** Başlangıçta tüm kutular boştur. Nesneler; geliş sıralarına göre, kutuların kalan kapasitesini kontrol etmek koşulu ile yerleştirilir. Yeterli kapasite bulamayan her bir nesne için yeni bir kutu açılmak suretiyle, tüm nesnelerin kutulara yerleştirilmesi sağlanır.

  İlk uygun yer algoritmasının çevrimdışı haline ise, ilk uygun azalan yer algoritması denir. İlk uygun yer algoritmasından farkı, yerleştirilecek olan nesnelerden oluşan $\mathcal{R}$kümesinin tamamıyla paketleme işlemi gerçekleşmeden bilinmesi dolayısıyla, ilgili özelliklerine göre (ağırlık, hacim, vd.) büyükten küçüğe doğru sıralanan nesneler, ilk uygun yer algoritmasına göre yerleştirilir.

- ***En Yakın Yer Algoritması(Next Fit-NF)*:** Başlangıçta tamamen boş olan kutulara yerleştirilecek nesneleri rasgele bir sırada işler. Her bir kutu için, yeterli kapasite olması durumu kontrol edilir. Eğer yeterli kapasite varsa, nesne o kutuya yerleşir. Aksi halde, bu nesne atlanarak sıradaki diğer nesneye geçilir. Yeni bir kutunun açılması için ancak daha önce açılan kutuların kapasitelerinin tamamen tüketilmesi veya kalan kapasiteye yüklenecek herhangi bir nesne kalmamış olması gerekir. Tüm nesneler kutulara yerleştirilinceye kadar bu prosedüre devam edilir.

  En yakın yer algoritmasının çevrimdışı hali olan azalan en yakın yer algoritması, $\mathcal{R}$kümesinin bilinmesi sebebiyle, nesnelerin ilgili özelliklerine göre (ağırlık, hacim, vd.) büyükten küçüğe doğru nesneleri sıralar ve en yakın yer algoritmasının prosedürüne göre yerleştirme gerçekleştirir.

- ***En Uygun Yer algoritması(Best Fit):*** Başlangıçta tamamen boş olan kutulara yerleştirilecek nesneleri geldikleri sıra ile işler. Her bir kutu için, nesnenin ilgili kutuya yerleştirildikten sonraki kalan kapasitenin en az kalması göz önünde bulundurulur. İlk uygun yer algoritmasına benzer şekilde, yeni bir kutunun açılması için ancak daha önce açılmış olan kutuların kapasitelerinin tamamen tüketilmesi veya kalan kapasiteye yüklenecek herhangi bir nesne kalmamış olması gerekir. Bu işlemler, tüm

nesneler kutulara yerleştirilinceye kadar devam eder. Diğer sezgisel yöntemlere benzer şekilde, çevrimdışı hali için nesnelerin büyükten küçüğe sıralanarak yerleştirilmenin gerçekleştirildiği azalan en uygun yer algoritması kullanılmaktadır.

- ***En Kötü Uygun Yer Algoritması(Worst Fit-WF)*:** WF sezgiseli en uygun yer algoritmasının aksine, nesneleri kutularda kalan maksimum hacimlerine göre atar. Bu algoritmanın da çevrimdışı versiyonu bulunmaktadır.

Kutulama problemleri algoritmalarında kullanılmak üzere, sonuçların en iyi ya da en kötü durumlarını incelemek için genel sınırlar kullanılmaktadır. Alt sınır bu amaçla kullanılan göstergelerden biri olup, optimal sonuç bilinmiyorsa, algoritmanın performansının tahmin edilmesine yardımcı olur. Bu nedenle, böyle belirsizlik durumlarında, kutulama problemlerinde minimum kutu sayısını bulmak için alt sınır hesaplanır. Alt sınırı bir başlangıç çözüm olarak kullanmanın yanı sıra, çözümde elde edilen kutu sayısının alt sınır değeri ile eşit çıkması durumunda optimuma ulaşıldığı kabul edilir. Eğer kutulama probleminde amaç fonksiyonu fayda/kar eniyilenmesi ise, üst sınır hesaplanarak, benzer süreç devam ettirilir [24].

Kötü durum performansı, bir algoritmanın en kötü durumlarda verdiği tepkileri ölçmek için kullanılmaktadır. Bu durumda, elde edilen sonuç "$OPT(I)$" ile karşılaştırılmak suretiyle algoritmaların verdiği sonuçlar değerlendirilmektedir.1BKP problemlerinin çözümlerinde kullanılan algoritmaların verdiği sonuçların değerlendirilmesi için Çizelge 2.1 verileri kullanılmaktadır[22].

Çizelge 2.1 Sezgisel Algoritmaların Karşılaştırılması [22]

| Sezgisel Algoritmalar | Zaman Karmaşıklığı | En Kötü Oran |
|---|---|---|
| *İlk Uygun Yer Algoritması* | "O(n)" | "2 * $OPT(I)$" |
| *En Uygun Yer Algoritması* | "O(n * log(n))" | "17 /10 * $OPT(I)$" |
| *En Kötü Yer algoritması* | "O(n *log(n))" | "17/ 10 * $OPT(I)$" |

## 2.3 İki Boyutlu Kutulama Problemleri

İki boyutlu kutulama problemlerinde 1BKP'den farklı olarak nesnelerin iki farklı özelliği (genişlik, uzunluk, derinlik, ağırlık hacim vd.) göz önünde bulundurulmak suretiyle, kutulara yerleştirme yapılır.

### 2.3.1 İki Boyutlu Kutulama Probleminin Yapısı

Her bir elemanı $w_i$ ve $h_i$ boyutlarına sahip dikdörtgensel $i$ nesnelerinden oluşan $n$ elemanlı $\mathcal{R}$ kümesinde yer alan tüm nesnelerin en az sayıda kutu kullanmak suretiyle kutulara yerleştirilmesi problemidir [25].

İki boyutlu kutulama problemlerini modellemeye yönelik ilk girişim, Gilmore ve Gomory [26] tarafından bir boyutlu kutulama algoritmalarından esinlenerek geliştirmiştir. Tek bir bölmeye doldurulabilen tüm alt grupların numaralandırılmasına dayanan bir sütun oluşturma yaklaşımı önermiştir [27]. Chung vd. [28]iki boyutlu kutulama problemlerinin çözümü için Sonraki Sığanların Azalan (NFDH) Yükseklik (Next Fit Decreasing Height –NFDH) algoritmasının[22] geliştirilmek suretiylehibrit bir algoritma önermiştir. Berkey vd.[29] iki boyutlu kutulama problemlerinin çözümü içinsonlu en iyi şerit (Finite Best-Strip), sonlu ilk şerit (Finite First Fit), sonlu dip-sol (Finite Bottom left) ve sonlu en yakın sığan (Finite Next Fit) sezgisel algoritmalarını geliştirmiştir. Lodi vd. [30] iki boyutlu kutulama problemlerinin matematiksel modellerinin alt sınırlarını[22], klasik yakınsama algoritmalarını[29], yeni buluşsal yöntemlerini[6], metasezgisel yöntemleri[31] ve tam sayılımodelleme yaklaşımlarını[32] incelemiştir.

### 2.3.2 İki Boyutlu Kutulama Probleminin Modeli

$\mathcal{R} = \{1, \dots, \mathrm{n}\}$ kümesi, $w_i$ ve $h_i$boyutlarında sahip kutulanacak $i$ dikdörtgensel nesneleri ifade etmek üzere, $W$ ve$H$ ebatlı kutulara yerleştirilmesi için, Pisinger ve Sigurd [25] iki boyutlu kutulama modeli aşağıdaki gibi ifade etmektedir:

$$Min \quad v \tag{2.6}$$

$$şka. \quad l_{ij} + l_{ji} + b_{ij} + b_{ji} + p_{ij} + p_{ji} \geq 1 \qquad i,j \in \mathcal{R}, i < j. \tag{2.7}$$

$$x_i - x_j + W l_{ij} \leq W - w_i \qquad i,j \in \mathcal{R}. \tag{2.8}$$

$$y_i - y_j + H b_{ij} \leq H - h_i \qquad i,j \in \mathcal{R}. \tag{2.9}$$

$$m_i - m_j + n p_{ij} \leq n - 1 \qquad i,j \in \mathcal{R}. \tag{2.10}$$

$$0 \leq x_i \leq W - w_i \qquad i \in \mathcal{R}. \tag{2.11}$$

$$0 \leq y_i \leq H - h_i \qquad i \in \mathcal{R}. \tag{2.12}$$

$$1 \leq m_i \leq v \qquad i \in \mathcal{R}. \tag{2.13}$$

$$l_{ij}, b_{ij}, p_{ij} \in \{0,1\} \qquad i,j \in \mathcal{R}. \tag{2.14}$$

$$x_i, y_i \in \mathbb{R} \qquad i \in \mathcal{R}. \tag{2.15}$$

$$m_i, v \in \mathbb{N} \qquad i \in \mathcal{R}. \tag{2.16}$$

2BKP probleminin matematiksel modelinde; $l_{ij}$, $i$ nesnesinin $j$ nesnesinin soluna yerleştirilmesi durumunda 1 değerini, aksi halde 0 değerini alan ikili değişkeni; $b_{ij}$, $i$ nesnesinin $j$ nesnesinin altına yerleştirilmesi durumunda 1 değerini, aksi halde 0 değerini alan ikili değişkeni göstermektedir. Modelde, $(x_i, y_i), i$ nesnesinin yerleştirildiği $m_i$ kutusunun sol-alt koordinatlarını, $v$kullanılan kutu sayısını göstermektedir ve $p_{ij}$ ise $m_i < m_j$ durumunda 1 değerini alan ikili değişkendir. Bu modele göre, iki nesnenin çakışmaması için aşağıdaki eşitsizliklerin sağlanması gerekmektedir.

$$l_{ij} + l_{ji} + b_{ij} + b_{ji} + p_{ij} + p_{ji} \geq 1 \tag{2.17}$$

$$l_{ij} = 1 \Longrightarrow x_i + w_i \leq x_j \tag{2.18}$$

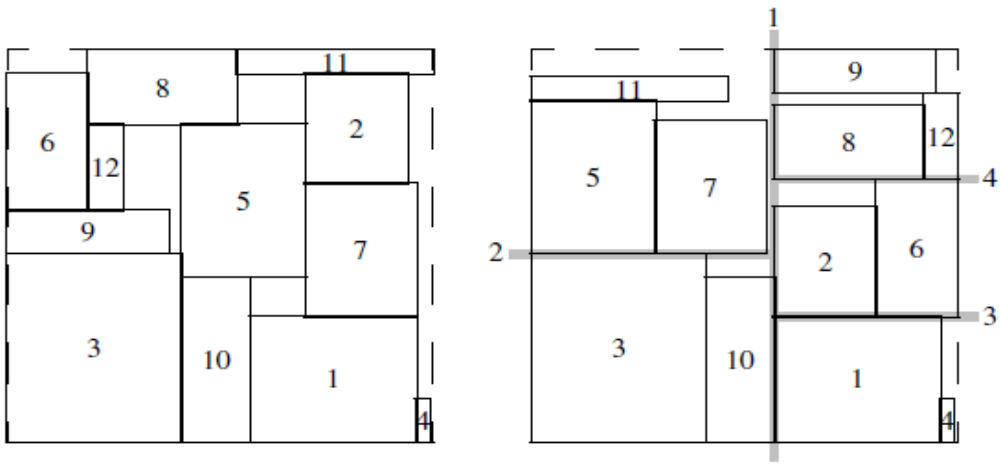
$$b_{ij} = 1 \Longrightarrow y_i + h_i \leq y_j \tag{2.19}$$

$$p_{ij} = 1 \Longrightarrow m_i + 1 \leq m_j \tag{2.20}$$

### 2.3.3 İki Boyutlu Kutulama Problemi Algoritmaları

İki boyutlu kutulama problemleri tek boyutlu kutulama probleminin genişletilmiş halidir ve Kes-Stokla kategorisine girmektedir [30]. Nesnelerin yönlendirilmesi ve giyotin kesme (nesneler, bölmenin kenarlarına paralel olacak şekilde, bir sıra halinde ve kenarlara bitişik kesiklerle ile elde edilmelidir. Şekil 2.4 Giyotinli kesme ve giyotinsiz kesme örneklerini göstermektedir. Lodi [30]'ye göre, kes-stokla problemleri dört sınıfta incelenir. Bunlar,

- 2BKP / O / G: Nesneler yönlendirilmiş (O) ve giyotin kesme (G) gereklidir;
- 2BKP / R / G: Nesneler 90 '(R) oranında döndürülebilir ve giyotin kesme gereklidir;
- 2BKP / O / F: Nesneler yönlendirilir ve kesim serbesttir (F);
- 2BKP / R / F: Nesneler 90 ° döndürülebilir ve kesim serbesttir.

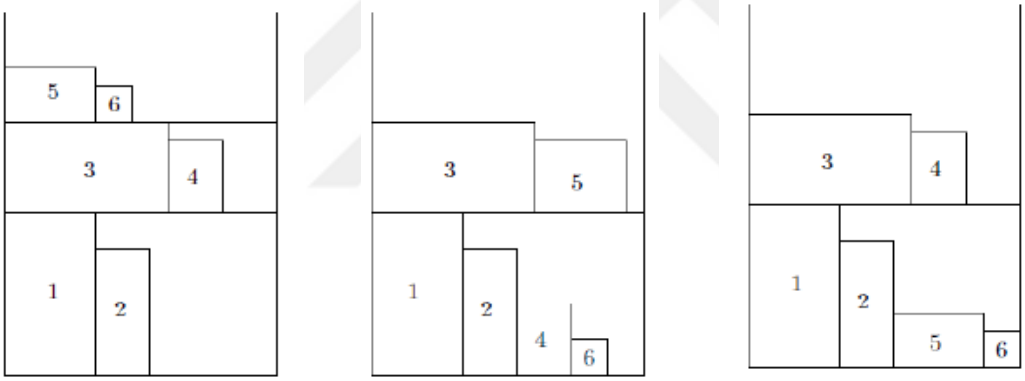



Şekil 2.3 Giyontinsiz Örnek ve Giyotinli Örnek [25].

İki boyutlu açgözlü kutulama algoritmaları, iki şekilde sınıflandırılabilir [29]; (1) nesnelerin sonlu kutulara doğrudan paketlendiği faz algoritmaları, (2) nesnelerin bir şerit halinde paketlendiği iki aşamalı şerit algoritmaları.

Şerit yerleştirmeile ilgili temel yaklaşım, nesnelerin sözderaflara paketlenmesini içerir. Bu durum, yerleştirilen nesnelerin yatay kısıtlamaları ile tanımlanan ve birbirleri üzerine yerleştirilen seviyeler oluşturduğu anlamına gelir. Şerit paketlemesi için önerilen algoritmalar, Bölüm 1.1.2'de açıklanan tek boyutlu kutulama algoritmalarından esinlenmektedir. Her durumda, nesneler başlangıçta artan yükseklik derecesine göre sıralanır ve ilgili rafta paketlenir. $j$ ilgili nesneyi ve $k$son oluşturulan rafı belirtmek üzere [30];

- ***Birinci Sığan Azalan Yükseklik (FFDH) stratejisi:*** $j$ nesnesi, sola hizalanarak paketlenir. Eğer $j$ nesnesi herhangi bir rafa yerleştirilemiyorsa, NFDH da olduğu gibi yeni raf oluşturulur.
- ***Sonraki Sığanların Azalan (NFDH) Yükseklik stratejisi***: $j$ nesnesi rafların üzerinde sola hizalanarak sığana kadar paketlenmesini içerir. Rafın dolması durumunda, aynı kutu içerisinde yeni bir raf oluşturulur ve yerleştirilecek olan nesneler sağa hizalanarak paketlenir. NFDH stratejisi FFDH'e kural olarak benzerdir. Ancak bu stratejide FFDH'den farklı olarak en son yerleşen nesnenin yeri hafızaya alınır ve bundan sonra gelen nesne kutuda yer olduğu sürece son nesnenin yanına yerleşmeye çalışır.
- ***En Uygun Azalan Yükseklik (BFDH) stratejisi:*** $j$ nesnesi sağa hizalanarak paketlenir. Kullanılmayan yatay alanlardan minimum olan rafa nesneler yerleştirilir. Herhangi bir rafa yeni gelen nesne yerleştirilemezse, yeni bir raf NFDH'de olduğu gibi tekrar oluşturulur.




NFDH FFDH BFDH

Şekil 2.4 RafYerleştirme Stratejileri[30]

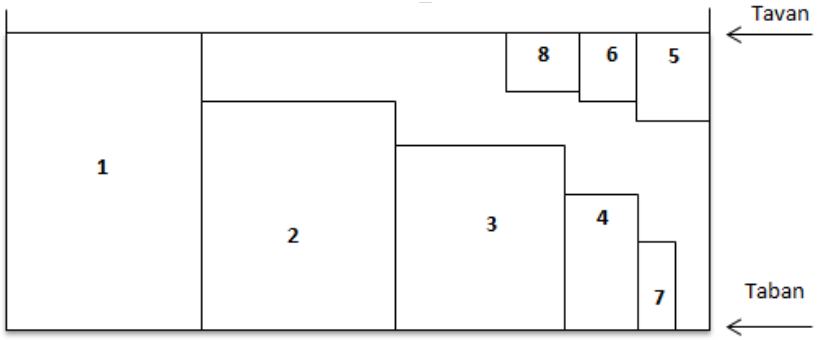
$OPT(l)$ mevcut şeridin mümkün olan en düşük yüksekliğini temsil etmek üzere, şerit paketlemedeki karmaşıklık ve en kötü durum performans oranı (APR)Çizelge 2.2'de gösterilmektedir [6].

Çizelge 2.2 Şerit Yerleştirme Algoritmaları[6]

| *Sezgisel Algoritmalar* | *Zaman Karmaşıklığı* | *En Kötü Performans Değerleri* |
|---|---|---|
| *SSA* | $\boldsymbol{O(n.\log(n))}$ | $\boldsymbol{2.OPT(l)+1}$ |
| *SSA* | $\boldsymbol{O(n.\log(n))}$ | $\boldsymbol{\frac{17}{10}.OPT(l)+1}$ |
| *SUAY* | $\boldsymbol{O(n.\log(n))}$ | $\boldsymbol{\frac{17}{10}.OPT(l)+1}$ |

Rafların oluşturulmasında kullanılan diğer yaklaşımlar şunlardır:

- Nesneler önce tabanda sonra tavandan ters olarak raflaraŞekil 2.5 gibi yerleşir.




Şekil 2.5 Raf Yerleştirme Stratejileri [30]

- Herbir nesnenin kârı, kullandığı alan olarak tanımlandığı durumda sırt çantası alt problemi bulunur.Bu bağlamda ilk olaraken uzun nesne seçilir ve bir raf oluşturulur; alt problemin amacı ise sırt çantası probleminin sonucuna göre rafa başka nesneler eklemektir.Sırt çantası problemi de NP zor problem sınıfına ait olduğu için, polinom zaman karmaşıklığının dışında kalma problemi oluşabilir, ancak alt problem genellikle çok küçük olduğundançok hızlı bir şekilde çözülebilmektedir[30].

Raflar oluşturulduktan sonra, nesnelerkutulara yerleştirilmektir. En temel iki fazlı algoritmalar, kutuları raflara yerleştirebilmek için 1BKPsezgiselleri kullanmaktadır. En yaygın kullanılan iki fazlı algoritmaları aşağıda listelenmektir.

- **Hibrid İlk Sıra Yerleşme (Hybrid First Fit- HFF)**: Choll[28] raflar oluşturulurken FFDH yöntemi kullanır ve1BKP yaklaşımı olarak FFD sezgiselineyaşvurulur. APR'nin sonucu ise 17 /8*OPT (I) + 5 'dir.

- **Sonlu En İyi Şerit Yerleşme (Finite Best Strip-FBS):**Araya ve Riff[33]bu algoritmada rafları oluştururken BFDH yöntemini ve1BKP yaklaşımı olarak en uygun yerleştirme yönteminin kullanımı önermiştir.

Raftabanlı algoritmaların bir özelliği giyotin kesilebilirliği olmasıdır ve 2BKP/*/G problemindeki gibi çözülebilir[30].

Tek fazlı algoritmalarda ise kutular doğrudan ve raflardan oluşturulmaktadır. Nesneler bu raflara yerleştirilirken iki fazlı algoritmaların aksine bu raflar kutuların içinde sürekli olarak oluşturulur. Bununla beraber, tek fazlı algoritmalar genellikle iki fazlı algoritmalardan daha kötü performans göstermektedir. İkinci tip denilenrafsız algoritmalar, nesneleri kutuların herhangi her bir alanına yerleştirmektedir.




Şekil 2.6 Alternatif Yönergeler Algoritması [30]

Bu yöntemlerden biri de Alternatif Yönler Algoritması'dır. Bualgoritma, alt sınır olan $L$ kutularının altlarına nesnelerin bir alt kümesini paketleyerek, azalan en uygun yer politikasını uygulamaktadır. Şekil 2.6 bu algoritmaya göre yerleştirilen kutuları göstermektedir.

#### 2.3.3.1 İki Boyutlu Raf Yerleştirme

Bu algoritmalar, yerleştirmeişlemini ve boşluğun boş alanını temsil etmek için kullanılan temel veri yapısına göre gruplar halinde sınıflandırılmaktadır.

Raf Algoritmaları, genişlik $W_b$ ve yükseklik $H_s$ olan bir bölmenin alt-dikdörtgen olabilmesi için raf tanımlanır. Yerleştirme devam ederken, kutunun boş alanı, nesnelerin soldan sağa yerleştirildiği, aşağıdan yukarıya doğru raflar halinde düzenlenir. Son raf (en üst raf) açık raf olarak adlandırılır. Dikdörtgenler aşağıdan yukarıya yerleştirildiğinden açık rafın üzerindeki alan her zaman kullanılmaz. Bu, rafın üzerine bir nesne yerleştirildiğinde açık rafın

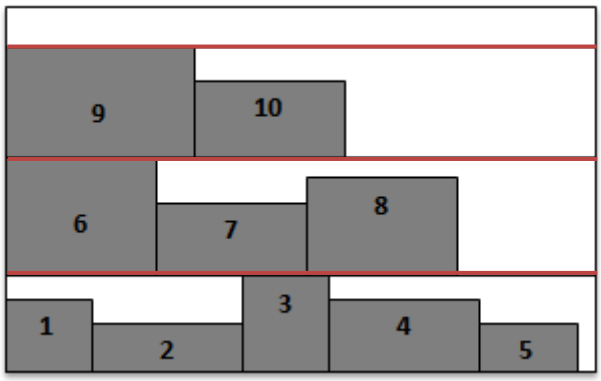
yüksekliğinin ayarlanmasını sağlar. Açık rafın altındaki raflarbu özgürlüğe sahip olmadığıiçin bunlara kapalı raflar denmektedir.

Rafın üzerine bir dikdörtgen ($w; h$) yerleştirirken dikdörtgenin döndürülüp döndürülmemesi, yani dikey açının dik olarak depolanıp depolanmadığına karar vermemiz gerekir $(store\ (min\ (w; h\ );maks\ (w; h))$ veya yanlara doğru $(store\ (max\ (w; h); min\ (w; h))$). Uygulamanın tüm varyantlarında, seçim şu sırada yapılır:

***1.Adım:*** Dikdörtgen yeni açık raftaki ilk dikdörtgense, yanlara doğru yerleştirilir. Bu, yeni rafın yüksekliğini en aza indirgemek içindir.

***2.Adım:*** Dikdörtgen dik ise, yani max ($w; h$) $<h_s$ ise, o zaman yerleştirilmeye uygundur. Bu, dikdörtgen üst kenar ile rafın tavanı arasındaki boşa harcanmış yüzey alanını en aza indirmeyi hedeflemektedir.

***3.Adım:*** Aksi takdirde dikdörtgeni mümkün olduğu şekilde yanlamasına yerleştirin.




Şekil 2.7 Raf Yerleştirme

Şekil 2.7, Raf Yerleştirme Algoritması tarafından üretilen örnek bir ambalajı göstermektedir. Dikdörtgenler, kutuya yerleştirildikleri sıra ile numaralandırılmış ve kırmızı çizgiler raf tavanlarını göstermektedir. Raf Algoritmasının tüm varyantları, bu resimde gösterilen paketlemelere çok benzer şekilde üretilir.

- ***Raf yerleşimde Sonraki Sığanların Azalan Yükseklik (S-NFD) Stratejisi*** *:* Bu tezde sunulan tüm algoritmalar ve varyantları arasından, Shelf Next Fit paketleri yerleştirmek için kesinlikle en basit yöntemdir(Şekil 2.5).Bu uygulamanın, başka algoritmaların gözden geçirmediğiyalnızca sabit bir miktarda iş belleği gerektiren özel

bir durumu vardır. Görüleceği üzere diğer algoritmalar, hali hazırda paketlenmiş olan dikdörtgenlerin sayısıyla doğrusal olan en azından bir çeşit veri yapısı kullanmaktadır. S-NF için yalnızca üç geçici kayıt gerekir. Bu özellik bazı uygulamalarda yararlı olabilse de ne yazık ki S-NF tarafından üretilen paketler, tezde sunulan en iyi yöntemlerden oldukça uzak olabilir.

- ***Raf yerleşimde İlk Sığanların Azalan Yükseklik (S-FFD) Stratejisi:*** Yeni bir raf açıldığında eski raflarda kalan boş alanın unutulması boşa giden bir durumdur. Bu nedenle, SHELF-NF algoritmasının tüm varyantları, daha önce kapalı rafların tümünü bir liste halinde paketler ve böylece mümkün olan kutulara yerleştirebilir. Ancak kutularda açılan raflardan birden fazla mevcutsa hangisini seçmeliyiz diye sorarsak bu seçimçeşitli şekillerde olabilir. Birinci aşamadanesnelerrafın içine daima en düşük indeksle yerleştirilir. Bu oldukça basit bir yöntemdir ancak algoritmanın hem çalışma süresi hem de bellek tüketimi, geçerli bölmede bulunan rafların sayısına göre doğrusaldır.

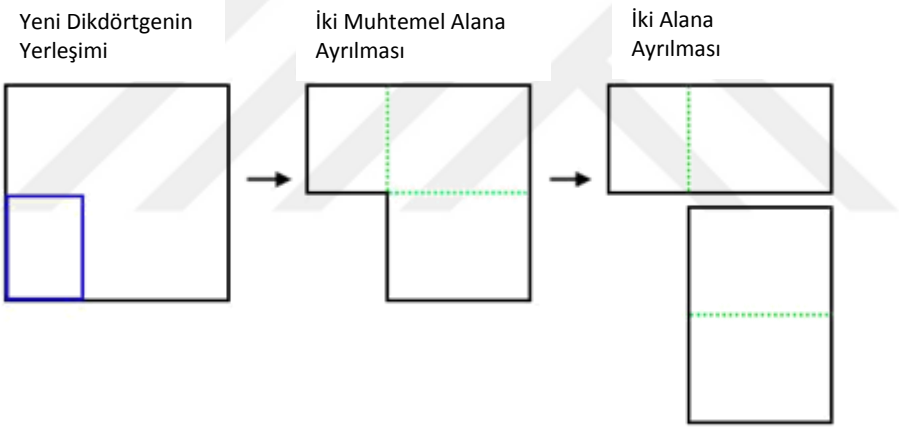
  S-FF kapalı bir rafta yönettiğimiz bir kutudaki alanı açık bir rafa koymamıza yardımcı olur. Dolayısıyla, S-FF'in S-NF'den daha iyi sonuç verdiği düşünülebilir, ancak bu doğru değildir. Bunun nedeni, yerleştirme kararı sezgiseldir ve S-FF'in daha optimum yerleştirdiğini söyleyemeyiz. Uygulamada S-FF SHELF-NF'den daha iyi performans göstermesine rağmen bazı durumlarda S-FF, S-NF'den daha iyi performans göstermeye çalışırken kötü sonuçlar elde edilebilir ve daha kötü bir yerleştirme oluşabilir. Bu, diğer algoritmaları karşılaştırırken tekrar eden bir işlemdir. S-FF algoritması ayrıca, O (*n* log *n*) zaman ve O (*n*) uzayında çalıştırılacak şekilde uygulanabilir [30].

- ***Rafyerleşimde En Uygun Azalan Yükseklik (S-BFD) stratejisi:*** S-FF'nin farklı bir versiyonu olarak görülebilir; olası tüm rafları bütün olarak düşünmeden, nesneler ilk rafın içine yerleştirilir. Algoritma, mümkün olan tüm raflara bakar ve yalnızca onlardan en uygun olanını bularak nesneleri yerleştirmeye devam eder. S-BFD'de, raf alanının kalan uzunluğunu en aza indirgeyen rafı seçme kuralı uygulanır. Bu

uygulamayı en uygun genişlik (En Uygun Azalan Genişlik (EUAG) stratejisi) ve en (En Uygun Azalan En (EUAE) stratejisi) olarak da çoğaltabiliriz.

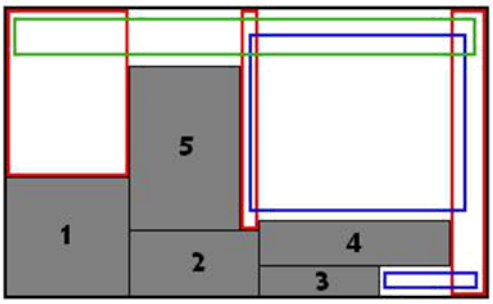
#### 2.3.3.2 İki Boyutlu En Büyük Dikdörtgensel Alan Algoritması

Önceki bölümde tanıtılan Giyotin algoritması, raf yerleştirme algoritmasının üzerinde yapılan bir iyileştirmedir. Ancak bölünmüş giyotin kesme sınırları pratik performans ile ilgili sorunlara neden olmaktadır. Bütün bu sorunları tamamen ortadan kaldırmak için Jylänki vd. [34]En Büyük Dikdörtgensel Alan Algoritmasını geliştirmişlerdir. Bu algoritma, Giyotin Bölme Yerleştirme kuralı uzantısına dayalıdır. Giyotin Kesme Algoritması gibi, Maksimum Dikdörtgen Algoritması, bölmenin serbest alanını temsil eden boş dikdörtgenlerin listesini depolar. Ancak, iki bölünmüş eksenden birini seçen Giyotin Algoritmasından farklı olarak, Maksimum Dikdörtgen algoritması, her iki bölme eksenini aynı anda toplayan bir işlemi gerçekleştirir.




Şekil 2.8 MAXRECT Veri Seti İçin Yerleşme Kuralı[30]

- **Sol Alt Maksimum Diktörtgensel Alan Algoritması (MAXRECTS-BL):** Alt Sol algoritması veya Tetris algoritması olarak anılmaktadır. Bu algoritma tarafından kullanılan sezgisel kural basittir. Her nesne, dikdörtgenin üst kenarındaki y-koordinatının en küçük olduğu konuma yönlendirilir ve yerleştirilir. Eğer böyle birkaç geçerli konum varsa, en küçük X-koordinat değeri olan seçilir. Bu algoritmayı uygulamak için en büyük dikdörtgen (maximal rectangles) veri yapısı kullanılır[35].




Şekil 2.9 MAXRECTS-BL Algoritmasıyla Üretilen Maksimum Dikdörtgenler [30]

- **Maksimum Dikdörtgen En İyi Alan Uyumu (MAXRECTS-BAF)**: Bu algoritma da Giyotin algoritmasında olduğu gibi M veri yapısında bulunan rasgele nesneleri seçerken aynı buluşsal kuralları kullanmaktadır. Maksimum Dikdörtgen En İyi Alan kuralında,sonraki dikdörtgen R'yi yerleştirmek için kutu içinde kalan en küçük alan seçilmelidir. Bu bölgede bir düğüm varsa, bu bağı koparmak için En İyi Kısa Taraf Eşleşmesi kuralı kullanılmaktadır[30].
- **En çok Dikdörtgen En İyi Kısa Taraf Eşleşmesi (MAXRECTS-BSSF):** Giyotin-BSSF'de olduğu gibi, Maksimum Dikdörtgen En İyi Kısa Taraf Eşleşmesi kuralında dikdörtgen R'yi yerleştirmek için kutuda kalan en kısa kenarlı bölgeyi seçmektedir. Başka bir deyişle, kalan tarafın uzunluğunu en aza indirmeyi hedefleyen bir algoritmadır[30].

## 2.4 Üç Boyutlu Kutulama Problemi

Gerçek hayattaki problemlere yakın olduğu için problemin yapısı karmaşıktır. 3BKP'inde paketler genellikle homojen, dış bükey olup, paketlerin boyutları bellidir.

### 2.4.1 Üç Boyutlu Kutulama Probleminin Yapısı

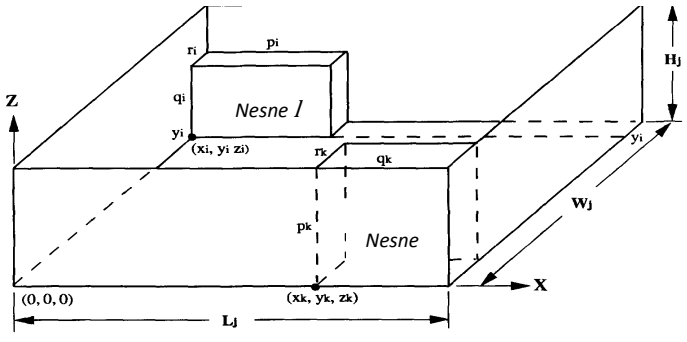
3BKPP boyutları $(w_i, h_i, l_i)$ olannesnelerin, boyutları$(W, H, L)$olan $n$ adet dikdörtgen prizma şeklindeki kutulara mümkün olan en az kutuyu kullanacak şekilde yerleşmesini amaçlar. Giyotin kesme kısıtlaması 2BKPP'de olduğu gibi yaygın değildir.3BKPP sınıflarını belirlerken, yönlendirme kısıtlaması olan3BKPP / O veya 3BP / F kuralları dikkate alınmaktadır. Literatürde 3BKPP / Oprobleminin çözümü için kullanılan sezgisel algoritmalar genellikle 2BKPP için kullanılan algoritmalardanesinlenmektedir. 3BKPP çözümü için, rafları ve 1BKPP

altproblemlerinin çözümünü içerenkesin algoritmalar mevcuttur[21]. Bu algoritmalar, iki seviyeden oluşur: (1)öncelikle nesneler, gerçek konumlarını belirtmeden kutulara atanır; (2) nesnelerin alt kümesinin tek bir bölmenin içine yerleştirilip yerleştirilemeyeceğini test etmek için özel bir algoritma kullanılır.

Martello ve Vigo [21] tarafından önerilen 3BKP algoritması 2BKP probleminin çözümü için sunulan iki seviyeli ayrışma ilkesine dayanmaktadır. Temel bir dallanma ağacı, gerçek konumlarını belirtmedennesneleri kutulara atarken aynı zamanda belirli karar düğümlerinde, nesnelerin bir alt kümesi tek bir bölme içine yerleştirilip yerleştirilmeyeceğini test etmek ve bu kutuyu belirlemek için kullanılır. Cevabın doğrulanması halinde nesneler yerleştirilir.

### 2.4.2 Üç BoyutluKutulama Probleminin Modeli

Üç Boyutlu kutulama problemlerinin algoritmaları, birçok araştırmacı tarafından konteyner yükleme problemlerinde uygulanmıştır. 3BKP,$m$ adet $\mathcal{R}$ kümesinin elemanı olan $(w_i, h_i, l_i)$ ebatındaki $i$ nesnelerinin, $(W, H, L)$ ebatındakikutulara yerleşmesi problemidir. Chen vd. [20], birden fazla konteynerin kullanılabileceği genel konteyner yükleme problemini bir 0-1 karma tamsayılı programlama modeli geliştirmiştir. Li vd. [15], çoklu konteyner yükleme problemlerinin çözümüne yönelik bir hibrid değişken algoritması geliştirmişlerdir. Şekil 2.11'de verilen$N = \{i | i = 1, .., n\}$adet nesneninyerleştirildiği koordinatları $(x_i, y_i, z_i)$ile gösterilmek üzere, $m$adet konteynereyerleştirilmesi problemi için:




*Konteyner j*

Şekil 2.10 Üç Boyutlu Kutu Kordinatları

$$Min \quad \sum_{j=1}^{m} L_j.W_j.H_j.n_j - \sum_{i=1}^{N} p_i.q_i.r_{.i} \tag{2.21}$$

şka. $\forall i \in M, \forall j$

$$x_i + p_i.l_{ix} + q_i.(l_{zi} - w_{yi} + \mathrm{h}_{zi}) + r_i.(l \quad - l_{xi} + l_{zi} + w_{yi} - \mathrm{h}_{zi}) \leq x_k + (1 - x_{ik}^{-}).M$$

$$\forall i \in M, \forall j \in N. \tag{2.22}$$

$$x_k + p_k.l_{ik} + q_k.(l_{zk} - w_{yk} + \mathrm{h}_{zk}) + r_k.(1 - l_{xk} + l_{zk} + w_{yk} - \mathrm{h}_{zk}) \leq x_i + (1 + x_{ik}^{+}).M$$

$$\forall i \in M, \forall j \in N. \tag{2.23}$$

$$y_i + q_i.w_{yi} + r_i.(l_{xi} + l_{zi} - w_{yi}) + p_i.(1 - l_{xi} + l_{zi}) \leq y_k + (1 - y_{ik}^{+}).M$$

$$\forall i, k \in \mathrm{M}, i < k \tag{2.24}$$

$$y_k + q_k.w_{yk} + r_k.(l_{xk} + l_{zk} - w_{yk}) + p_k.(1 - l_{xk} + l_{zk}) \leq y_i + (1 - y_{ik}^{-}).M$$

$$\forall i, k \in \mathrm{M}, i < k \tag{2.25}$$

$$z_i + r_i.h_{zi} + q_i.(1 - l_{zi} + h_{zi}) + p_i.l_{zi} \leq z_k + (1 - z_{ik}^{+}).M \qquad \forall i, k \in \mathrm{M}, i < k \tag{2.26}$$

$$z_k + r_k.h_{zk} + q_k.(1 - l_{zk} + h_{zk}) + p_k.l_{zk} \leq z_k + (1 - z_{ik}^{-}).M \quad \forall i, k \in \mathrm{M}, i < k \tag{2.27}$$

$$x_{ik}^{-} + x_{ik}^{-}, + y_{ik}^{+} + y_{ik}^{-} + z_{ik}^{+} + z_{ik}^{-} \geq s_{ij} + s_{kj} - 1 \qquad \forall i, k, j \; i < k \tag{2.28}$$

$$\sum_{j=1}^{m} s_{ij} = 1 \tag{2.29}$$

$$\sum_{i=1}^{N} s_{ij} \leq M.n_j \tag{2.30}$$

$$x_i + p_i.l_{ix} + q_i.(l_{zi} - w_{yi} + \mathrm{h}_{zi}) + r_i.(1 - l_{xi} + l_{zi} + w_{yi} - \mathrm{h}_{zi}) \leq L_j + \left(1 - s_{ij}\right).M \tag{2.31}$$

$$\forall i, k \in \mathrm{M}, i < k$$

$$y_i + q_i.w_{yi} + r_i.(l_{xi} + l_{zi} - w_{yi}) + p_i.(1 - l_{xi} + l_{zi}) \leq W_j + (1 - y_{ik}^{+}).M \tag{2.32}$$

$$\forall i, k \in \mathrm{M}, i < k$$

$$z_i + r_i.h_{zi} + q_i.(1 - l_{zi} + h_{zi}) + p_i.l_{zi} \leq H_j + (1 - z_{ik}^{+}).M \tag{2.33}$$

$$p_{ij}, u_j, \; x_{ik}^{+}, \; x_{ik}^{-}, y_{ik}^{+}, y_{ik}^{-}, z_{ik}^{+}, z_{ik}^{-} \in \{0,1\} \forall i, k \in M, \forall j \in N \tag{2.34}$$

$$x_i, y_i, z_i \geq 0 \;\; \forall i \in M \tag{2.35}$$

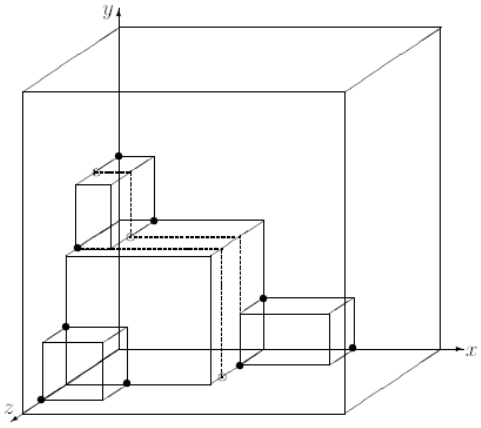
$s_{ij}$, $i$nesnesinin$j$ kutusuna yerleştirilmesi durumunda 1 değerini, aksi halde 0 değerini alan ikili değişkeni; $n_j$, j kutusunun kullanılması durumunda 1 değerini, aksi halde 0 değerini alan ikili değişkenini;$(x_i, y_i, z_i)$ nesnenin ön-sol-alt köşesinin koordinatlarını gösteren sürekli değişkeni; $(l_{xi}, l_{yi}, l_{zi})$i nesnesinin uzunluğunun sırasıyla X, Y ve Z eksenlerine paralel olması durumunu gösteren ikili değişkeni; $(w_{xi}, w_{yi}, w_{zi})$i nesnesinin genişliğinin sırasıyla X, Y ve Z eksenlerine paralel olması durumunu gösteren ikili değişkeni; $(h_{xi}, h_{yi}, h_{zi})$i nesnesinin yüksekliğinin sırasıyla X, Y ve Z eksenlerine paralel olması durumunu gösteren ikili değişkeni ve $(p_i, q_i, r_i)$i nesnesinin sırasıyla uzunluğu, genişliği ve yüksekliğini gösteren parametreyi;$(L_j, W_j, H_j)$j kutusunun sırasıyla uzunluğu, genişliği ve yüksekliğini gösteren parametreyiifade etmektedir. Ayrıca matematiksel modelde, nesnelerin birbirilerine göre konumlarını ifade etmek için; $x_{ik}^{-}i$ ve $k$ nesneleri aynı kutuya ve $i$ nesnesi $k$ nesnesinin soluna yerleştirilmesi durumunda 1 değerini, aksi halde 0 değerini alan ikili değişkeni; $x_{ik}^{+}i$ve $k$ nesneleri aynı kutuya ve $i$ nesnesi $k$ nesnesinin sağına yerleştirilmesi durumunda 1 değerini, aksi halde 0 değerini alan ikili değişkeni; $y_{ik}^{-}i$ ve $k$ nesneleri aynı kutuya ve $i$ nesnesi $k$ nesnesinin önüne yerleştirilmesi durumunda 1 değerini, aksi halde 0 değerini alan ikili değişkeni; $y_{ik}^{+}$i ve $k$ nesneleri aynı kutuya ve $i$ nesnesi $k$ nesnesinin arkasına yerleştirilmesi durumunda 1 değerini, aksi halde 0 değerini alan ikili değişkeni; $z_{ik}^{-}i$ ve $k$nesneleri aynı kutuya ve $i$ nesnesi $k$ nesnesinin üstüne yerleştirilmesi durumunda 1 değerini, aksi halde 0 değerini alan ikili değişkeni; $z_{ik}^{+}$i ve $k$ nesneleri aynı kutuya ve $i$ nesnesi $k$ nesnesinin altına yerleştirilmesi durumunda 1 değerini, aksi halde 0 değerini alan ikili değişkeni göstermektedir.

3BKP matematikselmodelinde,denklem (2.21) kullanılan tüm kutuların kullanılmayan boşluklarını enazlama amacını,(2.22)-(2.27) kısıtları herhangi iki nesnenin $i$ ve $k$'nin birbiriyle örtüşmemesini, denklem (2.28) bir nesne çiftinin aynı kutuya atanması durumunda örtüşmemesi durumunun kontrolünü, (2.29) denklemi her bir nesnenin kesinlikle bir kutuya atanmasını, (2.30) denklemi bir kutuya nesne atanması durumunda o kutunun kullanıldığını ve denklem (2.31)-(2.33) tüm nesnelerin kutunun fiziksel boyutlarına göre yerleştirildiğini göstermektedir.

### 2.4.3 Üç Boyutlu Kutulama Problemi Algoritmaları

3BKP probleminin çözümü için kullanılan sezgisel algoritmalar Martello ve Vigo [21] tarafından aşağıdaki gibi özetlenmektedir:

- **TEKKUTU Algoritması:** Dal ve Sınır algoritmasının değiştirilmiş bir versyonudur. Üç boyutlu kutulara, üç boyutlu nesnelerin yerleşmesiyle, en iyi doldurulmuş tek bir kutuyu bulmasını hedefleyen bir algoritmadır. Bunun için nesneleri x,y,z boyutunda sıfır başlangıç pozisyonundanbaşlayarak yerleştirir. Yerleştirilen nesneler, kutunun bitiş noktasına geldiğinde, kutuda boş hacim kalmışsa o bölgeye atanmamış nesneler yerleşmeye çalışır, yoksa başka bir kutuya geçerek algoritma en iyi kutuyu bulana kadar bu işleme devam eder.
- **Temel Dallanma Ağacı Algoritması**: Nesneler gerçek konumlarını belirtilmeden farklı kutulara atanır. Bu nesneler, öncelikli olarak hacimlerine göre azalanbir düzende sıralanır ve ilk-derinlik stratejisine göre yerleştirilir.




Şekil 2.11 Üç Boyutlu Kutuların Yerleşmesi

- **Tek Kutu Doldurma Algoritması:** ONEBIN algoritmasının M-kesikli bir sürümüne dayanan iki sezgisel algoritmadan biridir. Bir karar düğümü, alt sınırı geçmediği veya sezgisellerle yerleştirilemediğinde, $\mathcal{R}$listesindeki nesnelerin mevcut atamasının fizibilitesi TEKKUTU algoritması ile test edilebilir. Bu algoritmada yalnızca $J'$deki tüm nesnelerin paketlendiği bir çözüm bulmaya yöneliktir. Geçerli en iyi kutu dolduran küme bir değer ile sabitlenir(F).Algoritma F'nin değiştirilmemiş bir değerini verirse, tümüyle doldurma yok demektir.

# BÖLÜM 3

## KONTEYNER YÜKLEME PROBLEMİ

Günümüzde yüklerin %90 gibi önemli bir kısmı konteynerlerle taşınmaktadır ve bu durum yaklaşık 250 milyon konteynerin bir yıl içinde gemilere yüklendiğini göstermektedir [36]. Buveriler nakliye konteynerlerinindünya ekonomisindeki en önemli parçalardan biri olduğunu göstermektedir. Nakliye konteynerleri, tüm taşıma modları (hava, kara, deniz)için malların dünya çapında taşınmasında kullanılanöncelikli yöntemdir.

Maliyet enazlama hedefine ulaşmak, mevcut konteyner hacimlerinden yüksek seviyede faydalanmak için konteyner yükleme problemi araştırmacıların yoğun ilgisini çekmektedir. Konteyner yükleme problemleri NP zor problemlerden olduğu için, araştırmacılar etkin sezgisel algoritmalar geliştirmektedir. Bu problemin çözümüyle ilgili ilk sezgisel algoritmalar George ve Robinson[2]tarafından öne sürülmüştür. Bischo ve Marriott[3],George veRobinson tarafından öne sürülen sezgisel algoritmadan etkilenerek ondört tane sezgisel algoritma ortaya çıkarmıştır.



Şekil 3.1 Konteyner Yükleme

Konteyner yükleme problemi, 3 boyutlu nesnelerin 3 boyutlu konteynerlere atanması için geometrik bir atama modeli olarak tanımlanmaktadır. Genellikle tüm kutuların konteynerlere atanmasını sağlamak ve bu kutuların çakışan lokasyonlara yerleştirilmemesi kısıtları ile çalıştırılmaktadır.

Toffolo vd. [37], iki boyutlu konteyner yükleme probleminin çözümü için, "bir yüz yukarı" kısıtı göz önünde bulundurması sebebiyle, konteyner içi kullanımı arttırabilen bir yöntem önermiştir. Yerleşim yapıldıktan sonra son konteynerin içindeki kalan kapasitenin kullanımını sağlamak üzere, bir sonraki sevkiyata bekletilmesi önerilmiştir. Uygun çözümler hızlı bir şekilde oluşturularak, yerel arama ve 'Yık ve Yeniden Oluştur' (Ruin-and-recreate) sezgiseli ile sonuçlar iyileştirilmiştir.

Araújo vd. [38], Konteyner gemisini yüklemek ve boşaltmak için gerekli olan hareket sayısını en aza indirgemek ve gemideki dengesizliği azaltmak amaçlarıyla, üç boyutlu konteynerlerin gemiye yüklenmesi problemini çözmek üzere, öbek arama temelli yeni bir algoritma önermiştir. Bu yöntemin performansı, [39] tarafından önerilen pareto benzetimli tavlama yöntemine göre zaman/doğruluk açısından daha iyi sonuçlar üretmiştir.

Bortfeldt vd. [40], konteyner yükleme problemleri için geniş bir literatür araştırması yaparak, çalışmasında literatürde incelenen kısıtları detaylı olarak özetlemiştir.

Literatürde yer alan konteyner yükleme ile ilgili çalışmalar incelendiğinde, inceledikleri amaçlar ve kısıtların yüksek çeşitlilik içerdiği gözlemlenmiştir. Bu kapsamda, Çizelge 3.1 incelenen çalışmalar özetlenmektedir.

Çizelge 3.1 Literatürdeki Kutulama ve Konteyner Yükleme Çalışmaları

| Çalışma | Konteyner Sayısı | Boyut | Amaç (lar) |
|---|---|---|---|
| Lim vd. [41] | Tek/Çok | 3B | • Kullanılan konteyner sayısının minimize etmek<br>• Konteyner hacim kullanımını maksimize etmek |
| Geiger[9] | Çok | 1B | • Heterojenliği minimum yapmak<br>• Kullanılan konteyner sayısını minimum yapmak |
| Gonçalvez vd. [42] | Tek | 3B | • Yüklenen hacmin yüzdesini maksimize etmek |

Çizelge 3.2 Literatürdeki Kutulama ve Konteyner Yükleme Çalışmaları(devam)

| Çalışma | Konteyner Sayısı | Boyut | Amaç (lar) |
|---|---|---|---|
| Junquira vd.[43] | Tek | 3B | • Konteynere yüklenen kutuların toplam değerini maksimize etmek |
| Lim vd. [44] | Tek | 3B | • Akslarına göre tırlara yüklenen hacim kullanımını maksimize etmek<br>• Boşaltılan kutu hacmini minimize etmek. |
| Wang vd.[8] | Tek | 3B | • Önceliklendirilmiş paketlerin yüklenen toplam hacmini maksimize etmek |
| Araya vd.[33] | Tek | 3B | • Yüklenen kutuların hacmini maksimize etmek |
| Che vd. [45] | Çok | 3B | • Konteynerlerin her katmanında yer alan kutu sayısını maksimize etmek |
| Sheng vd. [46] | Tek | 3B | • Dikdörtgen kutuları yükleme planı oluşturulması için, yerleştirme hacmini maksimize eden kutu altkümelerinin belirlenmesi. |
| Beyaz[47] | Çok | 2B | • Kullanılan konteyner sayısını minimum yapmak |
| Ji ve ark. [48] | Çok | 3B | • Yükleme sırası ve yükleme lokasyonunun değişimini minimize etmek |
| Li vd. [15] | Çok | 3B | • Konteyner hacim kullanımını maksimize etmek |
| Wei vd.[49] | Çok | 3B | • Konteyner maliyetini minimize etmek |
| Zheng vd.[50] | Tek | 3B | • Konteynere yüklenen kutuların toplam değerini maksimize etmek<br>• Konteyner hacim kullanımını maksimize etmek |
| Araújo vd.[38] | Çok | 3B | • Konteyner hareketlerini minimize etmek<br>• Konteyner dengesizliği minimize etmek |
| Sheng vd.[51] | Çok | 3B | • Konteynerdeki ürünlerin hacmini maksimize etmek<br>• Her paletin altı, konteyner tabanı veya bir palet tarafından tam olarak desteklenmeli |

Konteyner Yükleme Problemleri literatürde konteyner sayısı, boyutu ve amaçlarına göre sınıflandırılmıştır (Çizelge 3.1). Araştırmacılar, konteyner sayısı enazlama, konteyner içi maksimum alan kullanma ve konteyner içinde yüklerin dengeli yerleştirilmesinin çözümüne yönelik algoritmalar geliştirmektedirler. Konteyner yükleme problemlerinde karşılaşılan bu problemlerin çözümüne yönelik geliştirdiğimiz İki Boyutlu Kombinatoryal Alan Maksimizasyon Yerleştirme Algoritması konteyner içi alanı maksimum şekilde kullanarak konteyner sayısını en az kılmaktadır.

## 3.1 Konteyner Yükleme Problemlerinin Çeşitleri

Wascher vd.[52] genel konteyner yükleme problemlerini altı özel problem başlığı altında sınıflandırmıştır. Bu altı kategori şunlardır:

- **Eş Olarak YerleştirmeProblemi:** Bu problem, mümkün olan en büyük sayıda birbirine eşboyutlu nesnelerin, belirli ve sınırlı sayıda olan kutular kümesine atanmasıdır.
- ***Yerleştirme Problemi*:** Bu problem, zayıf derecede heterojen olan nesneler kümesinin, belirli ve sınırlı sayıdakikutular kümesine yerleştirilmesidir. Burada, kutuların içine yerleştirilen nesnelerin sayısını ençoklamak veya bu kutuların içindekullanılmayan alanı en aza indirmek hedeflenmektedir.
- ***Sırt Çantası Problemi:*** Nesneler kümesinin çok fazla heterojen olmasın durumunda, nesnelerin belirli bir dizi kutulara atanması problemidir. Bu problemin amacı, büyük ve sınırlı alana sahip bir kutununiçine yerleştirilebilecek küçük nesne sayısının ençoklanmasıdır.
- ***Açık Boyut Problemi:*** Bir boyutu olan heterojen bir veya daha fazla kutuyu içerir. Bu kutular, nesneler kümesinin tamamını barındırmalıdır.
- ***Kesme Stok Problemi:*** Bu problem, zayıf heterojen nesneler kümesinin, tüm boyutları sabit ve özdeş olankutular kümesine atanması problemidir.
- ***Kutulama Problemleri:*** Bu problem, atanacak nesneler kümesinin oldukça heterojen olması dışında, prensip olarak kesme stok problemlerine benzemektedir. Kutulama problemlerinin temel amacı, nesneleri kutuda en az boşluk kalacak şekilde kutulara yerleştirmek ve/veya kullanılan kutu sayısı enazlamaktır.

Konteyner yükleme problemleri girdi enazlama problemleri olarak modellendiğinde, literatürde tek boyutlu stok kesme problemi, çok boyutlu stok kesme problemi, artık stok kesme problemleri, tek tipli kutulama problemi, çok tipli kutulama problemi, artık kutulama problemi ve açık boyutlu problemleri başlıklarında incelenmektedir.

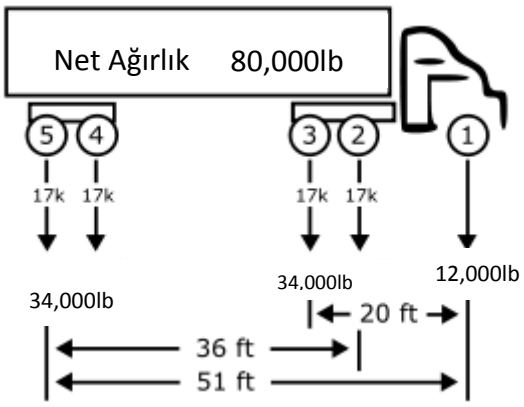
Konteyner yükleme problemleri çıktı ençoklama problemleri olarak modellendiğinde, eş kutulu yerleştirmeproblemi, tek kutu yerleştirme problemi, çoklu eş boyutlu kutu yerleştirme problemi, çoklu karma boyutlu kutu yerleştirme problemi, tekli sırt çantası problemi, çoklu eş sırt çantası problemi ve çoklu karma sırt çantası problemi başlıklarında incelenmektedir.

## 3.2 Konteyner Yükleme ProblemlerininKısıtları

Bu bölümde, konteyner yükleme problemlerinde karşılaşılabilecek kısıtlamaları tanımlanmaktadır. Konteyner yükleme kısıtları, kutu kaynaklı kısıtlar, paket kaynaklı kısıtlar ve konumlandırma kısıtlamalarını içermektedir.

### 3.2.1 Konteyner Kaynaklı Kısıtlar

Konteyner kaynaklı kısıtlar; konteynerin taşıyacağı maksimum ağırlık kısıtı ve yüklerin dengeli bir şekilde yerleşmesini gerektiren yük denge kısıtı olmak üzere iki çeşittir.




Şekil 3.2 Tır ve Konteyner Ölçüleri

- **Ağırlık Kısıtları:** Bir konteyner belirli bir ağırlık limiti aşılmadığı sürece, nesnelerle yüklenir. Ağırlık kısıtı, doğrusal sırt çantası problemine benzer olarak basit bir şekilde modellenebilir; yüklenen nesnelerin ağırlıklarının toplamı, konteynerin taşıyabileceği ağırlık sınırına eşit veya daha küçük olmalıdır.

Konteyner yükleme problemleri literatürde, ağırlık limitleri, Liu vd. [4], Gehring vd. [53] tarafından ele alınmıştır. Eşzamanlı olarak hem konteyner yükleme hem de araç rotalama problemin çözümünü araştıran çalışmalar genellikle ağırlık kısıtını dikkate almaktadır. Her bir paketin geometrik boyutlarına ek olarak ağırlığı ile karakterize edilen belirli bir paket kümesi, genellikle merkezi bir depoda bulunur ve coğrafi olarak dağınık müşterilere teslim edilir. Kutuların, genellikle homojen olduğu düşünülür ve belirli bir ağırlık taşıma kapasitesine sahip olduğu kabul edilir. Bu kutular depoda bulunan bir dizi araç ile taşınır. Böylece, genişletilen problem toplam mesafeyi enazlayan bir dizi güzergâh bulma amacıyla ve yalnızca alana göre değil, aynı zamanda her bir aracın ağırlık taşıma kapasitesine göre de uygulanabilen bir yükleme modeli içermektedir [54][55]. Dereli vd. [56] konteynerin yükleme kapasitesinin hacim ve ağırlık olarak sınırlı olduğu SBP tipi bir problemi ele almıştır. Bununla birlikte, maksimum konteyner hacim kullanımının yanı sıra, konteynerde en yüksek ağırlıkta kargo yüklemesi de hedeflenmektedir.

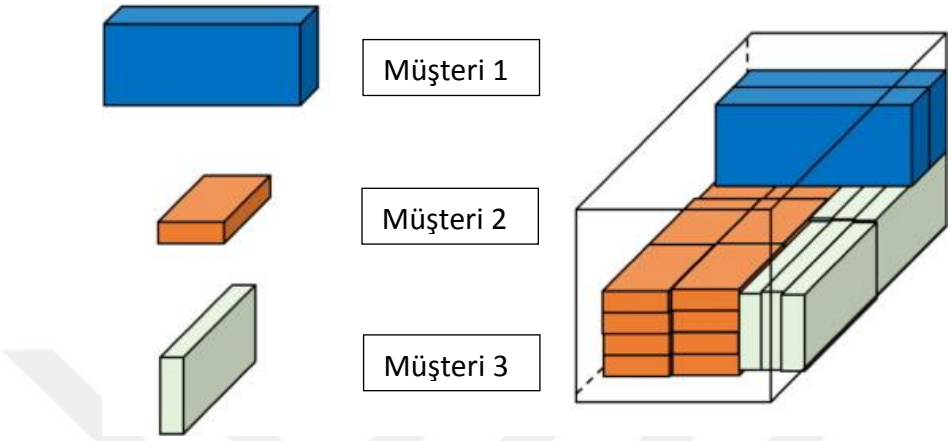
Paketlenecek nesnelerin köpük, kauçuk vb. hafif malzemedenoluşması durumunda, ağırlık kısıtı göz ardı edilir. Aksi durumda, ağırlık kısıtı konteynerlerin boyutlarına göre hesaplanan alan kısıtından her zaman daha önceliklidir.

- **Yük dengesi kısıtı:** Kutu veya konteyner taban yük dağılımı konteynerin dengesi açısından oldukça önemlidir. Yük dengesi kısıtı, paketağırlıklarının konteyner tabanınamümkün olduğunca eşit şekilde yayılmasıdır. Dengeli yükler, konteynerin taşınması sırasında kargonun kayma riskini azaltır. Dengesiz yükler ise, konteyner tır üzerine taşınırken, aksın parçalanması gibi istenmeyen sonuçlara neden olabilir[53].

  Uygun bir ağırlık dağılımı elde etmek için, yük ağırlık merkezinin konteyner tabanının geometrik orta noktasına yakın olmasıgerekmektedir [57]. Geometrik ağırlık merkezinin geometrik orta noktaya uzaklığının miktarıbelli bir mesafeyi aşmamalıdır[53][58].

### 3.2.2 Paket ile ilgili Kısıtlar

Paketler konteynerlere yerleştirilirken paket kaynaklı kısıtlamalar göz önünde bulundurulmalıdır. Paketler; dağıtım bölgelerine, paketiçeriğine, istiflenme kabiliyetine, döndürülme imkânına ve taşınma gereksinimlerine göre yerleştirilmesi gerekmektedir.



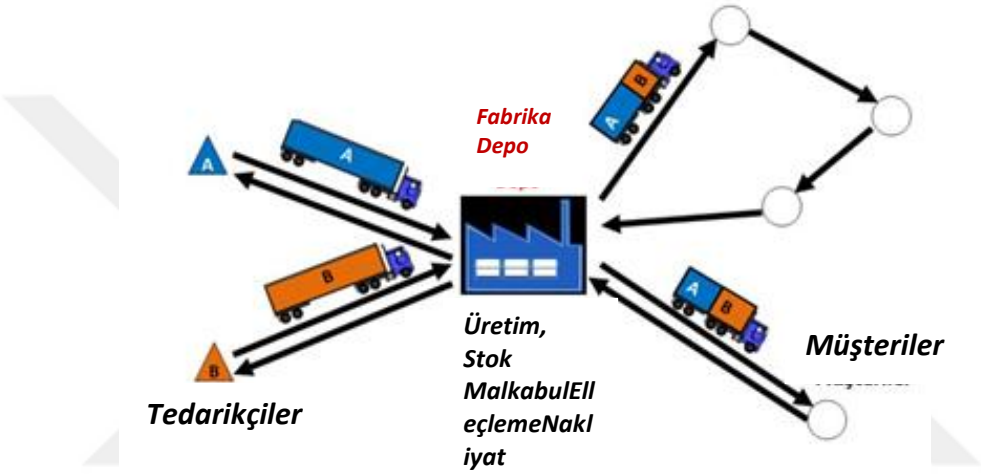

Şekil 3.3 Paket Kaynaklı Kısıtlar

- **Yükleme Önceliği Kısıtı:** Yükleme önceliği, teslimat sürelerinden veya ürünlerin tazeliği veya raf ömründen kaynaklanan gerekliliklerden kaynaklanabilir. Ayrıca, belirli paketlerin yüklenmesi diğer paketlerin yüklenmesinden daha cazip olabilir. Mevcut kutu alanı tüm paketlerin yerleştirilmesi için yeterli olmadığında, hangi nesnelerin yükleme için seçilmesine karar verilmelidir [13]. (Şekil 3.3) müşterilerin önceliklerine göre yerleştirilen paketleri göstermektedir.
- **Yönlendirme Kısıtlamaları:** Bir kutunun her bir boyutu, üç dikey yönlendirme oluşturur ve yüksekliğine bakılarak yönlendirilir. Yükseklik olarak belirli bir boyut seçilir ve kutunun dikey yönlendirmesi tanımlanır. Toplamda, 6 $(3 \times 2)$tane yönlendirme vardır; yani (dikdörtgen) bir paket konteyneredikey olarak yerleştirilebilir. Bununla birlikte, uygulamada bir kutunun kabul edilebilir sayıda yönelimi hem dikey hem de yatay yönde sınırlandırılabilir.
- **İstif kısıtlamaları:** Kutuların birbirinin üstüne nasıl yerleştirileceğini sınırlayan kısıtlardır. Birbiri üzerine yerleştirilecek kutular, sınırlı yük taşıma mukavemetine sahip olduğundan, istifleme gerçekleşmeden önce destekleme kutusunun birim alanına uygulanabilecekazami ağırlık ile basitçe ölçülmemelidir. Bunun yerine, yük

taşıma kuvvetini, kutunun yan duvarlarının mukavemetini ölçerek belirlemek daha uygundur [13].

### 3.2.3 Kargo Kaynaklı Kısıtlar

Kargo kaynaklı kısıtlar literatürde ve uygulamada iki şekilde gerçekleşmektedir:(1) aynı sevkiyatta yer alması gereken bir paket kümesinin komple sevkiyat kısıtlaması ve (2)bir müşteriye ulaştırılacak ürünlerin tek sevkiyatla ve aynı anda ulaşmasını gerektiren tahsis kısıtıdır.

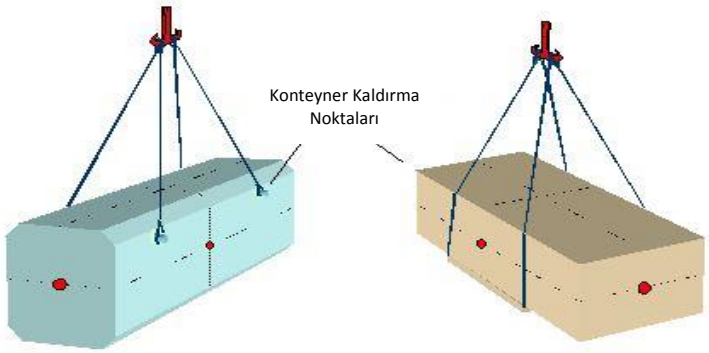



Şekil 3.4 Kargoların Müşterilere Teslimi

- **Komple Sevkiyat Kısıtlamaları:** Bir paket kümesininelemanlarından bir paketyüklenirse, bu kümenin diğer tüm elemanları da yüklenmelidir. Bir paket kümesine ait herhangi bir eleman yüklenemezse, bu kümenin hiçbir nesnesi de yüklenemez. Eksik sevkiyatlara izin verilmemesi için, komple sevkiyat kısıtlamalarına dikkat edilmelidir.
- **Tahsis Kısıtlamaları:** Tahsis kısıtlamaları yalnızca birden fazla konteyner yükleme probleminde ortaya çıkmaktadır. Bu tip kısıtlarda, belirli bir alt kümenin nesnelerinin aynı kutuya girmesinigerekmektedir [58].Tahsis kısıtlamaları, belirli paketleri aynı konteynere yerleştirmeyi engellemek için de kullanılabilir. Örneğin, gıda ve parfümeri eşyaları tek bir konteynere yüklenmemelidir [24].

### 3.2.4 Yükleme Kaynaklı Kısıtlar

Yükleme ile ilgili kısıtlamalar, konteynere yerleşmiş paketlerin özelliklerine göre yapılan en son düzenlemelerden kaynaklanır.




Şekil 3.5 Konteynerin Taşınması

- **Kararlılık Kısıtlamaları:** Literatürde yük istikrarı, konteyner alanı kullanımında en önemli konulardan biri olarak kabul edilmektedir[46]. (Şekil 3.4), yükleme ve boşaltma işlemlerinde kararsız yükler, halattan kopup ciddi hasarlara yol açabileceğini göstermektedir.

  Gerçek hayatta bu kadarönemliolan bu kısıt, literatürde araştırmacıların çok fazla dikkatini çekmemektedir. Yazarlar, yüksek konteyner içi alan kullanımı garanti edildiğinde, kararlılığın yük yoğunluğunun bir sonucu haline geldiğini savunmaktadır[59][60]. Pratik olarak, yük dengelemesi, ek destekler ya da kalan boşlukların içine yerleştirilen dolgu malzemelerinin (köpük parçaları gibi) kullanılmasıyla da sağlanabilmektedir [59][60].
- **Karmaşıklık Kısıtları:** Karmaşık yükleme şekilleri, manuel konteyner yüklemesi, mekanik ve otomatik yükleme için her zaman uygun olmamaktadır. Çizelgelerin görselliği yükleyici personel tarafından düzgün anlaşılmayabilirve uygulaması uzun zaman alabilir. Ayrıca, gelişmiş mekanik ve otomatik yükleme teknolojileri karmaşık yük düzenlemeleri için ek yatırım gerektirebilir. Karmaşıklık kısıtlamaları, teknolojik ve insan kaynaklarının sınırlamalarını yansıtan bir durumdur[13].

# BÖLÜM 4

# ÖNERİLEN YÖNTEM

Kutulama problemi, kullanılan kutu sayısını en aza indirmek amacıyla, çeşitli hacimlerdeki nesnelerin çeşitli hacimlerdeki kutulara yerleştirilmesi problemidir. Kutulama problemi NP-zor problemi olmasına rağmen, büyük boyuttaki problemler için ileri düzeydeki algoritmalarla optimal çözümler bulunabilmektedir.

Bu bölümde, iki boyutlu kutulama problemlerinin çözümü için önerilen İki Boyutlu Kombinatoryal Alan Maksimizasyon Yerleştirme Algoritması (2D Combinatorial Area Maximization Algorithm- 2D-CAMFA) açıklanmaktadır. Önerilen algoritmanın performansı, BF-Raf Yerleştirme Algoritması performansı ile test edilmektedir.

## 4.1 İki Boyutlu En Uygun Yerleştirme Raf Algoritması

Bölüm 2.3.3'de anlatılan 2B yerleştirme algoritmaları literatürde en iyi sonuçlar verdiği düşünülen temel modellerdir. Biz de bu çalışmada ilk olarak rassal seçtiğimiz (Ek-A1) iki boyutlu verilere BF-Raf Yerleştirme Algoritmasını uyguladık. (Şekil 4.2)'ye göre parametreler;

$p_j$= $j$ indisli kutunun başlangıç noktası($XAxis$)

$q_j$= $j$indisli kutunun başlangıç notası($YAxis$)

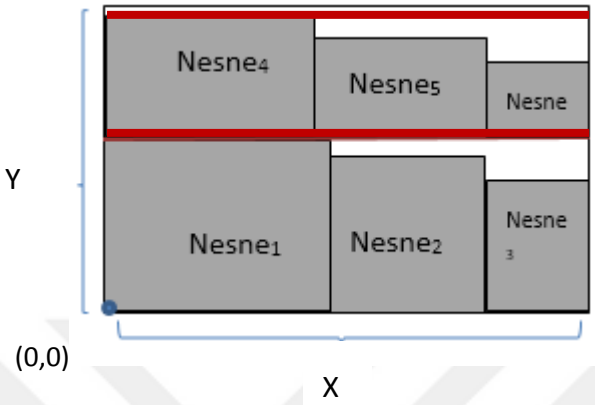
$j$=# kullanılan konteyner sayısı $j=\{1,2,\ldots\ldots,N\}$

$x,y$=Konteynerin Genişliği ve Uzunluğunun kordinatları

$x_i$=iindisli nesnenin eni

$y_i$=iindisli nesnenin uzunluğu

$d_j$=j indisli kutuda kalanuzunluk




Şekil 4.1 İki Boyutlu Nesne ve Konteyner Görünümü

Bu modele göre uzunluğu ve genişliği "$w_i,h_i$" olan $m$ adet paketler bir listede tutulur. Bu paketler uzunluğuna göre büyükten küçüğe sıralandıktan sonra; $K$ adet, eni $W$ve yüksekliği $H$ olan kutulara atanmak üzere Raf Yerleştirme Algoritmasına göre yerleşmektedir.

İlk olarak $x_j,y_j$ başlangıç noktaları (0,0) belirlenir. Kutu değeri ve raf değeri 1 olarak atanır (1)-(5). $\mathcal{R}$ Listesinde yer alan paketler uzunluğuna göre sıraladıktan sonra en uzun paket konteynerin sol alt köşesinden başlayacak şekilde ilk rafa yerleşir. Genişlik $W$dolana kadar devam eder. Rafın maksimum uzunluğuna sahip paket ilk pakettir ve bu uzunluk konteynerin uzunluğundan çıkartılırve ($d_j$) değeri bulunur. Yeni gelen listedeki paket ilk rafın genişliğinde kalan boşluktan büyükse ve konteynerin kalan uzunluğundan küçükse yeni bir raf açılır ve genişlik 0 olarak güncellenir. Konteyner dolana kadar bu işlem devam eder.

ALGORİTHM:2B Shelf –BF(BEST FIT)

Başlangıçta:

$Kur\ x \leftarrow 0$ (4.1)

$Kur\ y \leftarrow 0$ (4.2)

$Kur\ d \leftarrow 0$ (4.3)

$Kur\ \ F=\{(W,H)\}$ (4.4)

Paketle:

Her bir R={($w,h$)} için azalan sırada (4.5)

$if\ (w_i,h_i) <=(W,H)$ (4.6)

$Assign\ (w_i,h_i)\ to\ (W,H)$ (4.7)

$x_{(i+1)kj}=x_{ikj}+x_{(i+1)kj}\ \ y_{ikj}=y_{jki}$ (4.8)

$En\ büyük\ y_{kj}\ değerini\ bul$ (4.9)

$d_j=H\text{-}max\ y_k\ değeri$ (4.10)

$while\ x_j=W\ yeni\ raf\ aç$ (4.11)

$if\ y_{i+1}<=d_j\ and\ x_i<W$ (4.12)

$Set\ y_{(k+1)}=d_j\ and\ x_{(k+1)}=0$ (4.13)

$continue$

$else;$

$Başlangıç\ değerlere\ dön\ ve\ yeni\ kutu\ aç$ (4.14)

$k_{J+1}=k_J+1$ (4.15)

*Bitir*

Şekil 4.2 İki Boyutlu BFD-Raf Algoritması Pseudo Kodu

Literatürü incelediğinde 2B Yerleştirme Algoritmalarında Raf yerleştirme yöntemleri ile veri yapısına göre optimum sonuçlar elde edildiği görülmüştür. Bu tez çalışmasında rassal verilere

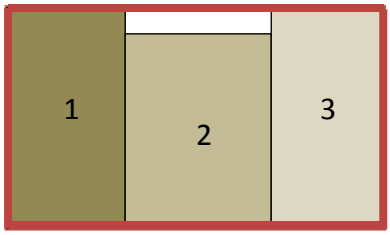
BF sezgiseli ve Raf Yerleştirme algoritması uygulanmıştır. Edinilen izlenimlere göre Raf yerleştirme algoritmalarında bazı eksiklikleri tespit edilmiştir. Bunlar;

- Konteyner içi alan verimli kullanılamadı. Paketlerin maksimum uzunluğuna göre kapatılan raflarda boş alanlar oluştu. Bu durum da daha fazla konteyner kullanımına sebep oldu.
- Paketler uzunluklarına göre sıralandığı ve genişlikleri bu sıralamada göz ardı edildiği için rafların boyutları eşit dağılmadı.
- Bu model paketleri konteynere tek tek uygunluğunu test ettiği için uzun sürede çalışmaktadır.
- Model ağırlık, üçüncü boyut, öncelikler, ürün çeşidi, denge vb. kısıtları içermediği için, bu modelin pratik hayatta uygulanabilirliği düşüktür.

Bu çalışmada Raf Yerleştirme Algoritmalarının eksikliklerini gidermek ve daha optimum çözümler elde edebilmek için 2D-CAMFA modeli geliştirilmiştir.

## 4.2 İki Boyutlu Kombinatoryal Alan Ençoklama Yerleştirme Algoritması

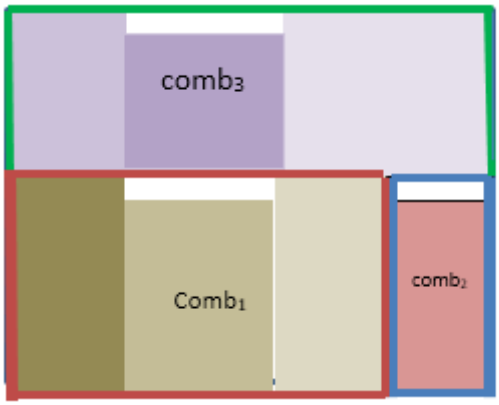
Konteyner yükleme modelleri kısıtları Bölüm 3.2'de ayrıntılı olarak anlatılmıştır. Bu kısıtlar altında optimum sonuçlar bulmak ve amaçlara ulaşmak için literatürden araştırılan Raf Yerleştirme Algoritması Bölüm 4.1 de rassal verilere uygulanmış ve sonuçlar elde edilmiştir. Bu algoritmada gözlemlenen bazı eksiklikleri gidermek adına 2B-Kombinatoryal Alan Ençoklama Yerleştirme Algoritması (2D-CAMFA) modellenmiştir. S-BFD Algoritmasının aksine 2B-CAMFA modeline Bölüm 3.2'de anlattılan bazı kısıtları eklenmiştir. Bunun için kutu ve nesnelerin İki Boyutuna ek olarak üçüncü bir kıriter belirlenmiştir. Öncelikle listedeki nesneler ürün çeşidine, ağırlığına ya da derinliğine göre gruplanmaktadır. Raf yerleştirme algoritmalarında listedeki her bir nesnenin kutulara uygunluğu test edilmektedir. Bu durum, sonuçların uzun sürede alınmasına sebep olmaktadır. Zaman tasarrufu yapmak için 2D-CAMFA modelinde nesneler kutulara yerleştirilmeden önce $x$ ve $y$ koordinatlarına göre bütün kombinasyonları bulunur (Şekil 4.3). Alandan maksimum şekilde yararlanmak için bu kombinasyonlardan maksimum alana sahip olan dikdörtgen, ilk rafa yerleşir (Şekil 4.4).




$xi$

Şekil 4.3 Kutulara Sığabilecek Nesnelerin Kombinasyon Örneği

Rafın $x$ koordinatında kalan boşluğuna göre kombinasyon tekrar hesaplanır. Bu kombinasyonlardan herhangi birinin alanı kalan alandan küçükse raf dolmaya devam eder; dolan rafın maksimum yüksekliği belirlenir ve kutunun yüksekliğinden çıkartılır. Aksi takdirde yeni bir raf açılır; kalan yükseklik ve kutunun genişliğine göre nesnelerden tekrar kombinasyon oluşturulur; maksimum alanlı kombinasyon seçilip yerleştirilir ve aynı kurallar kutu içi dolana kadar devam eder. Kutu içinde yer kalmadığında ise yeni bir boş kutu oluşturulup yerleştirme işlemi bütün nesneler yerleşene kadar devam eder.




Şekil 4.4 Nesnelerin Kutulara Yerleşmesi

Sonuç olarak her kutuda nesnelerin maksimum alan kombinasyonu kullanıldığı için raflardaki yük dağılımını en iyi şekilde sağlanmıştır. Bu da S-BFD'de ortaya çıkan yük dengesizliğini ortadan kaldırmıştır. Ayrıca ürünler önceden seçilen herhangi bir kıritere göre gruplandığı için istenilen ürünler kutunun en altına veya rafın en üstüne tercihe göre yerleştirilebilir.

### 4.2.1 İki Boyutlu Kombinatoryal Alan Maksimizasyon Yerleştirme Modeli

İki Boyutlu Kutulama Problemlerinin çözümüne yönelik önerilen 2D-CAMFA'nın en genel halinin sözde kodu şu şekildedir;

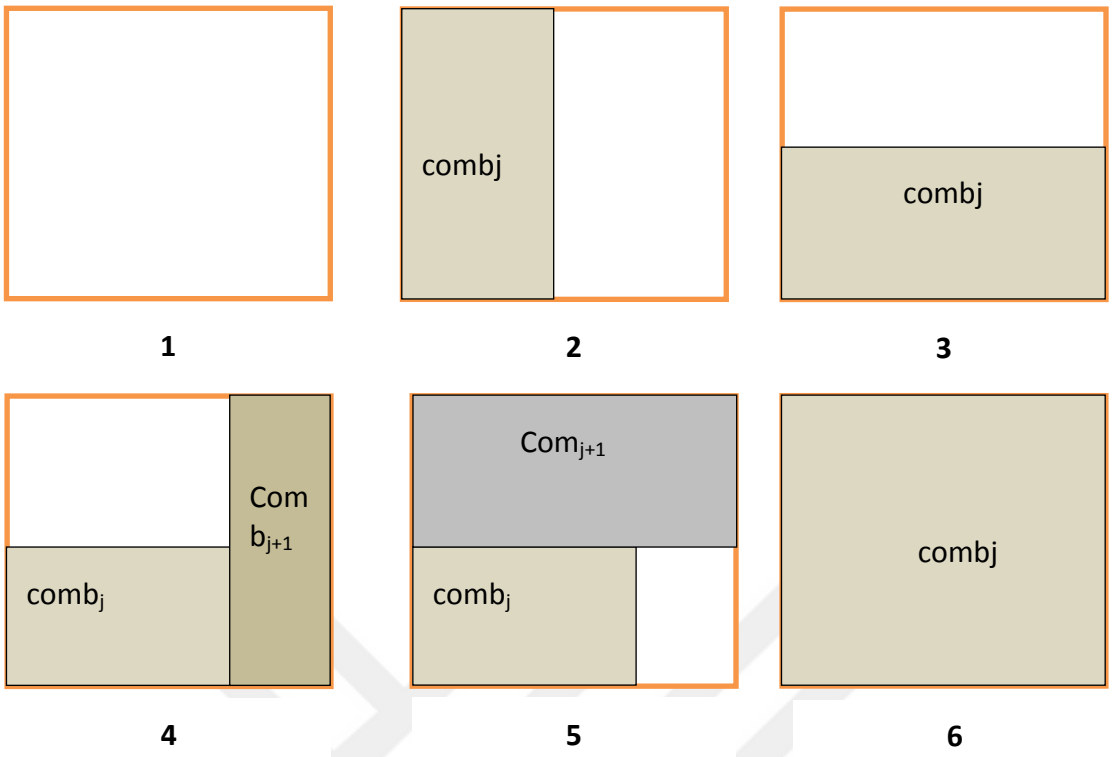
Çizelge 4.1 2D-CAMFA Sözde Kodu

| 2D-CAMFA PSEUDO CODE | |
|---|---|
| Başlangıç: | |
| Kur f={($w,h,d$)} | |
| Kur *Area* | |
| Kur F={($W,H,D$)} | |
| Define combination[xkutukalan, ykutukalan]: | (4.16) |
| comb= [ ] listesi oluştur | (4.17) |
| boxes=[ ] listesi oluştur | (4.18) |
| $x_{jkl} = x_{ijkl}$ *değerlerinin toplamı* | (4.19) |
| $y_{jkl} = \max\{y_{jkl}\}$ | (4.20) |
| $x_{jkl} <= x_{kl}$ ve $y_{jkl} <= y_{kl}$ | (4.21) |
| *Bu koşulları sağlayan değerler için comb listesini arttır* | |
| $Area = x_{jkl} * y_{jkl}$ | (4.22) |
| *Area listesini azalan sırada diz maksimum değerini bul* | |
| *Aynı d (paketz) boyuttaki kutuları grupla* | (4.23) |
| $y_{kl} = \max_j\{\max_i\{y_{ijkl}\}\}$ *l konteynerinin k rafindaki j kombinasyonlu i kutusu için,* | (4.24) |
| $x_{kl} = x_{ijkl}$ *l konteynerinin k rafindaki j kombinasyonlu i kutularının toplamı,* | (4.25) |
| $x_l$=xkutukalan=W-$x_{kl}$ | (4.26) |
| $y_l$=ykutukalan= H-$y_{kl}$ | (4.27) |
| $Area <= x_l * y_l$ | (4.28) |
| Paketle: | |
| *if xkutucomb<xkutu and ykutucomb= ykutu* | (4.29) |
| *box=box* | |
| *shelf=shelf* | |
| *if xkutu=xkutucomb and ykutucomb<ykutu* | (4.30) |

Çizelge 4.2 2D-CAMFA Sözde Kodu (devam)

| 2D-CAMFA PSEUDO CODE | |
|---|---|
| *box=box* | |
| *shelf=shelf* | |
| *if xkutu>xkutucomb and ykutucomb<ykutu* | (4.31) |
| *box=box* | |
| *if areakalanx<areakalany* | (4.32) |
| *shelf=shelf+1* | |
| *if areakalanx>areakalany* | (4.33) |
| *shelf=shelf* | |
| *if xkutu=xkutucomb and ykutucomb=ykutu* | (4.34) |
| *shelf=shelf+1* | |
| *box=box+1* | |
| *box değerinin en küçük değerini Bul* | (4.35) |
| *return boxes;* | |

(4.16) x ve y kordinatlarında kalan boşluğa göre kombinasyon foksiyon listesi tanımlanmaktadır. (4.17)-(4.18) boxes (kutular) ve comb (kombinasyon) listesi oluşturulur. (4.19)-(4.20) $x_{jkl}$ ve $y_{jkl}$ *l* konteynerinin *k* rafında ve *j* kombinasyonunda bulunan *i* kutularının uzunluğunun toplamını ve maksimum yüksekliğini hesaplamaktadır. (4.21) *l* konteynerinde bulunan *i* kombinasyonlarının uzunluk ve maksimum yükseklik toplamı *l* konteynerinden küçük veya eşit olmalıdır. (4.22) Kombinasyonlarda kullanılan paketlerin alanını hesaplamaktadır. Bu hesaplanan alanlar azalan sıraya dizilmekte ve maksimum alanlı kombinasyon seçilmektedir. (4.23) Kutular derinliğine göre veya kullanıcı tarafından seçilen herhangi bir krtitere göre gruplanmaktadır. (4.24)-(4.25)$x_{kl}$ ve$y_{kl}$ $l$ kutusunun $k$ rafında bulunan uzunluk ve maksimum yüksekliklerin toplamıdır. (4.26)-(4.27) $x_l$ ve$y_l$ konteynerinin toplam uzunluğu ve yüksekliğinden kalan değerlerdir.(4.28) Uygun kombinasyonla yerleşen paketlerin alanı kalan alandan küçük olmalıdır.




Şekil 4.5 2D CAMFA Şekilsel Gösterimi

Bu modeli Şekil 4.5'deki gibi ifade edebiliriz. Ilk olrak boş bir kutu oluşuturulur. Bu durumda box=box, shelf=shelf ,$x_{jkl} = 0$ ve $y_{jkl} = 0$ comb$[x_l, y_l]$'ye göre, $x_l - x_{jkl} > 0$ ve $y_l - y_{jkl} > 0$'dır. Nesneler seçtiğimiz herhangi bir kritere göre gruplanır ve nesnelerin kutuya sığabilecek tüm kombinasyonları hesaplanır ve maksimum alanlı kombinasyon seçilir(combj). Kombinasyonun maksimum yüksekliği konteyner ile aynı boyda ve uzunluğundan küçükse, o rafa yeni nesnelerden kombinasyon yapılır ve boş alan doldurulur. Bu durumda box=box, shelf=shelf (4.29), $0 < x_{jkl} < x_l$ ve $y_{jkl} = y_l$ comb$[x_l - x_{jkl}, y_l]$'ye göre, $x_l - x_{jkl} > 0$ ve $y_l - y_{jkl} = 0$'dır. Maksimum alanlı kombinasyon konteynerin uzunluğunu dolduruyor ve yükseklikten yer kalıyor ise box=box, shelf=shelf (4.30), $0 < y_{jkl} < y_l$ ve $x_{jkl} = x_l$ comb$[x_l, y_l - y_{jkl}]$'ye göre, $x_l - x_{jkl} = 0$ ve $y_l - y_{jkl} > 0$'dır.Bunun sonuncunda raf kapanır ve yeni raf açılıp tekrar en uygun kombinasyon seçilip burafa atanır. (4.31)-(4.33) Eğer bu maksimum alanlı kombinasyon hem yüksekliği hemde uzunluğu dolduramıyor ise yeni gelen kombinasyonu yapılmış nesneler en az boşluk kalacak şekilde ya rafa devam eder yada yeni rafa geçer. Birinci durumda box=box, shelf=shelf, $0 < x_{jkl} < x_l$ ve $0 < y_{jkl} < y_l$

comb$[x_l, y_l - y_{jkl}]$'ye göre, $x_l - x_{jkl} > 0$ ve $y_l - y_{jkl} > 0$'dır.İkinci durumda box=box, shelf=shelf+1, $0 < x_{jkl} < x_l$ ve $0 < y_{jkl} < y_l$ comb$[x_l - x_{jkl}, y_l]$'ye göre, $x_l - x_{jkl} > 0$ ve $y_l - y_{jkl} > 0$'dır ye comb$_{(j+1)l}$ yeni kutuya yerleştirilir. (4.35) Combj Kutuyu tam anlamıyla doldurursa veya kutunun içini dolduracak başka bir kombinasyon yoksa; box=box+1, shelf=shelf+1 ,$0 < x_{jkl} < x_l$ ve $y_{jkl} = y_l$ comb$[x_l - x_{jkl}, y_l]$'ye göre, $x_l - x_{jkl} > 0$ ve $y_l - y_{jkl} = 0$'dır. Son olarak işlem yenilenir, yeni boş bir kutu açılır ve tüm nesneler dolana kadar bu durum devam eder.

## 4.3 İki Boyutlu Kombinatoryal Alan Maksimizasyon Yerleştirme Algoritmasının Performansının Değerlendirilmesi

İki Boyutlu Kombinatoryal Alan Maksimizasyon Yerleştirme Algoritması( 2D-CAMFA) ile BFD-Raf Algoritması(S-BFD) karşılaştırılmıştır. Bu iki algoritma da Anaconda Python 2.7 Programlama dilinde yazılmıştır. Modele 250 adet Üç boyutlu rassal veri atanmıştır. Bu veriler üçüncü bir kritere göre gruplanmış ve en uygun alanlı kombinasyon seçilip konteynere yerleşmiştir. Ek A-1'de görüldüğü gibi 2D-CAMFA 80 adet konteynere paketleri yerleştirmiş, En Uygun Sığan Azalan Raf Yerleştirme Algoritması ise 82 adet konteynere yerleştirmiştir. Buna göre rassal verilerde 2D-CAMFA modelinin Raf-Yerleştirme Algoritmasına göre daha iyi sonuçlar verdiği kanıtlanmıştır.

# BÖLÜM 5

## LOJİSTİK SEKTÖRÜNDE BİR UYGULAMA

Uygulama yapılan firma ürünlerinde her zaman kaliteli hammaddeleri kullanmayı amaçlamış, müşteri memnuniyetini TSE ve İSO 9001:2008 TÜV CERT ISO + TÜVNORD 16949 belgeleriyle belgelendirmiş, üretimin her aşamasında da kalite standartlarına uygun olarak çalışmayı kendisine ilke edinmiş bir filtre fabrikasıdır. Ayrıca bu firma Avrupa, Rusya, Orta Doğu gibi yerlere ithalat yapmaktadır. Müşterilere tedarik sağlarken deniz aşırı tedarikler için forwader lojistik firmalarını, karayolu taşımalarında ise kendi tırlarını kullanmaktadır.



Şekil 5.1 İskenderun Lima'nından bir görüntü

Çalışma yapılan filtre fabrikasında, hava filtresi, yağ filtresi, yakıt filtresi ve hava kurutucu filtresi üretilmektedir(Şekil 5.2). Hava filtresi, havadaki toz ve partiküllerin motora ulaşmasını engelleyerek emilen havanın temiz olarak aktarımını sağlayan parçadır. Yağ filtresi, yağlama

sisteminin kirlenmesini önleyerek, motor yağının motorun sürtünme ve aşınmasını azaltması görevinin sürekliliğini sağlayan parçadır. Yakıt filtresi, içten yanmalı motorlar için kullanılan yakıtta bulunan toz partikülleri, pas ve suyun motora zarar vermesini engelleyen parçadır. Hava kurutucu filtreler ise, genellikle ticari araçlarda bulunan fren sistemindeki havanın nemden arındırılmasını sağlar.




Şekil 5.2 Filtre Çeşitleri

## 5.1 Problemin Tanımı

Geliştirdiğimiz algoritmayı uygulayacağımız bu firma yukarıda daha ayrıntılı anlatıldığı gibi çeşitli filtrelerin üretildiği bir filtre fabrikasıdır. Konteyner ya da tırlara bu filtre kolileri yüklenirken Bölüm 3'de anlatılan kısıtlar, yükleme operasyonlarını zorlaştırmaktadır. Örneğin; hava filtreleri diğer çeşitlere göre ağır değillerdir ama en çok hacmi bu tip filtreler kaplamaktadır. Dolayısı ile bu tarz hafif filtrelerin üstüne daha ağır bir koli yüklenemez. Bunun aksine, yakıt filtrelerinin içeriği demir olduğundan ağırdır ve konteynerin en alt rafına yerleşmelidir. Konteyner içi yük dengesi, taşıma sırasında akslara hasar vereceğinden çok önemli bir kısıttır. Bu firmanın konteyner yükleme operasyonlarında hacim ve ağırlık yönünden çok tipli ürünlerle karşılaşıldığından, paketlerin yük dağılımı dikkate alınarak çalışılmıştır.Bu filtreler Şekil 5.2' de gösterilen filtre çeşitleri, filtre boyutları ve koli ölçülerine göre çeşitli boyutlarda paketlenerek konteynerlere yüklenmektedir. Koli ölçüleri genelde kübik bir yapı sergilediği için paketlerin rotasyonu ihmal edilmektedir.

| | | TEKLİ KUTU ÖLÇÜLERİ | | | KOLİ ÖLÇÜLERİ | | |
|---|---|---|---|---|---|---|---|
| FİLTRE NO | KOLİ İÇİ | EN | BOY | YÜKSEKLİK | EN | BOY | YÜKSEKLİK |
| AD2000 | 8 | 145 | 145 | 175 | 305 | 305 | 365 |
| AD2006 | 8 | 145 | 145 | 175 | 305 | 305 | 365 |
| AF4303A | 1 | 335 | 335 | 265 | | | |
| AF4340 | 1 | 305 | 305 | 395 | | | |
| AF4347 / AF4348 | 1 | 360 | 360 | 380 | | | |
| AF4412 | 1 | 200 | 200 | 460 | | | |
| AF4532 | 1 | 210 | 210 | 460 | | | |
| EF1015 | 12 | 138 | 138 | 158 | 567 | 429 | 173 |
| EF1018 | 12 | 125 | 125 | 315 | 515 | 390 | 330 |

Şekil 5.3 Filtrelerin Çeşitleri, Filtre Boyutları Ve Koli Ölçüleri

3B-Konteyner yükleme probleminin tanımına uygun olan firmanın yükleme operasyon sorunlarını 2B Kutulama Algoritmalarından yararlanarak çözmeye çalışılmıştır. Ancak bunu Bölüm 4 de daha iyi sonuçlar veren2D-CAMFA firmanın konteyner yükleme operasyonlarına uygulanmıştır.Bu algoritma ile kolilerin, zaman kaybetmeden, raf ölçülerine göre kombinasyonlayanları yapılmış ve bu paketler üçüncü bir kritere göre( derinlik,ağırlık,ürün çeşidi vb.) gruplanıp, yerleştirilmiştir.



Şekil 5.4 Konteyner İçi Paket Yükleme

## 5.2 Konteynerlerin Özellikleri

Sahada kullanılan konteynerler ölçülerine ve tiplerine göre sekize ayrılmaktadır. Standart Konteynerler (Dry van), genel amaçlı kullanılan konteynerlerdir. Bu konteyner tipi, kuru yüklerin taşınmasında kullanılmaktadır. Ölçüleri genellikle 20' ve 40' olmakla birlikte 45' ve hatta son dönemde 54' uzunlukta konteynerler kullanılmaya başlanmıştır. High Cube

Konteynerler, standart konteynerlerle yapısal olarak benzer olmakla birlikte daha hacimli bir yüklemeye olanak sunan bir konteyner tipidir. Genellikle 40' olarak üretilmiştir.



Şekil 5.5 Sırasıyla 20' ve 40' Dry Van, Hig Cube Konteyner

Open Top Konteynerler, standart konteynerlerden farklı olarak tavan kısmı tamamen açık olan konteynerlerdir. Ölçüleri genellikle 20' ve 40' uzunluktadır. Flatrack Konteynerler, bu konteynerler bir taban platformu ve iki uç duvarından oluşan taşıma araçlarıdır. İstendiğinde uç duvarlar dik sabitlenebilir ya da kapatılıp tabana yatırılarak bir taşıma platformuna dönüştürülür. Genellikle 20' ve 40' ölçüde üretilmektedir. Kullanım alanları ağır tonajlı yükler, yüksek ya da geniş taşmalı yüklerdir. Platformlar (plats), bu taşıma ekipmanları yapısal olarak yalnızca taban platformu biçimindedir. Uç ve yan duvarları yoktur. Küçük alanda ağır tonajlı taşımaya uygundur. Genellikle 20' ve 40' ölçüde üretilmektedir. Doğal kullanım alanları gabari yükler ile ağır tonajlı yüklerdir. Havalandırmalı (Ventilated) Konteynerler, tepe ve taban profillerinde yer alan havalandırma kanallarından doğal hava dolaşımına izin veren konteyner tipidir. Genellikle 20' olarak üretilmektedir. Doğal kullanımı havadar ortamda taşınması gerekli yüklerdir. Soğutuculu (Reefer) Konteynerler, reefer konteynerler içerisinde soğutma ünitesi bulunan özel yalıtımlı konteynerlerdir. Genellikle 20' ve 40' ölçüde üretilmektedir. High cube olanları hafif ve hacimli yüklerin (çiçek, meyve vb.) taşınmasında kullanılır. Dökme Yük (Bulk) Konteynerleri, bu tip konteynerler tavan kısmında 3 adet yükleme kapağı ve kapı kısmında yer alan 2 adet tahliye kapakları olan konteynerlerdir. Genellikle 20' olarak üretilmektedir. Doğal kullanım alanı tahıl, hayvan yemi, baharat vb. yüklerin taşınmasıdır[13].

## 5.3 Amaçlar ve Kısıtlar

Konteyner yükleme problemlerindeki amaç fonksiyonu küçük iki boyutlu kutuları konteynerelere atayarak minimum konteyner sayısını maksimum alan kullanarak bulmaktır.

Bunun için kısıtlar ve varsayımlar;

- Konteynerler sadece bir çeşit ve kapasite hacmi sabit olarak varsayılmıştır. Konteyenerler 40'lık konteyner olup hacmi 60 $m^3$'tür.
- Kutular z boyutuna göre gruplandırılmış ve konteynerlerin x ve y boyutlarına göre atanmıştır. Ancak ürün çeşidine ya da ağrırlıklarına göre de gruplandırılabilir.
- Kutular konteynerin alanı dolana kadarraf atanmış, maksimum alandaki kutular altta olacak şekilde yan yana ve üst üste dizilmiştir.
- Kutular çeşitlidir ve boyutları farklıdır.
- Kutuların ve konteynerin ağırlıkları iki boyutlu çözümlerde göz ardı edilecektir.
- Kutulardaki yüklerin özellikleri (yanıcı maddeler, kırılabilir ürünler vb.)önemli değildir dolayısıyla istiflenebilir ürünler olarak varsayılmıştır.
- Kutulardaki yükler konteynerlere atanırken herhangi bir önceliğe sahip değildir.
- Yükler konteynere dengeli olarak yerleşmiştir.

## 5.4 2D-CAMFA Uygulaması

İki Boyutlu Kombinatoryel Alan Maksimizasyon Yerleştirme Algoritması ve En Uygun Azalan Raf Yerleştirme Algoritması firmanın üç boyutlu filtre paketlerine uygulanmıştır (Şekil 5.3). 1776 adet filtre paketi tüm konteynerlere atanmıştır. Buna göre S-BFD Algoritması ile bu paketler 8 adet konteynere atanmıştır. Bu atama yapılırken paketlerin üçüncü boyutu sisteme girmemiştir. 2D-CAMFA modeline göre ise konteyner içi maksimum alan kullanılmış ve bu veriler filtre çeşidine göre kategorize edilmiştir. Elde edilen sonuçlara göre paketler 5 adet konteynere atanmıştır (Ek A-2).

# BÖLÜM 6

## SONUÇLAR VE ÖNERİLER

Konteyner yükleme problemleri içerdiği kısıtlar ve karmaşık yük yapısından ötürü lojistik operasyonlarında en çok karşılaşılan problemlerdir. Etkin konteyner yükleme, lojistik ve nakliye masraflarını önemli ölçüde azaltma potansiyeline sahiptir. Konteyner yükleme problemi, hesaplama açısından karmaşıktır ve kapsamlı akademik çabaya rağmen algoritmalar tam anlamıyla gerçeğe uygun çalışmamaktadır. Gerçek problemler, öncelikle çok sayıda karmaşık kısıtlama ve hedefler yüzünden tatmin edici bir şekilde ele alınamamaktadır. Bu tez çalışmasında konteyner yükleme problemleri kutulama algoritmalarıyla çözülmeğe çalışılmış ve 2D-CAMFA modeli geliştirilmiştir.

Literatürde iki boyutlu kutulama problemlerinin çözümüne yönelik araştırmacılar tarafından geliştirilen S-BFD algoritması, Python 2.7 programında modellenmiş, Ek(A-1) ve Ek(A-2) sonuçları elde edilmiştir. Bu sonuçlara göre bu algoritmada bazı eksiklikler görülmüş ve 2D-CAMFA modeli tasarlanmıştır. 2D-CAMFA nesneleri kullanıcının istediği herhangi bir kısıta(ağırlık, ürün çeşidi, üçüncü boyut vb.) göre gruplayan, kutuların konteynere yerleşebilmesinitek tek denemek yerine, bunların tüm kombinasyonlarını yapan, konteyner içini maksimum seviyede doldurmak için bu kombinasyonlardan maksimum alanı seçip raflara yerleştiren, rafın en altına en yüksek alanlı kombinasyonu yerleştirdiği için konteynerde yük dengesini sağlayan bir modeldir. Bu model Python 2.7 programında yazılmış ve rassal veriler bu modelde çalıştırıp Ek(A-1) sonuçları elde edilmiştir. 2D-CAMFA, S-BFD algoritmasına göre %14 daha hızlı çalıştığı saptanmıştır. Bu sonuçlara göre 250 adet rassal boyutlu seçilen kutular 82 adet konteynere yerleşmiş ve aynı kutular 2D-CAMFA

Modeli ile 80 adet konteynere yerleştirildiği kanıtlanmıştır. Böylece maksimum alan kullanarak minimum konteyner kullanılmış ve bu durum lojistik operasyon maliyetini düşürmüştür.

Bu tez çalışmasında geliştirilen 2D-CAMFA algoritması gerçek bir firmanın Konteyner Yükleme Operasyonlarına uygulanmış ve kesin algoritmalarda kullanılan konteyner sayısından daha az konteyner sayısı edilmiştir Ek(A-2). Bu sonuçlara göre Shelf-Best Fit Algoritması 1776 adet kutuyu 8 adet konteyenere yerleştirirken, geliştirilen 2D-CAMFA modeli aynı adet kutuyu 5 konteynere yerleştirmiştir.

Sonuç olarak bu çalışmanın amacı; konteyner yükleme problemlerine kutulama algoritmalarının uygulanması ve yükleri konteynerlere geliştirilen 2D-CAMFA modeliyle 2 boyutlu ağırlık, ürün çeşidi yada derinliklerine göre konteyner içinde alan ençoklama yaparak yerleştirilip optimum konteyner sayısını bulmaktır. Bu çalışmanın sonucunda geliştirilen 2D-CAMFA algoritmasının S-BFD iki boyutlu kutulama algoritmasından daha optimum konteyner içinde alan doldurduğu ve optimum konteyner sayısı verdiği kanıtlanmıştır.

İleriki çalışmalarda amacımız bu kutuların konteynerlere gerçeğe daha uygun bir şekilde yerleştirilmesi için kısıtları arttırmak, 2D-CAMFA modelini Üç Boyutlu konteyner yükleme problemlerine uygulamak ve Kombinasyonlu paket seçimi yaparken Genetik Algoritma kullanmaktır.

## KAYNAKLAR


[1] Gilmore, P.C., Gomory,R.E., (1963).“A linear programming approach to the cutting stock problem”—part II, Oper. Res. 11:863–888.

[2] George, J.A., ve Robinson, D.F., (1980).“A heuristic for packing boxes into a container”, Computers and Operations Research, 7:147-156.

[3] Bischo E.E., Marriott,M.D., (1990). “A comparative evaluation of heuristics for container loading”, European Journal of Operational Research, 44:267.

[4] Liu, N.C. ve Chen, L.C., (1981). “A new algorithm for container loading”. In: Compsac ‘81 – Proceedings of the 5th International Computer Software and Application Conference of the IEEE,292–299.

[5] Wolke, A., (2015). “More than bin packing Dynamic resource allocation strategies incloud data centers Information Systems”,Mendley Data, 3.

[6] Coffman, E. G., Galambos, G., Martello, S., ve Vigo, D., (1999). “Bin packing approximation algorithms: Combinatorial analysis”,In Handbook of combinatorial optimization, 151-207.

[7] Paul A. K., ve Sahoo, B., (2015). “Dynamic Virtual Machine Placement in Cloud Computing” Doctora Tezi, National Institute of Technology, Rourkela.

[8] Wang N. , Lim A., Zhu W., (2013). “A multi-round partial beam search approach for the single container loading problem with shipment priority, International Journal of Production Economics”, 145(2): 531-540

[9] Geiger M. J., (2007). “Bin Packing Under Multiple Objectives a Heuristic Approximation Approach”, The Fourth International Conference on Evolutionary Multi-Criterion Optimization: Late Breaking Papers,53-56.

[10] Fernández A., GilC., Márquez, A. L.,Baños R.,Montoya, M. G., ve Parra, M., ( 2011). “A memetic algorithm for two-dimensional multi-objective bin-packing with constraints”, In Proceedings of the 13th annual conference companion on Genetic and evolutionary computation (GECCO '11), 341-346.

[11] Zitzler, E., ve Thiele, L., (1999). “Multiobjective evolutionary algorithms: Acomparative case study and the strength pareto approach”,IEEE Transactionson Evolutionary Computation, 3:257–271.

[12] Kang K., (2012),“A hybrid genetic algorithm with a new packing strategy for the three-dimensional bin packing problem”,Applied Mathematics and Computation,1287–1299.

[13] Bischoff E.E., Ratcliff, M.S.W., (1995). “Issues in the development of approaches to container loading”, 23(4):377-390.

[14] Lim A., Rodrigues, B., ve Yang, Y., ( 2005). "3-D container packing heuristics," Applied Intelligence, 22:125-134.

[15] Li X., Zhang K., (2015). “A hybrid differential evolution algorithm for multiple container loading problem with heterogeneous containers”, Computers & Industrial Engineering, 90:305-313.

[16] Liao C., Hsu C., (2013). “New lower bounds for the three-dimensional orthogonal bin packing problem”, European Journal of Operational Research, 225:244-252.

[17] Thapatsuwana P., Pongcharoena P.,Hicksb C., Chainatec W., (2012). “Development of a stochastic optimisation tool for solving the multiple container packing problems”,International Journal of Production Economics,737–748.

[18] Sheikhalishahi M., Wallace R.M., Grandinetti L., Vazquez-Poletti J.L., Guerriero F., (2016). “A multi-dimensional job scheduling”, Future Generation Computer Systems, 54:123-131.

[19] Choll A., Klein R., Jürgens C., (1997). “A fast hybrid procedure for exactly solving the one-dimensional bin packing problem”. Computers & Operations Research, 24:627-645.

[20] Chen C.S., Lee, S.M., Shen, Q.S., (1995).“An analytical model for the container loading problem”, European Journal of Operational Research, 80(1): 68-76.

[21] Martello S., D. Pisinger, and D. Vigo., (2000) “The three-dimensional bin packing problem”, Operation Research, 48:256–267.

[22] Coffman, E. G., Garey, M. R. and Johnson, D. S., (1996). “Approximation algorithms for bin-packing--an updated surve”, Algorithm Design for Computer System Design, 284:49-106.

[23] Martello S., Toth P., (1990).“Knapsack Problems : Algorithms and Computer Implementations”,Chichester: Wiley, 221.

[24] Eley, M., (2003). “Abottleneck assignment approach to the multiple container loading problem”, Operations Research Spectrum 25:45–60.

[25] Pisinger, D., ve M., Sigurd., (2007). “Using decomposition techniques and constraint programming for solving the two-dimensional bin-packing problem”,INFORMS Journal on Computing, 19:36–51.

[26] Gilmore P.C., Gomory,R.E., (1961). “A linear programming approach to the cutting stock problem”, Oper. Res. 9:849–859.

[27] Gilmore P.C., R.E. Gomory, (1965). “Multistage cutting problems of two and more dimensions”, Oper. Res. 13:94–119

[28] Chung, F. K. R., Garey,M. R., ve Johnson, D. S., (1982). “On packing two-dimensional bins”. SIAM Journal of Algebraic and Discrete Methods, 3:66–76.

[29] Berkey, J.O.,ve Wong P.Y., (1987). “Two dimensional finite bin packing algorithms”, Journal of Operational Research Society, 2:423—429

[30] Lodi, A., (1999). “Algorithms for Two-Dimensional Bin Packing and Assignment Problems”, Doctoral Tezi, Universita Degli Studi Di Bologna.

[31] Jacobs S., (1996). “On genetic algorithms for the packing ofpolygons”, European Journal of Operational Research, 165–181.

[32] Martello, S., Vigo, D., (1998).“Exact solution of the two-dimen-sional finite bin packing problem”, Management Science, 44:388–399.

[33] Araya, I., Riff, C., (2014). “A beam search approach to the container loading problem”, Computers & Operations Research, 43:100-107

[34] Jylänki J.,( 2010). “A Thousand Ways to Pack the Bin - A Practical Approach to Two-Dimensional Rectangle Bin Packing problem under home delivery service”, Asia–PacificJournal of Operational Research, 1–20.

[35] Chazelle B., (1983).“The bottomn-left bin-packing heuristic: An e-cient im-plementation” IEEE Transactions on Computers, 32:69-707

[36] Van de Voort, M., O'Brien, K.A., Rahman, A.,Valeri, (2003). “L. Seacurity: Improving the Security of the Global Sea-Container Shipping System”,Rand Coorparation.

[37] Toffolo Túlio A.M., Esprit E., Wauters T., Berghe G. V., (2017). “A two-dimensional heuristic decomposition approach to a three-dimensional multiple container loading problem”, European Journal of Operational Research, 257: (2):526-538.

[38] Araújo E. J., Chaves,A. A., Salles Neto, L. L., (2016). “Anibal Tavares de Azevedo, Pareto clustering search applied for 3D container ship loading plan problem”, Expert Systems with Applications, 44:50-57.

[39] Azevedo A.T.,Ribeiro C.M., Sena G.J., Chaves A.A, Neto L. L. S., Moretti A. C., (2014). “Solving the 3D container ship loading planning problem by representation by rules and meta-heuristics”. Int. J. Data Anal. Tech., 3:228-260.

[40] Bortfeldt A., (2004). “Conteiner loading problems-A state of art rewiew”,No 120007, FEMM Working Papers from Otto-von-Guericke University Magdeburg, Faculty of Economics and Management.

[41] Lim A. , Rodriguesb, B., Wangc Y., (2003). “A multi-faced buildup algorithm for three-dimensional packing problems”,Omega, 471–481

[42] Gonçalves, J.F., Mauricio, G.C., Resende, (2012), “A parallel multi-population biased random-key genetic algorithm for a container loading problem”, Computers & Operations Research, 39: 179-190

[43] Junqueira L., Morabito R., Yamashita D.S., (2012). “Three-dimensional container loading models with cargo stability and load bearing constraints”, Computers & Operations Research, 39:74-85.

[44] Lim, A., Ma H., Qiu, C., Zhu, W., (2013). “The single container loading problem with axle weight constraints, International Journal of Production Economics”, 144: 358-369

[45] Che, C. H.,Huang, W., Lim, A.,Zhu, W., (2011). “The multiple container loading cost minimization problem, European Journal of Operational Research”, 214.

[46] Sheng, L., Tan, W., Xu, Z., Liu, X., (2014). “A tree search algorithm for the container loading problem”, Computers & Industrial Engineering, 75:20-30

[47] Beyaz, M., Dokeroglu T., Cosar, A., (2015).“Robust hyper-heuristic algorithms for the offline oriented/non-oriented 2D bin packing problems”, Applied Soft Computing, 36:236-245.

[48] Ji, M., Guo, W., Zhu, H.,Yang, Y., (2015). “Optimization of loading sequence and rehandling strategy for multi-quay crane operations in container terminals”, Transportation Research Part E: Logistics and Transportation Review, 80: 1-19.

[49] Wei L., Zhu W.,Lim A., (2015). “A goal-driven prototype column generation strategy for the multiple container loading cost minimization problem”, European Journal of Operational Research, 241(1):39-49

[50] Zheng, J., Chien, C., Gen, M., (2015). “Multi-objective multi population biased random-key genetic algorithm for the 3-D container loading problem”, Computers & Industrial Engineering, 89: 80-87,

[51] Sheng, L., Zhao, H., Dong, X., Cheng., C., (2016). “A heuristic algorithm for container loading of pallets with infill boxes”, European Journal of Operational Research, 252:728–736.

[52] Wascher, G., Haubner, H., Schumann, H., (2007)."An improved typology of cutting andpacking problems," European Journal of Operational Research, 183:1109-1130.

[53] Gehring, H., Bortfeldt, A., (1997). “Agenetic algorithm for solving the container loading problem”. International Transactions in Operational Research, 4:401–418.

[54] Bortfeldt, A., (2011).“A hybrid algorithm for the capacitated vehicle routing problem with three-dimensional loading constraints”, Computers and Operations Research, 39(9):2248-2257.

[55] Wang, L., Guo, S., Chen, S., Zhu, W., Lim, A., (2010). “Two natural heuristics for 3D packing with practical loading constraints”, Zhang, B.-T., Orgun, M.A. (Eds.),Lecture Notes on Artificial Intelligence, 256–267.

[56] Dereli, Z.T., Das, G.S., (2010). “A hybrid simulated annealing algorithm for solvingmulti-objective container-loading problems”, Applied Artificial Intelligence 24: 463–486.

[57] Gehring, H., Menschner, K., Meyer, M., (1990). “A computer-based heuristic for packing pooled shipment containers”, European Journal of Operational Research ,44(2): 277–288

[58] Liu, J., Yue, Y., Dong, Z., Maple, C., Keech, M., (2011). “Anovel hybrid Tabu Search approach the container loading”. Computers &Operations Research 38:797–807.

[59] Parreño, F., Alvarez-Valdez R., Tamarit, J.M., Oliveira, J.F., (2008). “Amaximal-space algorithm for the container loading problem”. INFORMS Journal on Computing 20:412–422.

[60] Pisinge,r D., ( 2002). “Heuristics for the container loading problem”,European Journal of Operational Research, 141(2):382-392.

## EK-A

## VERİLER

**Rassal Veriler ve 2B-CAMFA Sonuçları**

SHELF BEST FİT SONUÇLARI RASSAL VERİ

{1: {1: [[number:200.0, number:56.0, number:248.0, number:118.0, number:179.0]]}, 2: {1: [[number:21.0, number:171.0]]}, 3: {1: [[number:130.0]]}, 4: {1: [[number:127.0]]}, 5: {1: [[number:68.0, number:70.0]]}, 6: {1: [[number:32.0]]}, 7: {1: [[number:247.0, number:110.0, number:165.0]], 2: [[number:151.0, number:129.0]]}, 8: {1: [[number:119.0, number:50.0]], 2: [[number:25.0]]}, 9: {1: [[number:125.0]], 2: [[number:236.0]]}, 10: {1: [[number:55.0]], 2: [[number:147.0]]}, 11: {1: [[number:178.0]], 2: [[number:187.0, number:142.0]]}, 12: {1: [[number:146.0]], 2: [[number:164.0, number:166.0]]}, 13: {1: [[number:143.0, number:103.0]], 2: [[number:2.0]]}, 14: {1: [[number:249.0, number:132.0]], 2: [[number:40.0]]}, 15: {1: [[number:128.0]], 2: [[number:73.0]]}, 16: {1: [[number:66.0, number:209.0]], 2: [[number:22.0, number:33.0]]}, 17: {1: [[number:190.0, number:91.0]], 2: [[number:58.0]]}, 18: {1: [[number:189.0]], 2: [[number:152.0]]}, 19: {1: [[number:141.0, number:140.0]], 2: [[number:108.0]]}, 20: {1: [[number:64.0, number:74.0]], 2: [[number:149.0, number:107.0, number:162.0]]}, 21: {1: [[number:112.0, number:154.0]], 2: [[number:231.0, number:175.0]]}, 22: {1: [[number:150.0, number:95.0]], 2: [[number:19.0, number:57.0, number:184.0, number:182.0]]}, 23: {1: [[number:155.0, number:198.0]], 2: [[number:102.0, number:235.0]]}, 24: {1: [[number:168.0]], 2: [[number:100.0, number:160.0]]}, 25: {1: [[number:53.0, number:113.0]], 2: [[number:14.0, number:239.0, number:213.0]]}, 26: {1: [[number:120.0]], 2: [[number:87.0]]}, 27: {1: [[number:80.0]], 2: [[number:244.0]]}, 28: {1: [[number:199.0, number:15.0]], 2: [[number:148.0]]}, 29: {1: [[number:5.0, number:31.0, number:216.0]], 2: [[number:145.0, number:173.0]]}, 30: {1: [[number:48.0, number:215.0]], 2: [[number:17.0]]}, 31: {1: [[number:161.0]], 2: [[number:133.0, number:16.0]]}, 32: {1: [[number:212.0]], 2: [[number:51.0, number:52.0]]}, 33: {1: [[number:11.0, number:109.0, number:214.0]], 2: [[number:122.0]]}, 34: {1: [[number:220.0, number:176.0]], 2: [[number:188.0, number:27.0]]}, 35: {1: [[number:46.0,

number:196.0, number:136.0]], 2: [[number:1.0]]}, 36: {1: [[number:232.0]], 2: [[number:3.0]]}, 37: {1: [[number:163.0]], 2: [[number:63.0, number:106.0]]}, 38: {1: [[number:226.0]], 2: [[number:137.0]]}, 39: {1: [[number:117.0]], 2: [[number:26.0, number:206.0]]}, 40: {1: [[number:210.0, number:79.0]], 2: [[number:13.0]]}, 41: {1: [[number:94.0]], 2: [[number:157.0, number:250.0, number:191.0]]}, 42: {1: [[number:227.0]], 2: [[number:44.0]]}, 43: {1: [[number:207.0]], 2: [[number:134.0, number:9.0, number:245.0]]}, 44: {1: [[number:104.0, number:105.0, number:174.0]], 2: [[number:30.0, number:41.0]]}, 45: {1: [[number:225.0]], 2: [[number:76.0]]}, 46: {1: [[number:172.0]], 2: [[number:28.0]]}, 47: {1: [[number:238.0]], 2: [[number:84.0, number:83.0]]}, 48: {1: [[number:8.0]], 2: [[number:241.0, number:208.0]]}, 49: {1: [[number:242.0]], 2: [[number:159.0, number:123.0]]}, 50: {1: [[number:42.0]], 2: [[number:39.0, number:60.0]], 3: [[number:121.0]]}, 51: {1: [[number:77.0]], 2: [[number:233.0]]}, 52: {1: [[number:23.0, number:67.0, number:246.0, number:71.0, number:217.0]], 2: [[number:86.0]]}, 53: {1: [[number:54.0]], 2: [[number:192.0]]}, 54: {1: [[number:92.0]], 2: [[number:205.0]]}, 55: {1: [[number:204.0]], 2: [[number:18.0, number:37.0]]}, 56: {1: [[number:219.0]], 2: [[number:135.0]]}, 57: {1: [[number:234.0]], 2: [[number:193.0]], 3: [[number:97.0]]}, 58: {1: [[number:72.0]], 2: [[number:10.0]], 3: [[number:82.0, number:6.0, number:185.0]]}, 59: {1: [[number:243.0]], 2: [[number:240.0]], 3: [[number:228.0]]}, 60: {1: [[number:223.0]], 2: [[number:81.0]], 3: [[number:237.0]]}, 61: {1: [[number:230.0]], 2: [[number:194.0, number:34.0]]}, 62: {1: [[number:43.0]], 2: [[number:180.0]]}, 63: {1: [[number:144.0, number:98.0, number:167.0]], 2: [[number:138.0]]}, 64: {1: [[number:101.0]], 2: [[number:229.0, number:85.0]]}, 65: {1: [[number:38.0, number:20.0, number:131.0]], 2: [[number:89.0, number:116.0]]}, 66: {1: [[number:4.0, number:126.0]], 2: [[number:59.0]], 3: [[number:156.0]], 4: [[number:124.0]]}, 67: {1: [[number:12.0]], 2: [[number:29.0]], 3: [[number:75.0]]}, 68: {1: [[number:221.0, number:65.0, number:45.0, number:93.0]], 2: [[number:24.0]]}, 69: {1: [[number:183.0]], 2: [[number:186.0, number:88.0, number:201.0]]}, 70: {1: [[number:170.0]], 2: [[number:177.0, number:197.0, number:7.0]]}, 71: {1: [[number:224.0]], 2: [[number:115.0]]}, 72: {1: [[number:47.0]], 2: [[number:195.0]]}, 73: {1: [[number:99.0]], 2: [[number:218.0]]}, 74: {1: [[number:114.0]], 2: [[number:96.0]]}, 75: {1: [[number:35.0]], 2: [[number:36.0, number:222.0]]}, 76: {1: [[number:153.0]], 2: [[number:49.0, number:78.0]]}, 77: {1: [[number:158.0]], 2: [[number:139.0]]}, 78: {1: [[number:181.0]], 2: [[number:69.0, number:211.0]]}, 79: {1: [[number:169.0]], 2: [[number:202.0]]}, 80: {1: [[number:203.0]], 2: [[number:62.0, number:61.0]]}, 81: {1: [[number:90.0]]}, 82: {1: [[number:111.0]]}}

algoritma sonucu min kutu sayısı  = 82

0.0

svmem(total=8487583744L, available=4768141312L, percent=43.8, used=3719442432L, free=4768141312L)

```
In [220]: print "algoritma sonucu min kutu sayısı  = %s" % kutu
algoritma sonucu min kutu sayısı  = 82
```

**2D-CAMFA RASSAL SONUÇLARI**

| | x | y | z | box | shelf |
|---|---|---|---|---|---|
| **9** | 16 | 4 | 1 | 80 | 2 |
| **205** | 20 | 4 | 1 | 80 | 1 |
| **70** | 2 | 2 | 1 | 79 | 2 |
| **88** | 8 | 10 | 2 | 79 | 1 |
| **99** | 11 | 17 | 1 | 79 | 2 |
| **146** | 2 | 1 | 1 | 79 | 2 |
| **184** | 4 | 2 | 1 | 79 | 2 |
| **192** | 19 | 11 | 1 | 79 | 3 |
| **233** | 5 | 17 | 1 | 79 | 1 |
| **246** | 15 | 17 | 1 | 79 | 1 |
| **1** | 18 | 9 | 2 | 78 | 4 |
| **87** | 12 | 13 | 2 | 78 | 1 |
| **98** | 2 | 15 | 2 | 78 | 1 |
| **129** | 4 | 12 | 2 | 78 | 2 |
| **133** | 4 | 9 | 2 | 78 | 2 |
| **212** | 12 | 13 | 2 | 78 | 2 |
| **217** | 19 | 11 | 2 | 78 | 3 |
| **247** | 6 | 14 | 2 | 78 | 1 |
| **71** | 4 | 15 | 2 | 77 | 1 |
| **123** | 15 | 14 | 2 | 77 | 1 |
| **126** | 15 | 3 | 2 | 77 | 2 |
| **170** | 13 | 8 | 2 | 75 | 2 |
| **243** | 20 | 12 | 2 | 75 | 1 |
| **4** | 3 | 3 | 3 | 73 | 1 |
| **18** | 14 | 1 | 3 | 73 | 1 |
| **187** | 3 | 2 | 2 | 73 | 2 |
| **225** | 20 | 15 | 2 | 73 | 1 |
| **231** | 10 | 5 | 2 | 73 | 2 |
| **12** | 19 | 7 | 3 | 72 | 1 |
| **84** | 14 | 5 | 3 | 72 | 2 |
| **158** | 2 | 8 | 3 | 72 | 2 |
| **239** | 3 | 9 | 3 | 72 | 2 |
| **35** | 2 | 15 | 3 | 70 | 1 |
| **102** | 3 | 17 | 3 | 70 | 1 |
| **136** | 6 | 16 | 3 | 70 | 1 |
| **156** | 3 | 20 | 3 | 70 | 1 |
| **163** | 5 | 9 | 3 | 70 | 1 |
| **46** | 13 | 4 | 4 | 69 | 1 |
| **131** | 5 | 5 | 4 | 69 | 1 |
| **181** | 18 | 2 | 4 | 69 | 2 |
| **13** | 6 | 19 | 4 | 68 | 1 |
| **172** | 12 | 12 | 4 | 68 | 1 |

| | | | | | |
|---|---|---|---|---|---|
| **19** | 16 | 20 | 4 | 66 | 1 |
| **40** | 17 | 19 | 4 | 65 | 2 |
| **185** | 1 | 1 | 4 | 65 | 2 |
| **195** | 7 | 20 | 4 | 64 | 1 |
| **200** | 5 | 12 | 4 | 64 | 1 |
| **223** | 8 | 19 | 4 | 64 | 1 |
| **127** | 18 | 6 | 5 | 63 | 1 |
| **198** | 4 | 12 | 5 | 63 | 2 |
| **28** | 7 | 15 | 5 | 62 | 3 |
| **49** | 5 | 11 | 5 | 62 | 2 |
| **65** | 7 | 16 | 5 | 62 | 1 |
| **83** | 11 | 9 | 5 | 62 | 3 |
| **90** | 7 | 19 | 5 | 62 | 1 |
| **125** | 4 | 4 | 5 | 62 | 4 |
| **144** | 4 | 4 | 5 | 62 | 4 |
| **151** | 6 | 17 | 5 | 62 | 1 |
| **232** | 15 | 13 | 5 | 62 | 2 |
| **79** | 15 | 19 | 5 | 60 | 1 |
| **219** | 4 | 20 | 5 | 60 | 1 |
| **148** | 19 | 5 | 6 | 59 | 1 |
| **213** | 20 | 1 | 6 | 59 | 2 |
| **58** | 5 | 5 | 6 | 58 | 1 |
| **67** | 8 | 12 | 6 | 58 | 1 |
| **68** | 7 | 16 | 6 | 58 | 1 |
| **95** | 8 | 3 | 6 | 58 | 2 |
| **138** | 11 | 9 | 6 | 58 | 3 |
| **50** | 9 | 14 | 6 | 57 | 1 |
| **128** | 1 | 6 | 6 | 57 | 1 |
| **180** | 19 | 4 | 6 | 57 | 2 |
| **241** | 9 | 16 | 6 | 57 | 1 |
| **37** | 18 | 12 | 7 | 56 | 1 |
| **69** | 17 | 5 | 7 | 56 | 2 |
| **168** | 3 | 3 | 7 | 56 | 2 |
| **104** | 14 | 16 | 7 | 54 | 2 |
| **160** | 7 | 4 | 7 | 54 | 3 |
| **226** | 11 | 4 | 7 | 54 | 3 |
| **238** | 6 | 7 | 7 | 54 | 2 |
| **86** | 1 | 3 | 7 | 53 | 1 |
| **106** | 4 | 12 | 7 | 53 | 1 |
| **188** | 15 | 20 | 7 | 53 | 1 |
| **210** | 12 | 2 | 8 | 52 | 2 |
| **222** | 12 | 6 | 8 | 52 | 1 |
| **10** | 15 | 10 | 8 | 51 | 1 |

| | | | | | |
|---|---|---|---|---|---|
| **140** | 15 | 6 | 8 | 51 | 2 |
| **97** | 10 | 3 | 8 | 50 | 1 |
| **147** | 9 | 16 | 8 | 50 | 1 |
| **169** | 15 | 2 | 8 | 50 | 2 |
| **8** | 18 | 15 | 8 | 49 | 2 |
| **61** | 6 | 1 | 8 | 49 | 3 |
| **82** | 1 | 19 | 8 | 49 | 2 |
| **199** | 14 | 1 | 8 | 49 | 3 |
| **173** | 15 | 17 | 8 | 48 | 1 |
| **197** | 5 | 20 | 8 | 48 | 1 |
| **66** | 12 | 13 | 9 | 47 | 1 |
| **137** | 4 | 3 | 9 | 47 | 1 |
| **166** | 17 | 1 | 9 | 47 | 2 |
| **5** | 20 | 3 | 9 | 46 | 2 |
| **108** | 12 | 14 | 9 | 46 | 1 |
| **203** | 5 | 3 | 9 | 46 | 1 |
| **0** | 18 | 4 | 9 | 45 | 2 |
| **72** | 3 | 6 | 9 | 45 | 1 |
| **177** | 15 | 14 | 9 | 45 | 1 |
| **23** | 19 | 13 | 9 | 44 | 2 |
| **44** | 14 | 4 | 9 | 44 | 3 |
| **162** | 4 | 6 | 9 | 44 | 3 |
| **2** | 4 | 14 | 9 | 43 | 1 |
| **53** | 6 | 19 | 9 | 43 | 1 |
| **207** | 4 | 20 | 9 | 43 | 1 |
| **214** | 5 | 19 | 9 | 43 | 1 |
| **234** | 1 | 6 | 9 | 43 | 1 |
| **81** | 10 | 3 | 10 | 42 | 2 |
| **117** | 18 | 2 | 10 | 42 | 1 |
| **73** | 17 | 3 | 10 | 41 | 2 |
| **208** | 13 | 5 | 10 | 41 | 1 |
| **24** | 9 | 10 | 10 | 40 | 1 |
| **48** | 10 | 10 | 10 | 40 | 1 |
| **218** | 17 | 5 | 10 | 40 | 2 |
| **51** | 7 | 18 | 10 | 38 | 1 |
| **76** | 5 | 12 | 10 | 38 | 1 |
| **159** | 8 | 19 | 10 | 38 | 1 |
| **215** | 3 | 1 | 10 | 38 | 2 |
| **91** | 15 | 20 | 10 | 36 | 1 |
| **100** | 6 | 9 | 11 | 36 | 1 |
| **228** | 4 | 20 | 10 | 36 | 1 |
| **143** | 16 | 10 | 11 | 35 | 2 |
| **179** | 19 | 10 | 11 | 35 | 1 |

| | | | | | |
|---|---|---|---|---|---|
| **3** | 15 | 13 | 11 | 34 | 1 |
| **101** | 15 | 7 | 11 | 34 | 2 |
| **130** | 7 | 5 | 11 | 33 | 2 |
| **132** | 19 | 14 | 11 | 33 | 1 |
| **174** | 4 | 6 | 11 | 33 | 2 |
| **47** | 2 | 19 | 11 | 32 | 1 |
| **54** | 7 | 15 | 11 | 32 | 1 |
| **92** | 19 | 16 | 11 | 32 | 2 |
| **96** | 9 | 3 | 11 | 32 | 4 |
| **141** | 10 | 15 | 11 | 32 | 3 |
| **186** | 11 | 16 | 11 | 32 | 1 |
| **229** | 8 | 16 | 11 | 32 | 3 |
| **249** | 1 | 13 | 11 | 32 | 2 |
| **224** | 15 | 1 | 12 | 31 | 1 |
| **235** | 14 | 1 | 12 | 31 | 2 |
| **62** | 19 | 2 | 12 | 30 | 1 |
| **167** | 20 | 1 | 12 | 30 | 2 |
| **34** | 10 | 5 | 12 | 29 | 3 |
| **152** | 17 | 5 | 12 | 29 | 2 |
| **153** | 20 | 6 | 12 | 29 | 1 |
| **52** | 19 | 3 | 12 | 28 | 2 |
| **142** | 14 | 6 | 12 | 28 | 1 |
| **237** | 4 | 16 | 12 | 28 | 1 |
| **56** | 3 | 1 | 12 | 27 | 3 |
| **64** | 10 | 8 | 12 | 27 | 3 |
| **201** | 18 | 11 | 12 | 27 | 2 |
| **216** | 7 | 9 | 12 | 27 | 3 |
| **30** | 12 | 15 | 13 | 26 | 1 |
| **105** | 20 | 20 | 12 | 26 | 1 |
| **16** | 12 | 8 | 13 | 25 | 2 |
| **32** | 17 | 10 | 13 | 25 | 3 |
| **75** | 6 | 3 | 13 | 25 | 2 |
| **190** | 18 | 11 | 13 | 25 | 1 |
| **196** | 2 | 3 | 13 | 25 | 1 |
| **6** | 3 | 9 | 13 | 24 | 3 |
| **17** | 2 | 2 | 14 | 24 | 2 |
| **77** | 3 | 5 | 13 | 24 | 3 |
| **89** | 20 | 18 | 13 | 24 | 2 |
| **110** | 12 | 18 | 13 | 24 | 3 |
| **111** | 3 | 19 | 14 | 24 | 2 |
| **121** | 14 | 2 | 14 | 24 | 2 |
| **189** | 1 | 6 | 13 | 24 | 3 |
| **85** | 8 | 11 | 14 | 23 | 3 |

| | | | | | |
|---|---|---|---|---|---|
| **171** | 20 | 9 | 14 | 23 | 2 |
| **204** | 12 | 3 | 14 | 23 | 3 |
| **93** | 17 | 15 | 14 | 22 | 1 |
| **113** | 3 | 20 | 14 | 22 | 1 |
| **124** | 11 | 19 | 14 | 21 | 1 |
| **175** | 8 | 16 | 14 | 21 | 1 |
| **244** | 18 | 1 | 14 | 21 | 2 |
| **15** | 12 | 4 | 15 | 20 | 3 |
| **245** | 6 | 6 | 15 | 20 | 3 |
| **248** | 16 | 7 | 15 | 20 | 2 |
| **14** | 7 | 15 | 15 | 19 | 1 |
| **57** | 20 | 5 | 15 | 19 | 2 |
| **122** | 9 | 8 | 15 | 19 | 1 |
| **242** | 4 | 10 | 15 | 19 | 1 |
| **119** | 16 | 18 | 15 | 18 | 1 |
| **134** | 3 | 18 | 15 | 18 | 1 |
| **11** | 5 | 1 | 15 | 16 | 2 |
| **45** | 15 | 19 | 15 | 16 | 1 |
| **55** | 5 | 12 | 15 | 16 | 1 |
| **27** | 19 | 19 | 15 | 15 | 1 |
| **59** | 16 | 1 | 15 | 15 | 2 |
| **36** | 16 | 6 | 16 | 14 | 1 |
| **42** | 4 | 6 | 16 | 14 | 1 |
| **135** | 5 | 8 | 16 | 14 | 2 |
| **139** | 14 | 5 | 16 | 14 | 2 |
| **165** | 7 | 4 | 16 | 14 | 3 |
| **21** | 11 | 9 | 16 | 13 | 2 |
| **107** | 8 | 8 | 16 | 13 | 2 |
| **150** | 19 | 10 | 16 | 13 | 1 |
| **29** | 5 | 18 | 16 | 12 | 2 |
| **31** | 12 | 7 | 17 | 12 | 1 |
| **63** | 8 | 13 | 16 | 12 | 2 |
| **74** | 5 | 11 | 16 | 12 | 2 |
| **116** | 18 | 14 | 16 | 12 | 1 |
| **209** | 2 | 17 | 16 | 12 | 1 |
| **38** | 9 | 12 | 17 | 11 | 1 |
| **39** | 3 | 5 | 17 | 11 | 2 |
| **103** | 13 | 1 | 17 | 11 | 2 |
| **114** | 1 | 6 | 17 | 11 | 1 |
| **157** | 11 | 10 | 18 | 11 | 1 |
| **211** | 10 | 14 | 17 | 11 | 1 |
| **230** | 16 | 12 | 17 | 11 | 3 |
| **22** | 10 | 14 | 18 | 10 | 1 |

| | | | | | |
|---|---|---|---|---|---|
| **25** | 11 | 13 | 18 | 9 | 1 |
| **164** | 12 | 1 | 18 | 9 | 2 |
| **155** | 12 | 18 | 18 | 8 | 1 |
| **161** | 17 | 2 | 18 | 8 | 2 |
| **94** | 16 | 6 | 18 | 7 | 2 |
| **183** | 17 | 14 | 18 | 7 | 1 |
| **240** | 2 | 9 | 18 | 7 | 1 |
| **112** | 4 | 15 | 18 | 6 | 1 |
| **118** | 15 | 16 | 18 | 6 | 1 |
| **182** | 13 | 4 | 18 | 6 | 2 |
| **202** | 6 | 3 | 18 | 6 | 2 |
| **26** | 19 | 9 | 19 | 5 | 1 |
| **109** | 14 | 2 | 19 | 5 | 2 |
| **20** | 3 | 12 | 19 | 4 | 4 |
| **41** | 20 | 11 | 19 | 4 | 3 |
| **120** | 8 | 3 | 19 | 4 | 5 |
| **149** | 15 | 12 | 19 | 4 | 4 |
| **154** | 10 | 3 | 19 | 4 | 5 |
| **176** | 2 | 19 | 19 | 4 | 2 |
| **193** | 6 | 14 | 19 | 4 | 2 |
| **221** | 11 | 14 | 19 | 4 | 2 |
| **227** | 1 | 5 | 19 | 4 | 2 |
| **60** | 7 | 12 | 20 | 3 | 2 |
| **78** | 20 | 2 | 19 | 3 | 3 |
| **80** | 1 | 15 | 19 | 3 | 2 |
| **145** | 6 | 18 | 19 | 3 | 2 |
| **191** | 20 | 10 | 20 | 3 | 1 |
| **194** | 10 | 11 | 20 | 3 | 2 |
| **206** | 13 | 17 | 19 | 3 | 2 |
| **7** | 18 | 14 | 20 | 2 | 2 |
| **33** | 2 | 4 | 20 | 2 | 3 |
| **43** | 1 | 14 | 20 | 2 | 2 |
| **236** | 15 | 2 | 20 | 2 | 3 |
| **115** | 3 | 9 | 20 | 1 | 1 |
| **178** | 13 | 19 | 20 | 1 | 1 |
| **220** | 4 | 20 | 20 | 1 | 1 |

**Gerçekler Veriler ve 2D-CAMFA ve S-BFD Sonuçları**

SHELF BEST FİT SONUÇLARI GERÇEK VERİ

{1: {1: [[number:747.0, number:748.0, number:750.0, number:749.0, number:751.0, number:1761.0, number:1758.0, number:1759.0, number:1762.0, number:1763.0, number:1764.0, number:1765.0, number:1760.0, number:726.0, number:725.0, number:727.0, number:728.0, number:729.0, number:1773.0, number:730.0, number:732.0, number:733.0]], 2: [[number:731.0, number:734.0, number:1774.0, number:1775.0, number:1772.0, number:425.0, number:426.0, number:430.0, number:431.0, number:432.0, number:434.0, number:437.0, number:438.0, number:439.0, number:440.0, number:442.0, number:443.0, number:444.0, number:445.0, number:447.0, number:448.0, number:449.0, number:427.0, number:428.0, number:429.0, number:433.0, number:435.0, number:436.0, number:441.0, number:446.0, number:735.0]], 3: [[number:737.0, number:738.0, number:739.0, number:206.0, number:207.0, number:208.0, number:209.0, number:210.0, number:211.0, number:212.0, number:213.0, number:214.0, number:215.0, number:216.0, number:217.0, number:218.0, number:219.0, number:220.0, number:221.0, number:222.0, number:223.0, number:224.0, number:205.0, number:62.0, number:106.0, number:107.0, number:108.0, number:109.0, number:110.0, number:111.0, number:112.0, number:113.0, number:114.0, number:115.0]], 4: [[number:117.0, number:118.0, number:119.0, number:120.0, number:121.0, number:122.0, number:123.0, number:124.0, number:125.0, number:126.0, number:127.0, number:128.0, number:129.0, number:130.0, number:131.0, number:132.0, number:133.0, number:134.0, number:135.0, number:136.0, number:137.0, number:138.0, number:139.0, number:140.0, number:141.0, number:142.0, number:143.0, number:144.0, number:145.0, number:146.0, number:147.0, number:148.0, number:149.0, number:150.0, number:151.0]], 5: [[number:153.0, number:154.0, number:155.0, number:159.0, number:157.0, number:156.0, number:158.0, number:160.0, number:161.0, number:162.0, number:163.0, number:164.0, number:165.0, number:166.0, number:167.0, number:168.0, number:170.0, number:171.0, number:172.0, number:173.0, number:174.0, number:175.0, number:176.0, number:177.0, number:178.0, number:179.0, number:180.0, number:181.0, number:182.0, number:183.0, number:184.0, number:185.0, number:186.0, number:188.0, number:189.0]], 6: [[number:191.0, number:192.0, number:169.0, number:193.0, number:194.0, number:195.0, number:196.0, number:197.0, number:198.0, number:200.0, number:201.0, number:202.0, number:203.0, number:204.0, number:50.0, number:51.0, number:52.0, number:53.0, number:54.0, number:55.0, number:56.0, number:57.0, number:58.0, number:59.0, number:60.0, number:61.0, number:63.0, number:64.0, number:65.0, number:66.0, number:67.0, number:68.0, number:69.0, number:70.0, number:71.0]]}, 2: {1: [[number:1771.0, number:736.0, number:116.0, number:152.0, number:190.0, number:72.0, number:73.0, number:74.0, number:75.0, number:76.0, number:77.0, number:78.0, number:79.0, number:80.0, number:81.0, number:82.0, number:83.0, number:84.0, number:85.0, number:86.0, number:87.0, number:88.0, number:89.0, number:90.0, number:91.0, number:92.0, number:93.0, number:94.0, number:95.0, number:96.0, number:97.0, number:98.0, number:99.0, number:100.0, number:101.0]], 2: [[number:103.0, number:104.0, number:105.0, number:187.0, number:199.0, number:1308.0, number:1310.0, number:1311.0, number:1313.0, number:1314.0, number:1317.0, number:1318.0, number:1319.0, number:1322.0, number:1325.0, number:1327.0, number:1328.0, number:1329.0, number:1331.0, number:1334.0, number:1303.0, number:1338.0, number:1339.0, number:1342.0, number:1304.0, number:1344.0, number:1345.0, number:1349.0,

number:1351.0, number:1352.0, number:1353.0, number:1355.0, number:1306.0, number:1315.0, number:1307.0, number:1309.0]], 3: [[number:1312.0, number:1320.0, number:1321.0, number:1323.0, number:1324.0, number:1326.0, number:1330.0, number:1332.0, number:1333.0, number:1335.0, number:1336.0, number:1337.0, number:1340.0, number:1341.0, number:1343.0, number:1346.0, number:1347.0, number:1348.0, number:1350.0, number:1354.0, number:1356.0, number:1357.0, number:1358.0, number:1359.0, number:1360.0, number:1361.0, number:1362.0, number:1363.0, number:1364.0, number:1365.0, number:1366.0, number:1367.0, number:1368.0, number:1369.0, number:1370.0, number:1371.0, number:1372.0]], 4: [[number:1374.0, number:1375.0, number:1376.0, number:1302.0, number:1305.0, number:1707.0, number:1708.0, number:1709.0, number:1710.0, number:1711.0, number:1712.0, number:1713.0, number:1714.0, number:1715.0, number:1716.0, number:1177.0, number:1178.0, number:1179.0, number:1180.0, number:1181.0, number:1182.0, number:1183.0, number:1184.0, number:1185.0, number:1186.0, number:1187.0, number:1188.0, number:1189.0, number:1190.0, number:1191.0, number:1192.0, number:1193.0, number:1194.0, number:1195.0]], 5: [[number:1197.0, number:1198.0, number:1199.0, number:1200.0, number:1201.0, number:1107.0, number:1108.0, number:1110.0, number:1111.0, number:1112.0, number:1113.0, number:1119.0, number:1124.0, number:1127.0, number:1128.0, number:1129.0, number:1133.0, number:1135.0, number:1136.0, number:1141.0, number:1146.0, number:1152.0, number:1102.0, number:1105.0, number:1117.0, number:1118.0, number:1120.0, number:1121.0, number:1156.0, number:1157.0, number:1158.0, number:1159.0, number:1160.0, number:1161.0, number:1162.0, number:1163.0, number:1164.0, number:1165.0]], 6: [[number:1167.0, number:1168.0, number:1169.0, number:1170.0, number:1171.0, number:1172.0, number:1173.0, number:1174.0, number:1175.0, number:1176.0, number:1106.0, number:1109.0, number:1114.0, number:1115.0, number:1116.0, number:1123.0, number:1125.0, number:1126.0, number:1130.0, number:1131.0, number:1132.0, number:1134.0, number:1137.0, number:1138.0, number:1139.0, number:1140.0, number:1142.0, number:1143.0, number:1144.0, number:1145.0, number:1147.0, number:1148.0, number:1149.0, number:1150.0, number:1151.0, number:1153.0, number:1154.0, number:1155.0]], 7: [[number:1104.0, number:1122.0, number:1671.0, number:1756.0, number:1757.0, number:1644.0, number:1645.0, number:1646.0, number:1647.0, number:1648.0, number:1649.0, number:1650.0, number:1651.0, number:1652.0, number:1653.0, number:1654.0, number:1655.0, number:1766.0, number:1706.0, number:1767.0, number:1768.0, number:1748.0, number:1749.0, number:1750.0, number:1751.0, number:1752.0, number:1753.0, number:1754.0, number:1755.0, number:1769.0]]}, 3: {1: [[number:102.0, number:1316.0, number:1373.0, number:1196.0, number:1166.0, number:1103.0, number:1770.0, number:1664.0, number:1665.0, number:1666.0, number:1667.0, number:1668.0, number:1669.0, number:1672.0, number:1656.0, number:1657.0, number:1658.0, number:1659.0, number:1660.0, number:1661.0, number:1662.0, number:1663.0, number:1673.0, number:1674.0, number:1675.0, number:1676.0, number:1677.0, number:1678.0, number:1679.0, number:1680.0]], 2: [[number:1682.0, number:1683.0, number:1684.0, number:1685.0, number:1686.0, number:1687.0, number:1688.0, number:1689.0, number:1690.0, number:1691.0,

number:1692.0, number:1693.0, number:1694.0, number:1695.0, number:1696.0, number:1697.0, number:1698.0, number:1699.0, number:1700.0, number:1670.0, number:1701.0, number:1702.0, number:1703.0, number:1704.0, number:1705.0, number:321.0, number:14.0, number:15.0, number:16.0, number:17.0, number:323.0]], 3: [[number:406.0, number:19.0, number:20.0, number:263.0, number:264.0, number:265.0, number:324.0, number:266.0, number:267.0, number:268.0, number:269.0, number:270.0, number:271.0, number:325.0, number:272.0, number:273.0, number:12.0, number:322.0, number:275.0, number:276.0, number:277.0, number:326.0, number:278.0, number:409.0, number:279.0, number:280.0, number:281.0, number:282.0, number:283.0, number:327.0, number:284.0, number:285.0, number:286.0, number:287.0, number:288.0, number:13.0, number:328.0, number:290.0, number:291.0]], 4: [[number:293.0, number:294.0, number:295.0, number:29.0, number:8.0, number:297.0, number:298.0, number:299.0, number:300.0, number:301.0, number:330.0, number:302.0, number:303.0, number:304.0, number:305.0, number:331.0, number:32.0, number:333.0, number:334.0, number:225.0, number:226.0, number:227.0, number:228.0, number:229.0, number:230.0, number:335.0, number:231.0, number:232.0, number:233.0, number:234.0, number:235.0, number:236.0, number:336.0, number:238.0, number:239.0, number:240.0, number:241.0, number:242.0, number:337.0]], 5: [[number:244.0, number:245.0, number:246.0, number:247.0, number:248.0, number:38.0, number:249.0, number:250.0, number:251.0, number:252.0, number:253.0, number:254.0, number:39.0, number:255.0, number:40.0, number:423.0, number:341.0, number:42.0, number:343.0, number:344.0, number:345.0, number:346.0, number:347.0, number:348.0, number:349.0, number:350.0, number:351.0, number:352.0, number:353.0, number:354.0, number:355.0, number:358.0, number:414.0, number:415.0, number:416.0, number:374.0, number:377.0, number:380.0, number:386.0]], 6: [[number:392.0, number:396.0, number:398.0, number:399.0, number:402.0, number:405.0, number:256.0, number:257.0, number:258.0, number:259.0, number:260.0, number:261.0, number:262.0, number:274.0, number:289.0, number:306.0, number:307.0, number:308.0, number:309.0, number:310.0, number:311.0, number:312.0, number:313.0, number:314.0, number:315.0, number:10.0, number:316.0, number:317.0, number:318.0, number:319.0, number:320.0, number:21.0, number:9.0, number:22.0, number:23.0, number:24.0, number:25.0, number:26.0, number:27.0]], 7: [[number:329.0, number:30.0, number:31.0, number:332.0, number:33.0, number:11.0, number:34.0, number:35.0, number:36.0, number:37.0, number:338.0, number:339.0, number:340.0, number:41.0, number:342.0, number:43.0, number:44.0, number:45.0, number:296.0, number:46.0, number:47.0, number:48.0, number:49.0, number:356.0, number:357.0, number:359.0, number:360.0, number:361.0, number:362.0, number:363.0, number:364.0, number:365.0, number:366.0, number:367.0, number:368.0, number:369.0, number:370.0, number:371.0, number:372.0]]}, 4: {1: [[number:1259.0, number:1260.0, number:1262.0, number:1264.0, number:1265.0, number:1266.0, number:1273.0, number:1274.0, number:1275.0, number:1279.0, number:1284.0, number:1287.0, number:1288.0, number:1290.0, number:1292.0, number:1296.0, number:1299.0, number:1252.0, number:1253.0, number:1254.0, number:1256.0, number:1257.0, number:1258.0, number:1261.0, number:1263.0, number:1267.0, number:1268.0, number:1269.0, number:1270.0, number:1271.0, number:1272.0, number:1276.0, number:1277.0, number:1278.0, number:1280.0, number:1281.0, number:1282.0, number:1283.0, number:1285.0,

number:1286.0, number:1289.0, number:1291.0, number:1293.0, number:1294.0, number:1295.0, number:1297.0, number:1298.0, number:1300.0, number:1301.0, number:1255.0, number:1227.0, number:1228.0, number:1229.0, number:1230.0, number:1231.0, number:1232.0, number:1233.0, number:1234.0, number:1235.0, number:1236.0, number:1237.0, number:1238.0, number:1239.0, number:1240.0, number:1241.0, number:1242.0, number:1244.0, number:1245.0, number:1246.0, number:1247.0, number:1248.0, number:1250.0, number:1251.0, number:1243.0, number:1249.0, number:1681.0, number:18.0, number:292.0, number:243.0, number:391.0, number:28.0, number:373.0, number:375.0, number:376.0, number:378.0, number:379.0, number:381.0, number:382.0, number:383.0, number:384.0, number:385.0, number:387.0, number:388.0, number:389.0, number:390.0, number:393.0, number:394.0, number:395.0, number:397.0, number:400.0, number:401.0, number:403.0, number:404.0, number:2.0, number:237.0, number:3.0, number:1.0, number:5.0, number:4.0, number:6.0, number:7.0, number:407.0, number:408.0, number:410.0]], 2: [[number:412.0, number:413.0, number:417.0, number:418.0, number:419.0, number:420.0, number:421.0, number:422.0, number:424.0, number:1056.0, number:1057.0, number:1058.0, number:1061.0, number:1063.0, number:1067.0, number:1068.0, number:1069.0, number:1070.0, number:1071.0, number:1072.0, number:1076.0, number:1077.0, number:1078.0, number:1080.0, number:1081.0, number:1082.0, number:1083.0, number:1085.0, number:1086.0, number:1089.0, number:1091.0, number:1093.0, number:1094.0, number:1095.0, number:1097.0, number:1098.0, number:1100.0, number:1101.0, number:1059.0, number:1060.0]], 3: [[number:1064.0, number:1065.0, number:1066.0, number:1073.0, number:1074.0, number:1075.0, number:1079.0, number:1084.0, number:1087.0, number:1088.0, number:1090.0, number:1092.0, number:1096.0, number:1099.0, number:1052.0, number:1053.0, number:1054.0, number:1055.0, number:968.0, number:856.0, number:857.0, number:858.0, number:859.0, number:860.0, number:861.0, number:862.0, number:863.0, number:864.0, number:865.0, number:866.0, number:867.0, number:970.0, number:868.0, number:869.0, number:870.0, number:871.0, number:872.0, number:873.0, number:874.0, number:875.0, number:876.0]], 4: [[number:878.0, number:879.0, number:880.0, number:881.0, number:882.0, number:883.0, number:884.0, number:885.0, number:886.0, number:887.0, number:888.0, number:889.0, number:890.0, number:891.0, number:892.0, number:893.0, number:894.0, number:895.0, number:896.0, number:897.0, number:898.0, number:899.0, number:900.0, number:901.0, number:902.0, number:903.0, number:904.0, number:905.0, number:756.0, number:757.0, number:758.0, number:759.0, number:760.0, number:956.0, number:761.0, number:762.0, number:763.0, number:764.0, number:765.0, number:766.0, number:957.0]], 5: [[number:768.0, number:769.0, number:770.0, number:771.0, number:772.0, number:773.0, number:774.0, number:775.0, number:776.0, number:777.0, number:778.0, number:959.0, number:779.0, number:780.0, number:781.0, number:782.0, number:783.0, number:784.0, number:960.0, number:785.0, number:786.0, number:787.0, number:788.0, number:789.0, number:790.0, number:961.0, number:791.0, number:792.0, number:793.0, number:794.0, number:795.0, number:796.0, number:962.0, number:797.0, number:798.0, number:799.0, number:800.0, number:801.0, number:802.0, number:963.0, number:803.0]], 6: [[number:805.0, number:964.0, number:966.0, number:969.0, number:971.0, number:972.0, number:973.0, number:975.0, number:976.0, number:978.0,

number:979.0, number:981.0, number:982.0, number:983.0, number:984.0, number:985.0, number:987.0, number:988.0, number:989.0, number:990.0, number:993.0, number:994.0, number:995.0, number:997.0, number:1000.0, number:1001.0, number:1003.0, number:1004.0, number:906.0, number:907.0, number:908.0, number:909.0, number:910.0, number:911.0, number:912.0, number:913.0, number:914.0, number:915.0, number:916.0, number:917.0, number:918.0]], 7: [[number:920.0, number:921.0, number:922.0, number:923.0, number:924.0, number:925.0, number:926.0, number:927.0, number:928.0, number:929.0, number:930.0, number:931.0, number:932.0, number:933.0, number:934.0, number:935.0, number:936.0, number:937.0, number:938.0, number:939.0, number:940.0, number:941.0, number:942.0, number:943.0, number:944.0, number:945.0, number:946.0, number:947.0, number:948.0, number:949.0, number:950.0, number:951.0, number:952.0, number:953.0, number:954.0, number:955.0, number:754.0, number:958.0, number:974.0, number:977.0, number:980.0]], 8: [[number:1506.0, number:1508.0, number:1509.0, number:1510.0, number:1511.0, number:1512.0, number:1514.0, number:1515.0, number:1516.0, number:1517.0, number:1518.0, number:1520.0, number:1521.0, number:1522.0, number:1523.0, number:1524.0, number:1526.0, number:1527.0, number:1528.0, number:1529.0, number:1530.0, number:1532.0, number:1533.0, number:1534.0, number:1535.0, number:1536.0, number:1538.0, number:1539.0, number:1540.0, number:1541.0, number:1542.0, number:1544.0, number:1545.0, number:1546.0, number:1547.0, number:1548.0, number:1550.0, number:1551.0, number:1507.0, number:1513.0, number:1519.0, number:1525.0, number:1477.0]]}, 5: {1: [[number:411.0, number:1062.0, number:877.0, number:767.0, number:804.0, number:919.0, number:986.0, number:991.0, number:992.0, number:996.0, number:998.0, number:999.0, number:1002.0, number:1005.0, number:752.0, number:753.0, number:755.0, number:806.0, number:807.0, number:808.0, number:809.0, number:810.0, number:811.0, number:812.0, number:813.0, number:814.0, number:815.0, number:816.0, number:817.0, number:818.0, number:819.0, number:820.0, number:821.0, number:822.0, number:823.0, number:824.0, number:825.0, number:826.0, number:827.0, number:828.0, number:829.0]], 2: [[number:831.0, number:832.0, number:833.0, number:834.0, number:835.0, number:836.0, number:837.0, number:838.0, number:839.0, number:840.0, number:841.0, number:842.0, number:843.0, number:844.0, number:845.0, number:846.0, number:847.0, number:848.0, number:849.0, number:850.0, number:851.0, number:852.0, number:853.0, number:854.0, number:855.0, number:1006.0, number:1007.0, number:1008.0, number:1009.0, number:1010.0, number:1011.0, number:1012.0, number:1013.0, number:1014.0, number:1015.0, number:1016.0, number:1017.0, number:1018.0, number:1019.0, number:1020.0, number:965.0]], 3: [[number:1022.0, number:1023.0, number:1024.0, number:1025.0, number:1026.0, number:1027.0, number:1028.0, number:1029.0, number:1030.0, number:1031.0, number:1032.0, number:1033.0, number:1034.0, number:1035.0, number:1036.0, number:1037.0, number:1038.0, number:1039.0, number:1040.0, number:1041.0, number:1042.0, number:1043.0, number:1044.0, number:1045.0, number:1046.0, number:1047.0, number:1048.0, number:1049.0, number:1050.0, number:1051.0, number:967.0, number:1478.0, number:1479.0, number:1480.0, number:1481.0, number:1482.0, number:1483.0, number:1484.0, number:1485.0, number:1486.0, number:1487.0, number:1488.0]], 4: [[number:1490.0, number:1491.0, number:1492.0, number:1493.0,

number:1494.0, number:1495.0, number:1496.0, number:1497.0, number:1498.0, number:1499.0, number:1500.0, number:1501.0, number:1502.0, number:1503.0, number:1504.0, number:1505.0, number:1531.0, number:1537.0, number:1543.0, number:1549.0, number:1392.0, number:1415.0, number:1393.0, number:1395.0, number:1396.0, number:1397.0, number:1416.0, number:1399.0, number:1401.0, number:1417.0, number:1418.0, number:1419.0, number:1420.0, number:1421.0, number:1422.0, number:740.0, number:741.0, number:743.0, number:746.0, number:1717.0, number:1718.0]], 5: [[number:1720.0, number:1721.0, number:1722.0, number:1723.0, number:1724.0, number:1725.0, number:1726.0, number:1727.0, number:1728.0, number:1729.0, number:1730.0, number:1731.0, number:1732.0, number:1734.0, number:1735.0, number:1736.0, number:1737.0, number:1738.0, number:1739.0, number:1740.0, number:1741.0, number:1742.0, number:1743.0, number:1744.0, number:1745.0, number:1746.0, number:1747.0, number:1437.0, number:1406.0, number:1407.0, number:742.0, number:1408.0, number:1409.0, number:744.0]], 6: [[number:745.0, number:1411.0, number:1412.0, number:1413.0, number:1423.0, number:1424.0, number:1425.0, number:1426.0, number:1427.0, number:1428.0, number:1429.0, number:1430.0, number:1431.0, number:1432.0, number:1433.0, number:1434.0, number:1377.0, number:1378.0, number:1379.0, number:1380.0, number:1435.0, number:1381.0, number:1382.0, number:1383.0, number:1384.0, number:1385.0, number:1386.0, number:1436.0, number:1387.0, number:1388.0, number:1389.0, number:1390.0, number:1391.0, number:1394.0, number:1398.0, number:1438.0, number:1400.0, number:1402.0, number:1403.0, number:1404.0, number:1439.0, number:1405.0, number:1440.0, number:1441.0]], 7: [[number:1443.0, number:1444.0, number:1445.0, number:1446.0, number:1447.0, number:1414.0, number:1448.0, number:1449.0, number:1450.0, number:1733.0, number:1451.0, number:1452.0, number:1453.0, number:1454.0, number:1455.0, number:1456.0, number:1457.0, number:1458.0, number:1459.0, number:1460.0, number:1461.0, number:1462.0, number:1463.0, number:1464.0, number:1465.0, number:1466.0, number:1467.0, number:1468.0, number:1469.0, number:1470.0, number:1471.0, number:1472.0, number:1473.0, number:1474.0, number:1475.0, number:1476.0, number:1627.0, number:1628.0, number:1629.0, number:1630.0, number:1632.0]], 8: [[number:1634.0, number:1635.0, number:1637.0, number:1638.0, number:1639.0, number:1640.0, number:1642.0, number:1643.0, number:1631.0, number:1636.0, number:1641.0, number:1561.0, number:1556.0, number:1567.0, number:1562.0, number:1578.0, number:1563.0, number:1564.0, number:1565.0, number:1557.0, number:1606.0, number:1607.0, number:1608.0, number:1568.0, number:1609.0, number:1610.0, number:1611.0, number:1613.0, number:1569.0, number:1570.0, number:1623.0, number:1571.0, number:1624.0, number:1625.0, number:1626.0, number:1572.0, number:1591.0, number:1573.0, number:1602.0, number:1574.0, number:1575.0, number:1576.0]]}, 6: {1: [[number:830.0, number:1021.0, number:1489.0, number:1719.0, number:1410.0, number:1442.0, number:1633.0, number:1577.0, number:1552.0, number:1553.0, number:1554.0, number:1612.0, number:1580.0, number:1581.0, number:1582.0, number:1583.0, number:1584.0, number:1615.0, number:1585.0, number:1616.0, number:1586.0, number:1617.0, number:1587.0, number:1588.0, number:1590.0, number:1619.0, number:1614.0,

number:1589.0, number:1620.0, number:1621.0, number:1622.0, number:1592.0, number:1593.0, number:1594.0, number:1595.0, number:1596.0, number:1603.0, number:1579.0, number:1597.0, number:1598.0, number:1599.0, number:1600.0, number:1601.0, number:1604.0, number:1605.0, number:1555.0, number:1618.0]], 2: [[number:1558.0, number:1559.0, number:1560.0, number:1206.0, number:1207.0, number:1208.0, number:1209.0, number:1210.0, number:1211.0, number:1212.0, number:1213.0, number:1214.0, number:1215.0, number:1216.0, number:1217.0, number:1218.0, number:1219.0, number:1220.0, number:1221.0, number:1222.0, number:1223.0, number:1224.0, number:1226.0, number:1202.0, number:1203.0, number:1204.0, number:1205.0, number:1225.0, number:714.0, number:656.0, number:658.0, number:660.0, number:661.0, number:662.0, number:663.0, number:664.0, number:665.0, number:666.0, number:667.0, number:668.0, number:670.0, number:671.0, number:672.0, number:673.0, number:674.0, number:675.0, number:676.0, number:677.0, number:678.0, number:679.0, number:680.0, number:681.0, number:682.0, number:683.0, number:684.0]], 3: [[number:686.0, number:688.0, number:689.0, number:690.0, number:691.0, number:692.0, number:693.0, number:694.0, number:695.0, number:696.0, number:697.0, number:698.0, number:700.0, number:701.0, number:702.0, number:703.0, number:718.0, number:704.0, number:719.0, number:657.0, number:659.0, number:669.0, number:706.0, number:707.0, number:709.0, number:712.0, number:715.0, number:716.0, number:720.0, number:721.0, number:723.0, number:724.0, number:687.0, number:708.0, number:710.0, number:711.0, number:713.0, number:699.0, number:717.0, number:705.0, number:722.0, number:650.0, number:651.0, number:652.0, number:653.0, number:654.0, number:655.0, number:450.0, number:451.0, number:453.0, number:454.0, number:455.0, number:506.0, number:507.0, number:508.0, number:509.0, number:510.0]], 4: [[number:512.0, number:513.0, number:514.0, number:515.0, number:516.0, number:517.0, number:518.0, number:519.0, number:520.0, number:521.0, number:522.0, number:523.0, number:524.0, number:525.0, number:526.0, number:527.0, number:528.0, number:529.0, number:530.0, number:531.0, number:532.0, number:533.0, number:534.0, number:535.0, number:536.0, number:537.0, number:538.0, number:539.0, number:540.0, number:541.0, number:542.0, number:543.0, number:544.0, number:545.0, number:546.0, number:547.0, number:548.0, number:549.0, number:550.0, number:551.0, number:552.0, number:553.0, number:554.0, number:555.0, number:456.0, number:457.0, number:458.0, number:459.0, number:460.0, number:461.0, number:462.0, number:463.0, number:464.0, number:465.0, number:466.0, number:467.0, number:468.0, number:469.0, number:470.0, number:471.0]], 5: [[number:473.0, number:474.0, number:475.0, number:476.0, number:477.0, number:478.0, number:479.0, number:480.0, number:481.0, number:482.0, number:483.0, number:484.0, number:485.0, number:486.0, number:487.0, number:488.0, number:489.0, number:490.0, number:491.0, number:492.0, number:493.0, number:494.0, number:495.0, number:496.0, number:497.0, number:498.0, number:499.0, number:500.0, number:501.0, number:502.0, number:503.0, number:504.0, number:505.0, number:606.0, number:607.0, number:608.0, number:609.0, number:610.0, number:611.0, number:612.0, number:613.0, number:614.0, number:615.0, number:616.0, number:617.0, number:618.0, number:619.0, number:620.0, number:621.0, number:622.0, number:623.0, number:624.0, number:625.0, number:626.0, number:627.0, number:628.0, number:629.0, number:630.0, number:631.0, number:632.0]], 6: [[number:634.0, number:635.0, number:636.0,

number:637.0, number:638.0, number:639.0, number:640.0, number:641.0, number:642.0, number:643.0, number:644.0, number:645.0, number:646.0, number:647.0, number:648.0, number:649.0, number:556.0, number:557.0, number:558.0, number:559.0, number:560.0, number:561.0, number:562.0, number:563.0, number:564.0, number:565.0, number:566.0, number:567.0, number:568.0, number:569.0, number:570.0, number:571.0, number:572.0, number:573.0, number:574.0, number:575.0, number:576.0, number:577.0, number:578.0, number:579.0, number:580.0, number:581.0, number:582.0, number:583.0, number:584.0, number:585.0, number:586.0, number:587.0, number:588.0, number:589.0, number:590.0, number:591.0, number:592.0, number:593.0, number:594.0, number:595.0, number:596.0, number:597.0, number:598.0, number:599.0]]}, 7: {1: [[number:601.0, number:602.0, number:603.0, number:604.0, number:605.0, number:452.0, number:1566.0, number:685.0, number:511.0, number:472.0, number:633.0, number:600.0]]}}

## 2D CAMFA SONUÇLARI

| no | FILTRE NO | En | boy | yükseklik | Alan | Okek | freq | x_toplam | raf_no | y_paket_top | kutu | y_cum |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 609 | AF4532 | 210 | 210 | 460 | 44100 | 818 | 114 | 23940 | 2 | 420 | 2 | 420 |
| 1550 | RA6219/RA6220 | 240 | 240 | 325 | 57600 | 627 | 75 | 18000 | 4 | 480 | 2 | 900 |
| 1475 | RA6113A | 275 | 275 | 515 | 75625 | 477 | 50 | 13750 | 6 | 550 | 2 | 1450 |
| 448 | AF4412 | 200 | 200 | 460 | 40000 | 902 | 161 | 32200 | 8 | 600 | 2 | 2050 |
| 1100 | RA6022 | 310 | 310 | 455 | 96100 | 376 | 75 | 23250 | 10 | 620 | 2 | 2670 |
| 1375 | RA6112 | 270 | 270 | 540 | 72900 | 495 | 100 | 27000 | 3 | 810 | 3 | 3480 |
| 1642 | SF8161A | 407 | 309 | 371 | 125763 | 287 | 63 | 25641 | 6 | 927 | 3 | 4407 |
| 1300 | RA6100/RA6101 | 325 | 325 | 400 | 105625 | 342 | 75 | 24375 | 9 | 975 | 3 | 5382 |
| 48 | AF4303A | 335 | 335 | 265 | 112225 | 322 | 175 | 58625 | 5 | 1675 | 4 | 7057 |
| 223 | AF4340 | 305 | 305 | 395 | 93025 | 388 | 200 | 61000 | 11 | 1830 | 4 | 8887 |
| 750 | RA6003/RA6004 | 290 | 290 | 350 | 84100 | 429 | 300 | 87000 | 8 | 2320 | 5 | 11207 |

| NO | FILTRE NO | En | boy | yükseklik | Alan | Okek | freq | x_toplam | raf_no | y_top | kutu | cum | x_raf |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1215 | RA6034/RA6050 | 220 | 220 | 410 | 48400 | 746 | 1 | 220 | 1215 | 1 | 220 | 1 | 220 | 1 |
| 1209 | RA6034/RA6044 | 220 | 220 | 410 | 48400 | 746 | 1 | 220 | 1209 | 1 | 220 | 1 | 440 | 1 |
| 1210 | RA6034/RA6045 | 220 | 220 | 410 | 48400 | 746 | 1 | 220 | 1210 | 1 | 220 | 1 | 660 | 1 |
| 1201 | RA6034/RA6036 | 220 | 220 | 410 | 48400 | 746 | 1 | 220 | 1201 | 1 | 220 | 1 | 880 | 1 |
| 1202 | RA6034/RA6037 | 220 | 220 | 410 | 48400 | 746 | 1 | 220 | 1202 | 1 | 220 | 1 | 1100 | 1 |
| 1203 | RA6034/RA6038 | 220 | 220 | 410 | 48400 | 746 | 1 | 220 | 1203 | 1 | 220 | 1 | 1320 | 1 |
| 1204 | RA6034/RA6039 | 220 | 220 | 410 | 48400 | 746 | 1 | 220 | 1204 | 1 | 220 | 1 | 1540 | 1 |
| 1205 | RA6034/RA6040 | 220 | 220 | 410 | 48400 | 746 | 1 | 220 | 1205 | 1 | 220 | 1 | 1760 | 1 |
| 1206 | RA6034/RA6041 | 220 | 220 | 410 | 48400 | 746 | 1 | 220 | 1206 | 1 | 220 | 1 | 1980 | 1 |
| 1207 | RA6034/RA6042 | 220 | 220 | 410 | 48400 | 746 | 1 | 220 | 1207 | 1 | 220 | 1 | 2200 | 1 |
| 1223 | RA6034/RA6058 | 220 | 220 | 410 | 48400 | 746 | 1 | 220 | 1223 | 1 | 220 | 1 | 2420 | 1 |
| 1211 | RA6034/RA6046 | 220 | 220 | 410 | 48400 | 746 | 1 | 220 | 1211 | 1 | 220 | 1 | 2640 | 1 |
| 1221 | RA6034/RA6056 | 220 | 220 | 410 | 48400 | 746 | 1 | 220 | 1221 | 1 | 220 | 1 | 2860 | 1 |
| 1220 | RA6034/RA6055 | 220 | 220 | 410 | 48400 | 746 | 1 | 220 | 1220 | 1 | 220 | 1 | 3080 | 1 |
| 1219 | RA6034/RA6054 | 220 | 220 | 410 | 48400 | 746 | 1 | 220 | 1219 | 1 | 220 | 1 | 3300 | 1 |
| 1218 | RA6034/RA6053 | 220 | 220 | 410 | 48400 | 746 | 1 | 220 | 1218 | 1 | 220 | 1 | 3520 | 1 |
| 1217 | RA6034/RA6052 | 220 | 220 | 410 | 48400 | 746 | 1 | 220 | 1217 | 1 | 220 | 1 | 3740 | 1 |
| 1216 | RA6034/RA6051 | 220 | 220 | 410 | 48400 | 746 | 1 | 220 | 1216 | 1 | 220 | 1 | 3960 | 1 |
| 1208 | RA6034/RA6043 | 220 | 220 | 410 | 48400 | 746 | 1 | 220 | 1208 | 1 | 220 | 1 | 4180 | 1 |
| 1214 | RA6034/RA6049 | 220 | 220 | 410 | 48400 | 746 | 1 | 220 | 1214 | 1 | 220 | 1 | 4400 | 1 |
| 1213 | RA6034/RA6048 | 220 | 220 | 410 | 48400 | 746 | 1 | 220 | 1213 | 1 | 220 | 1 | 4620 | 1 |
| 1212 | RA6034/RA6047 | 220 | 220 | 410 | 48400 | 746 | 1 | 220 | 1212 | 1 | 220 | 1 | 4840 | 1 |
| 1222 | RA6034/RA6057 | 220 | 220 | 410 | 48400 | 746 | 1 | 220 | 1222 | 1 | 220 | 1 | 5060 | 1 |

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1069 | RA6005/RA6025 | 295 | 295 | 345 | 87025 | 415 | 1 | 295 | 1069 | 1 | 295 | 1 | 5355 | 1 |
| 1070 | RA6005/RA6026 | 295 | 295 | 345 | 87025 | 415 | 1 | 295 | 1070 | 1 | 295 | 1 | 5650 | 1 |
| 1071 | RA6005/RA6027 | 295 | 295 | 345 | 87025 | 415 | 1 | 295 | 1071 | 1 | 295 | 1 | 5945 | 1 |
| 1072 | RA6005/RA6028 | 295 | 295 | 345 | 87025 | 415 | 1 | 295 | 1072 | 1 | 295 | 1 | 6240 | 1 |
| 1077 | RA6005/RA6033 | 295 | 295 | 345 | 87025 | 415 | 1 | 295 | 1077 | 1 | 295 | 1 | 6535 | 1 |
| 1074 | RA6005/RA6030 | 295 | 295 | 345 | 87025 | 415 | 1 | 295 | 1074 | 1 | 295 | 1 | 6830 | 1 |
| 1075 | RA6005/RA6031 | 295 | 295 | 345 | 87025 | 415 | 1 | 295 | 1075 | 1 | 295 | 1 | 7125 | 1 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 1055 | RA6005/RA6011 | 295 | 295 | 345 | 87025 | 415 | 1 | 295 | 1055 | 1 | 295 | 1 | 19515 | 2 |
| 1062 | RA6005/RA6018 | 295 | 295 | 345 | 87025 | 415 | 1 | 295 | 1062 | 1 | 295 | 1 | 19810 | 2 |
| 738 | FF3308 | 355 | 270 | 465 | 95850 | 377 | 1 | 355 | 738 | 1 | 270 | 1 | 20165 | 2 |
| 739 | FF3309 | 355 | 270 | 465 | 95850 | 377 | 1 | 355 | 739 | 1 | 270 | 1 | 20520 | 2 |
| 741 | FF3311 | 355 | 270 | 465 | 95850 | 377 | 1 | 355 | 741 | 1 | 270 | 1 | 20875 | 2 |
| 742 | FF3312 | 355 | 270 | 465 | 95850 | 377 | 1 | 355 | 742 | 1 | 270 | 1 | 21230 | 2 |
| 740 | FF3310 | 355 | 270 | 465 | 95850 | 377 | 1 | 355 | 740 | 1 | 270 | 1 | 21585 | 2 |
| 1200 | RA6034/RA6035 | 220 | 220 | 410 | 48400 | 746 | 2 | 440 | 1200 | 1 | 220 | 1 | 22025 | 2 |
| 743 | EFF1054 | 355 | 270 | 475 | 95850 | 377 | 2 | 710 | 743 | 1 | 270 | 1 | 22735 | 2 |
| 745 | EFF1091MB | 407 | 309 | 435 | 125763 | 287 | 2 | 814 | 745 | 1 | 309 | 1 | 23549 | 2 |
| 1773 | EFF1048T | 515 | 390 | 271 | 200850 | 180 | 3 | 1545 | 1773 | 1 | 390 | 1 | 25094 | 3 |
| 747 | LF5174H | 570 | 570 | 255 | 324900 | 111 | 3 | 1710 | 747 | 1 | 570 | 1 | 26804 | 3 |
| 1768 | SW8264 | 407 | 309 | 315 | 125763 | 287 | 5 | 2035 | 1768 | 1 | 309 | 1 | 28839 | 3 |
| 1715 | SF8266 | 440 | 270 | 315 | 118800 | 304 | 5 | 2200 | 1715 | 1 | 270 | 1 | 31039 | 3 |
| 1741 | SF8502 | 455 | 345 | 355 | 156975 | 230 | 5 | 2275 | 1741 | 1 | 345 | 1 | 33314 | 3 |
| 733 | EF1090 | 455 | 345 | 295 | 156975 | 230 | 5 | 2275 | 733 | 1 | 345 | 1 | 35589 | 3 |
| 728 | EF1018 | 515 | 390 | 330 | 200850 | 180 | 5 | 2575 | 728 | 1 | 390 | 1 | 38164 | 4 |
| 723 | EF1015 | 567 | 429 | 173 | 243243 | 149 | 5 | 2835 | 723 | 1 | 429 | 1 | 40999 | 4 |
| 1225 | RA6045 | 115 | 115 | 280 | 13225 | 2730 | 25 | 2875 | 1225 | 1 | 115 | 1 | 43874 | 4 |
| 1746 | SF8535 | 407 | 309 | 293 | 125763 | 287 | 10 | 4070 | 1746 | 1 | 309 | 1 | 47944 | 4 |
| 1705 | SF8256 | 427 | 324 | 258 | 138348 | 261 | 10 | 4270 | 1705 | 1 | 324 | 1 | 52214 | 5 |
| 30 | AD2006 | 305 | 305 | 365 | 93025 | 388 | 18 | 5490 | 30 | 1 | 305 | 1 | 57704 | 5 |
| 1756 | SF8738 | 500 | 500 | 150 | 250000 | 144 | 12 | 6000 | 1756 | 1 | 500 | 1 | 63704 | 6 |
| 1525 | RA6113C | 275 | 275 | 515 | 75625 | 477 | 25 | 6875 | 1525 | 1 | 275 | 1 | 70579 | 6 |
| 1625 | SF8104C | 415 | 255 | 265 | 105825 | 341 | 17 | 7055 | 1625 | 1 | 255 | 1 | 77634 | 6 |
| 1720 | SF8429 | 355 | 270 | 399 | 95850 | 377 | 21 | 7455 | 1720 | 1 | 270 | 1 | 85089 | 6 |
| 1175 | RA6031/RA6152 | 320 | 320 | 475 | 102400 | 353 | 25 | 8000 | 1175 | 1 | 320 | 1 | 93089 | 6 |
| 1250 | RA6064 | 175 | 175 | 475 | 30625 | 1179 | 50 | 8750 | 1250 | 1 | 175 | 1 | 101839 | 7 |
| 423 | AF4347/AF4348 | 360 | 360 | 380 | 129600 | 279 | 25 | 9000 | 423 | 1 | 360 | 1 | 110839 | 8 |
| 0 | AD2000 | 305 | 305 | 365 | 93025 | 388 | 30 | 9150 | 0 | 1 | 305 | 1 | 119989 | 8 |

[101 rows x 13 columns]

| x_raf | y_max |
|---|---|
| 1 | 295 |
| 2 | 309 |
| 3 | 570 |
| 4 | 429 |
| 5 | 324 |
| 6 | 500 |
| 7 | 175 |
| 8 | 360 |

y_max_top =
Name: y_paket_top, dtype: int64

Algoritmanın sonucu=5 konteyner sayisi

# ÖZGEÇMİŞ

**KİŞİSEL BİLGİLER**

**Adı Soyadı** :Sezgi TEKIL

**Doğum Tarihi ve Yeri** :09.04.1988

**Yabancı Dili** : İngilizce

**E-posta** :tekilsezgi@gmail.com

**ÖĞRENİM DURUMU**

| **Derece** | **Alan** | **Okul/Üniversite** | **Mezuniyet Yılı** |
|---|---|---|---|
| Lisans | Endüstri Mühendisliği(Ingilizce)(Burslu) | Doğuş Üniversitesi | 2012 |
| Lise | Fen | İskenderun Demirçelik Anadolu Lisesi | 2006 |

**İŞ DURUMU**

| Yıl | Firma/Kurum | Görevi |
|---|---|---|
| 2016 | İstanbul Gedik Üniversitesi | Endüstri Mühendisliği Araştırma Görevlisi |

**YAYINLARI**

**Makale**

1. S.TEKİL, V.ÖZKIR, KONTEYNER YÜKLEME PROBLEMLERİNİN İNCELENMESİ VE LOJİSTİK SEKTÖRÜNDE BİR UYGULAMA, Toros Üniversitesi İİSBF Sosyal Bilimler Dergisi, Cilt 3, Sayı 5 (2016)

**Bildiri**

1. 2016- TOROS ÜNİVERSİTESİ 5. ULUSAL LOJİSTİK VE TEDARİK ZİNCİRİ KONGRESİ(LODER),MERSİN Sezgi TEKİL, Vildan Ç. ÖZKIR,Büyük Boyutlu Kutulama Problemlerinin İncelenmesi Ve Çözümü İçin Sezgisel Algoritmaların Uygulanması 2016-

2. YÖNEYLEM ARAŞTIRMASI VE ENDÜSTRİ MÜHENDİSLİĞİ 36. ULUSAL KONGRESİNDE,(YAEM), İZMİR Vildan Özkır,Seda Erbayrak , Sezgi Tekil Tek Boyutlu Kutulama Problemlerinin Klasik Kutulama Algoritmaları ve Genetik Algoritma ile Çözümü

**Proje**

1. 2011-2012 THY Teknik A.Ş. Üretim Planlama Birimi Lisans Bitirme Projesi Uçak bakım kartlarının, ARENA programını kullanarak simülasyon modelinin kurulması ve sorunlu olan bakım hatlarının optimizasyonu

**ÖDÜLLERİ**

2012 -Doğuş Üniversitesi Endüstri Mühendisliği Bölüm Üçüncülüğü