

T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ÇOK AMAÇLI 3 BOYUTLU KUTULAMA PROBLEMLERİNİN
OPTİMİZASYONU İÇİN MATEMATİKSEL MODEL VE GENETİK
ALGORİTMA ÖNERİSİ

Seda ERBAYRAK

DOKTORA TEZİ

Endüstri Mühendisliği Anabilim Dalı

Endüstri Mühendisliği Programı

Danışman

Doç. Dr. Vildan ÇETİNSAYA ÖZKIR

Eş Danışman

Dr. Öğr. Üyesi Umman Mahir YILDIRIM

Ocak, 2022

T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**ÇOK AMAÇLI 3 BOYUTLU KUTULAMA PROBLEMLERİNİN
OPTİMİZASYONU İÇİN MATEMATİKSEL MODEL VE GENETİK
ALGORİTMA ÖNERİSİ**

Seda ERBAYRAK tarafından hazırlanan tez 27.01.2022 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalı, Endüstri Mühendisliği Programı **DOKTORA TEZİ** olarak kabul edilmiştir.

Doç. Dr. Vildan ÇETİNSAYA ÖZKİR
Yıldız Teknik Üniversitesi
Danışman

Dr. Öğr. Üyesi Umman Mahir
YILDIRIM
İstanbul Bilgi Üniversitesi
Eş-Danışman

Jüri Üyeleri

Doç. Dr. Vildan ÇETİNSAYA ÖZKİR, Danışman
Yıldız Teknik Üniversitesi

Prof. Dr. Prof. Dr. Umut Rıfat TUZKAYA, Üye
Yıldız Teknik Üniversitesi

Doç. Dr. Umut ASAN, Üye
İstanbul Teknik Üniversitesi

Doç. Dr. Nezir AYDIN, Üye
Yıldız Teknik Üniversitesi

Dr. Öğr. Üyesi Semih YALÇINDAĞ, Üye
Yeditepe Üniversitesi

Danışmanım Doç. Dr. Vildan ÇETİNSAYA ÖZKIR ve Eş danışmanım Dr. Öğr. Üyesi Umman Mahir YILDIRIM sorumluluğunda tarafımca hazırlanan Çok Amaçlı 3 Boyutlu Kutulama Problemlerinin Optimizasyonu İçin Matematiksel Model Ve Genetik Algoritma Önerisi başlıklı çalışmada çeşitli kaynaklardan aldığım bilgileri ana metin ve referanslarda eksiksiz gösterdiğimi, araştırma verilerine ve sonuçlarına ilişkin çarpıtma ve/veya sahtecilik yapmadığımı, çalışmam süresince bilimsel araştırma ve etik ilkelerine uygun davrandığımı beyan ederim. Beyanımın aksinin ispatı halinde her türlü yasal sonucu kabul ederim.

Seda ERBAYRAK

İmza

Aileme

ve

biricik kızıma

TEŞEKKÜR

Doktora tezi süresince desteklerini ve yardımlarını esirgemeyen, özveri ile bu süreçte benimle ilgilenen ve her zaman yanımda olan değerli danışman hocam Doç. Dr. Vildan ÇETİNSAYA ÖZKIR ve eş danışmanım Dr. Öğr. Üyesi Umman Mahir YILDIRIM'a teşekkür ederim. Tez izleme komitesinde bulunarak tezimin ilerlemesi ve yönlendirilmesinde büyük katkıda bulunan değerli hocalarım Prof. Dr. Umut Rıfat TUZKAYA ve Doç. Dr. Umut ASAN'a teşekkürlerimi sunarım. Özellikle kodlama çalışmalarımda yardımlarını esirgemeyen değerli hocam Dr. Ozan ÖZİŞİK'a çok teşekkür ederim. Sadece tezimde değil, hayatın her döneminde yanımda olan sevgili eşim Engin ERBAYRAK'a ve canım kızım Defne ERBAYRAK'a canı gönülden teşekkürlerimi sunarım.

Seda ERBAYRAK

İÇİNDEKİLER

SİMGE LİSTESİ	vii
KISALTMA LİSTESİ	viii
ŞEKİL LİSTESİ	ix
TABLO LİSTESİ	xi
ÖZET	xii
ABSTRACT	xiv
1 GİRİŞ	1
1.1 Literatür Özeti.....	3
1.1.1 Matematiksel Model Kullanılan Çalışmalar.....	4
1.1.2 Sezgisel Tabanlı Algoritmaların Kullanıldığı Çalışmalar.....	7
1.2 Tezin Amacı	11
1.3 Hipotez	12
1.4 Tezin İçeriği	12
2 KUTULAMA PROBLEMLERİ	14
2.1 Kutulama Problemlerinin Sınıflandırılması.....	14
2.1.1 Paketlenecek Nesne Boyutuna Göre Sınıflandırma	14
2.1.2 Kullanılan Amaç Türüne Göre Sınıflandırma	26
2.1.3 Nesne Çeşitliliğine Göre Sınıflandırma	28
2.1.4 Nesne Şekillerine Göre Sınıflandırma	28
2.1.5 Kutu Çeşitliliğine Göre Sınıflandırma	29
2.2 Kısıtlar	29
3 ÖNERİLEN YÖNTEM	33
3.1 Problem Tanımı	33
3.2 Önerilen Matematiksel Model	34
3.2.1 Karar Değişkenleri	36
3.2.2 Amaç Fonksiyonu	37
3.2.3 Kısıtlar	38
3.2.4 Doğrusal Olmayan Denklemlerin Doğrusal Hale Getirilmesi.....	41
3.3 Önerilen Genetik Algoritma Tabanlı Sezgisel Algoritma.....	45
3.3.1 Birinci Aşama: Genetik Algoritmaya Dayalı Sezgisel Algoritma ...	45
3.3.2 İkinci Aşama: Duvar Örne Algoritması.....	56

3.3.3 Üçüncü Aşama: Dengeli Yerleşim Planlarının Oluşturulması	60
4 BİLGİ VE SAYISAL ANALİZLER	63
4.1 Veri Seti	63
4.2 Önerilen Matematiksel Model Analizi	64
4.2.1 Matematiksel Modelin Doğrulanması	64
4.2.2 Uygulama	70
4.3 Önerilen Sezgisel Algoritmanın Analizi	72
5 SONUÇ VE ÖNERİLER	83
KAYNAKÇA	87
TEZDEN ÜRETİLMİŞ YAYINLAR	93

SİMGE LİSTESİ

f	Aile
j	Kutu
i,k	Nesne

KISALTMA LİSTESİ

3BKP	3 Boyutlu Kutulama Problemi
GAMS	General Algebraic Modeling System
FR	Doluluk Oranı
FUR	Aile Birliği Oranı

ŞEKİL LİSTESİ

Şekil 1.1 Yükleme planı	2
Şekil 2.1 Bir boyutlu kutulama problemi gösterimi.....	15
Şekil 2.2 En yakın yer algoritması.....	16
Şekil 2.3 İlk uygun yer algoritması	17
Şekil 2.4 Azalan ilk uygun yer algoritması.....	17
Şekil 2.5 En uygun yer algoritması	17
Şekil 2.6 Azalan en uygun yer algoritması	18
Şekil 2.7 En kötü yer algoritması	18
Şekil 2.8 Katman örneği	22
Şekil 2.9 Örnek yükleme adımları.....	22
Şekil 2.10 Katman örneği	23
Şekil 2.11 Maksimum boş alanlar	23
Şekil 2.12 Yatay ve dikey ekseninde oluşturulmuş blok örnekleri	24
Şekil 2.13 Ön üst sağ köşe algoritması (Gonçalves ve Resende, 2013)	24
Şekil 2.14 Arka sol alt algoritması	25
Şekil 2.15 Köşe noktaların gösterimi.....	25
Şekil 2.16 Uç noktaların gösterimi	26
Şekil 2.17 Kutulama problemlerinin amaç türüne göre sınıflandırılması.....	26
Şekil 2.18 Döndürme şekilleri.....	30
Şekil 3.1 Kutu koordinatları ve nesne yönelimi.....	34
Şekil 3.2 Kromozom yapısının gösterimi (Fan vd., 2020)	49
Şekil 3.3 Kromozom içeriği.....	49
Şekil 3.4 Ebeveyn 1 ve Ebeveyn 2	54
Şekil 3.5 Kromozomlar arası gen transferi	54
Şekil 3.6 Çaprazlama	55
Şekil 3.7 Oluşturulan yeni nesil	55
Şekil 3.8 Örnek başlangıç çözüm	58
Şekil 3.9 Yerleşim planı (Ürün ailesi dikkate alınarak)	59
Şekil 3.10 Yerleşim planı (Ürün Ailesi dikkate alınmadan)	59
Şekil 3.11 Aynalanmış katman.....	61
Şekil 3.12 Birbirleri arasında yer değiştirmiş katmanlar	62
Şekil 3.13 Kaydırılmış katman örneği	62

Şekil 4.1 Optimum yükleme planları	68
Şekil 4.2 Kutu kullanımının amaçlara göre değişimi	82

TABLO LİSTESİ

Tablo 1.1	Matematiksel modellerindeki kısıtlamalar.....	6
Tablo 1.2	Literatürde 3BKP çözümü için kullanılan sezgisel yaklaşımlar	10
Tablo 2.1	Kullanılan karar değişkenler	15
Tablo 2.2	Kullanılan karar değişkenler (2 Boyutlu)	18
Tablo 2.3	Kutulama problemlerinin birbirinden farklılıkları	28
Tablo 3.1	Matematiksel modeller arasındaki benzerlikler ve farklılıklar	35
Tablo 3.2	Kullanılan karar değişkenler	36
Tablo 3.3	Genetik algoritma temel aşamaları	47
Tablo 3.4	Kromozom yapıları	48
Tablo 3.5	Seçim oranı hesabı senaryoları.....	53
Tablo 3.6	Duvar örme algoritmasının temel aşamaları	57
Tablo 3.7	Nesne boyutları.....	57
Tablo 3.8	3 Boyutlu yerleşim planı (ürün ailesi dikkate alınarak)	59
Tablo 3.9	3 Boyutlu yerleşim planı (ürün ailesi dikkate alınmadan)	59
Tablo 3.10	Aynalanmış katman örneği	61
Tablo 3.11	Birbirleri arasında yer değiştirmiş katman örneği	61
Tablo 4.1	Kullanılan veri setleri	63
Tablo 4.2	16 nesneli 3BKP örneği (Trivella ve Pisinger,2006)	64
Tablo 4.3	Matematiksel modeller için karşılaştırmalı sonuçlar.....	65
Tablo 4.4	Farklı senaryolar için karşılaştırmalı sonuçlar	67
Tablo 4.5	Farklı ağırlık kombinasyonları için karşılaştırmalı sonuçlar.....	69
Tablo 4.6	Filtre çeşitleri	70
Tablo 4.7	Uygulama sonuçları	71
Tablo 4.8	Senaryo I: Karşılaştırmalı sonuçlar (ürün ailesi yok)	73
Tablo 4.9	Senaryo II: Karşılaştırmalı sonuçlar.....	75
Tablo 4.10	Homojen kutu oranı.....	76
Tablo 4.11	Sezgisel algoritmanın farklı ağırlık kombinasyonları.....	77

Çok Amaçlı 3 Boyutlu Kutulama Problemlerinin Optimizasyonu İçin Matematiksel Model ve Genetik Algoritma Önerisi

Seda ERBAYRAK

Endüstri Mühendisliği Anabilim Dalı

Doktora Tezi

Danışman: Doç. Dr. Vildan ÇETİNSAYA ÖZKIR

Eş-Danışman: Dr. Öğr. Üyesi Umman Mahir YILDIRIM

3 Boyutlu Kutulama Problemleri (3BKP), kullanılan kutu sayısını en aza indirmek amacıyla farklı boyutlarda dikdörtgen şeklinde ki nesnelerin, kutulara yerleştirilmesi olarak tanımlanır. Bu tez çalışmasında, 3BKP' nin çözümü için matematiksel ve sezgisel model önerisi yapılmıştır. Önerilen yöntemler yardımı ile nesnelerin 2-yönlü döndürülmesine izin verildiği, aynı ürün gruplarının bir arada yüklendiği ve kutulardaki yük dengesinin sağlandığı yükleme planları oluşturulmuştur. Ele alınan problem için aynı ürün gruplarını birlikte paketlemeye teşvik eden aile birliği adı verilen yeni bir kavram literatüre tanıtılmıştır. Pratikte ürün bazlı paketlemenin yaygın kullanılmasına rağmen, literatür de aile birliği kavramını içeren çözüm yöntemi bulunmamaktadır.

Önerilen matematiksel model, aile birliği oranını maksimize ederken, kullanılan kutu sayısını ve ideal ağırlık merkezinden sapmayı minimize etmiş ve optimum yerleşme planlarının oluşturulması sağlanmıştır. Ancak problem aile sayısı, nesne sayısı gibi parametrelerdeki artış sebebiyle karmaşık hal aldığında çözüm

sürelerinde artış olmuştur. Bu sebeple büyük ölçekli problemlerin çözümü için genetik algoritma tabanlı sezgisel algoritma geliştirilmiştir. Önerilen sezgisel algoritmada, ilk olarak problem bir boyutlu kutulama problemi olarak ele alınmış ve genetik algoritma kullanılarak nesnelerin ürün ailelerine göre kutulara atanması sağlanmıştır. Sonrasında, bu atamalar dikkate alınarak, problem 3BKP olarak ele alınıp duvar örme algoritması kullanılarak paketleme yapılmıştır. Son aşamada ise her kutu için, dengeli nihai yerleşim planları oluşturulmuştur.

Önerilen matematiksel ve sezgisel algoritmanın performansı literatürden örnekler kullanılarak karşılaştırılmış ve benzer sonuçlar elde edildiği görülmüştür.

Anahtar Kelimeler: Kutulama problemleri, yük dengesi, aile birliği, konteyner yükleme.

Multi-Objective 3D Bin Packing Optimization: A Mathematical Model and Genetic Algorithm

Seda ERBAYRAK

Department of Industrial Engineering

Doctor of Philosophy Thesis

Supervisor: Assoc. Prof. Dr. Vildan ÇETİNSAYA ÖZKİR

Co-supervisor: Assist. Prof. Dr. Umman Mahir YILDIRIM

Three-dimensional bin packing problems (3D-BPP) are defined as packing rectangular items in to minimum number of bins. In this study, it is aimed to propose a mathematical and heuristic model for the solution of load balanced 3D-BPP, in which the orientation constraint is taken into account, the same product families are loaded together. In the problem addressed a new concept called family unity, which promote a family of items to be packaged together. Despite the widespread use of item related concerns in practice, there is no solution method that includes the concept of family unity in the literature.

In the proposed mathematical model, it minimized the number of bins used and the deviation of the balance from the ideal barycenter, and loading plans were created while maximizing the family unity ratio. However, due to the increase in parameters such as the number of families and the number of items, the problem has become more complex and the run times have increased. Therefore, heuristic algorithm has been developed for the solution of large-scale problems. In the proposed heuristic algorithm, firstly, the problem is considered as a one-dimensional bin packing problem and the items are assigned to the bins according

to the product families by using the genetic algorithm. Then, considering these assignments, the problem was handled as 3D-BPP and the settlement was made using the wall-building algorithm. At this stage, the family unity of the items is preserved as much as possible. In the final stage, balanced final packing plan are created for each bin.

The performance of the proposed mathematical and heuristic algorithm was compared using examples from the literature and it was seen that similar results were obtained.

Keywords: Bin packing, load balance, family unity, container loading.

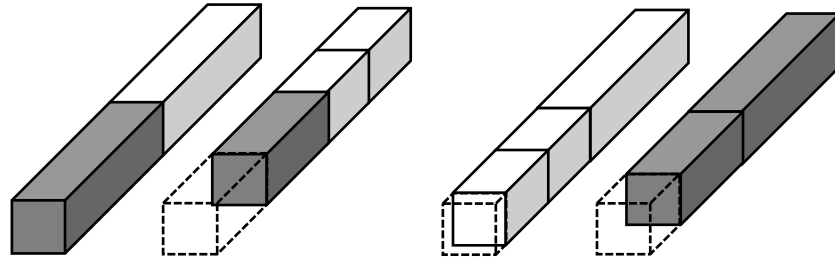
3 Boyutlu Kutulama Problemleri, ekonomi üzerinde güçlü bir etkisi olan, gerçek hayatta sıkça uygulanan bir kombinatorial optimizasyon problemidir (Ramos, Silva ve Oliveria, 2018). Geçmişten günümüze, işletmelerin nesneleri uygun kutularda paketleyip tedarik zincirleri boyunca farklı yerlere teslim etmesi lojistikte öne çıkan problemlerden birisidir. Paketleme ve nakliye için verimsiz planlama yöntemleri, maliyetleri artırır. Bu sebeple, ekonomik olarak verimlilik elde etmek isteyen araştırmacılar, NP-zor kutulama problemi için çeşitli matematiksel modeller ve sezgisel yöntemler önermişlerdir. Literatürdeki çok sayıda çalışma, birden fazla nesneyi birbirleri ile çakışmadan minimum sayıda kutuya yükleyerek nakliye maliyetlerini düşürmeye çalışmaktadır (Moon ve Nguyen, 2014).

Klasik 3BKP, minimum sayıda kutu kullanarak, dikdörtgen şeklinde ki bir dizi nesneyi paketlemeyi amaçlar. Bu amaca ek olarak, yük sabitliği, yük dengesi, istifleme gereksinimleri, aynı varış noktasına gidecek olan nesnelerin birlikte paketlenmesi gibi pratikte dikkate alınması gereken birçok faktör vardır. Literatürde yük sabitliği, nesnelerin taban alanının yerleştirildiği zemin tarafından kısmen veya tamamının desteklenmesi olarak tanımlanır. Yük dengesi ise, nesne ağırlıklarının kutu zemininde eşit olarak dağılması ile ilgilidir. Burada amaç, ideal ağırlık merkezi olarak belirlenen merkeze yakın yükleme planı oluşturmaktır. İdeal ağırlık merkezine uygun yükleme planları, kargo hasarları üzerindeki riskleri azaltır. Görünen önemine rağmen, yük sabitliği ve yük dengesi faktörleri genellikle kutulama literatüründe açıkça dikkate alınmamıştır (Bortfeldt ve Wascher, 2013). Bu nedenle, 3BKP'nin bu faktörleri hesaba katan uzantısına, yani yük dengeli 3BKP 'ye ilişkin literatür nispeten azdır.

Bu tez çalışmasında, ilk olarak kutulama problemleri için aile birliği adı verilen yeni bir kavram literatüre sunulmuştur. Literatür incelendiğinde, aile kavramı ile benzer kavramların bulunduğu görülmektedir. Ancak literatürde ürün ailesi

kavramı, aynı boyut veya özellikte ki öğeleri temsil etmek için kullanılırken, bu çalışmada biz dizi özelliği tanımlamak için kullanılmıştır. Bu yeni kavram, aynı varış noktasına sahip olma, taşıma gereksinimleri, depolama koşulları gibi pratik hususlar nedeniyle aynı kutuya paketlenmesi gereken nesneleri ele almaktadır. Aile birliğinin korunması, hem elleçleme için gerekli ekipman çizelgelemesinin karmaşıklığını azaltmada hem de elleçleme operasyonlarının yürütülmesinde daha iyi çözümler sağlar.

Şekil 1.1’de iki kutulu basit bir yükleme planını gösterilmiştir. Şekil 1.1 (a)’da her iki kutuyu yüklemek ve boşaltmak için birden fazla tipte taşıma ekipmanı gerekli olabilirken, Şekil 1.1 (b)’de her bir kutu tek tipte taşıma ekipmanı gerektirir. Ayrıca, herhangi bir elleçleme işlemi yapılmadan nihai varış noktasına vakit ve iş gücü kaybı olmadan gönderilebilir.



(a) Aile birliğinin dikkate alınmadığı plan (b) Aile birliğinin dikkate alındığı plan

Şekil 1.1 Yükleme planı

Bu çalışmada aile birliği kavramı ile literatürde mevcut olan aile kavramı birbirlerinden farklı kavramlar olduğunu dikkate almak gerekir. Literatürde, kutulama problemleri, tahsis kısıtlaması olarak tanımlanan kısıtlar altında incelenmiş ve müşteri bazlı yükleme planları oluşturulmuştur. Ancak bu kavram tam olarak bu çalışmada önerilen aile birliği kavramına karşılık gelmemektedir. Tahsis kısıtlamaları genellikle bağlantılı nesnelerin bir arada olması kısıtı ve çoklu aktarma kısıtı olarak karşımıza çıkmaktadır. Bağlantılı nesnelerin bir arada olması kısıtı, aynı müşteriye ait nesnelerin aynı kutuda birbirine yakın yerleştirilmesini sağlamak için tek kutu paketleme problemlerinde kullanılır. (Terno, Scheithauer, Sommerweiß, ve Riehme, 2000; Liu, Yue, Dong, Maple, ve Keech, 2011; Gimenez Palacios, Alonso, Valdes, ve Parreño, 2020). Tek kutulu paketleme probleminde,

bağlantılı nesnelerin bir arada olma kısıtını dikkate alan bir başka çalışma ise Egeblad, Garavelli, Lisi ve Pisinger (2010) tarafından sunulmuştur. Bu çalışmada, dikdörtgen olamayan nesneleri bir kutuya maksimum kar elde edecek şekilde yerleştirmeyi hedeflemiştir. Nesneleri kutuya yerleştirirken şekli aynı olan nesneleri birbirine yakın paketlemişlerdir. Ayrıca rotalama problemlerinde, çoklu aktarma kısıtı, karşımıza çıkmaktadır. Bu problemlerde, nesneler yerleştirilirken müşteriler dikkate alınarak atama yapılır. Her müşteriye ait nesne birbirine yakın yerleştirilmeli ve kutu müşteriye ulaştığında, müşteriye ait nesneler, başka müşterinin nesnelerini hareket ettirmeden erişim sağlanmalıdır. Genellikle kutu içerisinde, müşterilerin sipariş ettikleri nesnelerin alanı kadar alan müşterilere tahsis edilir. Sonrasında ise nesneler dağıtım sırasına göre, müşteri bazlı yerleştirilir (Gendreau, Iori, Laporte, ve Martello, 2006; Christensen ve Rousøe, 2009; Tarantilis, Zachariadis, ve Kiranoudis, 2009; Bortfeldt 2012; Ceschia ve Schaerf, 2013). Literatürdeki çalışmalar genellikle aynı müşteriye ait nesnelerin birbirine yakın paketlenmesine odaklanırken, bu tez çalışmasında, aynı ürün ailesindeki nesnelerin aynı kutuya atanması ve denge kısıtları dikkate alınarak kutu içinde paketlenmesine odaklanılmıştır.

Tez çalışmasının literatüre bir diğer katkısı ise, nesnelerin döndürülmesine izin verilen ürün aile birliğinin dikkate alındığı, yük dengeli 3BKP için çok amaçlı matematiksel model sunulmasıdır. Önerilen tam sayılı matematiksel model, denge ve döndürme gibi bir dizi kısıtın göz önünde bulundurulduğu, kullanılan kutu sayısı ve ideal ağırlık merkezinden sapmayı minimize ederken aile birliğini maksimum yapan ağırlıklandırılmış amaç fonksiyonuna sahip ilk matematiksel modeldir. Literatürde belirtilen kısıt ve amaçları aynı anda göz önünde bulunduran başka bir çalışma bulunmamaktadır. Son olarak, büyük ölçekli problemlerin çözümü için genetik algoritma tabanlı sezgisel algoritma geliştirmiş ve literatüre bu konuda da katkı sağlanmıştır.

1.1 Literatür Özeti

Kutulama problemleri, kesme ve paketleme ile ilgilenen araştırmacıların ilgisini çekmiştir. Problem gerçek hayat problemi olduğu için günümüzde popülerliğini hala devam ettirmektedir. 3BKP matematiksel modeller ile modellenip çözülebilsen

de, az sayıda kesin çözüm yöntemi vardır (Elhedhli, Gzara ve Yıldız, 2019). Kesin yöntemlere ek olarak, birçok araştırmacı problemin çözümü için sezgisel yöntem geliştirmeye odaklanmıştır (Chen, Lee ve Shen, 1995). Ayrıca, kutulama problemlerini çözmek için birçok meta-sezgisel ve hibrit yöntem önerilmiştir (Bortfeldt ve Wäscher, 2013; Elhedhli vd., 2019). Bu bölümde kutulama problemlerinin çözümü için kullanılan matematiksel modeller ve sezgisel tabanlı algoritmalar hakkında kapsamlı bir literatür özeti sunulmuştur.

1.1.1 Matematiksel Model Kullanılan Çalışmalar

Literatür özetinin bu bölümünde, kullanılan matematiksel modeller, problemin yapısı, kullanılan amaç ve kısıtlara göre incelenmiştir.

Literatürde, özellikle tekli kutulama problemlerinde, kullanılan alanı maksimum yapabilmek (Kang, Moon, ve Wang, 2012; Junqueira, Morabito ve Yamashita, 2012; Moon ve Nguyen, 2014; Araya, Guerrero ve Nuñez, 2017) veya bu amaca paralel olan kullanılmayan alanı minimum yapabilmek amaçlanmıştır (Chen vd., 1995). Egeblad ve Pisinger (2009) yaptıkları çalışmada, tekli kutulama problemlerinin çözümü için nesnelerin döndürülmesine izin verilmeyen bir tam sayılı matematiksel model önermişlerdir. Önerilen tam sayılı matematiksel model sayesinde yüklemekten elde edilen karın maksimum yapılması sağlanmıştır. Baldi, Perboli ve Tadei (2012), 3 boyutlu tekli kutulama problemlerinin çözümü için, karın maksimize edildiği ve denge kısıtlarının göz önünde bulundurulduğu bir tam sayılı matematiksel model literatüre sunmuşlardır. Önerilen matematiksel model, Egeblad ve Pisinger (2009) önerdikleri tam sayılı matematiksel model ile kıyaslanmış ve daha iyi performans gösterdiği kanıtlanmıştır. Ek olarak önerilen matematiksel model küçük boyutlu veri setlerinde optimum sonuçları elde ettiği gözlemlenmiştir. Moon ve Nguyen (2014) yaptıkları çalışmada tekli kutulama problemleri için, Baldi vd., (2012)'nin önerdikleri matematiksel modeli, hacim kullanımının maksimum düzeye çıkarılmasını dikkate alarak genişletmiştir. Ek olarak modele denge kısıtlarını da dahil etmişlerdir. Junqueira vd., (2012) tekli kutulama problemi için yük sabitliği ve yük dengesi sorunlarının ele alındığı tam sayılı matematiksel model sunmuşlardır.

Literatürde çoklu kutulama problemleri ile ilgili çalışmaların sayısında artış olmasına rağmen, hala çok azdır (Alonso, Alvarez-Valdes, Iori ve Parreño, 2019). Ayrıca Bortfeldt ve Wäscher (2013) çok kutulu paketleme problemlerindeki pratik kısıtların, tek kutulu paketleme literatürüne kıyasla neredeyse tamamının ihmal edildiğini ortaya koymuştur. Çoklu kutulama problemlerinde, en yaygın kullanılan amaç, kullanılan kutu sayısını minimize etmektir. Nesnelerin minimum kutu kullanılarak paketlenmesini amaçlayan ilk çalışma, Martello ve Vigo (1998) tarafından 2 boyutlu kutulama problemlerinin çözümü için formüle edilmiştir. Liu, Tan, Huang, Goh ve Ho (2008) iki boyutlu kutulama problemini çözmek için çok amaçlı matematiksel bir model önermiştir. Önerilen matematiksel modelde, kullanılan kutu sayısı minimize edilirken, aynı zamanda dengeli yükleme planlarının oluşturulması amaçlanmıştır. Ayrıca yaptıkları çalışmada kutuların döndürülme kısıtı ve sabitlik kısıtı göz önünde bulundurulmamıştır. Literatürde, 3 boyutlu kutulama problemlerinin çözümü için, döndürme kısıtının dikkate alındığı, farklı boyutlarda olan nesnelerin farklı boyutlarda olan kutulara yerleştirildiği ve kullanılmayan alanı minimize etmeye çalışan ilk tam sayılı matematiksel model Chen vd., (1995) tarafından sunulmuştur. Pedruzzi, Nunes, Rosa ve Arpini (2016) yaptıkları çalışmada, Chen vd., (1995) tarafından önerilen matematiksel modeli temel alarak yeni bir tam sayılı matematiksel model önermişlerdir. Önerilen matematiksel modele, nesnelerin taban alanının, yerleştirildiği zemin tarafından kısmen veya tamamının desteklenmesini sağlayan sabitlik kısıtı eklenmiştir. Ek olarak, nesneler kutulara yerleştirilirken teslim zamanları dikkate alınmış ve nesnelerin 2 yönlü döndürülmesine izin verilmiştir. Alonso, Alvarez-Valdes, Iori, Parreño ve Tamarit (2017) tarafından yapılan çalışmada, 3 boyutlu kutulama problemleri için, önce nesnelerin paletlere yüklenmesini sağlayan basit bir matematiksel model önerilmiştir. Sonrasında paletlerin kamyonlara yüklenmesi için önerilen matematiksel modeli daha karmaşık hale getirmişlerdir. Önerilen matematiksel model palet sayısını minimize ederken, kamyonlar için yük dengesi göz önünde bulundurulmuştur.

Bortfeldt ve Wäscher (2013) tarafından kutulama problemleri için kapsamlı bir literatür çalışması yapılmış ve pratikte kullanılan tüm kısıtları incelemişlerdir.

Tablo 1.1, literatürde 3 boyutlu çoklu kutulama problemi için önerilen matematiksel modellerindeki kısıtlamaları özetlemektedir.

Tablo 1.1 Matematiksel modellerindeki kısıtlamalar

Yayın	Amaç Fonksiyonu	Kutu		Kısıtlar		
		Eş	Değişken	Denge	Sabitlik	Dönme
Chen vd., (1995)	Min. Kullanılmayan alan		✓		✓	✓
Eley (2003)	Min. Kullanılan kutu sayısı		✓			
Trivella ve Pisinger (2016)	Amaç 1: Min. Kullanılan kutu sayısı Amaç 2: Min. İdeal ağırlık merkezinden sapma	✓		✓		
Pedruzzi vd., (2016)	Min. Kullanılmayan alan maliyeti		✓		✓	✓
Alonso vd., (2017)	Min. Kullanılan kutu sayısı	✓		✓		

3boyutlu kutulama problemleri için kullanılan matematiksel modeller üzerine yapılan çalışmalar, birkaç kısıtlamayı aynı anda karşılama konusunun tatmin edici bir şekilde ele alınmadığını göstermektedir. Bortfeldt ve Wäscher (2013) kutulama problemlerinin çözümüne yönelik, çok amaçlı yöntemlerin kullanılmasının iyi bir yöntem olduğunu vurgulamışlardır. Buna rağmen, literatürde çok amaçlı kutulama problemlerine nadir rastlanmaktadır. İlk çok amaçlı matematiksel model Liu vd., (2008) tarafından önerilmiştir. Önerilen matematiksel model 2 boyutlu kutulama problemlerin çözümü için kullanılmıştır. Literatürde 3 boyutlu kutulama problemlerinin çözümü için çok amaçlı matematiksel model önerisi yapılan ilk ve tek çalışma Trivella ve Pisinger'e (2016) aittir. Önerilen matematiksel model, kullanılan kutu sayısını ve ideal ağırlık merkezinden sapmayı minimize etmeyi amaçlamıştır. Ancak bu çalışmada nesnelerin döndürülmesine izin verilmemiştir.

1.1.2 Sezgisel Tabanlı Algoritmaların Kullanıldığı Çalışmalar

Literatürde matematiksel modellerin yetersiz kaldığı büyük boyutlu problemlerin çözümü için birçok sezgisel, meta-sezgisel ve hibrit yöntem önerilmiştir. Literatür özetinin bu bölümünde, kullanılan sezgisel tabanlı algoritmalar çeşitlerine göre gruplanmış ve kapsamlı şekilde sunulmuştur.

Çok boyutlu kutulama problemlerinin çözümüne yönelik ilk çalışma Gilmore ve Gomory (1963) tarafından yapılmıştır (Crainic, Perboli, Tadei, 2008). George ve Robinson (1980) 3 boyutlu kutulama problemlerinin çözümü için, nesnelerin döndürülmesine izin verilmeyen ve literatürde Duvar Örme Algoritması olarak isimlendirilen yeni bir algoritma önermişlerdir. Önerilen algoritma ile nesneler, kutu duvarlarına paralel katmanlar halinde yerleştirilmiştir. Yapılan bu çalışma, literatür de bulunan birçok çalışmanın öncüsü olmuştur. (Bischoff ve Ratcliff, 1995; Bortfeldt ve Gehring, 2001; Pisinger, 2002; Dereli ve Das, 2011; Lim, Ma, Qiu, Zhu, 2013, Sheng, Xiuqin, Changjian, Hongxia, Dayong ve Feiyue, 2017).

Moura ve Oliveira (2005) yaptıkları çalışmada, George ve Robinson (1980) tarafından önerilen algoritmayı modifiye ederek yeni bir algoritma önermişlerdir. Önerilen algoritmada, yerleştirilen nesnenin üzerinde oluşan yerleşim alanının sınırları belirlenirken, alttaki nesnenin boyutları dikkate alınmıştır. Böylelikle algoritmada, nesneler birbirini alttan destekledikleri için destekli yerleşim planı elde edilmiş ve kutuların sabitliği arttırılmıştır. Ayrıca nesnelerin döndürülmesine izin verilmiştir. Pisinger (2002) duvar örme yaklaşımına dayanan yeni bir sezgisel yöntem önermiştir. Bu çalışmada nesnelerin yerleştirme sırası belirlenirken klasik sıralama kriterleri yerine ağaç arama algoritması kullanılmıştır. Dereli ve Daş (2010) yükleme problemleri için duvar örme algoritması ve karınca kolonisi optimizasyonu yaklaşımını temel alan yeni algoritma önermiştir. Sheng vd., (2017) yaptıkları çalışmada, üç boyutlu kutulama problemini çözmek için sezgisel yaklaşım önermişlerdir. Önerilen algoritma, duvar örme algoritması tabanlı ağaç arama algoritmasıdır. Ayrıca, önerilen algoritma, nesnelerin 2-yönlü döndürme ve sabitlik kısıtını ele alır. Bu algoritma da, katmanlar belirlenirken kutunun hem enine hem boyuna paralel olacak şekilde denemeler yapılarak katmanlar

oluşturulmuş ve maksimum alan kullanımı sağlanmıştır. Ek olarak, nesneler yerleştirilirken son kullanma tarihlerine göre öncelik sırası belirlenmiştir.

Duvar örme algoritmasına benzer bir diğer algoritma ise katman oluşturma (layer building) algoritmasıdır. Duvar örme algoritmasından farkı, oluşturulan katmanlar kutunun duvarlarına paralel değil tabanına paralel olacak şekilde oluşturulur. Ratcliff ve Bischoff (1998) yaptıkları çalışmada, oluşturulan katmanlar kutunun tabanından yukarı doğru oluşturulmuştur. Katmanlar arasındaki yük taşıma limitleri göz önünde bulundurulmuştur. Önerilen algoritmada her adımda yük limiti kontrol edilerek, yük taşıma kabiliyeti yüksek olan nesnelerin öncelikli olarak yerleştirilmesi sağlanmıştır.

Loh ve Nee (1992) kutulama problemlerinin çözümü için yeni bir katman oluşturma algoritması önermişlerdir. Önerilen algoritma da nesnelerin yükleme sıralamasını belirleyen yeni bir strateji kullanılmıştır. Öncelikle nesneleri yüksekliklerine göre gruplara ayırmış ve grupları uzundan kısaya doğru sıralamıştır. Her bir grubun içerisindeki nesneleri ise kendi içlerinde taban alanına göre büyükten küçüğe doğru sıralamış ve bu sıralamaya göre katman katman yerleştirilmiştir. Lodi, Martello ve Vigo (2002), Loh ve Nee (1992) yaptıkları çalışmada ki sıralama stratejine benzer bir strateji izleyerek yeni bir yerleştirme algoritması önermişlerdir. Bu çalışmada gruplar oluşturulurken belirli bir yükseklik toleransı içindeyse nesneler gruba dâhil edilmiştir. Yani belirli yükseklik aralığı belirlenip, o aralıkta ki her bir nesnenin ilgili gruba dâhil olması sağlanır. Önerilen sezgisel algoritma tabu arama algoritması ile birleştirilmiştir.

Literatürde yerleştirme problemlerinin çözüm yöntemleri incelendiğinde genetik algoritmanın sıklıkla kullanıldığı görülmektedir. Genetik algoritma, yerleştirme algoritmaları ile birleştirildiğinde iyi sonuçlar verdiği açıkça gözlemlenmektedir (örn: Pimpawat ve Chaiyaratana, 2001; Bortfeldt ve Gehring, 2001; Wu, Li, Goh ve Souza, 2010; Ramos, Oliveira, Gonçalves ve Lopes, 2016; Jamrus ve Chien, 2016; Kabasakal ve Keskin, 2017; Ramos vd., 2018)

Lin, Foote, Pulat, Chang ve Cheung (1993) yaptıkları çalışmada genetik algoritma, katman oluşturma algoritması, tavlama benzetimi algoritması kullanarak melez bir yöntem önermişlerdir. Önerilen sezgisel yöntem ile ağır olan kutuları alta, hafif

olan kutular ise üste yerleştirilmesi sağlanmıştır. Önerilen algoritma da kutuların döndürülmesine izin verilmediği gibi denge kısıtları da dikkate alınmamıştır. Feng, Moon, ve Shin, (2015) genetik algoritma ve en geniş sol boş alana ilk sıradaki nesnenin atanması stratejisini kullanarak melez bir sezgisel algoritma önermişlerdir. Yapılan çalışmada nesnelerin döndürülmesine izin verilmiştir. Kang vd., (2012) döndürme ve sabitlik kısıtları altında kutulama problemlerinin çözümü için yeni bir yöntem geliştirmişlerdir. Literatürde en dip alt sol doldurma (deepest bottom left fill) algoritması olarak bilinen, Karabulut ve İnceoğlu (2004) tarafından önerilen yerleştirme algoritmasını genetik algoritma ile hibritlemiş ve üstünlüklerini kanıtlamışlardır. Moon ve Nguyen (2014) yaptıkları çalışmada genetik algoritma, en dip alt sol doldurma algoritması, açgözlü arama algoritması kullanarak melez bir yöntem önermişlerdir. Önerilen yöntemde nesnelerin döndürülme ve denge kısıtları dikkate alınmıştır.

Gonzalez, Mirvea ve Leon, (2016) yaptıkları çalışmada, kullanılan hacim ve kutu ağırlığını maksimum yapabilmek amacıyla, nesnelerin döndürme kısıtlarını içeren bir yöntem sunmuşlardır. Kullanılan yerleştirme algoritmasında katmanlar, kutunun enine ve boyuna paralel olacak şekilde iki farklı yöntemle yerleştirilmektedir. Önerilen yerleştirme algoritması genetik algoritma ile birleştirilmiş ve sonuçları incelenmiştir. Önerilen algoritmanın daha az çeşit nesne kullanıldığında daha iyi sonuç verdiği görülmüştür. Gonçaves ve Resende (2013) tarafından 2 ve 3 boyutlu kutulama problemlerinin çözümü için Rassal Anahtar (random key) Tabanlı Genetik Algoritma (BRKGA) önerilmiştir. Önerilen algoritmada klasik kısıtların yanı sıra nesnelerin döndürme kısıtı da kullanılmıştır. Ayrıca, genetik algoritmanın uygunluk fonksiyonu da yeni önerilmiş ve çözüme pozitif yönde katkı sağlamıştır. Yeung ve Tang (2005) yaptıkları çalışma da genetik algoritmayı alt sol köşe yaklaşımı yerleştirme algoritması ile hibritlemişlerdir. Önerilen algoritma sonuçları incelendiğinde dengeli yükleme planlarının oluşturulduğu gözlemlenmiştir.

Kutulama problemlerinin çözümü için sezgisel yöntemlerin kullanıldığı çalışmaların sayısı oldukça fazladır. Literatürdeki 3 boyutlu kutulama problemleri için kullanılan kısıtlar ve önerilen sezgisel yöntemler Tablo 1.2'de özetlenmiştir.

Tablo 1.2 Literatürde 3BKP çözümü için kullanılan sezgisel yaklaşımlar

Yayın	Tek Kutulu	Çok Kutulu		Kısıtlar			Yöntem ve Amaç
		Eş	Değişke	Denge	Sabitlik	Dönme	
Martello, Pisinger ve Vigo (2000)		✓					Yöntem: Dal ve Sınır Algoritması Amaç: Minimum kutu kullanımı
Eley (2002)		✓		✓	✓	✓	Yöntem: Açgözlü Algoritma + Ağaç Arama Algoritması Amaç: Minimum kutu kullanımı
Lodi vd., (2002)		✓					Yöntem: Tabu Arama Amaç: Minimum kutu kullanımı
Pisinger (2002)	✓				✓	✓	Yöntem: Ağaç arama tabanlı algoritması Amaç: Maksimum hacim kullanımı
Bortfeldt, Gehring ve Mack (2003)	✓				✓	✓	Yöntem: Tabu Arama Amaç: Maksimum hacim kullanımı
Castro Silva, Soma ve Maculan (2003)		✓		✓		✓	Yöntem: Açgözlü Algoritma Amaç: Minimum kutu kullanımı
Lodi, Martello ve Vigo (2004)		✓				✓	Yöntem: Tabu Arama Amaç: Minimum kutu kullanımı
Moura ve Oliveira (2005)	✓				✓	✓	Yöntem: Açgözlü tesadüfi adaptif arama prosedürü (GRASP) Amaç: Maksimum hacim kullanımı
Parreño, Alvarez-Valdes, Tamarit, Oliveira (2008)	✓						Yöntem: Açgözlü tesadüfi adaptif arama prosedürü Amaç: Minimum kutu kullanımı
Tarantilis vd., (2009)		✓			✓	✓	Yöntem: Tabu Arama + Yerel Arama Amaç: Minimum maliyet
Fuellerer, Doerner, Hartl ve Iori (2010)		✓			✓	✓	Yöntem: Karınca Kolonisi Algoritması Amaç: Minimum maliyet

Tablo 1.2 Literatürde 3BKP çözümü için kullanılan sezgisel yaklaşımlar (devamı)

Wang, Guo, Chen, Zhu ve Lim (2010)	✓				✓	✓	Yöntem: En Derin Alt Sol Algoritması+ Tabu Arama Amaç: Maksimum hacim kullanımı
Liu, vd., (2011)	✓			✓	✓	✓	Yöntem: Tabu Arama Amaç: Maksimum hacim kullanımı
Almeida ve Figueiredo (2010)		✓				✓	Yöntem: Alt sınır algoritması Amaç: Minimum kutu kullanımı
Fanslau ve Bortfeldt (2010)	✓				✓	✓	Yöntem: Ağaç Arama Algoritması Amaç: Maksimum hacim kullanımı
Lim, Ma, Xu ve Zhang (2012)		✓					Yöntem: Açgözlü Sezgisel Algoritma Amaç: Minimum kutu kullanımı
Araya ve Riff (2014)	✓				✓	✓	Yöntem: Işın Arama Stratejisi Amaç: Maksimum hacim kullanımı
Crainic vd., (2016)			✓				Yöntem: Aşamalı Riskten Korunma Prensibi Amaç: Minimum maliyet
Trivella ve Pisinger (2016)		✓		✓			Yöntem: Grafik Gösterim Yöntemi Amaç: Minimum kutu kullanımı+ Minimum İdeal ağırlık merkezinden sapma

Kutulama problemlerinin çözümü için matematiksel yöntemlere göre sezgisel algoritma tabanlı araştırmalar daha ileri düzeydedir. Ancak şimdiye kadar birkaç kısıtlamayı aynı anda karşılama konusunun tatmin edici bir şekilde ele alınmadığı gözlemlenmiştir.

1.2 Tezin Amacı

Lojistik operasyonlarda malzemelerin başlangıçtan nihai kullanıcıya kadar etkin ulaşımı çok önem taşımaktadır. Şirketler özellikle malzemelerin taşınma öncesinde kutulara nasıl paketleneyeceği ile ilgili çalışmalar yapmaktadırlar. Literatürde bu problem 3 boyutlu kutulama problemleri olarak isimlendirilir. 3 boyutlu kutulama problemleri hem teori hem pratik uygulamalarda sıklıkla kullanıldığı için günümüzde önemini korumaktadır.

Yapılan tez çalışmasında, paketleme için verimsiz planlama yöntemleri sebebiyle oluşan fazla iş gücü, gereğinden fazla kutu kullanılması, kargo hasarları, elleçleme gibi nedenlerden oluşan maliyetlerin düşürülmesi yönünde fayda sağlayacağı düşünülmektedir. Bu amaç doğrultusunda, nesnelerin 2-yönlü dönmesine izin verilen, ürün aile birliğinin dikkate alındığı, yük dengeli 3BKP çözümüne yönelik yöntemlerin geliştirilmesi sağlanmıştır.

Bu tez çalışmasının literatüre katkısı aşağıda maddeler halinde özetlenmiştir:

1. Aile birliği adı verilen yeni bir kavram sunmak
2. Literatürde aynı anda incelenmeyen pratik kısıtları ve amaçları aynı anda göz önünde bulunduran çok amaçlı tam sayılı matematiksel model geliştirmek
3. Büyük ölçekli problemlerin çözümü için genetik algoritma tabanlı sezgisel algoritma geliştirmek

Sunulan çözümle, küçük ve büyük boyutlu problemler için en uygun paketleme planları oluşturmak amaçlanmıştır.

1.3 Hipotez

Bu çalışmada, nesnelerin 2-yönlü dönmesine izin verilen, ürün aile birliğinin dikkate alındığı, yük dengeli 3BKP çözümüne yönelik matematiksel model ve genetik algoritma tabanlı sezgisel algoritma geliştirilmiştir.

Tezde incelenmek istenen ana hipotez aşağıda belirtilmiştir:

Hipotez: Önerilen sezgisel algoritma ve matematiksel model ile optimale yakın yükleme planlarının elde edilmesi.

1.4 Tezin İçeriği

Çalışmanın ilk bölümünde, tezin konusuyla ilgili kapsamlı bir literatür özeti sunulmuş, tezin amacı, hipotezi ve içeriği belirtilmiştir. Çalışmanın ikinci bölümünde, kutulama problemlerinin sınıflandırılması, çözüm yöntemleri ve kullanılan kısıtlar hakkında detaylı bilgi verilmiştir. Üçüncü bölümde, tez çalışmasında ele alınan problemin tanımı yapılmış ve problemin çözümüne yönelik önerilen matematiksel model ve genetik algoritma tabanlı sezgisel

algoritmanın ayrıntılı detayları sunulmuştur. Dördüncü bölümde, ilk olarak kullanılan veri kümesi tanıtılmıştır. Sonrasında önerilen matematiksel modelin performansını göstermek için karşılaştırmalı sayısal bir analiz sunulmuş ve önerilen modelin avantajları gösterilmiştir. Sonrasında, bir filtre fabrikasının ürün ailesi bazlı yükleme problemi ele alınmış ve önerilen matematiksel modelin uygulaması yapılmıştır. Dördüncü bölümün devam eden kısmında, önerilen sezgisel algoritmanın performansı literatür ile karşılaştırılmış ve ürün ailesi kavramının çözüm üzerindeki etkisi incelenmiştir. Çalışmanın son bölümü olan beşinci bölümde, gelecekteki araştırmalar için sonuçlar yorumlanmış ve öneriler sunulmuştur.

Kutulama Problemleri, NP-zor kombinatoriyal optimizasyon problemidir. Problem en genel şekliyle, farklı veya aynı boyutta dikdörtgen şeklindeki nesnelerin kutu kapasitesini aşmayacak şekilde paketlenmesi olarak tanımlanmaktadır. Problem polinom zamanda çözülmemesine rağmen, büyük boyuttaki problemler için geliştirilen algoritmalar sayesinde optimal çözümler elde edilebilmektedir.

Bu bölümde kutulama problemlerinin sınıflandırılması ve pratik kısıtları hakkında ayrıntılı bilgi verilecektir

2.1 Kutulama Problemlerinin Sınıflandırılması

Kutulama problemleri NP-zor kombinatoriyal problem sınıfına girmektedir. Mühendislik problemlerinin birçok alanında ana problem veya başka bir problemin bir parçası olarak karşımıza çıkmaktadır. Wäscher, Haußner ve Schumann (2007), kutulama problemlerini paketlenecek nesne boyutuna, kullanılan amaç türüne, nesne çeşitliliğine, nesne şekillerine ve nesne çeşitliliğine göre ilerleyen bölümlerde ayrıntısı verilen beş alt başlık altında sınıflandırmışlardır.

2.1.1 Paketlenecek Nesne Boyutuna Göre Sınıflandırma

Kutulama problemleri bir, iki ve üç boyutlu olmak üzere 3 kategoriye ayrılmaktadır.

2.1.1.1 Bir Boyutlu Kutulama Problemleri

Bir boyutlu kutulama problemlerinde kutu ve nesneler hacim veya ağırlık gibi tek bir parametre kullanılarak tanımlanmaktadır. Klasik bir boyutlu kutulama problemleri, kapasitesi c_j olan kutulara, hacimleri v_i olan n tane nesnenin kutu kapasitesini aşmadan yerleştirilmesi olarak tanımlanmaktadır. Bir boyutlu kutulama problemlerinin örnek gösterimi Şekil 2.1’de sunulmuştur.

Bir boyutlu problemlerinde en çok kullanılan amaçlar, kullanılan kutu sayısını veya kullanılmayan kutu hacmini minimum yapmaktır. Genellikle büyük yoğunluk ve boyutta olan nesneler yerleştirilirken problem bir boyutlu ele alınmaktadır. Klasik bir boyutlu kutulama probleminin matematiksel modeli aşağıda sunulmuştur.



Şekil 2.1 Bir boyutlu kutulama problemi gösterimi

Modelin karar değişkenleri, Tablo 2.1’ de özetlenmiştir.

Tablo 2.1 Kullanılan karar değişkenler

Sembol	Açıklama
p_{ij}	Eğer i nesnesi j kutusuna paketlenmişse 1, aksi halde 0.
u_j	Eğer j kutusu kullanılıyorsa 1, aksi halde 0.

1BKP’nin çözümüne yönelik yaygın olarak kullanılan toplam kutu sayısını en aza indirmektir. (Denklem 2.1)

$$\text{Minimize } \sum_j u_j \quad (2.1)$$

$$\sum_{i=1}^{|I|} v_i p_{ij} \leq c_j u_j \quad \forall j \in J \quad (2.2)$$

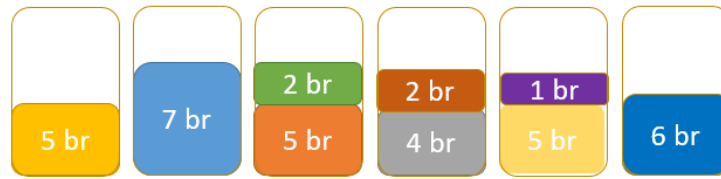
$$\sum_{j=1}^{|J|} p_{ij} = 1 \quad \forall i \in I \quad (2.3)$$

$$p_{ij}, u_j \in \{0,1\} \quad (2.4)$$

Kısıt (2.2), j kutusundaki nesnelerin toplam hacminin, kutunun kapasitesini aşmamasını sağlarken, Kısıt (2.3), her bir i nesnesinin paketlenmesini sağlar. Son olarak, kısıt (2.4)'de ikili değişkenler belirtilmiştir.

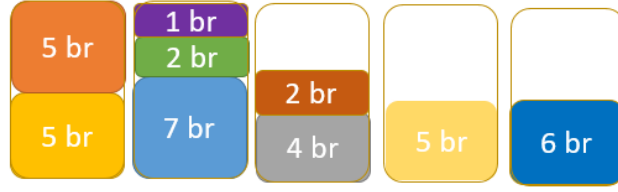
Bir boyutlu problemlerin çözümü için algoritmalar çevrimiçi ve çevrimdışı algoritmalar olarak iki gruba ayrılır. Çevrimiçi algoritmalar dinamik yapıdadır. Nesnelerin yerleşme sırası önceden bilinmez. Çevrimdışı algoritmalarda ise, paketleme işleminin başında paketlenen nesneler kümesinin tamamı bilinmektedir. Çevrimiçi algoritmalarından bazıları; En yakın yer algoritması, İlk uygun yer algoritması, En uygun yer algoritması, En kötü yer algoritmasıdır. Sıklıkla kullanılan çevrimdışı algoritmalarından bazıları ise; Azalan ilk uygun yer algoritması ve Azalan en uygun yer algoritmasıdır. Bu algoritmalar neredeyse optimum sonuca ulaşabilmektedir.

- En yakın yer algoritması (Next Fit): Yerleştirilecek nesne geldiğinde, nesnenin mevcut kutuya sığıp sığmadığı kontrol edilir. Eğer mevcut kutuda yer varsa içine yerleştirilir. Aksi durumda mevcut kutu kapatılır ve yeni bir kutu açılarak nesne yerleştirilir. Kapatılan kutular yeni nesne geldiğinde tekrar açılmazlar. Tüm nesneler kutulara yerleştirilinceye kadar bu prosedüre devam edilir. Şekil 2.1' de hacimleri verilen nesnelerin kapasitesi 10 birim olan kutulara en yakın yer algoritması ile yerleştirilmiş ve Şekil 2.2' de gösterilmiştir.



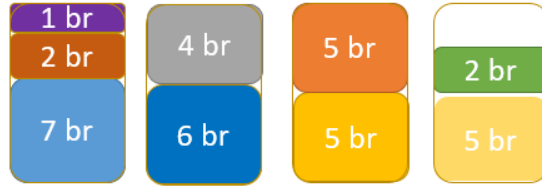
Şekil 2.2 En yakın yer algoritması

- İlk uygun yer algoritması (First Fit): Bu algoritmada yerleştirilecek nesne geldiğinde, mevcut kutular sıra ile taranır ve nesne ilk sığıldığı kutuya yerleştirilir. Eğer mevcut kutularda yer yoksa yeni bir kutu açılır ve nesne yerleştirilir. En yakın yer algoritmasına göre daha iyi sonuçlar elde edilir.
- Şekil 2.3'de ilk uygun yer algoritması ile nesnelerin kutulara nasıl yerleştiği gösterilmiştir. En yakın yer algoritmasına göre kullanılan kutu sayısı azalmıştır.



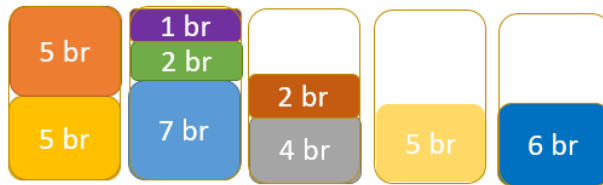
Şekil 2.3 İlk uygun yer algoritması

- Azalan ilk uygun yer algoritması (First Fit Decreasing): Bu algoritma ilk uygun yer algoritmasının çevrimdışı halidir. Nesneler boyutlarına göre azalan sıra ile sıralanır ve ardından ilk uygun yer algoritması uygulanır. Şekil 2.4’de azalan ilk uygun yer algoritması ile nesnelerin kutulara nasıl yerleştiği gösterilmiştir. Çevrimiçi algoritma sınıfında olan göre kullanılan kutu sayısı azalmıştır.



Şekil 2.4 Azalan ilk uygun yer algoritması

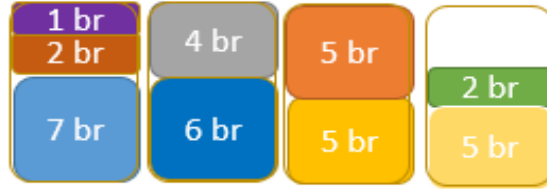
- En uygun yer algoritması (Best Fit): Algoritmanın yapısı ilk uygun yer algoritması ile benzerdir. Bu algortmada da yerleşecek nesne geldiğinde, mevcut kutular sıra ile taranır. Ancak burada kutular taranırken, nesne hangi kutuda daha az artık alan kalıyorsa ona yerleştirilir. Eğer mevcut kutulardan hiçbiri o nesneye uygun değilse yeni bir kutuya geçilir ve nesne bu kutuya atanır. Şekil 2.5’de En uygun yer algoritması ile nesnelerin kutulara nasıl yerleştiği gösterilmiştir.



Şekil 2.5 En uygun yer algoritması

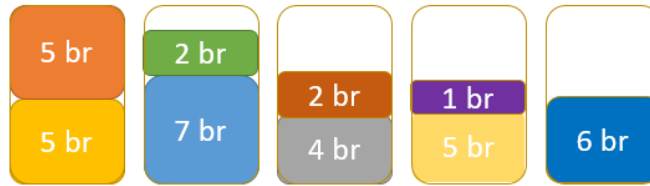
- Azalan en uygun yer algoritması (Best Fit Decreasing): Bu algoritma en uygun yer algoritmasının çevrimdışı halidir. Nesneler boyutlarına göre azalan sıra ile

sıralanır ve ardından en uygun yer algoritması uygulanır. Şekil 2.6’da Azalan en uygun yer algoritması ile nesnelerin kutulara nasıl yerleştiği gösterilmiştir.



Şekil 2.6 Azalan en uygun yer algoritması

- En kötü yer algoritması (Worst Fit): Bu algoritma, en uygun yer algoritmasının tam tersi prensiple çalışır. Yerleşecek nesne geldiğinde, kutular sıra ile taranırken hangi kutuda daha fazla artık alan kalıyorsa ona yerleştirilir. Eğer mevcut kutularda yer yoksa yeni bir kutu açılır ve nesne yerleştirilir. Şekil 2.7’de en kötü yer algoritması ile nesnelerin kutulara nasıl yerleştiği gösterilmiştir.



Şekil 2.7 En kötü yer algoritması

2.1.1.2 İki Boyutlu Kutulama Problemleri

İki boyutlu kutulama problemlerinde, nesnenin uzunluk (l_i)- genişlik (w_i) boyutları veya genişlik (w_i) - yükseklik (h_i) boyutları veya uzunluk (l_i)-yükseklik (h_i) boyutları gibi nesnenin iki boyutu dikkate alınır. Örneğin kâğıt, tel, kumaş gibi malzemelerin paketlemesinde problem iki boyutlu ele alınmaktadır. Klasik iki boyutlu kutulama probleminin matematiksel modeli aşağıda sunulmuştur. Modelin karar değişkenleri Tablo 2.2’ de özetlenmiştir.

Tablo 2.2 Kullanılan karar değişkenler (2 Boyutlu)

Sembol	Açıklama
p_{ij}	Eğer i nesnesi j kutusuna paketlenmişse 1, aksi halde 0.
u_j	Eğer j kutusu kullanılıyorsa 1, aksi halde 0.

Tablo 2.2 Kullanılan karar değişkenler (2 Boyutlu) (devamı)

x_{ik}^+	Eğer aynı kutuda i nesnesi k nesnesinin soluna yerleştirilmişse 1, aksi halde 0.
x_{ik}^-	Eğer aynı kutuda i nesnesi k nesnesinin sağına yerleştirilmişse 1, aksi halde 0.
y_{ik}^+	Eğer aynı kutuda i nesnesi k nesnesinin arkasına yerleştirilmişse 1, aksi halde 0.
y_{ik}^-	Eğer aynı kutuda i nesnesi k nesnesinin önüne yerleştirilmişse 1, aksi halde 0.
x_i, y_i, z_i	i nesnesinin yerleştiği sol-alt-arka koordinatları gösteren sürekli değişkenler

2BKP’de de klasik amaç kullanılan toplam kutu sayısını en aza indirmektir. (Denklem 2.5)

$$\text{Minimize } \sum_j^{|J|} u_j \quad (2.5)$$

$$p_{ij} \leq u_j \quad \forall i \in I, \forall j \in J \quad (2.6)$$

$$\sum_{j=1}^{|J|} p_{ij} = 1 \quad \forall i \in I \quad (2.7)$$

$$x_i + l_i \leq L_j u_j + (1 - p_{ij})M \quad \forall i \in I, \forall j \in J \quad (2.8)$$

$$y_i + w_i \leq W_j u_j + (1 - p_{ij})M \quad \forall i \in I, \forall j \in J \quad (2.9)$$

$$x_i + l_i \leq x_k + (1 - x_{ik}^+)M \quad \forall i, k \in I, i < k \quad (2.10)$$

$$x_k + l_k \leq x_i + (1 - x_{ik}^-)M \quad \forall i, k \in I, i < k \quad (2.11)$$

$$y_i + w_i \leq y_k + (1 - y_{ik}^+)M \quad \forall i, k \in I, i < k \quad (2.12)$$

$$y_k + w_k \leq y_i + (1 - y_{ik}^-)M \quad \forall i, k \in I, i < k \quad (2.13)$$

$$x_{ik}^+ + x_{ik}^- + y_{ik}^+ + y_{ik}^- \geq p_{ij} + p_{kj} - 1 \quad \forall i, k \in I, i < k \quad (2.14)$$

$$p_{ij}, u_j, x_{ik}^+, x_{ik}^-, y_{ik}^+, y_{ik}^- \in \{0,1\} \quad (2.15)$$

$$x_i, y_i, z_i \geq 0 \quad (2.16)$$

Modelin kısıtları Denklem (2.6)-(2.16)'da sunulmuştur. Denklem (2.6) i nesnesinin j kutusuna sadece j kutusunun kullanılması durumunda atanmasını sağlar. Denklem (2.7)'de her bir i nesnesinin paketlenmesini sağlar. Denklem (2.8)-(2.9), M 'nin büyük bir skaler olduğu durumlarda, nesnelerin kutunun boyutlarını aşmadan kutuya yerleştirilmesini sağlar. Denklem (2.10)-(2.13), Eğer i ve k nesneleri aynı kutuda paketleniyse, birbirleri ile çakışmasını önler. Denklem (2.14) ise, aynı kutuda i ve k nesnelerinin, birbirinin soluna, sağına, arkasına veya önüne yerleşmesini sağlar. Son olarak, Denklem (2.15)-(2.16)'de ikili ve negatif olmayan değişkenler belirtilmiştir.

İki boyutlu kutulama problemlerinin çözümü için klasik yerleştirme algoritmaları aşağıda özetlenmiştir.

- Aşağı sol algoritması: Yerleştirilecek dikdörtgen şeklindeki nesneler rassal veya istenilen öncelik sırasına göre sıralanır ve ilk nesne iki boyutlu dikdörtgen şeklindeki kutunun sol alt boş köşesine yerleştirilir. Tüm nesneler öncelik sırasına göre kutuda en sol boş alt alana yerleştirilmeye devam edilir. Eğer mevcut kutuda yer yoksa yeni bir kutu açılır ve nesne yerleştirilir.
- Geliştirilmiş aşağı sol algoritması: Yerleştirilecek istenilen öncelik sırasına göre sıralanır ve ilk nesne iki boyutlu dikdörtgen şeklindeki kutunun sol alt boş köşesine yerleştirilir. Sonrasında gelen nesne sağ üst köşeden başlayarak kaydırılır ve solda başka bir nesne ile temas edinceye kadar kaydırılır. En alt sol alanda temas edilen alanda yeterli alan varsa yerleştirilir. Tüm nesneler kaydırma tekniği kullanılarak kutunun sol alt alanına yerleştirilir.
- İlk uygun azalan algoritması: Yerleştirilecek nesnelerin verilen 2 boyutu birbirleri ile çarpılır ve alanları belirlenir. Alanları belirlenen nesneler büyükten küçüğe doğru sıralanır ve aşağı sol, geliştirilmiş sol gibi yerleştirme algoritmaları kullanılarak yerleştirilir.

İki boyutlu kutulama problemlerinin çözümü için klasik yöntemlerin yanında tabu arama, kalınca kolonisi, tavlama benzetimi, genetik algoritma gibi sezgisel algoritmalar sıklıkla kullanılmaktadır.

2.1.1.3 Üç Boyutlu Kutulama Problemleri

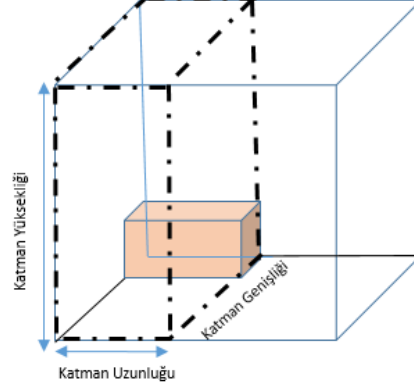
Klasik 3 Boyutlu Kutulama Problemleri, bir dizi nesnenin minimum sayıda kutuya yerleştirilmesi olarak tanımlanır. Bu problem türünde nesnenin uzunluk, genişlik ve yükseklik boyutları dikkate alınarak yerleştirme yapılmaktadır. Teoride ve pratikte en çok uygulaması bulunan problem çeşididir.

Üç boyutlu kutulama problemleri için kullanılan matematiksel modeller hakkında ayrıntılı bilgi Bölüm 3’de sunulmuştur.

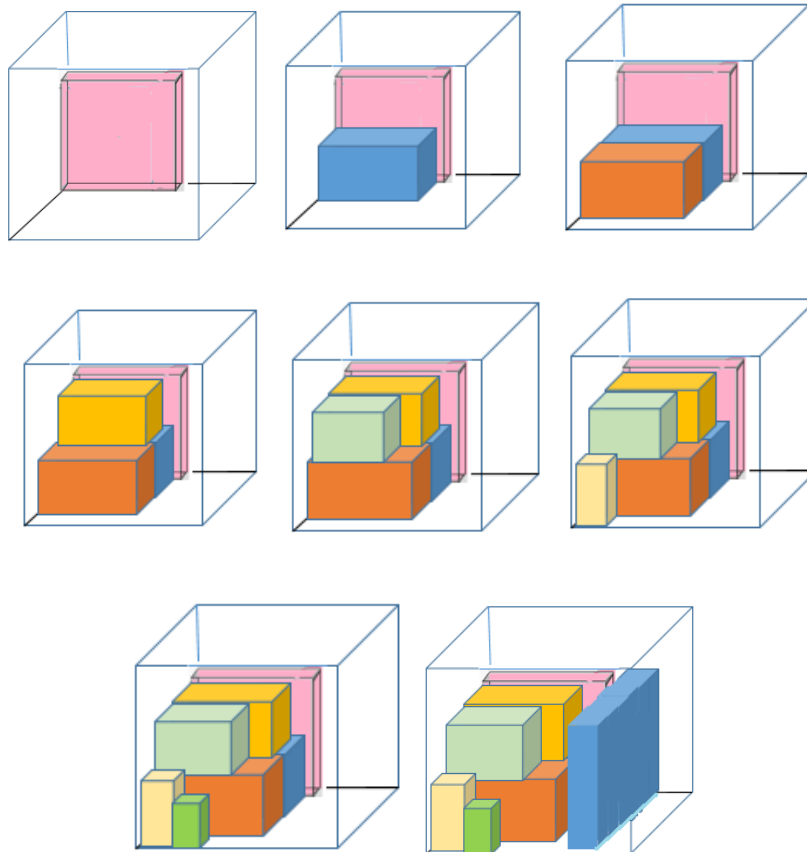
Üç boyutlu kutulama problemlerinin çözümü için birçok yerleştirme algoritması geliştirilmiştir. Bu algoritmalarından en çok kullanılan algoritmalar aşağıda özetlenmiştir. Ek olarak yerleştirme algoritmalarının daha iyi sonuç verebilmesi için genetik algoritma, tabu arama algoritması, tavlama benzetimi, parçacık sürü optimizasyonu, karınca algoritması gibi algoritmalar ile hibritlenmektedir.

- Duvar Örme Algoritması (Wall Building): Duvar örme algoritması literatürde en sık kullanılan algoritmalarından bir tanesidir. İlk olarak George ve Robinson (1980) tarafından 3 boyutlu konteyner yükleme problemlerinin çözümü için önerilmiş ve birçok çalışmaya öncülük yapmıştır. Bu sebeple literatürde birçok varyasyonu bulunmaktadır. Bu yöntemde, nesneler kutu duvarlarına paralel olacak şekilde katmanlar kullanılarak yerleştirilir. Yerleştirilecek nesnelerin öncelik sırasının nasıl belirleneceği hakkında farklı sıralama yöntemleri önerilmiştir. Bunlardan bazıları, nesneleri hacim, genişlik uzunluk veya taban alanlarına göre büyükten küçüğe sıralamayı içerir. Belirlenen sıralama yöntemine göre sıralanan nesnelerden birinci sıradaki nesne kutuya yerleştirilir ve ilk katman oluşturulur. Oluşturulan katmanın görsel hali Şekil 2.8’de sunulmuştur. Öncelik sırasına göre nesneler, bir önceki yerleştirilen nesnenin önce önüne, sonra yanına, en son üzerine yerleştirilmeye çalışılır. Katmanda yeterli alan kalmayınca kadar bu işlem devam eder. Katmanda yeterli alan kalmadığı zaman yeni katman oluşturulur. Bu işlem tüm

nesneler yerleştirilene kadar devam eder. İki katmanlı bir kutunun örnek yükleme adımları Şekil 2.9'da gösterildiği gibidir. Bu yöntem özellikle nesne çeşitliliğinin az olduğu durumlarda daha iyi sonuç verir. Ek olarak nesne çeşitliliğinin az olduğu durumlarda duvar örme algoritması kullanılması ile yük sabitliği de sağlanmış olur.



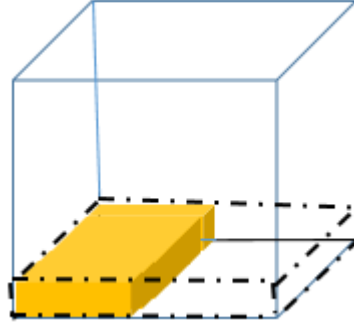
Şekil 2.8 Katman örneęi



Şekil 2.9 Örnek yükleme adımları

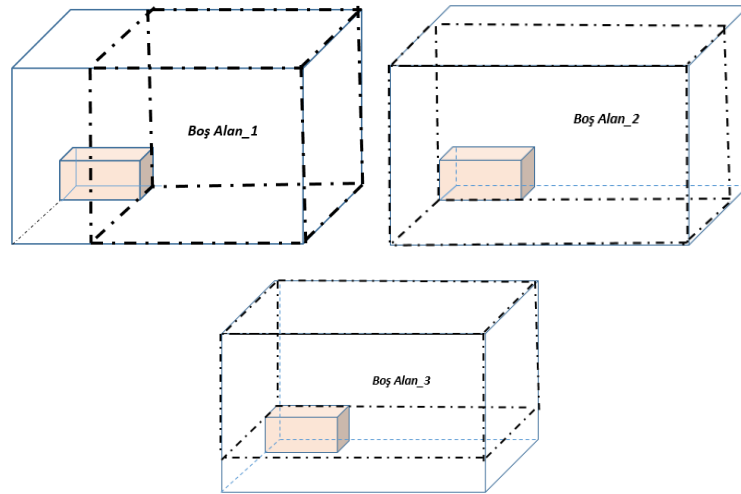
- Katman Oluřturma Algoritması (Layer Building): Duvar örme algoritmasına benzer bir dięer algoritma ise katman oluřturma

algoritmasıdır. Duvar örme algoritmasından farkı oluşturulan katmanlar kutunun duvarlarına paralel değil Şekil 2.10'da gösterildiği gibi tabanına paralel olacak şekilde oluşturulur. Bu yöntemde kutunun yüksekliği izin verdiği sürece yükleme yapılır.



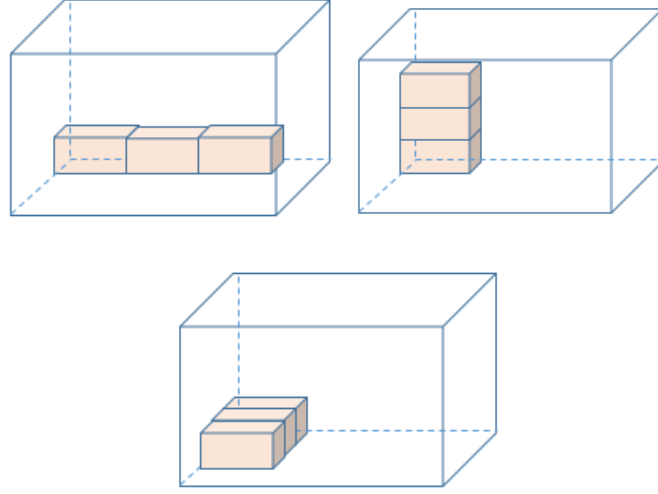
Şekil 2.10 Katman örneği

- Maksimum Boş Alan Algoritması (Maximal Space): İlk olarak kutulama problemlerin çözümü için Lai ve Chan (1997) tarafından önerilmiştir. Maksimum boş alan, kutunun boş hacmini tamamen kaplayan boşluklar kümesidir (Şekil 2.11). İlk nesne yerleştikten sonra boş alanlar listelenir. Bir sonraki nesne, nesnenin sığabileceği, kutunun sol alt köşesine en yakın olan boş alana yerleştirilir ve bu alan boş alan listesinden silinerek liste güncellenir. Tüm nesneler yerleşene kadar prosedür devam eder.



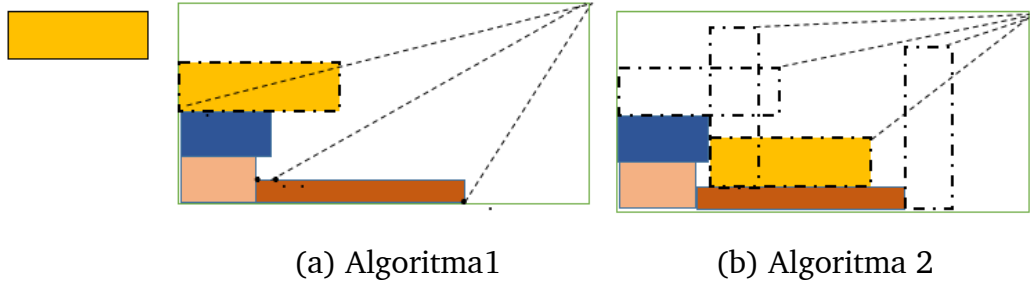
Şekil 2.11 Maksimum boş alanlar

- Blok Oluşturma Algoritması (Block Building): Bu yöntem, çoğunlukla aynı boyut ve özellikteki nesnelerin paketlenmesinde kullanılır. Burada nesneler Şekil 2.12'de gösterildiği gibi bloklar halinde yüklenir.



Şekil 2.12 Yatay ve dikey ekseninde oluşturulmuş blok örnekleri

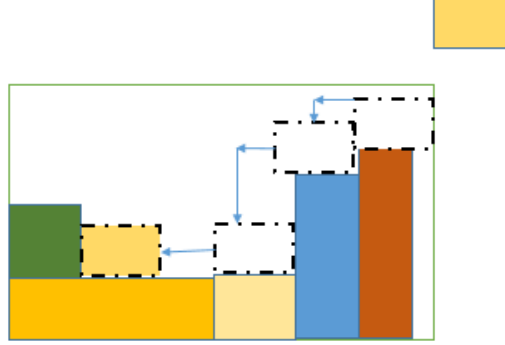
- **Ön Üst Sağ Köşe Algoritması (Distance to the Front-Top-Right Corner):** Gonçaves ve Resende (2013) tarafından literatüre sunulmuştur. Bu yöntemde öncelikle maksimum boş alan algoritmasına göre boş alanlar belirlenir. Belirlenen boş alanların her birinin sol alt köşe noktasının kutunun ön üst sağ köşe noktasına olan uzaklığı hesaplanır. Bu uzaklığın maksimum olduğu noktaya nesne yerleştirilir. (Şekil 2.13 (a)). Sonrasında bu yöntem geliştirilerek algoritma güncellenmiştir. Güncellenen algoritmada belirlenen boş alanların her birinin sol alt köşe noktasına (minimum koordinatı) nesne tüm yönleri ile yerleştirilir. Her bir denemede nesnenin sağ üst köşe noktasının kutunun sağ üst köşe noktasına uzaklığı hesaplanır ve maksimum olduğu noktaya yerleştirilir. (Şekil 2.13 (b)).



Şekil 2.13 Ön üst sağ köşe algoritması (Gonçaves ve Resende, 2013)

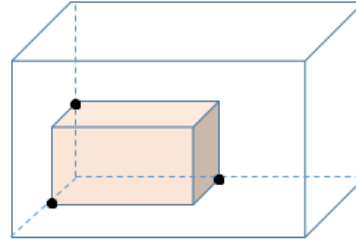
- **Arka Sol Alt Algoritması (Back Bottom Left):** Liu ve Teng (1999) yaptıkları çalışmada kutulama problemlerinin çözümü için Arka Sol Alt Algoritmasını

önermişlerdir. Önerilen algoritmada, yerleşecek nesne, kutunun sağ üst köşe noktasından başlayarak alt sol köşe noktasına kadar taranır ve nesne alt sol köşe noktasında en yakın olan alana yerleştirilir. (Şekil 2.14)



Şekil 2.14 Arka sol alt algoritması

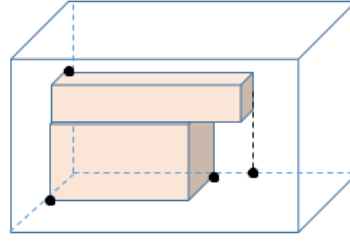
- Köşe noktalar algoritması (Corner Points): Bu algoritmada yerleşen nesnenin sol üst arka, sol alt ön ve sağ alt arka noktaları köşe noktalar olarak tanımlanır ve yeni yerleşecek nesne bu noktalardan bir tanesine yerleştirilir. Şekil 2.15’de kutuya yerleştirilen nesnenin köşe noktaları gösterilmiştir.



Şekil 2.15 Köşe noktaların gösterimi

- Uç Noktalar algoritması (Extreme Points): Uç Noktalar algoritması köşe noktalar algoritmasının bir uzantısıdır. Yüklenen nesnelerin köşe noktaları ve bu noktaların iz düşümleri yeni nesnelerin yerleşebileceği aday noktalar olarak listelenir.

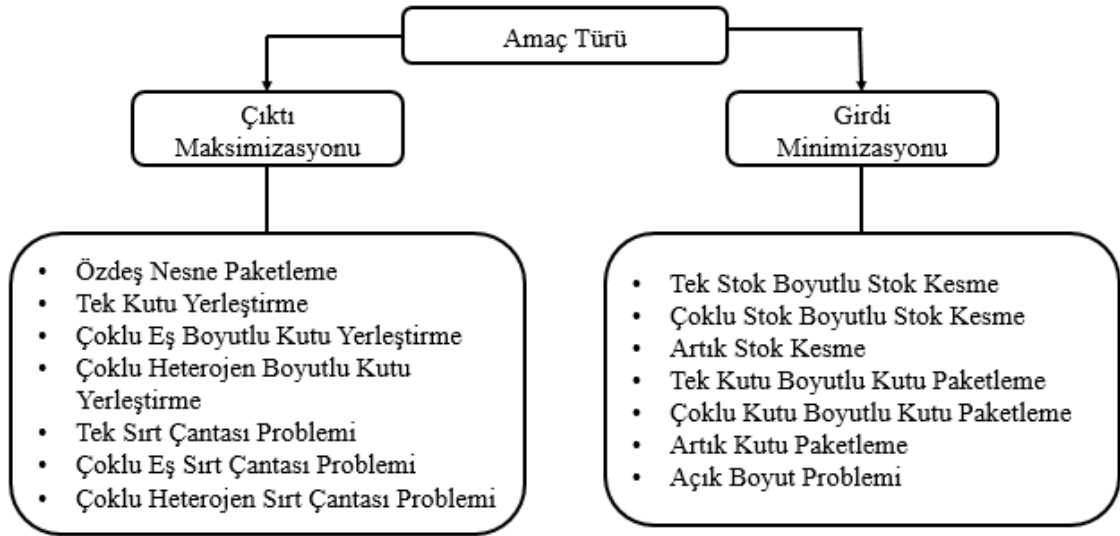
Şekil 2.16’da kutuya yerleştirilen kutuların köşe noktaları ve uygun iz düşümü gösterilmiştir. Uç noktalar algoritmasında tüm bu noktalar yerleşime aday noktalardır.



Şekil 2.16 Uç noktaların gösterimi

2.1.2 Kullanılan Amaç Türüne Göre Sınıflandırma

Kutulama problemlerinde amaçlar çıktı maksimizasyonu ve girdi minimizasyonu olarak ikiye ayrılmaktadır. Çıktı maksimizasyonu kutulara mümkün olduğu kadar fazla nesnenin atanması olarak tanımlanırken, girdi minimizasyonu, n tane nesnenin en az sayıda kutuya atanması olarak tanımlanır. Wäscher vd., (2007) yaptıkları çalışmada amaç türüne göre sınıflandırılan problemleri, kutu ve nesne çeşitliliğine göre Şekil 2.17’de gösterildiği gibi alt sınıflara ayırmışlardır (Uner, 2019).



Şekil 2.17 Kutulama problemlerinin amaç türüne göre sınıflandırılması

- **Tek Stok Boyutlu Kesme Problemi:** Tüm kutuların eş olduğu, nesne çeşitliliğinin ise zayıf heterojen olduğu problem türüdür. Nesnelerin en az sayıda kutuya yüklenmesi amaçlanmıştır.

- Çoklu Stok Boyutlu Stok Kesme Problemi: Hem nesne çeşitliliğinin hem de kutu çeşitliliğinin zayıf heterojen olduğu, en az sayıda kutu kullanımının amaçlandığı problem çeşididir.
- Artık Stok Kesme Problemi: Nesne çeşitliliğinin zayıf heterojen, kutu çeşitliliğinin ise güçlü heterojen olduğu problem türüdür.
- Tek Kutu Boyutlu Kutu Paketleme Problemi: Nesne çeşitliliğinin güçlü heterojen, kutuların ise eş olduğu problem türüdür.
- Çoklu Kutu Boyutlu Kutu Paketleme Problemi: Nesne çeşitliliği güçlü heterojen, kutu çeşitliliğinin ise zayıf heterojen olduğu problem çeşididir.
- Artık Kutu Paketleme: Güçlü heterojen kutuların ve nesnelerin yüklendiği problem türüdür.
- Açık Boyut Problemi: Problem tek kutulu paketleme problemidir. Bir veya daha fazla değişken boyuta sahip bir grup nesnenin tek bir kutuya yüklenmesi olarak tanımlanır.
- Özdeş Nesne Paketleme Problemi: Tek kutulu problem çeşitlerinden bir tanesidir. Birbirine eş nesneler kutu içerisine yerleştirilir.
- Tek Kutu Yerleştirme Problemi: Nesne çeşitliliğinin zayıf heterojen olan bir grup nesnenin tek kutuya paketlenmesi olarak tanımlanır.
- Çoklu Eş Boyutlu Yerleştirme Problemi: Çok kutulu problem çeşitlerinden bir tanesidir. Nesneler zayıf heterojen boyutlu, kutular ise birbirlerine eştir.
- Çoklu Heterojen Boyutlu Yerleştirme Problemi: Heterojen boyutlarda olan birden fazla kutuya, zayıf heterojen boyutlu nesnelerin yerleştirilmesi problemidir.
- Tek Sırt Çantası Problemi: Güçlü heterojen boyutlardaki nesnelerin tek bir kutuya yüklenmesi problemidir.
- Çoklu Eş Sırt Çantası Problemi: Birden fazla eş boyutlu kutuya, güçlü heterojen boyutlu nesnelerin yerleştirilmesi problemidir.
- Çoklu Heterojen Sırt Çantası Problemi: Heterojen boyutlarda olan birden fazla kutuya, güçlü heterojen boyutlu nesnelerin yerleştirilmesi problemidir.

Tablo 2.3' de problem çeşitlerinin kutu ve nesne özelliklerine göre birbirlerinden farklılıkları özetlenmiştir.

Tablo 2.3 Kutulama problemlerinin birbirinden farklılıkları

Amaç	Problem	Kutu				Nesne		
		Tüm Boyutlar Sabit			Değişken Boyutlu Tek Kutu			
		Eş	Zayıf Heteroje	Güçlü Heteroje		Eş	Zayıf Heteroje	Güçlü Heteroje
Çıktı Maksimizasyonu	Özdeş Nesne Paketleme	✓				✓		
	Tek Kutu Yerleştirme	✓					✓	
	Çoklu Eş Boyutlu Kutu Yerleştirme	✓					✓	
	Çoklu Heterojen Boyutlu Kutu Yerleştirme		✓	✓			✓	
	Tek Sırt Çantası Problemi	✓						✓
	Çoklu Eş Sırt Çantası Problemi	✓						✓
	Çoklu Heterojen Sırt Çantası Problemi		✓	✓				✓
Girdi Minimasyonu	Tek Stok Boyutlu Stok Kesme	✓					✓	
	Çoklu Stok Boyutlu Stok Kesme		✓				✓	
	Artık Stok Kesme			✓			✓	
	Tek Kutu Boyutlu Kutu Paketleme	✓						✓
	Çoklu Kutu Boyutlu Kutu Paketleme		✓					✓
	Artık Kutu Paketleme			✓				✓
	Açık Boyut Problemi				✓			

2.1.3 Nesne Çeşitliliğine Göre Sınıflandırma

Kutulama problemlerinde nesne çeşitliliği paketlenen nesnelerin ölçülerine göre belirlenmektedir. Literatürde nesne çeşitliliği özdeş, zayıf heterojen ve güçlü heterojen olarak sınıflanmaktadır. Tüm nesneler aynı çeşit ve boyutta ise özdeş olarak tanımlanmaktadır. Nesneler birkaç farklı boyutta ise nesne çeşitliliği zayıf heterojen olarak tanımlanırken nesnelerden birkaçı aynı boyutta ise nesne çeşitliliği güçlü heterojen olarak tanımlanmaktadır.

2.1.4 Nesne Şekillerine Göre Sınıflandırma

Literatürde nesne şekilleri en temel hali ile düzgün ve düzgün olmayan şekiller olarak ayrılmaktadır. İki ve üç boyutlu kutulama problemlerinde paketlenen

nesnelerin geometrik şekilleri kare, dikdörtgen, yuvarlak, silindir gibi düzgün şekilli olabileceği gibi düzgün olmayan şekillerde de olabilmektedir.

2.1.5 Kutu Çeşitliliğine Göre Sınıflandırma

Pratikte ve teoride yapılan tüm çalışmalarda, problem tipini belirleyen en önemli faktörlerden bir tanesidir. Kutulama problemleri tek ve çok kutulu olmak üzere iki şekilde gruplanmaktadır. Tek kutulu paketleme problemlerinde kutuların tüm boyutları eş olabileceği gibi bir veya birden fazla boyutu değişken olabilir. Çok kutulu paketleme problemleri, tıpkı nesne çeşitliliği gibi özdeş, zayıf heterojen ve güçlü heterojen olarak sınıflanmaktadır. Ele alınan tüm problem çeşitlerinde kutular dikdörtgen veya küp şeklinde olduğu varsayılmıştır.

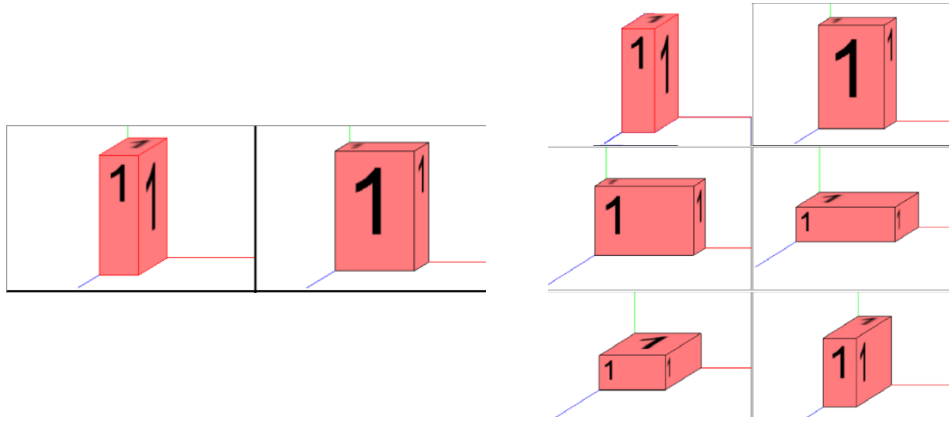
2.2 Kısıtlar

3BKP’i hem pratikte hem teoride farklı kısıtlar altında incelenmiştir. Kullanılan kısıtlar gerçek hayat problemleri dikkate alınarak belirlenirse, uygulama sonrasında elde edilen çözümler uygulanabilir olacaktır. Bortfeldt ve Wäscher (2013) tarafından kutulama problemlerinde kullanılan kısıtlamalar hakkında kapsamlı bir çalışma literatüre sunulmuştur.

Nesnelerin kutu sınırları içerisine yerleştirilmesi ve birbirleri ile çakışmamasını sağlamak kutulama problemlerinin en temel iki kısıtıdır. Ancak bu kısıtlamalar pratikte yetersiz kalmaktadır. Bu sebeple, problem birden çok kısıt altında incelenmiş ve problem çözümleri uygulanabilir hale getirilmiştir. Literatürde sıklıkla kullanılan kısıtlar aşağıda maddeler halinde açıklanmıştır.

- **Döndürme kısıtı:** Literatürde döndürme kısıtı sıkça ele alınmıştır. Literatürde döndürme kısıtı ile ilgili bazı çalışmalar şunlardır: Eley (2002); Pisinger (2002); Bortfeldt vd., (2003); Castro Silva vd., (2003); Lodi vd., (2004); Moura ve Oliveira (2005); Tarantilis vd., (2009); Fuellerer vd., (2010); Iori ve Martello (2010); Wang vd., (2010); He ve Huang, 2011; Liu vd., (2011); Almeida ve Figueiredo, 2010; Fanslau ve Bortfeldt, 2010; Queiroz vd., (2012); Lim vd., (2012); Araya ve Riff, 2014; Chen vd., (1995); Pedruzzi vd., (2016). Döndürme kısıtı, taşınacak olan nesnenin istiflenme ve taşınma özelliklerine bağlıdır. Bazı nesnelerin herhangi bir

eksende döndürülmesine izin verilmezken bazılarının Şekil 2.18' de gösterildiği gibi 2-yönlü veya 6-yönlü döndürülmesine izin verilmektedir. Örneğin buzdolabı gibi hassas taşınması gereken nesneler 2-yönlü döndürülmeye izin verilirken kuru gıdalar 6- yönlü döndürülerek yükleme yapılmasına izin verilmektedir. Özellikle gerçek hayat problemlerinde nesnelerin zarar görmemesi için sadece dikey eksende yani nesne tabanı üzerinde döndürülmesine izin verilmektedir.



(a) 2- yönlü döndürme

(b) 6-yönlü döndürme

Şekil 2.18 Döndürme şekilleri

- **Ağırlık Kısıtı:** Kutu içerisine yerleşen nesnelerin toplam ağırlığının, izin verilen ağırlık limitini aşmaması olarak tanımlanmaktadır. Paketlenecek nesneler köpük, kauçuk gibi hafif malzemelerden oluşuyorsa ağırlık kısıtı dikkate alınmamaktadır. Ancak ağır nesnelerin yüklenmesi yapıldığında bu kısıt araştırmacılar tarafından probleme dâhil edilmiştir (Bortfeldt ve Wäscher, 2013). Ağırlık kısıtının dikkate alındığı çalışmalara örnek olarak; Liu vd., (2011), Costa ve Captivo, (2016), ve Paquay vd., (2018) yaptıkları çalışmalar verilebilir.
- **Sabitlik (Stabilite) Kısıtı:** Yük sabitliği yatay ve dikey olmak üzere iki ayrı şekilde sağlanmaktadır. Literatürde dikey yük sabitliği, nesnelerin taban alanının, yerleştirildiği zemin tarafından kısmen veya tamamının desteklenmesi olarak tanımlanırken, paketlenmiş her nesne bir başka nesne veya kutu duvarına bitişik olacak şekilde paketlenmesi olarak tanımlanır. Yük sabitliğinin sağlanmasındaki başlıca amaç, nesnelerin kutu zeminine

veya birbirleri üzerine düşmesini önlemektir. Uygulamada nesneler arasındaki boşluklar köpük gibi dolgu malzemesi kullanılarak sabitlenir ve yük sabitliği sağlanır (Bortfeldt ve Wäscher, 2013). Böylelikle nakliye esnasında nesneler zarar görmeyecek şekilde sabitlenmiş olurlar. Pratikteki öneminden dolayı araştırmacılar yoğun olmasa da bu yönde çalışmalar yapmıştır. Literatürde sabitlik kısıtı ile ilgili bazı çalışmalar şunlardır: Eley (2002); Pisinger (2002); Bortfeldt vd., (2003); Moura ve Oliveira (2005); Tarantilis vd., (2009); Fuellerer vd., (2010); Iori ve Martello, 2010; Wang vd., (2010); Liu vd., (2011); Fanslau ve Bortfeldt (2010); Araya ve Riff (2014); Pedruzzi vd., (2016).

- **Denge Kısıtı:** Bu kısıt, ağırlığın kutu içerisinde nasıl dağıldığı ile ilgilenir. Bu konuda literatürde 2 farklı alan çalışma yapılmaktadır. Birincisi konteyner, palet gibi ekipmanlara yükleme yapıldığında, ağırlık merkezinin ideal ağırlık merkezine yakın olması amaçlanmaktadır. Kutunun ideal ağırlık merkezi, kutunun geometrik merkezi olduğu varsayılmıştır. İkincisi kamyon, tır gibi ekipmanlara yükleme yapıldığında her bir aks başına düşen ağırlık optimize edilmeye çalışılır. Çünkü burada aksların dayanacağı yük sınırlıdır. Eğer akslarını taşıyamayacağı şekilde yükleme yapılırsa, taşıma esnasında kargo ve yol güvenliği açısından riskli durumlar meydana gelebilir (Uner,2019). Ayrıca güvenlik açısından risk oluşturmamasının yanında, dengesiz yüklenen kutuların taşınması sırasında yakıt tüketiminde de artış olmaktadır (Trivella ve Pisinger, 2016). Literatürde denge kısıtını dikkate alan çalışmalara örnek olarak, Eley (2002); Costa ve Captivo (2016); Trivella ve Pisinger (2016); Alonso vd., (2017) yaptıkları çalışmalar verilebilir.
- **Yükleme Önceliği Kısıtı:** Bu kısıt, kutu içerisine yüklenecek nesnelerin hangilerinin öncelikli olarak yüklenmesi ile ilgilenir. Mesela birden çok teslimat noktasında gidecek olan kamyon yükleme yapılırken, teslimat sırasına göre nesneler yüklenir. Ek olarak, son kullanma süreleri yaklaşan nesnelerin veya raf ömrü kısa olan bozulacak nesnelerin öncelikli yüklenmesi gibi durumlarda problem bu kısıt altında incelenebilir. Jamrus

ve Chien (2016) yaptıkları çalışma yükleme önceliği kısıtını inceleyen çalışmalara örnek verilebilir (Kır, 2019).

- **Yük Dayanım Kısıtı:** Bu kısıt, nesnelerin yük taşıma kapasitesi dikkate alınarak üst üste nasıl yerleştirileceğini belirleme ihtiyacından doğmuştur. Nesnenin büyük veya ağır olması, nesnenin üzerine konulacak olan nesneyi taşıyabileceği anlamına gelmez. Bir nesnenin ne kadar ağırlığa veya basınca dayanabileceği en temel olarak birim başına uygulanabilecek maksimum ağırlığının belirlenmesi ile ölçülür. Yük dayanımı kısıtı dikkate alındığında nesnelerde hasarlar meydana gelmez.
- **İstifleme Kısıtı:** Nesneler tasarımları sebebiyle farklı istifleme özelliklerine sahip olabilirler. İstifleme kısıtı, bu tür nesneleri üst üste yüklememeye veya ayrı paketlemeye teşvik eder.
- **Bağlantı Kısıtı:** Bu kısıt, tek kutu paketleme problemlerinde, aynı müşteriye ait nesnelerin aynı kutuda birbirine yakın yerleştirilmesini sağlar. Literatürde bağlantı kısıtı ile ilgili bazı çalışmalar şunlardır: Terno vd., (2000); Liu vd., (2011); Palacios vd., (2020).
- **Çoklu Aktarma Kısıtı:** Çoklu aktarma kısıtı, rotalama problemlerinde, karşımıza çıkmaktadır. Bu problemlerde, müşteri odaklı yerleştirme yapılır. Aynı müşteriye ait nesneler kutu içerisinde birbirine yakın paketlenir. Kutu müşteriye ulaştığında, müşteriye ait nesneler, başka müşterinin nesnelerini hareket ettirmeden erişim sağlanmalıdır. Nesneler dağıtım noktalarına göre yüklenmelidir. Genellikle kutu içerisinde, müşterilerin sipariş ettikleri ürünlerin alanı kadar alan müşterilere tahsis edilir. Sonrasında ise nesneler dağıtım sırasına göre, LIFO gibi stratejiler kullanılarak müşteri bazlı yerleştirilir. Literatürde çoklu aktarma kısıtının kullanıldığı çalışmalara örnek olarak; Gendreau vd., (2006), Christensen ve Rousøe (2009), Tarantilis vd., (2009), Bortfeldt (2012), Ceschia ve Schaerf (2013) tarafından yapılan çalışmalar verilebilir.

Bu tez çalışmasında yük dengeli 3BKP çözümü için tam sayılı matematiksel model ve genetik algoritma tabanlı sezgisel algoritma geliştirilmiştir. Bu bölümde, ele alınan problem ve geliştirilen yöntemler hakkında ayrıntılı bilgi verilmiştir.

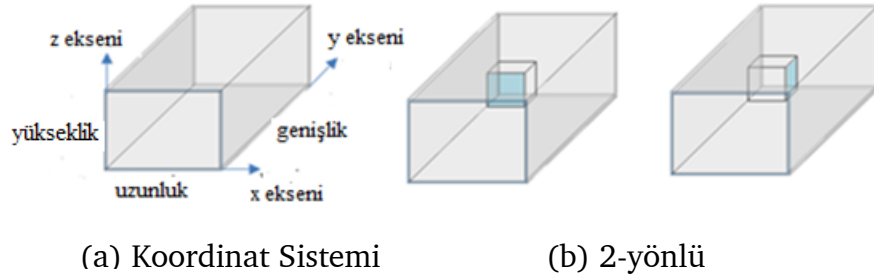
3.1 Problem Tanımı

Çoklu kutulama problemleri, NP-zor optimizasyon problemidir. Problem en genel şekliyle, farklı veya aynı boyutta dikdörtgen şeklindeki nesnelerin minimum sayıda kutu kullanılarak paketlenmesi olarak tanımlanmaktadır.

Bu tez çalışmasında ele alınan problem yük dengeli 3BKP'dir. Problemde, n nesne sayısı ve $i = \{1, 2, \dots, n\}$ olmak üzere, her bir nesnenin, uzunluğu l_i , genişliği w_i , yüksekliği h_i ve ağırlığı a_i olarak tanımlanmıştır. Probleme literatürden farklı olarak nesnelerin teslim aşamasını dolaylı yoldan kolaylaştırmak için ürün ailesi kavramı eklenmiştir. Bu kavram aynı hedefe gidecek bir dizi nesne olarak düşünülmüştür. Bir nesne sadece bir ürün ailesinin f üyesi olabilir. i nesnesinin f ürün ailesine ait olup olmadığı α_{if} ikili parametresi ile ifade edilmiştir. Ürün ailesi sadece bir nesneden oluşabileceği gibi, tüm nesneler tek bir ürün ailenin üyesi olabilir. Olabilecek en iyi yerleşim planı, aynı ürün ailesinden olan nesnelerin aynı kutuya yerleşerek aile birliğinin maksimum performans göstermesidir. Eğer bu yerleşim mümkün değilse, dağıtım noktası sayısının az olması için kutuların içerisindeki toplam ürün aile sayısı mümkün olduğunca az tutulmalıdır.

Nesne yüzeyleri kutu yüzeyine paralel olacak şekilde ortogonal yerleşim yapılmıştır. Nesneler uzunluğu L_j , genişliği W_j ve yüksekliği H_j olan kutulara yerleştirilmiştir. Yerleşim yapılırken, j kutusuna yerleşen nesnelerin toplam ağırlığı, yük taşıma(A_j) kapasitesini aşmamalıdır. Her j kutusu, orijini sol-alt-arka noktası olarak belirlenen üç boyutlu kartezyen koordinat sistemi ile tanımlanmıştır. Koordinat sisteminin x eksen, y eksen ve z eksen sırasıyla

kutunun uzunluęu, genişlięi ve yükseklięi olduęu varsayılmıřtır. Ek olarak, her bir kutunun ideal aęırlık merkezinin sırasıyla x eksenini, y eksenini ve z eksenini üzerindeki $\left(\frac{L_j}{2}, \frac{W_j}{2}, 0\right)$ noktalara karřılık gelen kutu tabanının orta noktasında olduęu varsayılmıřtır. Kutuları temsil eden koordinat sistemi řekil 3.1 (a)'da tanımlanmıřtır.



řekil 3.1 Kutu koordinatları ve nesne yönelimi

3BKP’de nesnelerin 6-yönlü döndürölmesine izin verilmektedir. Fakat yükleme operasyonların çoęunda nesnelerin sadece tabanları üzerinde döndürölmesine izin verilirken, bař üstü döndürölmesine izin verilmez. Bu tez alıřmasında, řekil 3.1(b)’de gösterildięi gibi nesnelerin 2-yönlü döndürölmesine izin verilmiřtir.

Bu alıřmada, 3BKP’ne kontrollü döndürme, kapasite, yük dengeleme, nesne ile baęlantılı kısıtlamalar dâhil edilerek daha kapsamlı bir kutulama problemi sunulmuřtur. Problemin amacı, yük dengesi ve aile birlięini saęlayanın yanında, bir dizi nesnenin minimum sayıda kutulara paketlenmesidir.

3.2 Önerilen Matematiksel Model

Bu bölümde, 3BKP özümü için, denge ve paketlenecek nesne ile ilgili pratik kısıtları içeren ok amaçlı tam sayılı matematiksel model sunulmuřtur. Yük dengeli 3BKP, NP-zor optimizasyon sınıfına girmektedir (Trivella ve Pisinger, 2016). Probleme yeni kısıtlar ve amaç dâhil edildięinde problem daha karmařık hale gelmiřtir. Önerilen matematiksel modelin odak noktası, büyük boyutlu problemleri özme becerisinden ok, aile birlięi dâhil edildięinde matematiksel modelin davranıřını incelemektir.

Önerilen model oluşturulurken, Chen vd., (1995), Liu vd., (2008), Trivella ve Pisinger (2016) yaptıkları çalışmalar temel alınmıştır. Tablo 3.1’de önerilen matematiksel modelin, temel alınan çalışmalardan farklılıkları gösterilmektedir.

Tablo 3.1 Matematiksel modeller arasındaki benzerlikler ve farklılıklar

Yayın	Amaç Fonksiyonu	Problem Boyutu	Kısıtlar			
			Döndürme	Denge	Sabitlik	Aile Birliği
Chen vd., (1995)	Min. Kullanılmayan alan	3 Boyutlu	6-yönlü		✓	
Liu vd., (2008)	Amaç 1: Min. Boş alan Amaç 2: Min. İdeal ağırlık merkezinden sapma	2 Boyutlu		✓		
Trivella ve Pisinger (2016)	Amaç 1: Min. Kullanılan kutu sayısı Amaç 2: Min. İdeal ağırlık merkezinden sapma	3 Boyutlu		✓		
Önerilen Model	Amaç 1: Min. Kullanılan kutu sayısı Amaç 2: Min. İdeal ağırlık merkezinden sapma Amaç 3: Mak. Aile birliği	3 Boyutlu	2-yönlü	✓	✓	✓

Önerilen matematiksel model, problemin doğası gereği kullanılan temel kısıtlamalar (nesnelerin çakışmamasını sağlayan, nesnelerin döndürülmesine izin veren, kutu kapasitesini aşmayan, her nesnenin bir kutuya paketlenmesini sağlayan) bakımından Chen vd., (1995) tarafından sunulan matematiksel model ile benzerlik gösterirken, denge ve aile birliği açısından farklılıklar göstermektedir. Önerilen matematiksel model, Liu vd., (2008) tarafından önerilen çok amaçlı matematiksel model ile yük dengeleme açısından benzerlik göstermektedir. Ancak bu model iki boyutlu kutulama problemlerin çözümü için kullanılmaktadır. Tablo 3.1’de görüldüğü gibi önerilen matematiksel modele en yakın Trivella ve Pisinger (2016) tarafından önerilen çok amaçlı matematiksel modeldir.

Bu tez çalışmasında yük dengeli 3BKP çözümü için Trivella ve Pisinger (2016) tarafından önerilen matematiksel modele ek aile birliği kısıtı eklenmiş ve aynı

ürün ailesinden olan nesnelerin aynı kutuya atanmasını sağlayan yeni bir amaç fonksiyonu oluşturulmuştur.

3.2.1 Karar Değişkenleri

Önerilen çok amaçlı tam sayılı matematiksel modelin karar değişkenleri Tablo 3.2’de özetlenmiştir.

Tablo 3.2 Kullanılan karar değişkenler

Sembol	Açıklama
p_{ij}	Eğer i nesnesi j kutusuna paketlenmişse 1, aksi halde 0.
u_j	Eğer j kutusu kullanılıyorsa 1, aksi halde 0.
x_{ik}^+	Eğer aynı kutuda i nesnesi k nesnesinin soluna yerleştirilmişse 1, aksi halde 0.
x_{ik}^-	Eğer aynı kutuda i nesnesi k nesnesinin sağına yerleştirilmişse 1, aksi halde 0.
y_{ik}^+	Eğer aynı kutuda i nesnesi k nesnesinin arkasına yerleştirilmişse 1, aksi halde 0.
y_{ik}^-	Eğer aynı kutuda i nesnesi k nesnesinin önüne yerleştirilmişse 1, aksi halde 0.
z_{ik}^+	Eğer aynı kutuda i nesnesi k nesnesinin altına yerleştirilmişse 1, aksi halde 0.
z_{ik}^-	Eğer aynı kutuda i nesnesi k nesnesinin üzerine yerleştirilmişse 1, aksi halde 0.
l_i^x	Eğer i nesnesinin uzunluğu X eksenine paralel ise 1, aksi halde 0.
l_i^y	Eğer i nesnesinin uzunluğu Y eksenine paralel ise 1, aksi halde 0.
w_i^x	Eğer i nesnesinin genişliği X eksenine paralel ise 1, aksi halde 0.
w_i^y	Eğer i nesnesinin genişliği Y eksenine paralel ise 1, aksi halde 0.
β_{jf}	Eğer f ailesinden en az bir nesne j kutusuna paketlenmişse 1, aksi halde 0.
x_i, y_i, z_i	i nesnesinin yerleştiği sol-alt-arka koordinatları gösteren sürekli değişkenler
g_j^x, g_j^y, g_j^z	Sırasıyla x, y ve z eksenlerinde j kutusunun ağırlık merkezini gösteren sürekli değişkenler

İkili değişkenler, atama değişkenleri (p_{ij}, u_j), konum değişkenleri ($x_{ik}^+, x_{ik}^-, y_{ik}^+, y_{ik}^-, z_{ik}^+, z_{ik}^-$), dönme değişkenleri ($l_i^x, l_i^y, w_i^x, w_i^y$) ve diğer değişkenler (β_{jf}) olarak

gruplandırılmıştır. Sürekli değişkenler ise nesnenin yerleştiği koordinatı (x_i, y_i, z_i) ve kutunun ağırlık merkezini (g_j^x, g_j^y, g_j^z) temsil eder. Konum değişkenleri, $i < k$ olduğunda nesnelerin birbirine göre konumunu belirtir.

3.2.2 Amaç Fonksiyonu

Nesnelerin kutulara aşağıda belirtilen üç hedef doğrultusunda yerleştirilmesi amaçlanmıştır.

1. Minimum kutu kullanımını sağlamak
2. Aynı ürün ailesinden olan nesnelerin aynı kutuya atanmasını sağlamak,
3. Kutularda yük dengesini korumak

Bu nedenle, kullanılan toplam kutu sayısını en aza indiren, aile birliğini en üst düzeye çıkaran ve her bir kutunun ideal ağırlık merkezinden sapmasını en aza indiren üç parçalı amaç fonksiyonu tanımlanmıştır.

Kullanılan ilk amaç, 3BKP'nin çözümüne yönelik yaygın olarak kullanılan toplam kutu sayısını en aza indirmektir. (Denklem 3.1)

$$\text{Minimize} \sum_j^{|J|} u_j \quad (3.1)$$

İkinci amaç ile aile birliğinin korunması hedeflenmiştir ve bu hedef doğrultusunda aynı kutuda aynı aileye ait nesnelerin sayısını maksimize etmek amaçlanmıştır.

$$\text{Minimize} \sum_{f=1}^{|F|} \sum_{j=1}^{|J|} \beta_{jf} \quad (3.2)$$

Bu nedenle, aynı ürün ailesinden olan nesnelerin aynı kutuya atanması Denklem (3.2) ile sağlanmaktadır. Aynı ürün ailesinden olan tüm nesnelerin aynı kutuya paketlenmesi çoğu durumda mümkün olmayabilir. Ancak önerilen matematiksel model, kutulara atanan nesnelerin mümkün olduğunca aynı ürün ailesinden olmasını sağlamaktadır. Olabilecek en iyi senaryo, $\sum_f \beta_{jf} = 1 \ (\forall j \in J)$ ve $\sum_j \beta_{jf} \leq \sum_j u_j \ (\forall f \in F)$ kısıtlarının sağlandığı her kutuda yalnızca aynı ürün ailesinden

nesnelerin bulunmasıdır. En kötü senaryo ise, $\sum_j \sum_f \beta_{jf} = |F| * |J|$ olmak üzere, her kutuda her ürün ailesinden en az bir nesne bulunmasıdır.

Son olarak Denklem (3.3) kullanılarak her bir kutunun ağırlık merkezinin ideal ağırlık merkezinden sapmasını minimize ederek kutularda yük dengesinin sağlanması amaçlanmaktadır.

$$\text{Minimize } \sum_{j=1}^{|J|} \left| g_j^x - \frac{L_j}{2} \right| + \left| g_j^y - \frac{W_j}{2} \right| + |g_j^z - 0| \quad (3.3)$$

Amaç fonksiyonunda yer alan amaçlar için karar vericinin tercihlerine göre ağırlık parametreleri $\theta_1, \theta_2, \theta_3 \geq 0$ kullanılarak ağırlıklı amaç fonksiyonu elde edilmiştir. Amaçların birimleri birbirinden farklı olduğu için, amaç fonksiyonlarını normalize etmek gerekmektedir. Bu sebeple, amaç fonksiyonu tipik bir orantı temelli normalleştirme prosedürü kullanılarak normalize edilmiştir. Amaç fonksiyonlarının pareto optimal alan içerisinde normalize edilmesi için, nadir ve ideal amaç vektörleri kullanılır (Deb, 2001). Her bir amaç fonksiyonu pareto optimal alan içerisinde nadir ve ütöpik değerler kullanılarak $[0,1]$ aralığında normalize edilir. o_i^U ütöpik değeri ve o_i^N nadir değeri ifade etmektedir. $f_i(x)$ olarak tanımlanan amaç fonksiyonunun normalize edilmiş hali $\frac{f_i(x) - o_i^U}{o_i^N - o_i^U}$ olarak tanımlanmıştır.

3.2.3 Kısıtlar

Önerilen çok amaçlı tam sayılı matematiksel modelin kısıtları tezin bu bölümünde ayrıntılı anlatılmıştır.

Kısıt (3.4), bir i nesnesinin yalnızca j kutusunun kullanıldığı durumda, j kutusuna atanabilmesini sağlar.

$$p_{ij} \leq u_j \quad \forall i \in I, \forall j \in J \quad (3.4)$$

Kısıt (3.5), her i nesnesinin paketlenmesini sağlar.

$$\sum_{j=1}^{|J|} p_{ij} = 1 \quad \forall i \in I \quad (3.5)$$

Kısıt (3.6)-(3.7), M 'nin büyük bir skaler olduğu durumlarda, nesnelerin kutunun uzunluğunu ve genişliğini aşmadan kutuya yerleştirilmesini sağlar. Problemde nesneler yerleştirilirken 2-yönlü döndürülmesine izin verildiği için kutu yüksekliğinin aşılmamasını sağlayan kısıtta (3.8) z eksenindeki dönme dikkate alınmamıştır.

$$x_i + l_i l_i^x + w_i w_i^x \leq L_j u_j + (1 - p_{ij})M \quad \forall i \in I, \forall j \in J \quad (3.6)$$

$$y_i + l_i l_i^y + w_i w_i^y \leq W_j u_j + (1 - p_{ij})M \quad \forall i \in I, \forall j \in J \quad (3.7)$$

$$z_i + h_i \leq H_j u_j + (1 - p_{ij})M \quad \forall i \in I, \forall j \in J \quad (3.8)$$

Kısıt (3.9)-(3.14), i ve k nesneleri aynı kutuda paketleniyse, birbirleri ile çakışmasını önler.

$$x_i + l_i l_i^x + w_i w_i^x \leq x_k + (1 - x_{ik}^+)M \quad \forall i, k \in I, i < k \quad (3.9)$$

$$x_k + l_k l_k^x + w_k w_k^x \leq x_i + (1 - x_{ik}^-)M \quad \forall i, k \in I, i < k \quad (3.10)$$

$$y_i + l_i l_i^y + w_i w_i^y \leq y_k + (1 - y_{ik}^+)M \quad \forall i, k \in I, i < k \quad (3.11)$$

$$y_k + l_k l_k^y + w_k w_k^y \leq y_i + (1 - y_{ik}^-)M \quad \forall i, k \in I, i < k \quad (3.12)$$

$$z_i + h_i \leq z_k + (1 - z_{ik}^+)M \quad \forall i, k \in I, i < k \quad (3.13)$$

$$z_k + h_k \leq z_i + (1 - z_{ik}^-)M \quad \forall i, k \in I, i < k \quad (3.14)$$

Kısıt (3.15) i ve k nesnelerinin aynı kutuda olması durumunda, birbirinin soluna, sağına, üstüne, altına, arkasına veya önüne yerleşmesini sağlar.

$$x_{ik}^+ + x_{ik}^- + y_{ik}^+ + y_{ik}^- + z_{ik}^+ + z_{ik}^- \geq p_{ij} + p_{kj} - 1 \quad \forall i, k \in I, i < k \quad (3.15)$$

Kısıt (3.16) - (3.19) her bir nesnenin tabanının, uygun pozisyonda yerleşmesini sağlar.

$$l_i^x + l_i^y = 1 \quad \forall i \in I \quad (3.16)$$

$$l_i^x + w_i^x = 1 \quad \forall i \in I \quad (3.17)$$

$$w_i^x + w_i^y = 1 \quad \forall i \in I \quad (3.18)$$

$$l_i^y + w_i^y = 1 \quad \forall i \in I \quad (3.19)$$

Kısıt (3.20), j kutusundaki nesnelerin toplam ağırlığının, kutunun kapasitesini aşmamasını sağlar.

$$\sum_{i=1}^{|I|} a_i p_{ij} \leq A_j u_j \quad \forall j \in J \quad (3.20)$$

Kısıt (3.21)-(3.23), her j kutusunun, x eksen, y eksen ve z eksenindeki ağırlık merkezini hesaplar.

$$g_j^x \sum_{i=1}^{|I|} a_i p_{ij} = \sum_{i=1}^{|I|} a_i p_{ij} \left(x_i + \frac{1}{2} l_i^x + \frac{1}{2} w_i^x \right) \quad \forall j \in J \quad (3.21)$$

$$g_j^y \sum_{i=1}^{|I|} a_i p_{ij} = \sum_{i=1}^{|I|} a_i p_{ij} \left(y_i + \frac{1}{2} l_i^y + \frac{1}{2} w_i^y \right) \quad \forall j \in J \quad (3.22)$$

$$g_j^z \sum_{i=1}^{|I|} a_i p_{ij} = \sum_{i=1}^{|I|} a_i p_{ij} \left(z_i + \frac{1}{2} h_i \right) \quad \forall j \in J \quad (3.23)$$

β_{jf} j kutusu içerisindeki f ürün ailesinin varlığını temsil etmektedir. Bu sebeple, kısıt (3.24) ürün ailesi ve kullanılan j kutusu arasında bir bağlantı kurulmasını sağlar. Kısıt (3.25) j kutusunda f ailesinden en az bir nesnenin bulunup bulunmadığını belirler. Kısıt (3.24)-(3.25) aynı aileden olan nesneler aynı kutuya tahsis edilmesini sağlar.

$$\beta_{jf} \leq u_j \quad \forall j \in J, \forall f \in F \quad (3.24)$$

$$\sum_{i=1}^{|I|} \alpha_{if} p_{ij} \leq \beta_{jf} M \quad \forall j \in J, \forall f \in F \quad (3.25)$$

Son olarak, kısıt (3.26)-(3.27)'de ikili ve negatif olmayan değişkenler belirtilmiştir.

$$p_{ij}, u_j, x_{ik}^+, x_{ik}^-, y_{ik}^+, y_{ik}^-, z_{ik}^+, z_{ik}^-, l_i^x, l_i^y, w_i^x, w_i^y, \beta_{jf} \in \{0,1\} \quad (3.26)$$

$$x_i, y_i, z_i, g_j^x, g_j^y, g_j^z \geq 0 \quad (3.27)$$

3.2.4 Doğrusal Olmayan Denklemlerin Doğrusal Hale Getirilmesi

Bu bölümde, önerilen matematiksel modele yeni değişkenler ve ek kısıtlar dâhil edilerek modelin doğrusal hale getirilmesi sağlanmıştır.

Kutularda yük dengesinin sağlandığı amaç fonksiyonu (3.3) mutlak değerli ifadelerden oluştuğu için doğrusal değildir. Amaç fonksiyonunu doğrusal hale getirebilmek için her bir mutlak değer içerisindeki pozitif ve negatif kısmı temsil eden ikişer tane negatif olmayan değişken tanımlanmıştır (3.29)-(3.31). Sonrasında, mutlak değer yeni değişkenlerin toplamı ile değiştirilmiştir. Kutularda ki yük dengesinin sağlandığı doğrusal amaç fonksiyonu için Denklem (3.28) elde edilmiştir.

$$\text{Minimize} \sum_j^{|I|} (e_j^{x+} + e_j^{x-} + e_j^{y+} + e_j^{y-} + e_j^{z+} + e_j^{z-}) \quad (3.28)$$

$$\left| g_j^x - \frac{L_j}{2} \right| = e_j^{x+} - e_j^{x-}, \quad e_j^{x+} \geq 0, e_j^{x-} \geq 0, \quad (3.29)$$

$$\left| g_j^y - \frac{W_j}{2} \right| = e_j^{y+} - e_j^{y-}, \quad e_j^{y+} \geq 0, e_j^{y-} \geq 0, \quad (3.30)$$

$$|g_j^z| = e_j^{z+} - e_j^{z-}, \quad e_j^{z+} \geq 0, e_j^{z-} \geq 0. \quad (3.31)$$

Sonuç olarak, amaç fonksiyonu doğrusal hali Denklem (3.32)' da gösterildiği gibidir.

$$\text{Minimize } \theta_1 \sum_j^{|J|} u_j + \theta_2 \sum_{f=1}^{|F|} \sum_{j=1}^{|J|} \beta_{jf} + \theta_3 \sum_{j=1}^{|J|} (e_j^{x+} + e_j^{x-} + e_j^{y+} + e_j^{y-} + e_j^{z+} + e_j^{z-}) \quad (3.32)$$

Matematiksel modelde, her bir j kutusunun ağırlık merkezini hesaplayan (3.21)-(3.23) kısıtlarda doğrusal olmayan denklemlerdir. Denklem (3.21)'de ki terimler Denklem (3.33)'de gösterildiği gibi yeniden düzenlenmiştir.

$$\sum_i a_i p_{ij} g_j^x - \sum_i a_i p_{ij} x_i - \sum_i a_i p_{ij} \frac{1}{2} l_i l_i^x - \sum_i a_i p_{ij} \frac{1}{2} w_i w_i^x = 0 \quad (3.33)$$

Denklem (3.33) iki tane ikili değişkenin çarpılması ve bir ikili değişken ile bir sürekli değişkenin çarpılmasından dolayı doğrusal değildir.

Bir ikili değişken ile bir sürekli değişkenin çarpımından dolayı doğrusal olmayan kısıtları doğrusal hale getirebilmek için, $d_{ij} = p_{ij} g_j^x$ olacak şekilde yeni bir pozitif değişken (d_{ij}) tanımlanmış ve (3.34)-(3.36) denklemlerinde verilen yeni kısıtlar modele eklenmiştir.

$$d_{ij} \geq g_j^x - M(1 - p_{ij}) \quad (3.34)$$

$$d_{ij} \leq M p_{ij} \quad (3.35)$$

$$d_{ij} \leq g_j^x \quad (3.36)$$

Aynı yöntem kullanılarak, $e_{ij} = p_{ij} x_i$ olacak şekilde yeni bir negatif olmayan değişken (e_{ij}) tanımlanmıştır. Denklem (3.37)-(3.39)'da belirtilen eşitsizlikler yeni kısıt olarak matematiksel modele dahil edilmiştir.

$$e_{ij} \geq x_i - M(1 - p_{ij}) \quad (3.37)$$

$$e_{ij} \leq M p_{ij} \quad (3.38)$$

$$e_{ij} \leq x_i \quad (3.39)$$

Önerilen matematiksel modelin doğrusal bir model olmasını engelleyen bir diğer ifade ise, $p_{ij} l_i^x$ gibi iki tane ikili değişkenin birbiri ile çarpımıdır. Bu durumda

modeli doğrusal hale getirebilmek için $b_{ij} = p_{ij}l_i^x$ olacak şekilde yeni bir ikili değişken (b_{ij}) tanımlanmış ve modele kısıt (3.40)-(3.42) eklenmiştir.

$$b_{ij} \leq l_i^x \quad (3.40)$$

$$b_{ij} \leq p_{ij} \quad (3.41)$$

$$b_{ij} \geq l_i^x + p_{ij} - 1 \quad (3.42)$$

Benzer şekilde, $(p_{ij}w_i^x)$ çarpımını doğrusal hale getirmek için aynı yöntem kullanılmıştır. $(p_{ij}w_i^x)$ ifadesine karşılık gelen negatif olmayan c_{ij} değişkeni tanımlanmıştır. Sonrasında matematiksel modele kısıt (3.43)-(3.45) dâhil edilmiştir.

$$c_{ij} \leq w_i^x \quad (3.43)$$

$$c_{ij} \leq p_{ij} \quad (3.44)$$

$$c_{ij} \geq w_i^x + p_{ij} - 1 \quad (3.45)$$

Sonuç olarak, önerilen matematiksel modelde denklem (3.21) denklem (3.46) ile değiştirilmiş ve (3.34)-(3.46) denklemlerindeki kısıtlar matematiksel modele eklenmiştir.

$$\sum_i a_i d_{ij} - \sum_i a_i e_{ij} - \sum_i \frac{1}{2} a_i l_i b_{ij} - \sum_i \frac{1}{2} a_i w_i c_{ij} = 0 \quad (3.46)$$

Benzer şekilde, Denklem (3.22)' de ifade edilen kısıt doğrusal hale getirebilmek için aynı yöntem kullanılmıştır. $t_{ij} = p_{ij}g_j^y$, $v_{ij} = p_{ij}y_i$, $r_{ij} = p_{ij}l_i^y$, $s_{ij} = p_{ij}w_i^y$ olacak şekilde negatif olmayan dört tane yeni değişken tanımlanmış ve denklem (3.22) doğrusal hale getirilerek kısıt (3.47)-(3.59) matematiksel modele dâhil edilmiştir.

$$\sum_i a_i t_{ij} - \sum_i a_i v_{ij} - \sum_i \frac{1}{2} a_i l_i r_{ij} - \sum_i \frac{1}{2} a_i w_i s_{ij} = 0 \quad (3.47)$$

$$t_{ij} \geq g_j^y - M(1 - p_{ij}) \quad (3.48)$$

$$t_{ij} \leq Mp_{ij} \quad (3.49)$$

$$t_{ij} \leq g_j^y \quad (3.50)$$

$$v_{ij} \geq y_i - M(1 - p_{ij}) \quad (3.51)$$

$$v_{ij} \leq Mp_{ij} \quad (3.52)$$

$$v_{ij} \leq y_i \quad (3.53)$$

$$r_{ij} \leq l_i^y \quad (3.54)$$

$$r_{ij} \leq p_{ij} \quad (3.55)$$

$$r_{ij} \geq l_i^y + p_{ij} - 1 \quad (3.56)$$

$$s_{ij} \leq w_i^y \quad (3.57)$$

$$s_{ij} \leq p_{ij} \quad (3.58)$$

$$s_{ij} \geq w_i^y + p_{ij} - 1 \quad (3.59)$$

Son olarak, $m_{ij} = p_{ij} g_j^z$ ve $n_{ij} = p_{ij} z_i$ olmak üzere iki yeni değişken tanımlanmış ve kısıt (3.60)-(3.66) matematiksel modele eklenerek kısıt (3.23) doğrusal hale getirilmiştir.

$$\sum_i a_i m_{ij} - \sum_i a_i n_{ij} - \sum_i \frac{1}{2} a_i h_i p_{ij} = 0 \quad (3.60)$$

$$m_{ij} \geq g_j^z - M(1 - p_{ij}) \quad (3.61)$$

$$m_{ij} \leq Mp_{ij} \quad (3.62)$$

$$m_{ij} \leq g_j^z \quad (3.63)$$

$$n_{ij} \geq z_i - M(1 - p_{ij}) \quad (3.64)$$

$$n_{ij} \leq Mp_{ij} \quad (3.65)$$

$$n_{ij} \leq z_i \quad (3.66)$$

Son olarak oluşturulan çok amaçlı tam sayılı matematiksel model, (3.4)-(3.20), (3.24)-(3.27) ve (3.34)-(3.66) kısıtları dikkate alarak Denklem (3.32)'deki fonksiyonları minimize etmeyi amaçlamaktadır.

3.3 Önerilen Genetik Algoritma Tabanlı Sezgisel Algoritma

Bu bölümde, aile birliği kavramının göz önünde bulundurulduğu yük dengeli 3BKP çözümü için genetik algoritma tabanlı sezgisel algoritma sunulmuştur. Önerilen sezgisel algoritma, kullanılan kutu sayısını ve ideal ağırlık merkezinden sapmayı minimize ederken, aynı ürün ailesinden olan nesnelerin mümkün olduğunca birlikte paketlenmesini sağlar.

Önerilen algoritma üç aşamadan oluşmaktadır. İlk aşamada problem bir boyutlu olarak ele alınmış ve nesneler kutu kapasitesini aşmayacak şekilde ürün ailelerine göre kutulara atanmıştır. Sonrasında birinci aşamadan elde edilen çözümler başlangıç çözüm olarak kabul edilmiştir. Bu aşamada aile birliği mümkün olduğunca korunmuş ve problem üç boyutlu ele alınarak nesneler tekrar yerleştirilmiştir. Son aşamada ise her bir kutunun dengeli paketlenmesi amaçlanmış ve nihai yerleşim planları hazırlanmıştır.

Önerilen algoritmanın aşamaları aşağıda ayrıntılı olarak açıklanmıştır.

3.3.1 Birinci Aşama: Genetik Algoritmaya Dayalı Sezgisel Algoritma

Genetik algoritmalar, 1970li yıllarda Holland tarafından literatüre sunulan evrimsel algoritmalarlardır. Genetik algoritmanın kullanımı 1989 yılında “Arama Optimizasyonu ve Makine Öğrenmesinde Genetik Algoritmalar” kitabının yayınlanması ile birlikte artmıştır. (Ayhan,2021) Günümüzde, genetik algoritmanın verimliliğini artırmak için çeşitli yaklaşımlar benimsenmiş ve bunlar çok sayıda optimizasyon problemine uygulanmıştır.

Genetik algoritmalar popülasyon tabanlı optimizasyon yöntemidir. İlk olarak birden fazla kromozomdan oluşan popülasyon havuzu oluşturulur. Kromozomun, yani çözümün kalitesi, algoritmayı daha iyi optimize edilmiş çözümlere

yönlendiren uygunluk fonksiyonu ile ölçülür. Uygunluk fonksiyonu değeri ne kadar yüksek olursa, çözüm kalitesi o kadar iyi olur. Oluşturulan popülasyondaki çözümleri iyileştirmek için çaprazlama ve mutasyon operasyonları uygulanır ve yeni popülasyon oluşturulur. Bu işlem en iyi çözüm bulunana kadar devam ettirilir.

Genetik algoritmanın avantaj olarak tanımlayabileceğimiz klasik algoritmalarından farklı özellikleri vardır: (Ayhan,2021)

- Genetik Algoritma amaç fonksiyonunu doğrudan kullanır. Ek bilgilere ihtiyaç duymaz.
- Çözüm uzayının bütününe taramak yerine belirli bir kısmını taradığı için çözüme kısa sürede ulaşır.
- Algoritmada deterministik kurallar yerine stokastik kurallara yer verilir.
- Genetik algoritma tek bir noktada arama yapmadığından dolayı yerel optimuma takılmadan çalışabilir.

Bu tez çalışmasında, ürün aile birliği korunarak nesnelerin kutulara atanması hedeflenmiştir. Önerilen sezgisel algoritma oluşturulurken, Falkenauer (1996)'ın yapmış olduğu çalışma temel alınmıştır. Bu kapsamda, nesneler ve kutular sadece hacimlerine göre tanımlanmış ve problem genetik algoritma tabanlı bir sezgisel yöntem kullanılarak bir boyutlu kutulama problemi olarak çözülmüştür.

Önerilen genetik algoritma ile Falkenauer (1996) yapmış olduğu çalışma arasında iki temel fark vardır:

- Kullanılan uygunluk fonksiyonu
- Çaprazlama aşamasında çaprazlama noktalarının belirlenmesi ve çaprazlama aşamasında nesnelerin tekrar yerleştirilmesi için kullanılan yöntem

Bu temel farklar tezin ilerleyen kısmında ayrıntılı olarak açıklanmıştır.

Genetik Algoritmaya Dayalı Sezgisel Algoritmanın temel aşamaları aşağıda Tablo 3.3'de gösterilmiştir.

Tablo 3.3 Genetik algoritma temel aşamaları

Aşama I: Genetik Algoritma Temel Aşamaları
1. Nesne listesi kullanılarak başlangıç popülasyonu rassal olarak oluşturulur 2. Her bir kromozomun uygunluk değeri hesaplanır 3. Rulet tekerleği yöntemi kullanılarak ebeveyn seçimi yapılır 4. Her ebeveyndeki en iyi gen (kutu) bulunur ve ebeveynler arasında çaprazlanır. 5. Çaprazlanan gen içerisinde bulunan nesneleri tekrar eden genler kromozomlardan silinir. 6. Silinen genlerin içerisinde olan nesneler ürün ailesine dikkat edilerek yeniden yerleştirilir. 7. Mutasyon: Mutasyon parametrelerine göre genler rassal olarak seçilir ve 5 ve 6. Adımlar tekrarlanır. 8. En iyi çözüm alınır ve Aşama II' ye gönderilir.

3.3.1.1 Kromozom Yapısı ve Başlangıç Popülasyonu

Genetik algoritmada her bir kromozom olası bir çözümü ifade ettiği için kromozomların probleme uygun şekilde kodlanması büyük önem taşımaktadır. Genetik algoritmalar ile kutulama problemleri çözülürken literatürde önerilen üç temel kromozom yapısı bulunmaktadır. Bunlardan ilki nesne tabanlı kromozom yapısıdır. Burada her bir gen bir nesneyi temsil eder. Örnek bir kromozom gösterimi |1 2| |3 4| şeklindedir. Bu kromozom yapısında, birinci ve ikinci nesneler ilk kutuya, üçüncü ve dördüncü nesneler ise ikinci kutuya paketlenmiştir. Kromozom yapılarından ikincisi ise, kutu tabanlı kromozom yapısıdır. Burada her bir gen bir kutuyu temsil eder. Her kromozom nesne sayısı ile eşit uzunluktadır. Örneğin, paketlenen nesne sayısı beş ise, örnek bir kromozom |2 3 1 1 2| şeklinde yazılabilir. Burada, birinci ve beşinci nesneler ikinci kutuya, ikinci nesne üçüncü kutuya ve üç ve dördüncü nesneler ise birinci kutuya paketlenmiştir. Kromozom yapılarından sonuncusu ise Falkenauer (1996) tarafından önerilen grup tabanlı kromozom yapısıdır. Grup tabanlı kromozom yapısı iki kısımdan oluşur. İlk kısım, kutu tabanlı kromozom yapısı ile aynıdır. İkinci kısım ise, kutu gruplarını ele alır. Kromozomun ilk kısmı sadece grup

üyelerini tanımlamak için kullanılır. Tüm genetik operasyonlar kutu gruplarının tanımlandığı ikinci kısım aracılığı ile yapılır (Mohamadi,2010).

Kutulama problemlerinin çözümü için genetik algoritmanın uygulandığı çalışmalarda kullanılan kromozom yapıları Tablo 3.4 de özetlenmiştir.

Tablo 3.4 Kromozom yapıları

Yayın	Kutu				Gen Yapısı
	Tek Kutulu	Çok Kutulu	Nesne Tabanlı	Kutu/Grup Tabanlı	
Jakobs (1996)	✓		✓		Paketleme Sırası
Liu ve Teng (1997)	✓		✓		Paketleme Sırası
Gehring ve Bortfeldt (1997)	✓		✓		Paketleme Sırası, Nesne yerleşme yönü
Wu vd., (2010)	✓		✓		Paketleme Sırası, Nesne yerleşme yönü
Kang vd., (2012)	✓		✓		Paketleme Sırası
Gançalves ve Resende (2013)	✓		✓		Paketleme Sırası, Nesne yerleşme yönü
Moon ve Nguyen (2014)	✓		✓		Paketleme Sırası, Nesne yerleşme yönü
Dokeroglu ve Cosar (2014)		✓	✓		Kutu
Gonzalez vd., (2016)	✓		✓		Nesne Çeşiti, Nesne sayısı, Nesne yerleşme yönü
Jamrus ve Chien (2016)	✓		✓		Paketleme Sırası
Kabasakal ve Keskin (2017)		✓		✓	Kutu

Tablo 3.4 Kromozom yapıları (devamı)

Ramos vd., (2018)		✓		✓	Paketleme Sırası, Nesne yerleşme yönü
Fan vd., (2020)		✓		✓	Kutu

Yapılan çalışmalar incelendiğinde tek kutulu problemlerde kullanılan kromozom yapısının nesne tabanlı formda olduğu gözlemlenmiştir. Birden fazla kutunun yüklendiği durumlarda ise grup tabanlı kromozom yapısının kullanıldığı görülmüştür.

Ayrıca Falkenauer (1996) yapmış olduğu çalışmada çoklu kutulama problemleri için grup tabanlı kromozom yapısının, nesne tabanlı kromozom yapısından daha iyi sonuç verdiğini kanıtlamıştır. Bu sebeple, bu çalışmada ele alınan probleme en uygun kromozom yapısı olarak grup tabanlı kromozom yapısı seçilmiştir. Şekil 3.2’de üç kutulu örnek bir kromozom verilmiştir. Her bir gen bir kutuyu, her bir kutu ise nesnelerin özelliklerini içerir. 4, 2, 7 ve 1 numaralı nesneler birinci kutuda, 8, 10 ve 9 numaralı nesneler ikinci kutuda ve 5, 6 ve 3 numaralı nesneler üçüncü kutuda yer almaktadır.

Kutu	1	1	3	1	3	3	1	2	2	2	:	1	2	3
Nesne	1	2	3	4	5	6	7	8	9	10	Kutu grupları			

Şekil 3.2 Kromozom yapısının gösterimi (Fan vd., 2020)

Her bir gen Şekil 3.3’de olduğu gibi nesne numarası, hacim ve ürün ailesi bilgilerini içinde barındırır.

$$A: \begin{pmatrix} \begin{bmatrix} 4 & 25 & 1 \\ 2 & 20 & 1 \\ 7 & 25 & 3 \\ 1 & 10 & 1 \\ \textbf{GEN1} \\ (\textbf{KUTU 1}) \end{bmatrix} & \begin{bmatrix} 8 & 10 & 4 \\ 10 & 40 & 3 \\ 9 & 20 & 4 \\ & & \\ \textbf{GEN2} \\ (\textbf{KUTU2}) \end{bmatrix} & \begin{bmatrix} 5 & 10 & 2 \\ 6 & 30 & 2 \\ 320 & 2 \\ & & \\ \textbf{GEN3} \\ (\textbf{KUTU3}) \end{bmatrix} \end{pmatrix}$$

Şekil 3.3 Kromozom içeriği

Kromozom yapısı belirlendikten sonra, olası çözümlerden oluşan başlangıç popülasyonu rastgele oluşturulmuş ve genetik algoritma adımları uygulanmıştır.

3.3.1.2 Uygunluk Fonksiyonu

Genetik algorithmada problemin amaç fonksiyonu, uygunluk fonksiyonu olarak kullanılmaktadır. Kromozomların bir sonraki nesile aktarılma olasılığını bu değer belirlemektedir.

Bu çalışmada kullanılan uygunluk fonksiyonu $f(x)$ iki kısımdan oluşmaktadır. Birinci kısım maksimum hacim kullanımı sağlayabilmek için doluluk oranını (FR) ele alınırken, ikinci kısım aile birliği oranını (FUR) ele almaktadır. Sonuç olarak bu iki amaç ağırlıklı toplanarak uygunluk fonksiyonu oluşturulmuştur.

Uygunluk fonksiyonunun birinci kısmında kullanılan doluluk oranı literatürde sıklıkla kullanılmıştır. Ancak ikinci kısımda aile birliğinin hesaba katılmasıyla literatürde olmayan yeni bir uygunluk fonksiyonu elde edilmiştir. Uygunluk fonksiyonu Denklem (3.67) kullanılarak hesaplanmaktadır.

$$f(x) = w_1FR + w_2FUR \quad (3.67)$$

Uygunluk fonksiyonunda yer alan amaçlar için karar vericinin tercihlerine göre ağırlık parametreleri $w_1, w_2 \geq 0$ kullanılarak ağırlıklı uygunluk fonksiyonu elde edilmiştir.

FR denklem (3.68) kullanılarak hesaplanmaktadır.

$$FR = \sum_{j=1}^m \frac{v_{ij}}{V_j} \quad (3.68)$$

v_{ij} , j kutusundaki i nesnesinin hacmini temsil ederken, V_j , j kutusunun hacmini (kapasitesi) temsil etmektedir. (Falkenauer, 1996)

Nesnelerin ürün ailesine göre kutulara yerleştirilmesini sağlayan ürün ailesi oranı denklem (3.69) yardımı ile hesaplanmıştır.

$$FUR = \frac{Nb_{f_beklenen}}{Nb_{f_gerçekleşen}} \quad (3.69)$$

$$Nb_{f_beklenen} = \left\lceil \frac{\sum_f v_{if}}{V_j} \right\rceil \quad (3.70)$$

$Nb_{f_beklenen}$, her aile için gerekli olan minimum kutu sayısı toplamı, $Nb_{f_gerçekleşen}$ her aile için kullanılan kutu sayısı toplamı ve v_{if} f ailesindeki i nesnesinin hacmi olarak tanımlanmıştır.

Üç kutudan oluşan örnek kromozom A'yı tekrar ele alacak olursak, örnek çözüm için Denklem (3.70) kullanılarak, $Nb_{f_beklenen}$ değeri sırasıyla [0,69], [0,75], [0,81] ve [0,38] olarak hesaplanmıştır. Kutu hacminin 80 olduğu varsayılmıştır. Sonuç olarak toplamda 4 kutuya ihtiyaç duyulduğu görülmüştür. Ancak örnek kromozomda kutular incelendiğinde, birinci, ikinci, üçüncü ve dördüncü aileye ait nesnelerin sırasıyla bir, bir, iki ve bir kutuya yerleştirildiği ve sonuç olarak $Nb_{f_gerçekleşen}$ değerinin beş olduğu görülmektedir. Elde edilen bu verilere FUR değeri 0,8 olarak hesaplanmıştır.

3.3.1.3 Seçme

Her bir kromozomun uygunluk değeri hesaplandıktan sonra yeni popülasyon (nesil) oluşturmak için mevcut çözümlerden ebeveynlerin seçilmesi gerekmektedir.

Algoritmada, uygunluk fonksiyonu hesaplanan kromozomlar için rulet tekerleği yöntemi kullanılmış ve ebeveyn kromozomlarının seçimi yapılmıştır. Rulet tekerleği yöntemi olasılık temelli bir seçme yöntemi olarak bilinmektedir. Bu yöntemde, her bir kromozomun seçilme olasılığı uygunluk fonksiyonu ile orantılıdır. Buna rağmen en yüksek uygunluk fonksiyonuna sahip kromozomun bir sonraki nesile aktarılmasını garanti etmez. Ancak yüksek orana sahip kromozomların seçilme şansının yüksek olacağı söylenebilir.

3.3.1.4 Çaprazlama ve Mutasyon

Çaprazlama, yüksek uygunluk fonksiyonuna sahip ebeveyn kromozomlardan daha iyi çocuk kromozomları üretmek için yeniden konumlandırma olarak tanımlanır.

Bu tez çalışmasında, uygunluk fonksiyonu yüksek bireyler için aile birliğini korumak amacı ile yeni bir çaprazlama yöntemi önerilmiştir. Tablo 3.5'de 11 tane açıklayıcı senaryo oluşturulmuştur ($n=9$). Her bir senaryo bir kutuyu temsil etmektedir. Örneğin senaryo 1'de bir kutuya dört tane nesne yerleştirilmiştir. Bu nesnelerden, birinci ve ikinci nesneler birinci ürün ailesine, üçüncü ve dördüncü

nesneler ise üçüncü ürün ailesine aittir. Önerilen yöntemde her bir gen için seçim değeri (3.71) eşitliği ile hesaplanmış ve en düşük değere sahip olan gen seçilerek çaprazlama işlemi başlatılmıştır. Böylece aile birliğini artıran genler diğer nesle aktarılmıştır.

$$\text{Seçim Değeri} = (1 - FR) + (f_{ij} + 1) * (n + 1) + (f_{ij} + 1) * (n - m_{ij}) + F_{HI} \quad (3.71)$$

Burada f_{ij} , m_{ij} ve F_{HI} , sırasıyla j kutusundaki farklı ailelerin sayısı, j kutusuna yerleştirilen nesne sayısı ve j kutusundaki her aile için nesnelerin sayısının çarpımı olarak tanımlanır. Önerilen formül (3.71) dört kısımdan oluşmaktadır. Birinci kısım aracılığı ile $(1 - FR)$ kutu içerisinde oluşan toplam boş alan hesaplanır. Kutu doluluk oranının yüksek olması bu genin en iyi gen olduğu anlamına gelmez. Çünkü önerilen sezgisel algoritmada ürün aile birliğinin oluşturulması amaçlanmaktadır. Bu sebeple formüle ikinci kısım $((f_{ij} + 1) * (n + 1))$ eklenmiştir. Formülün bu kısmı ürün aile çeşitliliğinin az olduğu genin seçilme olasılığını arttırmak için formüle dâhil edilmiştir. Yani kutudaki farklı ürün sayısı ne kadar düşükse bu değer de o kadar düşük olacaktır. Böylelikle ilgili genin seçilme olasılığı artacaktır. Kutu içerisinde bulunan ürün ailesi sayısının ve kutu içerisinde oluşan boş alanın az olduğu kutuların bulunduğu çözümlerin iyi çözüm olduğu söylenilebilir. Ancak senaryo 2 ve 4' de kutuların iki aileden oluştuğu ve her iki senaryoda da kutunun %30'unun boş olduğu görülmektedir. Hem ürün ailesi açısından hem de kutu içerisinde oluşan boş alan açısından bir eşitlik söz konusudur. Görüldüğü gibi, sadece iki kritere bakılarak en iyi gen seçimine karar vermek güçtür. Bu sebeple, ürün ailesi çeşitliliğinin az olduğu ve nesne sayısının fazla olduğu çözümlerin seçim olasılığını arttırmak için formüle üçüncü kısım $((f_{ij} + 1) * (n - m_{ij}))$ dâhil edilmiştir. Ancak, senaryo 10 ve 11 için değerlendirme kriterlerin hepsinde eşitlik söz konusudur. Bu gibi durumlarda en iyi gen seçimine karar verebilmek için, formüle son olarak aile heterojenliği indeksi (F_{HI}) kısmı eklenmiştir. Kutu içerisinde olan ürün ailesinin sayısı kadar o aileden kaç nesnenin paketlenildiği de önemlidir. Burada amaç, aynı ürün ailesinden olan nesnelerin sayıca çok olduğu çözümün öne çıkmasını sağlamaktır. Senaryo 10-11 incelendiğinde, senaryo 10'da, birinci ürün ailesinden beş tane nesne, üçüncü ürün ailesinden bir tane nesne vardır. F_{HI} değeri, j kutusundaki her aile için nesnelerin

sayısının çarpımı olarak tanımlandığından, beştir. Senaryo 11’de ise birinci ürün ailesinden üç tane nesne, üçüncü ürün ailesinden üç tane nesne vardır ve F_{HI} değeri dokuz olarak hesaplanmıştır. Bu sebeple, senaryo 10, senaryo 11’e göre daha iyi olduğu söylenebilir. Tüm kriterler göz önünde bulundurularak, Tablo 3.5’de sunulan senaryolar karşılaştırıldığında en iyi senaryonun 6 olduğu görülmüştür. Bu senaryoda, kutu tek bir aileden oluşmaktadır. Örneğin, senaryo 7 ile kıyaslandığında, her iki senaryoda ürün ailesi sayısı ve kutu içerisindeki nesne sayıları aynıdır. Ancak kutu içerisinde oluşan boş alan Senaryo 6’da daha az olduğu için toplam seçim değeri daha düşüktür. Bu sebeple Senaryo 6 daha iyidir. Senaryo 6 ile Senaryo 5 kıyaslandığında ise her iki senaryoda ürün aile sayısı ve kutu içerisinde kalan boş alan aynıdır. Ancak kutu içerisine yerleşen nesne sayısı farklıdır. Bu sebeple, nesne ve ürün çeşitliliği arasındaki bağlantıyı ilişkilendiren formülün üç ve dördüncü kısımlarında çözüm birbirinden ayrılmış ve senaryo 6’nın daha iyi olduğu görülmüştür.

Tablo 3.5 Seçim oranı hesabı senaryoları

Senaryo	Aile	f_{ij}	$(1 - FR)$	$(f_{ij} + 1) * (n + 1)$	$(f_{ij} + 1) * (n - m_{ij})$	F_{HI}	Toplam
1	1133	2	0.5	30	15	4	49.5
2	1133	2	0.3	30	15	4	49.3
3	113	2	0.5	30	18	2	50.5
4	113	2	0.3	30	18	2	50.3
5	2	1	0.4	20	16	1	37.4
6	222	1	0.4	20	12	3	35.4
7	222	1	0.6	20	12	3	35.6
8	1234	4	0.1	50	25	1	76.1
9	12344	4	0.6	50	20	2	72.6
10	111113	2	0.1	30	9	5	44.1
11	111333	2	0.1	30	9	9	48.1

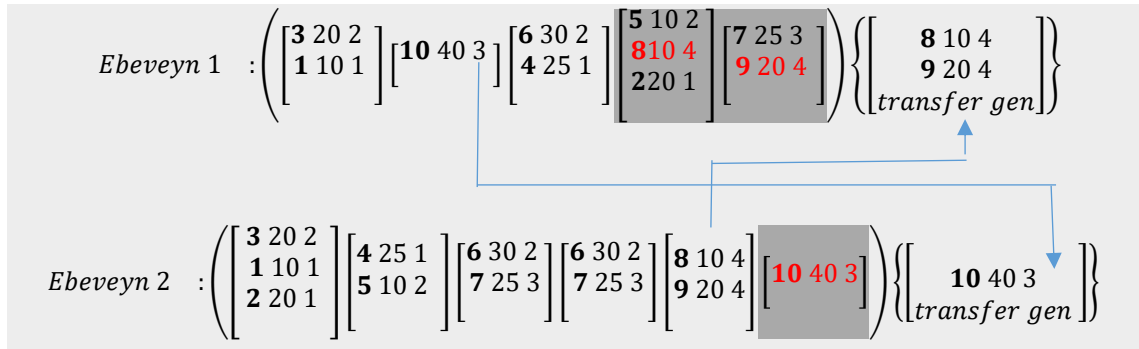
Böylelikle oluşturulan seçim değeri formülü ile her bir gen farklı birçok açıdan değerlendirmeye alınarak, ürün aile birliğinin yüksek olduğu genin seçilmesini sağlamaktadır. Çaprazlama aşamasının sonraki adımları için seçilen örnek ebeveyn bireyler Şekil 3.4’de gösterilmiştir.

$$Ebeveyn 1 : \left(\left(\begin{bmatrix} 3 & 20 & 2 \\ 1 & 10 & 1 \end{bmatrix} \begin{bmatrix} 10 & 40 & 3 \end{bmatrix} \begin{bmatrix} 6 & 30 & 2 \\ 4 & 25 & 1 \end{bmatrix} \begin{bmatrix} 5 & 10 & 2 \\ 8 & 10 & 4 \\ 2 & 20 & 1 \end{bmatrix} \begin{bmatrix} 7 & 25 & 3 \\ 9 & 20 & 4 \end{bmatrix} \right) \right)$$

$$Ebeveyn 2 : \left(\left(\begin{bmatrix} 3 & 20 & 2 \\ 1 & 10 & 1 \\ 2 & 20 & 1 \end{bmatrix} \begin{bmatrix} 4 & 25 & 1 \\ 5 & 10 & 2 \end{bmatrix} \begin{bmatrix} 6 & 30 & 2 \\ 7 & 25 & 3 \end{bmatrix} \begin{bmatrix} 6 & 30 & 2 \\ 7 & 25 & 3 \end{bmatrix} \begin{bmatrix} 8 & 10 & 4 \\ 9 & 20 & 4 \end{bmatrix} \begin{bmatrix} 10 & 40 & 3 \end{bmatrix} \right) \right)$$

Şekil 3.4 Ebeveyn 1 ve Ebeveyn 2

Denklem (3.68) kullanılarak, birinci ebeveyn için her bir genin seçim değeri sırasıyla 58.63, 41.50, 58.31, 73.50, 58.44 olarak hesaplanmıştır. Aynı şekilde ikinci ebeveyn için ise sırasıyla 56.17, 58.56, 58.31, 40.63, 41.50 olarak hesaplanmıştır. Her ebeveynde minimum seçim değerine sahip olan genler seçilmiş ve çaprazlanmıştır. Çaprazlama operasyonu Şekil 3.5’de gösterildiği gibi ebeveynler arasında gen transferi ile başlamıştır.



Şekil 3.5 Kromozomlar arası gen transferi

Çaprazlama işleminin ikinci adımında Falkenauer (1996) tarafından önerilen yöntem temel alınmış ve bu yöntemle aile birliğini korumak amacıyla yeni kurallar eklenmiştir. Transfer edilen gen içerisinde bulunan nesneleri tekrar eden genler kromozomlardan silinmiştir. Birinci ebeveyne transfer edilen gen, 8 ve 9 numaralı nesneleri içerdiği için bu nesneleri içeren genler kromozomdan silinmiştir. İkinci ebeveyn için aynı işlem uygulanarak beşinci gen kromozomdan silinmiştir. Sonrasında çözümden silinen genlerin içerisinde yer alan diğer nesneler kutulara tekrar yerleştirilmek üzere bir havuza atanmıştır. Çaprazlama aşamasının adımları Şekil 3.6’da gösterilmiştir.

$$Ebeveyn 1 : \left(\begin{bmatrix} 3 & 20 & 2 \\ 1 & 10 & 1 \\ 2 & 20 & 1 \end{bmatrix} \begin{bmatrix} 10 & 40 & 3 \end{bmatrix} \begin{bmatrix} 6 & 30 & 2 \\ 4 & 25 & 1 \end{bmatrix} \begin{bmatrix} 5 & 10 & 2 \\ 8 & 10 & 4 \\ 220 & 1 \end{bmatrix} \begin{bmatrix} 7 & 25 & 3 \\ 9 & 20 & 4 \end{bmatrix} \right) \left(\begin{bmatrix} 8 & 10 & 4 \\ 9 & 20 & 4 \\ transfer\ gen \end{bmatrix} \right)$$

$$Tekrar Atanacak Nesne Havuzu_1 = \begin{pmatrix} 5 & 10 & 2 \\ 7 & 25 & 3 \\ 220 & 1 \end{pmatrix}$$

$$Ebeveyn 2 : \left(\begin{bmatrix} 3 & 20 & 2 \\ 1 & 10 & 1 \\ 2 & 20 & 1 \end{bmatrix} \begin{bmatrix} 4 & 25 & 1 \\ 5 & 10 & 2 \end{bmatrix} \begin{bmatrix} 6 & 30 & 2 \\ 7 & 25 & 3 \end{bmatrix} \begin{bmatrix} 6 & 30 & 2 \\ 7 & 25 & 3 \end{bmatrix} \begin{bmatrix} 8 & 10 & 4 \\ 9 & 20 & 4 \end{bmatrix} \begin{bmatrix} 10 & 40 & 3 \end{bmatrix} \right) \left(\begin{bmatrix} 10 & 40 & 3 \\ transfer\ gen \end{bmatrix} \right)$$

$$Tekrar Atanacak Nesne Havuzu_2 = ([2. Ebeveynde Tekrar Atanacak Nesne Yoktur])$$

Şekil 3.6 Çaprazlama

Kutulara tekrar atama yapılırken ilk olarak nesnenin hangi ürün ailesinden olduğu dikkate alınarak atama yapılmıştır. Yani nesne hangi ürün ailesine aitse o ürün ailesine ait nesnelerin sayıca fazla olduğu kutuya kapasiteyi aşmamak koşulu ile atanmıştır. Eğer nesneler ait oldukları ürün ailesine göre yerleştirilemediğinde, yeterli boş alana sahip ilk kutuya yerleştirilmiştir. Eğer hacim olarak yerleşebileceği bir kutu yoksa yeni bir kutu oluşturulmuş ve nesne oluşturulan kutuya yerleştirilmiştir. Ürün aile birliğinin korunduğu çaprazlama kuralları sonrasında oluşturulan çocuk bireyler Şekil 3.7’de gösterilmiştir.

$$\text{Çocuk 1} : \left(\begin{bmatrix} 3 & 20 & 2 \\ 1 & 10 & 1 \\ 5 & 10 & 2 \\ 2 & 20 & 1 \end{bmatrix} \begin{bmatrix} 10 & 40 & 3 \\ 7 & 25 & 3 \end{bmatrix} \begin{bmatrix} 6 & 30 & 2 \\ 4 & 25 & 1 \end{bmatrix} \begin{bmatrix} 8 & 10 & 4 \\ 9 & 20 & 4 \end{bmatrix} \right)$$

$$\text{Çocuk 2} : \left(\begin{bmatrix} 3 & 20 & 2 \\ 1 & 10 & 1 \\ 2 & 20 & 1 \end{bmatrix} \begin{bmatrix} 4 & 25 & 1 \\ 5 & 10 & 2 \end{bmatrix} \begin{bmatrix} 6 & 30 & 2 \\ 7 & 25 & 3 \end{bmatrix} \begin{bmatrix} 8 & 10 & 4 \\ 9 & 20 & 4 \end{bmatrix} \begin{bmatrix} 10 & 40 & 3 \end{bmatrix} \right)$$

Şekil 3.7 Oluşturulan yeni nesil

Önerilen bu yöntemde aile birliği oranını artıran genler korunarak çaprazlama yapılmıştır. Çaprazlama işleminden sonra oluşan yeni popülasyonun çeşitli hale gelebilmesi için mutasyon işlemi yapılmıştır. Bu işlemde, mevcut kromozomların genleri belirlenen oranda (mutasyon oranı) değişime uğratarak yeni

kromozomlar elde edilmiştir. Mutasyona uğrıtılacak genler rastgele seçilmiştir. Seçilen gen kromozomdan çıkarılarak gen içerisinde ki nesneler çaprazlamada var olan yöntem kullanılarak yeniden yerleştirilmiştir.

3.3.2 İkinci Aşama: Duvar Örme Algoritması

Önerilen sezgisel algoritmanın bu aşamasında problem üç boyutlu kutulama problemi olarak ele alınmıştır. Birinci aşamada, problem tek boyutlu olarak çözülmüş ve nesneler kutulara ürün aile birliği dikkate alınarak yerleştirilmiştir. Ancak gerçek hayat problemlerinde kutu problemleri üç boyutlu ele alınmaktadır. Bu sebeple bu aşamada, problem üç boyutlu olarak ele alınmış ve nesneler duvar örme algoritması kullanılarak tekrar yerleştirilmiştir. Kullanılan duvar örme algoritması literatürden farklı kurallar içermektedir. Öncelikle katman boyutu belirlenirken, nesnelerin genişlik ve uzunluk değerleri karşılaştırılmış ve nesnelerin en büyük kenarı ($\max(l_i, w_i)$) bulunmuştur. Daha sonra nesnelerin büyük olan kenarlarına göre büyüktten küçüğe doğru sıralanmış ve yükleme sırası oluşturulmuştur. Sıralamada en başta olan nesne, kutunun uzun kenarına paralel olacak şekilde başlangıç noktasına yerleştirilmiştir. Böylece ilk katman oluşturulmuştur. Katmanlar doldurulurken, yerleştirilen ilk nesnenin ait olduğu ürün ailesinden olan nesneler sıralamada öne alınarak öncelikli olarak yerleştirilmiştir. Bu işlem katmanda boşluk kalmayana kadar öncelik sırasına göre devam etmiştir. Katmanda boşluk kalmamışsa yeni katmanlar oluşturulmuş ve yerleştirme işlemine devam edilmiştir. Ancak başlangıç çözümlerinin elde edildiği birinci aşamada problem bir boyutlu düşünöldüğünden dolayı bu aşamada kutu içerisine sığmayan nesneler olmuştur. Bu nesneler tekrar yerleştirilmek için havuza atılmıştır. Aynı zamanda kutu içerisinde ürün aile oranını azaltan nesneler var ise onlarda tekrar yerleştirilmek üzere havuza atılmıştır. Havuzda yer alan nesneler tekrar yerleştirilirken önce ürün ailelerine bakılarak mevcut kutulara yeterli alan var ise yerleştirilmiştir. Ancak yeterli alan yok ise yeni bir kutu oluşturulmuş ve kalan nesneler yerleştirilmiştir. Tablo 3.6' da duvar örme algoritmasının temel aşamaları yer almaktadır.

Tablo 3.6 Duvar örme algoritmasının temel aşamaları

Aşama II: Duvar Örme Algoritmasının Temel Aşamaları					
1. Aşama I den gelen çözüm alınır. 2. Her bir boyutlu kutu için üç boyutlu kutu oluşturulur. 3. Belirlenen sıralama ölçütlerine göre öncelik sırası oluşturulur. 4. Katman belirleyen nesne seçilir ve yerleştirilir. 5. Paketlemede önceliğe sahip nesneler kümesi belirlenir (yerleştirilen nesne ile aynı ürün ailesi numarasına sahip nesneler) 6. Öncelik sırasına göre katmanda yer kalmayınca kadar yerleştirilir. 7. Katmanda yeterli alan kalmayınca yeni katman açılır ve kutuda yer kalmayınca kadar devam edilir. 8. Kutuda yeterli alan kalmayınca yerleşemeyen nesneler ve homojenliği bozan nesneler bir havuza atanır. 9. Aşama I den gelen çözümdeki her bir kutu için işlem 3-8 arasındaki işlemler tekrarlanır. 10. Havuza gönderilen kutular ürün ailelerine göre diğer kutulara yerleştirilmeye çalışılır. 11. Kutulara sığmayan nesneler için yeni bir kutu oluşturulur.					

Aşağıda açıklayıcı bir örnek sunulmuştur.

Tablo 3.7 de uzunluk, genişlik ve yüksekliği 100 birim olan kutulara yerleşecek nesnelerin boyutları verilmiştir.

Tablo 3.7 Nesne boyutları

No	Uzunluk	Genişlik	Yükseklik	Hacim	Aile No
1	20	100	96	192000	1
2	34	71	100	241400	1
3	7	94	99	65142	3
4	25	80	68	136000	1
5	7	87	91	55419	2
6	21	69	72	104328	3
7	21	76	82	130872	1
8	15	79	99	117315	2
9	28	97	100	271600	3
10	18	91	73	119574	1

Birinci aşamadan elde edilmiş örnek başlangıç çözüm Şekil 3.8’de belirtildiği gibidir. Bu çözüm kullanılarak nesnelerin 3 boyutlu yerleşim planı oluşturulmuştur.

$$\text{Çözüm A} : \left(\begin{bmatrix} 4 & 136000 & 1 \\ 7 & 130872 & 1 \\ 10 & 119574 & 1 \\ 8 & 117315 & 2 \\ 2 & 241400 & 1 \\ 1 & 192000 & 1 \\ 5 & 55419 & 2 \end{bmatrix} \begin{bmatrix} 9 & 271600 & 3 \\ 6 & 104328 & 3 \\ 3 & 65142 & 3 \end{bmatrix} \right)$$

Şekil 3.8 Örnek başlangıç çözüm

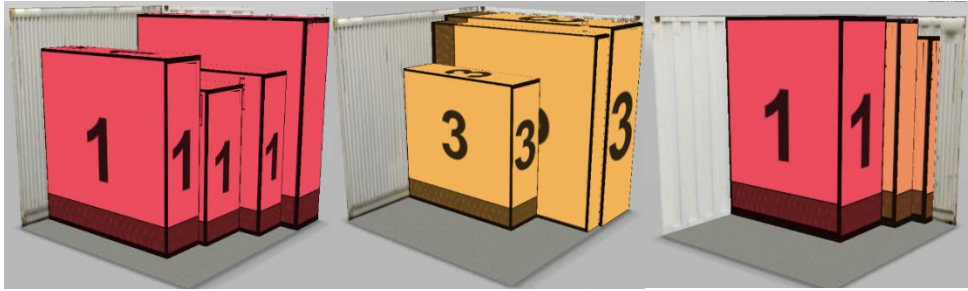
Yukarıda ayrıntısı verilen duvar örme algoritmasına göre birinci kutuda, ilk olarak nesnelerin öncelik sırası belirlenmiş ve birinci nesne kutunun (0,0,0) noktasına yerleştirilerek birinci katman oluşturulmuştur. İlk yerleşen nesne birinci aileden olduğu için ilk olarak birinci aileye ait nesneler kutuya yerleştirilmeye çalışılmıştır. Hem birinci aileden olup hem de yerleştirme sırası takip edildiğinde aynı aileden 1,10,4 ve 7 numaralı nesneler kutuya sığmış ancak 2 numaralı kutu sığmamış ve yeniden yerleştirilmek üzere ayrılmıştır. Ek olarak aile birliğinin korunması ve iyileştirilmesi için aynı aileden olmadığı için 5 ve 8 numaralı nesnelerde yeniden yerleştirilmek için ayrılmıştır. 2. Kutuda ise 3, 6 ve 9 numaralı nesneler aynı aileden olduğu için öncelik sırasına göre kutuya yerleştirilmiştir. Son olarak yeniden yerleştirilmesi gereken 2,5 ve 8 numaralı kutular ürün ailelerine göre başka bir kutuya yerleşemediği için yeni bir kutuya yerleştirilmiştir. Böylelikle ürün aileleri dikkate alındığında kullanılan kutu sayısı 3 olmuştur. Tablo 3.8’de nesnelerin kutu içerisindeki koordinatları verilmiş ve Şekil 3.9’da paketleme planları görselleştirilmiştir. Tezin bundan sonraki kısmında yükleme planları Easy Cargo yazılımı kullanılarak yapılmıştır.

Ek olarak aynı problem ürün aileleri dikkate alınmayıp çözüldüğünde birinci aşamadan gelen başlangıç çözümünde birinci kutuya 9,2,6,4 ve 1 numaralı nesneler, ikinci kutuya ise 3,7,10,5 ve 8 numaralı nesneler atanmıştır. Ürün ailesi ihmal edildiğinde, nesneler ürün ailelerine göre yerleşime zorlanmadığı için

kullanılan kutu sayısı 2 olarak bulunmuş ve Tablo 3.9’da ayrıntısı gösterilmiştir. Şekil 3.10’de ise paketlemenin görsel hali sunulmuştur.

Tablo 3.8 3 Boyutlu yerleşim planı (ürün ailesi dikkate alınarak)

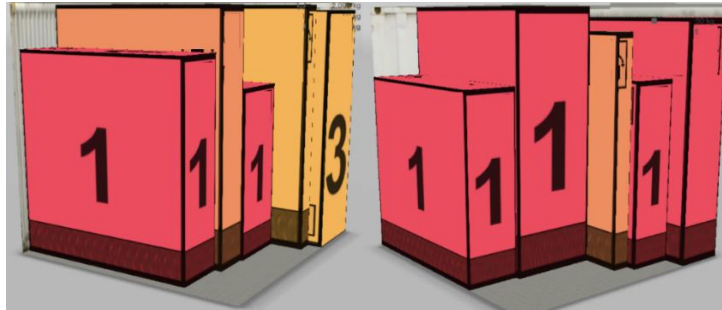
Kutu 1				Ürün Ailesi	Kutu 2				Ürün Ailesi	Kutu 3				Ürün Ailesi
Koordinat					Koordinat					Koordinat				
No	x	y	z		No	x	y	z		No	x	y	z	
1	0	0	0	1	9	0	0	0	3	5	0	0	0	2
10	0	20	0	1	3	0	28	0	3	8	0	7	0	2
4	0	38	0	1	6	0	35	0	3	2	0	22	0	1
7	0	63	0	1										



Şekil 3.9 Yerleşim planı (Ürün ailesi dikkate alınarak)

Tablo 3.9 3 Boyutlu yerleşim planı (ürün ailesi dikkate alınmadan)

Kutu 1				Ürün Ailesi	Kutu 2				Ürün Ailesi
Koordinat					Koordinat				
No	x	y	z		No	x	y	z	
9	0	0	0	3	1	0	0	0	1
3	0	28	0	3	10	0	20	0	1
4	0	35	0	1	5	0	38	0	2
8	0	60	0	2	2	0	45	0	1
7	0	75	0	1	6	0	79	0	1



Şekil 3.10 Yerleşim planı (Ürün Ailesi dikkate alınmadan)

Algoritmanın bu aşamasında önerilen yöntemle hem ürün aile birliği korunmuş hem de üç boyutlu yerleşim planlarının oluşturulması sağlanmıştır.

3.3.3 Üçüncü Aşama: Dengeli Yerleşim Planlarının Oluşturulması

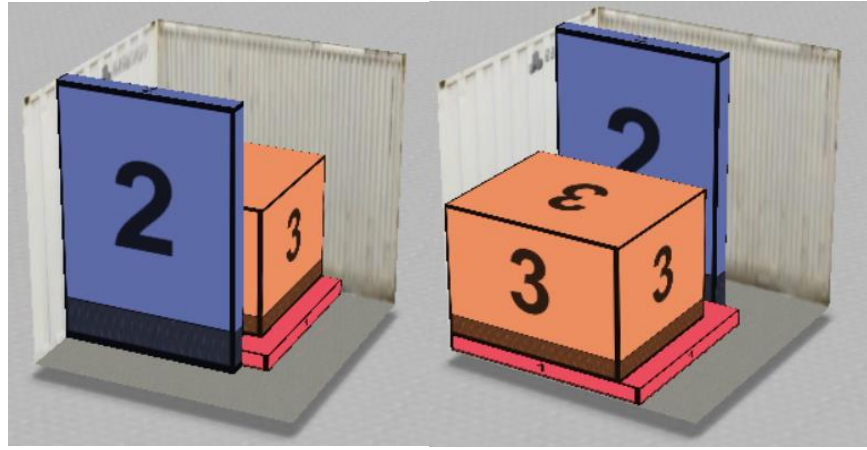
Üçüncü ve son aşamada, her bir kutu için, katmanların farklı yerleşim pozisyonları kullanılarak ideal ağırlık merkezinden sapma minimize edilmiş ve nihai yerleşim planları oluşturulmuştur.

Dengenin ihmal edildiği yüklemelerde taşıma sırasında yükler zarar görmektedir. Ayrıca dengesiz yük dağılımı nedeniyle tırlarda hasarlar meydana gelmektedir. Bu hasarlar taşıma maliyetlerini ciddi anlamda arttırmaktadırlar. Aynı zamanda müşteri memnuniyetini de olumsuz etkilemektedir. Dengenin ihmal edilmediği yerleştirme planları ile olası tüm riskler azaltılabilmektedir. Literatürde bu amaç doğrultusunda bazı çalışmalarda, nesnelerin taban alanının yerleştiği zemin tarafından kısmen veya tamamının desteklenmesi olarak ele alınırken bazı çalışmalarda ise nesne ağırlıklarının kutu zemininde eşit olarak dağılması ile ilgilidir. Ancak bir nesnenin boyutları büyük olmasına rağmen ağırlığı az olabilmektedir. Bu sebeple nesnelerin sadece alttan desteklenmesi yüklerin ağırlıktan dolayı zarar görmeyeceği anlamına gelmemektedir. Bu yüzden nesne ağırlıkları dikkate alınarak, nesnelerin ağırlık merkezinin, kutu tabanının ağırlık merkezine yakın olması amaçlanmıştır. Bu amaçla, Eley (2002) tarafından yapılan çalışma temel alınarak her bir kutu içerisinde yük dengelemesi yapılmıştır.

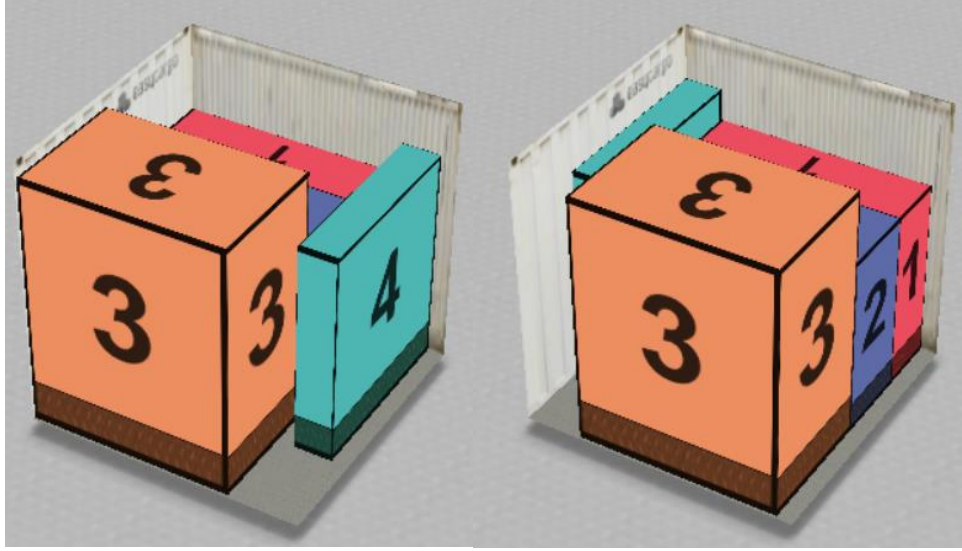
Bu aşamada, her bir kutu içerisinde oluşturulan katmanların kendi aralarında yer değiştirilmesine ve x-y ekseninde aynalanmasına izin verilmiştir. Tablo 3.10 ve Tablo 3.11’de koordinatları verilen, Şekil 3.11 ve Şekil 3.12’de görselleştirilen örnek yerleşim planları yer almaktadır.

Tablo 3.10 Aynalanmış katman örneği

Nesne				İkinci Aşamadan Gelen Yerleşim Koordinatı			Aynalanmış Yerleşim Koordinatı			Yerleştiği Kenar	Katman
No	w	l	h	x	y	z	x	y	z		
1	78	70	6	0	0	0	0	30	0	xyz	1
2	68	7	100	0	70	0	0	23	0	xyz	
3	58	70	50	0	0	6	0	42	6	yxz	

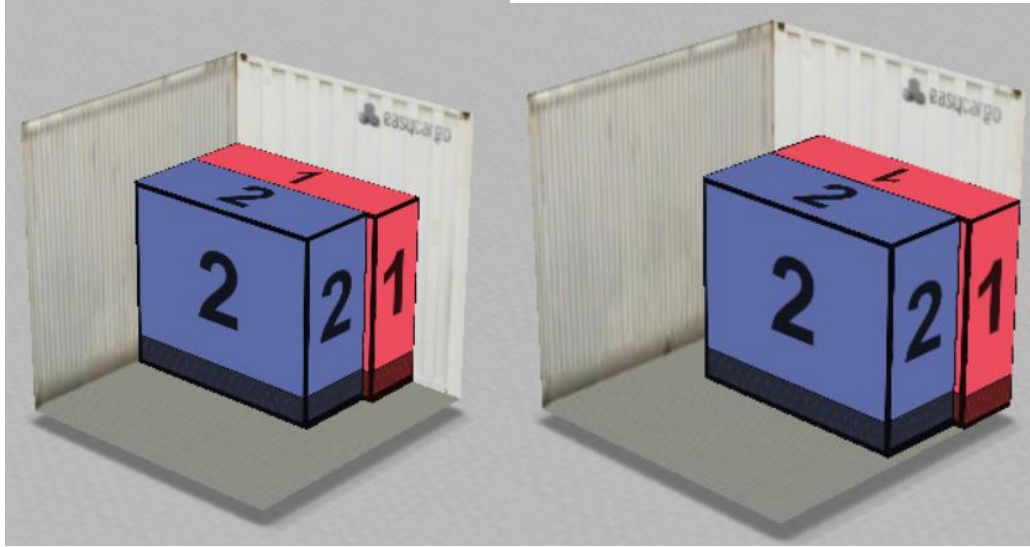
**Şekil 3.11** Aynalanmış katman**Tablo 3.11** Birbirleri arasında yer değiştirmiş katman örneği

Nesne				İkinci Aşamadan Gelen Yerleşim Koordinatı			Yer Değiştirmiş Yerleşim Koordinatı			Yerleştiği Kenar	Katman
No	w	l	h	x	y	z	x	y	z		
1	80	22	67	0	0	0	14	0	0	xyz	1
2	76	29	67	0	22	0	18	22	0	xyz	
3	74	48	94	0	51	0	20	51	0	xyz	
4	68	14	79	80	0	0	0	0	0	yxz	2



Şekil 3.12 Birbirleri arasında yer değiştirmiş katmanlar

Ayrıca her bir katman kutu içinde x ve/veya y eksenini yönünde kaydırılarak ideal ağırlık merkezine yakın konumlandırılmış ve paketleme planları oluşturulmuştur. Örneğin, Şekil 3.13’de gösterildiği gibi, katman ideal ağırlık merkezinden sapmayı iyileştirmek için x ekseninde dört birim sağa kaydırılmıştır.



Şekil 3.13 Kaydırılmış katman örneği

Bu bölümde, önerilen tam sayılı matematiksel modelin ve genetik algoritma tabanlı sezgisel algoritmanın güvenilirliğini test etmek amacıyla test problemleri kullanılarak karşılaştırmalı bir analiz sunulmuştur.

Bu çalışmada, tam sayılı matematiksel modelinin çözümünde GAMS 22.8 paket programı ve genetik algoritma tabanlı sezgisel algoritmanın kodlanmasında NETBEANS IDE 8.2 kullanılmıştır.

4.1 Veri Seti

Literatürde çoklu üç boyutlu kutulama problemleri için Martello vd., (2000) tarafından Tablo 4.1 de ayrıntısı verilen sekiz farklı grup veri seti sunulmuştur.

Tablo 4.1 Kullanılan veri setleri

(k)	Kutu Boyutu			Nesne Boyutu		
	Uzunluk (L)	Genişlik (W)	Yükseklik (H)	Uzunluk (l)	Genişlik (w)	Yükseklik (h)
1	100	100	100	$[1, \frac{1}{2}L]$	$[\frac{2}{3}W, W]$	$[\frac{2}{3}H, H]$
2	100	100	100	$[\frac{2}{3}L, L]$	$[\frac{2}{3}W, W]$	$[1, \frac{1}{2}H]$
3	100	100	100	$[\frac{2}{3}L, L]$	$[1, \frac{1}{2}W]$	$[\frac{2}{3}H, H]$
4	100	100	100	$[\frac{1}{2}L, L]$	$[\frac{1}{2}W, W]$	$[\frac{1}{2}H, H]$
5	100	100	100	$[1, \frac{1}{2}L]$	$[1, \frac{1}{2}W]$	$[1, \frac{1}{2}H]$
6	10	10	10	[1,10]	[1,10]	[1,10]
7	40	40	40	[1,35]	[1,35]	[1,35]
8	100	100	100	[1,100]	[1,100]	[1,100]

Trivella ve Pisinger (2016), sunulan bu veri setlerine yeni bir boyut getirerek nesne ağırlıklarını da dâhil etmişlerdir. Nesnelerin homojen yapıda olduğu ve p_i yoğunluğuna sahip olduğu varsayılmış ve kutu ağırlıkları $a_i = p_i * V_i$ olarak

hesaplanmıştır. (Trivella ve Pisinger,2016). Ek olarak, yapılan sayısal analizlerde nesnelerin yoğunluğunun 1 olduğu varsayılmıştır. Veri setleri oluşturulurken, k grubu için ($k = 1, \dots, 5$), nesnelerin %60'ı k grubu için belirlenen aralıklarda normal dağılıma göre üretilirken geriye kalan %40'ı diğer dört gruptan eşit olacak şekilde verilen aralıklarda normal dağılıma uygun olarak üretilmiştir. Grup 6,7 ve 8'e göre üretilmiş veri setlerinde ise tüm nesneler ilgili grubun uzunluk, genişlik ve yükseklik değerleri için verilen aralıklarda normal dağılıma uygun olarak üretilmiştir. (Martello vd., 2000)

Yapılan tez çalışmasında, literatürde ürün ailesinin dâhil edildiği veri seti olmadığından dolayı mevcut veri setine ürün ailesi bilgisi dâhil edilerek veri setine yeni bir boyut getirilmiştir. Ürün ailesi [1,3] aralığında rassal olarak üretilmiş ve analizler yapılmıştır.

4.2 Önerilen Matematiksel Model Analizi

Bu bölümde önerilen tam sayılı matematiksel modelin doğruluğu test edilmiş ve bir uygulama yapılarak sonuçlar incelenmiştir.

4.2.1 Matematiksel Modelin Doğrulanması

Literatürde yük dengeli 3BKP için tam sayılı matematiksel modelin çıktılarının tartışıldığı ilk ve tek çalışma Trivella ve Pisinger (2006)'e aittir. Bu sebeple önerilen matematiksel modelin testi için bu çalışma ile karşılaştırmalı bir analiz yapılmıştır. Tablo 4.2 'deki veri seti, Trivella ve Pisinger (2006) tarafından açıklayıcı bir örnek sunmak için kullanılmıştır. 16 nesneden oluşan bu veri seti Martello vd., (2000) tarafından sunulan grup 6'ya göre oluşturulmuştur. Nesne yoğunlukları 1 olarak belirlenmiştir. Ek olarak, kutular özdeş ve her bir kenarı 10 birim olan bir küp şeklinde olduğu varsayılmıştır.

Tablo 4.2 16 nesneli 3BKP örneği (Trivella ve Pisinger,2006)

Nesne	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
w_i	9	3	6	6	5	2	7	4	2	7	8	1	2	5	6	3
l_i	7	5	6	1	3	3	3	8	7	10	6	2	4	4	9	7
h_i	2	8	5	8	9	10	8	3	6	3	9	2	1	5	3	3

Trivella ve Pisinger (2016)'in yapmış oldukları çalışmada nesnelerin döndürülmesine izin verilmemiştir. Ayrıca aile birliği kavramı probleme dâhil edilmediğinden, sonuçların tutarlı bir şekilde kıyaslanabilmesi için ilk olarak model iki varsayımla kısıtlanmıştır:

(i) Nesnelerin döndürülmesine izin verilmemiştir

(ii) Aile birliği ile ilgili bileşenler ihmal edilmiştir

Matematiksel model, 8.0 GB RAM ve 64 bit işletim sistemine sahip Intel i5 2.50 GHz bilgisayarda GAMS 22.8'de kodlanmış ve optimum sonuç için çözülmüştür.

Sonrasında, önerilen modelde olduğu gibi nesnelerin 2 yönlü (x ve y ekseninde) döndürülmesine izin verilmiş ve beklendiği gibi daha iyi sonuçlar elde edilmiştir. İdeal ağırlık merkezinin koordinatı (5, 5, 0) olarak belirlenmiştir. İdeal ağırlık merkezinden toplam sapma (3.3)'deki formül kullanılarak hesaplanmış ve sonuçlar Tablo 4.3'te özetlenmiştir.

Tablo 4.3 Matematiksel modeller için karşılaştırmalı sonuçlar

	(Trivella ve Pisinger, 2006)			Önerilen Model (Döndürme Yok)			Önerilen Model (Döndürme Var)		
Kutu	1	2	3	1	2	3	1	2	3
g_j^x	5	5	5	5	5	5	5	5	5.07 1
g_j^y	5	5	5	5	5	5	5	5	5.03 1
g_j^z	3.716	3.341	4.265	3.716	3.341	4.265	3.93 3	2.61 3	4.24 4
Toplam Sapma	11.322			11.322			10.892		

Nesnelerin 2 yönlü döndürülmesine izin verildiğinde amaç fonksiyonunun iyileşmesi tesadüf değildir. Bu veri seti için, nesnelerin yalnızca tabanda döndürmesine izin verildiğinde bile, amaç fonksiyonunda yaklaşık %4'lük bir iyileştirme elde edilmiştir.

Modeli bu yönde test ettikten sonra modele ürün ailesi kavramı dahil edilmiştir. Bu tez çalışması ürün ailesi kavramını literatüre dahil eden ilk çalışmadır.

Bu sebeple, ürün ailesi kavramının önemini daha iyi vurgulamak için Tablo 4.2 de verilen sayısal örnek ile tek aileden oluşan senaryoya ek olarak iki ve üç ürün ailesinden oluşan 3 farklı senaryo oluşturulmuş ve sonuçları analiz edilmiştir. İlk olarak kontrol kolaylığı için, ilk 8 nesnenin (1-8) A ailesinde ve geri kalan nesnelerin (9-16) B ailesinde olduğu iki tür ürün ailesi tanımlanmıştır. Sonrasında, aile çeşitliliğinin sonuçlarını incelemek için, 3 ürün ailesinden oluşan iki farklı senaryo sunulmuştur. 3 ürün ailesinin temsil edildiği senaryo (a), (1-6) nesneler A ailesine, (7-11) nesneler B ailesine ve kalan nesneler ise C ailesine ait olduğu varsayılmıştır. 3 ürün ailesinin temsil edildiği senaryo (b)'de ise (1,5,6,8,12) nesneler A ailesine, (3,7,9,14,16) nesneler B ailesine ve geriye kalan (2,4,10,11,13,15) nesneler ise C ailesine ait olduğu varsayılmıştır.

Çalışmanın bu kısmında, aynı aileden gelen nesnelerin aynı kutuya paketlenmesini değerlendirmek için aile birliği oranı (FUR) olarak isimlendirilen bir kavram oluşturulmuştur.

$\mathcal{A}$, bir kutuda paketlenmiş nesneler kümesi, $\mathcal{F}$ ürün aileleri kümesi ve ν birebir (injektif) fonksiyon olarak tanımlanmıştır. ν fonksiyonu $\mathcal{A}$ 'nın farklı alt kümelerini $\mathcal{F}$ 'nin farklı öğeleri ile eşleştirilmesidir. $\mathcal{A}$ kümesi $\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_f$ alt kümelerinden oluştuğu varsayılmıştır. $\mathcal{A}_f$ alt kümesi denklem (3.72)'de ki gibi $\mathcal{F}$ kümesinde ki f ürün ailesiyle ilişkilendirilir.

$$\mathcal{A}_f = \{i | i \in \mathcal{A} \wedge \nu(i) = f\} \quad \forall f \in \mathcal{F} \quad (3.72)$$

Aile birliği oranı, en yüksek kardinaliteye sahip alt kümenin kardinalitesinin $\mathcal{A}$ 'nın kardinalitesine oranı olarak Denklem (3.73)'de gösterildiği gibi hesaplanmıştır.

$$FUR = \frac{\max_f |\mathcal{A}_f|}{|\mathcal{A}|} \quad (3.73)$$

Ürün aile sayısının bir olduğu durumunda, FUR değeri her bir kutu için %100 olarak hesaplanacağı için herhangi bir fikir vermeyecektir. Bu sebeple, ürün aile sayısının bir olduğu durumlarda bu değer dikkate alınmamıştır.

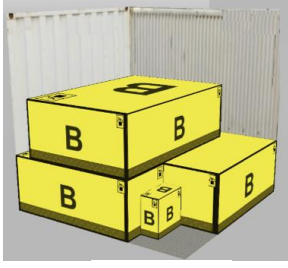
Tablo 4.4 Farklı senaryolar için karşılaştırmalı sonuçlar

	Tek Aileli Senaryo $ F = 1$			2-Aileli Senaryo $ F = 2$			3-Aileli Senaryo (a) $ F = 3$			3-Aileli Senaryo (b) $ F = 3$			
Kutu	1	2	3	1	2	3	1	2	3	1	2	3	4
g_j^x	5	5	5.071	5	5	4.797	5	5	5	5	4.621	5	5
g_j^y	5	5	5.031	5	5.226	5	5	5	5	5	5	5	5
g_j^z	3.933	2.613	4.244	2.603	4.149	4.418	4.193	3.795	4.119	2.806	3.262	4.309	4.401
Nesneler	3,6,7,10,13,14	1,8,12,15,16	2,4,5,9,11	10,12,15,16	1,2,3,4,5,7	6,8,9,11,13,14	7,9,11	8,10,12,13,14,15,16	1,2,3,4,5,6	10,15	1,5,6,8,12	2,4,11,13	3,7,9,14,16
Aile (%)				100% B	100% A	33% A 67% B	100% B	29% B 71% C	100% A	100% C	100% A	100% C	100% B
FUR	-	-	-	100%	100 %	67%	100%	71%	100%	100%	100%	100%	100%
Toplam Sapma	10.892			11.599			12.107			15.157			

Tablo 4.4’de sunulan sayısal analiz sonucunda matematiksel modeldeki üç amacın birbirleri ile çeliştiği gözlemlenmiştir. $|F| = 2$ ve $|F|_{(a)} = 3$ olan senaryolarda, tek aileli senaryoda olduğu gibi tüm nesneler 3 kutu kullanılarak paketlenmiştir. Ürün ailesi sayısında ki artışa rağmen kutu sayısı değişmemiş ancak aile birliği oranı %100 olacak şekilde yerleşim yapılamamıştır.

Ek olarak ürün ailesi birliği bakımından aile sayısının 3 olduğu Senaryo (b)’nin, Senaryo (a)’ dan daha iyi sonuç verdiği ancak kullanılan kutu sayısı ve toplam sapmanın arttığı görülmüştür. Özellikle toplam sapmanın önemli ölçüde arttığı görülmüştür. Bu aşamada çözümleri birden fazla amaç fonksiyonu açısından değerlendirmenin önemi vurgulanmıştır.

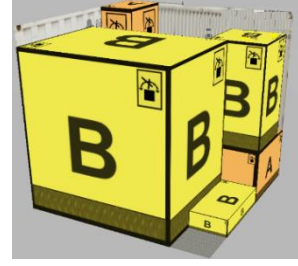
Şekil 4.1’de 2-Aileli Senaryo ve 3-Aileli Senaryo için en uygun paketleme planları gösterilmiştir.



Kutu 1



Kutu 2

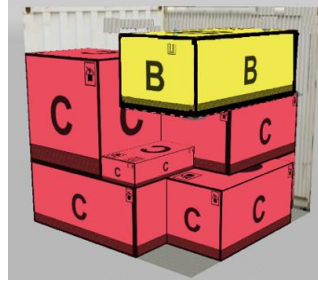


Kutu 3

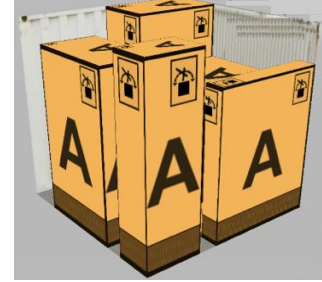
2- Aileli Senaryo



Kutu 1

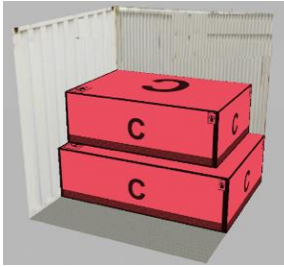


Kutu 2

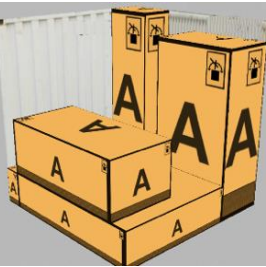


Kutu 3

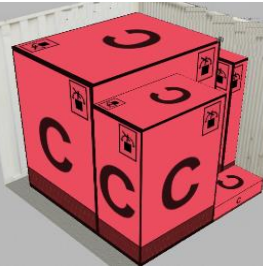
3- Aileli Senaryo (a)



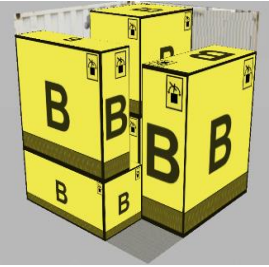
Kutu 1



Kutu 2



Kutu 3



Kutu 4

3- Aileli Senaryo (b)

Şekil 4.1 Optimum yükleme planları

Şekil 4.1' de gösterilen yükleme planları incelendiğinde, 2-Aileli Senaryo ve 3-Aileli Senaryo (a)'da iki kutunun ürün ailesi açısından tamamen homojen olduğu ve 3-Aileli Senaryo (b)'de kullanılan tüm kutuların homojen olduğu açıkça görülmektedir.

4.2.1.1 Ağırlık Parametrelerin Çözümdeki Etkisi

Bu kısımda, amaç fonksiyonunda yer alan amaçlar için belirlenen ağırlık parametrelerinin $(\theta_1, \theta_2, \theta_3)$ sonuçlar üzerindeki etkisini görmek amacıyla karşılaştırmalı bir analiz sunulmuştur.

Tablo 4.5’de 7 farklı senaryo incelenmiştir. İlk 3 senaryoda sadece bir amaç, Senaryo 4-6 da iki amacın farklı kombinasyonları ve son senaryoda ise tüm amaçlar göz önünde bulundurularak çözüm yapılmıştır.

Tablo 4.5 Farklı ağırlık kombinasyonları için karşılaştırmalı sonuçlar

Ağırlık	Senaryo 1 $\theta_1 = 1, \theta_2 = 0, \theta_3 = 0$			Senaryo 2 $\theta_1 = 0, \theta_2 = 0, \theta_3 = 1$			Senaryo 3 $\theta_1 = 0, \theta_2 = 1, \theta_3 = 0$					
Kutu	1	2	3	1	2	3	1	2	3	4		
g_j^x	5.164	4.299	5.261	5	5	5	3.977	4.18	4.266	4.863		
g_j^y	5.483	5.103	4.691	5	5	5	3.919	4.269	4.968	5.011		
g_j^z	5.021	5.591	4.22	4.282	2.976	4.309	4.256	6.838	4.247	4.262		
Aile (%)	80%A 20%C	20% A 40% B 40% C	12% A 50% B 33% C	20%A 60%B 20%C	40%B 60%C	83%A 17%C	100% B	100%C	100%B	100%A		
FUR	80%	40%	50%	60%	60%	83%	100%	100%	100%	100%		
Toplam Sapma	16.853			11.567			24.172					
Ağırlık	Senaryo 4 $\theta_1 = 0.5, \theta_2 = 0.5, \theta_3 = 0$			Senaryo 5 $\theta_1 = 0.5, \theta_2 = 0, \theta_3 = 0.5$			Senaryo 6 $\theta_1 = 0, \theta_2 = 0.5, \theta_3 = 0.5$			Senaryo 7 $\theta_1 = 0.33, \theta_2 = 0.33, \theta_3 = 0.33$		
Kutu	1	2	3	1	2	3	1	2	3	1	2	3
g_j^x	5.195	5.276	5.219	5	5	5	5	5	5	5	5	5
g_j^y	5.558	4.939	4.491	5	5	5	5	5	5	5	5	5
g_j^z	5.729	4.929	4.687	2.280	4.710	4.242	4.193	3.851	4.119	4.193	3.795	4.119
Aile (%)	67%B 33%C	100% A	100%C	67%B 33%C	50% A 50% C	33% A 50% B	100% B	29%B 71%C	100% A	100%B	29%B 71%C	100%A
FUR	67%	100%	100%	67%	50%	50%	100%	71%C	100% A	100%	71%	100%
Toplam Sapma	17.163			11.232			12.163			12.107		

Senaryo 1 ve 2’de kullanılan kutu sayısı 3 iken, Senaryo 3’de $\theta_2 = 1$ alındığında kullanılan kutu sayısı 4’e yükselmiş ve tüm kutularda FUR değeri %100 olarak bulunmuştur. Bununla birlikte, ağırlık merkezinden sapmanın çarpıcı biçimde arttığı görülmüştür. Diğer kombinasyonlarda ise çıktılardaki değişimlerin amaç fonksiyonuna verilen ağırlıklara paralel olduğu gözlemlenmiştir. Tüm senaryolar değerlendirildiğinde, optimal paketleme kararı kullanıcının belirlediği ağırlıklar doğrultusunda değişiklik göstermektedir. Önerilen model, karar vericinin amaç

fonksiyonlarından bir veya birden fazlasından feragat edebileceği ya da edemeyeceği konusunda farkındalık oluşturmaktadır.

4.2.2 Uygulama

Tezin bu kısmında, Avrupa, Rusya ve Orta Doğu bölgeleri olmak üzere ulusal ve uluslararası pazarlar için çeşitli tiplerde filtreler üreten bir filtre fabrikasından veriler alınarak uygulama yapılmıştır. Bu şirket lojistik operasyonlarını denizaşırı bölgeler için taşıyon firmalarına, karayolu ile ulaşılabilen bölgelere ise kendi tırları ile yaptırmaktadır. Fabrika, hava, yağ ve yakıt filtreleri üretmektedir. Hava filtresi, havadaki tozun motora ulaşmasını engelleyen köpük benzeri bir malzemeden oluşmaktadır. Yağ filtresi, motorda kullanılacak yağın süzülmesini ve motora daha temiz yağın gönderilmesini sağlar. Yakıt filtresi ise kullanılan yakıtının daha temiz halde kullanılabilmesi için yakıtın süzülmesine yardımcı olur.

Uygulama yapılacak fabrikada, aynı tipten olan filtreleri birlikte paketlemenin ve kutulardaki ürün çeşitliliğini azaltmanın, lojistik operasyonlarda zaman ve iş gücü kaybı konusunda avantaj sağlayacağı için yükleme yapılırken ürün aileleri dikkate alınmıştır. Tablo 4.6'da, 305×305×365 cm boyutlarında özdeş kutulara yükleme yapılacak olan filtre çeşitleri, boyutları ve adetleri verilmiştir. (Kullanılan veriler firmanın gizlilik kurallarına göre belirli oranda değiştirilmiştir.)

Tablo 4.6 Filtre çeşitleri

Filtre Çeşiti	Adet	Genişlik (cm)	Uzunluk (cm)	Yükseklik (cm)
Hava Filtresi	8	145	145	175
Yağ Filtresi	8	145	145	175
Yakıt Filtresi	12	138	138	158

Tablo 4.6'da gösterildiği gibi, hava ve yağ filtreleri boyut ve miktar olarak aynı olmasına rağmen farklı kullanım ve yükleme gereksinimleri vardır. Bu yüzden farklı kutulara yerleştirilmesi hedeflenmiştir.

Aile birliği kavramının sonuçlar üzerindeki etkisini gözlemlemek için iki farklı senaryo sunulmuştur:

(i) Senaryo I: Filtrelerin çeşidine bakılmaksızın ürün ailesi kavramının ihmal edildiği paketleme

(ii) Senaryo II: Filtrelerin çeşitleri göz önünde bulundurularak aile birliği kavramının probleme dâhil edildiği paketleme

Her iki senaryo için sonuçlar Tablo 4.7'de özetlenmiştir.

Tablo 4.7 Uygulama sonuçları

	Ürün ailesinin dikkate alınmadığı çözüm (Senaryo I)						Ürün ailesinin dikkate alındığı çözüm (Senaryo II)					
Kutu	1	2	3	4	5	6	1	2	3	4	5	6
g_j^x	152.0	152.00	152.00	152.00	152.00	152.00	152.00	152.00	152.00	152.00	152.00	152.00
g_j^y	152.0	152.00	152.00	152.00	152.00	152.00	152.00	152.00	152.00	152.00	152.00	152.00
g_j^z	81.46	85.68	87.50	81.46	83.68	165.79	87.50	145.83	87.50	87.50	79.00	158.00
Hava Filtresi		3	2	1	1	1	2	6				
Yağ Filtresi	1		2		1	4			4	4		
Yakıt Filtresi	3	1		3	2	3					4	8
FUR	75%	75%	50%	75%	50%	50%	100%	100%	100%	100%	100%	100%
Toplam Sapma	585.57						645.33					

Senaryo I ve Senaryo II'nin sonuçları karşılaştırıldığında, ürün ailesinin probleme dahil edildiği senaryoda, aynı aileden gelen ürünler, kullanılan kutu sayısını artırmadan aynı kutuya paketlenmiştir. Her iki senaryoda da kullanılan kutu sayısı 6 olmasına rağmen Senaryo II' de aile birliği oranı her kutu için %100 olarak bulunmuştur. Ancak Senaryo II'de ağırlık merkezinden sapmanın arttığı görülmektedir. Sonuç olarak, ideal ağırlık merkezinden sapma ile ürün aile birliğinin birbirleri ile çelişen amaçlar olduğu açıkça göze çarpmaktadır.

Son olarak, uygulama yapılan fabrikanın stratejisi, müşteriye zamanında teslimat ve doğru sipariş gönderme üzerine kuruludur. Bu sebeple, paketleme

problemlerini çözmek için aile birliği kavramını esas alan analitik yaklaşımın kullanılması fabrikaya birçok avantaj sağlar.

- Zaman ve maliyet açısından daha iyi yükleme planları ve daha iyi elleçleme operasyonları oluşturmalarına olanak tanır.
- Kutu homojenliğini sağlamak için önerilen yaklaşımı kullanmak, çeşitli taşıma ekipmanlarına olan ihtiyacı azaltır
- Teslimat operasyonlarının planlamasını basitleştirir
- Tek bir müşteriye çoklu ziyaretler en aza indirildiği için nakliye maliyetlerini azaltır.

Uygulama yapılan fabrikada sürekli iyileştirme kapsamında gerçekleştirilen bu çalışma, elleçleme, müşteri memnuniyeti, zaman yönetimi, fazla iş yükü ve nakliye maliyetleri açısından daima bir adım öteye taşıyacaktır.

4.3 Önerilen Sezgisel Algoritmanın Analizi

Bu bölümde önerilen sezgisel algoritmanın çözümünden elde edilen sonuçlar ve sonuçların karşılaştırmalı analizi anlatılmıştır. Bu karşılaştırmalı analiz tıpkı matematiksel modelde olduğu gibi Trivella ve Pisinger (2016) in yapmış oldukları çalışmanın sonuçları ile yapılmıştır.

Kullanılan veri setinin ayrıntısı Bölüm 4.1’de verilen standart örneklerden oluşur. Tablo 4.1 de aralıkları verilen her bir veri grubu için on veri seti oluşturulmuştur. Oluşturulan veri setleri aşağıdaki varsayımlarla oluşturulmuştur:

- (i) Öğelerin hepsi aynı aileye aittir. (Ürün ailesi ihmal edilmiştir.)
- (ii) Üç ürün ailesi olduğu varsayılmıştır. Öğeler sadece bir aileye aittir. (Ürün ailesi dahil edilmiştir.)

Genetik algoritmanın doğasında rassallık olduğu için her bir gruptan oluşturulan on farklı veri seti beş iterasyonda çalıştırılmış ve en iyi sonuçlar elde edilmiştir. Sonrasında on veri seti için elde edilen en iyi sonuçların ortalaması alınmıştır.

Elde edilen ortalama sonuçlar Tablo 4.8 ve Tablo 4.9’de özetlenmiştir.

Tablo 4.8 Senaryo I: Karşılaştırmalı sonuçlar (ürün ailesi yok)

Senaryo I: Önerilen Model (Ürün ailesi yok) Amaç Fonksiyonu: Minimum kutu kullanımı Ürün Ailesi=1						
Trivella ve Pisinger (2016) Sonuçları				Aşama I	Aşama II	Aşama III
k	N	Kutu	Sapma	Kutu (1D)	Kutu (3D)	Sapma
1	100	25.64	41.98	17.1	26.4	47.25
2	100	26.12	44.49	16	25.50	45.02
3	100	26.08	41.92	17,4	26.9	47.48
4	100	60.60	38.42	31,7	61	41.65
5	100	14.60	42.35	9,7	15.2	47.13
6	100	20.08	44.91	13.5	22.50	46.40
7	100	12.36	42.36	8.5	14.40	45.04
8	100	17.08	42.08	12.37	20.75	45.46
1	200	51.16	42.56	33.9	51.40	47.203
2	200	50.80	45.20	31.3	49.10	45.84
3	200	51.24	42.67	34.4	52.00	47.15
4	200	122.2	38.19	61.80	122.80	41.17
5	200	27.36	44.14	18.70	28.90	46.62
6	200	38.24	45.77	27.00	42.00	46.14
7	200	22.40	44.51	16.6	28.3	45.26
8	200	32.04	44.37	27.0	43.2	46.37

Tablo 4.8 de görüldüğü gibi elde edilen sonuçlara göre, kullanılan kutu sayısı bakımından Grup 1,2,3,4 ve 5'e göre üretilmiş veri setleri için literatür ile benzer sonuçlar elde edilmiştir. Özellikle taban alanı geniş ve yüksekliği az olan grup 2' ye göre üretilmiş nesnelerin daha iyi yerleştiği gözlemlenmiştir. Çünkü önerilen algoritma duvar örme algoritması temellidir. Duvar örme algoritmasının yapısı büyük uzunluk veya genişliğe sahip olan nesnelerin öncelikli yerleşmesini sağladığı için taban alanı geniş, yüksekliği düşük olan nesnelerin çok olduğu nesne kümeleri ile daha iyi uyum sağlamıştır.

Grup 6,7 ve 8'e göre üretilmiş veri setleri için yapılan analizde kullanılan kutu sayılarında artışın yüksek olduğu gözlemlenmiştir. Duvar örme algoritması nesne çeşitliliğinin zayıf heterojen veya eş olduğu veriler için daha uygundur. İlgili sınıflarda nesne boyutları çok geniş bir aralıkta üretildiğinden nesne çeşitliliği güçlü heterojen yapıdadır. Bu sebeple kutu kullanımında artış olmuştur.

Sonuçlar ideal ağırlık merkezinden sapma açısından değerlendirildiğinde, grup 2' ye göre üretilmiş veri setleri için ideal ağırlık merkezinden sapmanın literatür ile yakın sonuçlar elde edildiği görülmüştür. Çünkü daha önce de belirtildiği gibi veri seti duvar örme algoritmasına uygundur. Oluşan katmanlarda veri çeşitliliği zayıf heterojen olduğu için nesne ağırlıkları da çok değişken değildir. Bu sebeple ideal ağırlık merkezinden sapma da tıpkı kullanılan kutu sayısı gibi literatür ile yakındır.

Ek olarak, grup 6,7 ve 8'e göre üretilmiş veri setlerinde kullanılan kutu sayılarında artışın yüksek olduğu daha önce belirtilmiştir. Özellikle bu artış 200 nesnenin yerleştirildiği veri setlerinde daha fazladır. Bu sebeple kullanılan kutu sayılarında artış olduğu için kutu içlerinde boş alanlar da artmıştır. Önerilen sezgisel algoritmanın denge aşamasında kutu içerisinde oluşan boşluklar kullanılarak, katmanlar ideal ağırlık merkezine göre konumlandırılmaktadır. Bu işlem, boşluklar nedeniyle daha kolay hale gelmektedir. Bu sebeple, bu sınıflarda da ideal ağırlık merkezinden sapma literatür ile benzer sonuçlar elde edilmiştir.

Önerilen sezgisel algoritma ürün ailesi ihmal edilerek çözülmüş ve sonuçları literatür ile karşılaştırılarak güvenilirliği test edilmiştir. Sonrasında ürün ailesi kavramının önemini vurgulamak için farklı senaryolar oluşturulmuştur. Karşılaştırmalı analiz sonuçları Tablo 4.9'da özetlenmiştir.

Tablo 4.9 Senaryo II: Karşılaştırmalı sonuçlar

K	N	Senaryo I(a) Amaç Fonksiyonu: Minimum Kutu Kullanımı				Senaryo II (b) Amaç Fonksiyonu: Maksimum Aile Birliği			
		Aşama I	Aşama II		Aşama III	Aşama I	Aşama II		Aşama III
		Kutu (1D)	Kutu (3D)	Homojen Kutu (3D)	Sapma	Kutu (1D)	Kutu (3D)	Homojen Kutu (3D)	Sapma
1	100	17.10	26.90	12.30	47.82	17.20	27.90	21.10	47.90
2	100	16.00	26.70	13.90	44.90	16.00	27.80	18.90	44.89
3	100	17.40	27.50	11.60	47.84	17.40	28.60	21.40	47.52
4	100	31.90	61.10	49.60	41.61	31.90	63.30	61.80	41.32
5	100	9.70	15.80	5.30	47.83	9.70	16.10	11.50	47.20
6	100	13.70	22.70	7.11	46.35	13.70	22.50	16.80	45.80
7	100	8.50	15.00	2.60	45.77	8.50	15.70	8.50	44.04
8	100	11.5	20.7	9.1	45.60	11.5	21.8	16.1	44.90
1	200	34.10	52.40	23.96	47.13	34.10	54.10	42.10	47.22
2	200	31.40	49.80	25.93	46.43	31.40	51.00	35.60	45.81
3	200	34.20	53.20	28.4	47.84	34.20	54.40	39.80	47.62
4	200	64.50	121.90	98.96	41.60	64.50	125.00	122.00	41.32
5	200	18.70	29.60	9.93	47.10	18.70	30.30	22.80	47.20
6	200	27.40	42.50	17.7	46.00	27.40	43.40	32.10	45.80
7	200	16.60	28.60	4.96	45.04	16.60	28.70	16.60	44.04
8	200	27.9	46.9	26.3	45.80	27.9	48.9	37	44.90

Her grup için on ayrı veri seti üretilmiş ve algoritma beş iterasyonda çalıştırılmıştır. Sonuçlar iki farklı şekilde yorumlanmış ve analiz edilmiştir:

- (i) Senaryo II (a): Kullanılan kutu sayısı minimum olduğunda, bu kutulardan kaç tanesinin ürün ailesi bakımından homojen olduğu (%100 aynı aileden oluştuğu)
- (ii) Senaryo II (b): Homojen kutu sayısı maksimum olduğunda, elde edilen toplam kullanılan kutu sayısının kaç olduğu

Ürün ailesinin ihmal edildiği Senaryo I ve ürün ailesinin probleme dâhil edildiği Senaryo II sonuçları kullanılan kutu sayısına göre karşılaştırıldığında, Senaryo II (a)'da, kullanılan kutu sayısı çok artış olmadığı görülmektedir. Ancak Senaryo

II(b)'de, kullanılan toplam kutu sayısının arttığı gözlemlenmektedir. Kullanılan kutu sayısı ile ürün aile birliğinin birbirleri ile çelişen amaçlar olduğu önerilen matematiksel modelde de vurgulanmıştır.

Senaryo II(a) ve Senaryo II (b)'de, elde edilen kutuların yüzde kaçının homojen olduğu Tablo 4.10'da verilmiştir. Tablodan görüldüğü gibi maksimum aile birliği amaçlandığı Senaryo II (b) de homojen kutu oranı artmıştır.

Tablo 4.10 Homojen kutu oranı

k	N	Senaryo II(a)	Senaryo II(b)
		Oran I	Oran II
1	100	0.46	0.76
2	100	0.52	0.68
3	100	0.42	0.75
4	100	0.81	0.98
5	100	0.34	0.71
6	100	0.31	0.75
7	100	0.17	0.54
8	100	0.44	0.74
1	200	0.46	0.78
2	200	0.52	0.70
3	200	0.53	0.73
4	200	0.81	0.98
5	200	0.34	0.75
6	200	0.42	0.74
7	200	0.17	0.58
8	200	0.56	0.76

Sonrasında Aşama III sonuçları incelendiğinde, kutulardaki homojenlik artmasına rağmen ideal ağırlık merkezinden sapmaların artması beklenirken artmamıştır. Çünkü kullanılan kutu sayısı arttıkça kutularda daha fazla boş alan kalır. Önerilen algoritmanın denge aşamasında, katmanların ağırlık merkezini kutunun ideal ağırlık merkezine yakın yerleştirme işlemi, boşluklar nedeniyle daha kolay hale gelmektedir. Özellikle katmanlar x veya y yönünde ağırlık merkezine doğru kaydırılarak merkeze yakın konumlandırılabilir. Böylelikle ideal ağırlık merkezinden sapma amacı ile diğer amaçlar arasındaki çelişki ortadan kaldırılmaktadır.

Önerilen sezgisel algoritmanın uygunluk fonksiyonunda yer alan amaçlar için belirlenen ağırlık parametrelerinin sonuçlar üzerindeki etkisini görmek amacıyla karşılaştırmalı bir analiz sunulmuştur.

Tablo 4.11’de 200 nesnenin yerleştirildiği grup 6’ya göre üretilmiş veri seti için 7 farklı senaryo incelenmiştir. Senaryo 1 ve senaryo 2’de sadece bir amaç, Senaryo 2-6 da iki amacın farklı kombinasyonları göz önünde bulundurularak çözüm yapılmıştır.

Tablo 4.11 Sezgisel algoritmanın farklı ağırlık kombinasyonları

Senaryo 1					Senaryo 2					Senaryo 3				
Ağırlık $w_1 = 0, w_2 = 1$					$w_1 = 0.1, w_2 = 0.9$					$w_1 = 0.3, w_2 = 0.7$				
j/f	Aile%			FUR	j/f	Aile%			FUR	j/f	Aile%			FUR
	1	2	3			1	2	3			1	2	3	
1		100 %		100 %	1		100 %		100 %	1		100 %		100 %
2			100 %	100 %	2	100 %			100 %	2			100 %	100 %
3	100 %			100 %	3			100 %	100 %	3	100 %			100 %
4	100 %			100 %	4			100 %	100 %	4	100 %			100 %
5		100 %		100 %	5			100 %	100 %	5			100 %	100 %
6		100 %		100 %	6			100 %	100 %	6		100 %		100 %
7			100 %	100 %	7	100 %			100 %	7	100 %			100 %
8			100 %	100 %	8	100 %			100 %	8			100 %	100 %
9	100 %			100 %	9	100 %			100 %	9	100 %			100 %
10	100 %			100 %	10			100 %	100 %	10			100 %	100 %
11		100 %		100 %	11			100 %	100 %	11			100 %	100 %
12			100 %	100 %	12			100 %	100 %	12		100 %		100 %

Tablo 4.11 Sezgisel algoritmanın farklı ağırlık kombinasyonları (devamı)

13	100 %			100 %	13		100 %		100 %	13		100 %		100%
14		100 %		100 %	14	100 %			100 %	14		100 %		100%
15		100 %		100 %	15		100 %		100 %	15		100 %		100%
16			100 %	100 %	16		100 %		100 %	16			100 %	100%
17		100 %		100 %	17		100 %		100 %	17			100 %	100%
18	100 %			100 %	18		100 %		100 %	18			100 %	100%
19			100 %	100 %	19		100 %		100 %	19			100 %	100%
20		100 %		100 %	20	100 %			100 %	20	100 %			100%
21			100 %	100 %	21		100 %		100 %	21			100 %	100%
22			100 %	100 %	22			100 %	100 %	22			100 %	100%
23			100 %	100 %	23			100 %	100 %	23			100 %	100%
24			100 %	100 %	24			100 %	100 %	24	100 %			100%
25			100 %	100 %	25	100 %			100 %	25			100 %	100%
26	100 %			100 %	26	100 %			100 %	26	100 %			100%
27			100 %	100 %	27			100 %	100 %	27	25%		75%	75%
28			100 %	100 %	28			100 %	100 %	28	25%	75%		75%
29			100 %	100 %	29			100 %	100 %	29	67%	33%		67%
30			100 %	100 %	30	100 %			100 %	30		33%	67%	67%
31	100 %			100 %	31	13%		87%	87%	31		60%	40%	60%
32		22%	78%	78%	32	14%	86%		86%	32	60%	40%		60%
33	25%		75%	75%	33	14%	86%		86%	33	40%	60%		60%
34	25%	75%		75%	34	20%	80%		80%	34	22%	56%	22%	56%
35	26%	71%	3%	71%	35		25%	75%	75%	35	25%	25%	50%	50%
36	67%	33%		67%	36		25%	75%	75%	36		50%	50%	50%
37	33%	67%		67%	37	50%		50%	50%	37		50%	50%	50%

Tablo 4.11 Sezgisel algoritmanın farklı ağırlık kombinasyonları (devamı)

38	62%		38%	62%	38		50%	50%	50%	38	83%	17%		83%
39		57%	43%	57%	39		50%	50%	50%	39	33%	33%	33%	33%
40	50%	17%	33%	50%	40	43%	29%	28%	43%					
41	33%	33%	34%	34%										
Sapma 45.5		Homojen Kutu 31			Sapma 48.1		Homojen Kutu 30			Sapma 46.6		Homojen Kutu 26		
Senaryo 4					Senaryo 5					Senaryo 6				
$w_1 = 0.5, w_2 = 0.5$					$w_1 = 0.7, w_2 = 0.3$					$w_1 = 0.9, w_2 = 0.1$				
j f	Aile%			FUR	j f	Aile%			FUR	j f	Aile%			FUR
	1	2	3			1	2	3				1	2	
1	100 %			100 %	1	100 %			100 %	1	100 %			100%
2	100 %			100 %	2	100 %			100 %	2		100 %		100%
3	100 %			100 %	3	100 %			100 %	3		100 %		100%
4			100 %	100 %	4			100 %	100 %	4		100 %		100%
5			100 %	100 %	5			100 %	100 %	5	100 %			100%
6	100 %			100 %	6			100 %	100 %	6			100 %	100%
7			100 %	100 %	7	100 %			100 %	7	100 %			100%
8			100 %	100 %	8			100 %	100 %	8			100 %	100%
9		100 %		100 %	9		100 %		100 %	9			100 %	100%
10		100 %		100 %	10		100 %		100 %	10			100 %	100%
11		100 %		100 %	11	100 %			100 %	11			100 %	100%
12		100 %		100 %	12			100 %	100 %	12		100 %		100%
13			100 %	100 %	13		100 %		100 %	13		100 %		100%
14	100 %			100 %	14		100 %		100 %	14	100 %			100%
15		100 %		100 %	15			100 %	100 %	15		100 %		100%
16			100 %	100 %	16		100 %		100 %	16	100 %			100%
17			100 %	100 %	17		100 %		100 %	17			100 %	100%

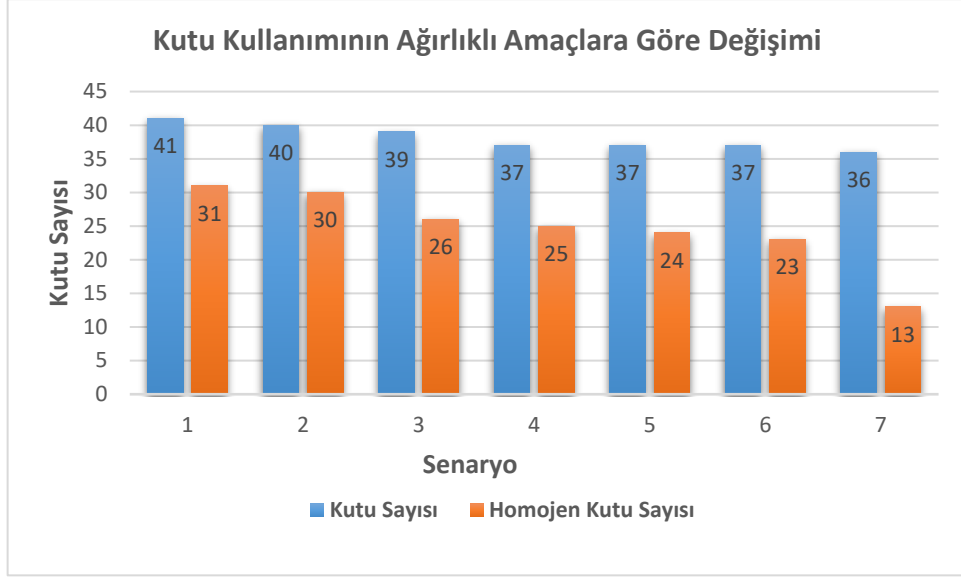
Tablo 4.11 Sezgisel algoritmanın farklı ağırlık kombinasyonları (devamı)

18			100 %	100 %	18		100 %		100 %	18			100 %	100%
19	100 %			100 %	19			100 %	100 %	19			100 %	100%
20		100 %		100 %	20			100 %	100 %	20			100 %	100%
21			100 %	100 %	21			100 %	100 %	21			100 %	100%
22			100 %	100 %	22			100 %	100 %	22			100 %	100%
23			100 %	100 %	23			100 %	100 %	23	100 %			100%
24			100 %	100 %	24	100 %			100 %	24		87%	13%	87%
25	100 %			100 %	25		89%	11%	89%	25		87%	13%	87%
26	33%		67%	67%	26		80%	20%	80%	26	71%		29%	71%
27	60%	40%		60%	27		25%	75%	75%	27	67%		33%	67%
28	22%	78%		78%	28	8%	17%	75%	75%	28	33%	67%		67%
29	33%	67%		67%	29	71%		29%	71%	29		33%	67%	67%
30	50%		50%	50%	30		33%	67%	67%	30	17%	17%	66%	66%
31		67%	33%	67%	31	31%	62%	7%	62%	31	36%		64%	64%
32		67%	33%	67%	32	60%		40%	60%	32	40%		60%	60%
33	17%	25%	58%	58%	33		44%	56%	56%	33		40%	60%	60%
34	40%	57%	3%	57%	34	50%	50%		50%	34	40%	60%		60%
35		50%	50%	50%	35		50%	50%	50%	35	57%	29%	14%	57%
36		50%	50%	50%	36	50%		50%	50%	36	25%	25%	50%	50%
37	33%	33%	33%	33%	37	42%	29%	29%	42%	37	33%	33%	33%	33%
Sapma 48.6		Homojen Kutu 25			Sapma 49.1		Homojen Kutu 24			Sapma 48.9		Homojen Kutu 23		
Senaryo 7														
$w_1 = 1, w_2 = 0$														
i / f	Aile%			FUR	i / f	Aile%			FUR	i / f	Aile%			FUR
f	1	2	3		f	1	2	3		f	1	2	3	

Tablo 4.11 Sezgisel algoritmanın farklı ağırlık kombinasyonları (devamı)

1	100 %			100 %	13	100 %			100 %	25	50%	25%	25 %	50%
2		100 %		100 %	14	75%		25%	75%	26	50%	25%	25 %	50%
3			100%	100 %	15		25%	75%	75%	27	50%	25%	25 %	50%
4			100%	100 %	16		33%	67%	67%	28		50%	50 %	50%
5	100 %			100 %	17		67%	33%	67%	29		50%	50 %	50%
6		100 %		100 %	18	17%	66%	17%	66%	30		50%	50 %	50%
7			100%	100 %	19	37%	63%		63%	31		50%	50 %	50%
8		100 %		100 %	20	60%		40%	60%	32	20%	30%	50 %	50%
9			100%	100 %	21	22%	22%	56%	56%	33	12%	44%	44 %	44%
10			100%	100 %	22		50%	50%	50%	34	43%	14%	43 %	43%
11			100%	100 %	23	50%	50%		50%	35	42%	29%	29 %	42%
12			100%	100 %	24	30%	20%	50%	50%	36	33%	33%	33 %	33%
Sapma 48.8				Homojen Kutu 13										

Tablo 4.11’de sunulan karşılaştırmalı analiz sonucunda, tüm kombinasyonlarda çıktılardaki değişimlerin verilen ağırlıklara paralel olduğu gözlemlenmiştir. Analiz sonuçlarının özet gösterimi Şekil 4.2’ de sunulmuştur. Görüldüğü gibi ürün aile birliğinin en yüksek olduğu senaryo $w_1 = 0, w_2 = 1$ alındığı birinci senaryodur. Aynı şekilde en düşük olduğu senaryo ise $w_1 = 1, w_2 = 0$ alındığı yedinci senaryodur. Dördüncü, beşinci ve altıncı senaryolarda kullanılan kutu sayısı değişmemiştir. Ancak ürün aile birliği amacına verilen ağırlık arttıkça kutularda homojenlik artmıştır.



Şekil 4.2 Kutu kullanımının amaçlara göre değişimi

Tüm senaryolar değerlendirildiğinde, paketleme kararı kullanıcının belirlediği ağırlıklar doğrultusunda değişiklik göstermektedir. Önerilen sezgisel algoritma, karar vericinin yükleme stratejilerine bağlı olarak minimum kutu kullanımı veya maksimum aile birliği hedeflerinden birinden feragat edebileceği ya da edemeyeceği konusunda farkındalık oluşturmaktadır.

İşletmeler arasındaki rekabet koşullarının gün geçtikçe arttığı lojistik süreçlerde işletmeler, maliyetlerini azaltmak ve müşterilerine daha verimli hizmet verebilmek için kendilerini geliştirmektedir. Lojistik süreçlerde, işletmeler öncelikle ürünlerin verimli bir şekilde kutulara yüklenmesi ile ilgilenirler. Ayrıca etkin paketleme planlarının lojistik ve nakliye maliyetlerini önemli ölçüde azaltmakla kalmayıp müşteri memnuniyetini de artırdığı bilinmektedir. Literatürde ürünler kutulara yerleştirilirken, çoğunlukla gerçek hayat problemlerinde kullanılan kısıtlar göz ardı edilmektedir. Özellikle, pratikte ürün bazlı paketleme sıklıkla kullanılmasına rağmen, literatürde konu ile ilgili çalışmalar nispeten azdır. Bu nedenle, gerçek hayat problemlerinin kısıtları göz önünde bulundurularak, nesnelerin döndürülmesine izin verilen ve ürün ailesi kavramı aracılığıyla nesne ile ilgili endişelerin dikkate alındığı yük dengeli 3BKP için matematiksel model ve genetik algoritma tabanlı sezgisel algoritma sunulmuştur. Problemden ele alınan ürün ailesi kavramı, aynı ürün ailesinden olan ürünlerin birlikte paketlenmesine teşvik eden ürün aile birliği kavramı olarak literatüre sunulmuştur. Bu yeni kavram, aynı varış noktasına sahip olma, taşıma gereksinimleri, depolama koşulları gibi pratik hususlar nedeniyle aynı kutuya paketlenmesi gereken ürünleri ele almaktadır. Literatürde ürün ailesi kavramına benzer kavramlar mevcuttur. Literatürde aynı boyut veya özellikte olan ürünler aynı ürün ailesinden olduğu varsayılırken bu çalışmada farklı boyut ve özellikte olan ürünler aynı aileye dâhil edilebilmektedir.

Önerilen matematiksel modelde öncelikle doğrusal olmayan kısıtlar ve amaç fonksiyonu doğrusal hale getirilmiş ve çözülmüştür. Modeli doğrulamak için, Trivella ve Pisinger (2016)'ın problemi, nesnelerin döndürülmesine izin verilmeden ve tüm nesnelerin aynı aileden olduğu varsayılarak önerilen model ile çözülmüştür. Ardından, önerilen modelde olduğu gibi nesnelerin 2 yönlü döndürülmesine izin verilmiş ve karşılaştırmalı sonuçlar incelenmiştir. Nesnelerin

2-yönlü dönmesine izin verildiğinde amaç fonksiyonunda %4 oranında iyileşme olmuştur. Son olarak ürün ailelerinin sayısı değiştirilerek yükleme planları oluşturulmuş ve aynı aileye ait nesnelerin mümkün olduğunca aynı kutuya atandığı gözlemlenmiştir. Ancak aile sayısı arttıkça ideal ağırlık merkezinden sapmanın da arttığı görülmüştür.

Ayrıca, önerilen matematiksel model gerçek bir 3BKP çözümü için kullanılmış ve sonuçlar değerlendirilmiştir. Firmanın üç çeşit ürün ailesi bulunmaktadır. Paketleme faaliyetlerini bu ailelerin özel gereksinimlerini göz ardı ederek yürütmektedir. Ürün aileleri arasındaki farklılıkların göz ardı edilmesi, yükleme işlemlerinde karmaşıklığı artırır ve dolaylı olarak lojistik maliyetlere neden olur. Bu nedenle, elleçleme, yükleme ve depolama operasyonlarında ürün ailelerine göre paketlemenin avantajından yararlanmak için ürün ailesini göz önünde bulundurularak paketleme kararı verilmektedir. Gerçek bir problemin önerilen matematiksel model ile sonuçları analiz edildiğinde, ürün ailesinin dikkate alınıp veya alınmadığı durumda da elde edilen kutu sayısının aynı kaldığını göstermektedir. Ancak ürün ailesin dikkate alındığı durumda ideal ağırlık merkezinden sapma artmıştır. Sonuçlar, ürün ailesi birliği ile yük dengesi hedefleri arasında bir çelişki olduğunu göstermektedir.

Sonuç olarak, önerilen matematiksel modelin amacı, nesnelerin 2-yönlü dönmesine izin verilen ve ürün ailesi birliği aracılığıyla ürünle ilgili endişelerin dikkate alındığı, yük dengeli 3BKP için bir çözüm sunmaktır. Önerilen matematiksel model problemin kapsamlı kısıtlar altında analiz edilebileceği ve kullanılan kutu sayısını, ürün aile birliğini ve ideal ağırlık merkezinden sapmanın minimum yapılabileceği bir dizi amaçtan oluşan ilk matematiksel modeldir.

Önerilen matematiksel model, yük dengelemenin yanında aile birliğinin de elde edilebileceğini göstererek hem pratik uygulamalar hem de teorik uygulamalar için umut vericidir. Ancak büyük boyutlu problemlerin çözümünde matematiksel modelleme yaklaşımı yetersiz kabul edilebilir. Bu sebeple, büyük boyutlu 3BKP'nin çözümü için genetik algoritma tabanlı sezgisel algoritma önerilmiştir.

Önerilen sezgisel algoritmanın ilk aşamasında problem bir boyutlu olarak ele alınmış ve nesneler genetik algoritma kullanılarak ürün ailelerine göre kutulara

atanmıştır. Önerilen genetik algoritmada kullanılan uygunluk fonksiyonu ve çaprazlama aşaması literatürden farklıdır. Uygunluk fonksiyonunda aile birliğinin hesaba katılması ile literatüre yeni bir uygunluk fonksiyonu sunulmuştur. Aynı şekilde çaprazlama aşamasında da ürün aile birliğinin korunmasını amaçlayan bir çaprazlama yöntemi önerilmiştir. İkinci aşamasında, birinci aşamadan elde edilen çözüm başlangıç çözüm olarak kabul edilmiş ve problem üç boyutlu ele alınmıştır. Bu aşamada duvar örme algoritması kullanılarak, nesneler üç boyutlu kutulara tekrar yerleştirilmiştir. Duvar örme algoritması, ürün ailelerinin bir arada paketlenmesi sağlayacak şekilde güncellenmiş ve sunulmuştur. Algoritmanın son aşamasında ise, her bir kutunun dengeli paketlenmesi amaçlanmış ve nihai yerleşim planları oluşturulmuştur.

Sezgisel algoritmayı doğrulamak için, Trivella ve Pisinger (2016)'in yaptıkları çalışma sonuçları kullanılmıştır. İlk olarak, nesnelerin aynı aileden olduğu varsayılarak çözülmüştür. Elde edilen sonuçlara göre, kullanılan kutu sayısı bakımından 1-5 arasında ki veri setleri için literatür ile benzer sonuçlar elde edilmiştir. Özellikle taban alanı geniş ve yüksekliği az olan grup 2'ye göre üretilmiş nesnelerin daha iyi yerleştiği gözlemlenmiştir. Ancak grup 6,7 ve 8'e göre üretilmiş veri setlerinde kullanılan kutu sayılarında artışın yüksek olduğu gözlemlenmiştir. Sonrasında, ürün aile sayısı [1,3] arasında rassal olarak üretilmiş ve karşılaştırmalı sonuçlar incelenmiştir. Ürün ailesi dahil edildiğinde sonuçlar iki farklı yönden analiz edilmiştir. Önce kullanılan bin sayısının minimum olduğu durumda ürün ailesi bakımından homojen kutu sayısına bakılmıştır. Bu durumda kullanılan kutu sayısı mevcut literatürle benzerlik göstermektedir. Ancak aynı ürün ailelerinin birlikte paketlenmediği kutu sayılarının oldukça düşük olduğu görülmüştür. Sonrasında ürün ailesi bakımından homojen kutu sayısının maksimum olduğu durumda kullanılan kutu sayısının arttığı görülmüştür.

Sonuçlar ideal ağırlık merkezinden sapma açısından değerlendirildiğinde, hem ürün ailesinin dikkate alınmadığı durumda hem de ürün ailesinin probleme dahil edildiği durumda grup 2'ye göre üretilmiş veri setinde literatür ile yakın sonuçlar elde edildiği görülmüştür. Ek olarak grup 6,7 ve 8'e göre üretilmiş veri setlerinde, özellikle 200 nesnenin paketlenmediği senaryolarda literatür ile benzer sonuçlar elde

edilmiştir. İlgili sınıflarda kutu kullanımı ciddi anlamda arttığı için kutu içerisinde yük dengelemesi daha kolay olabilmektedir.

Sonuç olarak, önerilen sezgisel algoritmanın amacı, nesnelerin 2-yönlü dönmesine izin verilen ve ürün ailesi birliği aracılığıyla ürünle ilgili endişelerin dikkate alındığı, yük dengeli büyük boyutlu 3BKP için bir çözüm sunmaktır. Önerilen algoritmanın, zayıf heterojen ve eş nesnelerin yerleştirilmesinde daha iyi sonuçlar elde edildiği gözlemlenmiştir. Ek olarak kutu içerisinde homojenlik arttıkça ideal ağırlık merkezinden sapmanın artması beklenirken fazla kutu kullanımından dolayı artmadığı görülmüştür. Böylelikle iki amaç arasında çakışma ortadan kaldırılmıştır.

- Alonso, M. T., Alvarez-Valdes, R., Iori, M., & Parreño, F. (2019). Mathematical models for multi container loading problems with practical constraints. *Computers & Industrial Engineering*, 127, 722-733.
- Alonso, M. T., Alvarez-Valdes, R., Iori, M., Parreño, F., & Tamarit, J. M. (2017). Mathematical models for multicontainer loading problems. *Omega*, 66, 106-117.
- Almeida, A., Figueiredo, M. B. 2010. A particular approach for the three-dimensional packing problem with additional constraints. *Computers and Operations Research*, 37, 1968-1976.
- Araya I., Riff M-C. (2014). A beam search approach to the container loading problem. *Comput Oper Res.*, 43, 100–7.
- Araya, I., Guerrero, K., & Nuñez, E. (2017). VCS: A new heuristic function for selecting boxes in the single container loading problem. *Computers & Operations Research*, 82, 27-35.
- Ayhan, Esra. (2021). Kan tedarik zinciri dağıtım ağı modellemesi ve hibrit genetik algoritma ile çözümü. Doktora Tezi, Yıldız Teknik Üniversitesi
- Baldi, M. M., Perboli, G., & Tadei, R. (2012). The three-dimensional knapsack problem with balancing constraints. *Applied Mathematics and Computation*, 218(19), 9802-9818.
- Bortfeldt, A. (2012). A hybrid algorithm for the capacitated vehicle routing problem with three-dimensional loading constraints. *Computers & Operations Research*, 39(9), 2248-2257.
- Bortfeldt, A., Wäscher, G. (2013). Constraints in container loading—A state-of-the-art review. *European Journal of Operational Research*, 229(1), 1-20.
- Bortfeldt, A, Gehring, H. (2001). A hybrid genetic algorithm for the container loading problem. *European Journal of Operational Research*, 131, 143-161.
- Bischoff, E.E., Ratcliff, M.S.W. (1995). Issues in the development of approaches to container loading. *Omega* 23, 377–390.
- Bortfeldt A, Gehring H, Mack D. (2003). A parallel tabu search algorithm for solving the container loading problem. *Parallel Comput*; 29(5), 641–62.
- Ceschia, S., & Schaerf, A. (2013). Local search for a multi-drop multi-container loading problem. *Journal of Heuristics*, 19(2), 275-294.
- 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.
- Christensen, S. G., & Rousøe, D. M. (2009). Container loading with multi-drop constraints. *International Transactions in Operational Research*, 16(6), 727-743.
- Crainic, T. G., Gobbato, L., Perboli, G., & Rei, W. (2016). Logistics capacity planning: A stochastic bin packing formulation and a progressive hedging meta-heuristic. *European Journal of Operational Research*, 253(2), 404-417.

- Crainic TG, Perboli G, Tadei R (2008) Extreme point-based heuristics for three-dimensional bin packing. *INFORMS J Comput* 20, 368–384
- Costa, M. da G., Captivo, M. E. 2016. Weight distribution in container loading: A case study. *International Transactions in Operational Research*, 23(1–2), 239–263.
- de Castro Silva, J., Soma, N.Y., Maculan, N., (2003). A greedy search for the three-dimensional bin packing problem: the packing static stability case. *International Transactions in Operational Research* 10, 141–153.
- Deb, K. (2001). *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, Inc., ISBN: 978-0-471-87339-6. USA.
- Dereli, T. ve Daş, G.S. (2011). A Hybrid Bee(s) Algorithm for solving Container Loading Problems. *Applied Soft Computing*, 11(2), 2854-2862
- EasyCargo [Computer Software]. 2021. Retrieved from <https://www.easycargo3d.com/>
- Egeblad, J., & Pisinger, D. (2009). Heuristic approaches for the two-and three-dimensional knapsack packing problem. *Computers & Operations Research*, 36(4), 1026-1049.
- Egeblad, J., Garavelli, C., Lisi, S., & Pisinger, D. (2010). Heuristics for container loading of furniture. *European Journal of Operational Research*, 200(3), 881-892.
- Eley, M. (2002). Solving container loading problems by block arrangement. *European Journal of Operational Research*, 141, 393-409.
- Eley, M. (2003). A bottleneck assignment approach to the multiple container loading problem. *OR Spectrum*, 25(1), 45-60.
- Elhedhli, S., Gzara, F., & Yildiz, B. (2019). Three-dimensional bin packing and mixed-case palletization. *Inform Journal on Optimization*, 1(4), 323-352.
- Fan, Y., Chu, J., & Xu, H. (2020). Improvement Grouping Genetic Algorithm for Solving the Bin Packing Problem. *Journal of Physics: Conference Series*, 1550
- Fanslau T, Bortfeldt A. (2010). A tree search algorithm for solving the container loading problem. *INFORMS J Comput*; 22(2), 222–35
- Falkenauer, E. (1996). A hybrid grouping genetic algorithm for bin packing. *Journal of Heuristics*, 2(2), 5–30.
- Feng, X., Moon, I., Shin, J. (2015). Hybrid genetic algorithms for the three-dimensional multiple container packing problem. *Flexible Services and Manufacturing Journal*, 27(2–3), 451–477.
- Fuellerer, Guenther & Doerner, Karl F. & Hartl, Richard F. & Iori, Manuel, (2010). Metaheuristics for vehicle routing problems with three-dimensional loading constraints. *European Journal of Operational Research*, Elsevier, 201(3), 751-759.

- Gimenez Palacios, I., Alonso, M. T., Alvarez Valdes, R., & Parreno, ~ F. (2020). Logistic constraints in container loading problems: the impact of complete shipment conditions. *Top*, 29, 177–203
- Gilmore, P. and Gomory, R. (1963). A linear programming approach to the cutting stock problem – part II. *Operations Research*, 11, 863–888.
- George, J.A. and Robinson, D.F. (1980). A heuristic for packing boxes into a container. *Computers & Operations Research*, 7(3), 147–156.
- Gendreau, M., Iori, M., Laporte, G., & Martello, S. (2006). A tabu search algorithm for a routing and container loading problem. *Transportation Science*, 40(3), 342-350.
- Gonzalez, Y., Miranda, G., & Leon, C. (2016). Multi-objective multi-level filling evolutionary algorithm for the 3d cutting stock problem. *Procedia Computer Science*, 96, 355–364.
- Gonçalves, J. F., Resende, M. G. C. (2013). A biased random key genetic algorithm for 2D and 3D bin packing problems. *International Journal of Production Economics*, 145(2), 500–510.
- He K, Huang W. (2011) An efficient placement heuristic for three-dimensional rectangular packing. *Comput Oper Res.*, 38(1), 227–33.
- Iori M, Martello S (2010). Routing problems with loading constraints. *TOP* 18(1),4–27
- Junqueira, L., Morabito, R., & Yamashita, D. S. (2012). Three-dimensional container loading models with cargo stability and load bearing constraints. *Computers & Operations Research*, 39(1), 74-85.
- Jamrus, T., Chien, C. F. (2016). Extended priority-based hybrid genetic algorithm for the less-than-container loading problem. *Computers and Industrial Engineering*, 96, 227–236.
- Karabulut, K., İnceoğlu, M. M. (2004). A hybrid genetic algorithm for packing in 3D with deepest bottom left with fill method. *Advances Information Systems, İzmir*, 441-450
- Kang, K., Moon, I., & Wang, H. (2012). A hybrid genetic algorithm with a new packing strategy for the three-dimensional bin packing problem. *Applied Mathematics and Computation*, 219(3), 1287-1299.
- Kabasakal, İ., & Keskin, F.D. (2017). Decision support system based on genetic algorithm for variable for variable sized bin packing problem with item conflicts. *International Review of Economics and Management*, 5(1), 1-17.
- Kır, Sena. (2019). Üç Boyutlu Palet Yükleme Probleminin Karişik Tam Sayili Programlama (Milp) Ve Hibrit Genetik Algoritma İle Çözümü. Doktora Tezi, Sakarya Üniversitesi.
- Liu, D. S., Tan, K. C., Huang, S. Y., Goh, C. K., & Ho, W. K. (2008). On solving multiobjective bin packing problems using evolutionary particle swarm optimization. *European Journal of Operational Research*, 190(2), 357-382.

- Liu, J., Yue, Y., Dong, Z., Maple, C., & Keech, M. (2011). A novel hybrid tabu search approach to container loading. *Computers & Operations Research*, 38(4), 797-807.
- Lim, A., Ma, H., Qiu, C., Zhu, W. (2013). The single container loading problem with axle weight constraints. *International Journal of Production Economics*, 144(1), 358-369.
- Loh, H. T., Nee, A. Y. C. (1992). A packing algorithm for hexahedral boxes. *Proceedings of the Industrial Automation Conference*, Singapore, 2, 115-126,
- Lodi, A., Martello, S., Vigo, D., (2002). Heuristic algorithms for the three-dimensional bin packing problem. *European Journal of Operational Research*, 141, 410-420
- Lin, J-L., Foote, B., Pulat, S., Chang, C-H., Cheung, J. Y. (1993). Hybrid genetic algorithm for container packing in three dimensions. *9th IEEE Conference on Artificial Intelligence for Applications*, Orlando, 353-359
- Lim, A., Ma, H., Xu, J., Zhang, X., (2012). An iterated construction approach with dynamic prioritization for solving the container loading problems. *Expert Systems with Applications* 39, 4292-4305.
- Lai, K., Chan, J., (1997). Developing a simulated annealing algorithm for the cutting stock problem. *Computers and Industrial Engineering* 32, 115-127.
- Liu, D., Teng, H., (1999). An improved BL-algorithm for genetic algorithm of the orthogonal packing of rectangles. *European Journal of Operational Research* 112, 413-420
- Lodi, A., Martello, S., Vigo, D., (2004). TSPack: a unified tabu search code for multi-dimensional bin packing problems. *Annals of Operations Research* 131, 203-213.
- Martello, S., & Vigo, D. (1998). Exact solution of the two-dimensional finite bin packing problem. *Management science*, 44(3), 388-399.
- Martello, S., Pisinger, D., & Vigo, D. (2000). The three-dimensional bin packing problem. *Operations Research*, 48(2), 256-267.
- Moon, I., & Nguyen, T. V. L. (2014). Container packing problem with balance constraints. *OR spectrum*, 36(4), 837-878.
- Moura A, Oliveira JF. (2005). A grasp approach to the container-loading problem. *IEEE Intell Syst*, 20(4), 50-7.
- Mohamadi, N. (2010). Application of genetic algorithm for the bin packing problem with a new representation scheme. *Mathematical Sciences*, 4(3), 253-266.
- Pedruzzi, S., Nunes, L. P. A., Rosa, R., & Arpini, B. (2016). A mathematical model to optimize the volumetric capacity of trucks utilized in the transport of food products. *Gest. Prod*, 23, 350-364.
- Pisinger, D. (2002). Heuristics for the container loading problem. *European Journal of Operational Research*, 141, 382-392.

- Pimpawat, C., Chaiyaratana, N. (2001). Using a co-operative co-evolutionary genetic algorithm to solve a three-dimensional container loading problem. Congress on Evolutionary Computation, Seoul, 1197-1204.
- Parreño F, Alvarez-Valdes R, Tamarit J, Oliveira J. (2008). A maximal-space algorithm for the container loading problem. *INFORMS J Comput.*, 20(3), 412–22.
- Paquay, C., Limbourg, S., Schyns, M. (2018). A tailored two-phase constructive heuristic for the three-dimensional Multiple Bin Size Bin Packing Problem with transportation constraints. *European Journal of Operational Research*, 267(1), 52– 64.
- Queiroz, T Ad, Miyazawa, F. K., Wakabayashi, Y., & Xavier, E. C. (2012). Algorithms for 3d guillotine cutting problems: Unbounded knapsack, cutting stock and strip packing. *Computers & Operations Research*, 39(2), 200–212
- Ratcliff, M. S. W., E. E. Bischoff. (1998). Allowing for weight considerations in container loading. *OR Spectrum*, 20, 65–71.
- Ramos, A. G., Silva, E., & Oliveira, J. F. (2018). A new load balance methodology for container loading problem in road transportation. *European Journal of Operational Research*, 266(3), 1140-1152.
- Ramos, A. G., Oliveira, J. F., Gonçalves, J. F., Lopes, M. P. (2016). A container loading algorithm with static mechanical equilibrium stability constraints. *Transportation Research Part B: Methodological*, 91, 565–581
- Sheng, L., Xiuqin, S., Changjian, C., Hongxia, Z., Dayong, S., Feiyue, W. (2017). Heuristic algorithm for the container loading problem with multiple constraints. *Computers and Industrial Engineering*, 108, 149–164.
- Tarantilis, C. D., Zachariadis, E. E., & Kiranoudis, C. T. (2009). A hybrid metaheuristic algorithm for the integrated vehicle routing and three-dimensional container-loading problem. *IEEE Transactions on Intelligent Transportation Systems*, 10(2), 255-271.
- Terno, J., Scheithauer, G., Sommerweiß, U., & Riehme, J. (2000). An efficient approach for the multi-pallet loading problem. *European Journal of Operational Research*, 123(2), 372-381.
- Trivella, A., & Pisinger, D. (2016). The load-balanced multi-dimensional bin-packing problem. *Computers & Operations Research*, 74, 152-164.
- Uner, Ömer. Pratik Kisitli Kutu Paketleme Probleminin Uç Noktalar Temelli Ve Kriter Ağırlıklı Çözümü İçin Sezgisel Model Önerisi. Doktora Tezi, Kocaeli Üniversitesi, 2019.
- Wu, Y., Li, W., Goh, M., de Souza, R. (2010). Three-dimensional bin packing problem with variable bin height. *European Journal of Operational Research*, 202(2), 347– 355.
- Wang L., Guo S., Chen S., Zhu W., Lim A. (2010). Two Natural Heuristics for 3D Packing with Practical Loading Constraints. *PRICAI 2010: Trends in Artificial Intelligence*, 256-267.

- Wäscher G, Hauner H, Schumann H (2007) An improved typology of cutting and packing problems. *Eur J Oper Res* 183(3), 1109–1130.
- Yeung, L. H. W., Tang, W. K. S. (2005). A hybrid genetic approach for container loading in logistics industry. *IEEE Transactions on Industrial Electronics*, 52(2), 617–627.
- Zhao, X., Bennell, J. A., Bektaş, T., & Dowsland, K. (2016). A comparative review of 3D container loading algorithms. *International Transactions in Operational Research*, 23(1-2), 287-320.

Makaleler

1. Erbayrak, S., Özkır, V., Yıldırım, U.M. (2021). Multi-objective 3D bin packing problem with load balance and product family concerns. *Computers and Industrial Engineering*, 159, 107518.