



**PALET YÜKLEME PROBLEMLERİNE BİR KARAR DESTEK MODELİ
ÖNERİSİ**

Sedat HAKYEMEZ

**DOKTORA TEZİ
YÖNETİM BİLİŞİM SİSTEMLERİ ANA BİLİM DALI**

**GAZİ ÜNİVERSİTESİ
BİLİŞİM ENSTİTÜSÜ**

Aralık 2021

Sedat HAKYEMEZ tarafından hazırlanan PALET YÜKLEME PROBLEMLERİNE BİR KARAR DESTEK MODELİ ÖNERİSİ” adlı tez çalışması aşağıdaki jüri tarafından OY BİRLİĞİ ile Gazi Üniversitesi Yönetim Bilişim Sistemleri Ana Bilim Dalında DOKTORA TEZİ olarak kabul edilmiştir.

Danışman: Prof. Dr. Uğur ÖZCAN

Ağaç İşleri Endüstri Mühendisliği Anabilim Dalı Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum.

.....

Başkan: Prof. Dr. Süleyman ERSÖZ

Endüstri Mühendisliği Anabilim Dalı, Kırıkkale Üniversitesi

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum.

.....

Üye: Prof. Dr. Aslıhan TÜFEKÇİ

Yönetim Bilişim Sistemleri Anabilim Dalı, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum.

.....

Üye: Prof. Dr. İsmail KARAOĞLAN

Endüstri Mühendisliği Anabilim Dalı, Konya Teknik Üniversitesi

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum.

.....

Üye: Doç. Dr. Talip KELLEGÖZ

Endüstri Mühendisliği Anabilim Dalı, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum.

.....

Tez Savunma Tarihi: 16/12/2021

Jüri tarafından kabul edilen bu tezin Doktora Tezi olması için gerekli şartları yerine getirdiğini onaylıyorum.

.....

Prof. Dr. Aslıhan TÜFEKÇİ

Bilişim Enstitüsü Müdürü

ETİK BEYAN

Gazi Üniversitesi Bilişim Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmada;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmada yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

Sedat HAKYEMEZ

16/12/2021

PALET YÜKLEME PROBLEMİNE BİR KARAR DESTEK MODELİ ÖNERİSİ

(Doktora Tezi)

Sedat HAKYEMEZ

GAZİ ÜNİVERSİTESİ
BİLİŞİM ENSTİTÜSÜ

Aralık 2021

ÖZET

Son yıllarda işletmeler palet yükleme, yerleşim, kesme ve paketleme ile ilgili faaliyetlerinde mümkün olan en iyi çözümü elde edebilmek için yoğun çaba harcamaktadırlar. Bu sebeple, bu tür problemlere çözüm aranırken yerleşim alanına mümkün olan fazla sayıda nesnenin, en az boşlukla yerleşimi hedeflenmektedir. Kare, dikdörtgen, daire vb. şekillerdeki nesnelerin paletlere veya yerleşim alanlarına katmansız olarak yerleştirilmesine iki boyutlu paketleme problemi denmektedir. Paketlenecek nesnelerin sayısının ve çeşidinin artması, ağırlık ve öncelik sırası, dayanım ve kırılabilirlik gibi ek kısıtlar problemi daha karmaşık hale getirmektedir. Bu karmaşıklık bu tür problemlerin çözümünün kesin çözüm metotlarıyla makul sürelerde çözülememesine neden olmaktadır. Bu sebeple, en iyi veya en iyiye yakın çözümler ancak metasezgisel yöntemlerle mümkündür. Bu çalışmada, daire şeklindeki paletler üzerinde iki boyutlu daire paketleme problemi (DPP) ele alınmıştır. Sabit yarıçaplı paletlerde daire sayısı ve alan maksimizasyonu amaç fonksiyonlarına yönelik özgün bir yerleşim algoritması geliştirilmiş ve genetik algoritma tabanlı bir çözüm modeli önerilmiştir. Önerilen çözüm modelinin performansı, literatürden elde edilen ve bir dizi test örneğinin kullanıldığı başka bir çalışmanın sonuçlarıyla karşılaştırılarak test edilmiştir. Gerçekleştirilen testler sonucunda, önerilen çözüm modelinin karşılaştırılan çalışmaya göre daha iyi performans değerleri elde ettiğini ortaya koymaktadır. Geliştirilen yerleşim algoritması ve önerilen çözüm modelinin kullanılmasıyla, DPP için bir karar destek sistemi geliştirilmiştir. Geliştirilen karar destek sistemi ile daire paketleme problemlerinde verimli ve etkili çözüm arayan karar vericilere karar vermelerinde yardımcı olması amaçlanmaktadır.

Bilim Kodu : 114606
Anahtar Kelimeler : Yönetim bilişim sistemleri, daire paketleme problemleri, karar destek sistemleri, genetik algoritma
Sayfa Adedi : 145
Danışman : Prof. Dr. Uğur ÖZCAN

A DECISION SUPPORT MODEL TO PALLET LOADING PROBLEM

(Ph. D. Thesis)

Sedat HAKYEMEZ

GAZI UNIVERSITY

INFORMATICS INSTITUTE

December 2021

ABSTRACT

In recent years, businesses are making great efforts to obtain the best possible solution in their activities related to pallet loading, placement, cutting and packaging. For this reason, while seeking solutions to such problems, it is aimed to place as many objects as possible in the settlement area with the least amount of space. Square, rectangle, circle etc. placing objects in shapes on pallets or loading areas without layers is called a two-dimensional packing problem. Increasing the number and variety of items to be packed, additional constraints such as weight, priority, strength and fragility complicate the problem. This complexity causes the solution of such problems not to be solved in reasonable time with exact solution methods. For this reason, the best or near-best solutions are only possible with metaheuristic methods. In this study, two-dimensional circle packing problem (CPP) on circular pallets is discussed. A unique loading algorithm has been developed for the number of circles and area maximization objective functions in fixed radius pallets and a genetic algorithm-based solution model has been proposed. The performance of the proposed solution model was tested by comparing it with the results of another study obtained from the literature and using a set of test examples. As a result of the tests carried out, it is revealed that the proposed solution model achieves better performance values than the compared study. A decision support system has been developed for DPP by using the developed layout algorithm and the proposed solution model. With the developed decision support system, it is aimed to help decision makers who are looking for efficient and effective solutions for circles packing problems in their decision making.

Science Code	: 114606
Key Words	: Management information systems, circle packing problems, decision support systems, genetic algorithm
Page Number	: 145
Supervisor	: Prof. Dr. Uğur ÖZCAN

TEŞEKKÜR

Doktora çalışmamın her aşamasında desteğini esirgemeyerek akademik gelişimime büyük katkı sağlayan ve fikirleriyle yol gösterici değerli danışman hocam Prof. Dr. Uğur ÖZCAN'a en içten teşekkürlerimi sunarım.

Bu çalışmanın yürütülmesi sırasında desteklerini daima hissettiğim ve tezin gelişmesine yönelik olumlu ve yapıcı katkıları bulunan değerli hocalarım Prof. Dr. İsmail KARAOĞLAN ve Doç. Dr. Talip KELLEĞÖZ teşekkürlerimi sunarım.

Dualarıyla her zaman yanımda olduklarını hissettiğim, maddi ve manevi desteklerini esirgemeyen, haklarını hiçbir zaman ödeyemeyeceğim sevgili babama ve anneme sonsuz minnet ve teşekkürlerimi sunarım.

Zor ve uzun bir süreç olan doktora çalışmasında sabır ve fedakârlıklarıyla bana destek olan, aşıladığı moral ve motivasyon ile çalışma azmimi artıran sevgili eşim Sevilay'a, varlıkları sayesinde hayatımızın güzelleştiği yavrularım Emir Kuzey ve Azra'ya gösterdikleri sabır ve destekleri için teşekkür ederim.

Son olarak bu süreçte olumlu ve yapıcı yaklaşımlarıyla her zaman yanımda olan Enstitünün değerli yöneticilerine, akademik ve idari personeline teşekkür ederim.

İÇİNDEKİLER

Sayfa

ÖZET	iv
ABSTRACT.....	v
TEŞEKKÜR.....	vi
İÇİNDEKİLER	vii
ÇİZELGELERİN LİSTESİ.....	x
ŞEKİLLERİN LİSTESİ.....	xi
RESİMLERİN LİSTESİ	xiv
SİMGELER VE KISALTMALAR.....	xv
1. GİRİŞ.....	1
2. TEMEL KAVRAMLAR.....	5
2.1. Palet Yükleme ve Paketleme Problemleri.....	5
2.1.1. Paketleme problemlerinin sınıflandırılması	6
2.1.2. Literatür özeti	15
2.2. Daire Paketleme Problemi.....	19
2.2.1. DPP'nin tanımı.....	19
2.2.2. DPP'nin tarihçesi	20
2.2.3. DPP'nin kullanım alanları.....	21
2.2.4. DPP'nin sınıflandırılması.....	24
2.2.5. Literatür özeti	24
2.3. Genetik Algoritma.....	27
2.3.1. Gen ve kromozom	30
2.3.2. Kodlama	30
2.3.3. Popülasyon	32
2.3.4. Uygunluk değeri ve uygunluk fonksiyonu	33
2.3.5. Seçim mekanizmaları	33

	Sayfa
2.3.6. Genetik operatörler.....	35
2.3.7. Durdurma kriteri.....	41
2.3.8. Genetik algoritma çeşitleri	41
2.4. Karar Destek Sistemleri	43
2.4.1. Karar verme ve karar verme süreci	43
2.4.2. Tanım ve kavramlar	44
2.4.3. KDS'nin temel bileşenleri.....	45
2.4.4. KDS'nin özellikleri, faydaları ve kısıtları.....	47
2.4.5. KDS'nin kullanım alanları	48
3. MATERYAL, YÖNTEM ve UYGULAMA.....	51
3.1. Problem Tanımı.....	51
3.1.1. Problemin amaç fonksiyonları ve kısıtları	51
3.1.2. Problemin matematiksel modeli.....	52
3.2. Çalışmanın Amacı.....	58
3.3. Çalışmanın Yöntemi ve Modelin Geliştirilmesi	59
3.3.1. DPP için tasarlanan yerleşim algoritması	59
3.3.2. DPP için geliştirilen genetik algoritma	73
3.3.3. GA bileşenleri için parametre optimizasyonu.....	79
3.4. DPP için Geliştirilen KDS Uygulaması	84
4. ARAŞTIRMA BULGULARI VE TARTIŞMA.....	89
4.1. Deney Ortamının Özellikleri.....	89
4.2. Problem Setlerinin Özellikleri.....	89
4.2.1. Problem seti 1'in özellikleri.....	90
4.2.2. Problem seti 2'nin özellikleri.....	91
4.2.3. Problem seti 3'ün özellikleri	91
4.3. Karşılaştırma Sonuçları.....	92

	Sayfa
4.3.1. Problem seti 1'in sonuçları	92
4.3.2. Problem seti 2'nin sonuçları	96
4.3.3. Problem seti 3'ün sonuçları.....	98
4.4. Sonuçların değerlendirilmesi	99
5. SONUÇ ve ÖNERİLER	101
KAYNAKLAR	105
EKLER.....	115
ÖZGEÇMİŞ	143

ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 2.1. Kesme ve paketleme problemlerinin gruplandırılması.....	7
Çizelge 2.2. Kesme ve paketleme problemlerinin simgesel gösterimi	12
Çizelge 2.3. Üretici palet yükleme problemi ile ilgili çalışmalar	17
Çizelge 2.4. Distribütör palet yükleme problemi ile ilgili çalışmalar.....	18
Çizelge 2.5. Özdeş olmayan daire paketleme ile ilgili çalışmalar	25
Çizelge 2.6. Özdeş daire paketleme ile ilgili çalışmalar	26
Çizelge 3.1. Önerilen yerleşim algoritmasının adımları	63
Çizelge 3.2. $n=6$ için dairelere atanan değerler.....	65
Çizelge 3.3. Parametre optimizasyonu için seçilen test problemleri	80
Çizelge 3.4. Test Sonuçlarına göre belirlenen parametreler	84
Çizelge 4.1. Problem Seti 1'deki test problemleri.....	90
Çizelge 4.2. Problem Seti 2'deki test problemleri.....	91
Çizelge 4.3. Problem Seti 3'deki test problemleri.....	92
Çizelge 4.4. Daire maksimizasyonu karşılaştırma sonuçları	93
Çizelge 4.5. Alan maksimizasyonu karşılaştırma sonuçları	95
Çizelge 4.6. Daire maksimizasyonu karşılaştırma sonuçları	96
Çizelge 4.7. Alan maksimizasyonu karşılaştırma sonuçları	97
Çizelge 4.8. Problem Seti 3 için daire sayısı maksimizasyonu karşılaştırma sonuçları .	98

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 2.1. Tek boyutlu stok kesme problemi.....	8
Şekil 2.2. İki boyutlu dörtgen parça yerleşimi.....	9
Şekil 2.3. İki boyutlu stok kesme.....	9
Şekil 2.4. Üç boyutlu yükleme problemi	10
Şekil 2.5. Üç boyutlu yerleşim probleminin iki boyuta indirgenmesi	10
Şekil 2.6. Kesme ve paketleme problemlerinin gruplandırılması.....	12
Şekil 2.7. Çıktı maksimizasyonu problem tipleri.....	14
Şekil 2.8. Girdi minimizasyonu problem tipleri	15
Şekil 2.9. Özdeş olmayan dairelerin paketlemesi	19
Şekil 2.10. Özdeş dairelerin paketlemesi	19
Şekil 2.11. Malfatti çemberleri	20
Şekil 2.12. Eşkenar üçgende daire paketleme.....	21
Şekil 2.13. Evrimsel algoritma süreçleri.....	28
Şekil 2.14. Genetik algoritma akış şeması	29
Şekil 2.15. İkili kodlama örnekleri	31
Şekil 2.16. Permütasyon kodlama örnekleri	31
Şekil 2.17. Değer kodlama örnekleri	32
Şekil 2.18. Ağaç kodlama gösterimi	32
Şekil 2.19. Turnuva seçimi örneği	34
Şekil 2.20. Rulet tekerleği gösterimi.....	35

Şekil	Sayfa
Şekil 2.21. Tek noktalı çaprazlama yöntemi.....	36
Şekil 2.22. İki noktalı çaprazlama yöntemi	37
Şekil 2.23. Tekdüze çaprazlama yöntemi	37
Şekil 2.24. Sıralı çaprazlamada kesme noktalarının gösterimi	38
Şekil 2.25. Sıralı çaprazlama işlemi ile yavru üretimi	38
Şekil 2.26. KDS'nin temel bileşenleri	46
Şekil 2.27. KDS'nin ilgili olduğu bilim dalları [137]	49
Şekil 3.1. Daire paletin koordinat gösterimi	53
Şekil 3.2. Yerleşik iki daireye üçüncü dairenin teğet konumlandırılması	55
Şekil 3.3. Palete ve yerleşik daireye teğet olacak şekilde daire konumlandırma	56
Şekil 3.4. Palete ve yerleşik daireye teğet olan dairenin koordinat tespiti	56
Şekil 3.5. 3 nolu dairenin yerleşim öncesi durum.....	61
Şekil 3.6. 3 nolu daire yerleşiminde mümkün olan komşuluklar	61
Şekil 3.7. Altı adet daire ve paletin gösterimi.....	64
Şekil 3.8. Daire ve palete ait parametrelerin gösterimi.....	65
Şekil 3.9. Dairelerin ve paletin yerleşim öncesi gösterimi	66
Şekil 3.10. 5 nolu dairenin yerleşimi sonrası gösterim.....	66
Şekil 3.11. 3 nolu dairenin komşuluk seçenekleri ve yerleşiminin gösterimi	67
Şekil 3.12. 6 nolu dairenin yerleşimi öncesi durum.....	68
Şekil 3.13. 6 nolu dairenin yerleşim adımlarının gösterimi.....	68
Şekil 3.14. 1 nolu dairenin yerleşimi öncesi durum.....	69

Şekil	Sayfa
Şekil 3.15. 1 nolu dairenin yerleşim adımlarının gösterimi.....	70
Şekil 3.16. 2 nolu dairenin yerleşimi öncesi durum.....	70
Şekil 3.17. 2 nolu dairenin yerleşim adımlarının gösterimi.....	71
Şekil 3.18. 4 nolu dairenin yerleşimi öncesi durum.....	72
Şekil 3.19. 4 nolu daireye uygun yer tespiti.....	73
Şekil 3.20. Altı dairesel paketleme probleminin çözümü	73
Şekil 3.21. DPP'nin çözümü için önerilen genetik algoritmanın akış şeması.....	74
Şekil 3.22. Genetik algorithmada örnek kromozom gösterimi	75
Şekil 3.23. Kromozomlardan oluşan popülasyonun gösterimi	75
Şekil 3.24. Çaprazlama noktası seçim yöntemi için birinci alternatif	78
Şekil 3.25. Çaprazlama noktası seçim yöntemi için ikinci alternatif.....	78
Şekil 3.26. Seçilen mutasyon yöntemlerinin kromozomda gösterimi	79
Şekil 4.1. $n=40$, $R_0=16.33$ probleminin daire maksimizasyonu karşılaştırması	94
Şekil 4.2. $n=40$, $R_0=16.33$ probleminin alan maksimizasyonu karşılaştırması	95

RESİMLERİN LİSTESİ

Resim	Sayfa
Resim 2.1. Bir coğrafi alanı en az keşişim ve en az boşlukla kapsama problemi	22
Resim 2.2. Fiber optik kabloların dış çapının minimizasyon problemi	22
Resim 2.3. Daire şeklindeki tasarlanan tarım arazileri [69].....	23
Resim 3.1. DPP için KDS arayüzü	85
Resim 3.2. KDS sonuç ekranı	85
Resim 3.3. KDS’de palet deseni gösterimi	86
Resim 3.4. KDS’de yerleşen dairelerin yarıçap ve koordinat verileri.....	87

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler

P_c

Açıklamalar

Çaprazlama oranı

P_m

Mutasyon oranı

P_s

Popülasyon büyüklüğü

Kısaltmalar

Açıklamalar

1PX

Tek noktalı çaprazlama

2PX

İki noktalı çaprazlama

BGA

Basit genetik algoritma

CPU

Merkezi işlem birimi (Centre)

CX

Dairesel çaprazlama

ÇN

Çaprazlama noktası

GA

Genetik algoritma

DPP

Daire paketleme problemi

KDS

Karar destek sistemi

KP

Komşuluk puanı

KDS

Konum seçim değeri

LOX

Doğrusal sıralı çaprazlama

MGA

Melez genetik algoritma

NP-Zor

Deterministik olmayan polinomsal-zor

OBX

Sıraya dayalı çaprazlama

OX

Sıralı çaprazlama

PBX

Pozisyona dayalı çaprazlama

PGA

Paralel genetik algoritma

Kısaltmalar**Açıklamalar****PMX**

Kısmı eşleştirmeli çaprazlama

TGA

Temel genetik algoritma

UX

Tek düze çaprazlama

 μ GA

Mikro genetik algoritma



1. GİRİŞ

Günümüz ticari rekabet ortamında düşük maliyet ve yüksek kârlılık, tüm sektörlerin odaklandığı başlıca hedeflerden biridir. Üretilen malların etkili ve verimli bir şekilde taşınması, işletmelerin maliyetlerini düşürmede ve kârlarını arttırmada önemli bir itici güç olmaktadır. Ayrıca üreticiden tüketiciye kadar olan lojistik faaliyetlerinde bulunan süreçlerin tümü, hasarsız, güvenli ve zamanında gerçekleştirilmesi, ürün kalitesini ve müşteri memnuniyetini arttırmada önemli bir rol oynamaktadır. Bu sebeple sektördeki işletmeler, lojistik faaliyetlerinde bulunan süreçlerde maliyet azaltıcı stratejiler geliştirmek ve faaliyetlerinde sürekliliği sağlamak için yoğun çaba harcamaktadırlar. Özellikle geniş lojistik ağı bulunan işletmelerde, taşınacak malların miktarı ve çeşidinin artmasıyla birlikte etkin, verimli ve optimal düzeyde paketleme, yükleme, taşıma bir zorunluluk halini almaktadır. Bir başka deyişle, malların etkin, verimli paketleme ve yükleme sürecinden geçmesiyle birlikte taşıma ve stoklama faaliyetlerinde maliyet düşüşü sağlanmaktadır. Ayrıca ürünlerin kısıt ve kurallara uygun bir şekilde en az boşlukla paketlenmesi, taşıma ve nakliye sırasındaki salınım hareketlerini azaltmakta ve ürünlerin zarar görmesi önlenmektedir.

Palet yükleme ve paketleme problemleri günlük hayatın birçok alanında karşımıza çıkmaktadır. İnşaat, elektronik, otomotiv, havacılık, denizcilik, savunma, gıda, tarım, ilaç, cam, tekstil endüstrilerinin hammadde tedarikinden müşteriye kadar olan tüm süreçlerinde çok çeşitli paketleme ve palet yükleme problemlerine çözüm aranmaktadır [1,2]. Bu probleme konu olan ürünler daire, kare, dikdörtgen, üçgen şeklinde olabilmektedir. Örneğin; beyaz eşya veya elektronik endüstrisindeki ürünler çoğunlukla kare ve dikdörtgen şeklinde iken; boya, kimya veya meşrubat endüstrisindeki ürünler ise genellikle daire ve kare şeklinde olmaktadır. Dolayısıyla paketlenecek ürünün veya yerleşecek nesnenin geometrik şekli paketleme faaliyetinin sınıflandırmasında ve adlandırılmasında önemli rol oynamaktadır. Bu sebeple dikdörtgen şeklindeki ürünlerin paketlenmesine kutu paketleme, daire şeklindeki ürünlerin paketlenmesi daire paketleme faaliyeti olarak görülmektedir [3].

Boya endüstrisindeki üretilen ürünlerin silindirik şeklindeki kutularla nakliyesi, varillerin depolara dizilmesi, şişe veya kavanozların kutulara yerleşimi, otomotiv endüstrisinde tekerleklerin depolanması gibi yerleşim faaliyetleri daire paketleme ve yerleşimi ile ilgilidir. Metal, cam veya ahşap levhalardan disklerin en az atık olacak şekilde kesilmesi, tarım ve

ağaçlandırmada etkin alan kullanımı, radar sistemlerinin belirli bir coğrafi alanı kaplaması, insansız hava araçlarının etki alanı, ilaç endüstrisindeki tabletlerin paketlenmesi gibi günlük hayatta sıkça karşılaşılan bu türdeki problemler, daire paketleme problemi için geliştirilen teknikler ile çözüme kavuşabilmektedir [4-6].

Palet yüklemede ürünlerin bir palet üzerinde birlikte istiflenerek paketlenmesi beraberinde palet konfigürasyon işlemlerini getirmektedir. Palet konfigürasyonu ise lojistik faaliyet akışının başarıyla sağlanabilmesi kritik rol oynamaktadır [7]. Bu çalışmada, daire paletlere daire nesnelerin istiflenmesini içeren ve daire paketleme problemlerinin çözümüne katkı sunan bir karar destek modeli önerilmiştir. Yerleşecek daire şeklindeki ürünlerin, daire şeklindeki paletin sınırları içinde bulunduğu ve yerleşim sırasında dairelerin birbiriyle çakışmayacak şekilde konumlandığı özgün bir yerleşim algoritması geliştirilmiştir. Geliştirilen bu yerleşim algoritması, genetik algoritma operatörleriyle birlikte çalıştırılarak palet üzerine yerleşecek olan daire sayısının ve dairelerin palet üzerinde kapladığı alanın maksimize edildiği bir karar destek sistemi geliştirilmiştir.

Bu kapsamda, çalışmanın ikinci bölümünde palet yükleme ve paketleme problemleri, daire paketleme problemi, karar destek sistemi ve genetik algoritma ile ilgili temel kavramlar üzerinde durulmuştur. Literatürde karşılaşılan palet yükleme ve paketleme problemleri, sınıflandırma şekilleri ve literatürü hakkında detaylı bilgiler verilmiştir. Paketleme problemleri türlerinden biri olan daire paketleme problemi (DPP) kapsamlı bir şekilde ele alınmıştır. Problemin tanımı, tarihçesi, kullanım alanları, sınıflandırılması ile ilgili detaylar verilmiştir. Daire paketleme probleminin çözüm modelinde kullanılan genetik algoritmanın yapısı, çalışma prensipleri ve çeşitleri hakkında bilgiler sunulmuştur. Karar destek sistemleri, bileşenleri, özellikleri ve kullanım alanları incelenmiştir.

Çalışmanın üçüncü bölümünde, daire palet üzerine dairelerin yüklemesi problemini ele alan daire paketleme probleminin amaç fonksiyonları ve kısıtları, matematiksel modeli üzerinde durulmuştur. Çalışmanın amacı ve yöntemi ayrıntılı bir şekilde açıklanmıştır. Bu çalışma kapsamında tasarlanan yerleşim algoritması ve geliştirilen genetik algoritma (GA) tabanlı karar destek modelinin detayları adım adım anlatılmıştır. Ayrıca geliştirilen genetik algorithmada kullanılacak olan parametrelerin belirlenebilmesi için bir parametre optimizasyon çalışması yapılmış ve GA'nın kalitesini ve başarısını arttıracak şekilde parametreler belirlenmiştir.

Çalışmanın dördüncü bölümünde, literatürden elde edilen ve 2016 yılına ait çalışmadaki üç farklı problem setinde bulunan toplam 36 test örneği, tasarlanan yerleşim algoritması ve geliştirilen GA ile çözülmüştür. Daire sayısı ve alan maksimizasyonu amaç fonksiyonlarına göre çözülen problemlerden elde edilen sonuçlar, ilgili çalışmadaki sonuçlarla karşılaştırılarak değerlendirmelerde bulunulmuştur.

Son olarak beşinci bölümde ise çalışmadan elde edilen sonuçlar ele alınmış ve literatüre katkısından bahsedilmiştir. DPP'nin çözümü için önerilen karar destek sisteminin faydaları açıklanmıştır. Ayrıca, çalışma ile ilgili gelecek çalışma önerilerine yer verilmiştir.





2. TEMEL KAVRAMLAR

Bu bölümde, palet yükleme ve paketleme problemleri, karar destek sistemleri ve genetik algoritma ile ilgili temel kavramlardan bahsedilmektedir. İlk olarak paketleme problemleri ile ilgili kavramlar açıklanmış ve problemin sınıflandırması ile ilgili literatürden örneklere yer verilmiştir. Palet yükleme problemine bir KDS geliştirilmesi sebebiyle KDS'nin tanımı, bileşenleri, özellikleri ve kullanım alanları üzerinde durulmuştur. Son olarak, geliştirilen çözüm modelinde metasezgisel algoritmalarından GA'nın kullanılması sebebiyle algoritmanın alt yapısı, mekanizmaları, operatörleri ve çeşitleri detaylı incelenmiştir.

2.1. Palet Yükleme ve Paketleme Problemleri

Günlük yaşamamızın her alanında sıklıkla karşılaştığımız problemlerden biri yükleme ya da paketleme problemleridir. Öyle ki evimizi bir yerden başka bir yere taşıırken araçlara en fazla eşyayı yükleme hedefi, farklı çaptaki kavanozların bir kutuya paketlenmesi, depoların düzenlenmesi gibi faaliyetler çoğu insanın günlük yaşamında çözmeye çalıştığı problemlerdendir.

Bir tesisin içinde bulunan bölümlerin verimi arttıracak şekilde konumlandırılması, tekstil atölyesindeki kumaş balyalarından en az atıkla en fazla ürünün elde edilmesi, dergi veya gazetelere haber, ilan ve reklamların yerleştirilmesi, ürünlerin palet, konteyner, kutu vb. içerisine en az boşlukla yüklenmesi gibi birçok problem yükleme ve yerleşim problemi örneği olarak karşımıza çıkmaktadır. Yükleme ve yerleşim problemlerinin ortak odak noktası kaynakları verimli kullanarak firenin ve atığın azaltılması, yerleşim için harcanan süresinin kısaltılması ve iş gücü kaynağının etkili kullanımı gibi unsurlardır. Bu tür problemlerin optimal çözüme kavuşturulması sonucunda işletmelerde maliyet düşüşü, kâr oranlarında ise artış olmaktadır.

Palet yükleme problemi kısıtlı kaynaklardan amacına yönelik en büyük faydayı elde etmek şeklinde ele alındığında, araç ve konteyner yükleme, paketleme, kutu istifleme, yerleşim, tesis planlama, sırt çantası yükleme, kesme ve paketleme problemleriyle aynı mantıksal yapıya sahip kendine has karakteristiği olan problemlerdir [8]. Bu tür problemlerin temelinde büyük parça olarak yerleşim alanı ya da kesim alanı, küçük parça olarak ise yerleşecek ya da kesilecek küçük nesneler bulunmaktadır. Bir büyük parça, birkaç küçük

parçaya bölünmesiyle problem, kesme problemi haline alırken, küçük nesnelerin, büyük parçanın içinde belirli bir düzen içinde dizilmesi de paketleme veya yükleme problemi haline dönüşmektedir [9]. Bir üretim tesisinin kurulum aşamasında birimlerin birbirleriyle verimli çalışacak şekilde konumlandırılması veya palet içine ürünlerin belirli kısıtlar göz önünde bulundurularak dizilmesi bir yerleşim problemidir. Hatta yerleşim problemlerine yazılım/oyun sektöründe de rastlanmak mümkündür. Tetris oyunu, iki boyutlu oyun alanındaki blokların, en az boşlukla dizilimin hedeflendiği tipik bir yerleşim problemidir. Bilgisayarda veya akıllı cihazlarda oynadığımız satrancın temelinde de yerleşim problemi çözümü bulunmaktadır.

2.1.1. Paketleme problemlerinin sınıflandırılması

Literatüre göre paketleme problemleri, kesme ve paketleme problemleri sınıfı içinde yer almaktadır. Kesme problemlerinde, büyük parçadan birden çok küçük parçanın elde edilmesi hedeflenirken, paketleme problemlerinde ise küçük nesnelerin büyük parçalara veya alanlara uygun bir şekilde yerleştirilmesi hedeflenmektedir [3].

Kesme ve paketleme problemleri ile ilgili sınıflandırma gösterim şekli Dyckhoff [10] tarafından geliştirilmiştir. Geliştirilen bu sınıflandırmada, kesme ve paketleme problemlerini boyutluluk, atama tipi, büyük parça seçimi, küçük nesnelerin seçimi olmak üzere dört temel özelliğe göre kategorize edilmektedir. Dyckhoff [10]'un yaptığı sınıflandırmadan sonra Wäscher vd. [11], sınıflandırmanın iyileştirildiği başka bir çalışma yaparak kesme ve paketleme problemleri için farklı bir sınıflandırma yöntemini literatüre kazandırmıştır. Belirtilen iki sınıflandırmanın ayrıntıları şu şekilde açıklanmıştır:

Dyckhoff sınıflandırması

1990 yılında yapılan bu sınıflandırmada boyutluluk, atama tipi, büyük parça seçimi, küçük nesnelerin seçimi olarak dört temel özellik üzerinde kesme ve paketleme problemlerini sınıflandırmaktadır. Çizelge 2.1'de gösterildiği şekliyle ifade edilen sınıflandırmada, kesme ve paketleme problemleri için 96 farklı tür ($4 \times 2 \times 3 \times 4$) problemin gösterimi yapılabilmektedir. Dyckhoff'a göre bu sınıflandırmadaki gösterim şekli dörtlü grupta şeklidir. Örneğin, 2/B/O/C şeklindeki bir gösterimin açıklaması iki boyutlu benzer boyuttaki küçük nesnelerin bir büyük parçaya yerleştirilmesi şeklinde ifade edilmektedir.

Bir başka örnekte ise benzer boyutta belirli bir sayıdaki üç boyutlu kutunun paletlere yerleştirilmesi problemi ise 3/V/I/R olarak gösterilmektedir.

Çizelge 2.1. Kesme ve paketleme problemlerinin gruplandırılması

1. Boyutluluk	
(1)	Bir boyutlu
(2)	İki boyutlu
(3)	Üç boyutlu
(N)	N boyutlu ($N > 3$)
2. Atama tipi	
(B)	Büyük parçaların tümü ve küçük nesnelerin seçimi
(V)	Büyük parçaların seçimi ve küçük nesnelerin tümü
3. Büyük parça çeşidi	
(O)	Bir büyük parça
(I)	Aynı ebatta büyük parça
(D)	Farklı ebatta büyük parça
4. Küçük nesne çeşidi	
(F)	Farklı boyutta az sayıda küçük nesneler
(M)	Farklı boyutta çok sayıda küçük nesneler
(R)	Benzer boyutta çok sayıda küçük nesneler
(C)	Benzer boyutta küçük nesneler

Çizelge 2.1’de belirtilen kesme ve paketleme problemi için yapılan sınıflandırmada bulunan boyutluluk, atama tipi, büyük parça çeşidi, küçük nesne çeşidi şu şekilde açıklanmıştır:

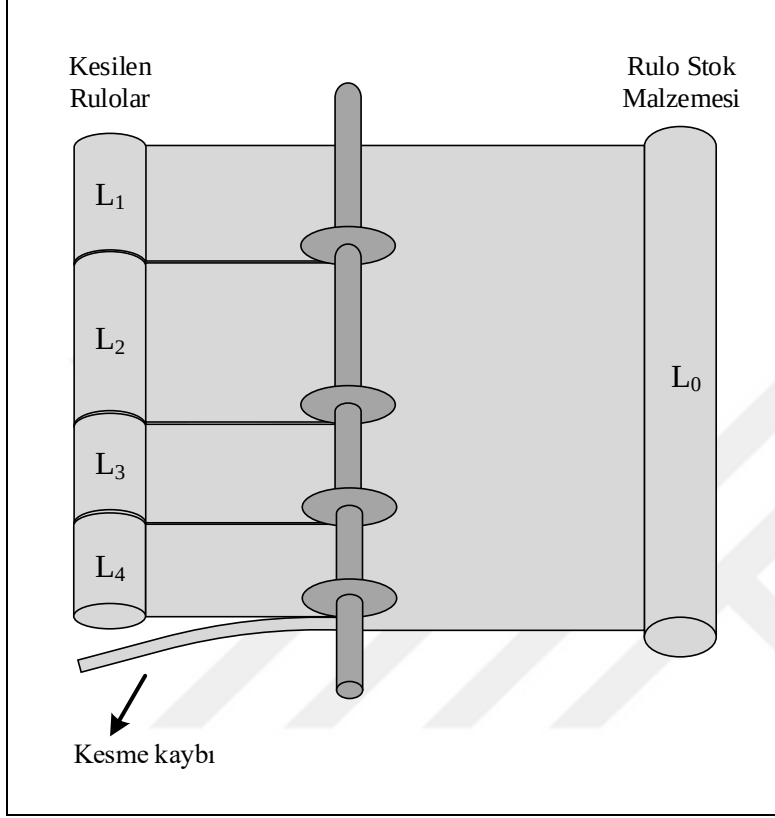
Boyutluluk

Kesme ve paketleme problemlerinde, küçük nesnelerin ve büyük parçanın geometrik boyut sayısını belirtmektedir. Bir, iki, üç ve daha fazla boyutlu olmak üzere sınıflandırılır.

Tek boyutluluk

Şekil 2.1’de gösterilen L_0 uzunluğundaki stok malzemesinden L_1, L_2, L_3, L_4 uzunluğundaki ruloların kesilerek elde edilmesi gösterilmektedir. Kağıt rulo veya demir çubuk

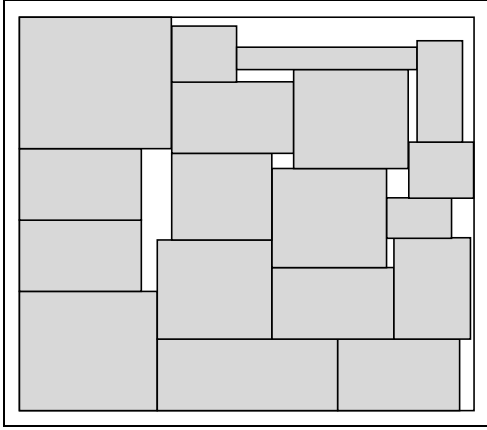
malzemelerin kesimi tek boyutlu kesme problem sınıfına örnektir. Bu problem sınıfında amaç, önceden belirlenmiş uzunluk taleplerine göre kesme kaybını veya kullanılacak stok malzeme miktarını en aza indirecek şekilde kesilmesidir.



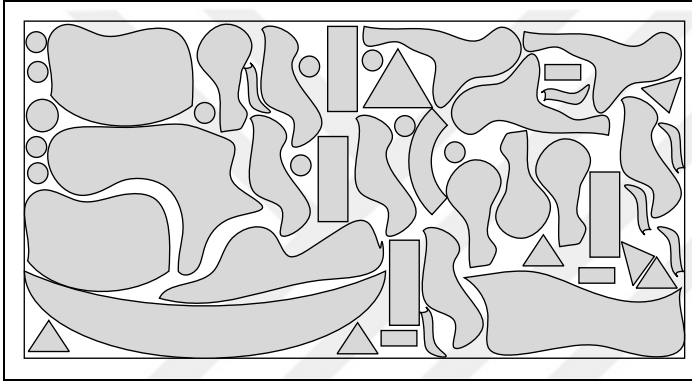
Şekil 2.1. Tek boyutlu stok kesme problemi

İki boyutluluk

Metal sac, kumaş, deri, ahşap, cam levha gibi ana malzemelerden küçük parçalar kesilmesi veya küçük parçaların başka bir ana malzemeye ya da bir alana yerleştirilmesi iki boyutlu problem sınıfındadır. Bu problem sınıfında ana hedef, kullanılan levha sayısını veya levha üzerindeki atık miktarını en aza indirmek olmaktadır. Ayrıca levhalar üzerinde kesilen küçük parçaların sayısının en büyüklenmesi de amaçlar arasındadır. Mobilya yapımı için kullanılan ahşap suntalardan küçük parçaların kesimi, gazete ve dergilere haber veya reklamların yerleştirilmesi, palet üzerine kutuların yüksekliklerine bakılmaksızın tek sıra halinde dizilmesi Şekil 2.2’de de gösterildiği iki boyutlu problem sınıfına girmektedir. Ayrıca tekstil sektöründe deri, kumaş tabakalarından kıyafet, ayakkabı yapımı için gerekli parçaların Şekil 2.3’de gösterildiği üzere en az atıkla kesilmesi gibi birçok örnek iki boyutlu problemlerdendir.



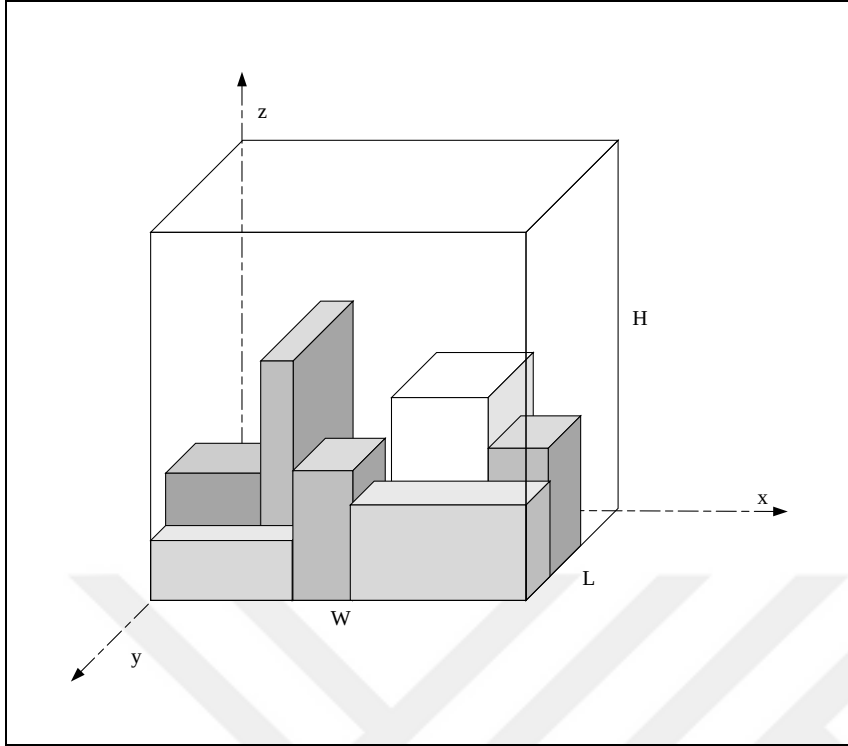
Şekil 2.2. İki boyutlu dörtgen parça yerleşimi



Şekil 2.3. İki boyutlu stok kesme

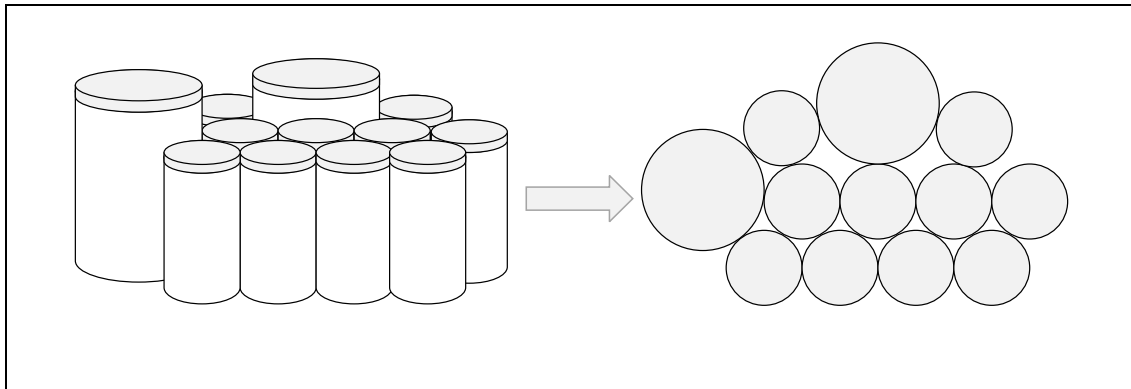
Üç boyutluluk

Üç boyutlu yükleme problemi, Şekil 2.4’de gösterildiği üzere n adet küçük nesnenin en (w), boy (l), yükseklik (h) değerleri dikkate alınarak daha büyük hacimli (W, L, H) kutu ya da konteynere taşma olmayacak şekilde yerleştirilmesi problemidir. Bu problem sınıfında sayısı ve hacmi belirli büyük kutulara, en fazla sayıdaki küçük nesnenin yerleşimi, maksimizasyon problemi iken, küçük nesnelerin tamamının mümkün olan en az sayıdaki büyük kutulara yerleştirilmesi ise minimizasyon problemi halini almaktadır. Ayrıca problem, yerleşim sırasındaki belirli kısıtlar sebebiyle farklı tür probleme dönüşebilmektedir. Örneğin, küçük nesnelerin ağırlıkları, hacimleri ve faydaları dikkate alınarak yerleşimin yapıldığı düşünüldüğünde, problem toplam faydanın maksimize edildiği bir sırt çantası problemi halini almaktadır.



Şekil 2.4. Üç boyutlu yükleme problemi

Üç boyutlu yükleme problemlerinin gerçek hayattaki uygulamaları iki boyuta indirgenebilmektedir [12]. Yerleştirilecek nesnelerin taban kesitlerinin geometrik şekline göre paletlere yerleşiminin yapıldığı durumlarda yerleşen nesnelerin oluşan desen yerleşim durumunu göstermektedir. Örneğin, Şekil 2.5’de gösterildiği üzere farklı taban çapları bulunan kavanozların yerleştirilmesi problemini, iki boyuta çevirerek iki boyutlu yerleşim problemi haline getirmek mümkün olmaktadır. Burada dikkat edilmesi gereken husus, bu tür yerleşim problemlerinde yerleşimi yapılacak olan nesnelerin katmansız olarak tek sıra halinde yerleştirilmesinin gerektiğidir.



Şekil 2.5. Üç boyutlu yerleşim probleminin iki boyuta indirgenmesi

Atama tipi

Küçük nesnelerin büyük parçalara atanması olarak, B ve V ile gösterilen iki tür belirtilmiştir [10].

- B, tüm büyük parçaların kullanılması gerektiği anlamına gelen Almanca "Beladeproblem" kelimesinden gelmektedir.
- V, tüm küçük nesnelerin bir dizi büyük parçaya yüklenmesi gereken bir durumu belirtir ve Almanca "Verladeproblem" kelimesinden gelmektedir.

Büyük parça çeşidi seçimi

Büyük parçaların çeşidini ifade etmektedir.

- O, tek bir büyük parçayı,
- I, birden fazla aynı boyuttaki benzer büyük parçayı,
- D, birden fazla farklı büyük parçayı temsil etmektedir.

Küçük nesne çeşidi seçimi

Bu ölçüt küçük nesnelerin çeşitlerinin gösterimini ifade etmektedir.

- F, farklı boyutta az sayıda küçük nesne (Örneğin, 10 kutunun araca yüklemesi),
- M, farklı boyutta çok sayıda küçük nesne (Örneğin, [0,1] birim uzunluğundaki yüzlerce farklı boyuttaki çubukların, 1 birim uzunluğundaki kutulara yerleştirilmesi)
- R, birbirine benzer çok sayıda küçük nesne, (Örneğin, yaklaşık 50 farklı küçük nesnenin, toplam 1000 adet kesilmesi)
- C, tümü birbiriyle benzer veya uyumlu küçük nesneler (Örneğin, palet yükleme)

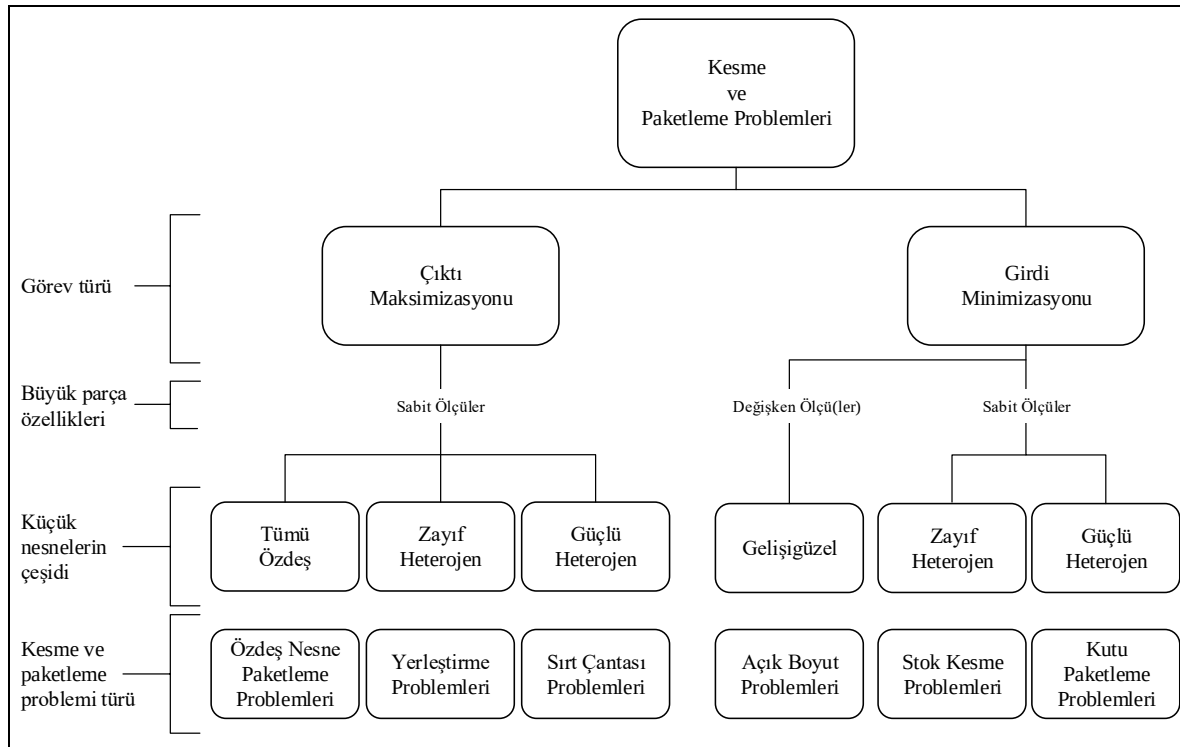
Dyckhoff'un yaptığı bu sınıflandırmaya göre, birçok farklı kesme ve paketleme probleminin gösterimi Çizelge 2.2'de örnek olarak verilmiştir.

Çizelge 2.2. Kesme ve paketleme problemlerinin simgesel gösterimi

Problem Türü	Simgesel Gösterim
Klasik Sırt Çantası Problemi	1 / B / O /
Palet Yükleme Problemi	2 / B / O / C
Çok boyutlu Sırt Çantası Problemi	/ B / O /
Araç yükleme Problemi	1 / V / I / F veya 1 / V / I / M
Konteyner Yükleme Problemi	3 / V / I / veya 3 / B / O /
Klasik Kutu Paketleme Problemi	1 / V / I / M
Klasik Stok Kesme Problemi	1 / V / I / R
2 boyutlu kutu paketleme Problemi	2 / V / D / M
Montaj Hattı Dengeleme Problemi	1 / V / I / M
Çok işlemcili zamanlama Problemi	1 / V / I / M
Bellek Ayırma/Tahsis Problemi	1 / V / I / M

Wäscher vd.'nin sınıflandırması

2007 yılında Wäscher vd. [11] tarafından yapılan çalışma, Dyckhoff'un kesme ve paketleme problemlerinin sınıflandırılmasının iyileştirildiği ve Şekil 2.6'da gösterildiği gibi problemin çıktı maksimizasyonu ve girdi minimizasyonu olmak üzere iki farklı amaç fonksiyonuna göre gruplandırıldığı bir sınıflandırma türüdür.



Şekil 2.6. Kesme ve paketleme problemlerinin gruplandırılması

Wäscher vd. [11], kesme ve paketleme problemlerinin aşağıda belirtilen beş temel kritere göre sınıflandırılmıştır. Bu kriterler şu şekildedir:

- Boyutluluk,
- Görev türü,
- Küçük nesnelerin çeşitliliği,
- Büyük parça çeşitliliği,
- Küçük nesnelerin şekilleri,

Boyutluluk

Dyckhoff sınıflandırmasında olduğu gibi problem bir, iki, üç, ve daha fazla boyutlu olmak üzere sınıflandırılır.

Görev türü

Sınıflandırmada problemin amaç fonksiyonu olarak nitelendirilen iki farklı görev türü bulunmaktadır. Çıktı maksimizasyonu ve girdi minimizasyonu olarak adlandırılan amaç fonksiyonlarının açıklamaları şu şekildedir:

Çıktı maksimizasyonu

Çıktı maksimizasyonu türündeki problemlerin ortak odak noktası, büyük parçalar içine küçük nesnelerin tamamının yerleştirilmesinin mümkün olamamasıdır. Bu sebeple büyük parçanın içine yerleştirilen küçük nesnelerin uygun seçimlerle beraber sayısının maksimize edilmesi gerekir. Diğer bir deyişle, bu gruptaki problemler, büyük parça içine en çok sayıdaki küçük nesnenin yüklenmesinin amaçlandığı problemlerdir. Örneğin, palet yükleme ve paketleme problemlerinde üretici ve distribütör palet yükleme problemleri çıktı maksimizasyonu problem tipine örnek verilebilir. Her iki problem tipi paletlere en fazla nesnenin yüklenmesinin veya paketlenmesinin amaçlandığı problemlerdir. Yüklenen nesnenin özdeş ya da zayıf ve güçlü heterojen olması durumuna göre farklılaşmaktadır. Üretici palet yükleme problemlerinde yüklenen nesneler özdeş olduğu için özdeş nesne paketleme problemi (IPPP) sınıfında bulunmaktadır. Distribütör palet yükleme problemlerinde ise yüklenen nesnelerin özdeş olmaması durumunda ise heterojenlik derecesine göre büyük parça yerleşim probleminde SLOOP, MILOPP, MHLOPP; sırt

çantası probleminde ise SKP, MIKP, MHKP tipinde çeşitlendirilmektedir. Çıktı maksimizasyonu problem tipleri Şekil 2.7’de gösterilmektedir.

büyük parça (yerleşim alanı) özellikleri		yerleşecek küçük nesne çeşitliği	özdeş	zayıf heterojen	güçlü heterojen
tüm boyutlar sabit	bir büyük parça		Özdeş Nesne Paketleme Problemi (IIPP)	Tekli Büyük Parça Yerleşim Problemi (SLOPP)	Tekli Sırt Çantası Problemi (SKP)
	özdeş			Çoklu Özdeş Büyük Parça Yerleşim Problemi (MILOPP)	Çoklu Özdeş Sırt Çantası Problemi (MIKP)
	heterojen			Çoklu Heterojen Büyük Parça Yerleşim Problemi (MHLOPP)	Çoklu Heterojen Sırt Çantası Problemi (MHKP)

Şekil 2.7. Çıktı maksimizasyonu problem tipleri

Girdi minimizasyonu

Girdi minimizasyonu problemlerinde, küçük nesnelerin tamamının yerleşebileceği en uygun büyük parçayı temin etmek gerekmektedir. Küçük nesnelerin tamamı eksiksiz yerleşeceğinden dolayı aralarında seçim sorunu yaşanmamaktadır. Ancak küçük nesnelerin tamamının yerleşebildiği büyük parçanın mümkün olabilecek en küçük ölçülerde olması amaçlanmaktadır. Bir başka ifade ile bu gruptaki problemler, belirli sayıdaki küçük nesnenin, olabilecek en az boyutlu büyük parçaya yüklenmesinin amaçlandığı problemlerdir. Girdi minimizasyonu problem tipleri Şekil 2.8’de gösterilmektedir.

büyük parça (yerleşim alanı) özellikleri yerleşecek küçük nesnelerin çeşitliği		zayıf heterojen	güçlü heterojen
tüm boyutlar sabit	özdeş	Tek Stok Boyutlu Stok Kesme Problemi (SSSCSP)	Tek Kutu Boyutlu Kutu Paketleme Problemi (SBSBPP)
	zayıf heterojen	Çoklu Stok Boyutlu Stok Kesme Problemi (MSSCSP)	Çoklu Kutu Boyutlu Kutu Paketleme Problemi (MBSBPP)
	güçlü heterojen	Artık Stok Kesme Problemi (RCSP)	Artık Kutu Paketleme Problemi (RBPP)
bir değişken boyutlu büyük parça		Açık Boyut Problemi (ODP)	

Şekil 2.8. Girdi minimizasyonu problem tipleri

Palet yükleme ve paketleme problemleri kompleks yapısı sebebiyle en iyi çözümün üretilmesinin zor olduğu problemlerdendir. Örneğin, küçük parçaların sayısı, boyutu, ağırlığı, paketlenme sırası gibi çeşitli kriterlerler problemdeki bu zorluğu daha da arttırmaktadır. Bu sebeple, bu tür problemleri NP-Zor problemler sınıfına dahil etmektedir [13]. Bununla birlikte üç boyutlu kutu paketleme problemleri (3D-BPP), iyi bilinen tek boyutlu kutu paketleme probleminin (1D-BPP) bir genelleştirilmesi olduğu için, güçlü bir şekilde NP-zor problem olarak kabul edilir [14]. Dolayısıyla, bu tür problemlerin kesin yöntemlerle kabul edilebilir zamanlarda çözülmesi zordur. Bu sebeple, geliştirilecek olan sezgiseller NP-Zor sınıfındaki problemlerin çözümü için büyük önem taşımaktadır.

2.1.2. Literatür özeti

Son 50 yılda palet yükleme problemleriyle ilgili birçok çalışma ortaya konmuştur. Palet yükleme problemi ile ilgili çalışmalar, literatürde üretici palet yükleme ve distribütör palet yükleme problemi olarak ikiye ayrılmaktadır. Üretici palet yükleme problemi, bir palete en fazla sayıda özdeş (uzunlukları, genişlikleri ve yükseklikleri aynı) kutunun yüklenerek

üreticiden bir dağıtım merkezine yapılan gönderileri değerlendiren problem şeklinde tanımlanırken, distribütör palet yükleme problemi ise bir palete en fazla olacak şekilde özdeş olmayan kutuların yerleştirilerek dağıtım merkezinden perakendecilere yapılan gönderiler şeklinde tanımlanmaktadır. Bu tanımlamada Dyckhoff [10]'un ve Wäscher vd. [11]'in sınıflandırmasında belirtilen küçük nesnelerin kendi aralarında homojen ya da heterojen olma durumu etkili olmaktadır.

Palet yükleme problemleri ile ilgili çalışmaları geniş çapta inceleyen 1992 yılında Ram [15], 2016 yılında Silva vd. [12] ve Kellegöz vd. [16]'nin palet yükleme problemleri ile ilgili literatür çalışmaları bulunmaktadır. Wäscher vd. [11] çalışmalarında konu ile ilgili 23 literatür çalışmasının olduğunu bildirmişlerdir. Üretici palet yükleme problemi literatürde birçok farklı yaklaşımla ele alınmıştır. Palet yükleme problemi NP-Zor kombinatoriyel problemler sınıfına girmektedir [13]. Bu sebeple, problemi makul süreler içinde en iyi veya en iyiye yakın çözümleri elde edebilmek için kesin çözüm yaklaşımının yanında sezgisel çözüm yöntemleri de kullanılarak problem ile ilgili çok sayıda çalışma yapılmıştır. Palet yükleme problemi ilk defa bu isimle 1982 yılında Hodgson [13] tarafından ele alınmış ve üretici palet yükleme problemi ve distribütör palet yükleme problemi olarak iki sınıfa ayrılmıştır.

Çizelge 2.3'de üretici palet yükleme problemi ile ilgili, Dyckhoff [10] ve Wäscher vd. [11]'in sınıflandırma gösterimine ve önerilen çözüm yöntemlerine yer verilmiştir. Gösterilen çalışmalarda problemler, üretici palet yükleme problemleri ile ilgili çalışmalar olması sebebiyle bütün küçük parçaların homojen yapıda olduğu ve bir büyük palete en fazla sayıda küçük nesnenin yerleştirildiği problemlerdir. Bu sebeple, Dyckhoff sınıflandırmasına göre çalışmalar 2/B/O/C ve Wäscher vd.'nin sınıflandırmasında ise özdeş nesne paketle problemler olarak IIPP şeklinde sınıflandırılmıştır. Üretici palet yükleme problemi 3-boyutlu olmasına rağmen, problem palet üzerine yüklemelerin katmanlar halinde yapılarak çözülebildiği için 2-boyutludur. Üretici palet yükleme probleminde dikdörtgen prizma şeklindeki kutular yerine silindir şeklindeki nesneler yer alırsa problem daire paketleme problemine dönüşmektedir. Problem 3-boyutlu olarak ele alınırsa problem özdeş kutulu konteynır yükleme, özdeş parça palet yükleme problemine dönüşmektedir. Bu sebeple, sınıflandırmasına göre çalışmalar 3/B/O/C ve IIPP şeklinde olmaktadır [16].

Çizelge 2.3. Üretici palet yükleme problemi ile ilgili çalışmalar

Yazar(lar), Yıl	Sınıflandırma Gösterimi		Yöntem	Açıklama
	Dyckhoff	Wäscher vd.		
Steudel [17], (1979)	2/B/O/C	IIPP	Sezgisel	4-blok sezgiseli
Smith ve De Cani [18], (1980)	2/B/O/C	IIPP	Sezgisel	Blok sezgiseli
Bischoff ve Dowsland [19], (1982)	2/B/O/C	IIPP	Sezgisel	5-blok sezgiseli
Steudel [20], (1984)	2/B/O/C	IIPP	Sezgisel	Blok sezgisel
Beasley [21], (1985)	2/B/O/C	IIPP	Kesin	0-1 matematiksel model
Carpenter ve Dowsland [22], (1985)	3/B/O/C	IIPP	Sezgisel	Yığınların stabilitesi ve kenetlenilebilirliği için bir teknik
Dowsland [23], (1987)	2/B/O/C	IIPP	Kesin	Graf teorisi
Dowsland [24], (1993)	2/B/O/C	IIPP	Sezgisel	Tavlama benzetimi
Arenales ve Morabito [25], (1995)	2/B/O/C	IIPP	Kesin	Ve/veya graf metodu
Scheithauer ve Terno [26], (1996)	2/B/O/C	IIPP	Sezgisel	G4 sezgiseli
Herbert ve Dowsland [27], (1996)	2/B/O/C	IIPP	Sezgisel	Genetik algoritma
Liu ve Hsiao [28], (1997)	3/B/O/C	IIPP	Sezgisel	Palet yükleme metodu
Bhattacharya ve Roy [29], (1998)	2/B/O/C	IIPP	Kesin	Ağaç arama algoritması
Morabito ve Morales [30], (1998)	2/B/O/C	IIPP	Sezgisel	Özyinelemeli 5-blok sezgiseli
Scheithauer ve Sommerweiß [31], (1998)	2/B/O/C	IIPP	Sezgisel	4-blok sezgiseli
Young-Gun ve Maing-Kyu [32], (2001)	2/B/O/C	IIPP	Sezgisel	Özyineli 5-blok sezgiseli
Amaral ve Wright [33], (2001)	2/B/O/C	IIPP	Sezgisel	Tabu arama algoritması
Lins vd. [34], (2003)	2/B/O/C	IIPP	Sezgisel	L-şekilli blok sezgiseli
Alvarez-Valdes vd. [35], (2005)	2/B/O/C	IIPP	Kesin	Dal ve kesme algoritması
Alvarez-Valdes vd. [36], (2005)	2/B/O/C	IIPP	Sezgisel	Tabu arama algoritması
Birgin vd. [37], (2005)	2/B/O/C	IIPP	Sezgisel	L-şekilli blok sezgiseli
Pureza ve Morabito [38], (2006)	2/B/O/C	IIPP	Sezgisel	Tabu arama algoritması
Yi vd. [39], (2009)	2/B/O/C	IIPP	Sezgisel	Pervane yükleme sezgiseli
Birgin vd. [40], (2010)	2/B/O/C	IIPP	Sezgisel	Özyineli bölümleme algoritması
Kocjan ve Holmström [41], (2010)	3/B/O/C	IIPP	Kesin	Tamsayılı programlama
de Querioz vd. [42], (2014)	2/B/O/C	IIPP	Sezgisel	L-şekilli blok sezgiseli
Lu ve Cha [43], (2014)	2/B/O/C	IIPP	Sezgisel	ERI metodu
Ahn vd. [44], (2015)	2/B/O/C	IIPP	Kesin	Geliştirilmiş en iyi ilk dal ve sınır algoritması
Ahn vd. [45], (2015)	2/B/O/C	IIPP	Kesin	En iyi ilk dal ve sınır algoritması

Distribütör palet yükleme problemi, bir palete en fazla olacak şekilde özdeş olmayan nesnelerin yerleştirilmesidir. Üretici palet yükleme problemi üzerine yapılan çalışmalarda gibi 2-boyutlu ve 3-boyutlu palet yükleme esasına dayanmaktadır. 3-boyutlu palet yükleme konusunda yapılan ilk çalışma Hodgson [13]'e aittir. Distribütör palet yükleme problemini Wäscher vd. [11] SLOPP sınıfına koymaktadır ve çalışmasında SLOPP üzerine 56 adet çalışmanın olduğunu bildirilmiştir. Bu çalışmaların tamamı sadece distribütör palet yükleme problemine değil sırt çantası, kalıp yerleşimi problemi, 2-boyutlu kesme problemi ve tekli konteyner yükleme problemi, özdeş olmayan daire paketleme problemi şeklinde sıralanabilecek problemlere de aittir [16].

Çizelge 2.4. Distribütör palet yükleme problemi ile ilgili çalışmalar

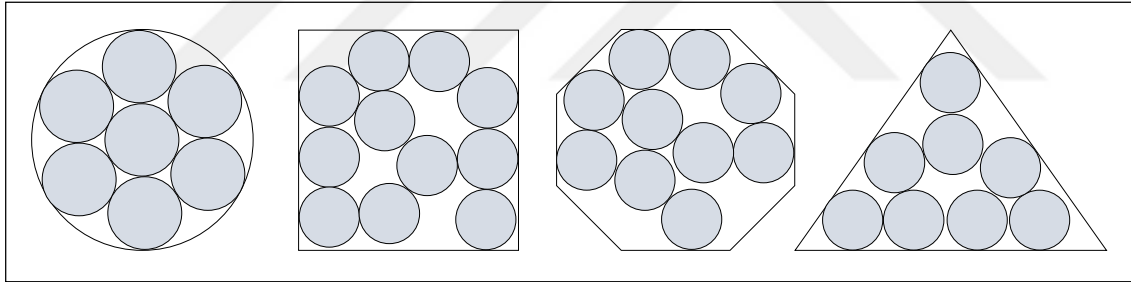
Yazar(lar), Yıl	Sınıflandırma Gösterimi		Yöntem	Açıklama
	Dyckhoff	Wäscher vd.		
Hodgson [13], (1982)	3/B/O/M	SKP	Sezgisel	Dinamik programlama ve sezgisel yöntemlerin kombinasyonu
Chen vd. [46], (1991)	3/B/O/R	SLOPP	Kesin	0-1 tamsayı programlama modeli
Tsai vd. [47], (1993)	3/B/O/M	SKP	Kesin	0-1 tamsayı programlama modeli
Abdou ve Yang [48], (1994)	3/B/O/M	SKP	Sezgisel	Değiştirilmiş lineer denklem yaklaşımı
Abdou ve Yang [49], (1995)	3/B/O/M	SKP	Kesin	Tamsayı Doğrusal programlama
Bischoff vd.[50], (1995)	3/B/O/R	SLOPP	Sezgisel	Katmanlama algoritması
Arghavani ve Abdou [51], (1996)	3/B/O/R	SLOPP	Kesin	Tamsayı Doğrusal programlama
Abdou ve Arghavani [52], (1997)	3/B/O/M	SKP	Kesin	Tamsayı Doğrusal programlama
Abdou ve Elmasry [53], (1999)	3/B/O/R	SLOPP	Sezgisel	Rasgelelik yaklaşımı
Furuholmen vd. [54], (2009)	3/B/O/R	SLOPP	Sezgisel	Gene dayalı Programlama ve İşbirliğine Dayalı Evrim
Ballew [55], (2000)	3/B/O/R	SLOPP	Kesin	0-1 doğrusal olmayan model
Baltacıoğlu vd. [56], (2006)	3/B/O/M	SKP	Sezgisel	İnsan zekasına dayalı sezgisel yaklaşım
Schuster vd. [57], (2010)	3/B/O/R	SLOPP	Sezgisel	Işın arama
Al-Shayea [58], (2011)	3/B/O/R	SLOPP	Kesin	0-1 tamsayı programlama modeli
Zúniga vd. [59], (2011)	3/B/O/R	SLOPP	Sezgisel	Renkli petri ağı optimizasyon yaklaşımı
Birgin vd.[60], (2012)	2/B/O/R	SLOPP	Sezgisel	Özyinelemeli bölümlene yaklaşımı
Taylor vd. [61], (2015)	3/B/O/R	SLOPP	Kesin	Karışık tamsayı programlama

2.2. Daire Paketleme Problemi

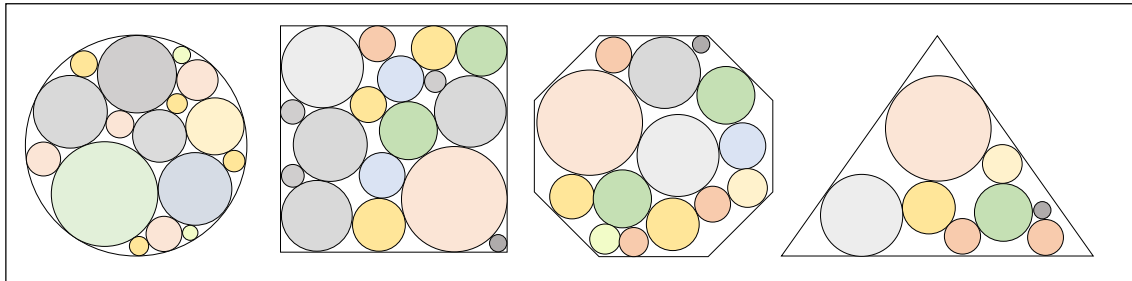
Bu bölümde kesme ve paketleme problemi olarak iki boyutlu daire paketleme problemi ele alınmıştır. Bu kapsamda, problemin tanımı yapılarak, problemin tarihçesi, kullanım alanları hakkında kapsamlı bir şekilde ele alınmıştır. Ayrıca problemin sınıflandırılmasından bahsedilerek, literatürü hakkında bilgi verilmiştir.

2.2.1. DPP'nin tanımı

Daire Paketleme Problemi (DPP), n adet özdeş veya özdeş olmayan dairenin üst üste gelmeyecek ve paketlenecek alan içinde bütünüyle konumlanacak (dışarı taşmayacak) şekilde yerleştirilmesi problemi olarak tanımlayabiliriz. Tanımlamadaki özdeş daireler Şekil 2.9'da gösterildiği gibi yarıçapları aynı olan daireleri ifade ederken, özdeş olmayan daireler ise Şekil 2.10'da gösterildiği gibi yarıçapları farklı çeşitli boyutlarda olan birbirinden farklı daireleri ifade etmektedir. Paketlemenin veya konumlamamanın yapıldığı alanlar ise kare, dikdörtgen, üçgen, daire, çokgen, elips vb. şekillerde olabilmektedir.



Şekil 2.10. Özdeş dairelerin paketlemesi



Şekil 2.9. Özdeş olmayan dairelerin paketlemesi

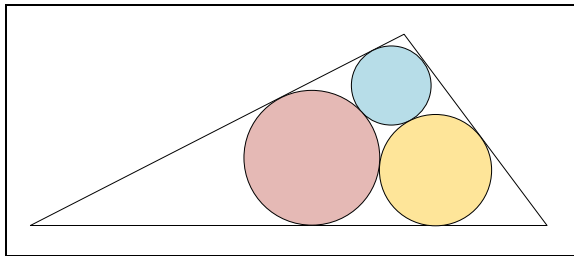
Bu problemin tanımı literatürde farklı araştırmacılar tarafından çok farklı şekillerde yapılmıştır. Örneğin Chen vd. [62], “Malzeme kullanımını maksimuma çıkaracak ve atıkları en aza indirecek şekilde, eşit veya eşit olmayan yarıçaplı çok sayıda dairenin büyük paletler

içine iyi bir düzenleme bulmasıyla ilgilenen klasik bir kombinatoryal optimizasyon problemidir.” şeklinde detaylı bir tanımlama yapmışken, Lopez ve Beasley [63] problem hakkında “Daire ambalajlama problemi, bir palet veya konteyner içindeki üst üste binmeme durumu ile n sayıdaki dairelerin düzenlenmesi ile ilgilidir.” demiştir. Ayrıca Stephenson [64] “daire paketleme, belirli bir teğet modeline sahip dairelerin bir konfigürasyonudur.” şeklinde kısa bir tanımlama yaparken, Castillo vd. [65] “daire paketleme, iki dairenin üst üste gelmeyeceği şekilde, bir konteyner (örneğin bir dikdörtgen veya bir daire) içindeki rastgele boyuttaki n dairenin optimize edilmiş bir düzenlemesidir.” şeklinde tanımlamıştır.

Kesme ve paketleme problemleri genel olarak en iyi çözümün geliştirilmesinin zor olduğu problemler olarak kabul edilir. Bu sebeple iki boyutlu daire paketleme problemleri NP-Zor kombinatoryal optimizasyon problemler arasındadır [66].

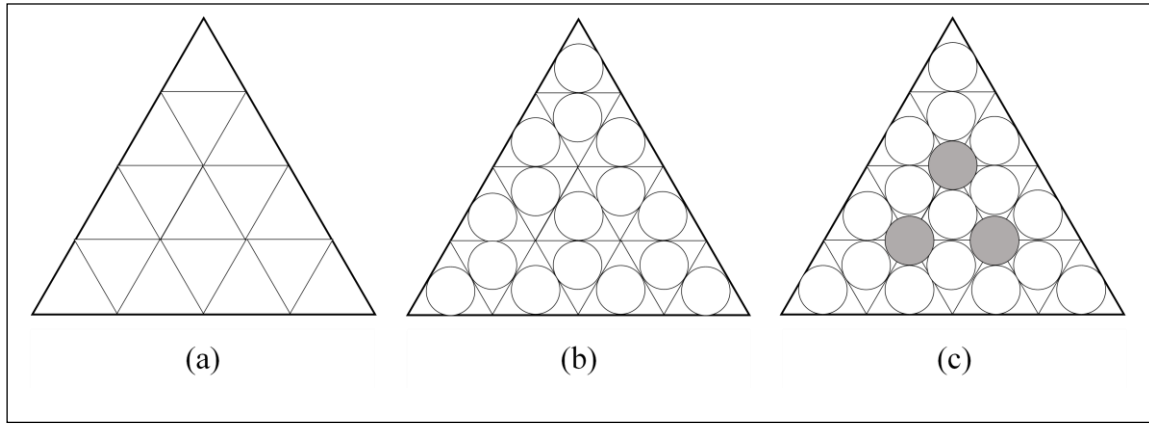
2.2.2. DPP’nin tarihçesi

DPP’nin çözümüne yönelik çalışmalar yüzyıllar öncesine dayanmaktadır. İtalyan bilim adamı Gianfrancesco Malfatti’nin, 1803’de yaptığı çalışmada [67] sorduğu soru ile problem çözümüne katkı sağlamıştır. Malfatti’nin, “Herhangi bir maddeden, örneğin, mermerden yapılmış bir üçgen dik prizmadan en az malzeme ziyan olacak şekilde üç dairesel dik silindir nasıl çıkarılır?” sorusundaki üç boyutlu yapının (üçgen prizma, silindir) dik izdüşümünün Şekil 2.11’de iki boyutlu hali gösterilmiştir. Çözümde her çember, diğer iki çembere ve üçgenin iki kenarına teğet olacak şekilde konumlandırmıştır. Probleminin çözümündeki amaç, dairelerin dışında oluşan atık alanı minimize etmektedir. Bir başka deyişle dairelerin alanlarını maksimize ederek fazla miktarda verim elde etmek hedeflenmektedir.



Şekil 2.11. Malfatti çemberleri

Macar matematikçi Farkas Bolyai'nin (1775-1856), daire yerleşimi ile ilgili çok sayıda çalışması olmuştur. Szabo vd.'ne göre [4], sınırlı bölgelerdeki daire paketlemenin yoğunluğu üzerine araştırma yapmış ve sonuçlarını yayınlamış olan Bolyai, 1832-33 yılında yapılan "Tentamen" olarak adlandırılan çalışmada [68] Şekil 2.12'de gösterildiği üzere eşkenar üçgende eşit dairelerin yoğun bir şekilde dizilmesi problemi üzerinde durmuştur. Çalışmaya göre, eşkenar üçgen daha küçük eşkenar üçgenlere bölünerek Şekil 2.12 (b) deki gibi üçgenlerin içine daireler yerleştirilmiştir. Daha sonra altı adet daire ile çevrili olan boşluklara, Şekil 2.12 (c)'deki gibi üç daire daha yerleştirilerek daire paketleme problemini en iyi şekilde çözülmüştür.

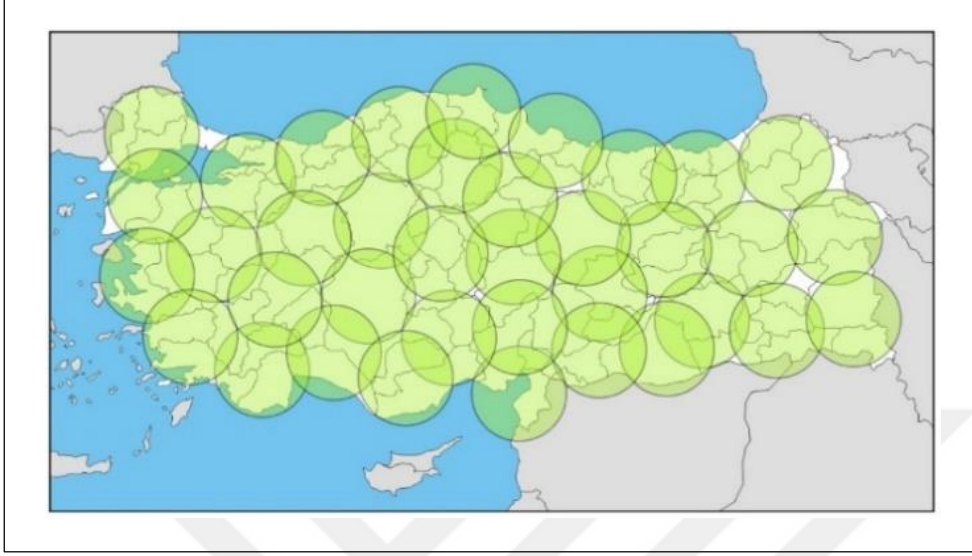


Şekil 2.12. Eşkenar üçgende daire paketleme

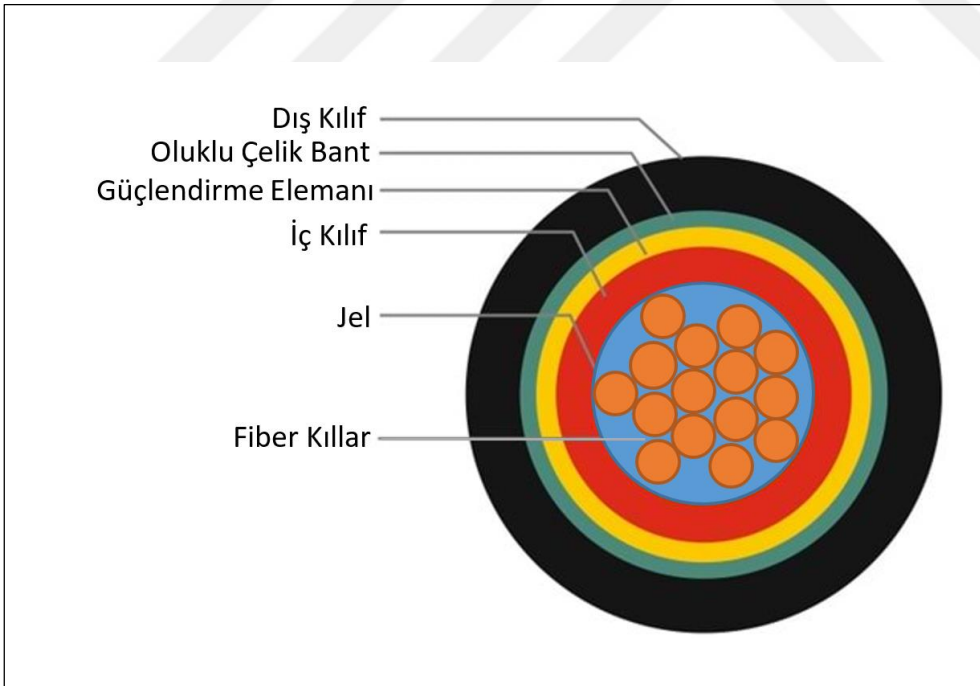
2.2.3. DPP'nin kullanım alanları

DPP, hayatın birçok alanında karşılaşılan problemlerden biridir. Doğa bilimlerinden, lojistik faaliyetlere, mühendislik tasarımından iletişim hatlarına geniş bir uygulama alanına sahiptir. Resim 2.1'de gösterilen radar sistemlerin bir coğrafi alanı en az keşişim ve en az fire ile verimli bir şekilde kapsaması DPP'ye örnektir. Ayrıca Resim 2.2'de görüldüğü üzere fiber optik kabloların dış çapının en küçük olacak şekilde çapları 125-900 mikrometre arasında değişen cam kıllarla döşenmesi iletişim ve elektronik sektörü için bir örnek oluşturmaktadır. Tarım arazilerinin verimini arttıracak şekilde sulanabilmesi için 1940 yılında A.B.D. Colorado'da bir çiftçi tarafından icat edilen Center Pivot Robotik Sulama Sistemi [69] dairesel şekillerde dizayn edilen tarlalar sayesinde verimli tarım faaliyetleri yürütülmektedir. Arazi kullanımının optimal seviyeye taşınabilmesi için DPP'nin çözümüne ihtiyaç duyulmaktadır. Resim 2.3'de gösterilen ve dairesel şekillerde oluşturulan tarlaların dairesel

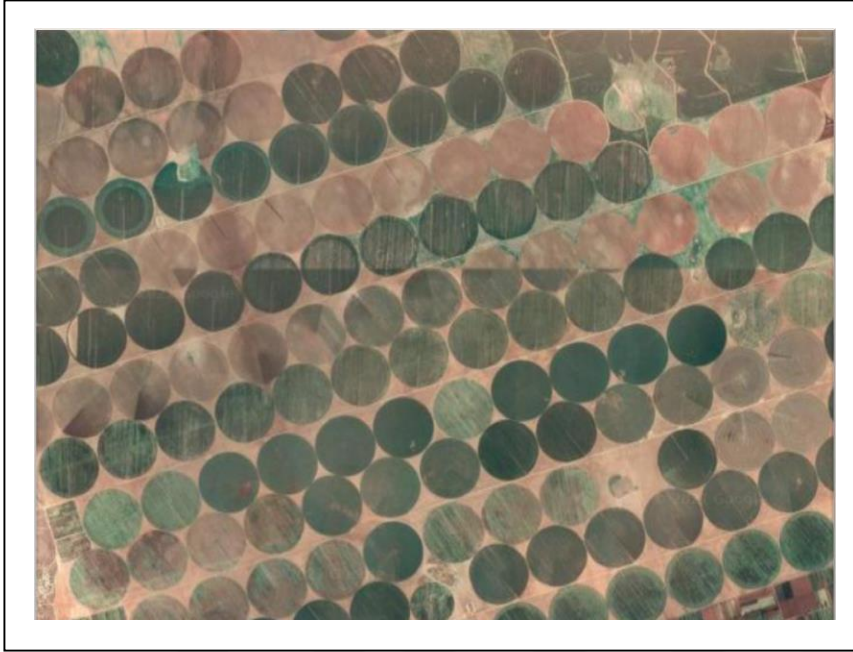
şekilde sulanması sayesinde işçi maliyetlerinde düşüş, büyük oranda su tasarrufu ve ürün verimi sağlamaktadır.



Resim 2.1. Bir coğrafi alanı en az keşişim ve en az boşlukla kapsama problemi



Resim 2.2. Fiber optik kabloların dış çapının minimizasyon problemi



Resim 2.3. Daire şeklindeki tasarlanan tarım arazileri [69]

Ayrıca, DPP'nin deniz taşımacılığı, motosiklet endüstrisi, malzeme kesimi, moda endüstrisi, kablosuz iletişim, gıda endüstrisi vb. alanlarda geniş bir uygulama alanının olduğu belirtilmiştir [5]. Szabo vd. [4] DPP'nin kullanım alanlarını şu şekilde örneklendirmişlerdir.

- *Kapsama alanı:* Coğrafi bir bölgeye yerleştirilen radyo kulelerinin mümkün olduğunca az keşimle maksimum kapsama alanının oluşturması.
- *Depolama:* Bir konteyner içine mümkün olduğunca çok sayıda nesnenin (örneğin varil) yerleştirilmesi.
- *Paketleme:* Belirli sayıda şişeyi içine sığdırabileceğiniz en küçük kutunun belirlenmesi.
- *Orman geliştirme:* Orman oluşumu için belirlenen bir alana, ağaçların maksimum genişleme alanı dikkate alarak dikim yapılması.
- *Kesim endüstrisi:* Bir parçadan mümkün olduğunca çok sayıda özdeş diskleri kesme.

Bir başka örnekte ise, otomotiv endüstrisinde, bir otomobilin sensörlerini ekran kartına bağlayan kablo demetini geçirmeyi planladıkları deliğin boyutunu göz önünde

bulundurulması DPP ile alakalıdır. Delik, tüm tellerin geçmesine izin verecek kadar büyük, ancak kasanın gereksiz yere zayıflamasını önlemek için olabildiğince küçük olmalıdır [6].

2.2.4. DPP'nin sınıflandırılması

DPP, bir yerleşim alanı (palet, konteyner, kutu, depo, vb.) üzerinde özdeş ya da farklı boyutlardaki dairelerin yerleştirilmesi problemidir. Bu nedenle bu çalışmada ele alınan problemin, iki boyutlu kutu paketleme problemi, stok kesim kaybı problemi, araç/konteyner yükleme problemi ve sırt çantası problemi gibi literatür içerisinde birçok problem tipi ile yakın ilişkisi bulunmaktadır.

DPP, paketleme problemlerinin sınıflandırılması ile ilgili Çizelge 2.1'de gösterilen ve Dyckhoff tarafından önerilen sınıflandırmaya göre 2/B/O/R şeklinde simgesel gösterimle tanımlama yapmak mümkündür. Ayrıca Wäscher vd. tarafından yapılan ve Şekil 2.7'de de gösterilen çıktı maksimizasyonu amaç türündeki çalışmaya göre ise SLOPP sınıfında bir problem olduğu görülmektedir.

2.2.5. Literatür özeti

Bu bölümde, DPP'nin çözüm yöntemleri üzerine, literatürde yer alan çalışmalara yer verilmiştir. Geçmişten günümüze DPP ile ilgili birçok araştırma ve çalışma yapılmıştır. Yapılan çalışmalarda DPP'nin farklı türleri, farklı hedefleri ve kısıtları bulunmaktadır. DPP ile ilgili erken dönem çalışmalarda daire sayısının (n) daha az sayı ve daha az karmaşık problemlere çözüm arandığını görmekle birlikte, daha sonraki yıllarda daire sayılarının ve kısıtların arttığı, palet türlerinin çeşitlendiği dolayısıyla problemlerin karmaşıklığının arttığı haldeki problemler karşımıza çıkmaktadır. [70]

DPP ile ilgili çalışmalar arttıkça, problemlerle ilgili yapılan literatür araştırması çalışmaları da gün geçtikçe artmaktadır. Hifi ve M'Hallah [66], daire palet üzerinde DPP'ye yönelik modeller ve metodolojiler ile ilgili kapsamlı bir inceleme sunmuştur. Ayrıca Szabo vd. [4] DPP ile ilgili detaylı literatür incelemesinde bulunup, DPP'nin kare palet üzerinde çözümünü geliştirdiği çalışmasında yer vermiştir. İki boyutlu DPP'yi farklı geometrik şekillerdeki paletler üzerinde inceleyerek sunan çalışmada [71], özdeş veya özdeş olmayan dairelerin yerleşimi ile ilgili karşılaştırmalı sonuçlara yer verilmiştir.

Literatürde DPP ile ilgili yapılan çalışmalar, paletin geometrik şekline, dairelerin özdeşlik durumuna ve sayısına, amaç fonksiyonlarına ve çözüm yaklaşımına göre çeşitlik göstermektedir. Paketlenecek dairelerin birbirleriyle farklı yarıçaplara sahip olduğu özdeş olmayan daire paketleme problemleri ile ilgili literatürdeki çalışmalar Çizelge 2.4’de, dairelerin birbirleriyle aynı yarıçaplara sahip olduğu özdeş daire paketleme problemleri ile ilgili literatürdeki çalışmaları ise Çizelge 2.5’de amaç fonksiyonuna, kesin veya sezgisel çözüm yöntemine göre listelenmiştir.

Çizelge 2.5. Özdeş olmayan daire paketleme ile ilgili çalışmalar

Çalışma	Palet Şekli	Amaç Fonksiyonu	Yaklaşım	Açıklama
Wang vd. [72]	Daire	Yarıçap minimizasyonu	Sezgisel	Tabu Araması
Huang vd. [73]	Daire	Yarıçap minimizasyonu	Sezgisel	İki Seviyeli Arama Strateji
Huang, Yan [74]	Daire	Çakışmama minimizasyonu	Sezgisel	Yarı Fiziksel Algoritma
Zhang, Huang [75]	Daire	Çakışmama minimizasyonu	Sezgisel	Tavlama Benzetimi
Zhang, Deng [76]	Daire	Çakışmama minimizasyonu	Sezgisel	Tavlama Benzetimi/ Tabu Araması
Akeb vd.[37]	Daire	Yarıçap minimizasyonu	Sezgisel	Işın Arama Algoritması
Carrabs vd. [77]	Daire	Yarıçap minimizasyonu	Sezgisel	Tabu Arama
Specht [78]	Daire	Yarıçap minimizasyonu ve daire sayısı maksimizasyonu	Sezgisel	Polidisperse Algoritması
Lopez ve Beasley [79]	Daire	Alan ve daire sayısı maksimizasyonu	Sezgisel	Boşluk Arama Formülasyonu
Orick vd. [80]	Daire	Daire sayısı maksimizasyonu	Kesin	Doğrusallaştırılmış Algoritma
Wang ve Zhang [81]	Daire	Daire sayısı maksimizasyonu	Sezgisel	Sürü Zeka İş Bölümü Algoritması
He, Huang, Yang [5]	Kare	Kenar uzunluk minimizasyonu	Sezgisel	ASGO Algoritması
Stoyan, Yaskov [82]	Dikdörtgen	Uzun kenar minimizasyonu	Kesin	Matematiksel Model/ Zıplama Algoritması
Hifi, M'Hallah [83]	Dikdörtgen	Sabit boyuttaki dikdörtgenin kullanım alanı maksimizasyonu	Sezgisel	Genetik Algoritma
Hifi, vd [84]	Dikdörtgen	Sabit boyuttaki dikdörtgenin kullanım alanı maksimizasyonu	Sezgisel	Tavlama Benzetimi
Stoyan, Yaskov [85]	Dikdörtgen	Uzun kenarı minimize etme	Kesin	Matematiksel Model
Zhang, Liu, Chen[86]	Dikdörtgen	Çakışmama minimizasyonu	Sezgisel	Tavlama Benzetimi
Akeb, Hifi, Negre[87]	Dikdörtgen	Uzun kenarı minimize etme	Sezgisel	Arttırılmış Algoritma
He, Wu [88]	Dikdörtgen	Uzun kenar minimizasyonu	Sezgisel	Genetik Algoritma
Torres-Escobar vd. [89]	Dikdörtgen	Dairelerin kapladığı alanı maksimize etmek	Sezgisel	Maymun Algoritması

Çizelge 2.4’de özdeş olmayan dairelerin daire, kare ve dikdörtgen palet üzerindeki yerleşim problemleri gösterilmiştir. Amaç fonksiyonu, yarıçap minimizasyonu olan çalışmalarda n adet dairenin tamamının yerleştirildiği, ancak dairesel paletin yarıçapının mümkün olduğu kadar minimize edilmeye çalışıldığı optimizasyon problemleridir. Bu tür minimizasyon problemleri kare palet için kenar uzunluğu minimizasyonu [5], dikdörtgen paletlerde ise uzun kenar minimizasyonu halini almaktadır.

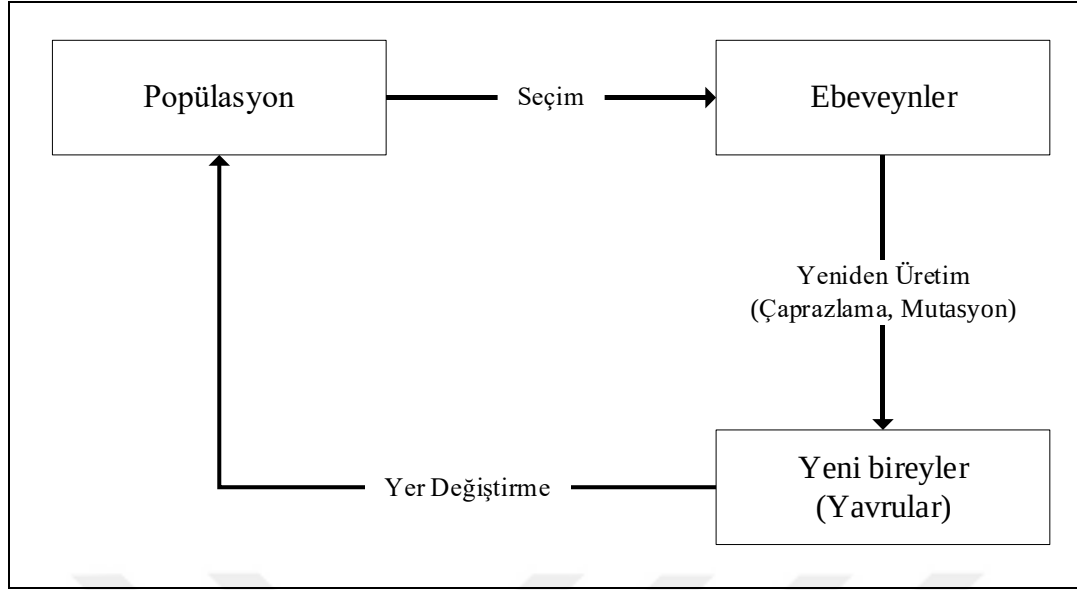
Çizelge 2.6. Özdeş daire paketleme ile ilgili çalışmalar

Çalışma	Palet Şekli	Amaç Fonksiyonu	Yaklaşım	Açıklama
Lubachevsky, Graham [90]	Daire	Daire sayısı maksimizasyonu	Sezgisel	Bilardo Benzetimi Algoritması
Graham vd. [91]	Daire	Daire sayısı maksimizasyonu	Sezgisel	Bilardo Benzetimi Algoritması
Birgin, Gentil[92]	Daire	Yarıçap minimizasyonu	Sezgisel	Doğrusal olmayan Programlama
Chen vd. [62]	Daire	Dış çap minimizasyonu	Sezgisel	Açgözlü Algoritması
He, Ye, Wang, Liu [93]	Daire	Dış çap minimizasyonu	Sezgisel	Yarı fiziksel –Yarı insan Algoritması
He vd. [94]	Daire	Daire sayısı maksimizasyonu	Sezgisel	Stokastik Parça İniş Yöntemi ile Yarı fiziksel–Yarı insan Algoritması
Goldberg [95]	Kare	Daire sayısı maksimizasyonu	Sezgisel	Çember Paketleme
Mollard, Payan [96]	Kare	Daire sayısı maksimizasyonu	Sezgisel	Cabri-Geometri
Theodoracatos, Grimsley [97]	Kare	Daire sayısı maksimizasyonu	Sezgisel	Tavlama Benzetimi
Nurmela, Östergard [98]	Kare	Daire sayısı maksimizasyonu	Kesin	50 daireye kadar kesin çözüm yaklaşımı
Boll vd. [99]	Kare	Daire sayısı maksimizasyonu	Sezgisel	Bilardo Benzetimi Algoritması
Locatelli, Raber[100]	Kare	Daire sayısı maksimizasyonu	Kesin	Dal–Sınır Algoritması
Markot, Csendes[101]	Kare	Daire sayısı maksimizasyonu	Kesin	Dal–Sınır Algoritması
Costa vd. [102]	Kare	Sabit boyuttaki palette dairelerin yarıçap maksimizasyonu	Kesin	Dal–Sınır Algoritması
Dowsland[103]	Dikdörtgen	Daire/silindir sayısı maksimizasyonu ve toplam palet sayısı minimizasyonu	Sezgisel	Geometrik analizle, daire komşuluklar arası açığı optimize etme.
Correia vd. [104]	Dikdörtgen	Sabit boyuttaki dikdörtgen içine yerleştirilen daire sayısını maksimize etme	Sezgisel	Tavlama Benzetimi
Birgin vd. [105]	Dikdörtgen	Paketlenen daire sayısının maksimize etmek	Sezgisel	Doğrusal Olmayan Programlama

DPP, NP-Zor problemler sınıfında olması sebebiyle metasezgisel yöntemlerle çözülebilmektedir. Küçük boyutlu problemler, kesin çözüm metotlarıyla kısa sürede optimal sonuç verebilirken, problemin daire sayısı, kısıtları ve karmaşıklığı arttıkça optimale yakın çözümler ancak sezgisel yöntemlerle bulunabilmektedir. Çizelge 2.4 ve Çizelge 2.5 de gösterildiği üzere, literatürde DPP'nin çözümü için birçok sezgisel yöntem kullanılmıştır. Bu sezgiseller arasında başta tavlama benzetimi, tabu araması, genetik gibi başlıca yöntemlerin yanı sıra açgözlü, ışın arama, süre zeka iş bölümü, bilardo benzetimi gibi bir çok farklı yöntem de DPP'nin çözümü sırasında başvurulmuş yöntemler arasındadır.

2.3. Genetik Algoritma

Temeli biyolojik evrim mekanizmalarından esinlenilerek oluşturulan, bilgisayar tabanlı problem çözme ve optimizasyon tekniklerin tamamına evrimsel algoritmalar adı verilmektedir. Evrimsel algoritmalar çok sayıda karmaşık problemi çözmeye yaygın olarak kullanılmaktadır. Şekil 2.13'de gösterildiği gibi, popülasyonu oluşturan bireylerin rekabet temelli seçim, değişim ve yeniden üretim süreçleriyle evrimleştirilmesi, güçsüz olanın yok olduğu bununla birlikte güçlü olanın hayatta kalarak sürece yeniden dahil edildiği fikrine dayanmaktadır. Bu tür algoritmaların alt başlığında genetik algoritmalar, genetik programlamalar, evrimsel stratejiler, evrimsel programlama algoritmaları bulunmaktadır [106]. Bunlar arasından en kullanışlı ve en çok tercih edilen evrimsel algoritma türü ise genetik algoritmadır. Genetik algoritma, Darwin'in en iyi olan hayatta kalır prensibine göre, doğal çevreye uyum sağlayabilen bireylerin hayatına devam edebildiği, üstün özellikli ebeveynlerden üstün özellikli bireylerin üretildiği fikrine dayalı doğal seçim ve doğal genetik süreçlerin benzeri bir yapı olarak karşımıza çıkmaktadır.



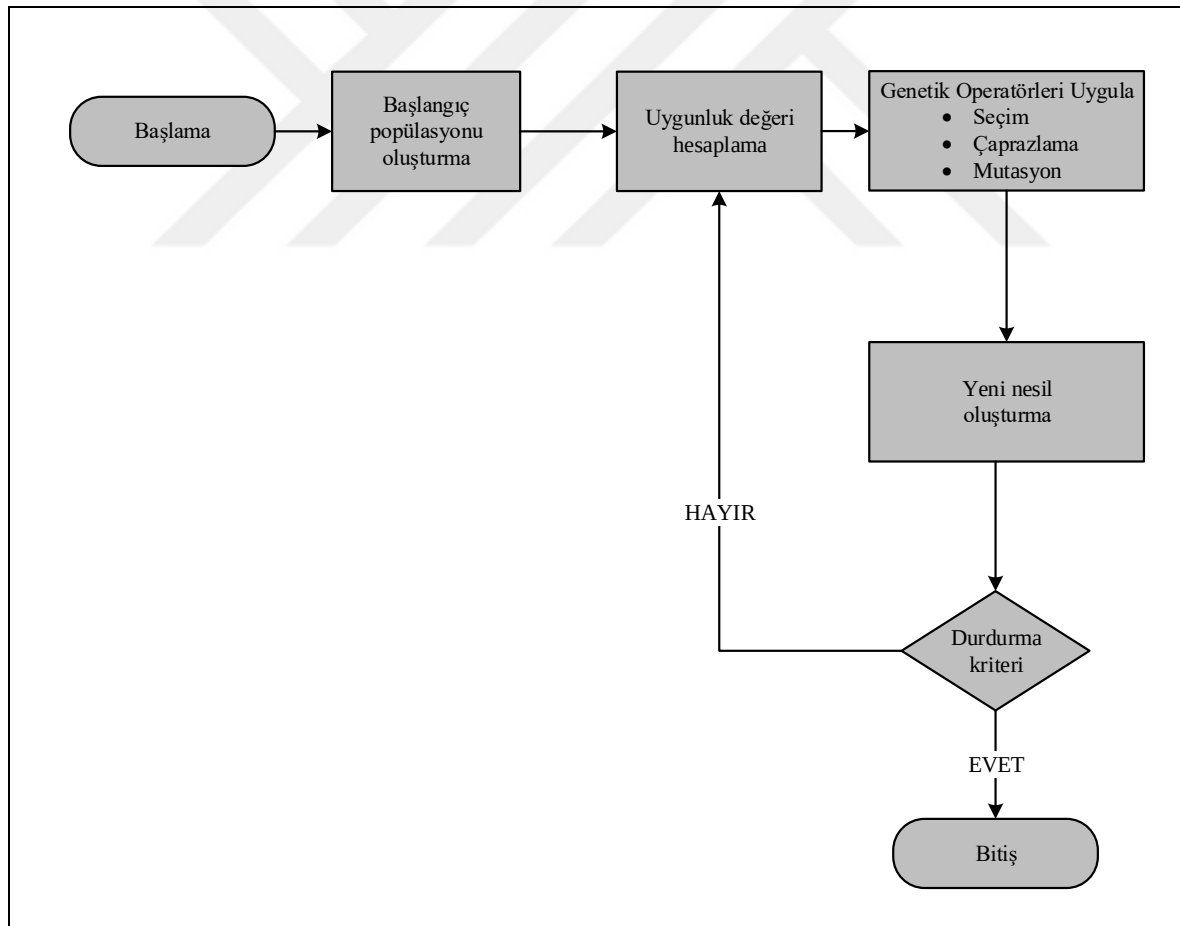
Şekil 2.13. Evrimsel algoritma süreçleri

Genetik algoritma (GA), deterministik yöntemlerle çözümü zor veya karmaşık olan problemlerin en iyi ya da en iyiye yakın çözümlerini en kısa sürede bulmayı amaçlayan, doğadaki doğal evrimsel sürecin bilgisayar tabanlı sistemler üzerinde taklit edilmesine dayalı, rassal tabanlı bir arama yöntemidir [107]. Bir başka deyişle GA, çözüm uzayı içindeki aramalarını rassal olarak gerçekleştiren ve bu sayede çeşitliliği üst düzeye çıkaran bir araştırma yapmaktadır. Bu rassallık ve bunun sonucu oluşan çeşitlilik, algoritmanın yerel optimuma takılmasının önüne geçmekte ve sürekli iyiye yakınsama durumu ortadan kaldırmaktadır.

Genetik algoritmanın temeli 1960'lı yıllarda John Holland'ın Darwin'in evrim kuramından etkilenecek yaptığı çalışmalara dayanmaktadır. Bu çalışmalarda, Holland bir problemin çözümü için bir algoritma geliştirmekten farklı olarak canlı mekanizmaların genetik süreçler neticesinde bulunduğu doğal ortam şartlarına adaptasyon sağlayıp en iyinin elde edilmesi ve bu mekanizmaların bir bilgisayar sistemi üzerinde uygulanabileceği yöntemlerin geliştirilmesi olmuştur. Holland'ın 1975'te yayınladığı çalışma ile beraber yöntem Genetik Algoritma olarak literatüre yerleşmiştir. Ancak Holland'ın öğrencisi David E. Goldberg, 1989 yılında yayınladığı ve konusunda klasik olarak nitelendirilen kitap yayınlanana kadar GA'nın, pratikte pek faydalı olmayan bir araştırma konusu olduğu düşünülüyordu [108]. Günümüzde GA, optimizasyon, otomatik programlama ve bilgi sistemleri, mekanik öğrenme, ekonomik ve sosyal sistem modelleri, finans, pazarlama, montaj hattı dengeleme problemi, çizelgeleme problemi, tesis yerleşim problemi, atama problemi, hücresel üretim

hattı problemi, sistem güvenliği problemi, taşıma problemi, gezgin satıcı problemi, araç rotalama problemi, minimum yayılan ağaç problemi gibi birçok alanda uygulanmaktadır [109].

Klasik bir genetik algoritma, kendine özgü bir takım süreçlerden oluşmaktadır. Şekil 2.14’de gösterilen süreç adımlarına göre; algoritma, potansiyel çözümleri içeren bir popülasyonun oluşturulması ile çalışmaya başlamaktadır. Rasgele olarak oluşturulan başlangıç popülasyonundaki her bireyin performansını gösteren uyum fonksiyonuna göre uygunluk değeri hesaplanır. Daha iyi bir uygunluk değeri elde edebilmek için yeni bireyler oluşturulur. Yeni birey oluşumu için seçim, çaprazlama ve mutasyon işlemleri devreye alınır. Bu süreç durdurma kriteri şartı sağlanana kadar devam eder. Durdurma kriteri, hedeflenen en iyi uygunluk değeri veya maksimum işlem sayısı olabilmektedir.



Şekil 2.14. Genetik algoritma akış şeması

Çoğu GA'nın başarısı büyük ölçüde, aday çözümlerin nasıl kodlandığı, popülasyon büyüklüğü, seçim, çaprazlama ve mutasyon operatörlerinin detayları ve ihtimallerine ek olarak izin verilen maksimum nesil sayısı gibi detaylara bağlıdır [110].

2.3.1. Gen ve kromozom

Genetik biliminde gen için, “bir canlının genetik bilgisini ve kalıtsal özelliklerini taşıyan, ebeveynden çocuğa aktarılan en temel fiziksel ve fonksiyonel kalıtım birimidir.” şeklinde bir tanımlama yapılabilmektedir. Daha basit bir ifade ile “anlamalı bir genetik bilgisi olan en küçük yapı birimidir.”

Birden fazla genin belirli bir sıra halinde dizilmesi ile ortaya çıkan genler dizisine kromozom adı verilmektedir. Örneğin, gezgin satıcı problemi gibi bir problemde, bir gen bir şehri temsil ederken, kromozomlar ise şehirden şehre giden bir yolu temsil etmektedir.

Her bir kromozom, temsil ettiği problemin çözümüne ilişkin bilgileri içermekte ve çözüm uzayında tek bir nokta ile ifade edilmektedir. GA'da tüm süreçler kromozomlar üzerinden yapılması sebebiyle kromozomların düzgün oluşturulması önem arz etmektedir.

2.3.2. Kodlama

Kodlamayı, genlerin kromozom üzerinde gösterim süreci olarak tanımlamak mümkündür. Bu gösterimde, bitler, sayılar, diziler, listeler, ağaçlar, semboller ya da nesneler kullanılabilir. GA'da bir problemin çözüme ulaşabilmesi için uygun bir şekilde kodlanması gerekmektedir. Çözümlerin kodlanmasıyla beraber problem bilgileri genetik algoritma için kullanıma hazır hale gelmektedir.

GA'da problemin türüne bağlı olarak farklı kodlama türleri kullanılabilir. İkili kodlama, permütasyon kodlama, değer kodlama, ağaç kodlama gibi kodlama türlerinin açıklaması şu şekildedir:

İkili kodlama

İkili kodlama yönteminde her bir kromozom ikili (bit) dizisinden oluşan genlerle gösterilmektedir. GA ile ilgili yapılan ilk çalışmalardan itibaren günümüze sıklıkla

kullanılmaktadır. Şekil 2.15’de A ve B kromozomlarına ait ikili kodlama örneği gösterilmiştir. İkili kodlama yöntemi ile çözülen problemler arasında sırt çantası problemi örnek olarak gösterilebilir. Nesnelerin çantaya alınıp alınmayacağına dair bilgiye kromozom içindeki 0 ya da 1 değerine göre karar verilmektedir.

Kromozom A	0	1	1	0	1	0	1	0	1	0	1
Kromozom B	1	1	0	0	1	1	0	1	1	0	1

Şekil 2.15. İkili kodlama örnekleri

Permütasyon kodlama

Permütasyon kodlama yöntemi, sıralamanın önemli olduğu durumlarda uygulanmaktadır. Çizelgeleme, iş sıralama, gezgin satıcı vb. problem türlerinde sıklıkla kullanılmaktadır. Şekil 2.16’da gösterildiği gibi kromozomdaki her bir gen bir tamsayı sayı ile temsil edilmektedir. Kromozomlardaki belirtilen tamsayı sıradaki çözümün indis numarasını ifade etmektedir. Örneğin, gezgin satıcı probleminde bir sonraki şehrin indisi, kutu paketleme probleminde çözüm alanına yerleştirme sırası gelen kutunun indisi permütasyon kodlama ile ifade edilmektedir.

Kromozom A	7	2	5	11	9	1	6	4	8	10	3
Kromozom B	1	8	4	7	2	5	3	10	11	9	6

Şekil 2.16. Permütasyon kodlama örnekleri

Değer kodlama

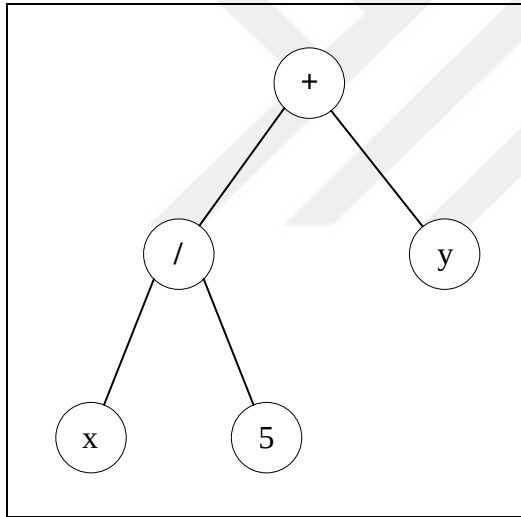
Bu kodlama türü bazı özel problemlerin çözümü için iyi sonuçlar verebilmektedir. Kromozom, Şekil 2.17’de gösterildiği gibi probleme özgü türden, form numarası, karakter, reel sayı, metin, sembol vs. kompleks değerleri içerebilir.

Kromozom A	2,18	3,146	11,68	14,66	0,33	7,871	6,33
Kromozom B	A	C	D	F	H	B	E
Kromozom C	üst	sağ	sol	alt	ön	sağ	arka

Şekil 2.17. Değer kodlama örnekleri

Ağaç kodlama

Ağaç kodlama yöntemi esas olarak genetik programlamada program ifadelerini geliştirmek için kullanılır. Her kromozom, bir programlama dilinin fonksiyonlar ve komutları gibi bazı nesnelerinin birer ağacıdır [111]. Şekil 2.18’da gösterildiği gibi ağaç şeklinde kodlanabilecek yapılar için uygundur.



Şekil 2.18. Ağaç kodlama gösterimi

2.3.3. Popülasyon

Bireylerin oluşturduğu topluluğa popülasyon denmektedir. Bir popülasyon, bireylerden, bireyleri tanımlayan parametrelerden ve arama alanı ile ilgili bilgilerden oluşmaktadır. GA’da çözümün kalitesi başlangıç popülasyonunun doğru yapılandırılması ile artmaktadır. Bu sebeple, başlangıç popülasyonun oluşturulma şekli ve popülasyon büyüklüğü GA’nın ölçeklenebilirliği ve performansını etkileyen en önemli faktörlerden biridir.

GA'da, başlangıç popülasyonunun rassal bir şekilde oluşturulması sık karşılaşılan bir durumdur. Ayrıca popülasyon büyüklüğünün doğru seçimi de problemin çözüm kalitesi açısından oldukça önemlidir. Popülasyon büyüklüğü, hem kaynak kullanımını olumsuz yönde etkileyecek kadar fazla olmamalı, hem de çözüm uzayını kısıtlı tarayacak ve erken yakınsamaya sebep olacak kadar da az olmamalıdır.

2.3.4. Uygunluk değeri ve uygunluk fonksiyonu

Her bir kromozom, temsil ettiği çözümün kalitesini göstermesi açısından yeteneği ile doğru orantılı sayısal bir uygunluk değerine sahiptir. Bu uygunluk değerinin hesaplanmasıyla ortaya çıkan sonuç, kromozomun çevreye uyum gücünü veya hayatına devam edip edemeyeceğini belirtmektedir. Bir başka deyişle, uygunluk değeri, bireylerin her seferinde yeniden üretilen popülasyonda temsil edilip edilmeyeceğinin belirlenmesinde kullanılan bir araçtır. Bir kromozomun uygunluk değeri ne kadar yüksekse yaşama ve çoğalma olasılığı o kadar fazla olmaktadır.

Kromozomların uygunluk değerlerini hesaplayabilmek için iyi oluşturulmuş bir fonksiyona ihtiyaç duyulmaktadır. Uygunluk fonksiyonları, problemin türüne göre istenilen şekilde oluşturulabilmektedir. Örneğin, mühendislik alanındaki problemlerin çözümü için geliştirilen GA'lardaki uygunluk fonksiyonu, genişlik, alan, hacim, uzunluk, ağırlık, maksimum parça sayısı, dayanıklılık oranı, teslim süresi, maliyet vs. olabildiği gibi bunlardan bir kaçının bir araya gelmesi ile oluşturulan fonksiyonlar da olabilmektedir.

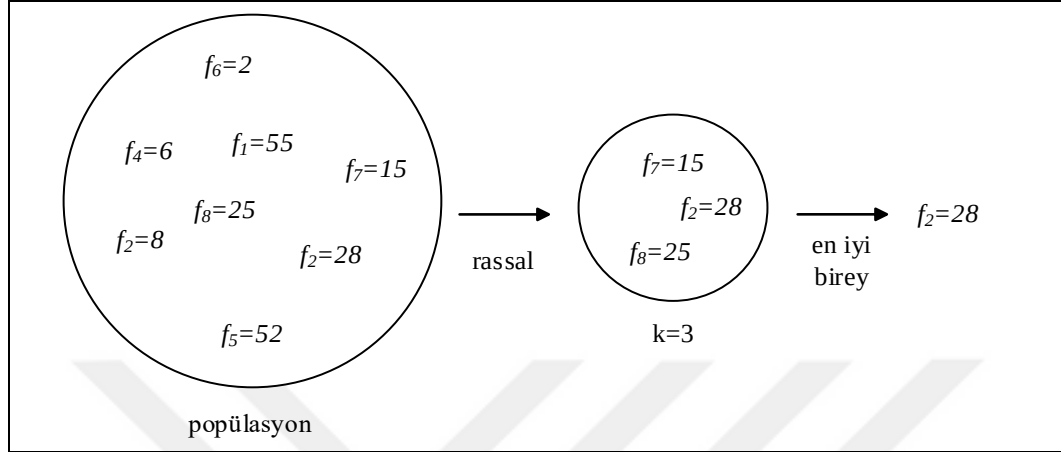
2.3.5. Seçim mekanizmaları

Seçim süreci, başlangıç popülasyonundaki bireylerin, en iyi uygunluk değerine sahip bireyler arasından bir kısmının seçilerek mevcut popülasyondan yeni popülasyona doğru aktarılması işlemidir. Literatürde birçok seçim yöntemi bulunmaktadır. Bu çalışmada, en çok kullanılan seçim yöntemlerinden olan turnuva seçimi ve rulet seçimi teknikleri açıklanmıştır [112].

Turnuva seçimi

Turnuva seçimi, basit, kullanılabilir ve etkin bir yöntem olması sebebiyle GA'da çok sık tercih edilmektedir. Şekil 2.19'de gösterilen örnekte de belirtildiği gibi, i adet bireyden

oluşan popülasyon içinden k (turnuva büyüklüğü) adet birey rasgele seçilir ve seçilen bireyler kendi aralarında turnuvada yarışır. Uygunluk değerlerinin (f_i) yarıştırdığı bu turnuvada, iyi olan birey yarış kazanarak bir sonraki nesli oluşturmak için seçilmiş olur.



Şekil 2.19. Turnuva seçimi örneği

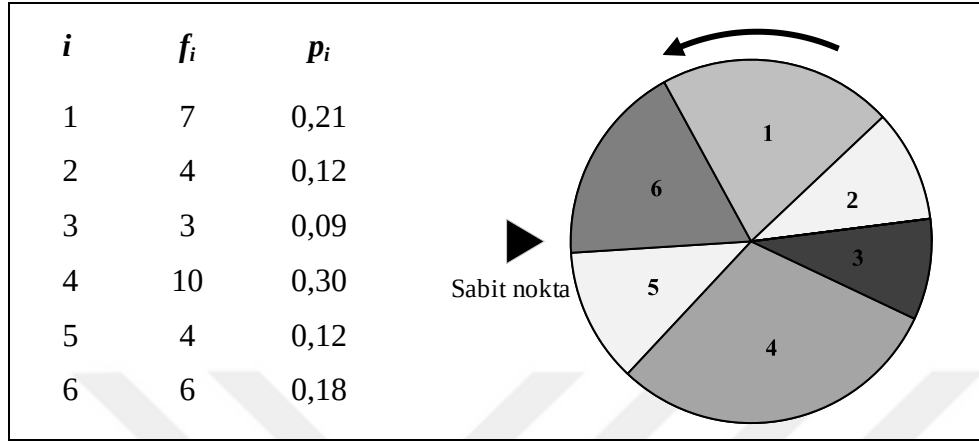
Popülasyondaki bireylerin seçiminde rasgelelik olması sebebiyle popülasyondaki tüm bireylerin turnuvada seçilme şansı bulunmaktadır. Bu yöntemle yarışma için seçilen bireyler popülasyonun diğer bireyelerine göre düşük uygunluk değerine sahip olsalar bile, sonraki nesillerde yer bulabilmektedirler. Bu durum çeşitliliği korumakta ve erken yakınsama durumunu ortadan kaldırmaktadır.

Rulet tekerleği yöntemi

Rulet tekerleği ya da rulet çarkı olarak da adlandırılan yöntemde, popülasyondaki tüm bireyler uygunluk değeri oranında rulet tekerleğine yerleşimi yapılır. Bir bireyin rulet tekerleği üzerindeki boyutu uygunluk değeri ile doğru orantılıdır. Bu sebeple uygunluk değeri yüksek olan bir bireyin rulet tekerleğinde kapladığı alan fazla olacağından seçilme şansı o oranda yüksek olur.

P popülasyonundaki i bireyinin uygunluk değeri (f_i) olmak üzere bu bireyin seçilme olasılığı Eş. 2.1’de hesaplanmaktadır. Şekil 2.20’de gösterildiği üzere, altı bireyden oluşan popülasyonun her bireyi, rulet tekerleğinde kendine p_i oranında yer bulmaktadır. Rulet tekerleğinin her çevriminde, sabit nokta önüne gelen bölge hangi bireyi temsil ediyorsa o birey seçilmiş olur. Bu işlem, seçilmek istenilen birey sayısı kadar tekrarlanır.

$$p_i = \frac{f_i}{\sum_{i=1}^n f_i} \quad (2.1)$$



Şekil 2.20. Rulet tekerleği gösterimi

Bu yöntemde uygunluk değeri, diğer bireylerin uygunluk değerinin aritmetik ortalamasından çok daha fazla olan bireyler, rulet tekerleğinde fazla alana sahip olmaları sebebiyle arama sürecinin ilk dönemlerinde erken yakınsamaya sebep olabilirler. Bir başka durumda ise, bütün bireylerin uygunluk değerleri birbirine yakın olduğu durumlarda ise değerli bireylerin seçilme ihtimalinin diğer bireylere göre çok farklı olmaması değerli bireyin seçilememesi ihtimalini doğurmaktadır.

2.3.6. Genetik operatörler

GA'da başlangıç popülasyonun oluşturulmasından sonraki nesillerin elde edilmesi için tekrar üretim sağlayan genetik operatörler kullanılmaktadır. Bu operatörler çaprazlama operatörü ve mustasyon operatörüdür. Genetik algoritma uygulamalarında yapılan işlemlerin %90-95'i seçim ve çaprazlama operatörleri ile ilişkilidir [113].

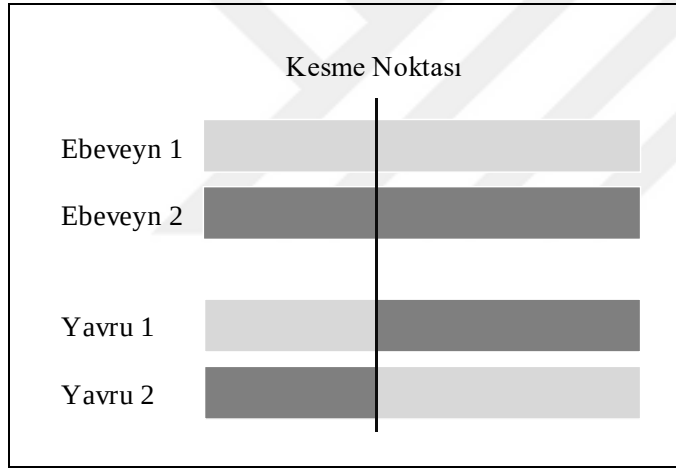
Çaprazlama

Geleneksel genetik algoritmalarda, çaprazlama operatörü temel operatör olarak kullanılır ve bir genetik sistemin performansını büyük ölçüde etkilemektedir [114]. Çaprazlama olasılığı (P_c) çaprazlamanın ne sıklıkla yapılacağını belirlemede etkili bir parametredir. Çaprazlama operatörünün kullanılmasındaki amaç, ebeveynlerin sahip olduğu özellikler

arasında deęişim yapılarak sonraki nesillere aktarımının saęlanmasıdır. aprazlama iřleminde iki bireyin kromozomları arasında eřitli yntemler kullanılarak karřılıklı yer deęiřiklięi yapılmaktadır. ok sayıda aprazlama operatr arařtırmacılar tarafından geliřtirilmiřtir. Bu yntemlerin bir kısmı ařaęıda detaylı bir řekilde aıklanmıřtır.

Tek noktalı aprazlama (One point crossover - 1PX)

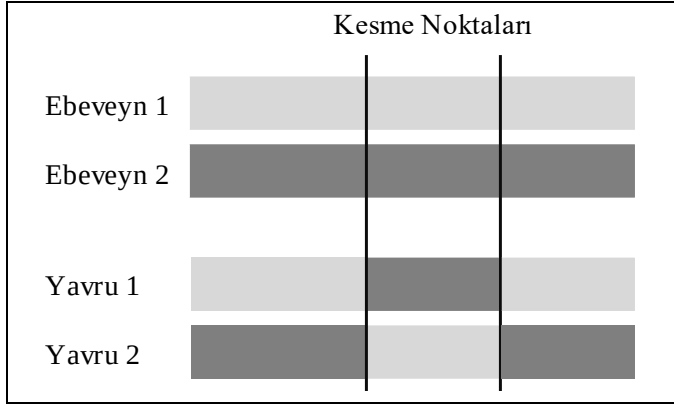
En basit aprazlama yntemidir. Kromozomlar zerinde kesme noktası olarak adlandırılan rasgele bir nokta seilir. Kromozomların, ilk genlerinden kesme noktasına kadar olan kısmı yeni bireylere doęrudan aktarılır. Kromozomun kalan kısımları ise yeni bireylere apraz bir řekilde karřılıklı olarak yerleřtirilir. řekil 2.21’de iki ebeveynden iki yavru retimi genel olarak gsterilmiřtir.



řekil 2.21. Tek noktalı aprazlama yntemi

İki noktalı aprazlama (Two point crossover - 2PX)

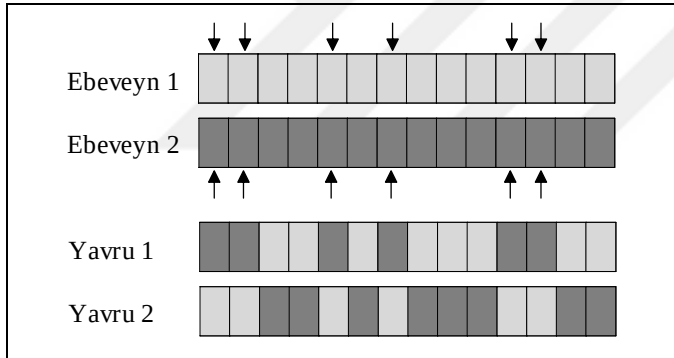
Bu aprazlama ynteminde kesme noktası olarak adlandırılan rasgele iki nokta seilir. Kromozom zerinde seilen iki kesme noktasının arasında kalan genler karřılıklı olarak yer deęiřtirir. Bylelikle iki yeni yavru elde edilmiř olur.



Şekil 2.22. İki noktalı çaprazlama yöntemi

Tek düze çaprazlama yöntemi (Uniform crossover –UX)

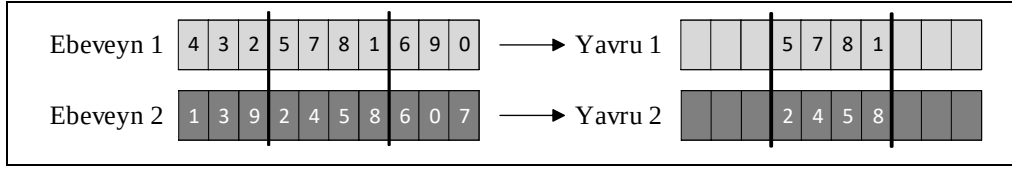
Şekil 2.23’de gösterildiği üzere İki yeni yavru üretimi için ebeveynlerden rasgele genler seçilir. Seçilen bu genler diğer ebeveynin aynı konumundaki genleri ile yer değiştirir.



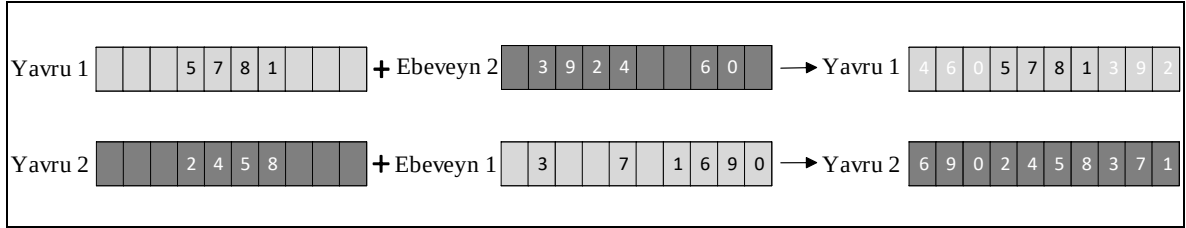
Şekil 2.23. Tekdüze çaprazlama yöntemi

Sıralı çaprazlama (Order crossover –OX)

Şekil 2.24’de gösterildiği üzere sıralı çaprazlamada rasgele iki kesme noktası tespit edilir ve iki nokta arasındaki genler ebeveynlerden yavrulara aktarılır. Daha sonra Şekil 2.25’deki gösterimde anlaşılaacağı üzere yavru 1’de bulunan genler ebeveyn 2’den, yavru 2’de bulunan genler ebeveyn 1’den çıkartılır. Ebeveyn 2’deki kullanılmayan genler yavru 1’e, ebeveyn 1’deki kullanılmayan genler yavru 2’ye göründükleri sıra ile her bir yavrunun ikinci kesme noktasından sonraki genden başlayarak aktarılır.



Şekil 2.24. Sıralı çaprazlamada kesme noktalarının gösterimi



Şekil 2.25. Sıralı çaprazlama işlemi ile yavru üretimi

Pozisyona dayalı çaprazlama (Position based crossover operator -PBX)

Bu yöntemde;

Adım 1: Ebeveyn 1 üzerinde rasgele seçilen bir dizi pozisyon seçilir.

Adım 2: Her seçilen pozisyonundaki semboller % 50 olasılıkla yavruya aktarılır.

Adım 3: Yavruya aktarılan semboller Ebeveyn 2'den silinir.

Adım 4: Ebeveyn 2'de kalan semboller sırasıyla yavruda boş kalan pozisyonlara aktarılır.

Sıraya dayalı çaprazlama (Order based crossover – OBX)

Sıraya dayalı çaprazlama yöntemi, pozisyona dayalı çaprazlama (PBX) yönteminin farklı bir versiyonu olarak Syswerda [115] tarafından önerilmiştir. OBX'in işlem adımları şu şekildedir:

Adım 1: Ebeveyn 1'den rasgele bir dizi gen seçilir.

Adım 2: Ebeveyn 1'den seçilen genlerin ebeveyn 2'de bulunan konumları tespit edilir.

Adım 3: Tespit edilen bu konumların yavrudaki karşılıklarına, ebeveyn 1'den seçilen genler sırasıyla aktarılır.

Adım 4: Yavrunun boş kalan konumlarına ebeveyn 2'den genler aktarılır.

Dairesel çaprazlama (Cycle crossover - CX)

Dairesel çaprazlama, 1987 yılında Oliver vd. [116] tarafından önerilmiş bir yöntemdir. Bu yöntem ilk ebeveynin ilk konumundan başlar ve bu ebeveyndeki genlerin diğer ebeveyndeki karşılığının tespitiyle bir döngü yaratılır. Adımları şu şekildedir [117]:

Adım 1: Ebeveyn 1'in ilk geninden başlayarak genler arasındaki ilişkinin döngüsü kurulur.

Adım 2: Döngü içindeki genler Ebeveyn 1'den alınarak yavrunun aynı pozisyonlarına yerleştirilir.

Adım 3: Döngüdeki genler Ebeveyn 2'den silinir.

Adım 4: Ebeveyn 2'deki kalan genler sırayla yavruya yerleştirilir.

Doğrusal sıralı çaprazlama (Linear order crossover operator - LOX)

Sıralı çaprazlama (OX) yönteminin farklı bir varyantı olarak Falkenauer ve Bouffouix [118] tarafından önerilmiştir. Adımları şu şekildedir:

Adım 1: Ebeveynlerde rasgele iki çaprazlama noktası belirlenir.

Adım 2: Ebeveyn 2'nin iki çaprazlama noktası arasındaki gen içerikleri Ebeveyn 1'den silinir.

Adım 3: Ebeveyn 1'de kalan genler, sırası değişmeden boşlukları dolduracak şekilde merkezden uçlara doğru kaydırılır.

Adım 4: Ebeveyn 2'nin çaprazlama noktaları arasındaki genler Ebeveyn 1'e aktarılarak yavru üretimi tamamlanır.

Ebeveynlerin rolleri değiştirildikten sonra ikinci yavru üretimi için aynı prosedür uygulanır.

Kısmi eşleştirmeli çaprazlama (Partially matched crossover- PMX)

Kısmi eşleştirmeli çaprazlama yöntemi 1985 yılında Golberg ve Lingle [119] tarafından önerilmiştir. Adımları şu şekildedir:

Adım 1: Ebeveynlerde rasgele iki çaprazlama noktası belirlenir.

Adım 2: Çaprazlama noktaları arasında kalan genler karşılıklı yer değiştirilerek, çaprazlama noktaları dışındakiler ise doğrudan aynı konumlara kopyalanarak yavruya aktarılır.

Adım 3: Ebeveynlerde tekrar eden genler, çaprazlama noktaları arasında bulunan genin diğer yavrudaki karşılığı ile değiştirilir.

Ebeveynlerin rolleri değiştirildikten sonra ikinci yavru üretimi için aynı prosedür uygulanır.

Mutasyon

Genetik algoritmanın bir diğer önemli operatörü olan mutasyon, çaprazlama işleminden sonra üretilen yavrlara uygulanarak yeni bireylerin oluşması sağlanmaktadır. Bu sayede popülasyona yeni genetik bilgiler eklenerek çeşitliliğin artması ve algoritmanın çözüm uzayında yeni alanlara yönelmesi sağlanmaktadır. Ayrıca, mutasyon algoritmaları yerel optimumdan kurtulma ve yeni çözümlere ulaşabilme özelliği getirmektedir.

Mutasyon işlemi önceden belirlenmiş bir olasılığa (P_m) göre rasgele uygulanır. Bu olasılık popülasyonun genetik bilgisinin çok fazla değişmemesi, bir başka deyişle ebeveynlerden gelen kalıtsal özelliklerin kaybolmaması için bu olasılık düşük tutulur.

Kromozomların mutasyona uğratabilmek için çeşitli mutasyon yöntemleri bulunmaktadır. Bu yöntemler şu şekilde tanımlanmıştır:

Karşılıklı değişim mutasyonu: Kromozom dizisi içerisinde iki gen rassal olarak seçilir ve karşılıklı olarak yer değiştirilir.

Ters Mutasyon: İkili kodlama ile oluşturulmuş bir kromozom dizisinde, rasgele seçilen bir genin 1 olan değer 0 veya 0 olan değer 1 olarak tersine çevrilir.

Ekleme mutasyonu: Kromozom dizisinde bir gen rasgele seçilir ve rasgele bir pozisyona yerleştirilir.

Komşu yer değiştirme mutasyonu: Bu yöntemde, kromozomda komşu iki gen seçilir ve kendi aralarında yer değiştirir.

2.3.7. Durdurma kriteri

Literatürde arama sonlandırma kriteri, yakınsama kriteri ya da tamamlama kriteri olarak da adlandırılan durdurma kriteri, GA'daki işlem döngüsünün hangi koşullarda sonlandırılması gerektiğini belirleyen bir mekanizmadır. Probleme özgü başarı veya başarısızlık durumlarında devreye giren bu mekanizma, aşağıda belirtilen farklı kriterlere göre kullanılır. Bu kriterler şu şekildedir:

1. Uygunluk eşiği: Hedeflenen en iyi uygunluk değerine ulaşılması durumunda GA'nın araması sonlandırılır.
2. Zaman sınırı: GA'nın çalışması için belirlenen maksimum süresinin tamamlanmasıyla birlikte algoritmanın çalışması durdurulur.
3. Jenerasyon sayısı: Kullanıcı tarafından belirlenen jenerasyon sayısına ulaşıldığı anda arama sonlandırılır ve bireylerdeki iyileşme gözlemlenir.
4. Yakınsama: Popülasyondaki en iyi olan bireyin veya tüm bireylerin uygunluk değerinde belirli bir jenerasyon sayısı sonrasında herhangi bir değişimin gözlenmemesi durumunda arama sonlandırılır.

2.3.8. Genetik algoritma çeşitleri

Genetik Algoritma (GA), problem alanında verimli ve etkili arama yapabilmesiyle birçok araştırmacı tarafından kullanılmaktadır. Günümüzde mühendislik, iş, bilim ve akademik dünyada sıklıkla karşılaştığımız problemlere bu algoritma sayesinde hızlı çözümler aranmaktadır. Geçmişten günümüze GA'nın çözüm için birçok kez tercih edilmesinin nedeni, hesaplama açısından daha az karmaşık olmasının yanında, algoritmanın iyileştirilmesinin mümkün olabilmesidir. Basit genetik algoritmanın operatör ve süreçlerinde yapılan değişikliklerle paralel genetik algoritma, hibrit genetik algoritma, mikro genetik algoritma gibi GA çeşitleri geliştirilmiştir.

Basit genetik algoritma

Holland'ın geliştirdiği algoritma olarak bilinmektedir. GA'nın yaygın olarak kullanılan çeşidi olmakla birlikte, basit genetik algoritma (BGA) veya temel genetik algoritma (TGA)

olarak adlandırılır. Şekil 2.14’de gösterildiği gibi basit genetik operatörler ve süreçlerden oluşmaktadır.

Paralel genetik algoritma

Birden fazla BGA’nın birlikte çalıştırılmasıyla oluşturulmuş bir GA çeşididir. Paralel genetik algoritmalar (PGA), geniş arama uzaylarında optimuma yakın çözümleri elde edebilmek ve basit genetik algoritmalarla ilişkili uzun işlem sürelerini kısaltmak için geliştirilmiştir. PGA’lar performans ve ölçeklenebilirlik açısından önemli kazanımlara sahiptir ve popülasyonun alt popülasyonlara bölünerek, her bir popülasyonun farklı bilgisayar ağlarında veya paralel ana bilgisayarlarda çalıştırılarak uygulanabilmektedir.

GA’ların paralelleştirilme şekli aşağıdaki unsurlara bağlıdır [111]:

- Uygunluğun değerlendirme ve mutasyonun uygulanma şekli,
- Seçimin yerel ya da genel uygulanması,
- Tekli veya çoklu alt popülasyonların kullanılması,
- Birden fazla popülasyon kullanılıyorsa, bireylerin nasıl değiştirildiği

Melez (hibrit) genetik algoritma

Basit genetik algoritmadaki çaprazlama operatörüyle üretilen yeni popülasyonun gelişiminin çeşitli sezgisel yöntemlerle birleştirilerek tasarlanan algoritmaya melez genetik algoritma (MGA) denir. Bu sezgisel yöntemlerden biri, başlangıç popülasyonunu oluşturmak için uygulanır. Diğer ikisi ise çaprazlama veya karıştırma ile elde edilen yavrulara uygulanmaktadır. Yavrulara uygulanan son iki sezgisel yöntem açgözlüdür, bu nedenle yerel optimuma takılmayı önlemek için karıştırma operatörünü kullanarak uygun miktarda rastgelelik içermektedir [111].

Mikro genetik algoritma

BGA’larda büyük popülasyonların nesilden nesile uygunluk fonksiyonlarının hesaplanmasında çalışma zamanının çok uzun olması bir dezavantaj olarak karşımıza çıkabilmektedir. Mikro Genetik Algoritmalarda (μ GA), popülasyon 5-6 bireyden oluşacak şekilde oluşturulur ve belirli aralıklarda popülasyondaki en iyi birey dışındaki tüm bireyler

tekrar rasgele üretilerek arama işlemi devam ettirilmektedir [120]. Ayrıca algoritmanın belirli bir sürede çözüm geliştirme işlemini tamamlamış olması gerekmektedir. Bu işlemi, popülasyon tabanlı algoritmalar, iteratif algoritmalarla göre daha uzun sürelerde yapmaktadır. Bu sürenin kısalması için az popülasyonla çalışan ve iyi performans gösteren genetik algoritmaların geliştirilmesi önemli hale gelmiştir[121]

2.4. Karar Destek Sistemleri

Günümüzde rekabet koşullarının zorlaşması sebebiyle, işletmelerin ayakta kalabilmeleri için tutarlı ve isabetli kararlar almaları gerekmektedir. İşletmelerin menfaatine olan bu kararların en uygun zamanda alınması işletmelerin geleceği için hayati önem taşımaktadır. Bu sebeple, önemli kararların zamanında alınabilmesi için hızlı erişilebilen, güvenli, çok sayıdaki bilgi yığınları arasından iyi analiz edilmiş özet bilgiye ihtiyaç duyulmaktadır. Bu özellikteki bilgilere kolay erişim ancak bilgi teknolojileri destekli karar destek araçlarıyla mümkündür. Bu tür araçlara genel ifadeyle karar destek sistemleri (KDS) denir. KDS, insan zekâsının zorlanacağı yarı yapısal veya yapısal olmayan karmaşık problemlere hızlı çözümler bulmak amacıyla, etkin veri analizi ve modellemeyi gerçekleştirerek, karar vericilerin karar verme sürecine destek olan bilgisayar temelli bilgi sistemleridir.

2.4.1. Karar verme ve karar verme süreci

İnsanoğlu, aile ya da iş ortamında karşılaştığı problemleri çözmek için var olan çözümler arasından birini seçmeye bir başka deyişle uygun çözüm için bir karar vermeye yönelmektedirler. Yerinde ve uygun kararlar insanoğlunun yaşamında olumlu değişimlere neden olurken, yerinde olmayan hatalı kararlar ise yaşamını olumsuz yönde etkilemektedir.

Karar verme kısaca, var olan bir problemin çözüm alternatifleri arasından birini veya birkaçını tercih etme eylemi olarak tanımlanabilir. Sorunu çözmeye yönelik yapılan bu eylemin uygun olup olmadığı tespit edebilmek için bazı aşamalardan geçmesi gerekmektedir. Karar verme sürecindeki aşamaları şu şekilde sıralayabiliriz [122]:

- Amaç ve problemin belirlenmesi,
- Amaç ve problemle ilgili verilerin toplanması,
- Toplanan verilerin analiz edilmesi,

- Çözüm için alternatiflerin belirlenmesi ve karşılaştırılması,
- Alternatiflerden uygun olan bir veya birkaçının seçilmesi,
- Kararın uygulanması,
- Kararın denetimi ve takibi,
- Kararın değerlendirilmesi,

2.4.2. Tanım ve kavramlar

1960'larda araştırmacılar, karar verme ve planlamaya yardımcı olmak için bilgisayarlı modellerin kullanımını sistematik olarak incelemeye başladılar. KDS ile ilgili tarihsel dönüm noktası ise 1971 yılında Scott Morton tarafından yapılan çalışmayla [123] olmuştur. “Yönetim Karar Sistemleri” adlı çalışmada Scott Morton, bilgisayarların ve analitik modellerin yöneticilere önemli iş planlama kararları vermeleri konusunda nasıl yardımcı olabileceğini incelemiştir.

İlk başlarda pek çok araştırmacı, KDS'nin öncüllerini oluşturan veri izleme ve yönetim bilgi sistemlerinden olan farklılıkları üzerinde durmuştur. KDS'nin kavramsal olarak tanımlanması ise, bilgisayar bilimleri, yönetim bilimleri, yöneylem araştırmaları gibi farklı disiplinlere mensup araştırmacıların katkılarıyla olmuştur. Sistemin gelişiminde bilgisayar yazılım ve donanım teknolojisindeki ilerlemeler, karar alma işlevine etkililik kazandırma çabalarına artan ilgi, tam, doğru, güvenilir bilgiye olan arzu, şiddetli rekabet baskısı ve öncü üniversitelerde yapılan araştırmalar önemli rol oynamıştır [124].

KDS genellikle küçük, basit ve bilgisayara dayalı, gerçekte var olan yönetim aktivitelerini matematiksel olarak örnekleyen modeller kullanır. Bu modellerden birinin ya da modele ilişkin girdilerin bir kısmının değiştirilmesi bir yönetsel probleme ilişkin alternatif çözümler sunar. Örneğin bir modeldeki satış fiyatının değiştirilmesi, o ürünün fiyatının gerçekten değişmesine yönelik karlılık değerlerini anlık olarak tespit etme imkânı verebilir [125].

Yönetim kararlarının alınmasında bilgisayar ve bilgisayar destekli bilgi sistemleri en önemli araçtır. Karar vericilere alacakları kararlarda destek sağlayan karar destek sistemleri, bilgisayar tabanlı bir sistem olarak ele alınabilir. Bu sistem işletme geneline yayılmış, tüm kullanıcı gruplarının yerel ağla veya bir sunucuyla web tabanlı olarak birbirine bağlandığı

bir sistem olabileceği gibi, bir yöneticinin odasındaki tek bir bilgisayarda çalışmak üzere tasarlanmış bir program da olabilir [126].

KDS'nin tanımı ilk defa Scott Morton ve Peter [127] tarafından 1978 yılında yapılmıştır. Bu tanımlamaya göre karar destek sistemi, yarı yapısal ve yapısal olmayan sorunların çözümünde karar alıcıya, veri ve modeller kullanmak suretiyle yardımcı olan etkileşimli bilgisayar sistemleridir [124]. KDS ile ilgili ilk tanımın yapıldığı tarihten günümüze birçok farklı tanımlamalar yapılmıştır. Bu tanımlamaların bir kısmı şu şekildedir:

KDS, kuruluşlardaki yöneticilerin ve diğer çalışanların karar alma faaliyetlerini desteklemek için hareket işleme sistemlerini (transaction processing systems) kullanan ve bilgi sistemlerinin tüm kısımlarıyla etkileşime giren bir tür bilgi sistemidir [128].

KDS, yarı yapılandırılmış ve yapılandırılmamış durumlarda veya ne yönde bir karar verilmesinin tam olarak kestirilemediği hallerde, karar vericilere modeller, bilgiler ve veri yönetme araçları sunan interaktif bilgi sistemleridir [129].

KDS, özellikle yapısal olmayan yönetim problemlerinin çözümünü desteklemede karar vermeyi kolaylaştırmak için geliştirilmiş etkileşimli, kullanışlı ve yeni koşullara uyarlanabilen bilgisayar destekli bir bilgi sistemidir [130].

KDS, değişik kaynaklardan topladığı bilgileri düzenleyerek, kararı modelleyerek, bilgileri analiz ederek ve değerlendirme sonuçlarını sunarak, belirli modeller kullanımı ile karar vericiye seçim sırasında destek veren bilgisayar temelli bir sistemdir [131].

KDS'ye sahip karar vericiler, her sorun için çözüm seçeneklerini formüle eder ve bilgisayara gönderir. Bilgisayar bu önerileri karşılaştırarak değerlendirir ve karar vericiye yollar. Karar verici de değerlendirilen öneriler arasında en iyi sonucu veren alternatifi seçer ya da yeni bilgilere göre yeni alternatifler hazırlayarak tekrar bilgisayarın değerlendirmesine sunar [132].

2.4.3. KDS'nin temel bileşenleri

Tipik bir KDS, Şekil 2.26'da gösterildiği üzere diyalog, model ve veri yönetimi olmak üzere üç temel bileşene sahiptir [122].

Diyalog Yönetimi

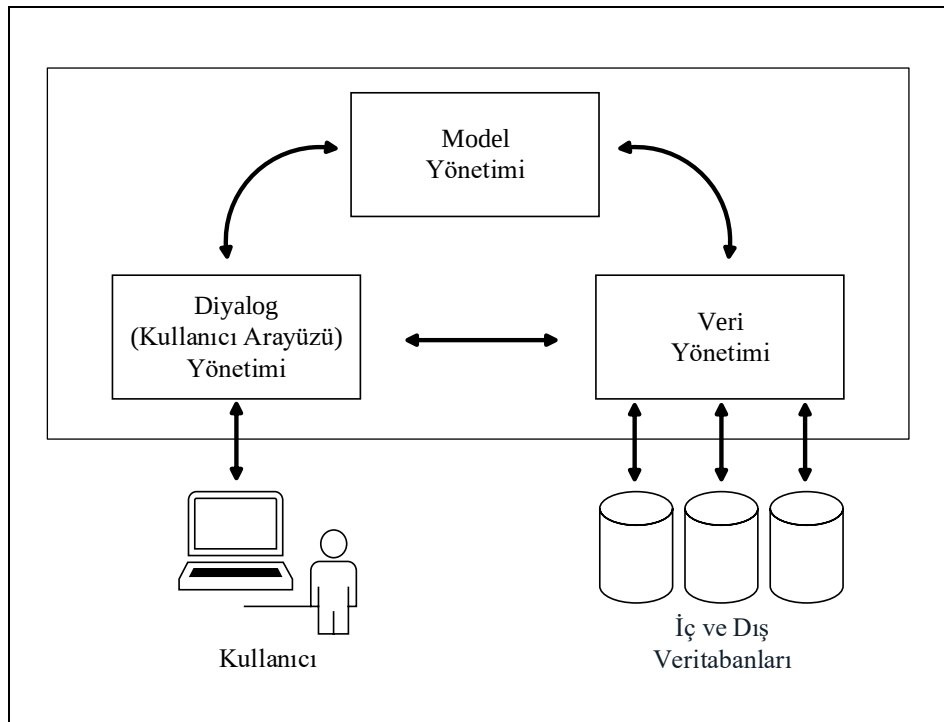
Kullanıcı arayüzü olarak da adlandırılan bu bileşen, kullanıcının KDS'ye erişimini sağlamaktadır. İyi tasarlanmış bir arayüz, KDS'nin yazılımı ve donanımıyla iletişim kurarak karar vericinin ihtiyacı olan verilere ve modellere kolaylıkla ulaşabilmektedir. Bu sayede kullanıcının karar verme başarısı artmaktadır.

Veri yönetimi

Birçok farklı uygulamadan elde edilen ve karar verilecek konularla ilgili verileri içerisinde barındıran veritabanlarından oluşur. Bu veritabanları bir veritabanı yönetim sistemi (VTYS) uygulaması tarafından yönetilir ve karar vericinin ihtiyacı olan verilerin çağırılması, depolanması, işlenmesi, dönüştürülmesi gibi işlemler gerçekleştirilmektedir.

Model yönetimi

KDS'de gerekli olan finansal, istatistiksel, yönetim bilimi, yöneylem araştırması veya analitik yetenek sağlayan sayısal modellerin saklama, depolama gibi organizasyonel aktivitelerin gerçekleştirildiği bir KDS bileşenidir.



Şekil 2.26. KDS'nin temel bileşenleri

2.4.4. KDS'nin özellikleri, faydaları ve kısıtları

Genel olarak ifade edilen KDS'nin özellikleri aşağıda sıralanmıştır [122]:

1. Geleceği planlamaya yöneliktir.
2. Yarı-yapısal ve yapısal olmayan kararlarda kullanılır.
3. Karar vericinin yerine geçmekten ziyade, ona karar vermesinde yardımcı olur.
4. Karar verme sürecinin tüm aşamalarını destekler.
5. Kullanıcının kontrolü altındadır.
6. Veri ve model tabanlarına erişimlidir.
7. Veri inceleme ve çözüm üretmede analitik modeller kullanır.
8. Kullanıcı etkileşimlidir. Bu nedenle karar verici, bir YBS uzmanından çok az yardım alarak ya da almadan kullanabilir.
9. Yoğun olarak stratejik ve taktik düzeydeki yöneticiler için, gerektiğinde düzeyler arası bütünleşmeye de destek vererek, karar verme desteği sağlar.
10. Birden fazla bağımsız ya da birbirine bağımlı kararlar için destek sağlayabilir.
11. Bireysel, grup tabanlı karar verme desteği sağlar.
12. Kullanım kolaylığı sağlar.
13. Değişen şartlara ve karar durumlarına uyum sağlayabilecek esnekliktedir.
14. Düzensiz ve planlanmamış zaman aralıklarında kullanılabilir.

KDS'nin faydaları şu şekildedir [133]:

1. Karar vericinin bilgiyi işleme kabiliyetini geliştirir.
2. Karar vericinin zaman alan, büyük çaplı karmaşık problemlerin üstesinden gelme kabiliyetini geliştirir.
3. Karar vermedeki gerekli olan zaman kısalır.
4. Karar sürecindeki sonucun güvenilirliği artar.
5. Karar verici tarafından araştırma ve keşfi teşvik eder.
6. Bir problem alanı veya karar bağlamı hakkında düşünmeye yönelik yeni yaklaşımlar ortaya çıkarır.
7. Bir kararı desteklemek veya var olan bir varsayımı onaylamak için yeni kanıtlar oluşturur.
8. Rakiplere göre rekabet gücünü artırır veya stratejik avantaj sağlar.

KDS, karar vericilerin kendi özel koşul ve tercihlerini anlamalarında oldukça değerli yardımcılardır. Birçok uygulamaya karşın karar destek sistemlerinin uygulamadaki başarıları genelde sınırlıdır [134,135]. Bu durumun başlıca nedenleri şöyle sıralanabilir:

- Karar destek sistemlerinin kullanıcıya yakınlığı yetersizdir. Genel olarak kullanıcılar, uygulama ve bilgisayar sistemi hakkında çok şey bilmek isterler. Ancak, kullanıcının yeterli düzeyde bilgisayar bilgisinin olmaması durumunda bu sistemlerden faydalanma sınırlı kalabilir.
- Karar destek sistemlerinin fonksiyonelliği ve esnekliği kısıtlıdır.
- Karar destek sistemlerinde çözüm için bir açıklama mekanizması yoktur.

KDS'nin kısıtları farklı bir bakış açısıyla şu şekilde sıralanmaktadır [133]:

- Karar destek sistemleri yaratıcılık, hayal gücü, sezgi gibi insani karar verme yetenekleriyle tasarlanamamaktadır.
- Karar destek sisteminin gücü kullanıldığı bilgisayar sistemi, tasarımı ve sahip olduğu bilgi ile sınırlıdır.
- Dil komut ara yüzleri kullanıcıya direktif ve sorgulamalarında doğal bir dil işlevi sağlayacak kadar gelişmiş değildir.
- Karar destek sistemleri normal olarak uygulama alanında sınırlı kalacak şekilde tasarlanırlar.

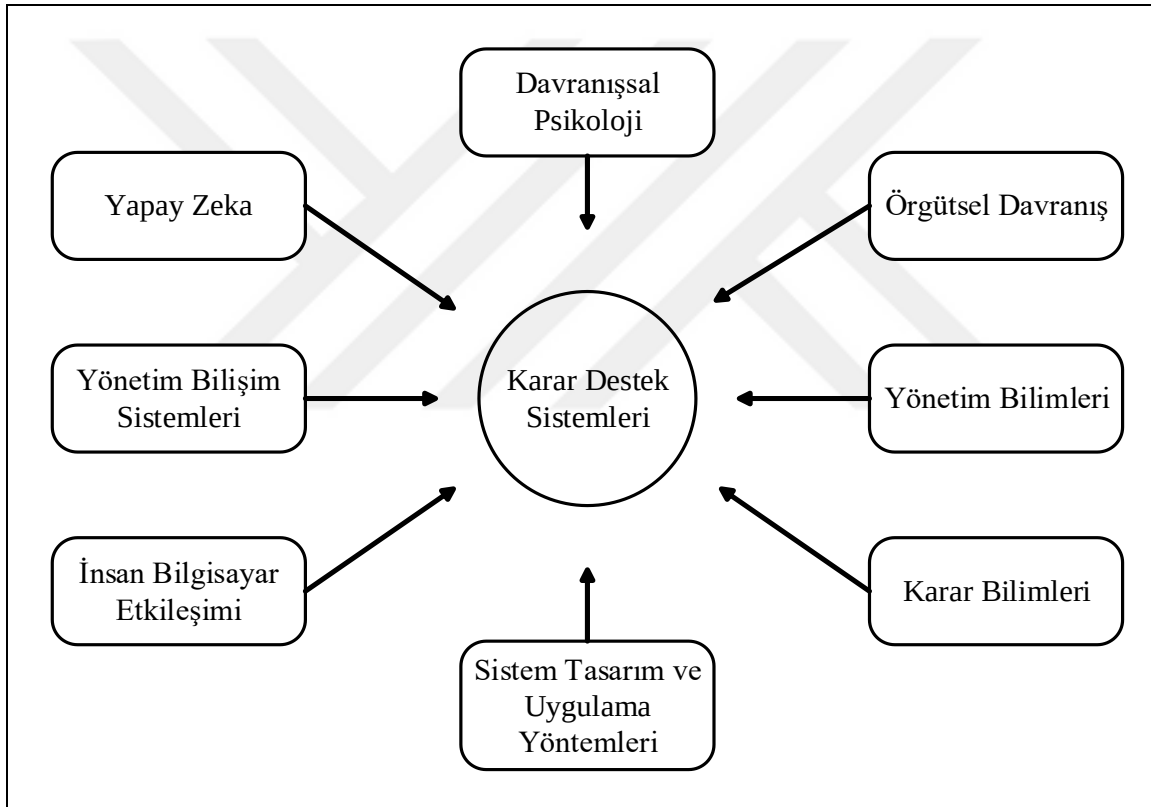
2.4.5. KDS'nin kullanım alanları

Son yıllardaki teknolojik gelişmeyle birlikte karar verici iş süreçlerinin başından sonuna kadar daha hızlı ve anlık veri elde edilebilmektedir. Örneğin, nesnelerin interneti, akıllı mobil cihazlar, robotik sistemler, insansız araçlar gibi ileri teknolojinin kullanımı sayesinde karar vericiler çok daha nitelikli veriler elde etmektedir. Bu sayede sürecin içinde kalan belirsiz noktalar azalmakta ve süreç sonunda daha tutarlı ve isabetli kararlar ortaya çıkmaktadır. Hatta sağlanan bu nitelikli veriler yapay zekâ teknikleriyle desteklenerek karar vericinin çözüm alternatifleri daha da çeşitlenebilmektedir.

KDS'lerinin geniş bir kullanım alanı bulunmaktadır. Bir kurumun ihtiyaç duyduğu herhangi bir alanda geliştirebilir. Pazarlama, tedarik zinciri yönetimi, lojistik, finansman-yatırım

kararları, operasyon yönetimi, sağlık, güvenlik, orman yönetimi, askeri uygulamalar, ulaştırma, bankacılık, sigorta, personel yönetimi, yol güvenliği, adli tıp, ulaştırma sistemleri vb. pek çok alanlarda KDS uygulamalarına rastlamak da mümkündür [136].

KDS birçok alandan kullanılmasının yanında, birçok bilim dalı ile ilgili olduğu görülmektedir. Şekil 2.27’de görüldüğü üzere KDS teoride, insan bilgisayar etkileşimi, psikoloji, örgütsel davranış, yapay zeka ve davranışsal psikoloji bilimi gibi araştırma alanları dahil olmak üzere KDS'nin tasarımı ve uygulanmasıyla ilgili olan disiplinler kümesi oldukça geniştir [137].



Şekil 2.27. KDS'nin ilgili olduğu bilim dalları [137]



3. MATERYAL, YÖNTEM VE UYGULAMA

Bu çalışmada, iki boyutlu yapıların daire palet üzerine dairelerin yüklenebilmesi için bir karar destek modeli önerilmiş ve genetik algoritma tabanlı bir sistem geliştirilmiştir. Geliştirilen sistemde dairelerin palet üzerindeki yerleşiminin belirli bir sistematik üzerinde gerçekleştirilebilmesi için bir yerleşim algoritması tasarlanmıştır. Tasarlanan yerleşim algoritmasıyla genetik algoritmasının başlangıç çözümü elde edildikten sonra genetik operatörlerle iyileştirmeler yapılmaktadır. Amaç fonksiyonu esas alınarak yapılan bu iyileştirmeler sayesinde optimal çözüme ulaşılması amaçlanmaktadır.

3.1. Problem Tanımı

Paletlerin veya paketleme yapılacak alanların geometrik yapısı ülkelerin nakliye standartları, yüklenecek ürünlerin özellikleri, yükleme problemindeki kısıtlara bağlı olarak farklılıklar gösterebilmektedir. Literatürde yapılan çalışmalar incelendiğinde paletlemenin çoğunlukla dikdörtgen, kare ve daire şeklindeki yükleme alanları üzerinde yapıldığı görülmektedir. Bununla beraber dik üçgen, eşkenar üçgen, çeyrek veya yarım daire, elips ve L tipi şeklindeki alanlara da yerleşimlerin yapıldığı çalışmalara da rastlamak mümkündür.

Bu tezde ele alınan problem, temel olarak kesme ve paketleme problemlerinden olan daire paketleme problemidir. Problem, özdeş veya özdeş olmayan n adet dairenin belirli amaçlar ve kısıtlar doğrultusunda çözüm alanına yerleştirilmesini içermektedir. Çözüm alanı, sabit ölçüdeki daire paletlerden oluşmakta ve yerleştirilecek n adet daire arasından uygun olanın palete yerleştirilmesine odaklanılmaktadır. Dairelerin palet üzerine yerleşiminin uygun bir şekilde yapılabilmesi için kısıt kontrolleri ve bununla beraber bir matematiksel hesaplama gerektirmektedir. DPP’de dairelerin, daire şeklindeki palet içine yerleştirilmesi probleminin amaç fonksiyonları ve kısıtları Bölüm 4.1.1’de, matematiksel modeli Bölüm 4.1.2’de sunulmaktadır.

3.1.1. Problemin amaç fonksiyonları ve kısıtları

Bu çalışmada, DPP’nin çözümü için iki amaç fonksiyonu ve iki temel kısıt ortaya konmaktadır. Sabit ölçüdeki daire paletlere n adet daire arasından uygun olanların yerleştirilmesinde temel amaç, palet kullanımında en büyük faydanın sağlanabilmesidir. Bu

sebeple ele alınan problemin amaç fonksiyonları çıktının en büyüklendiği problemler olarak şu şekilde belirtilmiştir:

1. Alan maksimizasyonu
2. Daire sayısı maksimizasyonu

Alan maksimizasyonu amaç fonksiyonunda sabit ölçüdeki palete yerleştirilen dairelerin palet üzerinde kaplandığı alanın en fazla olması hedeflenmektedir. Bu sayede palet üzerinde atıl olan atık alan azalmakta ve paletin kullanım alanı artmaktadır. Daire sayısı maksimizasyonunda ise palet içine yerleştirilen dairelerin sayısının mümkün olan en çok sayıda olması gerekmektedir. Bir başka deyişle, yerleştirilmesi hedeflenen n adet dairenin arasından mümkün olan en fazla sayıdaki dairenin sabit boyutlu palete yerleştirilmesine odaklanılmaktadır.

İki boyutlu palet yükleme ve paketleme problemlerinde dikkate alınan kısıtlar temel olarak yüklenecek olan parçaların birbiriyle üst üste binmeyecek şekilde konumlandırılması ve parçaların yükleme alanı dışına taşmayacak şekilde bütünüyle yerleştirilmesidir. Bu çalışma kapsamında dairelerin yerleşimi için temel iki kısıtla birlikte probleme özgü üçüncü bir kısıt ortaya konmuştur. Kısıtlar şu şekildedir:

1. Dairelerin palet dışına taşmaması
2. Dairelerin birbiriyle çakışmaması
3. Dairelerin en az iki noktadan teğet olması

İlk dairenin palet içine yerleştirilmesinden sonraki her daire yerleştirme işlemi en az iki noktadan teğet olacak şekilde olmalıdır. Bu iki nokta teğetliği “daire-daire” ya da “daire-palet” şeklinde olabilmektedir. Bu sayede her daire

Belirtilen kısıtların ihlal edilmeden kontrolü için probleme özgü matematiksel modeli bir sonraki bölümde sunulmuştur.

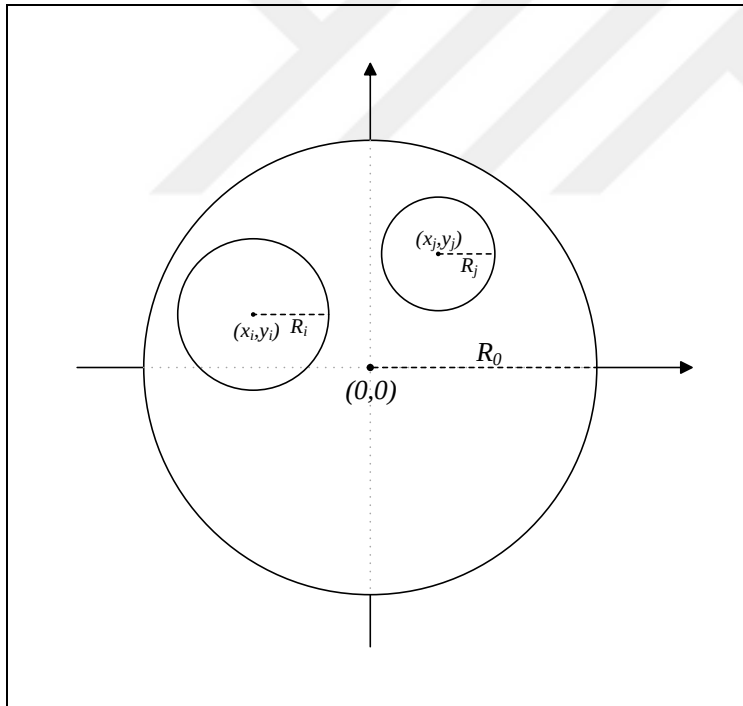
3.1.2. Problemin matematiksel modeli

Bu bölümde problemin çözümü için gerekli matematiksel yapı detaylı bir şekilde açıklanmıştır. Daire palette iki boyutlu DPP'nin çözümü için sunulan matematiksel modele göre, paletin merkez noktası Öklid düzleminde $(0,0)$ olarak kabul edilir. Bu merkez noktasına göre, yerleşimi yapılan her dairenin koordinatlarının hesaplanması, çakışma

kontrolü, paletten taşıma kontrolü gibi hesaplamalar bu matematiksel modelde detaylı açıklanmıştır.

Şekil 3.1’de gösterildiği üzere, R_0 yarıçaplı palet içine yerleştirilen R_i ve R_j yarıçaplı i ve j dairelerinin konumlandırılması gösterilmiştir. Buna göre;

- (i) paletin yarıçapı: R_0
- (ii) paletin merkez noktası (orijin): $(0,0)$
- (iii) yerleşmeye aday n adet dairenin yarıçapı: R_i ($i=1, 2, \dots, n$)
- (iv) yerleşmiş olan i dairesinin yarıçapı ve koordinatı: $R_i, (x_i, y_i)$
- (v) yerleşmiş olan j dairesinin yarıçapı ve koordinatı: $R_j, (x_j, y_j)$



Şekil 3.1. Daire paletin koordinat gösterimi

Lopez ve Beasley [79] tarafından DPP’nin çözümü için önerilen matematiksel modele göre palet üzerine yerleşmeye aday n adet özdeş olmayan daireler, yarıçaplarına (R_i) göre küçükten büyüğe sıralanarak ($R_i < R_{i+1}$ $i=1, \dots, n-1$) numaralandırılır. Bu daireler arasından amaç fonksiyonuna ve kısıtlara göre uygun olan dairenin seçimi yapılarak palet içine uygun bir yerleşim gerçekleştirilmektedir. DPP’nin çözümü için iki amaç fonksiyonu ve iki temel

kısıt ortaya konmaktadır. Eş. 3.1'deki V_i değerine göre farklılaşmaktadır. Alan maksimizasyonuna göre değer $V_i = \pi R_i^2$ olarak hesaplanırken, daire sayısı maksimizasyonunda ise $V_i = 1$ olmaktadır.

Yerleşme sırası gelen R_i yarıçaplı dairenin palette konumlandırılması için iki temel kısıt kontrolünün yapılması gerekmektedir. Palet dışına taşmama kısıtının kontrolü Eş. 3.2 ile sağlanırken, daire çiftlerinin birbiriyle çakışmama kısıtı ise Eş. 3.3 ile kontrol edilmektedir. Eş. 3.4'de gösterildiği üzere, kısıtlara göre yerleşmesi uygun olan dairelerde $\alpha_i = 1$, yerleşmesi uygun olmayan daireler için ise $\alpha_i = 0$ olarak Eş. 3.1'e katılmaktadır.

$$\max \sum_{i=1}^n \alpha_i V_i \quad (3.1)$$

$$x_i^2 + y_i^2 \leq \alpha_i (R_0 - R_i)^2 \quad i = 1, \dots, n \quad (3.2)$$

$$(x_i - x_j)^2 + (y_i - y_j)^2 \geq \alpha_i \alpha_j (R_i + R_j)^2 \quad \forall (i, j) \in Q \quad j > i \quad (3.3)$$

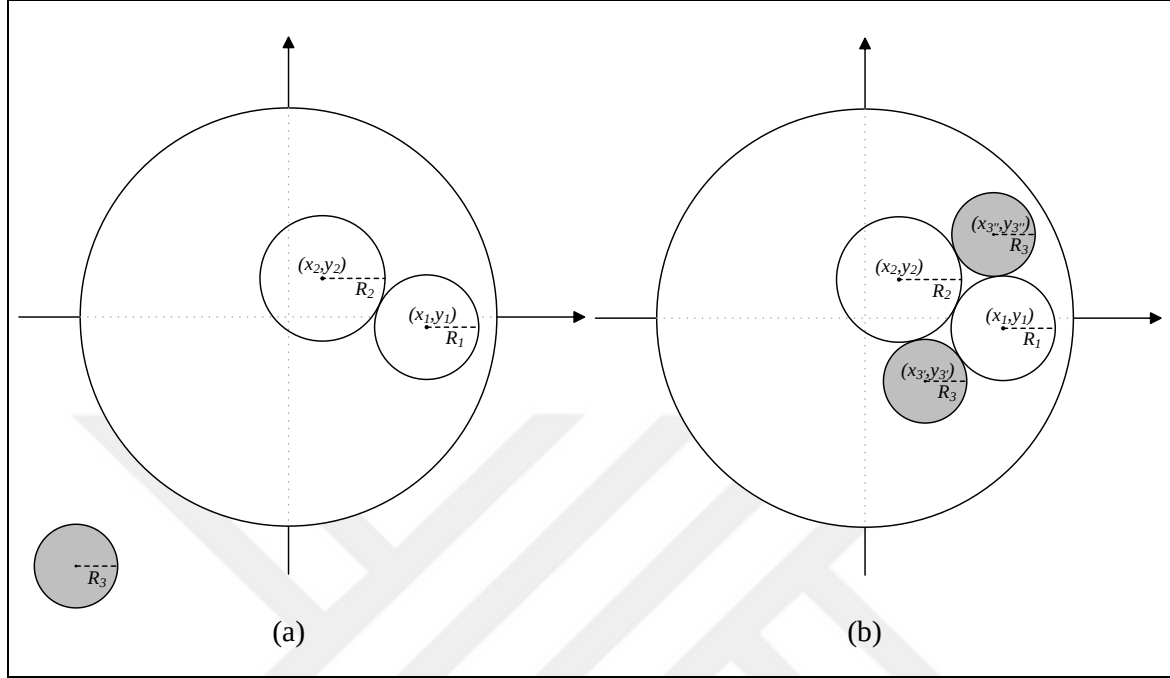
$$\alpha_i = 0 \text{ veya } 1 \quad i = 1, \dots, n \quad (3.4)$$

Önerilen yerleşim algoritmasına göre yerleşme sırası gelen ve yarıçapı bilinen her hangi bir dairenin koordinatlarının hesaplanabilmesi için en az iki noktada teğetlik oluşturması gerekmektedir. Belirtilen iki noktadaki teğetlik için iki farklı yerleşim durumu bulunmaktadır. Bu durumlardan birincisi iki daireye birden teğet olabilme; ikinci durum ise bir daire ve paletin iç noktasına teğet olabilme durumudur. Belirtilen iki durum için yerleşecek olan dairenin koordinatlarının tespiti için bir takım matematiksel modeller önerilmiştir. Bu modeller DPP'nin çözümü için tasarlanan yerleşim algoritmasında ve önerilen çözüme katkı sunmaktadır.

Daire yerleşim durumu-1: İki daireye teğet olacak bir dairenin yerleşmesi

Şekil 3.2(a)'da yarıçapları ve palet içi koordinatları bilinen R_1 ve R_2 daireleri ile yerleştirilmeye aday yarıçapı bilinen R_3 dairesi görülmektedir. R_3 dairesi, R_1 ve R_2 dairelerinin her ikisine de teğet olarak konumlandırılması gerektiğinde Şekil 3.2(b)'de görüldüğü gibi iki farklı konum mümkün olmaktadır. Görülen iki farklı konum arasından taşmama ve çakışmama problem kısıtlarına göre ya da detayları Bölüm 3.3.1'de belirtilen yerleşim algoritması kurallarına göre uygun olan konum seçilebilmektedir. R_3 yarıçaplı

dairenin mümkün olan iki ayrı konumuna göre formüller Eş. 3.5 ve Eş. 3.6'de gösterilmektedir [138].



Şekil 3.2. Yerleşik iki daireye üçüncü dairenin teğet konumlandırılması

$$(x_3', y_3') = \frac{1}{2} (x_1 + x_2, y_1 + y_2) + \frac{(R_1 + R_3)^2 - (R_2 + R_3)^2}{2(R_1 + R_2)^2} (x_2 - x_1, y_2 - y_1) + \quad (3.5)$$

$$\frac{1}{2} \sqrt{2 \frac{(R_1 + R_3)^2 + (R_2 + R_3)^2}{(R_1 + R_2)^2} - \frac{(R_1 + R_3)^2 - (R_2 + R_3)^2}{(R_1 + R_2)^4} - 1} (y_2 - y_1, x_1 - x_2)$$

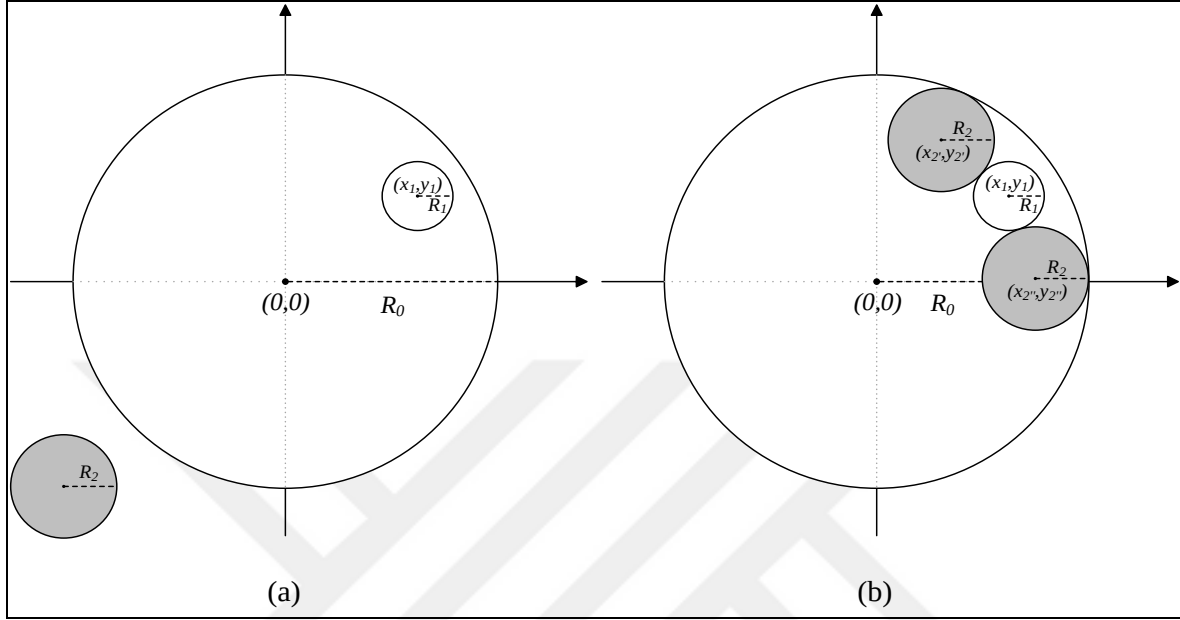
$$(x_3'', y_3'') = \frac{1}{2} (x_1 + x_2, y_1 + y_2) + \frac{(R_1 + R_3)^2 - (R_2 + R_3)^2}{2(R_1 + R_2)^2} (x_2 - x_1, y_2 - y_1) - \quad (3.6)$$

$$\frac{1}{2} \sqrt{2 \frac{(R_1 + R_3)^2 + (R_2 + R_3)^2}{(R_1 + R_2)^2} - \frac{(R_1 + R_3)^2 - (R_2 + R_3)^2}{(R_1 + R_2)^4} - 1} (y_2 - y_1, x_1 - x_2)$$

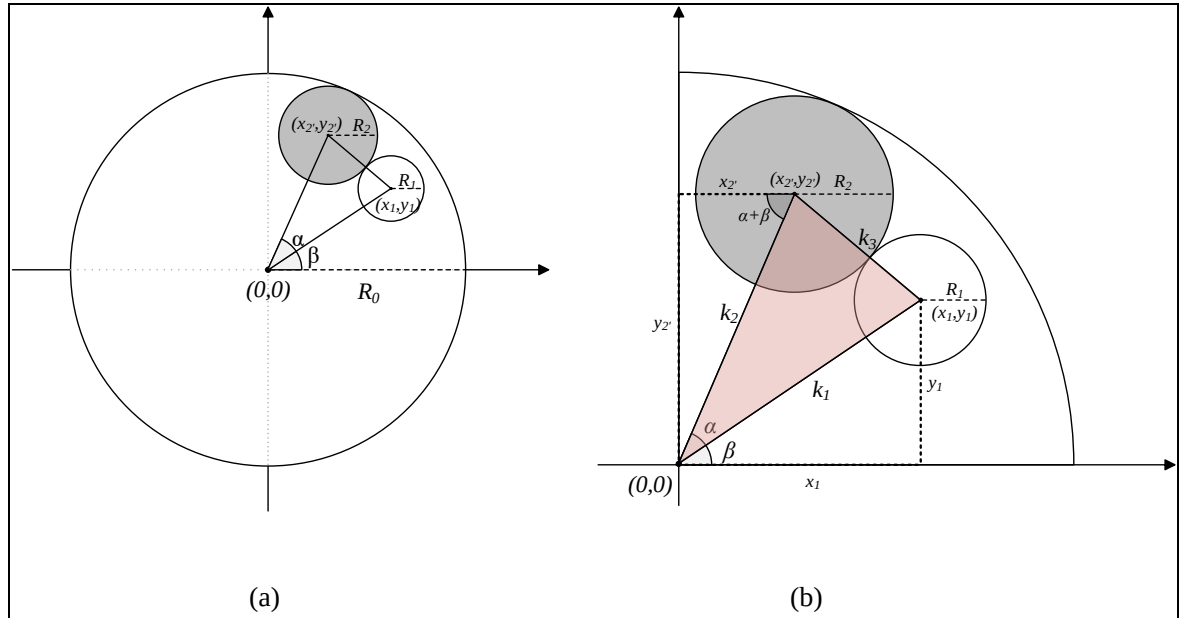
Daire yerleşim durumu-2: Palete ve daireye teğet dairenin yerleşimi

Şekil 3.3 (a)'da yarıçapı ve palet içi koordinatları bilinen \$R_1\$ daresi ile palete iç teğet olacak şekilde, yarıçapı bilinen \$R_2\$ daresi görülmektedir. \$R_2\$ daresinin, \$R_1\$ daresine ve palete iç teğet olarak konumlandırılması gerektiğinde Şekil 3.3 (b)'de görüldüğü üzere görüldüğü gibi

iki farklı konum mümkün olmaktadır. Görülen iki farklı konum arasından taşmama ve çakışmama problem kısıtlarına göre ya da detayları Bölüm 3.3.1’de belirtilen yerleşim algoritması kurallarına göre uygun olan konum seçilebilmektedir.



Şekil 3.3. Palete ve yerleşik daireye teğet olacak şekilde daire konumlandırma



Şekil 3.4. Palete ve yerleşik daireye teğet olan dairenin koordinat tespiti

R_2 yarıçaplı dairenin, palete ve R_1 dairesine teğet olabilecek iki alternatiften biri olan yerleşim Şekil 3.4 (a)'da, yerleşimin genişletilmiş detaylı gösterimi ise Şekil 3.4 (b)'de verilmiştir.

Bu yerleşim durumunda R_2 yarıçaplı daire, R_1 dairesine dıştan ve palete içten teğet olarak konumlandırılmak istenmektedir. Yerleşimden önce R_0 , R_1 , R_2 yarıçap değerleri ve (x_1, y_1) koordinatları bilinmekte ve (x_2, y_2) koordinatlarının tespit edilmesi hedeflenmektedir. Koordinat tespiti için Şekil 3.4 (b)'de gösterildiği gibi geometrik bir yapı ortaya çıkmaktadır.

Palet merkez noktası $(0,0)$ ve R_2 dairesinin merkez noktası arasında çizilen doğrunun, paletin x ve y doğruları ile oluşturduğu dik üçgen çözüm için kritik öneme sahiptir. Bu dik üçgen içindeki $\alpha + \beta$ açısındaki $\cos(\alpha + \beta)$ değerinin hesaplanması ile x_2 ve y_2 değerleri elde edilebilmektedir. x_2 ve y_2 değerlerini Eş. 3.7 ve Eş. 3.8'de belirtilen dik üçgende trigonometrik oran formülü ile bulmak mümkündür. Ayrıca bu dik üçgenin hipotenüsü olan k_2 , Eş. 3.9'da gösterildiği üzere palet yarıçapı ile R_2 daire yarıçapı farkı ile (R_2 dairesi palete iç teğet olduğundan) bulunabilmektedir.

Teğet iki dairenin merkez noktaları arasındaki uzaklık olan k_3 ise Eş. 3.10'da gösterildiği gibi iki dairenin yarıçapı toplamını vermektedir. Ayrıca palet merkez noktası $(0,0)$ ve R_1 dairesinin merkez noktası arasındaki doğru olan k_1 , x_1 ve x_2 ile beraber oluşturduğu dik üçgende hipotenüs olmaktadır. Bu sebeple k_1 uzunluğu, Pisagor Teoremi'ne göre Eş. 3.11'de gösterildiği gibi bulunabilmektedir.

$$x_2 = \cos(\alpha + \beta) k_2 \quad (3.7)$$

$$y_2 = \sin(\alpha + \beta) k_2 \quad (3.8)$$

$$k_2 = R_0 - R_2 \quad (3.9)$$

$$k_3 = R_1 + R_2 \quad (3.10)$$

$$k_1 = \sqrt{x_1^2 + y_1^2} \quad (3.11)$$

$$\cos(\beta) = \frac{x_1}{k_1} \quad (3.12)$$

$$\cos(\alpha) = \frac{k_1^2 + k_2^2 - k_3^2}{2k_1 k_2} \quad (3.13)$$

$$\sin(\beta) = \frac{y_1}{k_1} \quad (3.14)$$

$$\cos(\alpha + \beta) = \cos\alpha \cos\beta - \sin\alpha \sin\beta \quad (3.15)$$

$$\sin(\alpha) = \sqrt{1 - \cos^2\alpha} \quad (3.16)$$

$$x_2 = (\cos\alpha \cos\beta - \sin\alpha \sin\beta) k_2 \quad (3.17)$$

x_2 değerinin bulunabilmesi için Eş. 3.7’de verilen eşitlikte $\cos(\alpha + \beta)$ değerinin yerine Eş. 3.15’deki eşitlik konur. $\cos\alpha$, $\cos\beta$, $\sin\alpha \sin\beta$ değerlerinin açılımları Eş. 3.12, Eş. 3.13, Eş. 3.14 ve Eş. 3.16’deki değerlerinin yerine konmasıyla Eş. 3.17 şekline dönüşerek x_2 değeri bulunmaktadır. y_2 değeri ise Eş. 3.18 dik üçgen formülünde x_2 ve k_2 ’nin yerine konmasıyla bulunmaktadır. Böylelikle üçgende açı, kenar formüllerinin kullanılmasıyla eklenen dairenin (x,y) koordinatları tespit edilmiş olmaktadır.

$$x_2 = \left(\frac{x_1^2 + y_1^2 + (R_0 - R_2)^2 - (R_1 + R_2)^2 x_1}{2x_1^2 + y_1^2 (R_0 - R_2)} - \sqrt{1 - \left(\frac{x_1^2 + y_1^2 + (R_0 - R_2)^2 - (R_1 + R_2)^2 x_1}{2x_1^2 + y_1^2 (R_0 - R_2)} \right)^2} \frac{y_1}{\sqrt{x_1^2 + y_1^2}} \right) (R_0 - R_2) \quad (3.18)$$

$$y_2 = \sqrt{k_2^2 + x_2^2} \quad (3.19)$$

3.2. Çalışmanın Amacı

Palet yükleme ve paketleme problemleri, günlük hayatın birçok alanında karşılaşılan ve çözümü için yoğun çaba harcanan problemlerdendir. Genel olarak bu tür problemler NP-Zor sınıfındaki problemlerden olması sebebiyle daha az kaynak kullanımıyla en iyi veya en iyiye yakın sonuçların makul sürelerde alınması çözümün kalitesi açısından önemlidir. Küçük boyutlu problemler, kesin çözüm yöntemleriyle kısa sürede en iyi sonucu verebilmekte iken, problemin boyutunun ve kısıtlarının artmasıyla çözümün kaynak kullanımı ve hesaplama zamanları da artmaktadır. Bu ise istenmeyen bir durumdur. Yaklaşık çözüm veren metotlar en iyi sonucu garanti etmeseler bile kesin çözüm metotlarının aksine çok daha kısa sürede, düşük kaynak kullanımıyla en iyi ya da en iyiye yakın başarılı sonuçlar elde edilebilmektedir. Yaklaşık çözüm veren algoritmalar, en iyi çözümü bulmak için çözüm uzayında arama yapmaktadır. Etkin ve verimli arama yapabilmek için ise metasezgisel yöntemlerden faydalanılmaktadır. Metasezgisel yöntemler sayesinde arama süreci için etkin stratejiler geliştirilir ve arama uzayının hızlı bir şekilde taranması sağlanır.

DPP'nin daire palet üzerinde çözümünün amaç fonksiyonlarına uygun olarak yapılan bu iyileştirmelerle optimal çözümlere makul sürelerde ulaşılması büyük önem arz etmektedir. Bu çalışma kapsamında daire paketleme problemin amaçları şu şekilde sıralanmıştır:

- Problemin amaç fonksiyonunda belirtildiği gibi paletteki kullanım alanını ve palete yerleştirilecek daire sayısını arttırmak,
- Sabit boyutlu daire paletlere, belirtilen kısıtlar göz önünde bulundurularak yerleşmesi mümkün olan özdeş veya özdeş olmayan dairelerin mümkün olan en kısa sürelerde yüklenmesi amaçlanmaktadır.

3.3. Çalışmanın Yöntemi ve Modelin Geliştirilmesi

Bu çalışmada, DPP'nin çözümü için özgün bir yerleşim algoritması tasarlanarak genetik algoritma tabanlı bir karar destek modeli geliştirilmiştir. Genetik algoritma, NP-Hard sınıfındaki eniyileme problemlerini optimale yakın bir sürede çözebilen metasezgisellerden biridir. Tasarlanan yerleşim algoritması ile dairelerin palet üzerinde taşıma ve çakışma olmaksızın yerleşiminin gerçekleştirilip başlangıç çözümü elde edildikten sonra genetik algoritma operatörleriyle mevcut çözüm üzerinde iyileştirmeler yapılmaktadır.

Bu bölümde problemin çözüm yöntemi dört alt başlık altında toplanarak aşağıda sunulmuştur. Birinci alt başlıkta DPP'nin çözümü için tasarlanan özgün bir yerleşim algoritması anlatılmaktadır. İkinci alt başlıkta, DPP'nin çözümü için probleme özgü geliştirilen genetik algoritmaların ayrıntılarından bahsedilmektedir. Üçüncü alt başlıkta geliştirilen genetik algoritmanın parametrelerinin belirlenmesi ve algoritmanın başarımını arttıracak şekilde seçilmesi için yapılan parametre optimizasyonun detaylarından bahsedilmektedir. Dördüncü alt başlıkta ise DPP çözümü için karar destek sistemi uygulaması anlatılmaktadır.

3.3.1. DPP için tasarlanan yerleşim algoritması

DPP'de, uygun ve sistematik yerleşimin gerçekleştirilebilmesi için, daire paketleme stratejisinin düzgün belirlenmesi ve buna bağlı fonksiyonların oluşturularak, problemin tüm aşamalarında eksiksiz ve sorunsuz çalışan bir algoritmanın tasarlanması çözümün kalitesi açısından büyük önem arz etmektedir. Bu sebeple, “*Komşu ve Konum Seçimli Daire Yerleşim Algoritması*” olarak adlandırılan özgün bir çözüm kurucu algoritma tasarlanmıştır. Bu algoritma ile dairelerin birbirleriyle çakışmama, paletten dışarı taşmama kısıtlarına ek

olarak her bir dairenin sahip olduğu komşuluk ve konum seçim değerlerine göre yerleşim yapılmaktadır. Bu bilgiler ışığında, yerleşim algoritmasının çalışma sistemi aşağıda belirtilen yedi temel kural üzerine oturtulmuştur. Bu kurallar şu şekildedir:

Kural 1: Palete yerleşecek dairelerin yerleşim sırası belirlenmelidir.

Kural 2: Yerleşecek tüm dairelere ve palete benzersiz komşuluk puanı atanmalıdır.

Kural 3: Yerleşecek dairelere konum seçim değeri atanmalıdır.

Kural 4: İlk daire palete göre saat 3 yönünde yerleştirilmelidir.

Kural 5: İlk daireden sonraki daireler en az iki noktadan teğet olacak şekilde yerleştirilmelidir. Bu iki nokta teğetliği “daire-daire” ya da “daire-palet” olabilir.

Kural 6: Her yeni daire yerleştirme işlemi sırasında, dairenin yerleşebileceği çok sayıda alternatif ortaya çıkabilmektedir. Ortaya çıkan alternatifler arasından birinin belirlenmesi komşuluk puanı (KP) ve konum seçim değeri (KSD) mekanizmalarıyla olmalıdır.

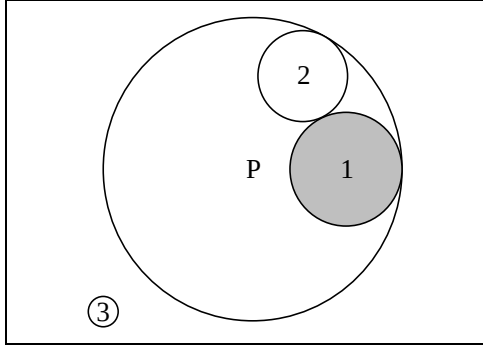
Kural 7: Kural 4, Kural 5, Kural 6’ya göre yerleşmesi mümkün olmayan daire bırakılmalı ve sıradaki dairenin yerleşim işlemi başlatılmalıdır.

Komşuluk puanı

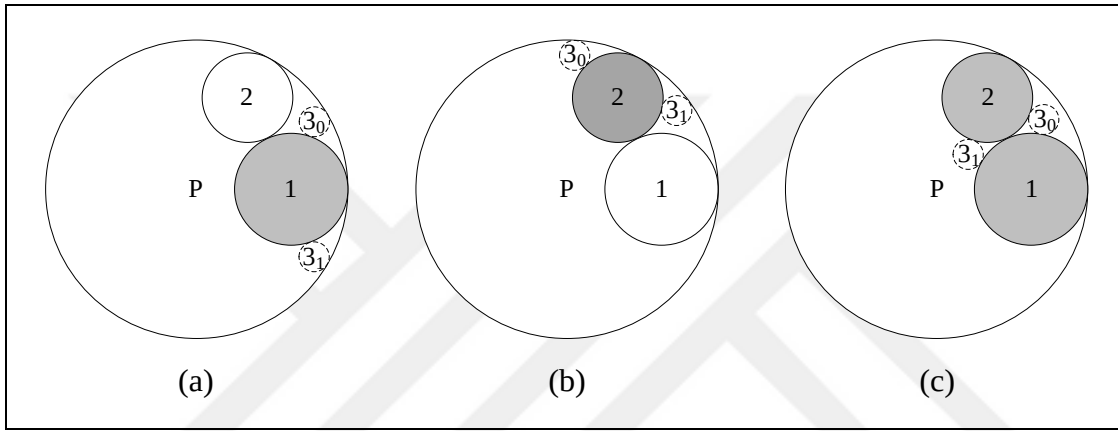
Yerleşecek olan her bir dairenin, hangi daire-daire ya da hangi palet-daire ikilisi ile komşu olacağı komşuluk puanına göre karar verilmektedir. Şekil 3.5’de gösterildiği üzere, 1 ve 2 nolu dairelerin bulunduğu palete, 3 nolu dairenin yerleşmesi öncesi durum gösterilmektedir. 3 nolu dairenin palete yerleştirilirken oluşabilecek komşuluklar üzerinde durulacaktır. Buna göre 3 nolu dairenin komşulukları şu şekilde gösterilmektedir:

Şekil 3.6. 3 nolu daire yerleşiminde mümkün olan komşuluklar

Şekil 3.6 (a)’da gösterildiği üzere daire-palet teğetliğinde 3 nolu daire, palete ve 1 nolu daireye komşu olabildiği görülmektedir. Şekil 3.6 (b)’de gösterildiği üzere yine daire-palet teğetliğinde 3 nolu daire, palete ve 2 nolu daireye komşu olabildiği görülmektedir. Şekil 3.6 (c)’de gösterildiği üzere daire-daire teğetliğinde 3 nolu daire, 1 ve 2 nolu dairelere komşuluğu komşu olabildiği görülmektedir.



Şekil 3.5. 3 nolu dairenin yerleşim öncesi durum



Şekil 3.6. 3 nolu daire yerleşiminde mümkün olan komşuluklar

3 nolu dairenin yerleşmesi sırasında Şekil 3.6'de gösterilen bu üç farklı komşuluktan hangisinin olacağı komşuluk puanı mekanizmasına göre belirlenmektedir. Her bir daireye ve palete başlangıçta verilen bu puan sayesinde, oluşan komşuluk alternatiflerinden birinin belirlenmesi sağlanmaktadır. Palet içinde yerleşmiş olan daire ve paletin komşuluk puanlarının büyükten küçüğe doğru sıralanarak, en yüksek komşuluk puanına sahip ikili, yeni yerleşecek olan dairenin komşuları olmaktadır.

En yüksek komşuluk puanına sahip ikiliye göre yapılacak olan yerleşimlerde taşmama ve çakışmama kısıtlarına ek olarak uygun mesafe kontrolü yapılmaktadır. Yerleşecek olan R_3 yarıçaplı dairenin R_1 ve R_2 dairelerine teğet olabilmesi için bu iki dairenin orijinleri arası mesafenin (R_x), Eş. 3.20'de gösterilen eşitliği sağlaması gerekmektedir.

$$R_x \leq R_1 + R_2 + 2R_3 \quad (3.20)$$

Taşma, çakışma ve uygun mesafe kontrolleri sağlandıktan sonra konum seçim değerine göre yerleşim yapılmaktadır. Eğer taşma, çakışma ve uygun olmayan mesafe sonucu yerleşim mümkün değilse, en yüksek komşuluk puanına sahip sıradaki ikiliye göre konum araması yapılmaktadır.

Konum seçim değeri

Şekil 3.6’de gösterildiği üzere 3 no’lu dairenin hangi ikili ile komşu olacağı komşuluk puanı (KP) ile belirlendikten sonra, konum seçim değeri (KSD) ile 3_0 ve 3_1 olan iki alternatif konumdan birinin seçilmesi gerekmektedir. Her bir daireye başlangıçta verilen 0 ve 1’den oluşan değerlerle birlikte, yerleşecek dairenin KSD’sine göre 3_0 ve 3_1 konumlardan birinin belirlenmesi sağlanmaktadır.

İki alternatif konum belirleme işleminde KP’de olduğu gibi taşma ve çakışma kısıtları da eş zamanlı kontrol edilmektedir. KSD mekanizması, iki alternatif konumdan ikisinin birden paletten taşmama ve yerleşen dairelerle çakışmama durumlarında kullanılabilir. İki alternatiften sadece birinde taşmama ve çakışmama olduğunda ise KSD’ye bakılmaksızın uygun olan alternatifin yerleşimi yapılmaktadır. Eğer iki alternatiften ikisinde de taşma ve çakışma oluyorsa, KP’ye göre en yüksek komşuluk puanına sahip sıradaki ikiliye geçilerek uygun konumlar aranır.

Algoritmanın daha iyi anlaşılması için Çizelge 3.1’de yerleşim algoritmasının adımları ve EK-1’de ise akış şeması gösterilmektedir. Çizelge 3.1’deki adımlarda girdi olarak yerleşecek daire listesi verilmekte, çıktı olarak ise yerleşmiş daire listesi alınmaktadır. Adımlar dairelerin yerleşim sırasının belirlenmesi ile başlamaktadır. Daha sonraki adımlar, daire ve palete komşuluk puanı ve her daireye konum seçim değerlerinin atanmasıyla devam etmektedir. Yerleştirme algoritmasında ilk daire daima saat 3 konumuna yerleştirilmektedir. Bu sayede ilk dairenin koordinatları kolaylıkla hesaplanmaktadır. Sonraki her daire palete ve dairelere iki noktadan teğet olacağından koordinat hesabı matematiksel modelde detayları anlatıldığı üzere yapılabilir. Sonraki her daire ekleme işlemi KP ve KSD’ye göre yapılmaktadır. Eğer yerleşecek dairenin iki teğet kuralına göre uygun konum bulamaması durumunda ise sıradaki daire için işlemler tekrarlanır. Yerleşecek daire listesinde yerleşmeye aday daire kalmadı ise işlem sonlandırılır.

Çizelge 3.1. Önerilen yerleşim algoritmasının adımları

Komşu ve Konum Seçimli Daire Yerleşim Algoritması
Girdi: Yerleşecek daire listesi Çıktı: Yerleşmiş daire listesi
Adım 1: Dairelerin yerleşim sırasını belirle; Adım 2: Dairelere ve palete benzersiz komşuluk puanı (KP) ata; Adım 3: Dairelere konum seçim değeri (KSD) ata; Adım 4: Yerleşecek daire listesinden ilk sıradaki daireyi seç; Adım 5: Seçili daireyi paletin saat 3 yönüne yerleştir; Adım 6: Yerleşecek daire listesini kontrol et; Eğer sırada seçilecek daire var ise daireyi seç ve Adım 7'ye git; Eğer sırada seçilecek daire yok ise Adım 10'a git; Adım 7: Paletin ve yerleşmiş olan dairelerin KP'sini büyükten küçüğe ikili olarak sırala; $KP_{ikili} = \{(KP_{max1}, KP_{max2}), (KP_{max1}, KP_{max3}) \dots (KP_{min2}, KP_{min1})\}$ Adım 8: Sıradaki KP_{ikili}'yi seç, Adım 9: Seçili daire ile seçili KP_{ikili}'nin ait olduğu palet-daire ya da daire-daire ikilisi için uygun konum ara, 1. Eğer tespit edilen uygun konum sayısı= 2 ise i. KSD'ye göre konum seç ii. Seçilen konuma göre daireyi yerleştir iii. Adım 6'ya git 2. Eğer tespit edilen uygun konum sayısı=1 ise, i. Konuma göre daireyi yerleştir ii. Adım 6'ya git 3. Eğer tespit edilen uygun konum sayısı=0 ise, i. Eğer seçili $KP_{ikili} \neq (KP_{min2}, KP_{min1})$ ise Adım 8'e git ii. Eğer seçili $KP_{ikili} = (KP_{min2}, KP_{min1})$ ise Adım 6'e git Adım 10: Bitir.

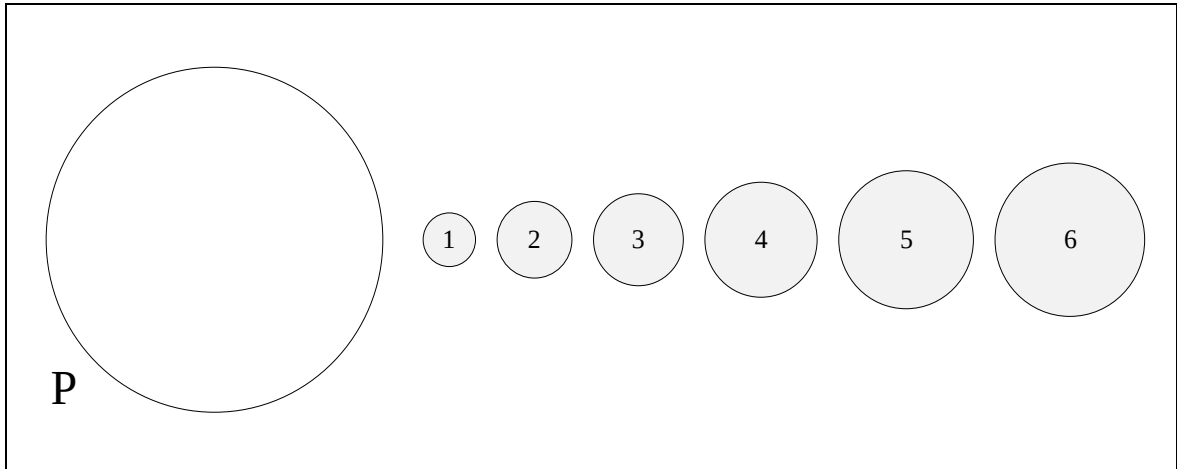
Algoritmada kullanılan kodlama

Algoritmanın kalitesini belirleyen en önemli faktörlerden biri ise arama uzayını en iyi şekilde temsil eden bir kodlamanın tasarlanmış olmasıdır. Bölüm 2.3.2’de detaylı anlatımı yapılan permütasyon kodlama ve ikili kodlama türleri tasarlanan yerleşim algoritmasında sıklıkla kullanılmaktadır. Algoritmada, dairelerin yerleşim sırası ve komşuluk puanlarının belirlenebilmesi için permütasyon (sıralı) kodlamadan yararlanılmıştır. Konum seçim değeri (KSD)’nin belirlenebilmesi için ise ikili (binary) kodlama kullanılmıştır. Kodlamaların detayları şu şekildedir:

- Yerleşim sırasının n adet daire için belirlenebilmesi için $[1, n]$ aralığında permütasyon kodlama ile rassal tekrarsız değerler verilmektedir.
- Komşuluk puanı (KP) dairelere ve palet verildiğinden, $[1, n+1]$ aralığında permütasyon kodlama ile rassal tekrarsız değerler verilmektedir.
- Konum seçim değeri (KSD), her daireye ikili kodlamaya göre $[0,1]$ aralığında rassal değer atanmaktadır.

Yerleşim algoritmasının örnek gösterimi

Yerleşim algoritmasının daha iyi anlaşılabilmesi için Şekil 3.7’de gösterildiği üzere küçük problem kümesi kullanılarak özdeş olmayan altı adet dairenin bir palet içine yerleşimi adım adım gerçekleştirilecektir.

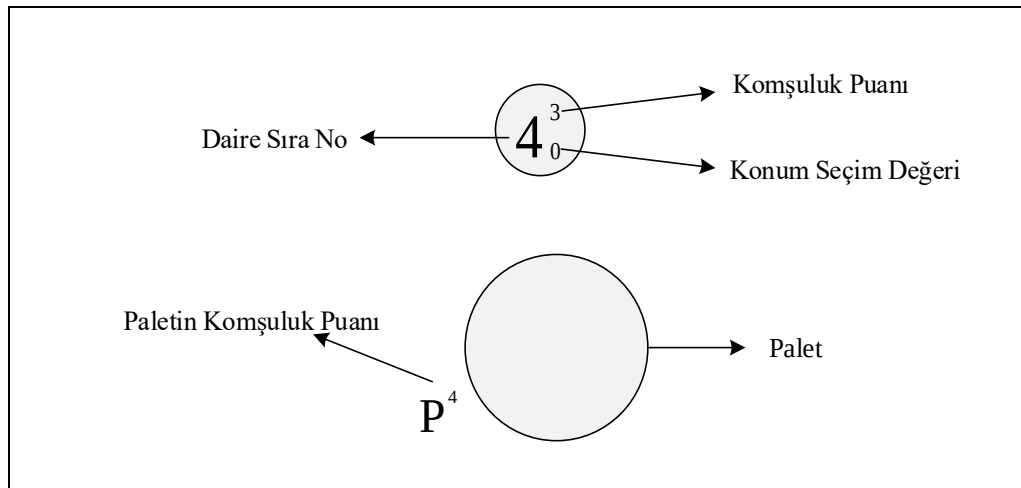


Şekil 3.7. Altı adet daire ve paletin gösterimi

Şekil 3.7’de gösterilen $n=6$ adet dairenin ve paletin permütasyon ve ikili kodlamaya göre alacağı puan ve değerler rassal olarak belirlenmiş ve Çizelge 3.2’de gösterilmiştir. Daire yerleşimi sırasında, işlem adımlarının daha anlaşılır olması için Şekil 3.8’de gösterildiği gibi bir gösterim yöntemi geliştirilmiştir. Bu yöntemle göre, daire sıra numarasında bulunan üst simge komşuluk puanı, alt simge alternatif konum seçim değerini ifade etmektedir. Ayrıca paletin üst simgesi ise dairelerle birlikte rekabete gireceği komşuluk puanını göstermektedir.

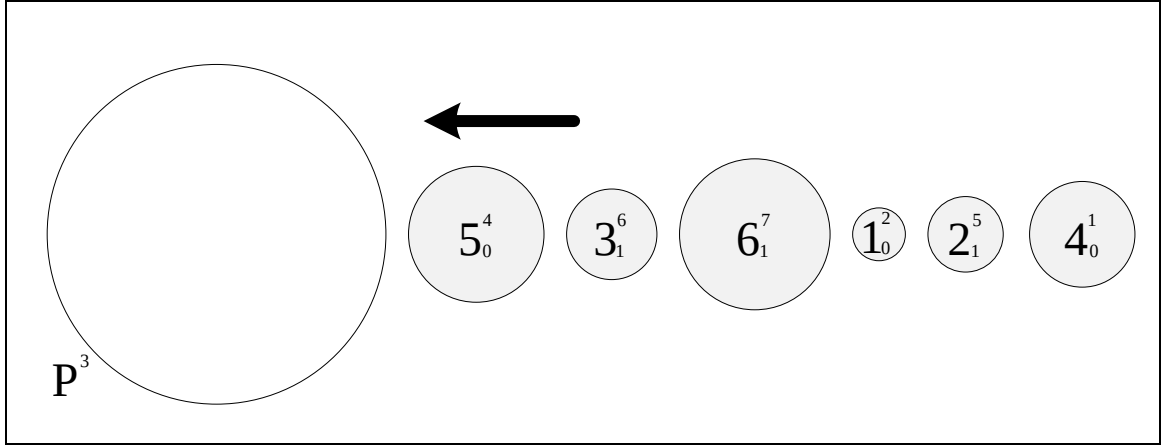
Çizelge 3.2. $n=6$ için dairelere atanan değerler

	Kodlama Türü	Palet	Daire (n=6)					
Daire Yerleşim Sırası	Permütasyon	-	5	3	6	1	2	4
Komşuluk Puanı (KP)	Permütasyon	3	4	6	7	2	5	1
Konum Seçim Değeri (KSD)	İkili	-	0	1	1	0	1	0



Şekil 3.8. Daire ve paletle ait parametrelerin gösterimi

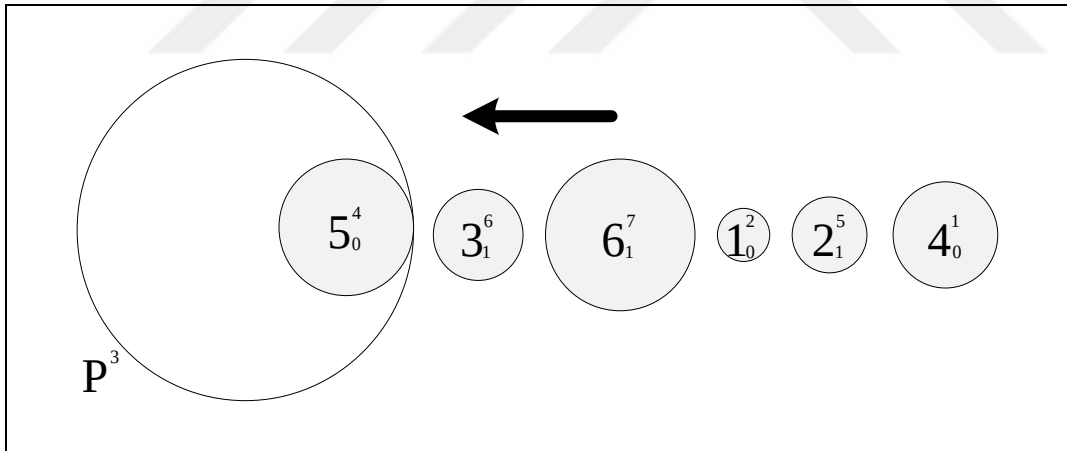
Tasarlanan gösterim şekline ve Çizelge 3.2’deki dizilime göre problemin başlangıç hali Şekil 3.9’de verilmiştir.



Şekil 3.9. Dairelerin ve paletin yerleşim öncesi gösterimi

Yerleşim 1

Kural 4’de belirtildiği üzere ilk yerleşecek daire saat 3 yönünde palete iç teğet olarak yerleştirilir. Şekil 3.10’da gösterildiği üzere 5 nolu daire, palete iç teğet olacak şekilde yerleştirilmiştir.



Şekil 3.10. 5 nolu dairenin yerleşimi sonrası gösterim

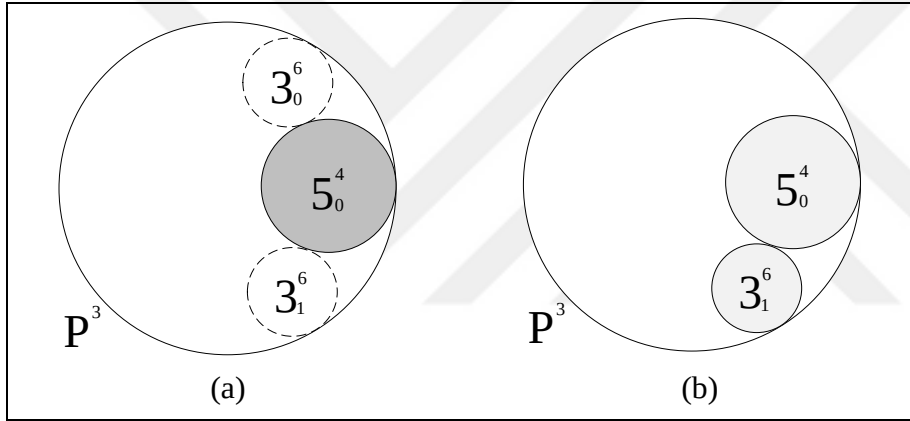
Yerleşim 2

Şekil 3.10’da gösterildiği üzere 5 nolu dairenin yerleşimi sonrası, 3 nolu dairenin yerleşimi gerçekleştirilecektir. 3 nolu dairenin yerleşimi öncesi, muhtemel komşularının tespiti için hesaplama gerekmektedir. Komşuluk ve konum tespiti için gereken işlem adımları aşağıda belirtilmiştir. Buna göre:

Adım 1: Palet ve palet içindeki dairelerin komşuluk puanları ($KP=\{4,3\}$) sıralanır ve en ikili olarak $(KP_1, KP_2)=\{(4,3)\}$ listelenir.

Adım 2: $(KP_1, KP_2)=\{(4,3)\}$ listesine göre, 3 nolu dairenin sadece palet ($KP=3$) ve 5 nolu daire ($KP=4$) ile komşuluğu bulunmaktadır. Bu komşuluğa göre Şekil 3.11(a)'da gösterildiği gibi iki konum alternatifi (3_0 ve 3_1) ortaya çıkmaktadır.

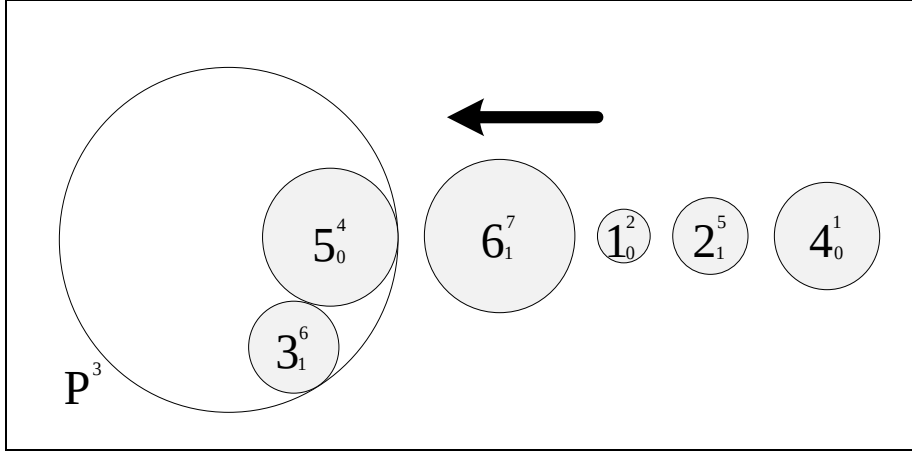
Adım 3: 3_0 ve 3_1 olmak üzere iki konum alternatifi bulunan palet ve 5 nolu daire komşuluğunda, bu alternatiflerden birinin seçilmesi gerekmektedir. 3 nolu dairenin Konum seçim değeri (KSD) Çizelge 3.2'de gösterildiği gibi "1" olarak tanımlandığından yerleşim Şekil 3.11 (b)'deki gibi gerçekleştirilmektedir.



Şekil 3.11. 3 nolu dairenin komşuluk seçenekleri ve yerleşiminin gösterimi

Yerleşim 3

Şekil 3.12'de gösterildiği üzere 3 nolu dairenin yerleşimi sonrası, 6 nolu dairenin yerleşimi gerçekleştirilecektir. 6 nolu dairenin, muhtemel komşularının tespiti için hesaplama gerekmektedir.



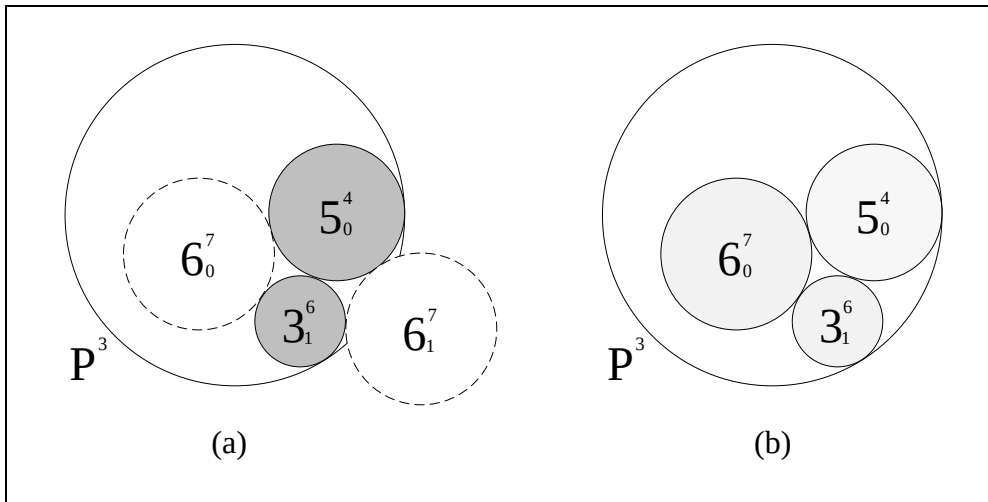
Şekil 3.12. 6 nolu dairenin yerleşimi öncesi durum

Komşuluk ve konum tespiti için gereken işlem adımları aşağıda belirtilmiştir. Buna göre:

Adım 1: Palet ve palet içindeki dairelerin komşuluk puanları ($KP=\{6,4,3\}$) sıralanır. En yüksek puanlılardan başlayarak ikili kombinasyonları listelenir. (KP_1, KP_2)= $\{(6,4), (6,3), (4,3)\}$

Adım 2: (KP_1, KP_2) listesindeki daire çiftlerinin ilk sırada olanları ile başlamak üzere, taşıma, çakışma, uygun aralık kontrolü yapılarak uygun yerleşim aranır. (KP_1, KP_2)= $(6,4)$ puanlarına sahip 3 ve 5 nolu dairelerle yapılacak komşulukta Şekil 3.13 'da gösterildiği gibi bir adet uygun konum (6_0) tespit edilmiştir.

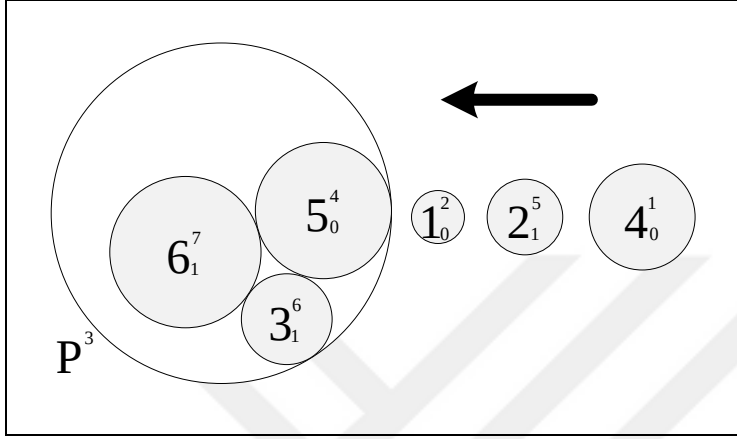
Adım 3: 3 ve 5 nolu daireler ile yapılan komşulukta sadece bir adet uygun konum tespit edildiğinde KSD'ye bakılmaksızın Şekil 3.13 (b)'deki gibi yerleşim gerçekleştirilmiştir.



Şekil 3.13. 6 nolu dairenin yerleşim adımlarının gösterimi

Yerleşim 4

Şekil 3.14’da gösterildiği üzere 6 nolu dairenin yerleşimi sonrası, 1 nolu dairenin yerleşimi gerçekleştirilecektir. 1 nolu dairenin, muhtemel komşularının tespiti için hesaplama gerekmektedir.



Şekil 3.14. 1 nolu dairenin yerleşimi öncesi durum

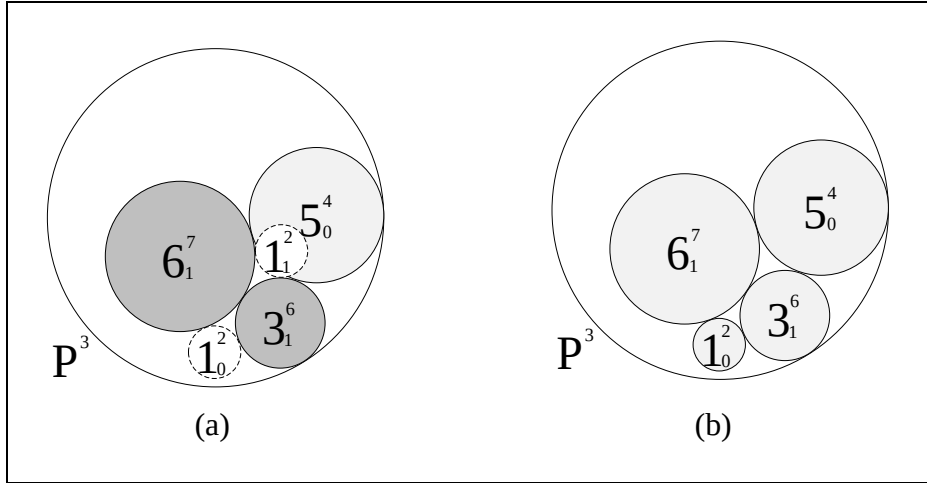
Komşuluk ve konum tespiti için gereken işlem adımları aşağıda belirtilmiştir. Buna göre:

Adım 1: Palet ve palet içindeki dairelerin komşuluk puanları ($KP = \{7, 6, 4, 3\}$) sıralanır. En yüksek puanlılardan başlayarak ikili kombinasyonları listelenir. (KP_1, KP_2) = $\{(7, 6), (7, 4), (7, 3), (6, 4), (6, 3), (4, 3)\}$

Adım 2: (KP_1, KP_2) listesindeki daire çiftlerinden ilki ile başlamak üzere, taşma, çakışma, uygun aralık kontrolü yapılarak uygun yerleşim aranır.

- (KP_1, KP_2) = (7, 6) puanlarına sahip 6 ve 3 nolu dairelere yapılacak komşulukta Şekil 3.15 (a)’da gösterildiği gibi bir adet uygun konum (1_0) tespit edilmiştir.

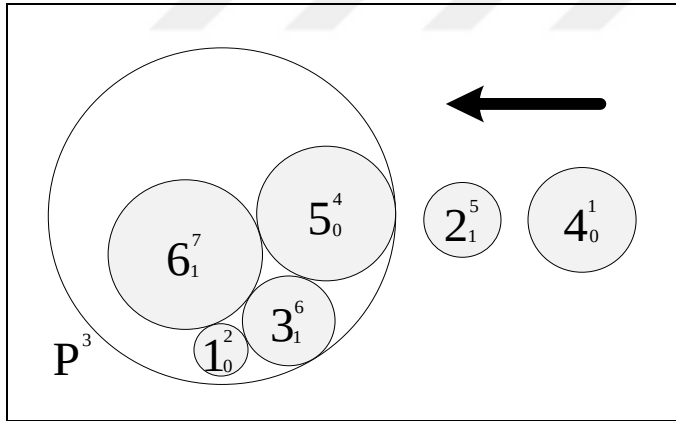
Adım 3: 6 ve 3 nolu daireler ile yapılan komşulukta bir adet uygun konum tespit edildiğinden KSD’ye bakılmaksızın yerleşim Şekil 3.15 (b)’de gösterildiği gibi gerçekleştirilmiştir.



Şekil 3.15. 1 nolu dairenin yerleşim adımlarının gösterimi

Yerleşim 5

Şekil 3.16’de gösterildiği üzere 1 nolu dairenin yerleşimi sonrası, 2 nolu dairenin yerleşimi gerçekleştirilecektir. 2 nolu dairenin, muhtemel komşularının tespiti için KP ve KSD kontrol edilmektedir.



Şekil 3.16. 2 nolu dairenin yerleşimi öncesi durum

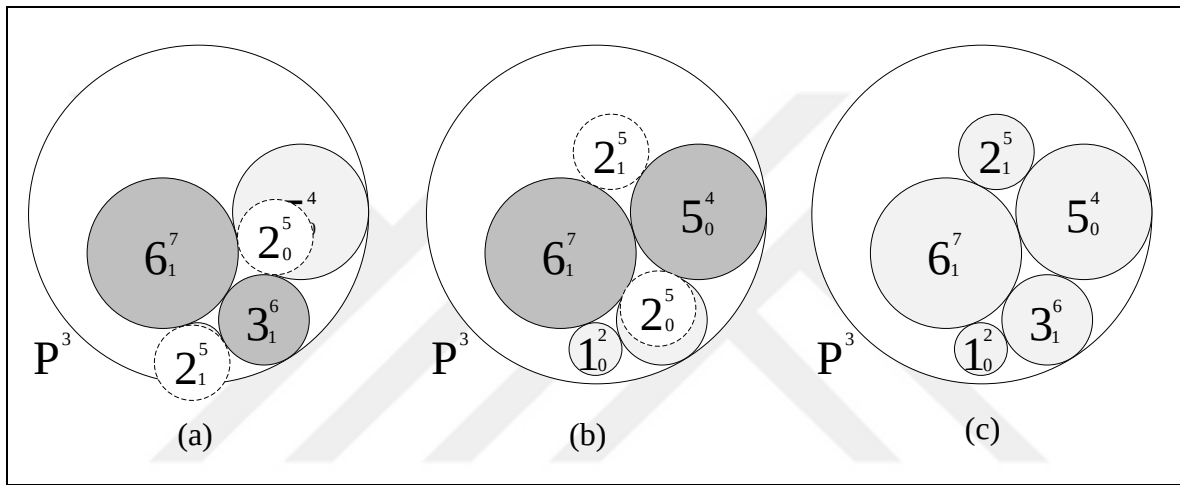
Komşuluk ve konum tespiti için gereken işlem adımları aşağıda belirtilmiştir. Buna göre:

Adım 1: Palet ve palet içindeki dairelerin komşuluk puanları ($KP = \{7, 6, 4, 3, 2\}$) sıralanır. En yüksek puanlılardan başlayarak ikili kombinasyonları listelenir. $(KP_1, KP_2) = \{(7, 6), (7, 4), (7, 3), (7, 2), (6, 4), (6, 3), (6, 2), (4, 3), (4, 2), (3, 2)\}$

Adım 2: (KP_1, KP_2) listesindeki daire çiftlerinden ilk sıradaki ile başlamak üzere, taşma, çakışma, uygun aralık kontrolü yapılarak uygun yerleşim aranır.

- $(KP_1, KP_2)_1 = (7,6)$ puanlarına sahip 6 ve 3 nolu dairelere yapılacak komşulukta Şekil 3.17 (a)'da gösterildiği gibi her iki konum için $(2_0$ ve $2_1)$ taşma ve çakışma olmaktadır. Bu komşuluk kabul edilmeyip, sıradaki daire çifti için komşuluk aranır.
- $(KP_1, KP_2)_2 = (7,4)$ puanlarına sahip 6 ve 5 nolu dairelerle yapılan komşulukta Şekil 3.17 (b)'de gösterildiği gibi bir adet uygun konum (2_1) tespit edilmiştir.

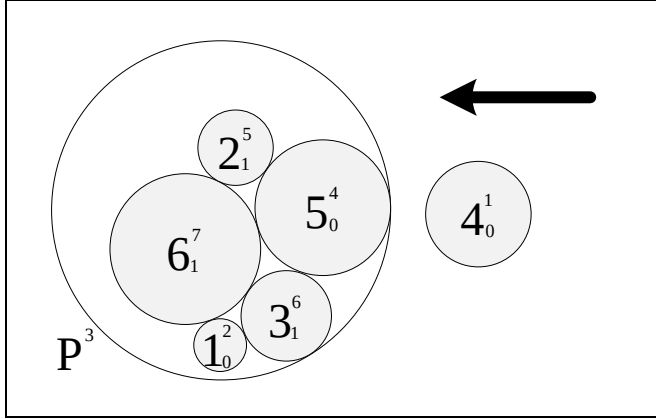
Adım 3: 6 ve 5 nolu daireler ile yapılan komşulukta sadece bir adet uygun konum tespit edildiğinden KSD'ye bakılmaksızın Şekil 3.17 (c)'deki gibi yerleşim gerçekleştirilmiştir.



Şekil 3.17. 2 nolu dairenin yerleşim adımlarının gösterimi

Yerleşim 6

Şekil 3.18'de gösterildiği üzere 2 nolu dairenin yerleşimi sonrası, sırada olan 4 nolu dairenin yerleşimi gerçekleştirilecektir. 4 nolu dairenin, muhtemel komşularının tespiti için KP ve KSD kontrol edilmektedir.



Şekil 3.18. 4 nolu dairenin yerleşimi öncesi durum

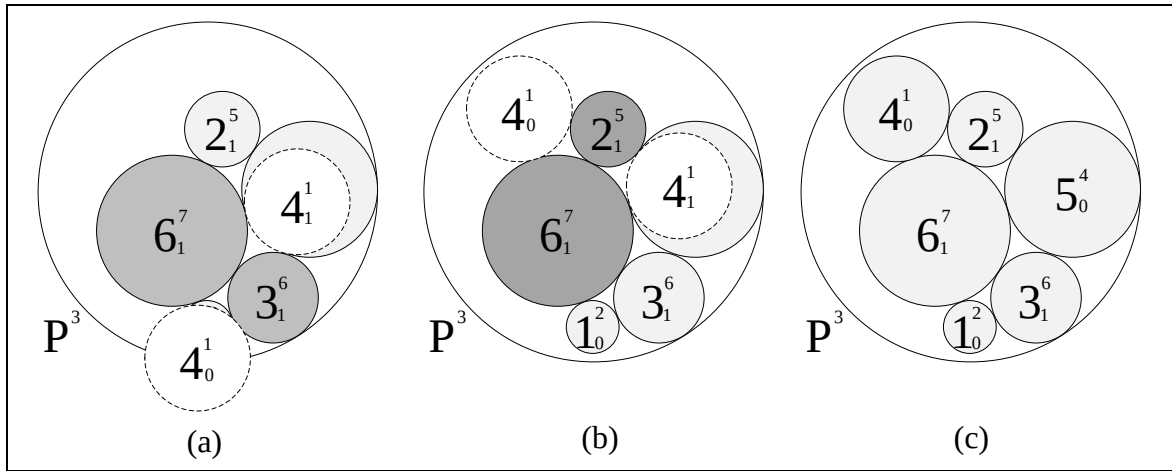
Komşuluk ve konum tespiti için gereken işlem adımları aşağıda belirtilmiştir. Buna göre:

Adım1: Palet ve palet içindeki dairelerin komşuluk puanları ($KP=\{7,6,5,4,3,2\}$) sıralanır. En en yüksek puanlılardan başlayarak ikili kombinasyonları listelenir. $(KP_1, KP_2)=\{(7,6), (7,5), (7,4), (7,3), (7,2), (6,5), (6,4), (6,3), (6,2), (5,4), (5,3), (5,2), (4,3), (4,2), (3,2)\}$

Adım 2: (KP_1, KP_2) listesindeki daire çiftlerinin ilkiyle başlamak üzere, taşıma, çakışma, uygun aralık kontrolü yapılarak uygun yerleşim aranır.

- $(KP_1, KP_2)_1=(7,6)$ puanlarına sahip 6 ve 3 nolu dairelere yapılacak komşulukta Şekil 3.19 (a)'da gösterildiği gibi taşıma ve çakışma tespit edilmiştir. Bu komşuluk kabul edilmeyip, sıradaki daire çifti için komşuluk aranır.
- $(KP_1, KP_2)_2=(7,5)$ puanlarına sahip 6 ve 2 nolu dairelere yapılacak komşulukta Şekil 3.19 (b)'de gösterildiği gibi bir adet uygun konum (4_0) tespit edilmiştir.

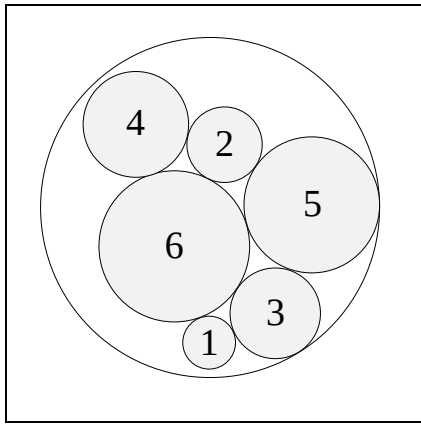
Adım 3: 6 ve 2 nolu daireler ile yapılan komşulukta bir adet uygun konum tespit edildiğinden KSD'ye bakılmaksızın Şekil 3.19 (c)'deki gibi bir yerleşim gerçekleştirilmiştir.



Şekil 3.19. 4 nolu daireye uygun yer tespiti

Yerleşim Sonucu

Altı adet daireden oluşan küçük boyutlu bir problemin çözümü gerçekleştirilerek Şekil 3.20'deki gibi bir desen ortaya çıkmıştır. İlk dairenin saat 3 yönünde yerleştirilmesi ile başlayan yerleşim işlemi, diğer beş dairenin paletten taşmama, dairelerin birbirleri ile çakışmama ve dairelerin en az iki nokta komşuluk sağlaması kuralları ile düzenli bir yerleştirme işlemi gerçekleştirilmiştir.

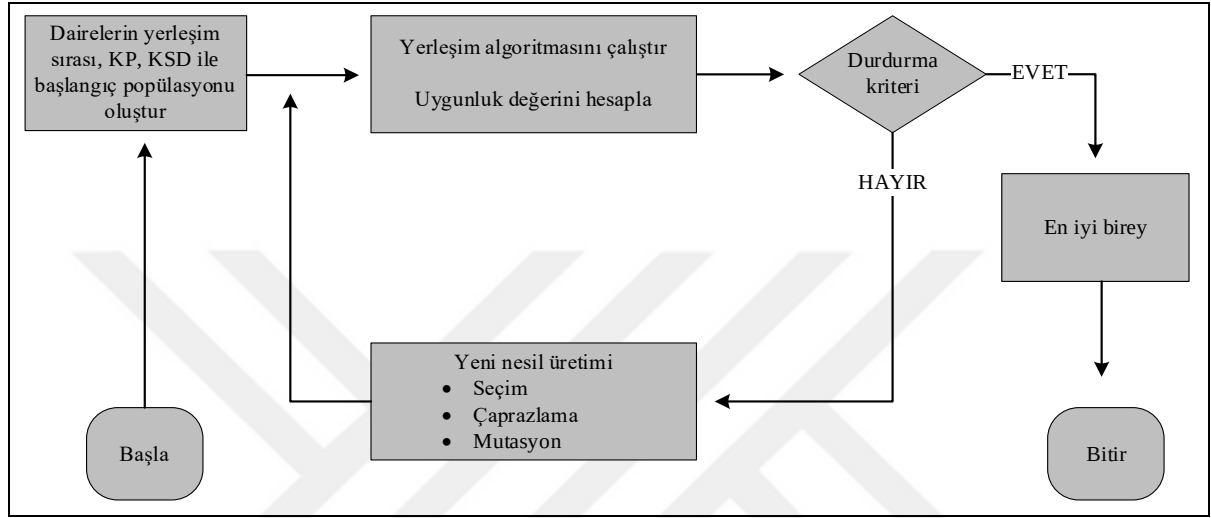


Şekil 3.20. Altı dairesel paketleme probleminin çözümü

3.3.2. DPP için geliştirilen genetik algoritma

Bu çalışmada, dairelerin çözüm alanına yerleştirilmesi için genetik algorithmadan yararlanılmıştır. Şekil 3.21'de önerilen sistemin tasarımına yönelik genetik algoritmanın adımlarını içeren akış şeması gösterilmiştir. Şemada gösterildiği üzere karar verici tarafından

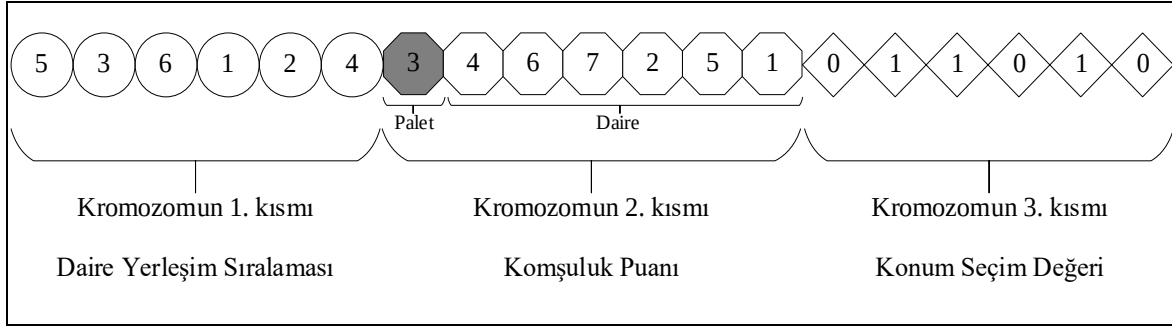
belirlenen başlangıç popülasyonundaki bireylerin, yerleşim algoritmasının çalıştırılması ile uygunluk değerleri hesaplanır. Hesaplanan uygunluk değerlerini de içeren durdurma kriteri şartları sağlandığı anda en iyi birey, problemin de en iyi çözümü olarak kayda alınır. Durdurma kriteri sağlanmadığı durumlarda ise yeni nesil üretimi için genetik operatörler çalıştırılarak mevcut popülasyonda değişim gerçekleştirilerek süreç tekrarlanır.



Şekil 3.21. DPP'nin çözümü için önerilen genetik algoritmanın akış şeması

Kromozom yapısının oluşturulması

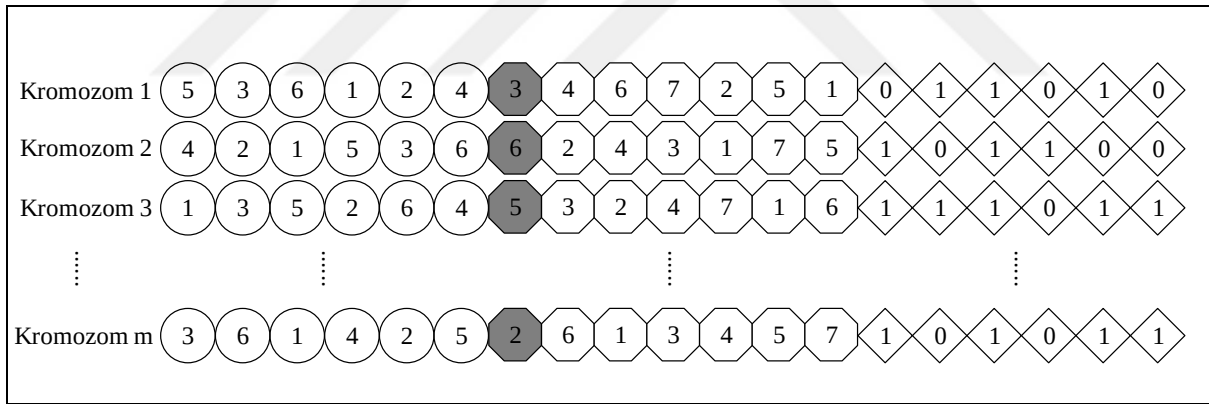
Kromozom yapısının anlaşılır bir şekilde tasarlanması genetik algoritmanın sonraki aşamaları için büyük kolaylık sağlamaktadır. Şekil 3.22’de gösterildiği üzere, DPP’de kromozom üç kısımdan oluşacak şekilde tasarlanmıştır. Kromozoma ait birinci kısmın her bir geni permüstasyon kodlamaya göre oluşturulmuş olan dairelerin yerleşim sırası göstermektedir. İkinci kısımdaki ilk gen paletin, diğer genler dairelerin komşuluk puanını belirtmektedir. Üçüncü kısım ise konum seçim değerini belirten ikili kodlama ile oluşturulan genlerdir.



Şekil 3.22. Genetik algorithmada örnek kromozom gösterimi

Başlangıç popülasyonunun oluşturulması

Genel bir genetik algoritma adımlarından biri başlangıç popülasyonunun oluşturulmasıdır. Gen diziliminin ve kromozom yapısının kodlama kurallarına göre oluşturulmasından sonra karar vericinin belirlediği bir büyüklüğe göre başlangıç popülasyonu oluşturulur. Şekil 3.23’de, 6 adet dairenin yerleşimi için oluşturulmuş, büyüklüğü m adet olan bir başlangıç popülasyonu gösterilmiştir.



Şekil 3.23. Kromozomlardan oluşan popülasyonun gösterimi

Popülasyon büyüklüğü genetik algoritmanın performansını ve verimliliğini etkileyen en önemli parametredir. Popülasyon büyüklüğü, hem kaynak kullanımını olumsuz yönde etkileyecek kadar fazla olmamalı, hem de çözüm uzayını kısıtlı tarayacak ve erken yakınsamaya sebep olacak kadar da az olmamalıdır. Bu sebeple, DPP çözümü için uygun olan popülasyon büyüklüğü parametre optimizasyonu yapılarak tespit edilmiştir. Bölüm 3.3.3’de parametrelerin belirlenmesi üzerinde durulmuştur.

Uygunluk değeri ve uygunluk fonksiyonu

Her bir kromozom, kodlandığı çözümün kalitesini göstermesi açısından uygunluk fonksiyonuyla elde edilen bir uygunluk değerine sahiptir. Her bir kromozomun bir sonraki aşamadaki varlığı bu değere göre değerlendirilmektedir. Kromozomların uygunluk değerlerini hesaplayabilmek için iyi oluşturulmuş bir fonksiyona ihtiyaç duyulmaktadır. Uygunluk fonksiyonları, problemin türüne göre istenilen şekilde oluşturulabilmektedir. DPP'nin çözümü için önerilen GA'daki uygunluk fonksiyonu, daire sayısı ve dairelerin alan kullanımının hesaplaması temeline oturtulmuştur.

Bu çalışma ile önerilen çözüm yönteminde uygunluk fonksiyonunun oluşturulmasında daire maksimizasyonu ve alan maksimizasyonu amaç fonksiyonlarının birbirinden bağımsız olarak ele alınmaktadır. Dairelerin, yerleştirme algoritması esaslarına göre yerleştirilmesi ile birlikte, daire sayısı maksimizasyonu problemlerinde uygunluk fonksiyonu Eş. 3.21'de gösterildiği gibi olmaktadır. Palet içine yerleştirilen dairelerin kullanım alanı en fazla olmasının hedeflendiği alan maksimizasyonu problemlerinde ise uygunluk fonksiyonu Eş. 3.22'deki verilmektedir.

n = toplam daire sayısı

R_i = i. dairenin yarıçapı

R_0 = Palet yarıçapı

α_i = i. dairenin palete yerleşme durumu (0: eklenmedi 1: eklendi)

1. Palet alanı üzerinde daire sayısı maksimizasyonuna göre uygunluk fonksiyonu

$$F(x) = \max \sum_{i=1}^n \alpha_i + \frac{\pi R_0^2 - \sum_{i=1}^n \alpha_i \pi R_i^2}{\sum_{i=1}^n \alpha_i \pi R_i^2 + 1} \quad (3.21)$$

2. Palet alanı üzerinde alan maksimizasyonu göre uygunluk fonksiyonu

$$F(x) = \max \sum_{i=1}^n \alpha_i \pi R_i^2 \quad (3.22)$$

Seçim mekanizması ve genetik operatörler için uygulanan yöntemler

Genetik algoritmanın en iyi sonucu elde edebilmesi için çözüm uzayının etkin bir şekilde taranması gerekmektedir. Bu tarama, aşağıda belirtilen genetik işlemler kullanılarak sağlanmaktadır.

Seçim

İyi, kaliteli olan bireylerin yaşaması ve genlerinin sonraki nesillere aktarılması ilkesiyle hareketle, popülasyon içindeki bireyler arasından en iyi uygunluk değerlerine sahip olanlar bir başka deyişle kaliteli genlere sahip olanlar arasından yeni nesil oluşturmak için bir kısmının seçilmesi işlemi bu operatör ile mümkün olmaktadır. DPP'nin çözümü için turnuva seçim yöntemi tercih edilmiştir.

Turnuva seçim yönteminde, popülasyondan rasgele seçilen k adet bireyden oluşturulan gruba turnuva genişliği denir. Bu gruptaki en iyi birey, yeni popülasyona kopyalanır. Bu işlem kullanıcı tarafından önceden kararlaştırılan çevrim sayısı kadar tekrarlanır [139]. Turnuva genişliği (k) kadar birey rasgele olarak seçilir ve turnuvada yarışır. Yarışı kazanan bireyler sonraki nesli oluşturmaktadır. Genelde turnuva iki birey arasında düzenlenmektedir ancak farklı sayıda genişliklerden ($k=2,3,5,\dots$) oluşan bireyler arasında da turnuva yapılabilmektedir.

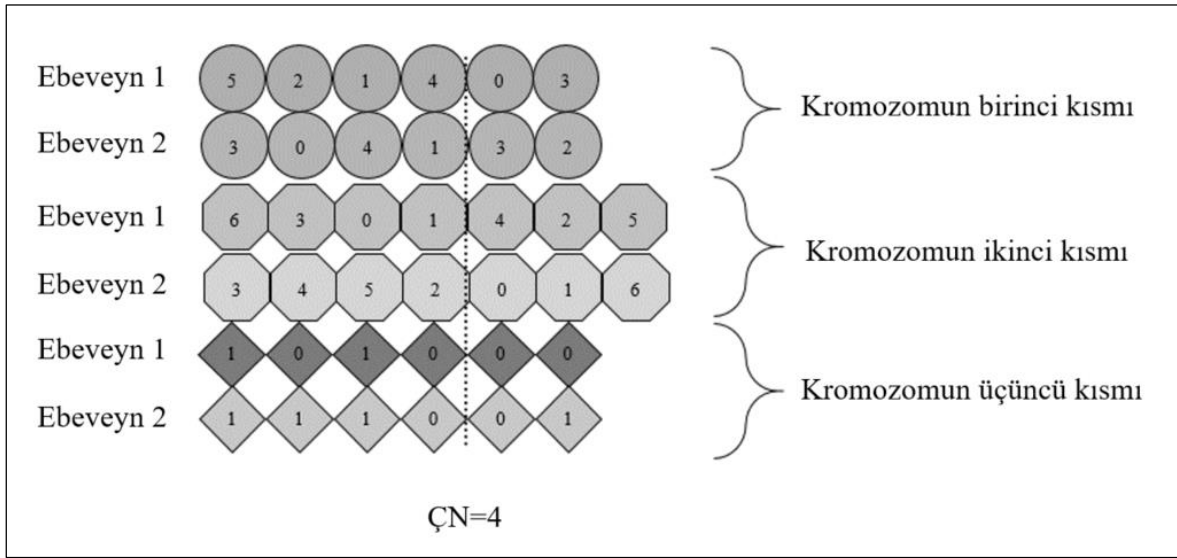
Bu çalışmada literatürde yapılan çalışmalar göz önünde bulundurularak rasgele seçilen birey sayısını (k) belirleyebilmek için ise bir dizi test ve analiz çalışması yapılmıştır. Yapılan test çalışması, turnuva büyüklüğü $k=2$ ve $k=3$ alınarak alan maksimizasyonu ve daire sayısı maksimizasyonunda elde edilen alan ve daire sayısı sonuçlarının CPU zamanı performansı ile birlikte analiz edilmesinden oluşmaktadır. EK-2'de gösterilen alan maksimizasyonuna göre, EK-3'de ise daire sayısı maksimizasyonuna göre elde edilen test sonuçlarında $k=3$ turnuva büyüklüğünün algoritmanın etkin çalışması için uygun olacağı görülmüştür.

Çaprazlama

Seçim yöntemi ile seçilen iki ebeveynin kromozomlar üzerinde rassal olarak tespit edilen noktalardan genlerin çapraz bir şekilde dizilimin değiştirilmesi yolu ile iki yavru kromozom elde edilmesi işlemidir.

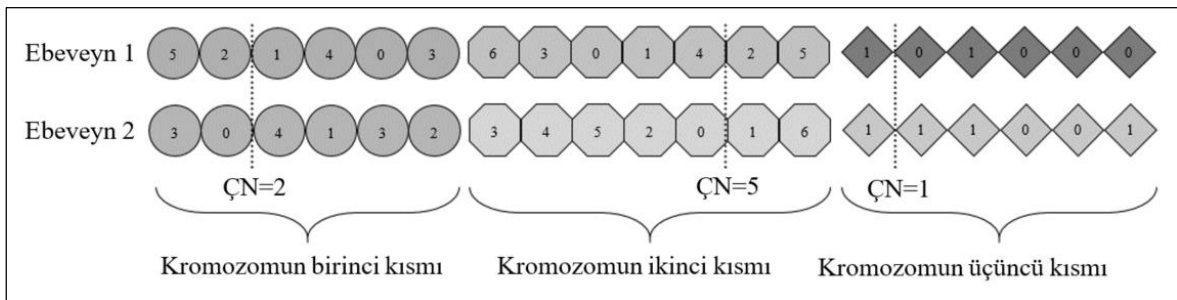
DPP'nin çözümü için tek noktali çaprazlama yöntemi önerilmektedir. Buna göre; kromozomun üç ayrı kısmı için kullanılacak bir rassal olarak belirlenecek çaprazlama noktasının belirlenme yöntemi için bir ön çalışma yapılmıştır. Yapılan ön çalışmada çaprazlama noktasının (ÇN) belirlenmesinde iki alternatif üzerinde durulmuştur. Bu iki alternatif şu şekildedir:

Birinci alternatif: Bu çaprazlama noktası belirleme alternatifinde, Şekil 3.24'de de gösterildiği üzere kromozomun her üç kısmı için aynı noktanın kullanılması üzerinde durulmaktadır.



Şekil 3.24. Çaprazlama noktası seçim yöntemi için birinci alternatif

İkinci alternatif: Bu çaprazlama noktası alternatifinde ise Şekil 3.25'de gösterildiği üzere her bir kromozomun üç kısmı için ayrı ayrı ÇN belirlenip, Ebeveyn 1 ve Ebeveyn 2'nin çaprazlama işlemlerinin üç ayrı ÇN'ye göre yapılması yöntemine dayanmaktadır.

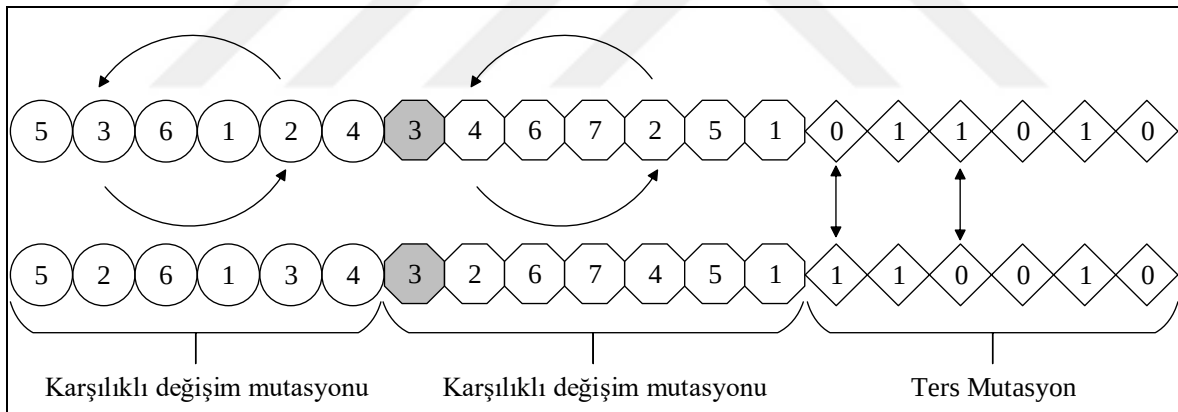


Şekil 3.25. Çaprazlama noktası seçim yöntemi için ikinci alternatif

Bu çalışmada, çaprazlama noktasının belirlenme yöntemi için sunulan iki alternatif arasından uygun olanın tespiti için bir dizi test ve analiz çalışması yapılmıştır. Yapılan test çalışması alan maksimizasyonu ve daire sayısı maksimizasyonunda elde edilen alan ve daire sayısı sonuçlarının CPU zamanı performansı ile birlikte analiz edilmesinden oluşmaktadır. EK-4’de gösterilen alan maksimizasyonuna göre, EK.5’de ise daire sayısı maksimizasyonuna göre elde edilen sonuçlarda, çaprazlama noktası belirlemede ikinci alternatifin daha uygun olduğu görülmektedir.

Mutasyon

Çaprazlama işleminden hemen sonra uygulanan mutasyon operatöründe, çaprazlama sonucu elde edilen yavruların mutasyona uğratılmasıyla yeni bireylerin oluşması sağlanmaktadır. Bölüm 2.3.6’da mutasyon yöntemleri açıklanmıştır. Şekil 3.26’da gösterilen kromozom yapısı dikkate alındığında, kromozomun birinci ve ikinci kısımları için karşılıklı değişim mutasyonu, üçüncü kısım için ise ters mutasyonun uygun olduğu görülmektedir.



Şekil 3.26. Seçilen mutasyon yöntemlerinin kromozomda gösterimi

3.3.3. GA bileşenleri için parametre optimizasyonu

Genetik algoritma ile çözülen problemlerde, algoritmaya ait operatörlerde kullanılan parametrelerin probleme özgü olması ve algoritmanın başarımını arttıracak şekilde seçilmesi son derece önemlidir. Popülasyon büyüklüğü (P_s), çaprazlama oranı (P_c) ve mutasyon oranı (P_m) gibi parametreler için seçilecek uygun değer ve oranlar uygunluk değeri, hesaplama süresi gibi algoritmanın başarısını etkileyen unsurları olumlu yönde etkilemektedir. Bir başka deyişle, iyi analiz edilerek belirlenmiş parametreler GA’nın kalitesini ve başarısını olumlu yönde etkilemektedir.

Bu sebeple DPP'nin çözümü için geliştirilen GA'da kullanılacak parametrelerin belirlenebilmesi için bir parametre optimizasyonu çalışması yapılmıştır. Bu çalışmada kullanılacak veri setleri, Çizelge 3.3'de gösterildiği üzere Lopez ve Beasley'in [79] çalışmasındaki test problemleri arasından üç tanesi rastgele seçilerek gerekli analiz çalışması gerçekleştirilmiştir. 10, 20, 40 dairesel test problemi, iki farklı amaç fonksiyonuyla (alan maksimizasyonu ve daire sayısı maksimizasyonu) birlikte altışar kez çalıştırılarak elde edilen 2916 ($3 \times 2 \times 3 \times 3 \times 3 \times 6$) adet sonuç popülasyon büyüklüğü, çaprazlama oranı, mutasyon oranı, durdurma kriteri parametrelerinin tespiti için IBM SPSS Statistics 23.0 yazılımı ile istatistiksel analizler gerçekleştirilmiştir. Dikkate alınan performans kriterlerinin istatistiksel analiz sonuçları EK.6, EK.7, EK.8 ve EK.9'de verilmiştir. Performans kriteri olarak daire sayısı maksimizasyonu, alan maksimizasyonu ve bunların CPU zaman değerleri seçilmiştir.

Çizelge 3.3. Parametre optimizasyonu için seçilen test problemleri

Daire sayısı (n)	Daire yarıçapları (R _i)	Palet yarıçapları (R ₀)
10	1.08, 1.10, 2.17, 3.21, 3.91, 4.27, 4.31, 4.34, 4.40, 4.72	6.58
20	1.02, 1.19, 1.23, 1.43, 1.55, 1.77, 1.85, 2.11, 2.14, 2.32, 2.65, 3.04, 3.13, 3.16, 3.29, 4.37, 4.38, 4.56, 4.60, 4.70	9.45
40	1.00, 1.02, 1.09, 1.28, 1.38, 1.45, 1.53, 1.60, 1.63, 1.66, 1.68, 1.72, 1.85, 2.05, 2.07, 2.57, 2.69, 2.75, 2.92, 2.94, 2.97, 3.10, 3.16, 3.44, 3.65, 3.67, 3.76, 3.76, 3.94, 3.97, 4.09, 4.13, 4.20, 4.21, 4.24, 4.35, 4.63, 4.72, 4.85, 4.96,	16.33

Popülasyon büyüklüğünün belirlenmesi

Genetik algoritmada parametre belirlemenin adımlarından ilki olan başlangıç popülasyon büyüklüğü için parametre seçimi, algoritmanın performansını etkileyen en önemli unsurlardan biridir. Popülasyon büyüklüğü ile ilgili literatür incelendiğinde, problemin büyüklüğüne bağlı olarak [10-100] aralığında seçimlerin uygun olabileceği belirtilmiş ve GA'daki popülasyon büyüklüğü 10 ile 160 arasında ve 10'nun katları şeklinde olması gerektiği bildirilmiştir [140]. Ayrıca, çok küçük popülasyon yetersiz örneklem büyüklüğü vermekte ve aramanın belirli bir yerel optimal noktaya doğru gitmesine neden olmaktadır.

Çok büyük popülasyon ise çok sayıda hiperdüzlemin temsilini içermekte ve yerel optimal çözüme yakınsamayı da engellemektedir. Ancak büyük popülasyon nesil gelişimi için daha fazla süreye ihtiyaç duymakta olduğundan hesaplama zamanını arttırmaktadır. Bu iki farklı parametre GA'nın performansı açısından istenmeyen durum oluşturmaktadır [141].

Bu çalışmada, literatürde yapılan çalışmalar göz önünde bulundurularak paketlenecek daire sayısına (n) göre, rastgele popülasyon büyüklüğü ($3n$, $4n$, $6n$) belirlenmiştir. Belirlenen popülasyon büyüklüğünün tespiti için bir dizi test çalışması yapılmıştır.

Daire sayısı performans kriterine göre ANOVA testi sonucu 0,997 düzeyinde anlamlılık vermiştir. 0,05 hata seviyesine göre popülasyon büyüklüğünün istatistiksel olarak daire sayısı performansına etki etmediği sonucu ortaya çıkmıştır. Ayrıca daire sayısı kriterinin CPU zamanı performans ölçütüne göre popülasyon büyüklüklerinin ANOVA testi sonucu istatistiksel olarak 0,05 hata seviyesinde anlamlı çıkmıştır. Hangi popülasyon büyüklüğü seviyesinde daha iyi performans elde edildiğinin tespiti için Duncan gruplandırma testi yapılmıştır ve sonuç olarak 0,05 hata seviyesinde $3n$ ve $4n$ 'nin istatistiksel olarak anlamlı bir farkının olmadığı fakat $6n$ 'nin istatistiksel olarak anlamlı bir farkının olduğu görülmüştür. CPU zamanının düşük değerlerde olmasının istenmesi sebebiyle $3n$ ve $4n$ popülasyon büyüklükleri bu aşamada seçilebilir. Alan performans kriteri için yapılan ANOVA testi sonucuna göre popülasyon sayısının anlamlılık düzeyi 0,999 çıkmıştır ve istatistiksel olarak 0,05 hata seviyesinde popülasyon sayıları arasında bir farklılık yoktur. Bu performans kriterinin CPU zamanı ile ilgili ANOVA sonucu ise 0,05 hata seviyesinde anlamlı çıkmıştır. Duncan testi gruplandırmasına göre yine $3n$ ve $4n$ arasında 0,05 hata seviyesinde istatistiksel olarak anlamlı bir farklılık yoktur fakat $6n$ ve diğerleri arasında 0,05 hata seviyesinde anlamlı bir fark bulunmaktadır. Çalışmada en düşük CPU zamanı $3n$ popülasyon büyüklüğünde elde edildiği için GA'nın bir parametresi olarak seçilmiştir.

Çaprazlama oranının belirlenmesi

Popülasyondan seçilen bireyler arasından en iyi olanların kalıtsal özelliklerini taşıyan yeni bireyler oluşturma işlemi için uygun çaprazlama oranının (P_c) belirlenmesi gerekmektedir. Çaprazlama oranı yeni birey oluşturma işleminin hangi sıklıkla yapılması gerektiğini belirtir. Bu oranın gerekenden büyük olması durumunda popülasyondaki değişkenlik hızlı bir şekilde gerçekleşmekte; küçük olması durumunda ise çaprazlama oranına bağlı çok küçük

değişimler meydana gelmekte ve algoritmadaki arama çok yavaş ilerlemektedir. Literatür incelendiğinde çaprazlama oranı 0.5-0.95 aralığında seçildiği görülmektedir. Grefenstette'e [141] göre çaprazlama oranının 0.95 seçilmesinin daha verimli sonuçlar meydana getirdiği belirtilmektedir. Yapılan başka bir çalışmada ise çaprazlama oranının 0.75 – 0.95 arasında seçilmesi gerektiği ortaya konmaktadır [142].

Bu çalışmada, literatürde yapılan çalışmalar göz önünde bulundurularak rasgele seçilen üç farklı P_c değeri (0.7, 0.8, 0.9) arasından uygun olanın tespiti için bir dizi test çalışması yapılmıştır.

Daire sayısı performans kriterine göre ANOVA testi sonucu 0,978 düzeyinde anlamlılık vermiştir. 0,05 hata seviyesine göre çaprazlama oranının istatistiksel olarak daire sayısı performansına etki etmediği sonucu ortaya çıkmıştır. Ayrıca daire sayısı kriterinin CPU zamanı performans ölçütüne göre çaprazlama oranının ANOVA testi sonucu istatistiksel olarak 0,05 hata seviyesinde anlamlı çıkmıştır. Duncan gruplandırma testine göre 0,05 hata seviyesinde çaprazlama oranlarının istatistiksel olarak anlamlı farklılığının olduğu ortaya çıkmıştır. $P_c=0,7$ oranının diğer oranlara göre daha iyi sonuç ürettiği görülmüştür. Alan performans kriteri için yapılan ANOVA testi sonucuna göre çaprazlama oranı anlamlılık düzeyi 0,991 çıkmıştır ve istatistiksel olarak 0,05 hata seviyesinde çaprazlama oranları arasında bir farklılık yoktur. Bu performans kriterinin CPU zamanı ile ilgili ANOVA sonucu ise 0,05 hata seviyesinde anlamlı çıkmıştır. Duncan testi gruplandırmasına göre yine $P_c=0,7$ oranının 0,05 hata seviyesinde daha iyi sonuçlar elde ettiği görülmektedir. Bu sebeple GA'nın bir parametresi olarak $P_c=0,7$ seçilmiştir.

Mutasyon oranının belirlenmesi

Mutasyon oranı (P_m) mutasyonun ne sıklıkla yapılacağını belirlemekte ve oran genellikle çok küçük değerlerden oluşmaktadır. Literatürdeki çalışmalar incelendiğinde, bu değer 0.005 – 0.01 arasındaki değerlerden seçilmesinin uygun olacağı ortaya konmaktadır [142]. P_m oranının büyük bir değerden oluşması, ebeveynden alınan kalıtsal özelliklerin kaybedilmesine ve yakınsamaya neden olabilmektedir. Ayrıca yüksek P_m değeri, GA'da rasgele aramaya dönüştüğü görülmektedir. Çok düşük P_m değeri ise algoritmanın yerel optimuma takılmasına sebep olmaktadır [143].

Bu çalışmada, literatürde yapılan çalışmalar göz önünde bulundurularak rasgele seçilen üç farklı P_m değeri (0.05, 0.1, 0.2) arasından uygun olanın tespiti için test çalışması yapılmış ve iki farklı amaç fonksiyonuna göre elde edilen uygunluk ve CPU zamanı değerleri için sonuçlar analiz edilmiştir.

Daire sayısı performans kriterine göre ANOVA testi sonucu 0,05 hata seviyesine göre mutasyon oranının istatistiksel olarak daire sayısı performansına etki etmediği sonucu ortaya çıkmıştır. Ayrıca daire sayısı kriterinin CPU zamanı performans ölçütüne göre de mutasyon oranının ANOVA testi sonucu istatistiksel olarak 0,05 hata seviyesinde anlamlı olmadığı ortaya çıkmıştır. Alan performans kriteri için yapılan ANOVA testi sonuçları daire sayısı performans kriterine göre yapılan ANOVA testi sonuçlarıyla benzer çıkmıştır. Ekte verilen varyans analizi sonuçlarına göre $P_m=0,2$ seçilmiştir.

Durdurma kriterinin seçilmesi

Durdurma kriteri, GA'daki işlem döngüsünün hangi koşulda sonlandırılması gerektiğini belirleyen bir yapıdır ve algoritmanın verimliliğinde önemli bir etkidir. Literatürde genetik algoritma tabanlı yöntemlerin kullanıldığı farklı optimizasyon problemleri için uygulanan çeşitli sonlandırma kriterleri mevcut olsa da, genel kabul görmüş bir sonlandırma kriteri yoktur[144].

Bu çalışma için durdurma kriteri olarak jenerasyon sayısı seçilmiştir. Literatürde yapılan çalışmalar göz önünde bulundurularak paketlenecek daire sayısına (n) bağlı rastgele jenerasyon sayısı ($25n$, $50n$, $100n$) değerleri belirlenmiştir. Belirlenen jenerasyon sayısı arasından uygun olanın durdurma kriteri olarak belirlenebilmesi için bir dizi test çalışması yapılmıştır.

Daire sayısı performans kriterine göre ANOVA testi sonucu 0,968 düzeyinde anlamlılık vermiştir. 0,05 hata seviyesine göre durdurma kriteri istatistiksel olarak daire sayısı performansına etki etmediği sonucu ortaya çıkmıştır. Ayrıca daire sayısı kriterinin CPU zamanı performans ölçütüne göre durdurma kriteri ANOVA testi sonucu istatistiksel olarak 0,05 hata seviyesinde anlamlı çıkmıştır. Duncan gruplandırma testine göre 0,05 hata seviyesinde durdurma kriterinin istatistiksel olarak anlamlı farklılığının olduğu ortaya çıkmış ve $25n$ değerinin diğer değerlere göre daha iyi sonuç ürettiği görülmüştür.

Alan performans kriteri için yapılan ANOVA testi sonucuna göre durdurma kriteri anlamlılık düzeyi 0,991 çıkmıştır ve istatistiksel olarak 0,05 hata seviyesinde durdurma kriterleri arasında bir farklılık yoktur. Bu performans kriterinin CPU zamanı ile ilgili ANOVA sonucu ise 0,05 hata seviyesinde anlamlı çıkmıştır. Duncan testi gruplandırmasına göre yine 25n değerinin 0,05 hata seviyesinde daha iyi sonuçlar elde ettiği görülmektedir. Bu sebeple GA'nın bir parametresi olarak durdurma kriteri 25n olarak seçilmiştir.

Elde edilen test sonuçların analizi sonucu Çizelge 3.4'deki parametreler belirlenmiştir.

Çizelge 3.4. Test Sonuçlarına göre belirlenen parametreler

Popülasyon Büyüklüğü	Çaprazlama Oranı	Mutasyon Oranı	Durdurma Kriteri
3n	0.7	0.2	25n

Parametre optimizasyonu ile belirlenen popülasyon büyüklüğü, çaprazlama oranı, mutasyon oranı ve durdurma kriteri değerleri önerilen genetik algoritma modelinde ve KDS uygulamasında kullanılmıştır. Ayrıca bölüm 4'te literatürden elde edilen ve dstayları sunulan problem setleri, önerilen genetik algoritma modeli ile çözülmüş ve elde edilen sonuçlar karşılaştırılarak sonuçlar değerlendirilmiştir.

3.4. DPP için Geliştirilen KDS Uygulaması

DPP'nin çözümü için geliştirilen genetik algoritma tabanlı çözüm modelinin kullanıcılar tarafından kolaylıkla kullanılabilmesi ve problem çözümünün anlaşılır bir şekilde çıktı verebilmesi için bir KDS geliştirilmiştir. Geliştirilen KDS'nin özellikle daire palet yükleme problemlerinde etkili ve hızlı karar verme ihtiyacı olan işletmelerde karar vericilere yardımcı olması amaçlanmıştır.

Sistem, Visual Studio platformunda C# programlama dili ile geliştirilmiştir. Yazılım, DPP'nin daire sayısı ve alan maksimizasyonu amaç fonksiyonları için özdeş ya da özdeş olmayan farklı yarıçaptaki dairelerin paletlere yerleşimini yapılabilmektedir. Resim 4.1'de gösterildiği üzere palet yarıçapının ve yerleşmeye aday dairelerin yarıçaplarının metin kutusuna girilmesiyle birlikte, problemin amaç fonksiyonu seçilmektedir. DPP'nin çözümü için istenen verilerin girilmesi sonrası yerleştirilme işlemi başlatılmaktadır.

DPP Karar Destek Sistemi

Menü

Daire Paketleme Problemi Karar Destek Sistemi

Palet Yanıçapı: 16.33

Daire Yanıçap Listesi: 1.00, 1.02, 1.09, 1.28, 1.38, 1.45, 1.53, 1.60, 1.63, 1.66, 1.68, 1.72, 1.85, 2.05, 2.07, 2.57, 2.69, 2.75, 2.92, 2.94, 2.97, 3.10, 3.16, 3.44, 3.65, 3.67, 3.75, 3.76, 3.94, 3.97, 4.09, 4.13, 4.20, 4.21, 4.24, 4.35, 4.63, 4.72, 4.85, 4.96

Amaç:

☒ Maksimum Alan Kullanımı

☐ Maksimum Daire Sayısı

YERLEŞTİR

Resim 3.1. DPP için KDS arayüzü

Resim 3.1’de “YERLEŞTİR” butonuyla birlikte DPP için tasarlanan genetik algoritma tabanlı model çalışmaya başlamaktadır. Modelin çalışmasıyla birlikte en iyi uygunluk değerine sahip sonuç elde edilmektedir. Resim 3.2’de gösterilen sonuç ekranında daire sayısı maksimizasyonu amaç fonksiyonuna göre yerleşen ve yerleşemeyen daire sayısı, dairelerin palet üzerindeki yerleşim sırası görülmektedir.

DPP Karar Destek Sistemi

Menü

Sonuç Ekranı:

Veriler

Amaç: Daire Sayısı Maksimizasyonu

Palet Yanıçapı: 16.33

Daire Sayısı: 40

Daire Yanıçapları: 1.00, 1.02, 1.09, 1.28, 1.38, 1.45, 1.53, 1.60, 1.63, 1.66, 1.68, 1.72, 1.85, 2.05, 2.07, 2.57, 2.69, 2.75, 2.92, 2.94, 2.97, 3.10, 3.16, 3.44, 3.65, 3.67, 3.75, 3.76, 3.94, 3.97, 4.09, 4.13, 4.20, 4.21, 4.24, 4.35, 4.63, 4.72, 4.85, 4.96

Sonuç

Yerleşen Daire Sayısı: 31

Yerleşmeyen Daire Sayısı: 9

Dairelerin Yerleşim Sırası: 2.57, 4.09, 3.94, 3.67, 3.44, 1.38, 1.09, 1.63, 2.94, 3.76, 1.68, 3.1, 1.72, 1.28, 2.75, 3.97, 2.97, 1.02, 3.65, 3.76, 1.53, 2.05, 2.92, 1.66, 3.16, 1.2, 2.69, 2.07, 1.6, 1.85, 1.45

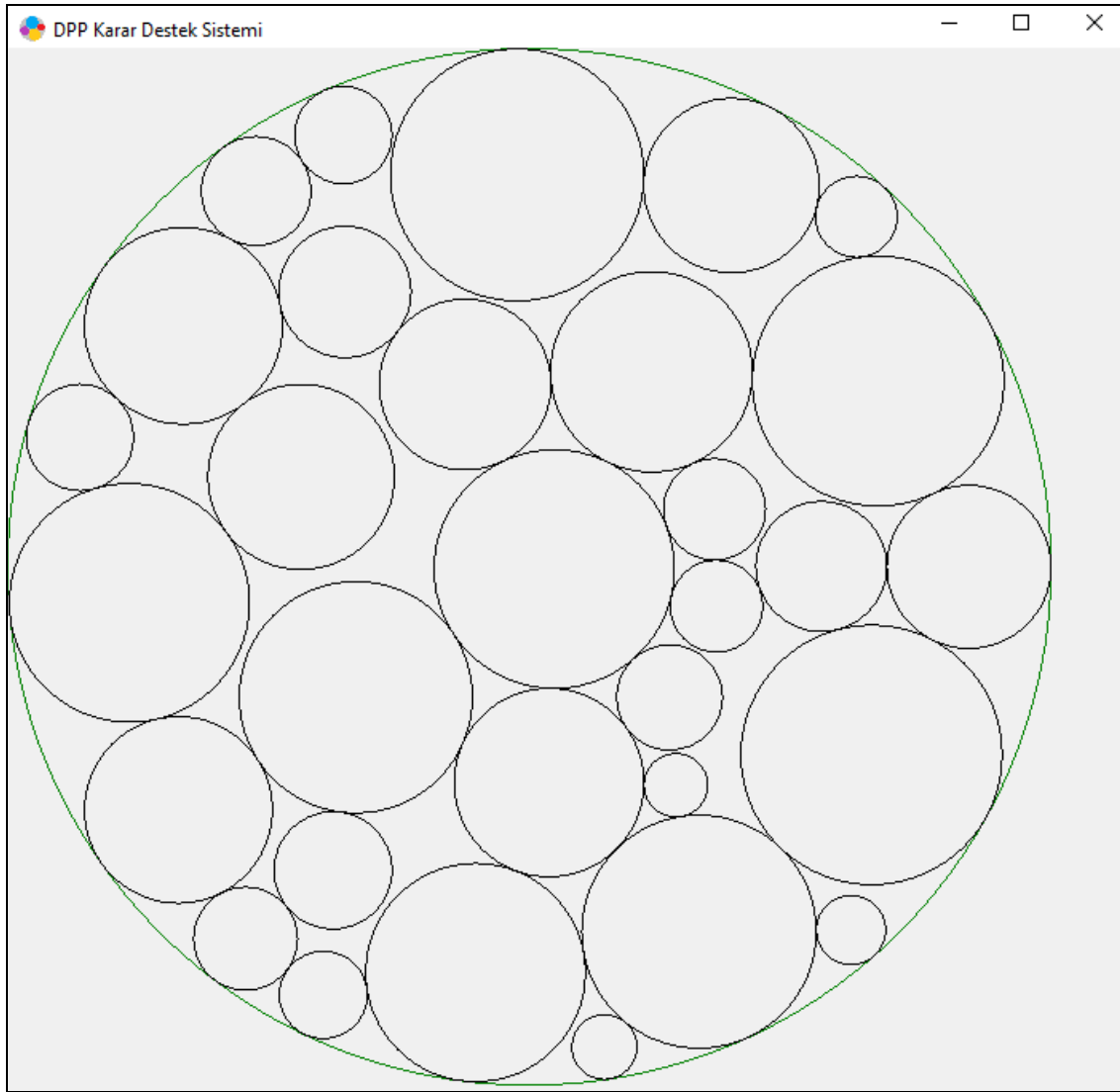
Daire Koordinatları Göster

Palet Desenini Göster

Resim 3.2. KDS sonuç ekranı

Resim 3.2’de daha ayrıntılı sonuçlar için iki ayrı buton konulmuştur. Eğer dairelerin palet üzerindeki deseninin görülmesi isteniyor ise, “Palet Desenini Göster” butonuyla birlikte elde edilen en iyi çözümün palet üzerindeki deseni oluşturulmaktadır. . Resim 3.3’de gösterilen palet deseni, problemin amaç fonksiyonuna göre en iyi uygunluk değerine sahip çözümün yerleşim durumun göstermektedir.

Yerleşen her bir dairenin paletin merkez noktasına göre koordinatları “Daire Koordinatlarını Göster” butonu ile elde edilmektedir. Resim 3.4’de gösterildiği üzere, palet deseninde gösterilen yerleşim şeklinin paletin orijinine göre dairelerin (x,y) koordinatlarını göstermektedir.



Resim 3.3. KDS’de palet deseni gösterimi

Yerleşen Daire Verileri		
Daire Yarıçapları	X Koordinatları	Y Koordinatları
2.57	13.76	0
4.09	10.7121947674419	-5.92169597872022
3.94	10.918226744186	5.85699793089775
3.67	5.31005691692461	-11.4925582678106
3.44	-1.68246023461478	-12.7797272098797
1.38	-6.45030322070632	-13.4868857917959
1.09	10.0697589473474	-11.4392986997594
1.63	-8.88215147950338	-11.7131287491923
2.94	-10.9854182965684	-7.65589216547758
3.76	-12.5187697855916	-1.133712068983
1.68	-14.0702988971643	4.0803417680945
3.1	-10.8319217316248	7.59620771174369
1.72	-8.55405339476384	11.843997235711
1.28	10.2367200125705	11.0323190392699
2.75	6.32843874983468	12.0152928798923
3.97	-0.383016688704658	12.3540640364284
2.97	0.61618301008833	-6.79605448875093
1.02	2.34807281408707	-15.1288682345953
3.65	-5.43288555736475	-4.10659925278751
3.76	0.77985781583512	-0.0680450834391584
1.53	-5.82329100004074	13.6062221769617
2.05	9.14002004918861	0.0136108082357265
2.92	-7.1464263863597	2.836215688638
1.66	4.38872114796891	-4.11186820642317
3.16	3.82412571029276	6.14636019677671
1	4.58531353685911	-6.87913794953789
2.69	-2.01218133722647	5.74633513871449
2.07	-5.7733389825361	8.66374720544883
1.6	5.79819668181603	1.81500431213858
1.85	-6.14831205952284	-9.55987028217991
1.45	5.8576208224927	-1.23441674299712

Resim 3.4. KDS'de yerleşen dairelerin yarıçap ve koordinat verileri



4. ARAŞTIRMA BULGULARI VE TARTIŞMA

Yerleşim algoritması ve önerilen genetik algoritmanın geliştirilen uygulama üzerinde performansı ve etkinliğini ölçebilmek için bir deneysel çalışma yapılmıştır. Bu deneysel çalışmada genetik algoritmanın parametreleri için bir parametre optimizasyonu yapılarak algoritmanın verimli ve etkin çalışabilmesi için testler gerçekleştirilmiştir.

Önerilen modelin daha önce çalışılmış verilerle test edilmesi ve elde edilen sonuçların karşılaştırılması için 2016 yılında yapılan Lopez ve Beasley'in [79] çalışmasından yararlanılmıştır. Araştırmacılar bu çalışmada, özdeş olmayan dairelerin, daire paletlere yerleşimi için oluşturdukları problem setlerini ve çözüm sonrası elde ettikleri sonuçları sunmuşlardır. Yerleşimde iki farklı amaç fonksiyonu ele alınmaktadır. Bu amaç fonksiyonları paketlenmiş daire sayısının maksimizasyonu ve paketlenmiş dairelerin palet üzerinde kapladıkları alanın maksimizasyonu olarak bildirilmiştir. Lopez ve Beasley ele aldıkları amaç fonksiyonunu tasarladıkları modelde uygulayabilmek için üç farklı problem seti kullanmıştır. Bu problem setlerinden birincisi Lopez ve Beasley tarafından oluşturulmuş olup 10 ile 40 adet özdeş olmayan dairelerin paketlenmesini içeren dört farklı daire grubundan oluşmaktadır. İkinci problem seti, Zhang ve Deng'in [76] çalışmasından alınmış olup 12 ile 17 adet özdeş olmayan dairelerin paketlenmesini içeren dört farklı daire grubundan oluşmaktadır. Üçüncü problem seti ise Packomania'dan [145] alınmıştır. Bu sette birim yarıçaplı paletlere, yarıçapları bilinen özdeş dairelerin yerleşimi gerçekleştirilmiştir.

4.1. Deney Ortamının Özellikleri

Lopez ve Beasley'in [79] çalışması ile karşılaştırmanın yapılabilmesi için Visual Studio platformunda C# tabanlı bir yazılım geliştirilmiştir. Yazılım, problem setleri içinde bulunan farklı sayıdaki daire gruplarının ve farklı amaç fonksiyonlarının tamamında sorunsuz çalışabilecek şekilde yazılmıştır. Yazılım, Intel i7-9400 3.00 Ghz x64 tabanlı işlemci ve 16 GB bellekli bilgisayar üzerinde çalıştırılarak, hesaplama sonuçları elde edilmiştir.

4.2. Problem Setlerinin Özellikleri

Bu bölümde Lopez ve Beasley'in [79] çalışmasında kullandıkları üç farklı problem setlerinin detaylı açıklamaları verilmiştir.

4.2.1. Problem seti 1'in özellikleri

Problem Seti 1'in elde edildiği çalışmadan [79] 12 farklı test problemi alınarak karşılaştırma için kullanılmıştır. Bu test problemlerindeki daire ve paletlerin yarıçapları şu şekilde belirlenmiştir:

- 10, 20, 30, 40 adet daireden oluşan 4 farklı daire grubu oluşturulmuştur.
- Her bir gruptaki dairelerin yarıçapları (R_i) [1,5] aralığında ve rasgele iki ondalıklı değer üretilerek elde edilmiştir.
- Her daire grubundaki dairelerin yerleşeceği üç farklı palet oluşturulmuştur. Paletlerin yarıçapları (R_0) gruptaki dairelerin alanlarının toplamının 1/3, 1/2, 2/3'ü alınarak üç farklı palet yarıçapı geliştirilmiştir.

Problem Seti 1 için belirlenen 12 farklı test probleminde kullanılan daire ve paletlerin sayıları ve yarıçapları Çizelge 4.1'de gösterilmiştir.

Çizelge 4.1. Problem Seti 1'deki test problemleri

Daire sayısı (n)	Daire yarıçapları (R_i)	Palet yarıçapları (R_0)
10	1.08, 1.10, 2.17, 3.21, 3.91, 4.27, 4.31, 4.34, 4.40, 4.72	6.58 8.06 9.31
20	1.02, 1.19, 1.23, 1.43, 1.55, 1.77, 1.85, 2.11, 2.14, 2.32, 2.65, 3.04, 3.13, 3.16, 3.29, 4.37, 4.38, 4.56, 4.60, 4.70	7.72 9.45 10.91
30	1.03, 1.38, 1.40, 1.46, 1.65, 1.99, 2.04, 2.15, 2.16, 2.21, 2.39, 2.53, 2.60, 2.87, 3.01, 3.09, 3.11, 3.17, 3.29, 3.39, 3.42, 3.79, 3.92, 3.98, 4.37, 4.43, 4.46, 4.88, 4.94, 4.99,	10.15 12.43 14.35
40	1.00, 1.02, 1.09, 1.28, 1.38, 1.45, 1.53, 1.60, 1.63, 1.66, 1.68, 1.72, 1.85, 2.05, 2.07, 2.57, 2.69, 2.75, 2.92, 2.94, 2.97, 3.10, 3.16, 3.44, 3.65, 3.67, 3.76, 3.76, 3.94, 3.97, 4.09, 4.13, 4.20, 4.21, 4.24, 4.35, 4.63, 4.72, 4.85, 4.96,	11.55 14.14 16.33

4.2.2. Problem seti 2'nin özellikleri

Lopez ve Beasley [79], Zhang ve Deng'in [76] çalışmasındaki 9 test problemi arasından özdeş olmayan daire paketleme problemini içeren 4 test problemini kullanarak karşılaştırma yapmıştır. Çizelge 4.2'de de gösterildiği üzere 12, 15, 17, 17 adet daireden oluşan 4 farklı daire grubu bulunmaktadır. Zhang ve Deng'in çalışmasında belirlenen palet yarıçapına ek olarak, Problem Seti 1'de kullanılan palet yarıçapının belirlenme yöntemindeki gibi her bir gruptaki dairelerin alanlarının toplamının $1/3$, $1/2$, $2/3$ 'ü alınarak üç palet yarıçapı (R_0) daha eklenmiştir. Bu şekilde 16 adet farklı test problemi elde edilmiştir.

Çizelge 4.2. Problem Seti 2'deki test problemleri

Daire sayısı (n)	Daire yarıçapları (R_i)	Palet yarıçapları (R_0)
12	23.72, 23.72, 23.72, 23.72, 23.72, 23.72, 48.26, 48.26, 48.26, 100, 100, 100	115.99 142.06 164.04 215.47
15	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15	20.33 24.90 28.75 39.37
17	25, 20, 15, 15, 10, 10, 10, 5, 5, 5, 5, 5, 5, 5, 5, 5	25.98 31.82 36.74 50
17	100, 100, 100, 100, 41.415, 41.415, 41.415, 41.415, 41.415, 20, 20, 20, 20, 20, 20, 20, 20, 20,	131.37 160.90 185.79 241.43

4.2.3. Problem seti 3'ün özellikleri

Sadece daire paketleme problemleri üzerine çalışmaların yapıldığı Packomania [145] adlı sitede çözülen problemlerden yararlanılarak bir problem seti oluşturulmuştur. Bu site dünyadaki birçok araştırmacının çeşitli formlardaki daire paketleme problemlerinin çözümüne katkı sunduğu bir site olarak bilinmektedir. Bu problemler, 1-5000 arasındaki

özdeş dairelerin birim yarıçaplı daire paletler içine yerleşmesi sırasında daire yarıçap maksimizasyonu olarak çözülmektedir.

Bu çalışmada, Packomania’da [145] çözüm sonuçları verilen 5, 10, 15, 20, 25, 30, 35, 40 dairesel problemler çalışılmıştır. Çizelge 4.3.’de gösterildiği üzere 8 farklı test problemi oluşturulmuştur. Her grup için palet yarıçapları 1 birimden oluşmakta ve her bir test problemindeki dairelerin yarıçapları Packomania’daki [145] sonuçlardan alınmıştır. Bu test problemlerinde sadece daire sayısı maksimizasyonu problemi çözülmüştür.

Çizelge 4.3. Problem Seti 3’deki test problemleri

Daire sayısı (n)	Daire yarıçapı (R _i)	Palet yarıçapı (R ₀)
5	0.370191908158750137702237641058	1
10	0.262258924190165855095630653709	1
15	0.221172539086390937264316484926	1
20	0.195224011018748878291305694833	1
25	0.173827661421222233359259594413	1
30	0.161349109064689865167358846536	1
35	0.149316776635116022601721177537	1
40	0.140373604202714704755923179430	1

4.3. Karşılaştırma Sonuçları

Bu bölümde Lopez ve Beasley’in [79] çalışmasında kullandıkları üç farklı problem setinin DPP’nin çözümü için önerilen modelle karşılaştırmalı sonuçlarına yer verilmiştir.

4.3.1. Problem seti 1’in sonuçları

Problem seti 1’deki 12 farklı test problemi iki amaç fonksiyonuna göre çözülmüştür. Her bir test problemi için elde edilen sonuçlar Lopez ve Beasley’in [79] çalışmasındaki sonuçlarla

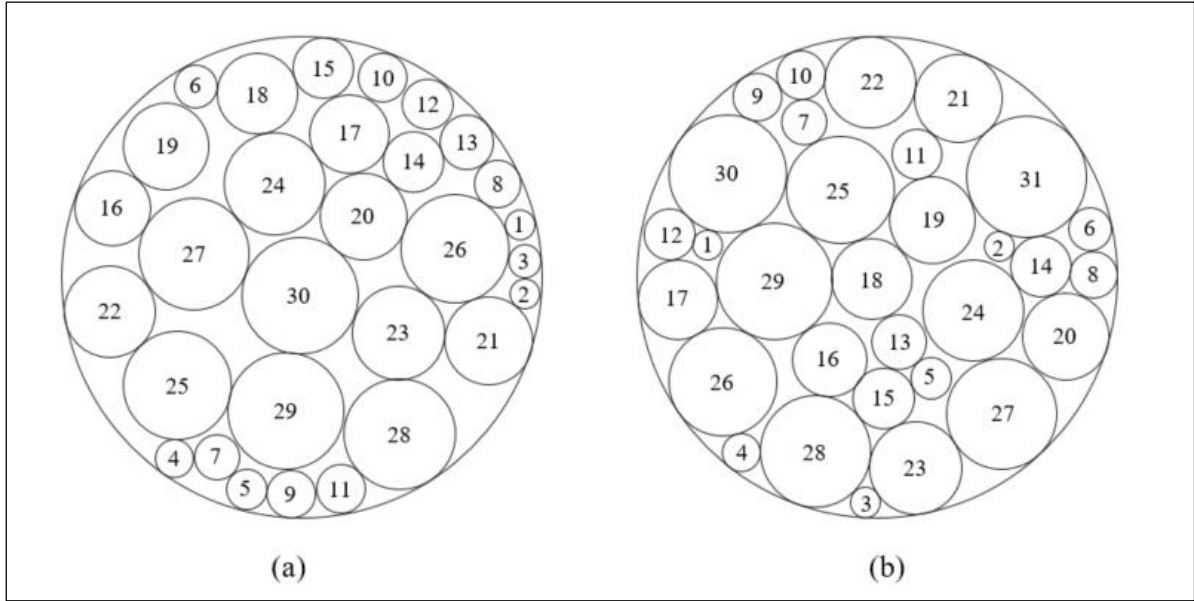
karşılaştırılarak, daire maksimizasyonu amaç fonksiyonu ve alan maksimizasyonu amaç fonksiyonuna ait sonuçlar sırasıyla Çizelge 4.4’de ve Çizelge 4.5’de gösterilmiştir.

Daire maksimizasyonu amaç fonksiyonuyla çözülen ve sonuçları Çizelge 4.4’de gösterilen 12 test problemi, önerilen genetik algoritmanın sonuçlarıyla karşılaştırıldığında 12 problem arasından sadece $n=40$ ve $R_0 = 16.33$ olan problemde daire sayısı açısından bir iyileşmenin olduğu görülmektedir. Bununla birlikte önerilen modelin CPU zamanları, karşılaştırılan çalışmaya göre ciddi oranda düşüşler gözlemlenmiştir. Lopez ve Beasley, $n=40$ ve $R_0 = 16.33$ olan test probleminde 30 dairenin yerleşimini 2800sn’de gerçekleştirmişken, önerilen modelle palet 31 daire 13,32 sn’ye yerleştirilmiştir.

Lopez ve Beasley’in çalışmasındaki $n=40$ ve $R_0 = 16,33$ probleminin deseni Şekil 4.1 (a)’da, önerilen modelle yerleşimi yapılan paletin deseni Şekil 4.1 (b) gösterilmiştir.

Çizelge 4.4. Daire maksimizasyonu karşılaştırma sonuçları

Daire Sayısı (n)	Palet Yarıçapı (R_0)	Lopez ve Beasley [79] Sonuçları		Önerilen Genetik Algoritma Sonuçları		Daire Sayısındaki İyileşme Oranı (%)
		En İyi Daire Sayısı	En İyi CPU Zaman (sn)	En İyi Daire Sayısı	En İyi CPU Zaman (sn)	
10	6,58	4	1	4	0,04	0
	8,06	5	1	5	0,03	0
	9,31	6	102	6	0,04	0
20	7,72	12	381	12	0,26	0
	9,45	14	212	14	0,28	0
	10,91	15	658	15	0,35	0
30	10,15	16	1416	16	1,87	0
	12,43	20	1600	20	1,26	0
	14,35	22	1500	22	2,25	0
40	11,55	22	2000	22	3,92	0
	14,14	26	2000	26	6,72	0
	16,33	30	2800	31	13,32	3,22



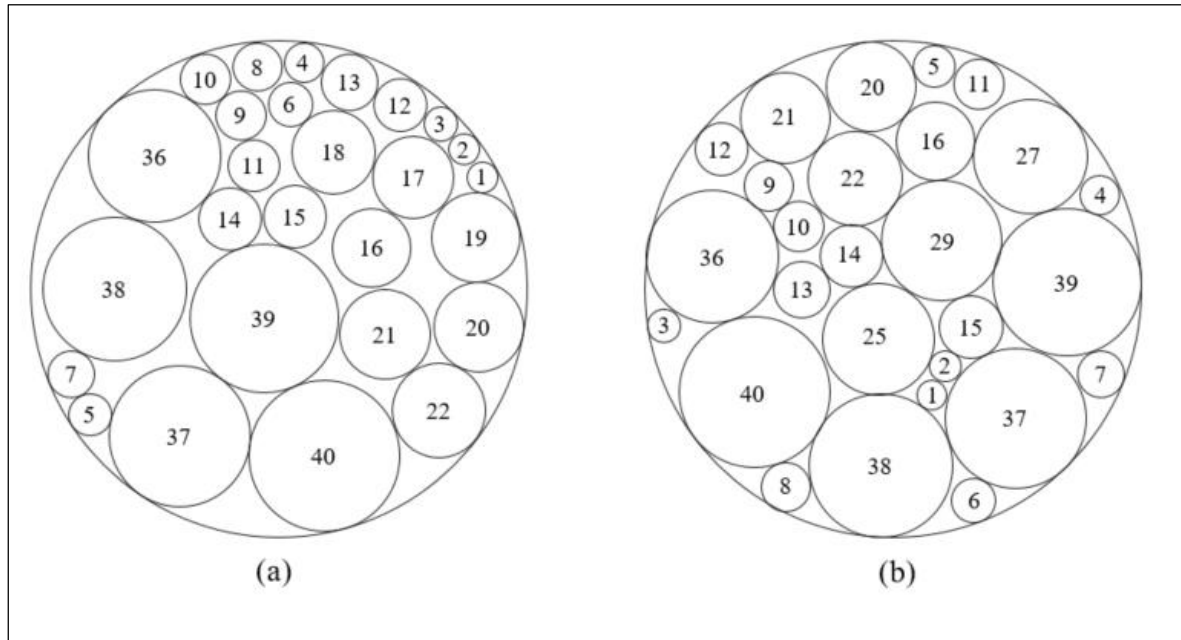
Şekil 4.1. $n=40$, $R_0=16.33$ probleminin daire maksimizasyonu karşılaştırması

Alan maksimizasyonu amaç fonksiyonuyla çözülen ve sonuçları Çizelge 4.5’de gösterilen 12 test problemi, önerilen genetik algoritmanın sonuçlarıyla karşılaştırıldığında, $n=10$ olan test problemleri haricindeki bütün test problemlerinde palet alanı kullanımında % 3,94-%9,07 arası iyileşmenin olduğu görülmektedir. Bununla birlikte bütün problemlerde önerilen modelin CPU zamanları, karşılaştırılan çalışmayla karşılaştırıldığında ciddi oranda düşüşlerin olduğu gözlemlenmiştir. Öyle ki, $n=40$ ve $R_0 = 16,33$ olan test probleminde daha önce 13585 sn olan çözüm süresi, önerilen modelle birlikte daha iyi sonucu 39,15 sn’de elde etmiştir.

Şekil 4.2(a)’da gösterilen Lopez ve Beasley’in [79] çalışmasındaki $n=40$ ve $R_0=16,33$ olan test problemindeki alan maksimizasyonu sonucunda 642,9087 br² olan yerleşim alanı, önerilen genetik algoritmanın çalıştırılmasıyla birlikte alan kullanımı Şekil 4.2(b)’de gösterilen palette alan kullanımı 704,4324 br² çıkarak %8,73 oranında iyileşmenin olduğu görülmüştür.

Çizelge 4.5. Alan maksimizasyonu karşılaştırma sonuçları

Daire Sayısı (n)	Palet Yarıçapı (R_0)	Lopez ve Beasley [79] Sonuçları		Önerilen Genetik Algoritma Sonuçları		Alandaki İyileşme Oranı (%)
		En İyi Alan	En İyi CPU Zaman (sn)	En İyi Alan	En İyi CPU Zaman (sn)	
10	6,58	83,0804	577	83,0804	0,04	0
	8,06	124,6201	17	124,6201	0,04	0
	9,31	197,0718	3919	197,0718	0,05	0
20	7,72	143,7552	4851	151,9764	0,37	5,41
	9,45	219,0655	5164	229,2404	0,43	4,44
	10,91	290,5062	5689	303,3551	0,37	4,24
30	10,15	254,2444	9549	264,6657	0,86	3,94
	12,43	372,0648	7585	398,2182	1,42	6,57
	14,35	502,0171	7046	530,1438	1,98	5,31
40	11,55	318,9141	8967	350,7425	4,21	9,07
	14,14	486,9406	8967	524,9108	6,97	7,23
	16,33	642,9087	13585	704,4324	39,15	8,73

Şekil 4.2. n=40, $R_0=16.33$ probleminin alan maksimizasyonu karşılaştırması

4.3.2. Problem seti 2'nin sonuçları

Zhang ve Deng'e [76] ait 4 test problemi, Lopez ve Beasley'in [79] tarafından genişletilerek 16 test problemi haline getirilmiş ve iki farklı amaç fonksiyonuna göre çözülmüştür. Lopez ve Beasley'in sonuçları önerilen modelle birlikte karşılaştırılarak daire maksimizasyonu amaç fonksiyonuna göre olan sonuçlar Çizelge 4.6'da, alan maksimizasyonu amaç fonksiyonuna göre olan sonuçlar ise Çizelge 4.7'de gösterilmiştir.

Daire maksimizasyonu amaç fonksiyonuyla çözülen ve sonuçları Çizelge 4.6'da gösterilen 16 test problemi, önerilen genetik algoritmanın sonuçlarıyla karşılaştırıldığında, 7 test probleminde daire sayısında % 6,25 ile % 11,11 oranında iyileşmenin olduğu görülmektedir. Ayrıca bu setteki bütün problemler önerilen modelle çözüldüğünde CPU zamanlarında, karşılaştırılan çalışma göre ciddi oranda düşüşün olduğu gözlemlenmiştir.

Çizelge 4.6. Daire maksimizasyonu karşılaştırma sonuçları

Daire Sayısı (n)	Palet Yarıçapı (R ₀)	Zhang ve Deng [51] probleminin Lopez ve Beasley [43] göre sonuçları		Önerilen Genetik Algoritma Sonuçları		Daire Sayısındaki İyileşme Oranı (%)
		En İyi Daire Sayısı	En İyi CPU Zaman (sn)	En İyi Daire Sayısı	En İyi CPU Zaman (sn)	
12	115,99	8	282	8	0,02	0
	142,06	9	1	9	0,02	0
	164,04	9	161	10	0,02	11,11
	215,47	11	162	12	0,02	9,09
15	20,33	9	202	9	0,03	0
	24,90	10	202	10	0,05	0
	28,75	11	203	12	0,09	9,09
	39,37	14	202	15	0,08	7,14
17	25,98	12	229	12	0,07	0
	31,82	13	230	14	0,10	7,70
	36,74	15	4	15	0,11	0
	50	16	228	17	0,10	6,25
17	131,37	13	4	13	0,07	0
	160,90	13	227	13	0,09	0
	185,79	14	1	14	0,08	0
	241,43	16	420	17	0,16	6,25

Alan maksimizasyonu amaç fonksiyonuyla çözülen ve sonuçları Çizelge 4.7’da gösterilen 16 test problemi, önerilen genetik algoritmanın sonuçlarıyla karşılaştırıldığında, 12 test probleminde palet alanı kullanımında, % 0,62 ile % 8,63 oranında iyileşmenin olduğu görülmektedir. Bununla birlikte bütün problemlerde önerilen modelin CPU zamanları, karşılaştırılan çalışmayla karşılaştırıldığında ciddi oranda düşüşlerin olduğu gözlemlenmiştir. Öyle ki, $n=17$ ve $R_0 = 241,43$ olan test probleminde daha önce 7672 sn olan çözüm süresi, önerilen modelle birlikte daha iyi sonucu 0,18 sn’de elde etmiştir.

Çizelge 4.7. Alan maksimizasyonu karşılaştırma sonuçları

Daire Sayısı (n)	Palet Yarıçapı (R_0)	Zhang ve Deng [43] Sonuçları		Önerilen Genetik Algoritma Sonuçları		Alandaki İyileşme Oranları (%)
		En İyi Alan	En İyi CPU Zaman (sn)	En İyi Alan	En İyi CPU Zaman (sn)	
12	115,99	31415,93	2682	31415,93	0,01	0
	142,06	42021,41	2033	42021,41	0,03	0
	164,04	63971,98	4814	63971,98	0,03	0
	215,47	126803,84	614	126803,84	0,03	0
15	20,33	1014,73	4475	1021,02	0,06	0,62
	24,90	1460,84	3562	1545,66	0,11	5,81
	28,75	1979,20	4693	2079,73	0,10	5,08
	39,37	3741,64	5347	3895,58	0,16	4,11
17	25,98	1963,50	3792	1963,50	0,01	0
	31,82	2513,27	3857	2513,27	0,10	0
	36,74	3377,21	3857	3377,21	0,16	0
	50	6361,73	3491	6361,73	0,36	0
17	131,37	41469,02	2811	41469,02	0,07	0
	160,90	63384,81	7949	64641,45	0,10	1,98
	185,79	68411,36	2892	68411,36	0,09	0
	241,43	149730,85	7672	162659,14	0,18	8,63

4.3.3. Problem seti 3'ün sonuçları

Problem seti 3'deki 8 farklı test problemi daire sayısı amaç fonksiyonuna göre çözülmüştür. Her bir test problemi için elde edilen sonuçlar Lopez ve Beasley'in [79] çalışmasındaki sonuçlarla karşılaştırılarak sonuçlar Çizelge 4.8'de gösterilmiştir. Çizelgede görüldüğü üzere sadece daire sayısında karşılaştırması yapılan çalışmayla kıyaslandığında bir iyileşme görülmemekle beraber CPU zamanlarında belirgin bir düşüşün olduğu saptanmıştır. $n=5, 10, 15, 20, 25, 30$ dairesel problemlerde tüm dairelerin yerleşiminin gerçekleşmesi sebebiyle iyileşmenin olması beklenmemektedir. $n=35, 40$ dairesel problemlerde ise karşılaştırması yapılan çalışma ve önerilen çalışmada 1 dairenin yerleşemediği görülmektedir.

Çizelge 4.8. Problem Seti 3 için daire sayısı maksimizasyonu karşılaştırma sonuçları

Daire Sayısı (n)	Palet Yarıçapı (R_0)	Lopez ve Beasley [79] Sonuçları		Önerilen Genetik Algoritma Sonuçları		Daire Sayısındaki İyileşme Oranları (%)
		En İyi Daire Sayısı	En İyi CPU Zaman (sn)	En İyi Daire Sayısı	En İyi CPU Zaman (sn)	
5	1	5	1	5	0,03	0
10	1	10	2	10	0,07	0
15	1	15	7	15	0,26	0
20	1	20	7	20	0,54	0
25	1	25	596	25	1,41	0
30	1	30	1009	30	3,34	0
35	1	34	2682	34	5,00	0
40	1	39	1928	39	15,25	0

4.4. Sonuçların değerlendirilmesi

DPP için önerilen karar destek modelinde özgün bir yerleşim algoritması geliştirilmiş ve optimizasyon problemlerinin çözümünde sıklıkla başvurulmuş GA ile optimal çözümün arandığı bir çalışma yapılmıştır. GA, rasgele araştırma algoritmalarının bir türüdür ve diğer evrimsel algoritmalar gibi araştırma uzayında bulunan çözümlerin bir kısmının oluşturduğu bir başlangıç popülasyonu kullanmaktadır. Başlangıç popülasyonu genellikle rasgele sayı üreticisi kullanılarak oluşturulur [121]. Bu çalışmada, probleme özgün gen dizilimi ve kromozomlarla başlangıç popülasyonunun oluşturulması, turnuvaya girecek bireylerin seçimi, çaprazlamada birey ve çaprazlama noktasının belirlenmesi, mutasyona uğratılacak bireylerin seçimi sırasında GA'nın bu rasgele araştırma özelliği sıkça kullanılmıştır.

Bu çalışmada önerilen karar destek modelinin performansı bir başka çalışma ile karşılaştırılarak ölçülmüştür. Literatürde DPP'nin dairesel palet üzerindeki çözümünü hem daire sayısı maksimizasyonu, hem de alan maksimizasyonu amaç fonksiyonu ile gerçekleştiren Lopez ve Beasley'in [79] çalışması performans karşılaştırması için uygun bir çalışma olarak görülmüştür. Çalışmanın sunduğu üç farklı problem setinde, daire maksimizasyonuna göre 36 farklı problem, alan maksimizasyonuna göre 28 farklı problem olmak üzere toplam 64 farklı problem bulunmaktadır. Bütün problemler önerilen modelle çözülmüş ve genel olarak karşılaştırma sonuçları şu şekildedir:

Uygunluk değerine göre;

- Daire maksimizasyonu amaç fonksiyonu için çözülen 36 problemde 8'inde,
- Alan maksimizasyonu amaç fonksiyonu için çözülen 28 problemde 15'inde,

CPU zamanına göre;

- Daire maksimizasyonu amaç fonksiyonu için çözülen 36 problemin 36'sında
- Alan maksimizasyonu amaç fonksiyonu için çözülen 28 problemin 28'inde,

iyileşmenin olduğu görülmektedir.

Ayrıca, uygunluk değerine göre oluşan iyileşmenin iki amaç fonksiyonu için çözülen toplam 64 problemin 23'ünde iyileşme gerçekleşmiştir. İyileşme gerçekleşmeyen 41 problemin bir kısmında optimum sonuca ulaşıldığı için iyileşmenin mümkün olmadığı görülmektedir.

Lopez ve Beasley'in [79] çalışmasındaki problemlerin çözümü Intel i5-2400S 2.50 Ghz işlemci kullanılmış iken önerilen model Intel i7-9400 3.00 Ghz işlemcisi üzerinde çalıştırılmıştır. Karşılaştırması yapılan çalışmada kullanılan işlemci ile önerilen algoritmanın çalıştırıldığı işlemcinin, CPU performans uygulamasına [146] göre kıyas skorları sırasıyla 3132 ve 13437 dir. Sonuçlara bakıldığında, iki bilgisayar arasındaki CPU karşılaştırmasında görülen yaklaşık 4,3 katlık iyileşmenin getirdiği faydadan daha fazlasının elde edildiği görülmektedir. Bu fayda önerilen GA ile çözülen bütün problemlerde CPU'nun çalışma sürelerindeki yaklaşık %99 oranında düşüşle birlikte, uygunluk değerlerinde ise % 11,11'e kadar iyileşme görülmektedir.



5. SONUÇ VE ÖNERİLER

Günümüzde inşaat, elektronik, otomotiv, havacılık, denizcilik, savunma, gıda, ilaç, cam, tekstil endüstrilerinin hammadde tedarikinden müşteriye kadar ki tüm süreçlerinde farklı türlerde paketleme ve paletleme problemlerine çözüm aranmaktadır. Çoğunlukla daire paketleme ve yerleşimi olarak karşımıza çıkan boya endüstrisindeki ürünlerin nakliyesi, varillerin depolanması, şişelerin kutulanması, disklerin en az atık olacak şekilde kesilmesi, tarım alanlarının verimli yerleşimi gibi birçok faaliyet daire paketleme problemi olarak karşımıza çıkmaktadır. Bu çalışma kapsamında, daire paketleme problemi (DPP) ele alınmıştır.

DPP, birçok yerleşim, yükleme, kesme ve paketleme problemi ile benzerlik göstermektedir. Bu tür problemlerin çözümündeki ortak motivasyon, kaynakların verimli kullanımı, firenin ve atığın azaltılması, çözüm için gereken süresinin kısaltılmasıdır. Bu sebeple, birçok araştırmacı problemin çözümünde en iyiyi bulmak için yoğun çaba harcamaktadır. NP-zor sınıfında olan bu tür problemler kesin çözüm metotlarının aksine yaklaşık çözüm veren metotlarla bir başka deyişle metasezgisel yöntemlerle çözülebilmektedir. Metasezgisel yöntemler sayesinde arama süreci için etkin stratejiler geliştirilir ve arama uzayının etkili ve hızlı bir şekilde taranması sağlanır.

Bu çalışmada, DPP'nin çözümü için bir genetik algoritma tabanlı bir model önerilmiştir. Genetik algoritma, literatürde sıklıkla kullanılan NP-Zor sınıfındaki eniyileme problemlerini optimale yakın bir sürede çözebilen metasezgisellerden biridir. Dairelerin yerleşiminin belirli bir sistematikte olabilmesi için “Komşu ve Konum Seçimli Daire Yerleşim Algoritması” olarak adlandırılan özgün bir çözüm kurucu algoritma tasarlanmıştır. Bu algoritma ile dairelerin birbirleriyle çakışmama, paletten dışarı taşmama kısıtlarına ek olarak her bir daireye atanan komşuluk ve konum seçim değerlerine göre yerleşim yapılmaktadır. Geliştirilen yerleşim algoritmasıyla oluşturulan başlangıç çözümleri, genetik algoritma süreçlerinden geçirilerek en iyi çözümler aranmıştır.

Genetik algoritma geliştirme süreçlerinde kullanılacak parametre ve yöntemler için test ve analizler yapılmıştır. Seçim işlemi için turnuva yöntemi kullanılmıştır. Ancak uygun olan turnuva genişliğinin (k) belirlenmesi bir çalışma yapılmıştır. Literatürde sıklıkla kullanılan iki turnuva genişliği (k=2, 3) kullanılması sebebiyle bu iki değer için bir performans çalışması yapılmıştır. Yapılan çalışma neticesinde turnuva genişliğinin k=3 olması

algoritmanın etkin çalışması açısından uygun olduğu sonucuna varılmıştır. Çaprazlama operatöründe tek noktalı çaprazlama yönetimi (1PX) kullanılmıştır. Önerilen modelde kromozom üç farklı kısımdan oluşmaktadır. 1PX yönteminde belirlenen çaprazlama noktasının kromozomun üç kısmı için aynı gen aralığında olup olmayacağı ile ilgili bir çalışma yapılmıştır. Yapılan çalışma neticesinde, kromozomun her bir kısmı için farklı bir çaprazlama noktasının belirlenmesi algoritmanın etkin çalışması açısından daha uygun olduğu sonucuna varılmıştır.

Turnuva genişliği ve çaprazlama noktasının belirlenmesinden sonra parametre optimizasyonu çalışması yapılmıştır. GA'nın temel adımlarında kullanılacak olan parametrelerin iyi analiz edilerek belirlenmesi algoritmanın kalitesini ve başarısını olumlu yönde etkilemektedir. Bu sebeple, Lopez ve Beasley'in [79] çalışmasından elde edilen test problemlerinin bir kısmı kullanılarak popülasyon büyüklüğü (P_s), çaprazlama oranı (P_c), mutasyon oranı (P_m), durdurma kriteri parametrelerinin tespiti için önerilen algoritma kullanılarak sonuçlar üzerinde istatistiksel analizler gerçekleştirilmiştir. Analiz sonucunda n adet daire yerleşiminin yapıldığı bir DPP'de parametrelerin $P_s=3n$, $P_c=0,7$, $P_m=0,2$ ve durdurma kriteri= $25n$ olması algoritmanın etkin çalışması açısından daha uygun olduğu sonucuna varılmıştır.

Parametre optimizasyonu çalışması sonrası, önerilen modelin performansının karşılaştırılması için bir başka çalışma yapılmıştır. Literatürde DPP'nin dairesel palet üzerindeki çözümünü hem daire sayısı maksimizasyonu, hem de alan maksimizasyonu amaç fonksiyonu ile gerçekleştiren Lopez ve Beasley'in [79] çalışmasında kullandığı problem setleri karşılaştırma için kullanılmıştır. Üç farklı setten oluşan verilerde palet yarıçapı (R_0), yerleşecek dairelerin sayısı (n) ve dairelerin yarıçapları (R_i) bulunmaktadır. Bu verilerle, iki amaç fonksiyonuna yönelik karşılaştırma çalışması yapılmıştır.

Karşılaştırma yapılan çalışmanın daire maksimizasyonu test sonucunda uygunluk fonksiyonu değerlerinde 36 farklı test probleminin 8'inde % 11,11 'e kadar iyileştirilmenin olduğu görülmüştür. CPU zamanında ise 2800 sn olan çözüm süresinin 13,32 sn'ye kadar düşürüldüğü gözlemlenmiştir. Ayrıca $n=40$ ve $R_0=16,33$ palet yarıçaplı test probleminin daire maksimizasyonu problemi çözümünde, karşılaştırması yapılan çalışmada palete maksimum 30 daireyi çok uzun sürelerde yerleştirebildiği; önerilen modelin çalıştırılması sonucunda ise palette kalan boşluklara bakıldığında yeni yerleşiminin zor olacağı düşünülen palette, 31 dairenin çok daha kısa sürede yerleştirebildiği görülmüştür.

Alan maksimizasyonu amaç fonksiyonu için yapılan testlerde problem seti 3'ün dışındaki bütün problemlerde iyileşmenin olduğu görülmektedir. 28 test probleminin 15'inde %9.98' kadar iyileşmenin olduğu görülmektedir.

Bu tez çalışması kapsamında geliştirilen modelin sunduğu daire sayısı ve alan maksimizasyonundaki iyileşmeler yerleşen parça sayısının artmasını, atık alanların azalmasını, palet yüzeyinin etkin kullanımı gerçekleştirmektedir. Bu sayesinde bu tür problemlerinin çözümünde rol alan karar vericilerin karar verme süreçlerine olumlu katkı sunacağı görülmektedir. Ayrıca sağlanan iyileşmenin işletmelerin kar oranlarına, teslimat sürelerine ve nakliye maliyetlerine olumlu etki sağlaması kaçınılmazdır. Geliştirilen algoritma ve model dairesel alan üzerindeki daire paketleme ve yerleştirme problemlerine uyarlanabilir nitelik geliştirilmiştir. Bu durum yeni çözüm metotları açısından probleme eklenecek yeni kısıt, parametre ve değişken ile problemin farklı versiyonlarının geliştirilmesine imkân sağlamaktadır.

Gelecekte bu tez kapsamınsa yapılan çalışma ile ileri teknoloji cihazları için (insansız hava araçları, drone vb.) yapılan çalışmalarında kullanılan algoritmaların birleşmesiyle birlikte yararlı ve verimli projeler geliştirilebilir. Örneğin, tarım alanlarının ilaçlanması veya takibi sırasında kullanılacak drone gibi hava araçlarının havadan yere olan etki alanının dairesel olması sebebiyle verimli yüzey taraması DPP çözümleriyle mümkün olmaktadır. Ayrıca bu tür hava araçlarının gelecekte nesnelerin interneti (IoT) teknolojisini de kullanarak sürü halinde çok geniş alanı tarama ve işleme faaliyetini gerçekleştirebileceği düşünülmektedir. Bu faaliyet sırasında kullanılacak cihaz sayısının minimizasyonu ve kapsama alanının maksimizasyonunun yapılması verimlilik açısından gereklidir. Bu sebeple bu çalışma kapsamınsa önerilen model bu tür ihtiyaçlara çözüm olabilir. Bu bağlamda DPP'nin daha geniş alan ve sektörlerde çözüme katkı sunacağı gelecek çalışma hedeflerinde yer alacağı düşünülmektedir.

Bu çalışmada, dairelerle ilgili üst üstte binmeme ve taşmama kısıtlarına ek olarak daha etkin bir yerleşim sağlaması açısından dairelerin en az iki noktada teğetlik kısıtı kullanılmıştır. Yerleşim sırasında dairelerle ilgili ağırlık, yüke dayanım, yerleşim sırası veya önceliği, kırılabilirlik, yük dengesi gibi ek kısıtlar göz ardı edilmiştir. Dairelerin ağırlıklarına göre yerleştirilmesi paletin dengesinin sağlanmasında; yerleşim sırası veya önceliğine göre yerleştirilmesi ise nakliyedeki teslim sırasına göre paletin içi kısmına veya dış kısmına yerleştirilmesi açısından önemli bir kısıt olabilir. Hatta palet içinde yan yana

konumlandıklarında birbirlerini kötü etkileyecek ürünlerin birbirlerinden belirli bir mesafede olmalarını sağlayacak komşuluk yasağı kısıtı da eklenerek DPP için farklı çözüm modeli geliştirilebilir. Ayrıca belirtilen bu kısıtlar üç boyutlu DPP için çözüm modeli geliştirildiğinde ise farklı alanlarda bir DPP olabilmektedir.



KAYNAKLAR

1. Hashim, N., A., Zulkipli, F., Januri, S., S., Shariff, S., Sarifah R. (2013). *An Alternative Heuristics for Bin Packing Problem*. International Conference on Industrial Engineering and Operations Management.
2. Romanova, T., Pankratov, A., Litvinchev, I., Pankratova, Y. and Urniaieva, I. (2019). Optimized Packing Clusters of Objects in a Rectangular Container. *Mathematical Problems in Engineering*, 2019, Article 4136430.
3. Hoare, N.P. and Beasley, J.E. (2001). Placing Boxes on Shelves: A Case Study. *The Journal of the Operational Research Society*, 52(6), 605-614.
4. Szabó, P., Markot, M., Csendes, T., Specht, E., Casado, L. G. and García, I. (2007). *New Approaches to Circle Packing in a Square: With Program Codes*. Springer, pp. 1-8.
5. He, K., Huang, M. and Yang, C. (2015). An action-space-based global optimization algorithm for packing circles into a square container. *Computers & Operations Research*, 58, 67-74.
6. Liu, J., Xue, S., Liu, Z. and Xu, D. (2009). An improved energy landscape paving algorithm for the problem of packing circles into a larger containing circle. *Computers & Industrial Engineering*, 57(3), 1144-1149.
7. Saghir, M. and Jönson, G. (2001). Packaging Handling Evaluation Methods in the Grocery Retail Industry. *Packaging Technology and Science*, 14, 21-29.
8. Onursal, F.S. (2014). *Ambalaj Sektöründe Stok Çeşitliliği Optimizasyonu* (Birinci Baskı). İstanbul: Nobel Akademik Yayıncılık.
9. Hinxman, A.I. (1980). The trim-loss and assortment problems: A survey. *European Journal of Operational Research*, 5(1), 8-18.
10. Dyckhoff, H. (1990). A typology of cutting and packing problems. *European Journal of Operational Research*, 44(2), 145-159.
11. Wäscher, G., Haußner, H. and Schumann, H. (2007). An improved typology of cutting and packing problems. *European Journal of Operational Research*, 183(3), 1109-1130.
12. Silva, E., Oliveira, J.F. and Wäscher, G. (2016). The pallet loading problem: a review of solution methods and computational experiments. *Int. Trans. Oper. Res.*, 23, 147-172.
13. Hodgson, T.J. (1982). A Combined Approach to the Pallet Loading Problem. *AIIE Transactions*, 14(3), 175-182.
14. Martello, S., Pisinger, D. and Vigo, D. (1998). The Three-Dimensional Bin Packing Problem. *Operations Research*, 48, 256-267.

15. Ram, B. (1992). The pallet loading problem: A survey. *International Journal of Production Economics*, 28(2), 217-225.
16. Kellegöz T., Özcan.U., İşleyen S.K., Hakyemez S., Doğan A., Söylemez İ., Şahin M. (2016, 19-22 May 2016). *Pallet loading problem Literature reviewand research topics*. International Conference on Research in Educationand Science, Muğla,Türkiye.
17. Steudel, H.J. (1979). Generating Pallet Loading Patterns: A Special Case of the Two-Dimensional Cutting Stock Problem. *Management Science*, 25, 997–1004.
18. Smith, A., De Cani, P. (1980). An algorithm to optimize the layout of boxes in pallets. *Journal of the Operational Research Society*, 31, 573–578.
19. Bischoff, E., Dowsland, W.B. (1982). An application of the micro to product design. *Journal of the Operational Research Society*, 33, 271–280.
20. Steudel, H. (1984). Generating pallet loading patterns with considerations of item stacking on end and side surfaces. *Journal of Manufacturing Systems*, 3, 135–143.
21. Beasley, J.E. (1985). An exact two-dimensional non guillotine cutting tree search procedure. *Operations Research*, 33, 49–64.
22. Carpenter, H., Dowsland, WB. (1985). Practical considerations of the pallet-loading problem. *Journal of the Operational Research Society*, 36, 489–497.
23. Dowsland, K. (1987). An exact algorithm for the pallet loading problem. *European Journal of Operational Research*, 31, 78–84.
24. Dowsland, K.A. (1993). Some experiments with simulated annealing techniques for packing problems. *European Journal of Operational Research*, 68, 389–399.
25. Arenales, M., Morabito, R. (1995). An AND/OR-graph approach to the solution of two-dimensional non-guillotine cutting problems. *European Journal of Operational Research*, 84, 599–617.
26. Scheithauer, G., Terno, J. (1996). The G4-heuristic for the pallet loading problem. *Journal of the Operational Research Society*, 47, 511–522.
27. Herbert, E.A., Dowsland, K.A. (1996). A family of genetic algorithms for the pallet loading problem. *Annals of Operations Research*, 63, 415–436.
28. Liu, F.-H., Hsiao, C-J. (1997). A three-dimensional pallet loading method for single-size boxes. *Journal of the Operational Research Society*, 48, 726–735.
29. Bhattacharya, S., Roy, R. (1998). An exact depth-first algorithm for the pallet loading problem. *European Journal of Operational Research*, 110, 610–625.
30. Morabito, R., Morales, S. (1998). Simple and effective recursive procedure for the manufacturer's pallet loading problem. *Journal of the Operational Research Society*, 49, 819–828.

31. Scheithauer, G., Sommerweiß, U. (1998). 4-Block heuristic for the rectangle packing problem. *European Journal of Operational Research*, 108, 509–526.
32. Young-Gun, G., Maing-Kyu, K. (2001). A fast algorithm for two-dimensional pallet loading problems of large size. *European Journal of Operational Research*, 134, 193–202.
33. Amaral, A., Wright, M. (2001). Experiments with a strategic oscillation algorithm for the pallet loading problem. *International Journal of Production Research*, 39, 2341–2351.
34. Lins, L., Lins, S., Morabito, R. (2003). An L-approach for packing (l, w)-rectangles into rectangular and L-shaped pieces. *Journal of the Operational Research Society*, 54, 777–789.
35. Alvarez-Valdes, R., Parreño, F., Tamarit, JM. (2005). A branch-and-cut algorithm for the pallet loading problem. *Computers and Operations Research*, 32, 3007–3029.
36. Alvarez-Valdes, R., Parreño, F., Tamarit, JM. (2005). A tabu search algorithm for the pallet loading problem. *OR Spectrum*, 27, 43–61.
37. Birgin, E.G., Morabito, R., Nishihara, F. H. (2005). A note on an L-approach for solving the manufacturer's pallet loading problem. *Journal of the Operational Research Society*, 56, 1448–1451.
38. Pureza, V., Morabito, R. (2006). Some experiments with a simple tabu search algorithm for the manufacturer's pallet loading problem. *Computers and Operations Research*, 33, 804–819.
39. Yi, J., Chen, X., Zhou, J. (2009). The pinwheel pattern and its application to the manufacturer's pallet-loading problem. *International Transactions in Operational Research*, 16, 809–828.
40. Birgin, E.G., Lobato, R., Morabito, R. (2010). An effective recursive partitioning approach for the packing of identical rectangles in a rectangle. *Journal of the Operational Research Society*, 61, 306–320.
41. Kocjan, W., Holmström, K. (2010). Computing stable loads for pallets. *European Journal of Operational Research*, 207 (2), 980-985.
42. de Queiroz, T.A., Miyazawa, F.K., Wakabayashi, Y. (2014). On the L-approach for generating unconstrained two dimensional non-guillotine cutting patterns. *4OR*, 13 (2), 199-219.
43. Lu, Y., Cha, J. (2014). A fast algorithm for identifying minimum size instances of the equivalence classes of the pallet loading problem. *European Journal of Operational Research*, 237 (3), 794-801.
44. Ahn, S., Yoon, K., Park, J. (2015). A best-first branch and bound algorithm for the pallet-loading problem. *International Journal of Production Research*, 53 (3), 835-849.

45. Ahn, S., Park, C., Yoon, K. (2015). An improved best-first branch and bound algorithm for the pallet-loading problem using a staircase structure. *Expert Systems with Applications*, 42 (21), 7676-7683.
46. Chen, C., Sarin, S., Ram, B. (1991). The pallet packing problem for non-uniform box sizes. *International Journal of Production Research*, 29 (10), 1963-1968.
47. Tsai, R., Malstrom, EL., Kuo, W. (1993). Three dimensional palletization of mixed box sizes. *IIE Transactions*, 25, 64-75.
48. Abdou, G., Yang, M. (1994). A systematic approach for the three-dimensional palletization problem. *International Journal of Production Research*, 32, 2381–2394.
49. Abdou, G., Yang, M. (1995). Multi-layer palletisation of multi-size boxes. *International Journal of Advanced Manufacturing Technology*. *International Journal of Advanced Manufacturing Technology*, 10(4), 292–297.
50. Bischoff, E.E., Ratcliff, M. S. W. (1995). Loading multiple pallets. *Journal of the Operational Research Society*, 46, 1322–1336.
51. Arghavani, J., Abdou, G. (1996). 3D Volumetric pallet-loading optimisation. *International Journal of Advanced Manufacturing Technology*, 11, 425-429.
52. Abdou, G., Arghavani, J. (1997). Interactive ILP procedures for stacking optimization for the 3D palletization problem. *International Journal of Production Research*, 35, 1287–1304.
53. Abdou, G., Elmasry, M. (1999). 3D random stacking of weakly heterogeneous palletization problems. *International Journal of Production Research*, 37, 1505–1524.
54. Furuholmen, M., Glette, K., Hovin, M., Torresen, J. (2009). *Coevolving heuristics for the distributor's pallet packing problem*. IEEE Congress on Evolutionary Computation (CEC 2009).
55. Ballew, P. (2000). *The distributor's three-dimensional pallet-packing problem: A mathematical formulation and heuristic solution approach*, MS Thesis, Air Force Institute of Technology Graduate School of Engineering, Ohio.
56. Baltacioğlu, E., Moore, JT., Hill Jr., RR. (2006). The distributor's three-dimensional pallet-packing problem: A human intelligence-based heuristic approach. *International Journal of Operational Research*, 1 (3), 249-266.
57. Schuster, M., Bormann, R., Steidl, D., Haertle, SR., Stilman, M. (2010). *Stable stacking for the distributor's pallet packing problem*. IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan.
58. Al-Shayea, A. (2011). Solving the Three-dimensional pallet-packing problem using mixed 0 - 1 model. *Journal of Service Science and Management*, 4, 513-522.

59. Zúñiga, C., Piera, MA, Narciso, M. (2011). Revisiting the pallet loading problem using a discrete event system approach to minimise logistic costs. *International Journal of Production Research*, 49 (8), 2243–2264.
60. Birgin, E., Lobato, RD., Morabito, R. (2012). Generating unconstrained two-dimensional non-guillotine cutting patterns by a recursive partitioning algorithm. *Journal of the Operational Research Society*, 63, 183–200.
61. Taylor, G.S., Chan, Y., Rasool, G. (2015). A three-dimensional bin-packing model: exact multicriteria solution and computational complexity. *Annals of Operation Research*, 251, 397-427.
62. Chen, M., Tang, X., Song, T., Zeng, Z., Peng, X. and Liu, S. (2018). Greedy heuristic algorithm for packing equal circles into a circular container. *Computers & Industrial Engineering*, 119, 114-120.
63. López, C.O. and Beasley, J.E. (2013). Packing unequal circles using formulation space search. *Computers & Operations Research*, 40(5), 1276-1288.
64. Stephenson, K. (2005). *Introduction to circle packing: The theory of discrete analytic functions* Cambridge University Press,
65. Castillo, I., Kampas, F.J. and Pintér, J.D. (2008). Solving circle packing problems by global optimization: Numerical results and industrial applications. *European Journal of Operational Research*, 191(3), 786-802.
66. Hifi, M. and M'Hallah, R. (2009). A Literature Review on Circle and Sphere Packing Problems: Models and Methodologies. *Advances in Operations Research*, 2009, 1-22.
67. Malfatti, G. (1803). Memoria sopra un problema stereotomico. *Memorie di Matematica e di Fisica della Società Italiana delle Scienze*, 10, 235-244.
68. Bolyai, F., Bolyai, J.n., Evans, H.M. and Library, B. (1832-1833). *Tentamen iuventutem studiosam in elementa matheseos purae elementaris ac sublimioris methodo intuitiva evidentiaque huic propria introducendi : cum appendice triplici.*
69. Internet: Center pivot irrigation. URL: https://en.wikipedia.org/wiki/Center_pivot_irrigation, Son Erişim Tarihi: 21.09.2021.
70. Akeb, H., Hifi, M. and M'Hallah, R. (2009). A beam search algorithm for the circular packing problem. *Computers & Operations Research*, 36(5), 1513-1528.
71. Soto, L. and Orquidea, C. (2013). *Formulation space search for two-dimensional packing problems.*
72. Wang, H., Huang, W., Zhang, Q. and Xu, D. (2002). An improved algorithm for the packing of unequal circles within a larger containing circle. *European Journal of Operational Research*, 141(2), 440-453.
73. Huang, W., li, y., Jurkowiak, B., Li, C.-M. and Xu, R. (2003). *A Two-Level Search Strategy for Packing Unequal Circles into a Circle Container,*

74. Huang, W. and Yan, K. (2004). A Short Note on a Simple Search Heuristic for the Diskpacking Problem. *Annals OR*, 131, 101-108.
75. Zhang, D. and Huang, W. (2004). *A Simulated Annealing Algorithm for the Circles Packing Problem*. International Conference on Computational Science 2004, pp. 206-214
76. Zhang, D.-f. and Deng, A.-s. (2005). An effective hybrid algorithm for the problem of packing circles into a larger containing circle. *Computers & Operations Research*, 32(8), 1941-1951.
77. Carrabs, F., Cerrone, C. and Cerulli, R. (2014). *A Tabu Search Approach for the Circle Packing Problem*. 17th International Conference on Network-Based Information Systems, pp. 165-171.
78. Specht, E. (2015). A precise algorithm to detect voids in polydisperse circle packings. *Proceedings: Mathematical, Physical and Engineering Sciences*, 471(2182), 1–19.
79. López, C.O. and Beasley, J.E. (2016). A formulation space search heuristic for packing unequal circles in a fixed size circular container. *European Journal of Operational Research*, 251(1), 64-73.
80. Orick, G.L., Stephenson, K. and Collins, C. (2017). A linearized circle packing algorithm. *Computational Geometry*, 64, 13-29.
81. Ying-cong WANG, L.Z. (2019). Swarm intelligence labor division algorithm for solving unequal circle packing problem. *Journal of ZheJiang University (Engineering Science)*, 53(11), 2129-2138.
82. Stoyan, Y. and Yaskov, G. (2013). Packing unequal circles into a strip of minimal length with a jump algorithm. *Optimization Letters*, 8(3), 949-970.
83. Hifi, M. and M'Hallah, R. (2004). Approximate algorithms for constrained circular cutting problems. *Computers & Operations Research*, 31(5), 675-694.
84. Hifi, M., Paschos, V.T. and Zissimopoulos, V. (2004). A simulated annealing approach for the circular cutting problem. *European Journal of Operational Research*, 159(2), 430-448.
85. Stoyan, Y.G. and Yaskov, G. (2004). A mathematical model and a solution method for the problem of placing various-sized circles into a strip. *European Journal of Operational Research*, 156(3), 590-600.
86. Zhang, D., Liu, Y. and Chen, S. (2004). *Packing Different-sized Circles into a Rectangular Container Using Simulated Annealing Algorithm*. 2004 International Conference on Computational Intelligence, pp. 388-391.
87. Akeb, H., Hifi, M. and Negre, S. (2011). An augmented beam search-based algorithm for the circular open dimension problem. *Computers & Industrial Engineering*, 61(2), 373-381.

88. He, Y. and Wu, Y. (2012). Packing non-identical circles within a rectangle with open length. *Journal of Global Optimization*, 56(3), 1187-1215.
89. Torres-Escobar, R., Marmolejo-Saucedo, J.A. and Litvinchev, I. (2020). Binary monkey algorithm for approximate packing non-congruent circles in a rectangular container. *Wireless Networks*, 26(7), 4743-4752.
90. Lubachevsky, B.D. and Graham, R.L. (1997). Curved Hexagonal Packings of Equal Disks in a Circle. *Discrete & Computational Geometry*, 18(2), 179-194.
91. Graham, R.L., Lubachevsky, B.D., Nurmela, K.J. and Östergård, P.R.J. (1998). Dense packings of congruent circles in a circle. *Discrete Mathematics*, 181(1), 139-154.
92. Birgin, E., G. and Gentil, J., M. (2010). New and improved results for packing identical unitary radius circles within triangles, rectangles and strips. *Computers & Operations Research*, 37(7), 1318-1327.
93. He, K., Ye, H., Wang, Z. and Liu, J. (2018). An efficient quasi-physical quasi-human algorithm for packing equal circles in a circular container. *Computers & Operations Research*, 92, 26-36.
94. He, K., Zhang, M., Zhou, J., Jin, Y. and Li, C. (2020). Stochastic Item Descent Method for Large Scale Equal Circle Packing Problem. *ArXiv*, abs/2001.08540.
95. Goldberg, M. (1970). The Packing of Equal Circles in a Square. *Taylor & Francis, Ltd. on behalf of the Mathematical Association of America*, 43, 24-30.
96. Mollard, M. and Payan, C. (1990). Some progress in the packing of equal circles in a square. *Discrete Mathematics*, 84(3), 303-307.
97. Theodoracatos, V.E. and Grimsley, J.L. (1995). The optimal packing of arbitrarily-shaped polygons using simulated annealing and polynomial-time cooling schedules. *Computer Methods in Applied Mechanics and Engineering*, 125(1), 53-70.
98. Nurmela, K.J. and Östergård, P.R.J. (1997). Packing up to 50 Equal Circles in a Square. *Discrete & Computational Geometry*, 18(1), 111-120.
99. Boll, D.W., Donovan, J., Graham, R. and Lubachevsky, B. (2000). Improving Dense Packings of Equal Disks in a Square. *The Electronic Journal of Combinatorics*, 7(1).
100. Locatelli, M. and Raber, U. (2002). Packing equal circles in a square: a deterministic global optimization approach. *Discrete Applied Mathematics*, 122(1), 139-166.
101. Markót, M.C. and Csendes, T. (2005). A New Verified Optimization Technique for the Packing Circles in a Unit Square Problems. *SIAM Journal on Optimization*, 16(1), 193-219.
102. Costa, A., Hansen, P. and Liberti, L. (2013). On the impact of symmetry-breaking constraints on spatial Branch-and-Bound for circle packing in a square. *Discrete Applied Mathematics*, 161(1-2), 96-106.

103. Dowsland, K.A. (1991). Optimising the palletisation of cylinders in cases. *Operations-Research-Spektrum*, 13(4), 204-212.
104. Correia, M., Oliveira, J. and Soeiro Ferreira, J. (2000). Cylinder packing by simulated annealing. *Pesquisa Operacional*, 20.
105. Birgin, E.G., Martínez, J.M. and Ronconi, D.P. (2005). Optimizing the packing of cylinders into a rectangular container: A nonlinear approach. *European Journal of Operational Research*, 160(1), 19-33.
106. Michalewicz, Z. (1996). Heuristic methods for evolutionary computation techniques. *Journal of Heuristics*, 1(2), 177-206.
107. Michalewicz, Z. (1996). *Genetic algorithm data structures evolution programs*, Springer Science & Business Media, New York.
108. Albayrak, M. (2008). *Baskı devre kartı delik delme makineleri için genetik algoritmalar yardımı ile güzergâh belirleme*. Doktora tezi, Selçuk Üniversitesi Fen Bilimleri Enstitüsü, Konya.
109. Emel, G.G. and Taşkın, Ç. (2002). Genetik Algoritmalar ve Uygulama Alanları. *Uludağ Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, 21(1), 129-152.
110. Mitchell, M. and Taylor, C.E. (1999). Evolutionary Computation: An Overview. *Annual Review of Ecology and Systematics*, 30(1), 593-616.
111. Sivanandam, S.N. and Deepa, S.N. (2008). *Introduction to Genetic Algorithms* Springer Publishing Company, Springer, Germany.
112. Maza, M.D. and Tidor, B. (1992). Increased Flexibility In Genetic Algorithms: The Use Of Variable Boltzmann Selective Pressure To Control Propagation. *Computer Science and Operations Research*, 425-440.
113. Banzhaf, W., Nordin, P., Keller, R. and Francone, F. (1998). *Genetic Programming: An Introduction on the Automatic Evolution of computer programs and its Applications*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
114. Gen, M. and Cheng, R. (2000). *Genetic algorithms and engineering optimization*. New York: John Wiley and Sons.
115. Syswerda, G. (1991). *Schedule Optimization Using Genetic Algorithms, Handbook of Genetic Algorithms*. Van Nostrand Reinhold, New York.
116. Oliver, I.M., Smith, D.J. and Holland, J.R.C. (1987). *A study of permutation crossover operators on the traveling salesman problem*. Proceedings of the Second International Conference on Genetic Algorithms on Genetic algorithms and their application, Cambridge, Massachusetts, USA.
117. Gen, M. and Cheng, R. (1997). *Genetic Algorithms and Engineering Design*. New York: John Wiley and Sons.

118. Falkenauer, E. and Bouffouix, S. (1991). *A genetic algorithm for job shop*. International Conference on Robotics and Automation, 1, 824-829.
119. Goldberg, D.E. and Lingle, R. (1985). *Alleles, Loci and the Traveling Salesman Problem*. Proceedings of the 1st International Conference on Genetic Algorithms, 154, 154-159.
120. Krishnakumar, K. (1990). Micro-Genetic Algorithms For Stationary And Non-Stationary Function Optimization. *SPIE Intellegent Control and Adaptive System*, 1196, 289-296.
121. Karaboğa, D. (2011). *Yapay Zeka Optimizasyon Algoritmaları*. Nobel Yayın Dağıtım.
122. Gökçen, H. (2011). *Yönetim bilgi sistemleri*. Palme Yayıncılık.
123. McKenney, J.L. and Scott, M.M. (1971). *Management decision systems : computer-based support for decision making*. Harvard Business School Press.
124. Bensghir, T. (1996). *Bilgi Teknolojileri ve Örgütsel Değişim*. Türkiye ve Orta Doğu Amme İdaresi Enstitüsü Yayını.
125. Scott, G.M. (1986). *Principles of Management Information Systems*. McGraw-Hill.
126. Power, D. (1997). What is a DSS. *The On-Line Executive Journal for Data-Intensive Decision Support*, 1(3), 223-232.
127. Keen, P.G.W. and Scott Morton, M.S. (1978). *Decision support systems: An organizational perspective*. Addison-Wesley Pub. Co.
128. Sprague, R.H., Jr. and Carlson, E.D. (1982). *Building effective decision support systems*. Englewood Cliffs (N.J.) : Prentice-Hall.
129. Alter, S. (1999). *Information systems: a management perspective*. 2nd ed., Menlo Park, CA: The Benjamin/Cummings.
130. Turban, E. (1995). *Decision support and expert systems Management support systems*. Prentice-Hall.
131. Sauter, V. (1997). *Decision support systems: an applied managerial approach*. John Wiley & Sons.
132. Ülgen, H. (1980). *İşletme yönetiminde bilgisayarlar*. İstanbul Üniversitesi.
133. Marakas, G. (1999). *Decision Support Systems in the 21st Century*. Prentice Hall.
134. Holtzman, S. (1989). *Intelligent decision systems*. Reading, Mass.: Addison-Wesley,
135. Turban, E. (1990). *Decision Support and Expert Systems: Management Support Systems*. New York: Prentice Hall PTR,

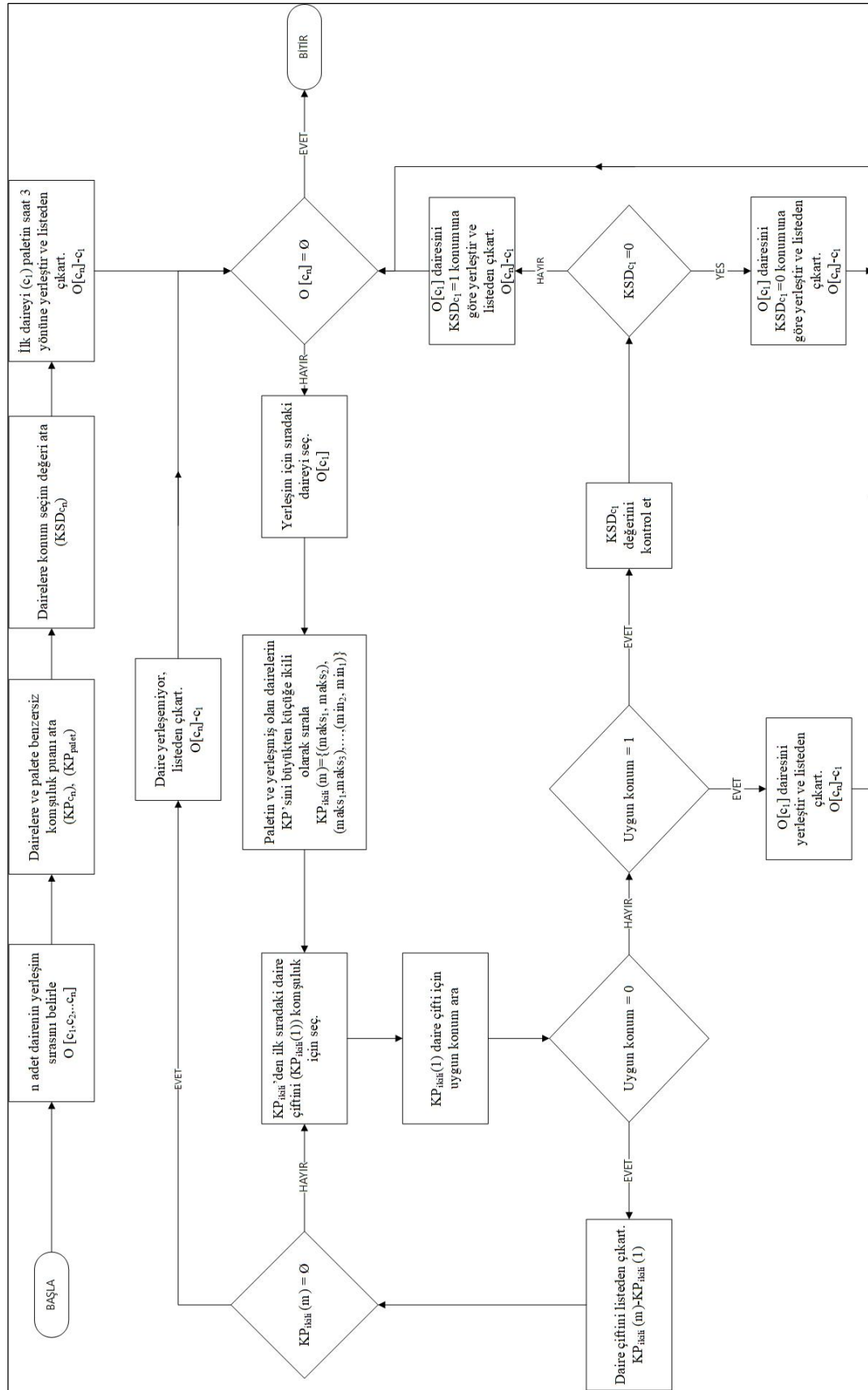
136. Türkiye Bilişim Derneği (TBD). (2010). Kamuda Karar Destek Sistemlerinin Kullanımı ve Bir Model Önerisi, *Türkiye Bilişim Derneği: II. Çalışma Grubu Raporu*.
137. Angehrn, A.A.and Jelassi, T. (1994). DSS research and practice in perspective. *Decis. Support Syst.*, 12, 267-275.
138. Internet: math.stackexchange. URL: <https://math.stackexchange.com/questions/256100/how-can-i-find-the-points-at-which-two-circles-intersect>, Son Erişim Tarihi: 22.04.2021
139. Bolat, B., Erol, K.and İmrak, C. (2004). Mühendislik Uygulamalarında Genetik Algoritmalar ve Operatörlerin İşlevleri. *Journal of Engineering and Natural Sciences*, 22(4), 264-271.
140. Ghedjati, F. (1999). Genetic algorithms for the job-shop scheduling problem with unrelated parallel constraints: Heuristic mixing method machines and precedence. *Computers & Industrial Engineering*, 37(1), 39-42.
141. Grefenstette, J.J. (1986). Optimization of Control Parameters for Genetic Algorithms. *IEEE Transactions on Systems, Man, and Cybernetics*, 16(1), 122-128.
142. Mitchell, M. (1998). *An introduction to genetic algorithms*. MIT press.
143. Pham, D.T.and Karaboga, D. (1997). Genetic algorithms with variable mutation rates: Application to fuzzy logic controller design. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 211(2), 157-167.
144. Görener, A. (2012). *Depolama Sistemlerinde Sipariş Toplama İşlemlerinin Optimizasyonu*, Doktora Tezi, İstanbul Üniversitesi Sosyal Bilimler Enstitüsü, İstanbul.
145. Internet: Packomania. URL: www.packomania.com, Son Erişim Tarihi: 08/09/2021.
146. Internet: Cpubenchmark. URL: www.cpubenchmark.net, Son Erişim Tarihi: 08/10/2021.



EKLER



EK-1. Yerleşim algoritması akış şeması





EK-2. Alan maksimizasyonuna göre turnuva seçiminde büyüklüğü belirleme testi sonucu

Paired T-Test and CI: k=2 Alan; k=3 Alan

Paired T for k=2 Alan - k=3 Alan

	N	Mean	StDev	SE Mean
k=2 Alan	36	333,3	262,0	43,7
k=3 Alan	36	332,3	261,5	43,6
Difference	36	0,966	4,314	0,719

95% CI for mean difference: (-0,494; 2,425)

T-Test of mean difference = 0 (vs $\neq$ 0): T-Value = 1,34 P-Value = 0,188

P-değeri 0.05'den büyük. H_0 hipotezi kabul edilir. Her iki ana kütlenin alanları arasında istatistiksel olarak anlamlı bir farklılık yoktur.

Paired T-Test and CI: k=2 Alan_CPU; k=3 Alan_CPU

Paired T for k=2 Alan_CPU - k=3 Alan_CPU

	N	Mean	StDev	SE Mean
k=2 Alan_CPU	36	4,063	5,375	0,896
k=3 Alan_CPU	36	3,885	5,221	0,870
Difference	36	0,179	1,601	0,267

95% CI for mean difference: (-0,363; 0,720)

T-Test of mean difference = 0 (vs $\neq$ 0): T-Value = 0,67 P-Value = 0,507

P-değeri 0.05'den büyüktür. H_0 hipotezi kabul edilir. Her iki ana kütlenin CPU zamanları arasında istatistiksel olarak anlamlı bir farklılık yoktur.



EK-3. Daire sayısı maksimizasyonuna göre turnuva seçiminde büyüklüğü belirleme testi sonucu

Paired T-Test and CI: k=2 Daire; k=3 Daire

Paired T for k=2 Daire - k=3 Daire

	N	Mean	StDev	SE Mean
k=2 Daire	36	16,09	10,40	1,73
k=3 Daire	36	16,23	10,55	1,76
Difference	36	-0,1355	0,4614	0,0769

95% CI for mean difference: (-0,2916; 0,0206)

T-Test of mean difference = 0 (vs $\neq$ 0): T-Value = -1,76 P-Value = 0,087

P-değeri 0.05'den büyüktür. H0 hipotezi kabul edilir. Her iki ana kütlenin daire sayıları arasında istatistiksel olarak anlamlı bir farklılık yoktur.

Paired T-Test and CI: k=2 Daire_CPU; k=3 Daire_CPU

Paired T for k=2 Daire_CPU - k=3 Daire_CPU

	N	Mean	StDev	SE Mean
k=2 Daire_CPU	36	4,80	6,54	1,09
k=3 Daire_CPU	36	4,74	6,44	1,07
Difference	36	0,057	1,280	0,213

95% CI for mean difference: (-0,376; 0,490)

T-Test of mean difference = 0 (vs $\neq$ 0): T-Value = 0,27 P-Value = 0,792

P-değeri 0.05'den büyüktür. H0 hipotezi kabul edilir. Her iki ana kütlenin CPU zamanları arasında istatistiksel olarak anlamlı bir farklılık yoktur.



EK-4. Alan maksimizasyonuna göre aprazlama ynteminde aprazlama noktası belirleme testi sonucu

Paired T-Test and CI: CY1_Alan; CY2_Alan

Paired T for TK_Alan - IK_Alan

	N	Mean	StDev	SE Mean
CY1_Alan	36	332,5	261,5	43,6
CY2_Alan	36	333,0	262,0	43,7
Difference	36	-0,497	5,255	0,876

95% CI for mean difference: (-2,275; 1,281)

T-Test of mean difference = 0 (vs $\neq$ 0): T-Value = -0,57 P-Value = 0,574

Her iki aprazlama ynteminin ortalamalarının birbirinden farklı olduėuna dair istatistiksel olarak %95 gven dzeyinde yeterli kanıt bulunmamaktadır. Bununla birlikte CY2 ynteminin ortalaması yksektir.

Paired T-Test and CI: CY1_Alan_CPU; CY2_Alan_CPU

Paired T for TK_Alan_CPU - IK_Alan_CPU

	N	Mean	StDev	SE Mean
CY1_Alan_CPU	36	3,986	5,300	0,883
CY2_Alan_CPU	36	3,962	5,298	0,883
Difference	36	0,024	1,486	0,248

95% CI for mean difference: (-0,479; 0,526)

T-Test of mean difference = 0 (vs $\neq$ 0): T-Value = 0,10 P-Value = 0,924

%95 gven dzeyinde her iki yntemin ortalama CPU deėerlerinde istatistiksel olarak anlamlı bir farklılık yoktur. Bununla birlikte CY2 ynteminin ortalama CPU deėeri dşktr.



EK-5. Daire sayısı maksimizasyonuna göre aprazlama ynteminde aprazlama noktası belirleme testi sonucu

Paired T-Test and CI: CY1_Daire; CY2_Daire

Paired T for TK_Daire - IK_Daire

	N	Mean	StDev	SE Mean
CY1_Daire	36	16,12	10,44	1,74
CY2_Daire	36	16,20	10,51	1,75
Difference	36	-0,0769	0,4143	0,0690

95% CI for mean difference: (-0,2170; 0,0633)

T-Test of mean difference = 0 (vs $\neq$ 0): T-Value = -1,11 P-Value = 0,273

Her iki aprazlama ynteminin ortalamalarının birbirinden farklı olduėuna dair istatistiksel olarak %95 gven dzeyinde yeterli kanıt bulunmamaktadır. Bununla birlikte CY2 ynteminin ortalaması yksektir.

Paired T-Test and CI: CY1_Daire_CPU; CY2_Daire_CPU

Paired T for TK_Daire_CPU - IK_Daire_CPU

	N	Mean	StDev	SE Mean
CY1_Daire_CPU	36	4,93	6,66	1,11
CY2_Daire_CPU	36	4,62	6,32	1,05
Difference	36	0,308	1,631	0,272

95% CI for mean difference: (-0,244; 0,860)

T-Test of mean difference = 0 (vs $\neq$ 0): T-Value = 1,13 P-Value = 0,265

%95 gven dzeyinde istatistiksel olarak her iki aprazlama ynteminin hem daire sayısı performans lt ve hem de CPU time performans lt aısından bir farkı bulunmamaktadır. Her iki hipotez testinin ortalamalar birbirine eřittir řeklinde kurulan H0 hipotezi reddedilemez. Bununla birlikte, CY2 ynteminin ortalama CPU deėeri dřktr.

P-deėerleri daire sayısı performans lt iin 0.273 ve CPU time performans lt iin 0.265 olarak hesaplanmıřtır.



EK-6. Daire sayısı maksimizasyonunda uygunluk değerine göre istatistiksel analiz sonuçları

Between-Subjects Factors

	N
Pc	486
	486
	486
Pm	486
	486
	486
Populasyon	486
	486
	486
Durdurma	486
	486
	486

Tests of Between-Subjects Effects

Dependent Variable: FF_Daire

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Corrected Model	18,365 ^a	80	,230	,002	1,000
Intercept	385070,420	1	385070,420	3332,971	,000
Pc	5,163	2	2,582	,022	,978
Pm	,094	2	,047	,000	1,000
Populasyon	,590	2	,295	,003	,997
Durdurma	7,402	2	3,701	,032	,968
Pc * Pm	,305	4	,076	,001	1,000
Pc * Populasyon	,549	4	,137	,001	1,000
Pc * Durdurma	1,123	4	,281	,002	1,000
Pm * Populasyon	,150	4	,038	,000	1,000
Pm * Durdurma	,166	4	,042	,000	1,000
Populasyon * Durdurma	,012	4	,003	,000	1,000
Pc * Pm * Populasyon	,372	8	,047	,000	1,000
Pc * Pm * Durdurma	,288	8	,036	,000	1,000
Pc * Populasyon * Durdurma	,241	8	,030	,000	1,000
Pm * Populasyon * Durdurma	,691	8	,086	,001	1,000
Pc * Pm * Populasyon * Durdurma	1,218	16	,076	,001	1,000
Error	159089,876	1377	115,534		
Total	544178,661	1458			
Corrected Total	159108,241	1457			

a. R Squared = ,000 (Adjusted R Squared = -,058)



EK-6. (devamı) Daire sayısı maksimizasyonunda uygunluk değerine göre istatistiksel analiz sonuçları

FF_Daire

Duncan^{a,b}

Pc	N	Subset
		1
,70	486	16,1773
,80	486	16,2540
,90	486	16,3230
Sig.		,844

Means for groups in homogeneous subsets are displayed.

Based on observed means.
The error term is Mean Square(Error) = 115,534.

a. Uses Harmonic Mean
Sample Size = 486,000.

b. Alpha = 0,05.

FF_Daire

Duncan^{a,b}

Pm	N	Subset
		1
,20	486	16,2432
,10	486	16,2487
,05	486	16,2623
Sig.		,979

Means for groups in homogeneous subsets are displayed.

Based on observed means.
The error term is Mean Square(Error) = 115,534.

a. Uses Harmonic Mean
Sample Size = 486,000.

b. Alpha = 0,05.

FF_Daire

Duncan^{a,b}

Populasyon	N	Subset
		1
3n	486	16,2268
4n	486	16,2514
6n	486	16,2761
Sig.		,947

Means for groups in homogeneous subsets are displayed.

Based on observed means.
The error term is Mean Square (Error) = 115,534.

a. Uses Harmonic Mean Sample
Size = 486,000.

b. Alpha = 0,05.

FF_Daire

Duncan^{a,b}

Durdurma	N	Subset
		1
25n	486	16,1593
50n	486	16,2621
100n	486	16,3328
Sig.		,815

Means for groups in homogeneous subsets are displayed.

Based on observed means.
The error term is Mean Square (Error) = 115,534.

a. Uses Harmonic Mean Sample
Size = 486,000.

b. Alpha = 0,05.



EK-7. Daire sayısı maksimizasyonu probleminde CPU zamanı göre istatistiksel analiz sonuçları

Between-Subjects Factors

	N
Pc	486
	486
	486
Pm	486
	486
	486
Populasyon	486
	486
	486
Durdurma	486
	486
	486

Tests of Between-Subjects Effects

Dependent Variable: CPU_Daire

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Corrected Model	716684,193 ^a	80	8958,552	2,370	,000
Intercept	1447711,041	1	1447711,041	382,991	,000
Pc	240274,867	2	120137,434	31,782	,000
Pm	3988,135	2	1994,067	,528	,590
Populasyon	69700,146	2	34850,073	9,220	,000
Durdurma	284319,751	2	142159,876	37,608	,000
Pc * Pm	1930,291	4	482,573	,128	,972
Pc * Populasyon	8084,633	4	2021,158	,535	,710
Pc * Durdurma	89961,474	4	22490,369	5,950	,000
Pm * Populasyon	176,691	4	44,173	,012	1,000
Pm * Durdurma	1244,834	4	311,208	,082	,988
Populasyon * Durdurma	10985,282	4	2746,321	,727	,574
Pc * Pm * Populasyon	690,587	8	86,323	,023	1,000
Pc * Pm * Durdurma	1062,429	8	132,804	,035	1,000
Pc * Populasyon * Durdurma	2280,203	8	285,025	,075	1,000
Pm * Populasyon * Durdurma	669,725	8	83,716	,022	1,000
Pc * Pm * Populasyon * Durdurma	1315,143	16	82,196	,022	1,000
Error	5205076,697	1377	3780,012		
Total	7369471,931	1458			
Corrected Total	5921760,890	1457			

a. R Squared = ,121 (Adjusted R Squared = ,070)



EK-7. (devamı) Daire sayısı maksimizasyonu probleminde CPU zamanı göre istatistiksel analiz sonuçları

CPU_Daire

Duncan^{a,b}

Pc	N	Subset		
		1	2	3
,70	486	18,7495		
,80	486		26,7090	
,90	486			49,0745
Sig.		1,000	1,000	1,000

Means for groups in homogeneous subsets are displayed.

Based on observed means.

The error term is Mean Square(Error) = 3780,012.

a. Uses Harmonic Mean Sample Size = 486,000.

b. Alpha = 0,05.

CPU_Daire

Duncan^{a,b}

Pm	N	Subset
		1
,05	486	30,2510
,20	486	30,4345
,10	486	33,8475
Sig.		,394

Means for groups in homogeneous subsets are displayed.

Based on observed means.

The error term is Mean Square(Error) = 3780,012.

a. Uses Harmonic Mean Sample Size = 486,000.

b. Alpha = 0,05.

CPU_Daire

Duncan^{a,b}

Populasyon	N	Subset	
		1	2
3n	486	23,9717	
4n	486	29,8883	
6n	486		40,6730
Sig.		,134	1,000

Means for groups in homogeneous subsets are displayed.

Based on observed means.

The error term is Mean Square(Error) = 3780,012.

a. Uses Harmonic Mean Sample Size = 486,000.

b. Alpha = 0,05.

CPU_Daire

Duncan^{a,b}

Durdurma	N	Subset		
		1	2	3
25n	486	16,9430		
50n	486		27,2477	
100n	486			50,3423
Sig.		1,000	1,000	1,000

Means for groups in homogeneous subsets are displayed.

Based on observed means.

The error term is Mean Square(Error) = 3780,012.

a. Uses Harmonic Mean Sample Size = 486,000.

b. Alpha = 0,05.



EK-8. Alan maksimizasyonu probleminde uygunluk değerine göre istatistiksel analiz sonuçları

Between-Subjects Factors

	N
Pc	486
	486
	486
Pm	486
	486
	486
Populasyon	486
	486
	486
Durdurma	486
	486
	486

Tests of Between-Subjects Effects

Dependent Variable: FF_Alan

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Corrected Model	3052,403 ^a	80	38,155	,001	1,000
Intercept	162623065,2	1	162623065,2	2274,376	,000
Pc	1280,631	2	640,316	,009	,991
Pm	8,331	2	4,165	,000	1,000
Populasyon	135,702	2	67,851	,001	,999
Durdurma	1290,505	2	645,252	,009	,991
Pc * Pm	2,774	4	,693	,000	1,000
Pc * Populasyon	26,717	4	6,679	,000	1,000
Pc * Durdurma	46,417	4	11,604	,000	1,000
Pm * Populasyon	8,317	4	2,079	,000	1,000
Pm * Durdurma	4,616	4	1,154	,000	1,000
Populasyon * Durdurma	28,067	4	7,017	,000	1,000
Pc * Pm * Populasyon	34,536	8	4,317	,000	1,000
Pc * Pm * Durdurma	15,184	8	1,898	,000	1,000
Pc * Populasyon * Durdurma	38,663	8	4,833	,000	1,000
Pm * Populasyon * Durdurma	37,266	8	4,658	,000	1,000
Pc * Pm * Populasyon * Durdurma	94,677	16	5,917	,000	1,000
Error	98458622,07	1377	71502,267		
Total	261084739,7	1458			
Corrected Total	98461674,48	1457			

a. R Squared = ,000 (Adjusted R Squared = -,058)



EK-8. (devam) Alan maksimizasyonu probleminde uygunluk değerine göre istatistiksel analiz sonuçları

FF_Alan

Duncan^{a,b}

Pc	N	Subset
		1
,70	486	332,9356
,80	486	333,7791
,90	486	335,2064
Sig.		,902

Means for groups in homogeneous subsets are displayed.

Based on observed means.
The error term is Mean Square(Error) = 71502,267.

a. Uses Harmonic Mean
Sample Size = 486,000.

b. Alpha = 0,05.

FF_Alan

Duncan^{a,b}

Pm	N	Subset
		1
,05	486	333,8772
,20	486	333,9823
,10	486	334,0617
Sig.		,992

Means for groups in homogeneous subsets are displayed.

Based on observed means.
The error term is Mean Square(Error) = 71502,267.

a. Uses Harmonic Mean
Sample Size = 486,000.

b. Alpha = 0,05.

FF_Alan

Duncan^{a,b}

Populasyon	N	Subset
		1
3n	486	333,6043
4n	486	333,9655
6n	486	334,3514
Sig.		,968

Means for groups in homogeneous subsets are displayed.

Based on observed means.
The error term is Mean Square (Error) = 71502,267.

a. Uses Harmonic Mean Sample Size = 486,000.

b. Alpha = 0,05.

FF_Alan

Duncan^{a,b}

Durdurma	N	Subset
		1
25n	486	332,7865
50n	486	334,0472
100n	486	335,0875
Sig.		,901

Means for groups in homogeneous subsets are displayed.

Based on observed means.
The error term is Mean Square (Error) = 71502,267.

a. Uses Harmonic Mean Sample Size = 486,000.

b. Alpha = 0,05.



EK-9. Alan maksimizasyonu probleminde CPU zamanına göre istatistiksel analiz sonuçları

Between-Subjects Factors

	N
Pc	486
	486
	486
Pm	486
	486
	486
Populasyon	486
	486
	486
Durdurma	486
	486
	486

Tests of Between-Subjects Effects

Dependent Variable: CPU_Alan

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Corrected Model	309623,154 ^a	80	3870,289	1,766	,000
Intercept	943809,558	1	943809,558	430,649	,000
Pc	90504,970	2	45252,485	20,648	,000
Pm	6,066	2	3,033	,001	,999
Populasyon	47025,983	2	23512,991	10,729	,000
Durdurma	129835,067	2	64917,533	29,621	,000
Pc * Pm	753,641	4	188,410	,086	,987
Pc * Populasyon	4056,444	4	1014,111	,463	,763
Pc * Durdurma	21557,376	4	5389,344	2,459	,044
Pm * Populasyon	1251,431	4	312,858	,143	,966
Pm * Durdurma	821,607	4	205,402	,094	,984
Populasyon * Durdurma	4327,023	4	1081,756	,494	,740
Pc * Pm * Populasyon	1399,558	8	174,945	,080	1,000
Pc * Pm * Durdurma	2068,570	8	258,571	,118	,999
Pc * Populasyon * Durdurma	1447,481	8	180,935	,083	1,000
Pm * Populasyon * Durdurma	2422,906	8	302,863	,138	,997
Pc * Pm * Populasyon * Durdurma	2145,032	16	134,065	,061	1,000
Error	3017830,674	1377	2191,598		
Total	4271263,386	1458			
Corrected Total	3327453,828	1457			

a. R Squared = ,093 (Adjusted R Squared = ,040)



EK-9 (devam) Alan maksimizasyonu probleminde CPU zamanına göre istatistiksel analiz sonuçları

CPU_Alan

Duncan^{a,b}

Pc	N	Subset		
		1	2	3
,70	486	16,7389		
,80	486		23,7700	
,90	486			35,8192
Sig.		1,000	1,000	1,000

Means for groups in homogeneous subsets are displayed.
Based on observed means.
The error term is Mean Square(Error) = 2191,598.
a. Uses Harmonic Mean Sample Size = 486,000.
b. Alpha = 0,05.

CPU_Alan

Duncan^{a,b}

Pm	N	Subset
		1
,05	486	25,3534
,20	486	25,4713
,10	486	25,5034
Sig.		,963

Means for groups in homogeneous subsets are displayed.
Based on observed means.
The error term is Mean Square(Error) = 2191,598.
a. Uses Harmonic Mean Sample Size = 486,000.
b. Alpha = 0,05.

CPU_Alan

Duncan^{a,b}

Populasyon	N	Subset	
		1	2
3n	486	19,7111	
4n	486	23,4360	
6n	486		33,1811
Sig.		,215	1,000

Means for groups in homogeneous subsets are displayed.
Based on observed means.
The error term is Mean Square(Error) = 2191,598.
a. Uses Harmonic Mean Sample Size = 486,000.
b. Alpha = 0,05.

CPU_Alan

Duncan^{a,b}

Durdurma	N	Subset		
		1	2	3
25n	486	15,2648		
50n	486		23,0561	
100n	486			38,0072
Sig.		1,000	1,000	1,000

Means for groups in homogeneous subsets are displayed.
Based on observed means.
The error term is Mean Square(Error) = 2191,598.
a. Uses Harmonic Mean Sample Size = 486,000.
b. Alpha = 0,05.



GAZİLİ OLMAK AYRICALIKTIR..