

**KARINCA KOLONİSİ OPTİMİZASYONU  
ALGORİTMASI İLE DEPO ROTA PLANLAMASI**

**Furkan ULU**



T.C.  
BURSA ULUDAĞ ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ

**KARINCA KOLONİSİ OPTİMİZASYONU ALGORİTMASI İLE DEPO ROTA  
PLANLAMASI**

Furkan ULU  
0000-0002-6651-4659

Prof. Dr. Fatih ÇAVDUR  
(Danışman)

YÜKSEK LİSANS TEZİ  
ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI

BURSA – 2023  
Her Hakkı Saklıdır

**Her Hakkı Saklıdır**

**TEZ ONAYI**

Furkan ULU tarafından hazırlanan “Karıncı Kolonisi Optimizasyonu Algoritması ile Depo Rota Planlaması” adlı tez çalışması aşağıdaki jüri tarafından oy birliği/oy çokluğu ile Bursa Uludağ Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalı’nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

**Danışman** : Prof. Dr. Fatih ÇAVDUR

**Başkan** : Prof. Dr. Fatih ÇAVDUR  
0000-0001-8054-5606  
B.U.Ü. Mühendislik Fakültesi,  
Yöneylem Araştırması Anabilim Dalı İmza

**Üye** : Doç. Dr. Tülin İNKAYA  
0000-0002-6260-0162  
B.U.Ü. Mühendislik Fakültesi,  
Endüstri Mühendisliği Anabilim Dalı İmza

**Üye** : Dr. Öğr. Üyesi Yunus DEMİR  
0000-0003-3868-1860  
B.T.Ü. Mühendislik ve Doğa Bilimleri Fakültesi,  
Yöneylem Araştırması Anabilim Dalı İmza

**Yukarıdaki sonucu onaylarım**

**Prof. Dr. Hüseyin Aksel EREN**  
**Enstitü Müdürü**  
**02/02/2023**

**B.U.Ü. Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmasında;**

- tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- ve bu tezin herhangi bir bölümünü bu üniversite veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

**beyan ederim.**

02/01/2023

**Furkan ULU**

**TEZ YAYINLANMA  
FİKRİ MÜLKİYET HAKLARI BEYANI**

Enstitü tarafından onaylanan lisansüstü tezin/raporun tamamını veya herhangi bir kısmını, basılı (kâğıt) ve elektronik formatta arşivleme ve aşağıda verilen koşullarla kullanıma açma izni Bursa Uludağ Üniversitesi'ne aittir. Bu izinle Üniversiteye verilen kullanım hakları dışındaki tüm fikri mülkiyet hakları ile tezin tamamının ya da bir bölümünün gelecekteki çalışmalarda (makale, kitap, lisans ve patent vb.) kullanım hakları tarafımıza ait olacaktır. Tezde yer alan telif hakkı bulunan ve sahiplerinden yazılı izin alınarak kullanılması zorunlu metinlerin yazılı izin alınarak kullandığını ve istenildiğinde suretlerini Üniversiteye teslim etmeyi taahhüt ederiz.

Yükseköğretim Kurulu tarafından yayınlanan “**Lisansüstü Tezlerin Elektronik Ortamda Toplanması, Düzenlenmesi ve Erişime Açılmasına İlişkin Yönerge**” kapsamında, yönerge tarafından belirtilen kısıtlamalar olmadığı takdirde tezin YÖK Ulusal Tez Merkezi / B.U.Ü. Kütüphanesi Açık Erişim Sistemi ve üye olunan diğer veri tabanlarının (Proquest veri tabanı gibi) erişimine açılması uygundur.

Prof. Dr. Fatih ÇAVDUR  
02/01/2023

Furkan ULU  
02/01/2023

## ÖZET

Yüksek Lisans Tezi

### KARINCA KOLONİSİ OPTİMİZASYONU ALGORİTMASI İLE DEPO ROTA PLANLAMASI

**Furkan ULU**

Bursa Uludağ Üniversitesi  
Fen Bilimleri Enstitüsü  
Endüstri Mühendisliği Anabilim Dalı

**Danışman:** Prof. Dr. Fatih ÇAVDUR

Günümüzde tedarik zincirlerinde malzeme akışının hacmi günden güne artmaktadır. Artan malzeme akış hızı, müşteri taleplerini zamanında ve doğru karşılamayı zorlaştırdığından depolarda verimlilik artırıcı çalışmaların yapılması büyük önem arz etmektedir. Bu tez çalışması kapsamında, dikdörtgen şekilli depolarda toplayıcı rotalama problemi için bütünsel ve esnek bir çözüm yaklaşımı önerilmektedir. Ele alınan rotalama probleminde, birbirine paralel raflardan oluşan bir depoda malzeme toplama işlemlerini yapan tek bir aracın olduğu varsayılmaktadır. Söz konusu malzeme toplama işlemlerinin gerçekleştirileceği herhangi bir turda belirli bir başlangıç noktasından başlayan, müşterinin talep ettiği malzemelerin bulunduğu noktalardaki talepleri toplayarak tekrar aynı başlangıç noktasına geri dönen bir toplayıcı için minimum mesafeli rotanın oluşturulması amaçlanmaktadır. Buradan ele alınan problemin literatürdeki Gezgin Satıcı Problemi (GSP) olduğu çıkarımı yapılabilir. Çalışma kapsamında önerilen çözüm yaklaşımına göre, her tur için ilgili turdaki talep noktaları da dikkate alınarak depo geometrisini gösteren temsili bir ağ modeli dinamik olarak oluşturulmaktadır. Oluşturulan ağ modelindeki bazı düğümler, toplayıcının ziyaret etmesi gereken noktaları temsil etmektedir. Sonraki adımda, yine her tur için karşılık gelen gezgin satıcı problemine ait mesafe matrisi oluşturulmakta ve bu amaçla ilgili talep noktaları arasındaki en kısa yolları hesaplamak için Floyd-Warshall algoritması kullanılmaktadır. Son olarak, oluşturulan gezgin satıcı problemini çözmek için Karınca Kolonisi Optimizasyonu (KKO) algoritması kullanılmıştır. Önerilen yaklaşımın farklı depo yapılarında ve farklı senaryolarda uygulanabilmesi amacıyla Python programlama dili kullanılarak bir kullanıcı arayüzü tasarlanmış ve oluşturulan yazılım yardımıyla sezgisel çözümün simüle edilmesi sağlanmıştır.

**Anahtar Kelimeler:** En kısa yol problemi, Gezgin satıcı problemi, Floyd-Warshall algoritması, Karınca kolonisi optimizasyonu, Rota planlama, Depo lojistiği

**2023, vii + 46 sayfa.**

## **ABSTRACT**

MSc Thesis

### **WAREHOUSE ROUTE PLANNING USING ANT COLONY OPTIMIZATION ALGORITHM**

**Furkan ULU**

Bursa Uludag University  
Graduate School of Natural and Applied Sciences  
Department of Industrial Engineering

**Supervisor:** Prof. Dr. Fatih ÇAVDUR

Nowadays, the volume of material flow in supply chains is increasing day by day. As the increasing material flow rate makes it difficult to meet customer demands on time and accurately, it is of great importance to carry out productivity-enhancing studies in warehouses. In this thesis, an integrated and flexible solution approach is proposed for the picker routing problem in rectangular warehouses. In this routing problem discussed, it is assumed that there is only one picker that performs material picking operations in a warehouse consisting of parallel shelves. In any tour where material picking will be carried out, it is aimed to create a minimum distance route for a picker that starts from a certain starting point and returns to the same starting point by picking the demands from the points where the materials requested by customer. It can be concluded that the problem addressed here is the Traveling Salesman Problem (TSP) in the literature. According to the proposed solution approach within the scope of the study, a representative network model showing the warehouse geometry is dynamically created for each tour, considering the demand points in the relevant tour. Some nodes in the generated network model represent points that the picker must visit. In the next step, the distance matrix is created to be used in the traveling salesman problem for each tour, and for this purpose, the Floyd-Warshall algorithm is used to calculate the shortest paths between the relevant demand points. At last, Ant Colony Optimization (ACO) algorithm is used to solve the traveling salesman problem. In order to implement the proposed approach in different warehouse structures and different scenarios, a user interface was designed using the Python programming language and the heuristic solution that obtained was simulated with the help of the created software.

**Key words:** Shortest path problem, Traveling salesman problem, Floyd-Warshall algorithm, Ant colony optimization, Route planning, Warehouse logistics

**2023, vii + 46 pages.**

## TEŐEKKÜR

Bu alıőmanın hazırlanma sűrecinde, deęerli bilgilerini benimle paylaőan, kendisine her danıőtıęımda bana kıymetli zamanını ayırıp sabırla ve bűyűk bir ilgiyle bana faydalı olabilmek adına elinden gelenden fazlasını sunan, gűler yűzűnű ve samimiyetini benden esirgemeyen ve gelecekteki mesleki hayatımda da bana verdięi deęerli bilgilerden faydalanacaęımı dűőűndűęűm saygıdeęer danıőmanım Prof. Dr. Fatih AVDUR'a teőekkűrlerimi sunarım.

Yalnız bu alıőmamda deęil, hayatımın her anında desteęini benden esirgemeyen ve benim iin hibir fedakârlıęı yapmaktan bir an olsun geri durmayan kıymetli aileme sonsuz teőekkűrlerimi sunarım.

Furkan ULU  
02/01/2023



## İÇİNDEKİLER

	<b>Sayfa</b>
ÖZET .....	i
ABSTRACT .....	ii
TEŞEKKÜR .....	iii
KISALTMALAR DİZİNİ .....	v
ŞEKİLLER DİZİNİ .....	vi
ÇİZELGELER DİZİNİ .....	vii
1. GİRİŞ .....	1
2. KAYNAK ARAŞTIRMASI .....	3
3. MATERYAL ve YÖNTEM.....	14
3.1. Materyal .....	14
3.1.1. Dikdörtgen Şekilli Depoların Ağ Gösterimi .....	14
3.1.2. En Kısa Yol Problemi .....	16
3.1.3. Gezgin Satıcı Problemi .....	21
3.1.4. Sezgisel Algoritmalar .....	23
3.2. Yöntem.....	26
3.2.1. Örnek Deponun Ağ Gösterimi ile Tanımlanması .....	26
3.2.2. Örnek Depo için En Kısa Yol Hesaplamaları .....	28
3.2.3. Depo için Gezgin Satıcı Problemi Kurgusunun Oluşturulması .....	30
3.2.4. Problem için Karınca Kolonisi Optimizasyonu Uygulaması .....	31
4. BULGULAR ve TARTIŞMA .....	35
4.1. Depo Boyutlarının Değiştirilmesi .....	35
4.2. Toplama Yeri Sayısının Değiştirilmesi .....	37
4.3. Karınca Kolonisi Optimizasyon Parametrelerinin Değiştirilmesi .....	39
5. SONUÇ .....	42
KAYNAKLAR .....	44
ÖZGEÇMİŞ .....	46

## KISALTMALAR DİZİNİ

<b>Kısaltmalar</b>	<b>Açıklama</b>
TZÜ	Tam Zamanında Üretim
GSP	Gezgin Satıcı Problemi
KKO	Karınca Kolonisi Optimizasyonu
FW	Floyd-Warshall

## ŞEKİLLER DİZİNİ

	<b>Sayfa</b>
Şekil 2.1. Standart bir depo için genel depo operasyon akışı .....	5
Şekil 2.2. Depolarda yapılan operasyonlara göre alan kullanımı .....	5
Şekil 2.3. Geleneksel bir depo örneği .....	6
Şekil 2.4. Depolardaki lojistik süreç akışları .....	7
Şekil 2.5. Sipariş toplama sistemlerinin sınıflandırılması.....	8
Şekil 2.6. Tipik bir depoda sipariş toplama zamanının dağılımı .....	9
Şekil 2.7. Tek bloklu bir depoda örnek rotalama metotları .....	13
Şekil 3.1. Dikdörtgen şekilli basit bir depo örneği .....	14
Şekil 3.2. Dikdörtgen şekilli depo yapılarında genel ağ gösterimi .....	15
Şekil 3.3. Depo ağ yapısının detay gösterimi .....	16
Şekil 3.4. Dijkstra algoritmasının akış şeması .....	18
Şekil 3.5. Belman-Ford algoritmasının akış şeması.....	19
Şekil 3.6. Floyd-Warshall algoritmasının sözde kod gösterimi.....	20
Şekil 3.7. Johnson algoritmasının sözde kod gösterimi.....	21
Şekil 3.8. Karıncaların yuva-yiyecek arası en kısa mesafeyi bulması .....	24
Şekil 3.9. İki yatay, üç dikey koridorlu; her sırada dört bitişik depolama yerinden oluşan dikdörtgen şekilli bir depo örneği .....	27
Şekil 3.10. Deponun matris gösterimi.....	27
Şekil 3.11. 16 koridor düğümü için Floyd-Warshall başlangıç matrisi .....	28
Şekil 3.12. 16 koridor düğümü için Floyd-Warshall uygulanmış uzaklık matrisi .....	29
Şekil 3.13. 16 koridor düğümü için Floyd-Warshall uygulanmış öncelik matrisi .....	29
Şekil 3.14. 1;1 düğümünden 4;4 düğümüne giden en kısa rotanın bulunması ....	30
Şekil 3.15. Oluşturulan yazılımın kullanıcı arayüzü.....	32
Şekil 3.16. Örnek depo yapısının program tarafından çizilmesi.....	33
Şekil 3.17. Örnek depo yapısı üzerinden rotalama .....	34
Şekil 4.1. Depo parametre setlerine göre oluşan depo yapıları.....	36
Şekil 4.2. Toplama yeri parametre setlerine göre oluşan iki örnek depo.....	38

## ÇİZELGELER DİZİNİ

	<b>Sayfa</b>
Çizelge 3.1. Karınca kolonisi algoritması parametreleri .....	25
Çizelge 4.1. İncelenecek depo parametre setleri .....	35
Çizelge 4.2. Toplama yeri sayısı değişiminin incelenmesi .....	37
Çizelge 4.3. Karınca sayısı değişiminin incelenmesi .....	39
Çizelge 4.4. Buharlaşma katsayısı değişiminin incelenmesi .....	40
Çizelge 4.5. Feromon yoğunluğu değişiminin incelenmesi .....	41

## 1. GİRİŞ

Müşteriyi kazanmanın ve elde tutmanın her geçen gün daha da önem kazanmasıyla işletmeler, ürün çeşitliliğini artırmakta ve bu çeşitlilik içerisinde müşterinin istediği ürünü en kısa zamanda müşteriye ulaştırmayı amaçlamaktadır. Bunu sağlamanın en önemli yolu ise tedarik zinciri operasyonlarının verimliliğini artırmak ve bunu sürdürülebilir kılmaktır. Tedarik zincirinin etkin bir şekilde yönetilmesi müşteri memnuniyetinin artışı ve işletme maliyetlerinin azalmasını sağlamanın en önemli yollarından biridir. Lojistik operasyonları ürün üzerinde değer yaratmayan faaliyetlerin başında geldiğinden bu alanda yapılacak en küçük iyileştirme dahi işletmenin israflarını azaltması ve ortadan kaldırması için çok önemlidir. Verimli bir lojistik operasyonu oluşturmak işletmelerin genel maliyetlerini de azaltacağından rekabet gücünü önemli ölçüde artıracaktır.

Lojistik operasyonları geniş ölçekte incelendiğinde iç süreçleri birbirinden bağımsız iki sistemin birbirleri arasındaki akışlar görülmekte olup daha küçük bir ölçekte incelendiğinde ise bağlantılı süreçler arasındaki akışlar görülmektedir. Bu operasyonları malzeme akışının yönetilmesi ve ürünlerin müşterilere dağıtımının haricinde nakliye, dağıtım, depolama ve sipariş işleme süreçlerini de içerir. Mikro ölçekte bu operasyonlar sistemin iç dinamikleri ile etkileşim halinde olduğundan yapılacak iyileştirmeler sonuçlara ve sistemin yapısına doğrudan etki etmektedir.

Geçmişten günümüze tedarik zincirinde önemli bir bileşen olan depolama, lojistik faaliyetleri destekleyen temel yapı taşlarından biridir. Üretilen malzemelerin depolanması ve oluşturulan bu stok alanının işletilmesi tüm endüstriyel faaliyetlerin içerisinde büyük öneme sahiptir. Depolar, hammaddeden bitmiş ürüne kadar tüm seviyelerin stoklandığı alanlardır. Tüm bu malzemelerin tedarik edilmesi, üretilmesi ve dağıtım süreçlerinde merkezde yer almaktadır. Müşteriye sevkiyatların yapıldığı depolar, hizmetin oluşturulduğu nokta olduğundan müşteri memnuniyeti açısından kritik önemdedir.

Depolardaki en önemli faaliyetlerden biri sipariş toplama operasyonudur. Sipariş toplama, depolardaki malzemelerin müşteri ya da müşteri proseslerin taleplerine göre toplanması sürecidir. Sipariş toplama, genellikle depolarda dağıtım merkezi olarak çalışan bir birimde gerçekleştirilir. Sipariş toplama operasyonunda, müşterinin talep ettiği ürünün doğru miktarlarda ve doğru zamanda teslim edilmesi amaçlanmaktadır. Bu süreç, müşteri siparişlerine göre depodaki stokların taranması, ürünlerin toplanması ve

paketlenmesi adımlarını içerir. Sipariş toplama, depolarda önemli bir işlemdir ve müşteri memnuniyeti açısından kritik öneme sahiptir. Sipariş toplama süreci etkin bir şekilde gerçekleştirildiğinde, müşteriler talep ettikleri ürünü hızlı ve doğru bir şekilde teslim alabilirler.

Sipariş toplama operasyonu yapılırken, toplamayı yapan personelin toplama noktalarını dikkate alarak bir rota oluşturması gerekir. Rotalama, depolarda müşterinin istediği malzemelerin yerleştirildiği bölgelerin belirlenmesi ve bu bölgelerin yine müşteri siparişleri dikkate alınarak doğru şekilde taranması sürecidir. Rotalama ile depodaki malzemeler bir plan doğrultusunda sipariş toplama sürecine dahil edilebilir.

Rotalama yapılırken temel hedef optimum rotanın bulunmasıdır. Optimum rota, depolarda malzemelerin en etkin şekilde taranmasını sağlayacak olan yol olarak tanımlanabilir. Optimum rota, malzemelerin depoda düzenli bir şekilde stoklandığı ve toplayıcıların doğru şekilde rotalandığı durumlarda belirlenir. Optimum rota bulunarak depodaki toplanması planlanan tüm malzemelerin en az zaman ve emek harcanarak taranması sağlanır, bu da depolama maliyetlerini azaltmaya yardımcı olur.

Depolarda optimum rotalamanın yapılabilmesi için sezgisel yöntemler, matematiksel modellemeler ve yapay zekâ yöntemleri kullanılabilir. Bu yöntemler arasından hangisinin kullanılacağı, depodaki malzemelerin yerleşimi ve deponun yapısına göre belirlenir. Her yöntemin kendine özgü avantajları ve dezavantajları vardır ve optimum rotanın bulunması sürecinde hangisinin kullanılacağı, depodaki çeşitli faktörlere ve optimizasyonu yapacak kişinin seçimine göre değişiklik gösterebilir.

Çalışmanın ilerleyen bölümlerinin organizasyonunda, “Kaynak Araştırması”, bölümünde literatürde yer alan ilgili çalışmalar özetlenmektedir. Sonraki bölüm olan “Materyal ve Yöntem” bölümünde tez kapsamında önerilen yaklaşımlar, kullanılan model ve algoritmalar açıklanmakta, takip eden “Bulgular ve Tartışma” bölümünde ise önerilen yaklaşımların uygulanmasıyla elde edilen sonuçlar sunulmaktadır. Son kısımda yer alan “Sonuç” bölümü tez çalışmasının genel bir değerlendirmesini sunmaktadır.

## 2. KAYNAK ARAŞTIRMASI

Depolar, faaliyet gösterdikleri tedarik zincirlerinin ayrılmaz bir parçasıdır ve bu nedenle artan piyasa değişkenliği, ürün yelpazesinin çoğalması ve müşteri tedarik sürelerinin kısılması gibi son trendlerin tümü, depoların yerine getirmesi gereken roller üzerinde bir etkiye sahiptir (Rushton vd. 2017). Depoların, bir bütün olarak değerlendirildiğinde içinde buldukları tedarik zincirinin özel gereksinimlerine uygun olarak tasarlanması ve işletilmesi gerekmektedir. Yapılan bu tasarım ve işletme süreci müşteri memnuniyet düzeyini artırma ve maliyeti minimize etme odaklı olmalıdır.

Tesisin oluşturulması, işletilmesi, personel ve ekipman kullanımı gibi nedenlerden tedarik zincirinin en maliyetli unsurlarından biri depolar olduğundan, depo tasarımı ve sonrasındaki işletme sürecinin başarılı yönetilmesi depolanan ürünün rekabetçi olabilmesi için kritik önem arz etmektedir. Çoğu deponun en temel amacı, ürünlerin nihai tüketiciye hareketini kolaylaştırmaktır.

Lambert vd. (1998), dünya çapında, son teknolojiye uygun bir biçimde profesyonelce yönetilen depoların yanı sıra şirketlerin özel depolama tesisleri de dahil olmak üzere 750.000 adetten fazla depo tesisi bulunduğunu belirtmektedir. Depolar genellikle büyük yatırımlar ve işletme maliyetleri (örneğin arazi maliyeti, tesis ekipmanı, işçilik vb.) içerir.

Lambert ve arkadaşlarına göre (1998) depolar, şirketin aşağıda da belirtilen pek çok sayıda misyonuna katkıda bulunurlar:

- Nakliye verimliliğinin sağlanması
- Üretim verimliliğinin geliştirilmesi
- Satın alma indirimlerinden ve vadeli alım fırsatlarından yararlanılması
- Firmanın müşteri hizmetleri politikalarının desteklenmesi
- Değişen pazar koşullarına ve belirsizliklere karşı uyum sağlanması
- Üreticiler ve müşteriler arasında var olan zaman ve alansal uyumsuzlukların ortadan kaldırılması
- İstenen müşteri hizmeti seviyesiyle orantılı olarak en düşük toplam maliyetli lojistik operasyonların gerçekleştirilmesi
- Tedarikçilerin ve müşterilerin tam zamanında üretim (TZÜ) programlarının desteklenmesi

- Müşterilere her siparişte tek bir ürün yerine bir ürün grubu sağlanması (konsolidasyon)
- Bertaraf edilecek veya geri dönüştürülecek malzemelerin geçici olarak depolanmasının sağlanması
- Aktarma için bir tampon konumu sağlanması

Rushton vd. (2017) tedarik zincirlerindeki depoların yapısının birçok farklı sınıflandırma türü olduğunu belirtmiştir. Bunlar:

- Tedarik zincirindeki aşamaya göre: ham maddeler, yarı mamuller, bitmiş ürünler veya iade ürünler.
- Coğrafi alana göre: küresel, bölgesel ya da ulusal.
- Ürün tipine göre: örneğin, küçük parçalar, büyük parçalar, dondurulmuş gıdalar, çabuk bozulan maddeler, güvenlik maddeleri ve tehlikeli mallar.
- Fonksiyona göre: örneğin, envanter tutma veya sınıflandırma.
- Mülkiyete göre: direkt kullanıcıya veya bir üçüncü taraf lojistik şirketine ait olabilir.
- Şirket kullanımına göre: örneğin, bir şirket için ayrılmış bir depo veya birkaç şirket için tedarik zincirlerini yöneten ortak kullanıcıya bir depo.
- Hacim ve kapasite kullanımına göre: merkezi depolar, tali depolar, bayi depoları, aktarma depoları
- Ekipman kullanım türüne göre: manuel veya otomatik.

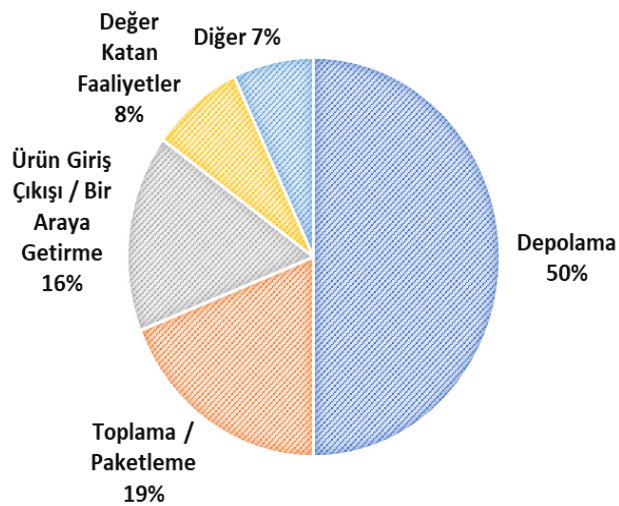
Çoğu depoda ortak olan temel bazı operasyonlar vardır. Bu operasyonlar, deponun ekipman kullanım türüne göre (manuel veya otomatik) farklılık gösterebilir de temelde aynıdır. Standart bir depo için, tipik depo operasyonları ve malzeme akışları Şekil 2.1’de gösterilmektedir.





**Şekil 2.1.** Standart bir depo için genel depo operasyon akışı (Rushton vd. 2017)

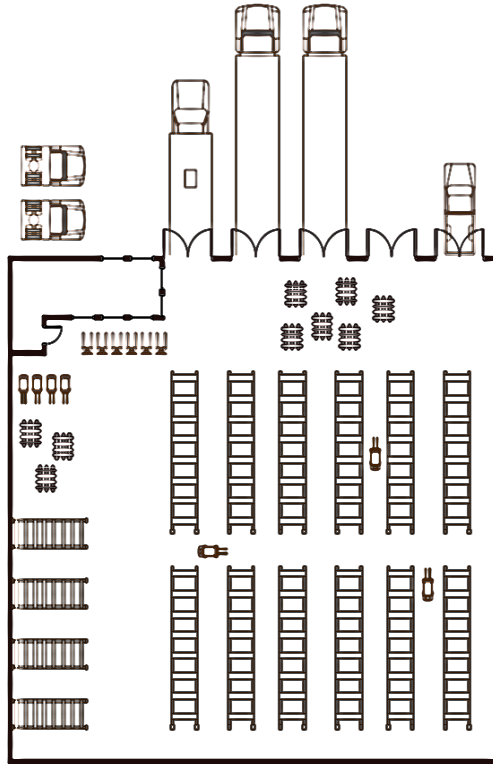
Şekil 2.1’de verilen genel depo operasyon akışında her operasyon belirli bir alan kullanmaktadır. Alan kullanımı ile ilgili grafik Şekil 2.2’deki gibidir. Buna göre en büyük alan kullanımı depolamaya aittir. Daha sonra sırasıyla toplama ve paketleme, ürün giriş çıkışı/bir araya getirme, değer katan faaliyetler gelmektedir.



**Şekil 2.2.** Depolarda yapılan operasyonlara göre alan kullanımı (Baker ve Perotti 2008)

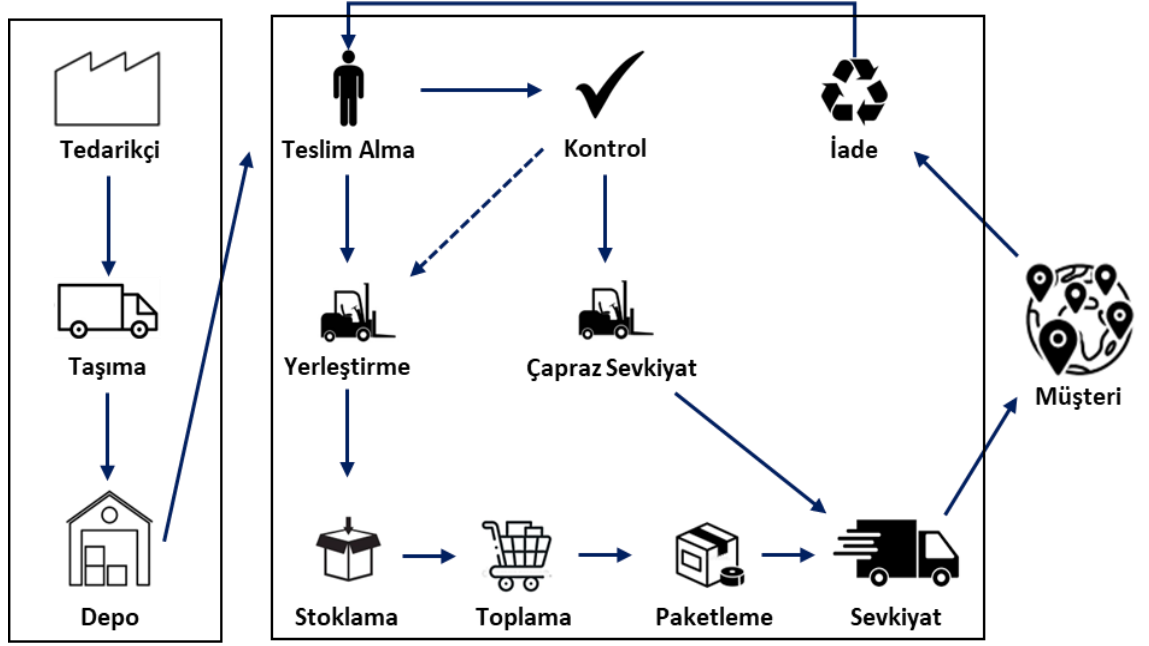
İşletmeler, depo yönetimini ele aldığıında iki önemli faktörü göz önünde bulundurmaktadır. İlk faktör müşteri talebini tam zamanında ve doğru bir biçimde karşılamak, ikincisi ise bunu minimum maliyet ile gerçekleştirmektedir. Ürünlerin müşteriye sevki gerçekleştirilirken herhangi bir hata kabul edilemez ve sürecin hatasız ve minimum maliyet ile ilerletilmesi amaçlanır.

Birbirini büyük ölçüde zıt yönde etkileyen bu iki faktör sebebiyle depolarda her iki faktörü de dikkate alacak şekilde optimizasyon yapılması önem arz etmektedir. Depo optimizasyonu, bir depodaki zaman ve alan gibi tüm kaynakların müşteri memnuniyetini olumsuz etkilemeyecek şekilde verimli kullanımının sağlanması sürecidir. Şekil 2.3'te geleneksel bir örneği verilen depoda optimizasyon faaliyetleri üç gruba ayrılabilir. Bunlar, deponun temel teknik yapısı ve kurulumu, depodaki operasyonel ve organizasyonel çerçeve ve son olarak depo operasyonları için koordinasyon ve kontrol sistemleridir (Karásek 2013).



**Şekil 2.3.** Geleneksel bir depo örneği (Karásek 2013)

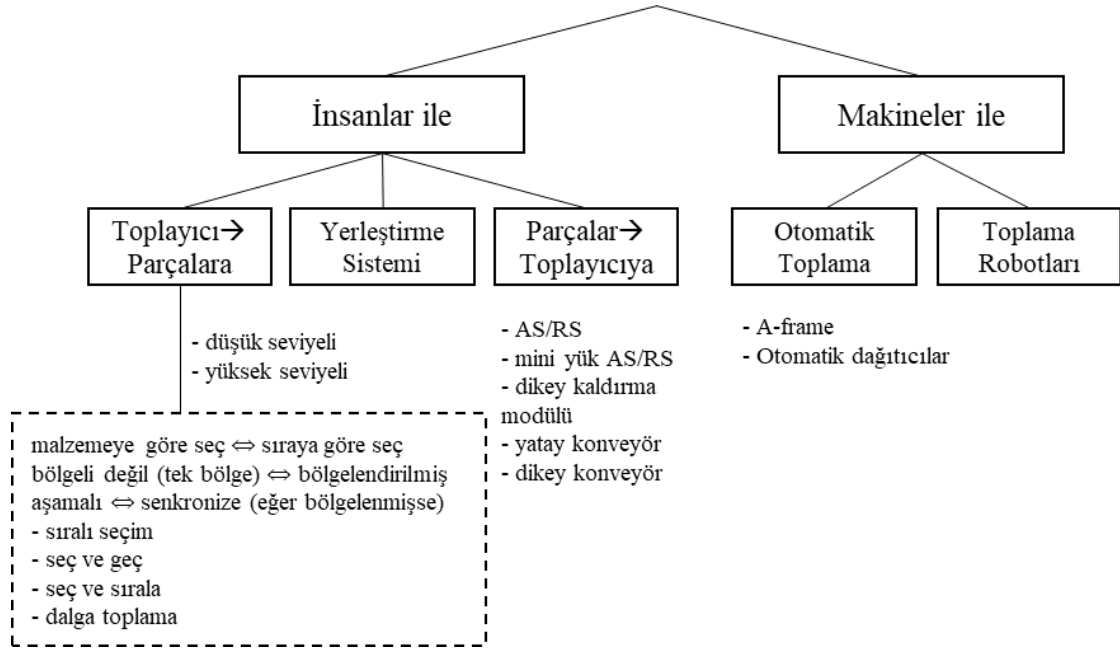
Depolarda Şekil 2.4'te de temel olarak gösterilen, teslim alma, depolama, iç taşıma, toplama ve sevkiyat operasyonları bulunmaktadır. Teslim alma, süreç akışına göre bir depodaki ilk faaliyettir. İlgili süreç, tedarikçi procesten ürün geliş bildiriminin yapılmasıyla başlayıp indirme, tanımlama, kontrol ve ürün kabul süreçleri ile devam etmektedir. Kabulü yapılan ürünler depoda yerleştirilerek stoklanır. Gelen müşteri siparişlerine göre toplama ve paketleme faaliyetleri gerçekleştirilir. Süreç son aşaması olan bitmiş ürünlerin müşteriye sevki ile tamamlanmaktadır.



Şekil 2.4. Depolardaki lojistik süreç akışları (Karasek 2013)

Sipariş toplama, müşteri siparişlerinin gruplandırılması ve programlanması, sipariş edilen referansların depo adreslerinden alınması ve toplanan ürünlerin sevk edilmesi sürecini içerir. Sipariş toplama yöntemleri Şekil 2.5'te görüldüğü üzere operasyonu makine ya da insanın yapması durumlarına göre ikiye ayrılır. Her iki durumun da kendi içerisinde alt detayları ve dinamikleri bulunmaktadır.

## Sipariş Toplama Yöntemleri



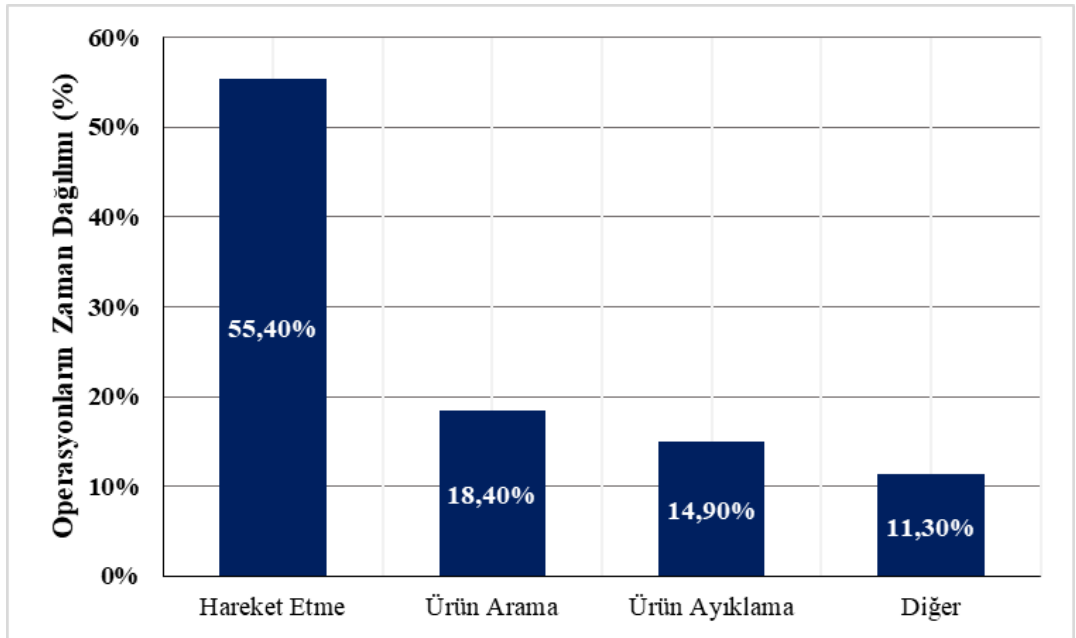
**Şekil 2.5.** Sipariş toplama sistemlerinin sınıflandırılması (De Koster 2004)

Depoların çoğu, sipariş toplama operasyonu için insan faktörünü kullanır. Bunlar arasında, sipariş toplayıcının ürünleri bulup almak için koridorlar boyunca hareket ettiği toplayıcıların parçalara gittiği sistemler en yaygın olanıdır (De Koster 2004). Toplayıcıların parçalara gittiği sistemler düşük seviyeli toplama ve yüksek seviyeli toplama olarak ikiye ayrılır. Düşük seviyeli sipariş toplama sistemlerinde, sipariş toplayıcı, koridorlar boyunca hareket ederken, talep edilen ürünleri raflardan alır. Diğer sipariş toplama sistemlerinde yüksek depolama yerleri kullanılır; sipariş toplayıcılar, sipariş toplama vinci ile toplama yerlerine giderler. Vinç, uygun toplama yerinin önünde otomatik olarak durur ve sipariş toplayıcının toplama işlemini gerçekleştirmesini bekler. Bu tür bir sisteme yüksek seviyeli sipariş toplama sistemi denir.

Parçaların toplayıcıya geldiği sistemler, çoğunlukla bir veya daha fazla birim yükü (paletleri veya kutuları) alan, koridora bağlı vinçlerin kullanıldığı otomatik depolama ve alma sistemlerini (Automated Storage and Retrieval Systems – AS/RS) içerir. Bu konumda, sipariş toplayıcı gerekli sayıda parçayı alır ve ardından kalan yük tekrar depolanır.

Yerleştirme sistemleri veya sipariş dağıtım sistemleri bir alım ve dağıtım sürecinden oluşur. İlk olarak, parçalar toplayıcıya veya toplayıcı parçalara yöntemlerinden biriyle malzemelerin alınması gerekir. İkinci olarak, bu malzemeler önceden seçilmiş birimlere sahip taşıyıcı tarafından, bunları müşteri siparişleri üzerinden dağıtan bir sipariş toplayıcıya sunulur.

İç lojistik operasyonları arasında, sipariş toplama genellikle en pahalı faaliyet olarak kabul edilir, çünkü emek ve sermaye yoğun bir yapısı vardır (Frazelle 2016). Sipariş toplama operasyonlarının zaman dağılımı Şekil 2.6'da gösterilmiştir. Grafikten de görüleceği üzere sipariş toplama süresinin yarıdan fazlası toplanacak noktalar arasında yapılan hareketten oluşmaktadır. Sipariş toplama ve bu operasyon esnasında rota oluşturulması depo operasyonlarının iyileştirilmesi için en yüksek öncelikli faaliyetlerin başında gelmektedir. Sipariş toplama esnasında optimum veya optimuma yakın bir rotanın bulunması, müşteri talebine hızlı cevap verme (tam zamanında sevkiyat hedefini gerçekleştirme), yalın süreçler oluşturma gibi amaçların yanı sıra işletme maliyetlerinin azaltılmasına da hizmet eder.



Şekil 2.6. Tipik bir depoda sipariş toplama zamanının dağılımı (Frazelle 2016)

Arařtırmalar sonucunda sipariř toplamadaki hareket mesafesini ve dolayısıyla sipariř toplama süresini etkileyen faktörlerin ařağıdaki řekilde ortaya ıktığı görölmektedir (Gu, Goetschalckx, ve McGinnis 2007):

- Deponun boyut ve yerleřim düzeni aısından genel yapısı
- Sipariře özel toplama ve toplu toplama gibi operasyonel stratejiler
- Depolama atama prosedürü
- Alan toplamanın kullanımı
- Toplayıcı yönlendirme
- Depoda aynı anda daha fazla toplayıcı alıřtıėında tıkanıklık olasılıėını artıran dar koridorların kullanımı

Sipariř toplama sistemlerinin en temel amacı, emek, makine ve sermaye gibi kaynak kısıtlamalarına baėlı olarak hizmet düzeyini maksimuma ıkartmaktır (Goetschalckx ve Ashayeri 1989). Hizmet düzeyi, sipariř teslim süresinin ortalama ve varyansı, sipariř bütünlüėü ve doėruluėu gibi eřitli faktörlerden oluşur. Sipariř toplamada hizmet düzeyi, bir sipariř geldiėinde müşteriye sevkiyatın ne kadar hızlı olduėu ile doėru orantılıdır. Bir sipariřin gönderiminde ge kalındıėında, bir sonraki gönderim dönemine kadar beklemek zorunda kalınabilir. Bu nedenle, sipariř toplama süresinin en aza indirilmesi, herhangi bir depo için önemli bir ihtiyaçtır.

Manuel sipariř toplama sistemleri için, seyahat süresi, seyahat mesafesinin artan bir fonksiyonudur (Jarvis ve McDowell 1991). Sonuç olarak, seyahat mesafesi genellikle ambar tasarımı ve optimizasyonunda birincil hedef olarak kabul edilir. Sipariř toplama literatüründe yaygın olarak ortalama ve toplam seyahat mesafesi olmak üzere iki tür seyahat mesafesi kullanılmaktadır. Bununla birlikte, belirli bir dizi sipariř için, ortalama tur uzunluėunu en aza indirmenin, toplam seyahat mesafesini en aza indirmekle eřdeėer olduėu söylenebilir.

Ortalama seyahat mesafesini (veya eřdeėer olarak toplam seyahat mesafesini) en aza indirmek, birçok seenekten yalnızca biridir. Diėer bir önemli hedef, toplam maliyeti (hem yatırım hem de iřletme maliyetlerini) en aza indirmek olacaktır. Ambar tasarımında

ve optimizasyonunda genellikle dikkate alınan diğer hedefler şunlardır (De Koster vd. 2007):

- Siparişin toplama süresinin en aza indirilmesi
- Alan, ekipman ve işgücü kullanım verimliliğinin en üst düzeye çıkarılması
- Tüm malzemelere erişilebilirliğin en üst düzeye çıkarılmasıdır.

Bir depoda sipariş toplayıcıların yönlendirilmesi sorunu aslında gezgin satıcı probleminin özel bir durumudur (Lawler vd. 1995). Sipariş toplayıcı, bir toplama listesi aldığı depoda başlangıç düğümünden başlayarak tüm toplama yerlerini ziyaret eder ve son olarak başlangıç noktasına geri dönmesi gerekir.

Cornuejols vd. (1985), Ratliff ve Rosenthal'ın (1983) algoritmasının, Steiner Gezgin Satıcı Problemi'ni tüm sözde seri-paralel ağ yapılarında çözmek için genişletilebileceğini göstermiştir. De Koster ve Van der Poort'un (1998) algoritması, merkezi olmayan depoya sahip, bir bloktan oluşan bir depoda en kısa sipariş toplama rotalarını belirleyebilmektedir. Roodbergen ve De Koster (2001c), biri önde, biri arkada ve biri ortada olmak üzere üç çapraz koridorlu bir depo için bir algoritma geliştirmiştir.

Depolarda rota planlaması, basit sezgisel yöntemlerden, hesaplamalı kompleks prosedürlere çeşitlilik göstermektedir. Uygulamada şirketler, genellikle optimal bir hareket mesafesi sağlamasalar da iç lojistik faaliyetlerinin rotalaması için daha hızlı çözüm sağlayan çok basit yönlendirme stratejilerini benimsemeyi tercih ederler. Bu durum, gerçek yaşamda optimal rotalamanın bazı dezavantajlarından kaynaklanmaktadır. Sipariş toplayıcılar, optimal rotaları mantıksız olarak görebilmekte ve bunun sonucunda bu rotalardan sapabilmektedirler (Gademann ve Van de Velde 2005). Basit rotalama stratejilerine; S-şekilli yönlendirme, en büyük aralık, geri dönüş veya orta noktaya göre yönlendirme örnek olarak verilebilir. Malzemeleri depodan toplayan operatörler bu stratejilere hızlı bir biçimde uyum sağlarlar ve böylece hata riskini en aza indirirler (Petersen 1999).

Sipariş toplayıcıları için en basit sezgisel yöntemlerden biri, S-şekilli rotalama sezgiselidir. S-şekilli rotalama yöntemini kullanarak sipariş toplayıcıları rotalamak, en az bir toplama noktası içeren herhangi bir koridorun tamamen geçilmesi anlamına

gelmektedir. Toplama noktası olmayan koridorlara girilmez. Son ziyaret edilen koridordan, sipariş toplayıcı başlangıç noktasına geri döner.

Sipariş toplayıcıları yönlendirmek için başka bir basit sezgisel yöntem, bir sipariş toplayıcının her koridora aynı uçtan girip çıktığı geri dönüş yöntemidir. Sadece toplama noktası olan koridorlar ziyaret edilir. Orta nokta yöntemi, depoyu temel olarak iki alana ayırır. İlk yarıdaki toplama noktalarına ön çapraz koridordan ve ikinci yarıdaki toplama noktalarına arka çapraz koridordan erişilir. Sipariş toplayıcı, ziyaret edilecek son veya ilk koridordan ikinci yarıya geçer. Hall'a (1993) göre bu yöntem, koridor başına toplama yeri sayısı az olduğunda, S-şekilli rotalama sezgiselinden daha iyi performans gösterir.

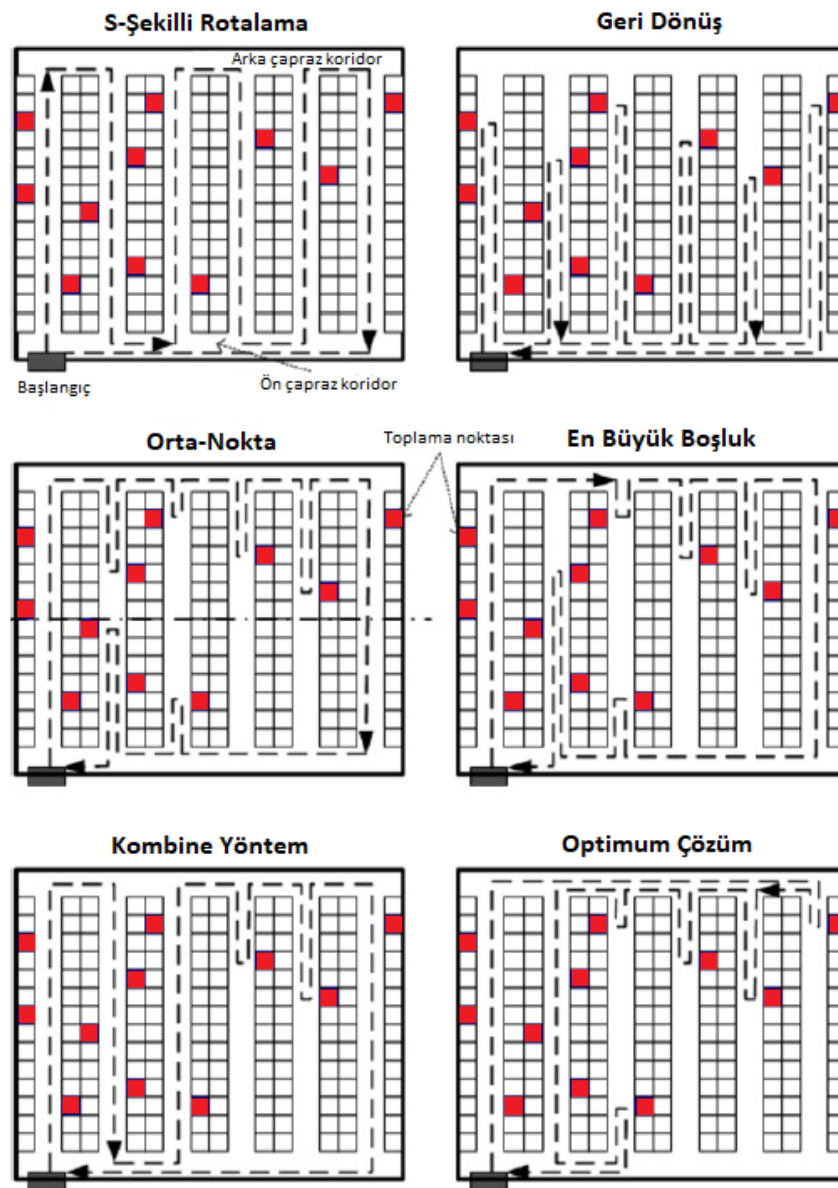
En büyük boşluk stratejisi, orta nokta yöntemine benzer, ancak bir sipariş toplayıcı, orta nokta yerine bir koridordaki en büyük boşluğa kadar ilerler. Boşluk, herhangi iki bitişik toplama noktası arasındaki, ilk toplama noktası ile ön koridor arasındaki veya son toplama noktası ile arka koridor arasındaki aralığı temsil eder. En büyük boşluk iki bitişik toplama noktası arasındaysa, sipariş toplayıcı koridorun her iki ucundan bir dönüş yolu gerçekleştirir. Aksi takdirde, ön veya arka koridordan bir dönüş rotası kullanılır. Bu nedenle, bir koridor içindeki en büyük boşluk, koridorun sipariş toplayıcının geçmediği kısmıdır. Arka koridora yalnızca ilk veya son koridordan erişilebilir. En büyük boşluk yöntemi her zaman orta nokta yönteminden daha iyi performans gösterir (Hall 1993). Ancak uygulama açısından orta nokta yöntemi daha basittir. Kombine (veya bileşik) bir sezgisel yöntem için, toplama noktası olan koridorlar ya tamamen geçilir ya da aynı uçtan girilir ve bırakılır. Ancak ziyaret edilen her koridor için seçim dinamik programlama kullanılarak yapılır (Roodbergen ve De Koster, 2001b).

Petersen (1997), S-şekilli rotalama, geri dönüş, en büyük boşluk ve orta nokta gibi sezgisel rotalama yöntemlerini karşılaştırmak için çeşitli sayısal deneyler gerçekleştirmiştir. Çalışmaları sonucunda en iyi sezgisel çözümün optimal çözümden ortalama %5 daha fazla olduğu sonucuna varmıştır. Lin ve Kernighan'ın (1973) k-optimal metodolojisini kullanan bir rota iyileştirme yöntemi, Makris ve Giakoumakis (2003) tarafından sunulmuştur. Verilen sezgisel rotalama stratejileri ile ilgili örnekler Şekil 2.7'de paylaşılmıştır.

Optimal prosedürler, bu rotalama problemine en genel manada gezgin satıcı probleminin (GSP) özel bir durumu olarak yaklaşır ve bu problem için kesin çözüm bulmaya çalışır.



Ratliff ve Rosenthal (1983) basit yapıda depolar için GSP'yi baz alarak bir optimal prosedür geliştirmişlerdir. Sonrasında literatürde bunun gibi birçok çalışma yapılmıştır. Depo büyüklüğü ve konfigürasyonunun karmaşıklığı arttıkça GSP çerçevesi kullanılarak problemin çözümü daha zor hale gelmektedir. Bu nedenle, daha karmaşık depo konfigürasyonlarında problemin çözümü için sezgisel algoritmaların kullanılması önem arz etmektedir. Burada önemli olan, kullanılan sezgisel algoritmanın kesin çözüm veren diğer algoritmalara göre başarısının kıyaslanarak doğru sonuçlar ürettiğinin doğrulanmasıdır.



Şekil 2.7. Tek bloklı bir depoda örnek rotalama metotları (Roodbergen 2001a)

### 3. MATERYAL ve YÖNTEM

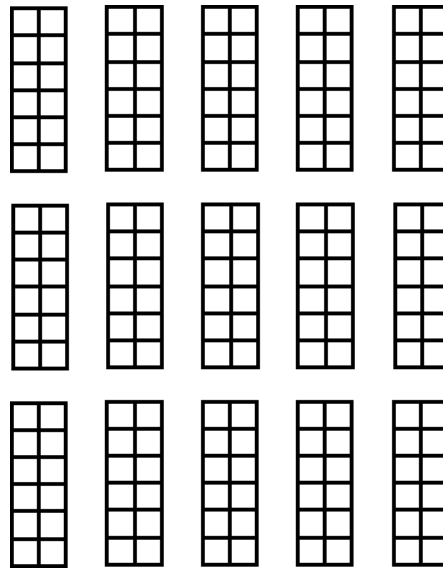
Çalışmanın bu bölümünde dikdörtgen şekilli yerleşim düzeni olan depo yapıları ve bu yapıdaki depolarda optimal ve sezgisel rotalama prosedürleri ile ilgili bilgiler aktarılacaktır. Bölümün devamında tez çalışmasına konu olan problem tanımlanarak problemin nasıl ele alınacağı üzerinde durulacaktır.

#### 3.1. Materyal

Bu alt bölümde tez çalışması kapsamında ele alınan problemin çözümünde kullanılacak teorik yapılar ve çözüm yöntemlerinin formülasyonları ile ilgili bilgiler açıklanacaktır.

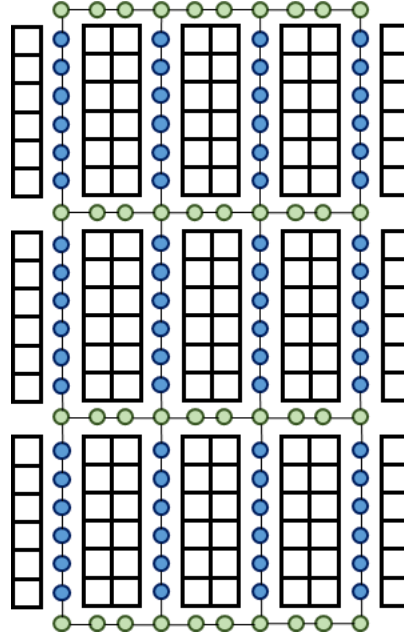
##### 3.1.1. Dikdörtgen Şekilli Depoların Ağ Gösterimi

Malzemelerin stoklanmasında kullanılan depolar çeşitli yerleşim düzenlerinde olabilmektedir. En yaygın kullanımı olan yapılarından biri dikdörtgen şekilli, kuşbakışı görünüme göre yatay ve dikey koridorlar ile ayrılmış depolardır (Karasek 2013). Şekil 3.1’de bu tanıma uyan basit bir deponun kroki görüntüsü verilmiştir.



Şekil 3.1. Dikdörtgen şekilli basit bir depo örneği

Geleneksel dikdörtgen yapıları bir depoda temel parametreler stok yerleri ve koridorlardır. Bu bilgiler kullanılarak deponun nasıl bir yapıda olduğunun bilgisi verilebilir. Depodaki koridor düğümlerinin ağ yapısındaki bağlantılı gösterimi Şekil 3.2’de görüldüğü gibidir. Koridor düğümlerinden birbirleri arasında hareket imkânı olan düğümler arasındaki bağlantılar gösterilmiştir. Bu şekilde bir gösterim deponun sayısal olarak modellenmesi ve iç lojistik problemlerinin çözülebilmesi adına önem arz etmektedir.



**Şekil 3.2.** Dikdörtgen şekilli depo yapılarında genel ağ gösterimi

Analiz edilen depo ağ yapısı Şekil 3.3’te daha küçük bir ölçekte gösterilmiştir. Buradan da görülebileceği gibi dikey koridorlarda bulunan koridor düğümleri komşu depolama yerleri ile sipariş toplama operasyonunun yapılması açısından ilişkilidir. Depolama yerleri için parametre tanımlaması yapıldığında bu ilişkinin de gösterilmesi problem çözümüne katkı sağlayacaktır.

Ayrıca Şekil 3.3’te depo tasarımını tanımlayan iki önemli parametre olan  $k_x$  ve  $k_y$  de gösterilmiştir. Burada  $k_x$ , yatay koridorlar arasındaki iki düğüm arası mesafeyi,  $k_y$  ise dikey koridor düğümleri arası mesafeyi ifade etmektedir.



Burada, Denklem 3.1 modelin amaç fonksiyonunu ifade etmektedir. Amaç fonksiyonu düğümler arası toplam hareket maliyetini en aza indirmeyi hedeflemektedir. Denklem 3.2 düğümler arası döngü oluşmama kısıtını ve Denklem 3.3 ise atama değerinin sıfırdan büyük olma kısıtını göstermektedir. Ele alınan problemin yapısına göre modelin amaç fonksiyonlarında veya kısıtlarında farklılıklar söz konusu olabilmektedir.

En kısa yol problemi, problemin durumuna göre çeşitli alt varyasyonlara ayrılmaktadır. Bunlar:

- Tek-eşli (single-pair) en kısa yol problemi
- Tek-kaynaklı (single-source) en kısa yol problemi
- Tek-hedefli (single-destination) en kısa yol problemi
- Tüm eşler (all-pairs) arası en kısa yol problemi

Tek-eşli en kısa yol problemi, belirli bir düğüm çifti arasındaki en kısa yolun hesaplandığı en kısa yol problemidir. Bu problemi çözmek adına en çok kullanılan algoritma A\* arama algoritmasıdır.

A\* arama algoritmasında Denklem 3.4'te verilen formülasyon temel alınarak hesaplamalar yapılmaktadır:

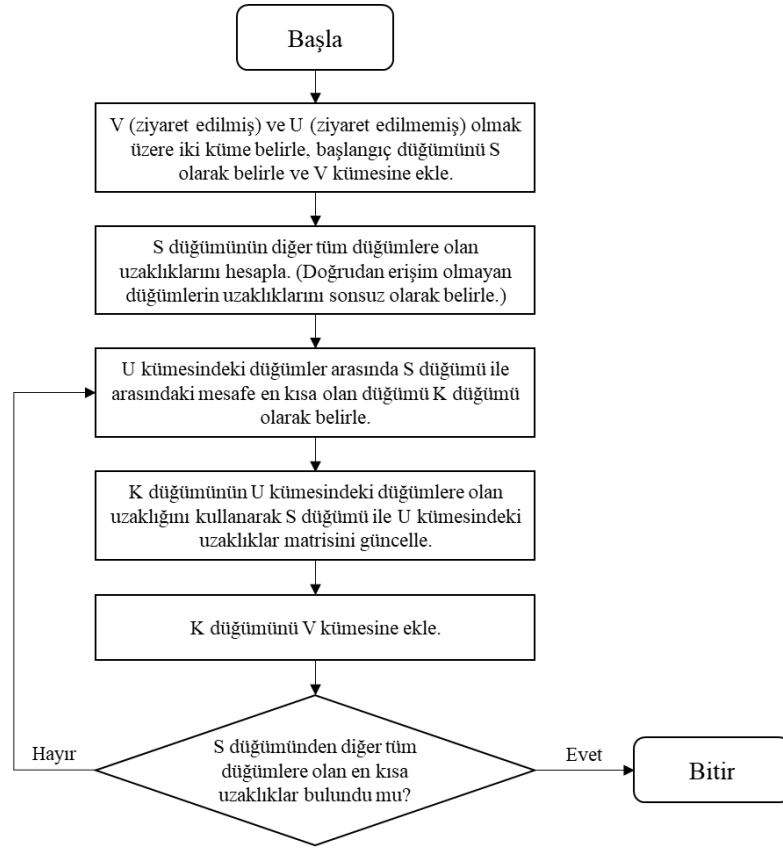
$$f(n) = g(n) + h(n) \quad (3.4)$$

Burada  $f(n)$  mesafenin hesaplandığı sezgisel fonksiyonu,  $g(n)$  başlangıç düğümünden mevcut düğüme gelmenin maliyetini,  $h(n)$  ise mevcut düğümden hedef düğüme varmak için tahmin edilen mesafeyi ifade etmektedir.  $f(n)$  fonksiyonunun sezgisel olma sebebi, içerisindeki  $h(n)$  fonksiyonunun tahmine dayalı sezgisel bir fonksiyon olmasıdır.

Algoritma, basit bir toplama işleminin tanımlanan öncelik sırası kuyruğuna göre iterasyonlar ile devam etmesi üzerine kuruludur. Algoritma her adımda en düşük maliyetli değeri olan düğümü seçer ve bu düğümü öncelik kuyruğundan çıkarır. Ardından seçilen bu düğüme komşu olan bütün düğümlerin değerleri güncellenir. Algoritma yukarıdaki adımları hedefe varana kadar veya kuyrukta başka düğüm kalmayana kadar tekrarlar.

Tek-kaynaklı en kısa yol problemi, belirli bir düğümden diğer tüm düğümlere giden en kısa yolun bulunması problemidir. Dijkstra algoritması ve Bellman Ford algoritması, tek-kaynaklı en kısa yol probleminin çözümünde kullanılan en popüler algoritmalarındandır.

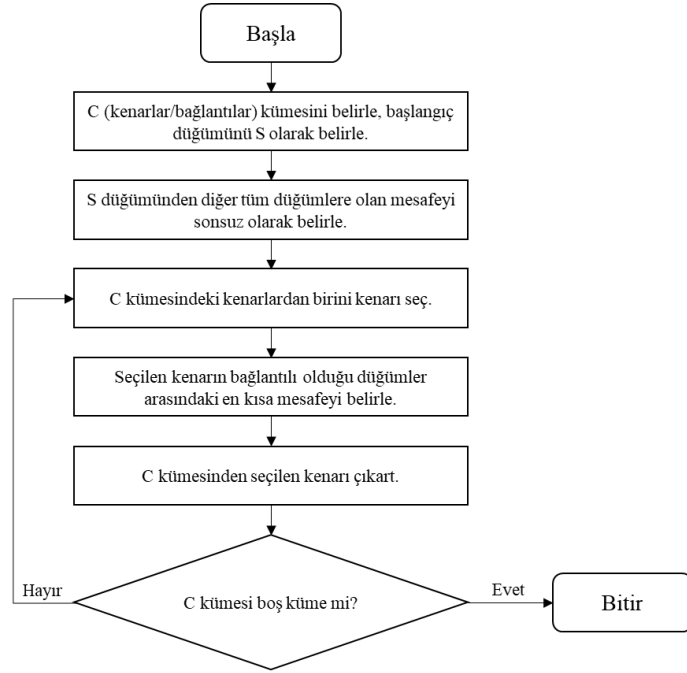
1959'da Edsger Dijkstra tarafından geliştirilen Dijkstra algoritması, düğümler arası bağlantıları negatif olmayan bir ağın tek kaynaklı en kısa yol problemini çözümünde kullanılmaktadır. Algoritma en kısa yolu belirlerken açgözlü yaklaşımı kullanır. Dijkstra algoritması için akış şeması Şekil 3.4'te verilmiştir. Algoritma, temelde bir başlangıç düğümü seçimi ile başlar ve diğer düğümlerin sırasıyla kontrol edilmesi ile devam eder. Algoritmanın karmaşıklığı,  $n$  düğüm sayısını belirtmek üzere  $O(n^2)$  şeklindedir.



Şekil 3.4. Dijkstra algoritmasının akış şeması

Belman-Ford algoritması da Dijkstra algoritması ile benzer şekilde bir düğümden diğer tüm düğümlere olan en kısa mesafenin bulunmasında kullanılmaktadır (Bellman 1958, Ford ve Fulkerson 1962). Belman-Ford algoritmasının Dijkstra algoritmasından farkı

düğümler arası bağlantıları (kenar ağırlıkları) negatif olabilen ağ yapılarında da çalışabilmesidir. Ayrıca Belman-Ford algoritması Dijkstra algoritması gibi açgözlü bir algoritma değildir. Belman-Ford algoritması için akış şeması Şekil 3.5'te verilmiştir. Algoritma, temel olarak bir kaynak düğümünden diğer düğümlere olan minimum uzaklıkları düğümler arası bağlantılar listesini kullanarak bulmaktadır. Bu algoritma için de  $n$  düğüm sayısı olmak üzere karmaşıklık  $O(n^2)$  şeklindedir.



**Şekil 3.5.** Belman-Ford algoritmasının akış şeması

Tek-hedefli en kısa yol problemi, diğer tüm düğümlerden tek bir hedef düğüme giden en kısa yolun bulunması problemidir. Dijkstra Algoritması aynı zamanda bu problemin çözümü için de uyarlanmış en önemli algoritmadır.

Son olarak tüm eşler arası en kısa yol problemi ağ üzerindeki tüm düğümlerin birbirleri arasındaki en kısa mesafenin bulunması problemidir. Floyd-Warshall algoritması (Floyd 1962, Warshall 1962) ve Johnson algoritması (Johnson 1977), bu problem türünün çözümünde yaygın olarak kullanılan algoritmalarıdır.

Bahsi geçen algoritmalarından, bu çalışma kapsamında da kullanılmış olan Floyd-Warshall algoritması, bir ağ yapısındaki tüm düğümlerin birbirleri ile arasındaki minimum mesafeyi bulmak amacıyla kullanılmaktadır. Algoritmanın temel hali iki düğüm arasındaki en kısa rotanın yol bilgisini vermese de küçük değişiklikler ile rotanın da bulunması mümkün hale gelmektedir. Algoritmanın karmaşıklığı  $n$  düğüm sayısı olmak üzere  $O(n^3)$  şeklindedir. Algoritmanın sözde kodu Şekil 3.6’da verilmiştir. Algoritma ilk olarak, bir uzaklık matrisinin oluşturulması ile başlar. Uzaklık matrisinin değerleri başlangıçta sonsuz olarak atanır. Her kenar bilgisi için uzaklık matrisi bir kez güncellenir, matrisin diyagonali 0’a eşitlenir. Her iki düğüm çifti için diğer düğümler kontrol edilerek daha kısa bir yolun varlığı döngüler ile araştırılır.

```

dist matrisi  $|V| \times |V|$  boyutunda minimum uzaklıkları gösteren
başlangıçta tüm değerleri  $\infty$  olan bir matris olsun.

for each edge (u, v) do // her kenar için
    dist[u][v]  $\leftarrow$  w(u, v) // (u, v) kenarının ağırlığı

for each vertex v do // her düğüm için
    dist[v][v]  $\leftarrow$  0

for k from 1 to |V|
    for i from 1 to |V|
        for j from 1 to |V|
            if dist[i][j] > dist[i][k] + dist[k][j]
                dist[i][j]  $\leftarrow$  dist[i][k] + dist[k][j]
            end if

```

**Şekil 3.6.** Floyd-Warshall algoritmasının sözde kod gösterimi

Tüm düğümler arası uzaklıkların bulunmasında kullanılan diğer bir algoritma Johnson algoritmasıdır. Çoğunlukla dağınık ve az sayıda düğüme sahip yönlü ağ yapılarında kullanılmaktadır. Johnson algoritması bu çeşitteki ağ yapılarında hızlı çözümler sağlamaktadır. Johnson algoritmasının sözde kod gösterimi Şekil 3.7’de görüldüğü gibidir.



Sözde kod yapısında;  $G$  üzerinde çalışılan ağ,  $G.V$  ağ üzerindeki düğüm kümesini,  $G.E$  ise ağ üzerindeki kenar kümesini göstermektedir. Johnson algoritmasında ilk olarak ağ yapısına diğer tüm düğümler ile arasındaki bağlantı sıfır olan sanal bir düğüm eklenir. Her düğüm için Belman-Ford algoritması çalıştırılır ve düğümlerin ağırlıkları belirlenir. Bir sonraki aşamada ağ üzerinde Dijkstra algoritması çalıştırılır. Düğümler arası mesafe matrisi bu adım sonunda hesaplanmış olur.

```

Johnson(G)
1.
 $G'.V = G.V + \{s\}$  olacak şekilde  $G'$  değişkeni oluştur,
 $G'.E = G.E + \{(s, u) \text{ for } u \text{ in } G.V\}$ , ve
ağırlık( $s, u$ ) = 0 for  $u$  in  $G.V$ 
2.
if Bellman-Ford( $s$ ) == False
    return "Ağ yapısında negatif ağırlıklı bir döngü var."
else:
    for düğüm  $v$  in  $G'.V$ :
         $h(v) = \text{dist}(s, v)$  Bellman-Ford ile hesaplanır.
    for bağlantı  $(u, v)$  in  $G'.E$ :
        ağırlık'( $u, v$ ) = ağırlık( $u, v$ ) +  $h(u) - h(v)$ 
3.
 $D$  = başlangıçta sonsuz değerli yeni uzaklık matrisi
for düğüm  $u$  in  $G.V$ :
     $\text{dist}'(u, v)$  for all  $v$  in  $G.V$ 'yi hesaplamak için Dijkstra( $G$ , ağırlık',  $u$ ) çalıştır.
    for each düğüm  $v$  in  $G.V$ :
         $D(u, v) = \text{dist}'(u, v) + h(v) - h(u)$ 
return  $D$ 

```

Şekil 3.7. Johnson algoritmasının sözde kod gösterimi

### 3.1.3. Gezgin Satıcı Problemi

Flood'un 1956'daki çalışmasında tanıttığı Gezgin Satıcı Problemi (GSP), bilgisayar bilimi ve yöneylem araştırmasında büyük önemi olan NP-zor karmaşıklıkta bir optimizasyon problemidir. Problem temel olarak, bir ağ yapısındaki tüm düğümlerin en az bir kez ziyaret edilerek başlangıç düğümüne geri dönen en kısa rotanın bulunmasıdır.

GSP, literatürde en yoğun çalışılan problemlerden biridir. Birçok optimizasyon yöntemi, bu problemi performans karşılaştırması yapmak amacıyla kullanılmaktadır. Problem her ne kadar hesaplama karmaşıklığı açısından zor olsa da çözüm için birçok sezgisel ve kesin algoritma bilinmektedir.

GSP'nin, tedarik zinciri rotalama problemlerinden talaşlı imalat maliyet minimizasyonuna, DNA dizilimi analizlerinden teleskop rotalamaya kadar çok çeşitli uygulamaları bulunmaktadır. Lojistik süreçlerinde başlangıç noktasına geri dönecek en düşük maliyetli rotanın bulunması zaman ve maliyet kazancı açısından işletmelere büyük faydalar sağlamaktadır. Talaşlı imalat süreçlerinde kesici uçların minimum mesafede hareket ederek işlemi tamamlaması kesici uç ömrünü uzatarak benzer şekilde maliyet kazancını sağlamaktadır. DNA'daki gen dizilimleri GSP kullanılarak analiz edilebilmekte ve bilimsel açıdan anlamlı sonuçların elde edilmesine öncülük etmektedir. Astronomide teleskopların görüntü yakalamak için kısıtlı süreleri olduğundan en kısa sürede istenen rotanın tamamlanması yeni keşiflerin yapılmasında büyük önem arz etmektedir.

GSP, bir tam sayılı doğrusal programlama modeli ile ifade edilebilir.  $x_{ij}$ ,  $i$  düğümünden  $j$  düğümüne hareket edilip edilmeyeceğini belirleyen ikili bir karar değişkeni ve  $N$  düğüm sayısı olmak üzere model aşağıdaki gibidir:

Denklem 3.5 problemin amaç fonksiyonunu belirtmektedir. Amaç toplam maliyeti minimize etmektir. Denklem 3.6 ve Denklem 3.7'da belirtilen kısıtlar  $i - j$  yolunun rotada kullanılıp kullanılmadığını belirlemektedir. Denklem 3.8 ve Denklem 3.9 ile sırasıyla rota içerisinde alt tur oluşması engellenmekte ve değişken tanımlı yapılmaktadır. Denklem 3.10'da ise karar değişkeni ikili değişken olarak tanımlanmıştır.

$$\min z = \sum_{i=1}^N \sum_{j=1}^N c_{ij} x_{ij} \quad (3.5)$$

s. t.

$$\sum_{i=1}^N x_{ij} = 1, \quad j = 1, \dots, N \quad (3.6)$$

$$\sum_{j=1}^N x_{ij} = 1, \quad i = 1, \dots, N \quad (3.7)$$

$$u_i - u_j + Nx_{ij} \leq N - 1, \quad \forall i, j \ i \neq j \quad (3.8)$$

$$u_j \geq 0, \quad \forall j \quad (3.9)$$

$$x_{ij} \in \{0,1\}, \quad \forall i, j \quad (3.10)$$

GSP ile en kısa yolu bulma problemi arasındaki temel fark, GSP'nin ağ üzerindeki her düğümün bir permütasyonunu içerecek şekilde en kısa yolu bulması, en kısa yolu bulma probleminin ise herhangi bir en kısa rotayı tercih etmesidir. GSP'nin çözümü, cevabının bir döngü oluşturmasını gerektirir. GSP'nin çözüm karmaşıklığı NP-zor iken en kısa yol problemine polinom zamanlarda çözüm bulunabilmektedir.

#### 3.1.4. Sezgisel Algoritmalar

Literatürde karmaşık problemleri çözmek amacıyla kullanılan algoritmalarından bazıları şunlardır (Dreo 2007):

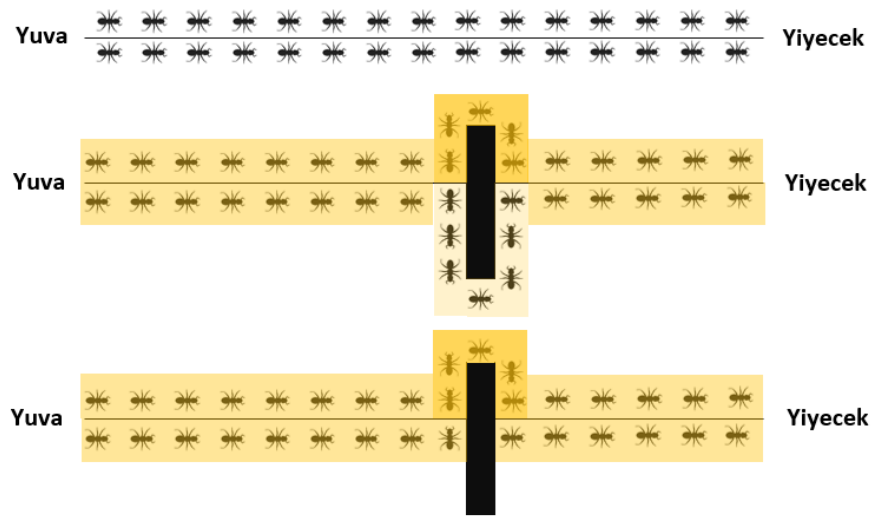
- Tavlama benzetimi
- Tabu arama
- Genetik algoritmalar
- Yapay sinir ağları
- Sürü zekâsı algoritmaları (karınca kolonisi, yapay arı kolonisi algoritmaları vb.)

Her sezgisel algoritma kendisine has temel kurallar içermekte ve algoritmayı kullanan kişi tarafından şekillendirilmektedir. Tavlama benzetimi, 1982 yılında Kirkpatrick vd. tarafından optimizasyon problemlerinin çözümünde kullanılmak üzere geliştirilen bir algoritmadır. Amaç problemin türüne göre optimizasyon yönünde iyileşmeyi sağlayacak parametreyi bulmak ve sonucu iyileştirmektir. İlk defa Glover tarafından 1986'da literatüre geçen tabu arama yaklaşımı, çözüme giderken belirli yönde ilerlemeyi yasaklayarak yerel optimumları aşmayı amaçlar. Genetik algoritmalarından ilk kez 1975'te Holland tarafından bahsedilmiştir. Bu algoritmalar temel olarak bir problem için birden fazla çözümü (popülasyon) oluşturmakta ve bu çözümleri oluştururken ve çoğaltırken

evrimsel yasaları kullanmaktadır. Yapay sinir ağıları da diğer algoritmalar gibi gerçek yaşamda insan beyninin bilgi işleme tekniğinden esinlenilerek oluşturulmuştur.

Sürü zekâsı algoritmaları da aynı şekilde doğadaki sürülerin hareketlerinden esinlenilerek oluşturulmuştur. Gerçek doğada sürüler, birbirleri ile etkileşim halinde bulunarak karşılaştıkları problemlere çözüm getirmektedir. En bilinen sürü algoritmaları karınca kolonisi ve yapay arı kolonisi algoritmalarıdır.

Dorigo vd. karınca kolonisi optimizasyon algoritmasını (KKO) ilk olarak 1990'ların başlarında tanıtmışlardır. Bu algoritmanın geliştirilmesi sürecinde, bilindiği üzere sosyal bir topluluk olan karınca kolonilerinin yaşam biçimlerinden ilham alınmıştır. Karıncalar kolonilerde birbirleri ile etkileşim halinde yaşarlar ve bireysel hedeflere göre değil, koloni hedeflerine göre hareket ederler. Karınca kolonisi optimizasyon algoritması, karıncaların yiyecek arama mekanizmasında bulunan besin kaynaklarına giden en kısa yolu bulma güdüsünü temel almaktadır. Karıncalar, başlangıçta rassal olarak farklı noktalara dağılmakta ve keşfettikleri bölgelerin özelliklerini feromonlar yardımıyla diğer karıncalara bildirmektedir. Karıncalar, hedef rotayı tayin ederken topluluktaki diğer karıncaların yoğun feromon bıraktığı noktaları seçme eğilimindedir. Yiyeceğe ulaşan karınca, yine aynı feromonları takip ederek yuvasına geri döner ve takip ettiği yola hedefteki yiyeceğin kalitesi ve miktarına göre feromon bırakmaya devam eder. Bu bilgi paylaşımı, diğer karıncaların da seçecekleri rotayı yönlendirmektedir.



**Sekil 3.8.** Karıncaların yuva-yiyecek arası en kısa mesafeyi bulması

Şekil 3.8’de karınca kolonisi optimizasyon algoritmasının temeli olan karıncaların yuva-yiyecek arası en kısa yolu bulma yöntemi gösterilmiştir.

Karınca kolonisi algoritmasında kullanılan parametreler Çizelge 3.1’deki gibidir.

**Çizelge 3.1.** Karınca kolonisi algoritması parametreleri

Parametre	Açıklama
$n$	Problemdeki düğüm sayısı
$m$	Sistemdeki karınca sayısı
$d_{ij}$	$i$ ve $j$ düğümleri arası mesafe
$t$	İterasyon sayısı
$T$	Maksimum iterasyon sayısı
$\rho$	Buharlaşma katsayısı
$\tau_{ij}$	$i$ ve $j$ düğümleri arası feromon yoğunluğu
$\eta_{ij}$	$i$ ve $j$ düğümleri arası görünürlük
$p_{ij}^k$	$k$ karıncasının $i$ düğümünden $j$ düğümüne geçme olasılığı,
$L^k(t)$	$k$ karıncasının $t$ iterasyonunda kat ettiği mesafe
$L_{eniyi}(t)$	$t$ iterasyonunda bulunan en kısa mesafe
$\alpha$	Feromonun önem katsayısı
$\beta$	Uzaklığın önem katsayısı
$Q$	Feromon güncelleme sabiti

Karınca kolonisi algoritması ile aşağıda belirtilen adımlar izlenerek optimum çözüme yaklaşılr:

1. Çözüm için temel parametreler ( $n, m, d_{ij}, \rho, \alpha, \beta, Q$ ) ve başlangıç feromon değerleri ( $\tau_{ij}$ ) belirlenir.
2. Karıncalar her düğüme rassal olarak yerleştirilir.
3. Her karınca, sonraki düğümü Denklem 3.11’de belirtilen lokal arama olasılığına bağlı olarak seçmek üzere turunu tamamlar.

$$p_{ij}^k = \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum [\tau_{ij}]^\alpha [\eta_{ij}]^\beta} \quad (3.11)$$

4. Sistemdeki her karınca için hareket mesafesi hesaplanır. Lokal feromon güncellemesi Denklem 3.12 ve 3.13’te belirtildiği şekilde yapılır.

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \sum_{k=1}^m \Delta\tau_{ij}^k(t+1) \quad (3.12)$$

$$\Delta\tau_{ij}^k(t+1) = \begin{cases} 1/L^k(t+1) & \text{eğer } k \text{ karıncası } i-j \text{ yolunu kullandıysa} \\ 0 & \text{diğer durumlarda} \end{cases} \quad (3.13)$$

5. En iyi çözüm hesaplanır ve global feromon güncellemesi Denklem 3.14 ve 3.15'te belirtildiği şekilde yapılır.

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \sum_{k=1}^m \Delta\tau_{ij}^k(t+1) \quad (3.14)$$

$$\Delta\tau_{ij}^k(t+1) = \begin{cases} 1/L_{eniyi}(t+1) & \text{eğer en iyi tura aitse} \\ 0 & \text{diğer durumlarda} \end{cases} \quad (3.15)$$

6. Maksimum iterasyon sayısı ( $T$ ) ya da belirlenen bir başka yeterlilik kriteri sağlanana kadar 2. Adım'dan devam edilir.

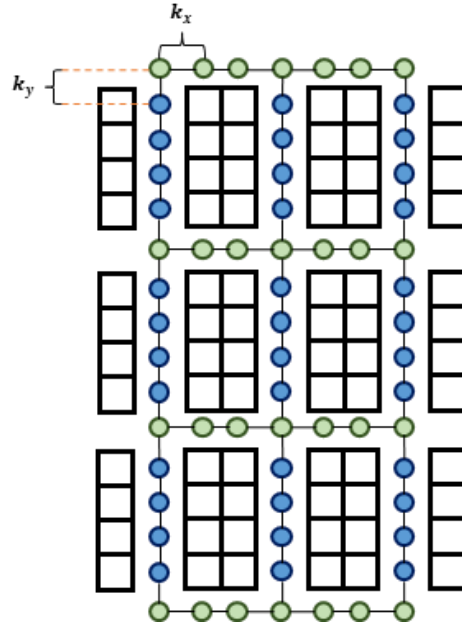
### 3.2. Yöntem

Bu bölümde, dikdörtgen şekilli depolarda en kısa rotanın bulunması için materyal başlığı altında genel çerçevesi verilen yöntemlerin bütünleşik halde kullanımını açıklanacaktır.

Tez kapsamında ele alınan problem, dikdörtgen şekilli depolarda gezgin satıcı problemine uygun şekilde sipariş toplama operasyonunun en kısa yoldan tamamlanmasıdır. Bu doğrultuda sipariş toplayıcı, toplama yapacağı depolama yerlerini en kısa yoldan ziyaret ederek rotasını tamamlamalı ve başlangıç noktasına topladığı malzemeleri geri getirmelidir.

#### 3.2.1. Örnek Deponun Ağ Gösterimi ile Tanımlanması

Problemin çözümüne ilk olarak mevcut sistem dinamikleri tanımlanarak başlanmalıdır. Dikdörtgen şekilli bir depoda, depolama yerleri ile ilişkili koridor düğümlerinin tanımlanması gerekmektedir. Şekil 3.9'da iki yatay koridor ve üç dikey koridor ile ayrılmış, bitişik depolama yeri sayısı dört olan basit bir dikdörtgen şekilli depo yapısı verilmiştir.



**Şekil 3.9.** İki yatay, üç dikey koridorlu; her sırada dört bitişik depolama yerinden oluşan dikdörtgen şekilli bir depo örneği

Depodaki uzaklık hesaplamalarının daha sistematik yapılabilmesi için Şekil 3.9’da ağ gösterimi verilen depo, Şekil 3.10’da görüleceği üzere iki boyutlu matris haline dönüştürülmüştür. Matris yardımıyla düğümler arası mesafeler kolaylıkla hesaplanabilmekte ve koridor düğümlerinin ilişkili olduğu depolama yerleri daha kolay bulunabilmektedir.

-	0	1	2	3	4	5	6	7	8
0		K	K	K	K	K	K	K	K
1	D	K	D	D	K	D	D	K	D
2	D	K	D	D	K	D	D	K	D
3	D	K	D	D	K	D	D	K	D
4	D	K	D	D	K	D	D	K	D
5		K	K	K	K	K	K	K	
6	D	K	D	D	K	D	D	K	D
7	D	K	D	D	K	D	D	K	D
8	D	K	D	D	K	D	D	K	D
9	D	K	D	D	K	D	D	K	D
10		K	K	K	K	K	K	K	
11	D	K	D	D	K	D	D	K	D
12	D	K	D	D	K	D	D	K	D
13	D	K	D	D	K	D	D	K	D
14	D	K	D	D	K	D	D	K	D
15		K	K	K	K	K	K	K	

-	0	1	2	3	4	5	6	7	8
0		1;0	2;0	3;0	4;0	5;0	6;0	7;0	
1	0;1	1;1	2;1	3;1	4;1	5;1	6;1	7;1	8;1
2	0;2	1;2	2;2	3;2	4;2	5;2	6;2	7;2	8;2
3	0;3	1;3	2;3	3;3	4;3	5;3	6;3	7;3	8;3
4	0;4	1;4	2;4	3;4	4;4	5;4	6;4	7;4	8;4
5		1;5	2;5	3;5	4;5	5;5	6;5	7;5	
6	0;6	1;6	2;6	3;6	4;6	5;6	6;6	7;6	8;6
7	0;7	1;7	2;7	3;7	4;7	5;7	6;7	7;7	8;7
8	0;8	1;8	2;8	3;8	4;8	5;8	6;8	7;8	8;8
9	0;9	1;9	2;9	3;9	4;9	5;9	6;9	7;9	8;9
10		1;10	2;10	3;10	4;10	5;10	6;10	7;10	
11	0;11	1;11	2;11	3;11	4;11	5;11	6;11	7;11	8;11
12	0;12	1;12	2;12	3;12	4;12	5;12	6;12	7;12	8;12
13	0;13	1;13	2;13	3;13	4;13	5;13	6;13	7;13	8;13
14	0;14	1;14	2;14	3;14	4;14	5;14	6;14	7;14	8;14
15		1;15	2;15	3;15	4;15	5;15	6;15	7;15	

**Şekil 3.10.** Deponun matris gösterimi

### 3.2.2. Örnek Depo için En Kısa Yol Hesaplamaları

Temel tanımlamaları yapılan depo için probleme giden yolda bir sonraki aşama, en kısa yol hesaplamalarının yapılacağı sistematiğin kurulmasıdır. Bu amaçla ilk olarak düğümler arası mesafe parametreleri tanımlanarak başlangıç uzaklık matrisi oluşturulmalıdır. Ardından en kısa yol algoritmalarından tüm düğümler arası uzaklıkların bulunabileceği bir algoritma seçilerek uzaklık matrisi problemin çözümünde kullanılmak üzere hazır hale getirilmelidir.

Verilen depo örneği için  $k_x = 2 m$ ,  $k_y = 1,5 m$  olsun. Bu parametreler yardımıyla koridor düğümleri arasındaki mesafeleri hesaplamak için öncelikle komşu koridor düğümleri arası mesafeler matrise işlenerek  $D_0$  uzaklık matrisi oluşturulmaktadır. Sonrasında tüm düğümlerin birbirleri ile aralarındaki en kısa mesafeyi bulmak için Floyd-Warshall algoritması uzaklık matrisine uygulanmaktadır.

Örnekteki deponun 16 koridor düğümü için düğümler arası uzaklık başlangıç matrisi  $D_0$  Şekil 3.11'de görüldüğü gibidir.  $D_0$  başlangıç matrisi üzerinde Floyd-Warshall algoritması iterasyonları uygulandığında tüm düğümler arası uzaklık matrisi elde edilecektir. Depo modellemesi yapıldıktan sonra düğümler arası mesafenin bir sefere mahsus hesaplanması daha sonra yapılacak hesaplamalar için yeterli olmaktadır.

D_0	1;0	1;1	1;2	1;3	1;4	1;5	2;0	2;5	4;0	4;1	4;2	4;3	4;4	4;5	3;0	3;5
1;0	0	$k_y$	$\infty$	$\infty$	$\infty$	$\infty$	$k_x$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
1;1	$k_y$	0	$k_y$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
1;2	$\infty$	$k_y$	0	$k_y$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
1;3	$\infty$	$\infty$	$k_y$	0	$k_y$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
1;4	$\infty$	$\infty$	$\infty$	$k_y$	0	$k_y$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
1;5	$\infty$	$\infty$	$\infty$	$\infty$	$k_y$	0	$\infty$	$k_x$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
2;0	$k_x$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$k_x$
2;5	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$k_x$	$\infty$	0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$k_x$
4;0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	0	$k_y$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$k_x$
4;1	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$k_y$	0	$k_y$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
4;2	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$k_y$	$k_y$	0	$k_y$	$\infty$	$\infty$	$\infty$	$\infty$
4;3	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$k_y$	0	$k_y$	$\infty$	$\infty$	$\infty$	$\infty$
4;4	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	0	$k_y$	$\infty$	$\infty$	$\infty$
4;5	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	0	$k_y$	$\infty$	$k_x$
3;0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$k_x$	$k_x$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	0	$\infty$
3;5	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$k_x$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$k_x$	$\infty$	0

Şekil 3.11. 16 koridor düğümü için Floyd-Warshall başlangıç matrisi



$D_0$  matrisine Floyd-Warshall algoritması uygulandığında Şekil 3.12'deki  $D$  matrisi elde edilmiş olur.  $D$  matrisi kullanılarak herhangi iki koridor düğümü arasındaki en kısa mesafe ve bu en kısa mesafenin hangi düğümler ziyaret edilerek kat edilebileceği bulunabilir.

D	1;0	1;1	1;2	1;3	1;4	1;5	2;0	2;5	4;0	4;1	4;2	4;3	4;4	4;5	3;0	3;5
1;0	0,0	1,5	3,0	4,5	6,0	7,5	2,0	9,5	6,0	7,5	9,0	10,5	12,0	13,5	4,0	11,5
1;1	1,5	0,0	1,5	3,0	4,5	6,0	3,5	8,0	7,5	9,0	10,5	12,0	13,5	12,0	5,5	10,0
1;2	3,0	1,5	0,0	1,5	3,0	4,5	5,0	6,5	9,0	10,5	12,0	13,5	12,0	10,5	7,0	8,5
1;3	4,5	3,0	1,5	0,0	1,5	3,0	6,5	5,0	10,5	12,0	13,5	12,0	10,5	9,0	8,5	7,0
1;4	6,0	4,5	3,0	1,5	0,0	1,5	8,0	3,5	12,0	13,5	12,0	10,5	9,0	7,5	10,0	5,5
1;5	7,5	6,0	4,5	3,0	1,5	0,0	9,5	2,0	13,5	12,0	10,5	9,0	7,5	6,0	11,5	4,0
2;0	2,0	3,5	5,0	6,5	8,0	9,5	0,0	11,5	4,0	5,5	7,0	8,5	10,0	11,5	2,0	11,5
2;5	9,5	8,0	6,5	5,0	3,5	2,0	11,5	0,0	11,5	10,0	8,5	7,0	5,5	4,0	13,5	2,0
4;0	6,0	7,5	9,0	10,5	12,0	13,5	4,0	11,5	0,0	1,5	3,0	4,5	6,0	7,5	2,0	9,5
4;1	7,5	9,0	10,5	12,0	13,5	12,0	5,5	10,0	1,5	0,0	1,5	3,0	4,5	6,0	3,5	8,0
4;2	9,0	10,5	12,0	13,5	12,0	10,5	7,0	8,5	3,0	1,5	0,0	1,5	3,0	4,5	5,0	6,5
4;3	10,5	12,0	13,5	12,0	10,5	9,0	8,5	7,0	4,5	3,0	1,5	0,0	1,5	3,0	6,5	5,0
4;4	12,0	13,5	12,0	10,5	9,0	7,5	10,0	5,5	6,0	4,5	3,0	1,5	0,0	1,5	8,0	3,5
4;5	13,5	12,0	10,5	9,0	7,5	6,0	11,5	4,0	7,5	6,0	4,5	3,0	1,5	0,0	9,5	2,0
3;0	4,0	5,5	7,0	8,5	10,0	11,5	2,0	13,5	2,0	3,5	5,0	6,5	8,0	9,5	0,0	11,5
3;5	11,5	10,0	8,5	7,0	5,5	4,0	11,5	2,0	9,5	8,0	6,5	5,0	3,5	2,0	11,5	0,0

Şekil 3.12. 16 koridor düğümü için Floyd-Warshall uygulanmış uzaklık matrisi

Şekil 3.12'deki matrisini yorumlayabilmek adına Floyd-Warshall hesaplamaları esnasında düğümlere giden en kısa yolun hangi yol üzerinden gidileceği bilgisini de saklamak gerekmektedir. Şekil 3.13'teki  $P$  matrisinde yukarıdaki 16 düğüme giden en kısa yolların hangi düğüm üzerinden geçtiği görülmektedir.

P	1;0	1;1	1;2	1;3	1;4	1;5	2;0	2;5	4;0	4;1	4;2	4;3	4;4	4;5	3;0	3;5
1;0	-	1;0	1;1	1;2	1;3	1;4	1;0	1;5	3;0	4;0	4;1	4;2	4;3	4;4	2;0	2;5
1;1	1;1	-	1;1	1;2	1;3	1;4	1;0	1;5	3;0	4;0	4;1	4;2	4;3	3;5	2;0	2;5
1;2	1;1	1;2	-	1;2	1;3	1;4	1;0	1;5	3;0	4;0	4;1	4;4	4;5	3;5	2;0	2;5
1;3	1;1	1;2	1;3	-	1;3	1;4	1;0	1;5	3;0	4;0	4;1	4;4	4;5	3;5	2;0	2;5
1;4	1;1	1;2	1;3	1;4	-	1;4	1;0	1;5	3;0	4;2	4;3	4;4	4;5	3;5	2;0	2;5
1;5	1;1	1;2	1;3	1;4	1;5	-	1;0	1;5	3;0	4;2	4;3	4;4	4;5	3;5	2;0	2;5
2;0	2;0	1;0	1;1	1;2	1;3	1;4	-	1;5	3;0	4;0	4;1	4;2	4;3	4;4	2;0	2;5
2;5	1;1	1;2	1;3	1;4	1;5	2;5	1;0	-	4;1	4;2	4;3	4;4	4;5	3;5	2;0	2;5
4;0	2;0	1;0	1;1	1;2	1;3	1;4	3;0	3;5	-	4;0	4;1	4;2	4;3	4;4	4;0	4;5
4;1	2;0	1;0	1;1	1;2	1;3	2;5	3;0	3;5	4;1	-	4;1	4;2	4;3	4;4	4;0	4;5
4;2	2;0	1;0	1;1	1;2	1;5	2;5	3;0	3;5	4;1	4;2	-	4;2	4;3	4;4	4;0	4;5
4;3	2;0	1;0	1;1	1;4	1;5	2;5	3;0	3;5	4;1	4;2	4;3	-	4;3	4;4	4;0	4;5
4;4	2;0	1;0	1;3	1;4	1;5	2;5	3;0	3;5	4;1	4;2	4;3	4;4	-	4;4	4;0	4;5
4;5	2;0	3;5	1;3	1;4	1;5	2;5	3;0	3;5	4;1	4;2	4;3	4;4	4;5	-	4;0	4;5
3;0	2;0	1;0	1;1	1;2	1;3	1;4	3;0	3;5	3;0	4;0	4;1	4;2	4;3	4;4	-	4;5
3;5	1;1	1;2	1;3	1;4	1;5	2;5	3;0	3;5	4;1	4;2	4;3	4;4	4;5	3;5	4;0	-

Şekil 3.13. 16 koridor düğümü için Floyd-Warshall uygulanmış öncelik matrisi

Örnek bir hesaplama “1;1” düğümünden “4;4” düğümüne yapılabilir. Burada  $D$  matrisinden gelen en kısa yol uzunluğu 13,5 m olarak bulunur. En kısa yolun rotasını belirlemek için ise  $P$  matrisi kullanılır.  $P$  matrisinin değerlendirilerek rotanın bulunma adımları Şekil 3.14’te gösterilmiştir.

<b>Başlangıç</b>	<b>Bitiş</b>	<b>Kullanılan Yol</b>
1;1	4;4	4;3
1;1	4;3	4;2
1;1	4;2	4;1
1;1	4;1	4;0
1;1	4;0	3;0
1;1	3;0	2;0
1;1	2;0	1;0
<b>1;1</b>	<b>1;0</b>	<b>1;1</b>

1;1 → 1;0 → 2;0 → 3;0 → 4;0 → 4;1 → 4;2 → 4;3 → 4;4

**Şekil 3.14.** 1;1 düğümünden 4;4 düğümüne giden en kısa rotanın bulunması

### 3.2.3. Depo için Gezgin Satıcı Problemi Kurgusunun Oluşturulması

Arka plandaki tüm sistem dinamikleri oluşturulduktan sonra depodaki rotalama probleminin çözümü için literatürdeki gezgin satıcı problemi kullanılarak çözüm mekanizması kurgulanmıştır. Ele alınan problem, gezgin satıcı probleminde olduğu gibi ağ üzerindeki seçili tüm düğümler ziyaret edilerek başlangıç düğümüne en kısa rota izlenerek geri dönülmesidir.

Problem için matematiksel programlama modeli daha önce sunulan gezgin satıcı problemi ile yapısal olarak aynı olup aşağıdaki gibi tekrar sunulmuştur. Amaç fonksiyonu Denklem 3.16’da verildiği gibi,  $x_{ij}$   $i - j$  yolunun kullanılıp kullanılmaması ve  $d_{ij}$   $i - j$  düğümleri arası mesafeyi ifade etmek üzere, kat edilen toplam mesafenin minimize edilmesi üzerine kurulmuştur. Denklem 3.17 ve Denklem 3.18 kısıtları ile rotadaki kullanılacak yollar belirlenmektedir. Denklem 3.19 ve Denklem 3.20 kısıtları ile rota içerisinde alt tur oluşması engellenmiş ve ilgili değişken tanımları yapılmıştır. Denklem 3.21 karar değişkeninin ikili değişken olarak tanımlanması amacıyla kullanılmıştır.

$$\min z = \sum_{i=1}^N \sum_{j=1}^N d_{ij} x_{ij} \quad (3.16)$$

s. t.

$$\sum_{i=1}^N x_{ij} = 1, \quad j = 1, \dots, N \quad (3.17)$$

$$\sum_{j=1}^N x_{ij} = 1, \quad i = 1, \dots, N \quad (3.18)$$

$$u_i - u_j + N x_{ij} \leq N - 1, \quad \forall i, j \ i \neq j \quad (3.19)$$

$$u_j \geq 0, \quad \forall j \quad (3.20)$$

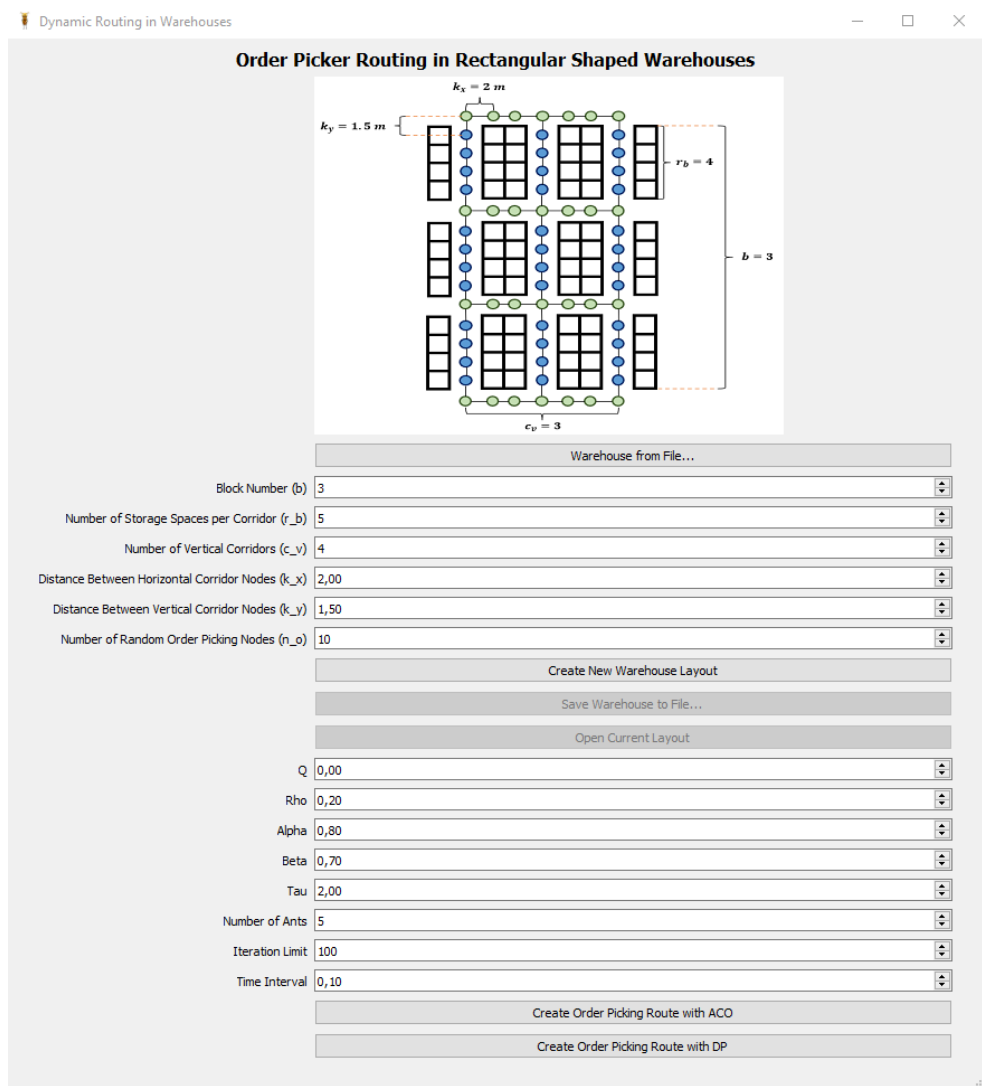
$$x_{ij} \in \{0,1\}, \quad \forall i, j \quad (3.21)$$

Problemin çözümü için yukarıda verilen matematiksel programlama modeli kullanılmaktadır. Modelin kullanımı öncesinde depo parametreleri tanımlanarak deponun ağ ve matris gösterimi yapılmalıdır. Bu gösterimler yapılırken koridor düğümleri ile depolama yerleri arasındaki ilişkiler tanımlanmalıdır. Sonraki aşamalardaki hesaplamalar için tüm koridor düğümleri arasındaki en kısa mesafe ve rotalar oluşturulacak çözüm yöntemlerinde kullanılmak üzere hazır hale getirilmelidir.

### 3.2.4. Problem için Karınca Kolonisi Optimizasyonu Uygulaması

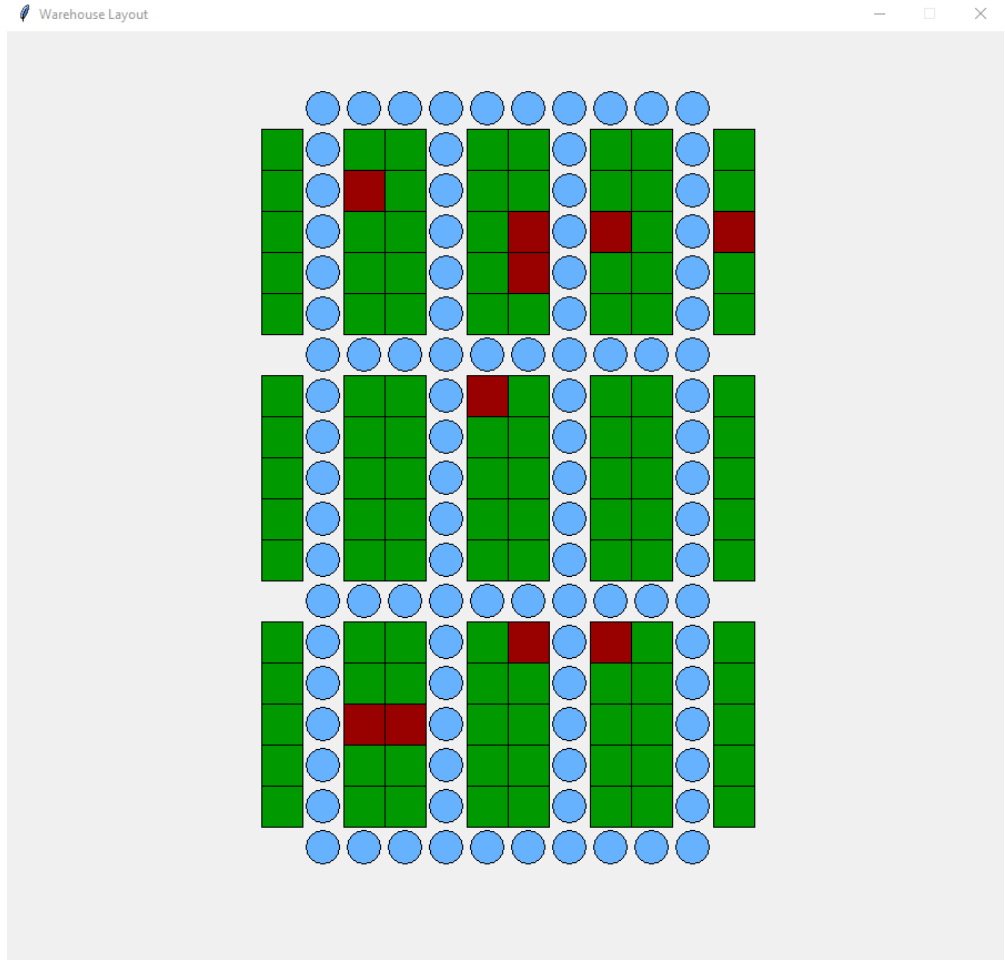
Tez çalışması kapsamında önceki bölümlerde tanıtılan ve örnekleri verilen yöntemleri entegre olarak kullanan bir yazılım Python programlama dili ve bu dilin açık kaynak kodlu çeşitli kütüphaneleri kullanılarak geliştirilmiştir. Hazırlanan yazılımın kullanıcıdan aldığı depo parametrelerine göre dinamik bir depo yapısı oluşturması ve problemin çözümünde kullanılacak matrisleri hesaplaması hedeflenmiştir. Ayrıca hazırlanan çalışmada oluşturulan depo yapısı kullanıcıya görsel olarak sunulmakta ve seçilen depolama noktalarını minimum maliyet ile gezecek gezgin satıcı problemi çözümü de simüle edilerek kullanıcıya verilmektedir.

Oluşturulan yazılımın kullanıcı arayüzü Şekil 3.15'te verilmiştir. Arayüzde depoyu oluşturan parametrelerin değiştirilebileceği alanlar, bu parametrelere göre depo yapısının oluşturulması için bir buton, oluşturulan depo yapısının sonraki çalışmalarda kullanılabilmesi için dosyaya kaydedilmesini sağlayan bir buton, kaydedilen depo yapılarının açılabilmesi için bir buton ve seçilen depo yapısının krokisinin kontrol edilebilmesi için bir buton olmak üzere çeşitli bileşenler bulunmaktadır. Ayrıca çalışma kapsamında KKO algoritması ile rotanın oluşturulabilmesi için algoritma parametrelerinin girilebileceği bölümler ve rotanın oluşturularak kullanıcıya görsel olarak sunulmasını sağlayan butonlar da bulunmaktadır.



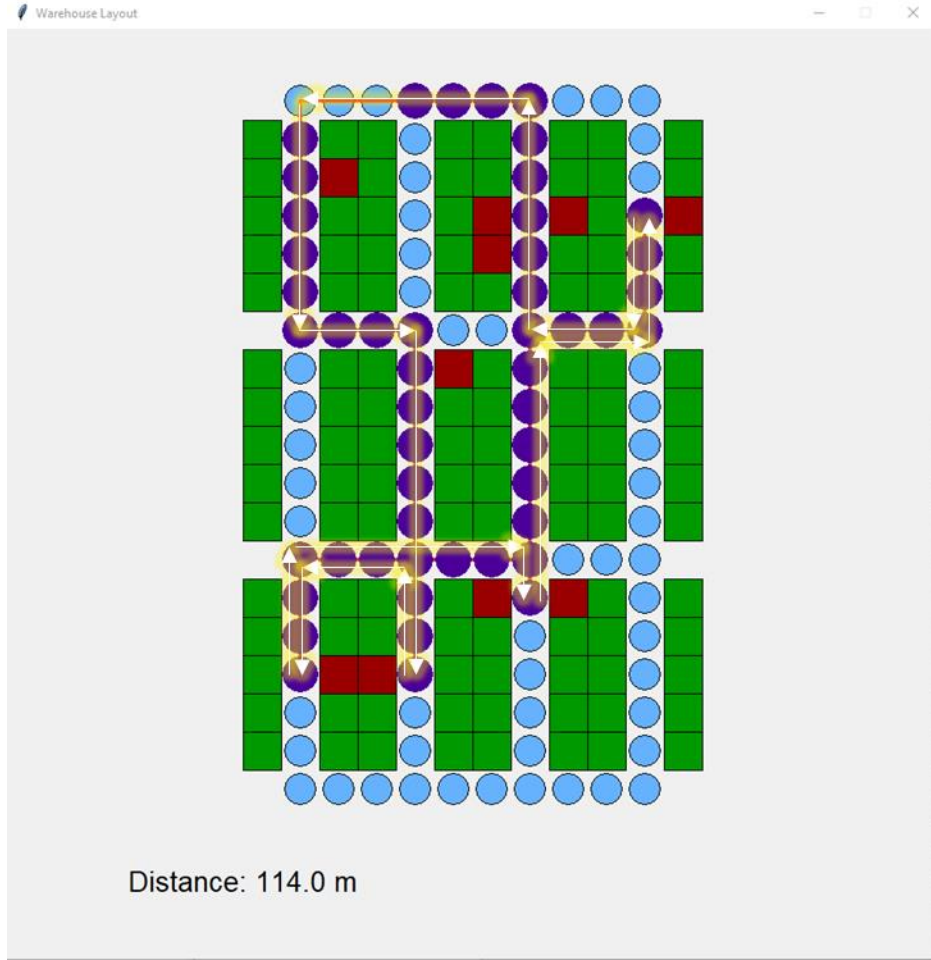
Şekil 3.15. Oluşturulan yazılımın kullanıcı arayüzü

“Open Current Layout” butonu yardımıyla parametrelere göre en baştan oluşturulan ya da daha önce dosyaya kaydedilip programda açılan depo yapısı görsel olarak kullanıcıya sunulmaktadır. Girilen parametrelere göre çizimi yapılan depo yapısı örneği Şekil 3.16’da görülmektedir. Şekilde mavi daireler koridor düğümlerini, yeşil ve kırmızı kareler depolama yerlerini, depolama yerleri arasındaki kırmızı kareler ise program tarafından seçilen rassal depolama yerlerini göstermektedir.



**Şekil 3.16.** Örnek depo yapısının program tarafından çizilmesi

“Create Order Picking Route with ACO” butonu ile programa girilen karınca kolonisi optimizasyon parametreleri dikkate alınarak sezgisel GSP çözümü bulunmakta ve bulunan rotanın hangi düğümler üzerinde hareket edilerek oluşturulduğu kullanıcıya adım adım gösterilmektedir. Şekil 3.17’de karınca kolonisi optimizasyon algoritması kullanılarak rota çiziminin tamamlanmış hali görülmektedir.



**Şekil 3.17.** Örnek depo yapısı üzerinden rotalama

Bulgular ve tartışma bölümünde her iki yöntem ile elde edilen çözümler karşılaştırılarak çözüm kaliteleri analiz edilecektir.

#### 4. BULGULAR ve TARTIŞMA

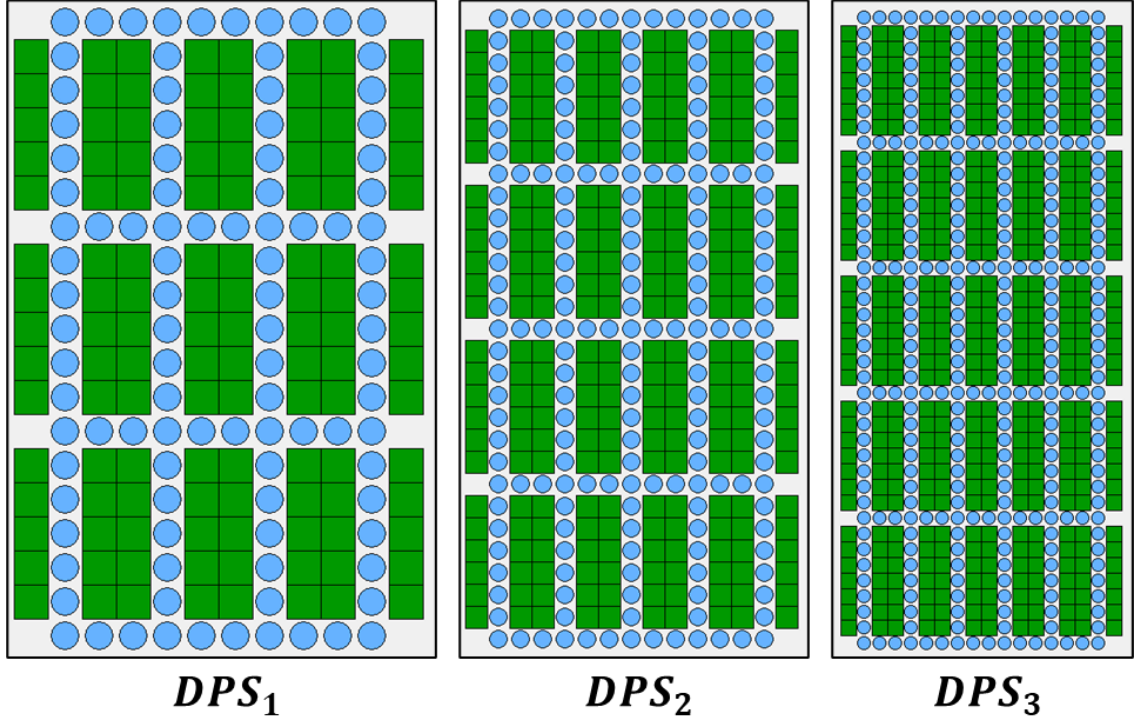
Çalışmanın bu bölümünde, materyal ve yöntem bölümünde metodolojisi verilen dikdörtgen şekilli depolarda rotalama prosedürünün farklı depo boyutları ve toplama noktası sayılarına göre uygulama örnekleri incelenecektir. Çözümler, önceki bölümde tez çalışması kapsamında Python programlama dili ile geliştirildiği belirtilen yazılım arayüzü kullanılarak elde edilecektir. Çözüm sonuçları, Intel Core i7-7700HQ 2.80GHz işlemci ve 16 GB RAM'e sahip Windows 10 işletim sistemli bir bilgisayar yardımıyla hesaplanmıştır.

##### 4.1. Depo Boyutlarının Değiştirilmesi

İlk olarak depo boyutları, Çizelge 4.1'de verilen parametrelere göre değiştirilerek analiz edilecektir. Deponun gösteriminde oluşan koridor düğümü sayıları ve depolama yeri sayıları ve bu depo kurgularında koridor düğümleri arası Floyd-Warshall matrisi hesaplama süreleri yine Çizelge 4.1'de belirtilmiştir. Ayrıca oluşan depo yapıları Şekil 4.1'de gösterilmiştir.

Çizelge 4.1. İncelenecek depo parametre setleri

Depolama Parametre Seti	$DPS_1$	$DPS_2$	$DPS_3$
Blok Sayısı ( $b$ )	3	4	5
Koridor Tarafı Başına Depolama Yeri Sayısı ( $r_b$ )	5	6	7
Dikey Koridor Sayısı ( $c_v$ )	4	5	6
Koridor Düğümü Sayısı	100	185	306
Depolama Yeri Sayısı	120	240	420
FW Matris Hesaplama Süresi (s)	0,48	2,77	13,44



**Şekil 4.1.** Depo parametre setlerine göre oluşan depo yapıları

İncelenen depo parametre setlerinde;  $b$ ,  $r_b$  ve  $c_v$  değerleri arttıkça deponun büyüklüğünün ve koridor düğüm sayısının arttığı görülmektedir. Depo parametrelerine göre koridor düğümü sayısı Denklem 4.1’de, depolama yeri sayısı Denklem 4.2’de verilmiştir. Floyd-Warshall algoritmasının hesaplama karmaşıklığı  $O(n^3)$  kaynaklı hesaplama süresi, koridor düğüm sayısının artışının küpü ile doğru orantılı olarak artmaktadır.

$$\text{Koridor Düğümü Sayısı} = (b + 1) \times (3c_v - 2) + b \times r_b \times c_v \quad (4.1)$$

$$\text{Depolama Yeri Sayısı} = 2 \times b \times r_b \times c_v \quad (4.2)$$

Gerçek yaşamda daha büyük boyutlu depolarla karşılaşmak mümkündür. Büyük boyutlu bir depo için yalnızca bir kez Floyd-Warshall algoritması çalıştırılarak uzaklık ve öncül düğüm matrislerinin bulunması ve oluşan uzaklık ve öncül matrislerinin daha sonra kullanılmak üzere dosyaya kaydedilmesi kullanıcı açısından büyük avantaj sağlayacaktır.



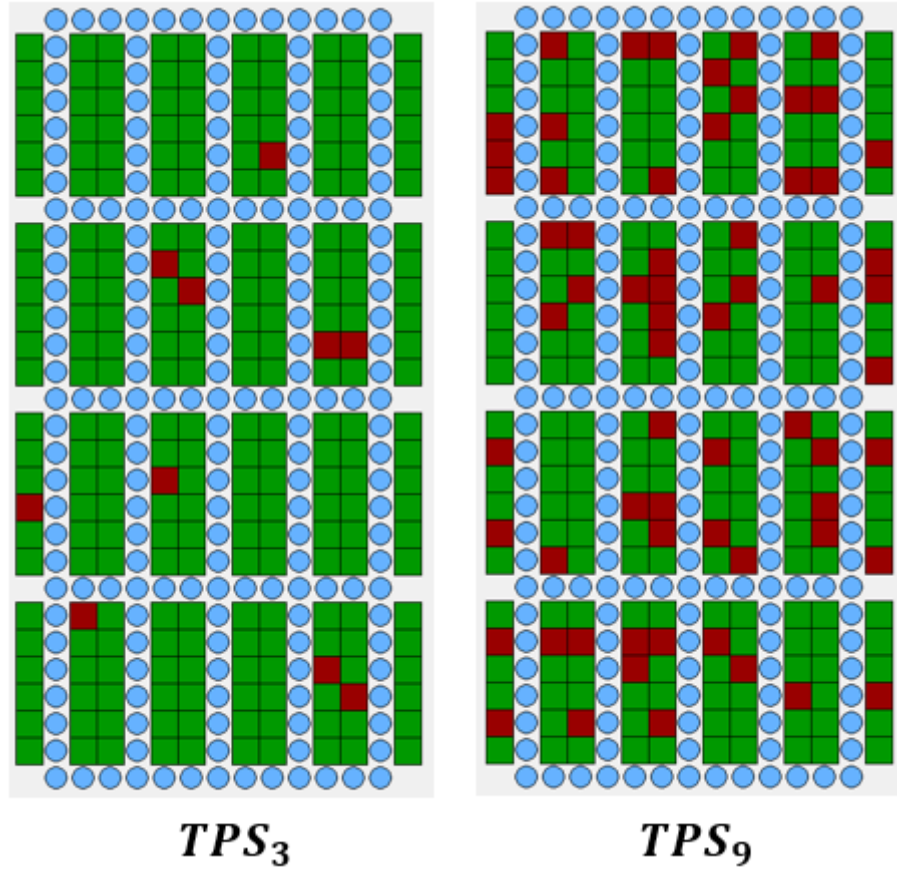
## 4.2. Toplama Yeri Sayısının Değiştirilmesi

Çalışma kapsamında incelenecek ikinci durum toplama yeri sayısının değiştirilmesidir. Bir önceki bölümde, depo parametreleri deponun yapısıyla ilgili olduğundan değişmesi daha zor bir parametre seti iken, toplama operasyonuna başlanılan her anda toplama yerleri sipariş durumuna göre değişecektir. Bu durum, toplama yeri sayısı parametresini inceleme açısından daha önemli hale getirmektedir.

Toplama yeri sayısı değiştirilirken depolama yeri sabit olarak bir önceki bölümde oluşturulan  $DPS_2$  kullanılacaktır.  $DPS_2$ , 240 depolama yerinden oluşan bir depoyu ifade etmekte olup rota oluşturabilmek için seçilebilecek toplama yeri sayısı ( $n_o$ )  $1 \leq n_o \leq 240$  şeklinde olacaktır. Çizelge 4.2'de seçilebilecek toplama yeri sayısı değişiminin sonuçları gösterilmiştir.

**Çizelge 4.2.** Toplama yeri sayısı değişiminin incelenmesi

Toplama Yeri Parametre Seti	$TPS_1$	$TPS_2$	$TPS_3$	$TPS_4$	$TPS_5$	$TPS_6$	$TPS_7$	$TPS_8$	$TPS_9$	$TPS_{10}$
Seçilen Rassal Depolama Yeri Sayısı	1	5	10	20	30	40	50	60	70	80
Seçilen Toplama Yerlerinin Toplam Depolama Yeri Sayısına Oranı (%)	0,42	2,08	4,17	8,33	12,50	16,67	20,83	25,00	29,17	33,33
KKO ile Bulunan Rota Uzunluğu (m)	24	117	189	195	258	297	351	375	360	414
KKO Çözüm Oluşturma Süresi (s)	0,01	0,03	0,11	0,26	0,36	0,64	0,76	0,66	0,81	1,3
Koridor Dolaşma Oranı (%)	6,06	29,55	47,73	49,24	65,15	75,00	88,64	94,70	90,91	104,55



**Şekil 4.2.** Toplama yeri parametre setlerine göre oluşan iki örnek depo

Toplama yeri sayısı 1'den 80'e kadar farklı 10 parametre seti kullanılarak değiştirilmiştir. Şekil 4.2'de bu 10 parametre setinden iki örnek gösterilmiştir. Parametre değişimleri ile toplama yeri sayısı arttıkça karınca kolonisi optimizasyonu parametreleri sabit tutulduğundan rota uzunluğunda artış gözlenmiştir. Toplama yeri seçimindeki ve algoritmadaki rassallık da göz önünde bulundurulduğunda,  $TPS_9$  parametre seti ile algoritmanın bulduğu rota uzunluğunun direkt olarak toplama yeri sayısı ile bağlantılı olmayacağı görülmüştür. Karınca kolonisi optimizasyon algoritmasının çözüm süresi toplama yeri sayısı arttıkça düzenli olarak artmıştır.

Koridor dolaşma oranı, algoritma sonucunda bulunan rota uzunluğunun depodaki tüm koridorlar dolaşarak başlangıç düğümüne tekrar geri dönülmesi durumunda kat edilecek mesafeye oranını göstermektedir.  $DPS_2$ 'de tüm koridorlar dolaşarak kat edilecek mesafe 396 metre olarak bulunmuştur. Koridor dolaşma oranını incelediğimizde, toplama yeri sayısı arttıkça algoritma sonucu oluşan rota uzunluğunun olası en yüksek kat edilecek

mesafeye ulaşarak bu mesafeyi dahi geçebildiği görülmüştür. Sonuçlar incelendiğinde, seçilen toplama yeri sayısı tüm depolama yerlerinin %20'sini aştığında depodaki toplam kat edilebilecek mesafenin %88'inden daha fazlası kat edildiğinden herhangi bir sezgisel optimizasyon uygulamadan tüm koridorların dolaşarak malzemelerin toplanmasının daha akıllıca olacağı sonucu çıkartılabilir. Diğer yandan toplama yeri sayısı arttığında karınca kolonisi optimizasyon parametrelerinde ayarlamalar yapılarak algoritma başarısında artış sağlanabilir.

### 4.3. Karınca Kolonisi Optimizasyon Parametrelerinin Değiştirilmesi

Çalışma kapsamında incelenecek üçüncü ve son durum karınca kolonisi optimizasyon algoritması parametrelerinin değiştirilmesidir. Önceki bölümlerde üzerinde çalışılan depo parametre setlerinden  $DPS_2$  (240 depolama yeri) ve toplama yeri parametre setlerinden  $TPS_4$  (20 toplama yeri seçimi) kullanılarak karınca kolonisi optimizasyon parametrelerinin değişimi analiz edilecektir.  $TPS_4$  bir önceki bölümde toplama noktası sayısı artarken koridor dolaşma oranının en az arttığı parametre olduğundan seçilmiştir.  $i$  algoritmayı çalıştırma sırası olmak üzere  $R_i$  algoritma çalıştırmayı ifade etmektedir.

**Çizelge 4.3.** Karınca sayısı değişiminin incelenmesi

Karınca Sayısı	5		10		15	
	Rota Uzunluğu (m)	Çözüm Süresi (s)	Rota Uzunluğu (m)	Çözüm Süresi (s)	Rota Uzunluğu (m)	Çözüm Süresi (s)
$R_1$	249	0,16	219	0,34	219	0,41
$R_2$	213	0,32	198	0,65	201	0,52
$R_3$	240	0,18	204	0,66	195	0,48
$R_4$	252	0,28	207	0,47	210	0,43
$R_5$	228	0,18	192	0,33	228	0,47
$R_6$	231	0,33	219	0,35	198	0,45
$R_7$	213	0,32	204	0,31	219	0,43
$R_8$	234	0,20	204	0,39	216	0,41
$R_9$	219	0,19	216	0,30	225	0,49
$R_{10}$	252	0,22	201	0,30	222	0,46
<b>En İyi Çözüm</b>	213	0,16	192	0,30	195	0,41
<b>En Kötü Çözüm</b>	252	0,33	219	0,66	228	0,52
<b>Standart Sapma</b>	15,10	0,07	9,03	0,14	11,70	0,04
<b>Ortalama</b>	233,10	0,24	206,40	0,41	213,30	0,46

İlk olarak Çizelge 4.3'te görüleceği üzere diğer parametreler sabit tutularak ( $\rho = 0.25$ ,  $\tau = 0.25$ ,  $Q = 0.1$ ,  $\alpha = 0.8$ ,  $\beta = 0.7$ ,  $T = 100$ ) karınca sayısı değişiminin algoritma sonucu bulunan rota uzunluğu ve algoritma çalıştırma süresine etkisi incelenmiştir. İnceleme sonucunda, seçilen üç parametre arasında karınca sayısının 10 olarak belirlenmesinin ortalama çözüm sonuçları değerlendirildiğinde en başarılı sonuçların elde edildiği görülmüştür. Ayrıca karınca sayısının artmasının algoritma çözüm süresini artırması da diğer bir sonuçtur.

**Çizelge 4.4.** Buharlaşma katsayısı değişiminin incelenmesi

Buharlaşma Katsayısı ( $\rho$ )	0,25		0,50		0,75	
Algoritma Çalıştırma	Rota Uzunluğu (m)	Çözüm Süresi (s)	Rota Uzunluğu (m)	Çözüm Süresi (s)	Rota Uzunluğu (m)	Çözüm Süresi (s)
$R_1$	228	0,28	219	0,17	297	0,65
$R_2$	222	0,25	243	0,16	267	0,14
$R_3$	231	0,27	225	0,19	270	0,66
$R_4$	225	0,28	267	0,18	279	0,13
$R_5$	207	0,27	246	0,16	261	0,67
$R_6$	216	0,28	237	0,16	237	0,15
$R_7$	219	0,27	225	0,15	303	0,65
$R_8$	213	0,33	240	0,17	294	0,14
$R_9$	228	0,27	225	0,15	300	0,66
$R_{10}$	222	0,24	210	0,19	246	0,15
<b>En İyi Çözüm</b>	207	0,24	210	0,15	237	0,13
<b>En Kötü Çözüm</b>	231	0,33	267	0,19	303	0,67
<b>Standart Sapma</b>	7,49	0,02	16,34	0,01	23,19	0,27
<b>Ortalama</b>	221,10	0,27	233,70	0,17	275,40	0,40

Buharlaşma katsayısı değişiminin çözüm kalitesi ve süresine olan etkisi Çizelge 4.4'te verilmiştir. Buradaki sonuçlar diğer parametreler sabit tutularak incelendiğinde ( $m = 10$ ,  $\tau = 0.25$ ,  $Q = 0.1$ ,  $\alpha = 0.8$ ,  $\beta = 0.7$ ,  $T = 100$ ), tek başına buharlaşma katsayısı artışının, bulunan rota uzunluklarını artırdığı ve rassal olarak bulunan çözümlerin standart sapmalarını da artırdığı görülmüştür. Buharlaşma katsayısı değişiminin çözüm sürelerine anlamlı bir etkisi olmamıştır.

Karınca kolonisi optimizasyon parametrelerinde son olarak feromon yoğunluğu değişiminin etkisi incelenmiştir. Bu inceleme ile ilgili sonuçlar Çizelge 4.5'te görülmektedir. Diğer parametreler sabit tutularak ( $m = 10$ ,  $\rho = 0.25$ ,  $Q = 0.1$ ,  $\alpha = 0.8$ ,  $\beta = 0.7$ ,  $T = 100$ ) incelenen durumlar göz önüne alındığında, feromon yoğunluğu artışı, bulunan ortalama rota uzunluğunu anlamlı bir şekilde olmasa da azaltmıştır. Bu azalma algoritmanın temelini oluşturan karınca feromonlarının, bulunan lokal optimum sonuçlara olan yakınsama olasılığını artırması kaynaklıdır. Çözüm süresi, feromon yoğunluğu değişikliğinden etkilenmemiştir.

**Çizelge 4.5.** Feromon yoğunluğu değişiminin incelenmesi

Feromon Yoğunluğu ( $\tau$ )	0,25		0,50		0,75	
Algoritma Çalıştırma	Rota Uzunluğu (m)	Çözüm Süresi (s)	Rota Uzunluğu (m)	Çözüm Süresi (s)	Rota Uzunluğu (m)	Çözüm Süresi (s)
$R_1$	219	0,64	231	0,71	234	0,27
$R_2$	225	0,38	225	0,64	234	0,30
$R_3$	204	0,64	225	0,28	210	0,35
$R_4$	234	0,37	234	0,31	222	0,33
$R_5$	222	0,33	228	0,34	228	0,26
$R_6$	237	0,33	219	0,32	213	0,30
$R_7$	240	0,38	225	0,66	207	0,39
$R_8$	237	0,35	222	0,29	201	0,66
$R_9$	243	0,46	243	0,34	234	0,66
$R_{10}$	216	0,40	210	0,37	210	0,33
En İyi Çözüm	204	0,33	210	0,28	201	0,26
En Kötü Çözüm	243	0,64	243	0,71	234	0,66
Standart Sapma	12,53	0,12	8,85	0,17	12,61	0,15
Ortalama	227,70	0,43	226,20	0,43	219,30	0,39

## 5. SONUÇ

Genel olarak bir tedarik zincirinde gerçekleştirilen lojistik faaliyetleri arasında depolama ve malzeme taşımaya yönelik operasyonların önemli bir yer teşkil ettiği ve bu operasyonların tüm operasyonlar arasında önemli bir maliyet oluşturduğu görülmektedir. Özellikle iç lojistik olarak adlandırılan sistem içi lojistik faaliyetleri kapsamında, malzeme taşıma operasyonlarının en önemli bileşenlerinden birisi olarak sipariş toplama sürecinden bahsedilebilir. Benzer operasyonların geleneksel depolarda olduğu gibi süpermarket adı verilen üretime doğrudan besleme yapan depolama kurgularında ve yeni nesil insansız depolarda da karşılıkları olabilir. Özellikle geleneksel depolarda, belirli periyotlarda besleme yapılacak üretim sisteminin ilgili periyottaki taleplerinin karşılanması için taşıyıcı bir aracın deponun belirlenmiş bir başlangıç noktasından başlayarak, talep edilen malzemeleri toplanarak araca yüklendiği, sonra tekrar başlangıç noktasına geri dönerek toplanan malzemelerin bir sonraki aşamaya aktarılması için aracın boşaltıldığı bir rotalama süreci söz konusudur.

Sipariş toplamada optimum rotanın bulunması süreci, stoklanan malzemelerin en etkin şekilde taranmasını amaçlamaktadır. Optimum rota bulunarak emek ve zaman kayıplarının en aza indirilmesi, maliyetlerin azaltılması hedeflenmektedir. Optimum ya da optimuma en yakın rota, sezgisel yöntemler, matematiksel modelleme ve yapay zekâ gibi çeşitli bilimsel yöntemler kullanılarak bulunabilmektedir.

Bu tez çalışmasında, karınca kolonisi optimizasyonu (KKO) algoritması ile depo rota planlaması konusu ele alınmıştır. Çalışma kapsamında dikdörtgen şekilli depolarda rota planlaması için bir yaklaşım önerilmektedir. Önerilen yaklaşımla, öncelikle, depo tasarımının bir ağ gösterimi oluşturulmaktadır. Sonrasında toplayıcı aracın, ilgili turda toplayacağı malzemelere bağlı olarak belirlenen toplama noktaları arasındaki en kısa yollar, son aşamadaki rotalama problemi için dikkate alınacak olan uzaklık matrisini oluşturmak üzere Floyd-Warshall algoritması ile belirlenmekte ve son aşamada da KKO algoritması kullanılarak GSP olarak formüle edilmiş olan rotalama problemi için optimize edilmiş bir rota oluşturulmaktadır. Buna ek olarak, önerilen yaklaşımın kolay kullanımının sağlanması için bir arayüz tasarımı gerçekleştirilmiş ve yaklaşım örnek problemler kullanılarak test edilmiştir.

Gelecek alıřmalar kapsamında, depolama sistemi karakteristiklerinin genelleřtirilmesine ek olarak (farklı depo tasarımlarının dikkate alınması gibi), genel olarak önerilen yaklaşımın yöntemsel ve sistemsel olarak geliştirilmesi ele alınabilir. Örneğın, en kısa yolların ve rotanın belirlenmesi amacıyla Floyd-Warshall ve KKO algoritmaları yerine farklı algoritmaların kullanılması ve bunların hesaplama performanslarının incelenmesine ek olarak, tasarlanmış olan arayüzün yeteneklerinin artırılması gibi konular gelecek alıřmalar kapsamında dikkate alınabilecek öneriler olarak sunulabilir.

## KAYNAKLAR

- Baker, P., Perotti, S., 2008. UK Warehouse Benchmarking Report, Cranfield School of Management.
- Bellman, R. (1958). On a routing problem. *Quarterly of applied mathematics*, 16(1), 87-90.
- Cornuejols, G., Fonlupt, J., Naddef, D., 1985. The traveling salesman problem on a graph and some related integer polyhedra. *Mathematical Programming* 33, 1–27.
- De Koster, R., 2004. How to assess a warehouse operation in a single tour. Report, RSM Erasmus University, the Netherlands.
- De Koster, R., Le-Duc, T., & Roodbergen, J. 2007. Design and control of warehouse order picking: A literature review. *European Journal of Operational Research*, 182 (2), 481–501.
- De Koster, R., Van der Poort, E.S., 1998. Routing orderpickers in a warehouse: A comparison between optimal and heuristic solutions. *IIE Transactions* 30, 469–480.
- Dijkstra, E.W. A note on two problems in connexion with graphs. *Numer. Math.* 1, 269–271 (1959).
- Dorigo M, Maniezzo V, Colomi A. 1996. Ant System: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics*, 26(1), 29-41.
- Dreo, J., “Dreaming of Metaheuristics”, <http://metah.nojhan.net>, (2007).
- Flood, M. M. (1956). The traveling-salesman problem. *Operations research*, 4(1), 61-75.
- Floyd, R. W. (1962). Algorithm 97: shortest path. *Communications of the ACM*, 5(6), 345.
- Ford Jr, L. R., & Fulkerson, D. R. (1962). *Flows in networks*. Princeton University Press.
- Frazelle, E. H. 2016. *World-class warehousing and material handling*. McGraw-Hill Education.
- Gademann, N., Van de Velde, S., 2005. Batching to minimize total travel time in a parallel-aisle warehouse. *IIE Transactions* 37 (1), 63–75.
- Glover, F. 1986. Future paths for integer programming and links to artificial intelligence, *Comput. Oper. Res.* 13:533–549.
- Goetschalckx, M., Ashayeri, J., 1989. Classification and design of order picking systems. *Logistics World* (June), 99–106.
- Gu, J., Goetschalckx, M. and McGinnis, L. F. 2007. Research on warehouse operation: A comprehensive review. *European journal of operational research*, 177(1), 1-21.
- Hall, R.W., 1993. Distance approximation for routing manual pickers in a warehouse. *IIE Transactions* 25, 77–87.



- Holland, J. H. 1975. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press.
- Jarvis, J.M., McDowell, E.D., 1991. Optimal product layout in an order picking warehouse. *IIE Transactions* 23 (1), 93–102.
- Johnson, D. B. (1977). Efficient algorithms for shortest paths in sparse networks. *Journal of the ACM (JACM)*, 24(1), 1-13.
- Karásek, J. 2013. An overview of warehouse optimization. *International journal of advances in telecommunications, electrotechnics, signals and systems*, 2(3), 111-117.
- Kirkpatrick S., Gelatt C. D., ve Vecchi M. P. 1983. *Optimization by Simulated Annealing*, Science, vol. 220, pp. 671-680.
- Lambert, D.M., Stock, J.R., Ellram, L.M. (Eds.), 1998. *Fundamentals of Logistics Management*. McGraw-Hill, Singapore.
- Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A.H.G., Shmoys, D.B., 1995. *The Traveling Salesman Problem*. Wiley, Chichester.
- Lin, S., Kernighan, B.W., 1973. An effective heuristic algorithm for the traveling salesman problem. *Operations Research* 21 (2), 498–516.
- Makris, P.A., Giakoumakis, I.G., 2003. k-Interchange heuristic as an optimization procedure for material handling applications. *Applied Mathematical Modelling* 27 (5), 345–358.
- Petersen C. 1999. The impact of routing and storage policies on warehouse efficiency. *International Journal of Operations & Production Management*, 19 (10), pp. 1053-1064
- Petersen, C.G., 1997. An evaluation of order picking routing policies. *International Journal of Operations & Production Management* 17 (11), 1098–1111.
- Ratliff, H.D., Rosenthal, A.S., 1983. Orderpicking in a rectangular warehouse: A solvable case of the traveling salesman problem. *Operations Research* 31 (3), 507–521.
- Roodbergen, K.J., 2001a. *Layout and routing methods for warehouses*. Ph.D. thesis, RSM Erasmus University, the Netherlands.
- Roodbergen, K.J., De Koster, R., 2001b. Routing methods for warehouses with multiple cross aisles. *International Journal of Production Research* 39 (9), 1865–1883.
- Roodbergen, K.J., De Koster, R., 2001c. Routing order-pickers in a warehouse with a middle aisle. *European Journal of Operational Research* 133, 32–43.
- Rushton A., Croucher P., Baker P. 2017, *The Handbook of Logistics and Distribution Management: Understanding the Supply Chain*, London, UK, 291-300.
- Warshall, Stephen. "A theorem on boolean matrices." *Journal of the ACM (JACM)* 9.1 (1962): 11-12.

## ÖZGEÇMİŞ

Adı Soyadı	: Furkan ULU
Doğum Yeri ve Tarihi	: Bursa / 21.12.1996
Yabancı Dili	: İngilizce
Eğitim Durumu (Kurum ve Yıl)	
Lise	: İnegöl Mediha Hayri Çelik Fen Lisesi (2014)
Lisans	: Bursa Uludağ Üniversitesi Mühendislik Fakültesi Endüstri Müh. (2018)
Yüksek Lisans	: Bursa Uludağ Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalı
Çalıştığı Kurum/Kurumlar ve Yıl	: OYAK Renault Otomobil Fabrikaları A.Ş. (Temmuz 2020 – Şubat 2021) – Proje Lideri Maysan Mando Otomotiv Parçaları Sanayi ve Ticaret A.Ş. (Mart 2021-Eylül 2022) – Üretim Planlama Mühendisi Karsan Otomotiv Sanayi ve Ticaret A.Ş. (Eylül 2022- ...) – Lojistik Operasyonları Mühendisi
İletişim (e-posta)	: furkanulu96@gmail.com