

**T.C.
PAMUKKALE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**

**MODÜLER DEPO YÖNETİMİNDE A* TEMELLİ YENİ BİR
ALGORİTMA ÖNERİSİ**

YÜKSEK LİSANS TEZİ

ELİF GÜLFİDAN DAYIOĞLU

DENİZLİ, TEMMUZ - 2017

**T.C.
PAMUKKALE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**



**MODÜLER DEPO YÖNETİMİNDE A* TEMELLİ YENİ BİR
ALGORİTMA ÖNERİSİ**

YÜKSEK LİSANS TEZİ

ELİF GÜLFİDAN DAYIOĞLU

DENİZLİ, TEMMUZ - 2017

KABUL VE ONAY SAYFASI

ELİF GÜLFİDAN DAYIOĞLU tarafından hazırlanan “**MODÜLER DEPO YÖNETİMİNDE A* TEMELLİ YENİ BİR ALGORİTMA ÖNERİSİ**” adlı tez çalışmasının savunma sınavı 11.07.2017 tarihinde yapılmış olup aşağıda verilen jüri tarafından oy birliği / oy çokluğu ile Pamukkale Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı Yüksek Lisans Tezi olarak kabul edilmiştir.

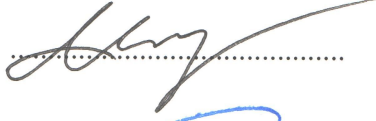
Jüri Üyeleri

İmza

Danışman
Yrd. Doç. Dr. Kenan KARAGÜL



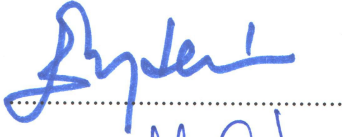
Üye
Doç. Dr. Michael G. KAY
North Carolina State University



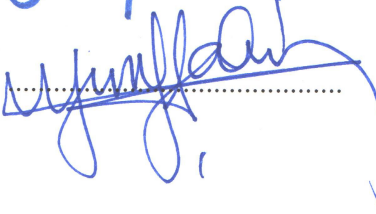
Üye
Prof. Dr. Sezai TOKAT
Pamukkale Üniversitesi



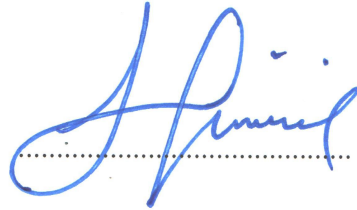
Üye
Yrd. Doç. Dr. Erdal AYDEMİR
Süleyman Demirel Üniversitesi



Üye
Yrd. Doç. Dr. Yusuf ŞAHİN
Mehmet Akif Ersoy Üniversitesi



Pamukkale Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun
19/07/2017 tarih ve 28/19 sayılı kararıyla onaylanmıştır.



Prof. Dr. Uğur YÜCEL

Fen Bilimleri Enstitüsü Müdürü

Bu tezin tasarımı, hazırlanması, yürütülmesi, arařtırmalarının yapılması ve bulgularının analizlerinde bilimsel etięe ve akademik kurallara özenle riayet edildiđini; bu çalışmanın doğrudan birincil ürünü olmayan bulguların, verilerin ve materyallerin bilimsel etięe uygun olarak kaynak gösterildiđini ve alıntı yapılan çalışmalara atfedildiđine beyan ederim.

ELİF GÜLFİDAN DAYIOĐLU



ÖZET

**MODÜLER DEPO YÖNETİMİNDE A* TEMELLİ YENİ BİR
ALGORİTMA ÖNERİSİ
YÜKSEK LİSANS TEZİ
ELİF GÜLFİDAN DAYIOĞLU
PAMUKKALE ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
(TEZ DANIŞMANI: YRD. DOÇ. DR. KENAN KARAGÜL)**

DENİZLİ, TEMMUZ - 2017

Günümüzde depo yönetimi lojistik zincirinin en önemli bileşenlerinden birini oluşturmaktadır. Etkin bir yönetim ve maliyetlerin en aza indirilmesi için depoların tam otomatik hale getirilmesi (yükleme - boşaltma, ayırma - istifleme, paketleri depolama ve geri alma işlemlerinin otomatikleştirilmesi) önemli bir konu haline gelmiştir. Depodaki işlemlerin tam otomatik olarak gerçekleştirildiği bu depolar modüler depo olarak tanımlanmaktadır. Bu çalışmada, dikey olarak yükselip alçalan tekerlekli birim modüllerin bir araya gelmesiyle oluşan modüler bir depo için A* temelli bir sezgisel yöntem geliştirilmiştir. Depo içerisindeki paketlerin hedeflenen çıkış noktalarına minimum süre ve adımda ulaşmaları amaçlanmıştır. Önerilen yöntem mevcut veri kümeleri ile denenmiş ve çözüm süreleri bilimsel yazınlardaki çalışmalarla karşılaştırılmıştır. Sonuç olarak, geliştirilen A* temelli yöntemin karşılaştırılan yöntemlerden daha iyi sonuçlar verdiği belirlenmiştir.

ANAHTAR KELİMELEER: Modüler Depo, A* Algoritması, Sezgisel Yöntemler

ABSTRACT

A NEW ALGORITHM SUGGESTION BASED ON A* IN A MODULAR WAREHOUSE MANAGEMENT

MSC THESIS

ELİF GÜLFİDAN DAYIOĞLU

PAMUKKALE UNIVERSITY INSTITUTE OF SCIENCE

COMPUTER ENGINEERING

(SUPERVISOR:ASSIST. PROF. DR. KENAN KARAGÜL)

DENİZLİ, JULY 2017

Nowadays, warehouse management is one of the most important components of the logistics chain. Full automation of the warehouses (automating loading - unloading, sorting - stacking, the storage and retrieval of packages processes) has become an important issue for effective management and lower costs. These warehouses, which are fully automated, are defined as modular warehouses. In this study, an A* based heuristic for modular warehouse consisting of orthogonal pop-up wheeled unit modules was developed. It is aimed to reach the targeted exit points of the packages in the warehouse in minimum time and step. The proposed method was tested with the available data sets and the resolution times were compared with those in the literature. As a result, it has been determined that the developed A * based method gives better results than the comparative methods.

KEYWORDS: Modular Warehouse, A* Algorithm, Heuristic Methods

İÇİNDEKİLER

Sayfa

ÖZET.....	i
ABSTRACT	ii
İÇİNDEKİLER	iii
ŞEKİL LİSTESİ.....	v
TABLO LİSTESİ.....	vii
TEZ KAPSAMINDA YÜRÜTÜLEN ÇALIŞMALAR İÇİN İNGİLİZCE – TÜRKÇE SÖZLÜK.....	viii
KISALTMA LİSTESİ	ix
SEMBOL LİSTESİ	x
ÖNSÖZ.....	xi
1. GİRİŞ.....	1
2. BİLİMSEL YAZIN TARAMASI.....	3
2.1 15 – Kayan Blok Problemi	3
2.2 Yoğun Trafik Problemi	6
2.3 Depocu Problemi	7
3. LOJİSTİK ve LOJİSTİK YÖNETİMİ	12
3.1 Lojistik Yönetimi	12
3.1.1 Lojistik ve Lojistik Yönetimi Kavramları	12
3.1.2 Lojistiğin Amacı ve Önemi.....	14
3.1.3 Lojistik Faaliyetler	14
3.1.4 Lojistik Süreçler.....	16
3.1.5 Lojistik Maliyetler	17
3.1.6 Kamusal Lojistik Ağ.....	18
3.1.7 Dağıtım Merkezleri.....	19
3.1.8 KLA ve OLA Arasındaki Farklar	22
3.1.9 Eve Dağıtım Lojistik Ağı.....	25
4. Depo Yönetimi	28
4.1 Depo ve Depolama	28
4.2 Depolama Fonksiyonları	32
4.2.1 Taşıma Fonksiyonu	32
4.2.2 Stoklama Fonksiyonu	34
4.2.3 Bilgi Transfer Fonksiyonu	35
4.3 Depo Çeşitleri.....	35
4.4 Modüler Depo Yönetimi	37
4.4.1 Modüler Depo Tanımı	37
4.4.2 Modüler Depo Kavramı ve Kısıtlamaları	37
4.4.3 Temel İşleyiş.....	38
5. ÖNERİLEN ÇÖZÜM YAKLAŞIMI.....	39
5.1 Problemin Tanımı.....	39
5.2 Önerilen Yaklaşım.....	43
5.2.1 Rota Planlama	43
5.2.2 Etiketleme Süreci	49
5.2.3 Ana Kontrol	52
5.3 Veri Kümeleri.....	56
5.4 Önerilen Yaklaşımın Test Problemlerindeki Başarısı	57

5.4.1	Tüm Algoritmaların Çalışma Süreleri ve Göreceli Karşılaştırmaları	59
5.4.2	Tüm Algoritmaların Çözüm Adım Sayılarının Karşılaştırması...	66
6.	SONUÇ VE ÖNERİLER	71
7.	KAYNAKLAR.....	73
8.	EKLER.....	79
9.	ÖZGEÇMİŞ	92

ŞEKİL LİSTESİ

Sayfa

Şekil 2.1: 15 – Kayan Blok Bulmacası.....	3
Şekil 2.2: x ve y karolarının yer değiştirmesi.....	4
Şekil 2.3: Örnek yapılandırma.....	5
Şekil 2.4: Verilen başlangıç durumundan (a) elde edilmesi imkansız olan bitiş durumunun (b) gösterimi.....	5
Şekil 2.5: Yoğun trafik problemi.....	6
Şekil 2.6: Depo sistemi (Sarrafzadeh ve diğ. 1995).....	7
Şekil 3.1: Lojistik ve tedarik zinciri arasındaki ilişki (Rushton ve diğ. 2014).....	13
Şekil 3.2: Lojistik yönetimi örneği (Christopher 2005).....	14
Şekil 3.3: Lojistik faaliyetler (Lambert ve diğ. 1998).....	15
Şekil 3.4: Lojistik yönetimi bileşenleri (Lambert ve diğ. 1998).....	17
Şekil 3.5: KLA’da paket taşınması (Jain 2004).....	19
Şekil 3.6: Tek bir modülün üstten görünümü (Kay 2004).....	20
Şekil 3.7: Yükselmiş durumdaki klavuzlar (Kay 2004).....	20
Şekil 3.8: Yüklerin taşınması için alçalıp yükselen tekerlekler (Kay 2004).....	20
Şekil 3.9: Rihtımlı bir DM örneği (Kay 2004).....	21
Şekil 3.10: DM içerisindeki bir yükün taşınması örneği (Kay 2004).....	21
Şekil 3.11: Domi-No-Drivers.....	25
Şekil 3.12: EDLA örneği (Kay 2013).....	26
Şekil 3.13: 3 Katmanlı depo kontrol sistemi (Kay 2013).....	27
Şekil 4.1: Lojistik sistemdeki maliyetler (Lambert ve diğ. 1998.).....	29
Şekil 4.2: Transfer maliyetini azaltmak için kullanılan depo çeşitleri (Kay 2015).....	31
Şekil 4.3: Depo fonksiyonları (Şahin 2009).....	33
Şekil 4.4: Sipariş toplamanın depolama ve sipariş işlemedeki yeri.....	34
Şekil 4.5: Depo çeşitleri.....	36
Şekil 4.6: Modüler depo örneği.....	38
Şekil 5.1: ALG-B2 çözüm adımları.....	40
Şekil 5.2: ALG-B1 çözüm adımları.....	42
Şekil 5.3: 4 × 4 depo örneği.....	47
Şekil 5.4: Etiketleme süreci örneği.....	50
Şekil 5.5: Önerilen sezgisel yaklaşımın çözüm adımları.....	54
Şekil 5.6: ALG-P5 çözüm adımları.....	56
Şekil 5.7: 4 × 4 boyutundaki problem örneği.....	58
Şekil 5.8: 6 × 6 boyutundaki problem örneği.....	58
Şekil 5.9: 16 × 32 boyutundaki problem örneği.....	59
Şekil 5.10:1. Grup için tüm algoritmaların çalışma sürelerinin karşılaştırması.....	61
Şekil 5.11: 1. Grup için tüm algoritmaların ALG-B1’e göre göreceli karşılaştırması.....	62
Şekil 5.12: 2. Grup için tüm algoritmaların çalışma sürelerinin karşılaştırması.....	63
Şekil 5.13: 2. Grup için tüm algoritmaların ALG-B1’e göre göreceli karşılaştırması.....	63
Şekil 5.14: 3. Grup için tüm algoritmaların çalışma sürelerinin karşılaştırması.....	64

Şekil 5.15: 3. Grup için tüm algoritmaların ALG-B1'e göre göreceli karşılaştırması.....	65
Şekil 5.16: 1. Grup için tüm algoritmaların çözüm adım sayısının karşılaştırması.....	66
Şekil 5.17: 1. Grup için tüm algoritmaların ALG-B1'e göre göreceli karşılaştırması.....	67
Şekil 5.18: 2. Grup için tüm algoritmaların çözüm adım sayısının karşılaştırması.....	68
Şekil 5.19: 2. Grup için tüm algoritmaların ALG-B1'e göre göreceli karşılaştırması.....	68
Şekil 5.20: 3. Grup için tüm algoritmaların çözüm adım sayısının karşılaştırması.....	69
Şekil 5.21: 3. Grup için tüm algoritmaların ALG-B1'e göre göreceli karşılaştırması.....	70

TABLO LİSTESİ

Sayfa

Tablo 2.1: Bilimsel Yazın Taraması.	11
Tablo 3.1: KLA ve İnternet karşılaştırması (Kay 2016).	18
Tablo 3.2: Ağ topolojileri (internet 5, internet 6).....	23
Tablo 3.3: KLA ve OLA karşılaştırması (Kay 2016).	24
Tablo 5.1: A* algoritması.	45
Tablo 5.2: LB hesabı için çözüm örneği.	47
Tablo 5.3: Rota planlamada komşular arası dolaşma.....	48
Tablo 5.4: Rota planlama algoritması.	49
Tablo 5.5: Etiketleme süreci algoritması.	51
Tablo 5.6: Ana kontrol süreci algoritması.....	53
Tablo 5.7: İncelenen tüm algoritmaların karşılaştırması.....	60
Tablo 5.8: 1. Grup için tüm algoritmaların çalışma süreleri karşılaştırmaları. .	61
Tablo 5.9: 1. Grup için tüm algoritmaların göreceli karşılaştırmaları.	61
Tablo 5.10: 2. Grup için tüm algoritmaların çalışma süreleri karşılaştırmaları.	62
Tablo 5.11: 2. Grup için tüm algoritmaların ALG-B1'e göre göreceli karşılaştırmaları.	63
Tablo 5.12: 3. Grup için tüm algoritmaların çalışma süreleri ve göreceli karşılaştırmaları.	64
Tablo 5.13: 3. Grup için tüm algoritmaların çalışma süreleri ve göreceli karşılaştırmaları.	65
Tablo 5.14: 1. Grup için tüm algoritmaların çözüm adım sayısı karşılaştırması.....	66
Tablo 5.15: 1. Grup için tüm algoritmaların ALG-B1'e göre göreceli çözüm adım sayısı karşılaştırması.....	66
Tablo 5.16: 2. Grup için tüm algoritmaların çözüm adım sayısı karşılaştırması.....	67
Tablo 5.17: 2. Grup için tüm algoritmaların ALG-B1'e göre göreceli çözüm adım sayısı karşılaştırması.....	68
Tablo 5.18: 3. Grup için tüm algoritmaların çözüm adım sayısı karşılaştırması.....	69
Tablo 5.19: 3. Grup için tüm algoritmaların ALG-B1'e göre göreceli çözüm adım sayısı karşılaştırması.....	70

**TEZ KAPSAMINDA YÜRÜTÜLEN ÇALIŞMALAR İÇİN
İNGİLİZCE – TÜRKÇE SÖZLÜK**

15 – Puzzle Problem	: 15 – Kayan Blok Problemi
Rush Hour Problem	: Yoğun Trafik Problemi
Warehouseman Problem	: Depocu Problemi
Iterative Deepening A* (IDA*)	: Yinelemeli Derinleştirme A* (YDA*)
Manhattan Pair Distance (MPD)	: Manhattan Çifti Mesafesi (MCU)
Private Logistics Network	: Özel Lojistik Ağ (OLA)
Public Logistics Network (PLN)	: Kamusal Lojistik Ağ (KLA)
Home Delivery Logistics Network (HDLN)	: Eve Dağıtım Lojistik Ağları (EDLA)
Generalized Version of Rush Hour (GRH)	: Genelleştirilmiş Yoğun Trafik Problemi (GYT)

KISALTMA LİSTESİ

- KLA** : Kamusal Lojistik Ağ
DM : Dağıtım Merkezi
OLA : Özel Lojistik Ağ
YDA* : Yinelemeli Derinleştirme A* Algoritması
MCU : Manhattan Çifti Uzaklığı
OYH : Olasılıklı Yol Haritası
EDLA : Eve Dağıtım Lojistik Ağları
SDA : Sürücüsüz Dağıtım Aracı
LB : Alt sınır

SEMBOL LİSTESİ

- G(n)** : Başlangıç noktasından hedef noktaya ulaşmak için gidilmesi gereken yolun maliyeti
- H(n)** : Verilen bir noktadan hedef noktaya ulaşmak için gidilmesi gereken yolun sezgisel maliyeti
- $LB_{(x,y)}^t$** : Başlangıç noktasından (x, y) koordinatına gitmek için hesaplanan alt sınır değeri
- $T_{(x,y)}^{(a_0,b_0)}$** : (a_0, b_0) başlangıç noktasından (x, y) komşusuna olan tahmini dolanma süresi
- $T_{(x,y)}^{(a,b)}$** : (a, b) noktasından (x, y) noktasına gitmek için hesaplanan dolanma süresinin ağırlıklı toplam değeri
- $T_{(a_n,b_n)}^{(x,y)}$** : (x, y) ve (a_n, b_n) arasındaki Manhattan uzaklığı
- $T_{(x,y),t}^{(a,b)}$** : t zaman adımında (x, y) modülünün doluluk durumuna karşılık gelen değer
- w_t** : Keyfi ağırlık değeri

ÖNSÖZ

Bu çalışmada yardımlarını, desteklerini ve bilgi birikimlerini benden esirgemeyen danışmanım **Yrd. Doç. Dr. Kenan KARAGÜL** hocam ve **Doç. Dr. Michael G. KAY** başta olmak üzere, lisans ve yüksek lisans eğitim hayatım boyunca her birinden çok şey öğrendiğim saygıdeğer hocalarıma ve bu çalışmam boyunca benden bir an olsun yardımlarını esirgemeyen arkadaşlarım **Neriman İnak** ve **Merve Özdeş**'e, bu zorlu süreçte her zaman yanımda olan ve manevi desteğini hiçbir zaman esirgemeyen canım aileme çok teşekkür ederim.

1. GİRİŞ

Lojistik; malzeme, parça ve bitmiş envanteri tedarik etme, taşıma ve depolama süreçlerini, mevcut ve gelecek maliyet karlılığının en üst düzeye çıkarılmasını sağlayacak şekilde stratejik olarak yönetme sürecidir. İlk olarak askeri güçler tarafından anlaşılan lojistiğin kritik rolü iş hayatında sonraları tanınmaya başlamış ve lojistik yönetiminin rekabet avantajı elde etmekteki önemi kavranmıştır (Christopher 2005).

Lojistik zincirinin birimlerinden biri olan depolar; hammaddelerin, işlem görmekte olan (yarı mamul) veya bitmiş ürünlerin ihtiyaç anına kadar saklandığı (Kay 2015), ürünlerin başlangıç aşamasından müşteriye teslimine kadar olan süreçteki faaliyetler dizisinin gerçekleştirilmesinde kritik rol oynayan önemli bir bileşendir (Öztürk 2011). Müşterinin ihtiyaç ve talepleri doğrultusunda, minimum maliyetle ürünlerin muhafaza edildiği yerlerdir (Yener 2014). Ticaretin ve rekabetin gelişmesiyle birlikte depolama, bir işlem olmanın yanı sıra arz ve talep dengesini sağlamada stratejik rol üstlenen bir özellik kazanmıştır. Aynı zamanda da depo yönetimi konusu en yüksek maliyetli lojistik unsurdur (Hopbağlı 2009).

Bilimsel yazınlarda depo yönetimi maliyetlerinin minimuma düşürülmesi ile ilgili pek çok çalışma bulunmaktadır. Bu tez çalışmasında ise depo yönetim maliyetlerini en aza indirmek için Kay ve Parlikad (2002), Jain (2004) ve Kay (2008)'ın özel lojistik ağlara alternatif olarak önerdikleri kamusal lojistik ağ (KLA) kavramından faydalanılmış ve KLA'da kullanılan dağıtım merkezleri (DM) ele alınmıştır.

İkinci bölümde; modüler depo yönetimi ile benzer olan problem türleri 15-kayan blok problemi, yoğun trafik ve depocu problemleri açıklanmıştır. Bilimsel yazınlarda bulunan çözüm yöntemleri ve geliştirilen sezgisel algoritmalar taranmış ve bu çalışmalara yer verilmiştir.

Üçüncü bölümde; lojistik ve lojistik yönetimi kavramları ile lojistik faaliyetleri tanımlanmıştır. Lojistik süreçleri ve bu süreçlerin bir işletme açısından

önemi açıklanmıştır. Lojistik ağı türlerinden olan Özel Lojistik Ağ (OLA) ve KLA konuları ele alınmıştır.

Lojistik kavramı açıklandıktan sonra, dördüncü bölümde lojistik zincirinin en önemli bileşenlerinden biri olan depo ve depolama kavramları, depo yönetimi ve lojistik zinciri içerisindeki önemi, depolama fonksiyonları açıklanmıştır. Kay (2013)'ın önerdiği modüler depo kavramı tanımlanmış ve temel işleyişi anlatılmıştır.

Beşinci bölümde ise ele aldığımız problem tanımlanmış, kullanılan veri kümeleri açıklanmıştır. Modüler depo yönetimi için tez çalışmasında önerilen sezgisel algoritma ile Sittivijan (2015) ile Datar (2011)'ın önerdiği yöntemler karşılaştırılmıştır. Bunun sonucunda elde edilen çalışma süreleri ve çözüm sonuçları da tablo ve grafikler ile gösterilmiştir.

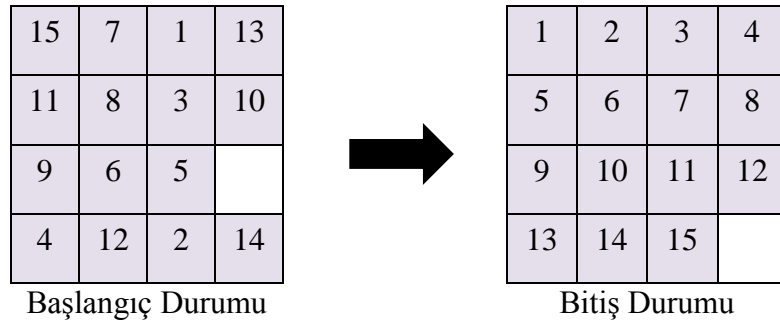
Son bölümde ise bu çalışmada elde edilen sonuçlara ilişkin yorumlara ve önerilere yer verilmiştir.

2. BİLİMSEL YAZIN TARAMASI

Tezin amacı sınırlı sayıda boş alanın olduğu modüler bir depoda birim boyutlu paketlerin taşınması esaslı bir yöntem tasarlamaktır. Dolayısıyla ele alınan problem hareket planlama problemi ile yakından ilişkilidir. Bu bölümde 15-kayan blok problemi, yoğun trafik ve depocu problemleriyle ilgili bilimsel yazınlar incelenecektir. Bu problemler de sınırlı boş alana sahip bir alanda, nesnenin bir noktadan başka bir noktaya taşınmasıyla ilgili olduğundan dolayı incelenen problemin karmaşıklığını anlamada yardımcı olacaktır.

2.1 15 – Kayan Blok Problemi

15-kayan blok problemi, bu tezde ele alınan problemin yapısı ile benzerlik taşımaktadır. Sınırlı bir alanda birden fazla objenin taşınması probleminin daha basit bir biçimidir (Datar 2011). Bu problemde; boyutu 4×4 olan bir kare alanda, belirli bir hedef yapılandırmasına göre yeniden düzenlenecek olan 1’den 15’e kadar numaralandırılmış 15 dolu karo ve 1 adet boş karo bulunur. Tanımlanan boş karo sayesinde bitişik bir karo ortogonal olarak bu konuma kaydırılabilir (Bauer 1994). Amaç, Şekil 2.1’de görüldüğü gibi başlangıç durumundan, karoları sadece yatay veya dikey olarak hareket ettirerek hedeflenen son duruma ulaşmaktır.



Şekil 2.1: 15 – Kayan Blok Bulmacası.

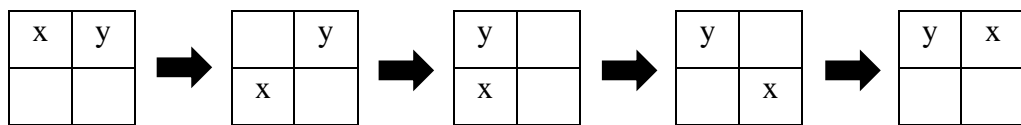
Bu tezde önerilen yapı, kayan blok bulmacasına benzer şekilde çalışacaktır. Tezde önerilen algoritmanın kayan blok bulmacasından farkı, verilen herhangi bir

yapılandırmadan paketlerin alınması ve yerleşimdeki paket yoğunluğunun değişken olmasıdır.

Reinefeld (1993), 8-kayan blok problemini ele almış ve yinmeli derinleştirme A* (YDA*) algoritmasını kullanarak düğüm sıralamasının faydasını değerlendirmiştir. Diğer yinlemeli derinlemesine arama algoritmalarındaki gibi, YDA* da iyi bir düğüm genişleme düzeniyle hedef yapılandırmaya ulaşmadaki iterasyon sayısına harcanan süreyi azaltır. Bulgulara göre sabit bir operatör sırasıyla (örneğin; yukarı, sol, sağ, aşağı) gerçekleştirilen YDA* uygulamalarının basit bir rastgele operatör seçimi ile yapılanlardan daha kötü performans sergilediği sonucuna ulaşılmıştır.

Gue ve Kim (2007), 15-kayan blok tabanlı bir depo sistemi geliştirmiştir. Kayan blok probleminden farklı olarak birden fazla boş alan için de hesaplama yapılmış ve boş alan sayısı arttıkça geri alma zamanının azaldığı sonucuna ulaşılmıştır. Ayrıca 15-kayan bloktan daha büyük boyutlu problem senaryoları ele alınmıştır. Önerilen algoritma, bir birim zamanda tek bir karonun taşınmasına izin verir ve bir boş alandan faydalanmaktadır. Birden fazla boş karo durumunda ise her bir zaman adımında mevcut boş karolar arasından seçim yapmak için sezgisel kullanılmaktadır. Bir karoyu diğer konuma taşımak için gereken minimum boş karo hareket sayısını arayan bir sezgisel sunmuşlardır.

Bauer (1994), YDA* algoritması ile Manhattan mesafe fonksiyonunun ve yeni çift uzaklık sezgisel aracının bir kombinasyonu olan Manhattan çifti uzaklığı (MCU) sezgiselini sunmuştur. Önerilen sezgisel ile 15-kayan blok için sezgisel aramadaki düğüm sayısı %80 oranında azaltılmıştır. Şekil 2.2’de gösterildiği gibi Manhattan uzaklığına göre x ve y ’nin pozisyonlarını değiştirmesi için en az 4 hamle gerekmektedir.



Şekil 2.2: x ve y karolarının yer değiştirmesi.

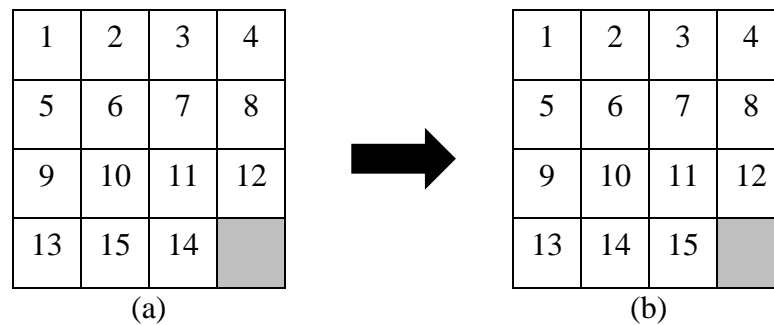
	1	2	11
5	4	7	6
10	9	8	3
14	15	12	13

Şekil 2.3: Örnek yapılandırma.

Önerilen çifti uzaklık yöntemine göre Şekil 2.3'te görüldüğü gibi karolardan 11 ile 3, 10 ile 8, 5 ile 4, 7 ile 6, 14 ile 12, 15 ile 13 çift oluşturmaktadır. Ancak önerilen sezgiselin tek başına faydalı olmadığı belirtilmiştir. Ayrıca küçük boyutlu $(n^2 - 1)$ -kayan blok problemlerinde çift olma olasılığı daha düşük olduğundan dolayı tam bir çözüm elde edilememiştir.

Ratner ve Warmuth (1990), 8-kayan blok ve 15-kayan blok problemlerinin $n \times n$ 'e genişletilmiş halinin NP-zor problem olduğunu kanıtlamışlardır. Başlangıç yapılandırmasından bitiş yapılandırmasına en fazla k adımda geçiş sağlanabilir mi sorusunu tartışmışlardır.

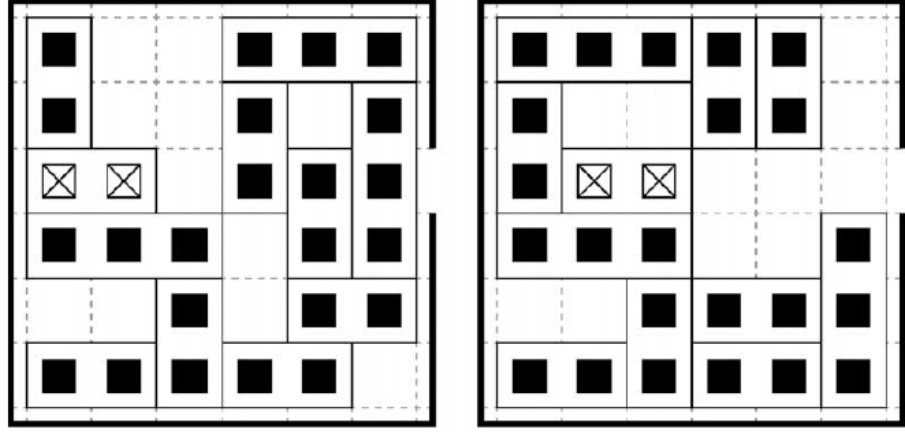
Spitznagel (1967), başlangıç yapılandırmasında bitiş yapılandırmasını elde etmenin ancak çift sayılı permütasyonla mümkün olduğunu ispatlamıştır. Şekil 2.4'te görüldüğü gibi tek bir aktarımla yani tek sayılı permütasyonla yapılandırma değişiminin olması gereken durumların çözümünün olmadığını kanıtlamıştır.



Şekil 2.4: Verilen başlangıç durumundan (a) elde edilmesi imkansız olan bitiş durumunun (b) gösterimi.

2.2 Yoğun Trafik Problemi

Yoğun trafik problemi Şekil 2.5'te görüldüğü gibi, modül tabanlı, çıkış noktasına taşınacak bir hedef araç ve sadece yatay veya dikey doğrultuda hareket eden birkaç araçtan oluşan bir oyundur (Flake ve Baum 2002). Modüldeki diğer arabalar, hedef arabanın belirlenmiş çıkışına giden yolunu açacak şekilde hareket ettirilir.



Şekil 2.5: Yoğun trafik problemi.

Flake ve Baum (2002), hedef aracın modülden çıkış yapıp yapmayacağı kararının PSPACE-Tam olduğunu göstermişlerdir. Ayrıca orijinal yoğun trafik probleminden farklı olarak, keyfi genişlik ve yükseklik ile çıkışın, ızgaranın çevresindeki herhangi bir yerinde bulunma seçeneğiyle yoğun trafik probleminin geliştirilmiş bir versiyonunu (GYT – Genelleştirilmiş Yoğun Trafik Problemi) sunmuşlardır. Yoğun trafik probleminden farkı, modülün boyutunun rastgele genişlikte bir dikdörtgen olabilmesi ve hedef arabanın çıkışının herhangi bir konumda olmasına izin verilmesidir. GYT devrelerini, hedef arabayı serbest bırakmanın tek yolunun tersine çevrilebilir çift raylı rastgele erişimli bir makineyi taklit edebileceği biçimde inşa etmek olduğunu ileri sürmüşlerdir.

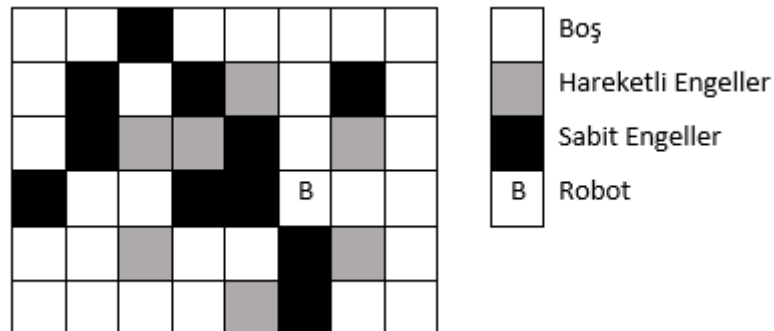
Hearn ve Demaine (2005), minimum akış kısıtlamaları ile ağırlıklı yönlü çizgelerdeki ters kenar yönlerine dayanan deterministik olmayan bir hesaplama modeli önermişlerdir. Geliştirdikleri çerçeve, Flake ve Baum (2002) tarafından geliştirilen GYT'den esinlenerek geliştirilmiştir. Ayrıca, herhangi bir 1×2 boyutlu blok kaymanın PSPACE-zor olduğunu kanıtlamışlardır.

Hauptman ve diğ. (2009), bir PSPACE-Tam sorun olan yoğun trafik probleminin 6×6 ve 8×8 sürümlerinin çözümü için genetik programlama protokolleri ile geliştirilmiş YDA* temelli bir sezgisel önermişlerdir.

2.3 Depocu Problemi

$n \times n$ kayan blok probleminin bir uzantısı olan depocu problemi (Sittivijan 2015), sınırlı bir alandaki çok sayıda bağımsız nesnenin koordine edilebilen hareket planlamasını içerir (Sharma ve Aloimonos 1992). Amaç, depodaki nesnelere hareket ettirerek nesnelere başlangıç yapılandırmasından son yapılandırmaya ulaştırmaktır (Datar 2011). Engellerin varlığında çok sayıda üç boyutlu nesnenin koordineli hareket planlaması, karmaşıklığı düzenlemek için önemli olduğu bir hesaplama problemidir (Hopcroft ve diğ. 1984).

Sarrafzadeh ve Maddila (1995), ızgara çizgileri boyunca yatay ve dikey olarak hareket etmesine izin verilen, Şekil 2.6'da görüldüğü gibi iki boyutlu birim kare nesnelere (robot ve engeller) oluşan iki boyutlu depo sistemi oluşturmuşlardır. Çalışmada ele alınan problem, hareketli engellerin olduğu hareket planlama problemi olarak ele alınıyor. Engellerin hareketi için uzaktan mekanizma ve doğrudan temas temelli (engellerin yalnızca robotun doğrudan teması ile hareket ettirilmesi) hareketler incelenmiş ve bu hareketleri oluşturmak için gerekli durumları araştırmışlar, nesnenin başlangıçtan son konumuna hareketini garantilemek için gereken koşulları açıklamışlardır. Hem uzaktan hareket hem de doğrudan temas ile hareketlerde, robotun başlangıç konumundan son konumuna nasıl taşınacağı ve bu iki konum arasında nasıl bir yol inşa edileceği sorunları üzerine çalışmışlardır.



Şekil 2.6: Depo sistemi (Sarrafzadeh ve diğ. 1995).

Guo ve Parker (2002), birden fazla robot için dağıtık ve en iyi hareket planlama algoritması önermişlerdir. Hesaplama karmaşıklığı fazla olan bu problem yol planlama ve hız planlama olarak iki modüle ayrılmış ve her iki modüle de D* arama yöntemi uygulanmıştır. En iyileme, hem bireysel robot seviyesinde hem de takım seviyesinde maliyet fonksiyonlarını en küçüklemeyi başarmıştır.

Mevcut çok robotlu hareket planlama algoritmaları, robotlar arasındaki bilgi taşıma yapısına göre, genellikle merkezi ve merkezi olmayan olmak üzere sınıflandırılmaktadır. Merkezi olmayan planlamada, her robotun yani hareket eden her nesnenin hareketi ayrı ayrı planlanır, merkezi yaklaşımda ise tüm nesnelere hareketi merkezi bir yol planlayıcı kullanarak planlanır. Guo ve Parker (2002) çoklu robotlar için merkezi olmayan bir hareket planlama algoritması tasarlamışlardır.

Hopcroft ve diğ. (1984) 2 boyutlu dikdörtgen şeklindeki bir alan içinde hareket etmek için sınırlandırılmış ayrı farklı boyutlu dikdörtgen cisimler topluluğu için eşzamanlı hareket planlama sorununun PSPACE-zor olduğunu kanıtlamışlardır.

Yamashita ve diğ. (2003), üç boyutlu bir ortamda büyük bir nesnenin birlikte taşınması için çoklu mobil robotların hareket planlamasına yönelik iki aşamalı bir yöntem önermişlerdir. Buna bağlı olarak hareket planlayıcısını, küresel bir yol planlayıcı ve yerel bir hareket planlayıcısı olarak iki aşamaya ayırıp bunların entegrasyonunu gerçekleştirmişlerdir. Küresel yol planlamacısında, nesne hareketinin kısıtlamaları, A* aramasındaki maliyet fonksiyonu ve sezgisel fonksiyon olarak düşünülür. Yerel hareket planlayıcısı için konum kontrolüne dayalı olarak robotlar için hareket tekniği geliştirilmiştir.

LaValle ve Hutchinson (1998), çoklu robotların hareket planlaması için oyun kuramı bilimsel yazınlardaki kavramlarla tutarlı bir optimallik yöntemi sunmuşlardır.

Leroy ve diğ. (1999), çoklu robotların hareket planlaması için geometrik tabanlı bir yöntem önermişlerdir. Tüm robotların yolları birbirinden bağımsız olarak hesaplanırken, robotların kendi yollarındaki hareketlerini birbirleriyle çarpışmayacak şekilde koordine etme sorununa değinilmiştir. Önerilen algoritma, $n \times n$ boyutlu koordinasyon diyagramında arama yapmaktan oluşur.

Olasılıksal yol haritaları (OYH), bir robotun çarpışmayan alanının bağlantısını yakalamak ve yol planlama problemlerini pek çok serbestlik derecesi ile çözmek için etkili bir araçtır. Shancez ve Latombe (2002), birden fazla etkileşimli robot için çarpışmadan serbest yollar planlayan OYH kullanmışlardır. TIC (Tekli Sorgu, İki Yönlü, Çarpışmayı kontrol etmekte tembel) olarak adlandırılan, tek sorgulamalı iki yönlü örnekleme stratejisini tembel bir çarpışma kontrol bağlantı stratejisi ile birleştiren yeni bir OYH planlayıcısı geliştirmişlerdir.

Sharma ve Aloimonos (1992), birim boyutlu olmayan nesnelere için nesnelere boyutlarına kısıtlamalar getirerek ve boş alanların dağılımıyla depocu probleme çözüm önermişlerdir. Boş alanların kısıtlanması için geçici depolama kavramını tanıtmışlardır. Bu geçici depolama alanı kullanımı, boş alanın gerçek dağılımı üzerinde kısıtlamalar getirmenin yanı sıra, kullanılabilir alanın sezgisel fikrinin biçimlendirilmesine yardımcı olur.

Svestka ve Overmars (1998), çoklu robot yol planlaması problemine alışılmış merkezi olmayan planlamanın aksine koordine edilmiş bir yaklaşım sunmuşlardır. Önerilen sistemde çoklu robot hareket bilgilerini depolayan veri yapısı iki adımda oluşturulmuştur. İlk adımda, olasılıklı yol planlayıcısı kullanılarak sadece bir robot için bir yol haritası oluşturulmuş, ikinci adımda da bu basit yol haritalarının bir kısmı, bileşik robot için bir yol haritası haline getirilmiştir.

Lumelsky ve Harinarayan (1997), sabit engellerin bilinmediği 2 boyutlu bir ortamda çalışan çoklu robot için merkezi olmayan gerçek zamanlı hareket planlaması için bir yaklaşım önermişlerdir. Bir robotun çevresindeki nesnelere algılayabildiği, mevcut ve hedef konumunu bildiği, bir robotu bir engelden ayırt edebildiği ve başka bir robotun anlık hareketini değerlendirebildiği bir ortam ele alınmıştır. Bunların aksine, bir robotun ortam hakkında veya diğer robotların yol ve hedefleri hakkında bilgisi yoktur. Labirent arama tekniklerine dayalı bir strateji sunulmuştur.

Azarm ve Schmidt (1997), çoklu robotların çarpışma sorununun çözülmesi için merkezi olmayan ve paralel karar vermeye izin veren bir anlaşma çerçevesi geliştirdi. Önerilen çerçeve paralel yol hesaplamasına ve dinamik öncelik atmasına izin verir.

Liu ve diğ. (2006), çoklu robot sistemleri için çarpışma önleme özellikli ve karınca koloni algoritmasına dayalı bir yol planlaması sunmuştur. Hareketli robotlar arasındaki çarpışmayı çözmek için "ilk gelen ilk hizmet alır" üzerine bir davranış stratejisi benimsemişlerdir. Yeung ve Bekey (1987), problemi küresel yol planlama sorunu ve yerel yol planlama problemi olmak üzere iki alt problem haline getirilmesine dayanan merkezi olmayan bir yaklaşım önermişlerdir.

Yapılan araştırmalar ve bu çalışma ile olan ilişkiler Tablo 2.1'de gösterilmiştir.

Tablo 2.1: Bilimsel Yazın Taraması.

Benzer Araştırmalar	İncelenen Kaynaklar	Tez İle Olan İlişkisi
15-kayan blok problemi ve türevleri	Spitznagel 1967 Ratner ve Warmuth 1990 Reinefeld 1993 Bauer 1994 Gue and Kim 2007	Problemin karmaşıklığını anlamayı ve algoritmayı nasıl tasarlayacağımız hakkında fikir sahibi olmamızı sağlar. Boş alanların sınırlı olduğu alandaki nesnelerin hareketinin kontrolünü anlamakta fayda sağlar.
Yoğun Trafik Problemi	Flake ve Baum 2002 Hearn ve Demaine 2005 Hauptman ve diğ. 2009	15 puzzle probleminin geliştirilmiş bir türüdür. Hedef aracın çıkış noktasına varmasında kullanılan yöntemlerden faydalanılarak tezimizde ele aldığımız nesnelerin başlangıç yapılandırmasından bitiş yapılandırmasına olan yolun tanımlanmasında yeni fikirler üretilebilir.
Depocu Problemi	Hopcroft ve diğ. 1984 Yeung ve Bekey 1987 Sharma ve Aloimonos 1992 Sarrafzadeh ve Maddila 1995 Lumelsky ve Harinarayan 1997 Azarm ve Schmidt 1997 Svestka ve Overmars 1998 LaValle ve Hutchinson 1998 Leroy ve diğ. 1999 Shancez ve Latombe 2002 Guo ve Parker 2002 Yamashita ve diğ. 2003 Liu ve diğ. 2006	Birden çok nesnenin hareketini kontrol etmeye yönelik ileri sürülen yöntemler yeni algoritmalar geliştirmede faydalı fikirler üretmek için kullanılabilir.

3. LOJİSTİK ve LOJİSTİK YÖNETİMİ

3.1 Lojistik Yönetimi

Bu bölümde lojistik ve lojistik yönetimi kavramları ile lojistik faaliyetleri tanımlanmıştır. Lojistiğin bir işletme açısından önemine ve lojistik süreçlerine değinilmiştir. Lojistik ağlarına farklı bir çerçeveden bakan özel lojistik ağ (OLA) ve kamusal lojistik ağ (KLA) kavramları ele alınmıştır.

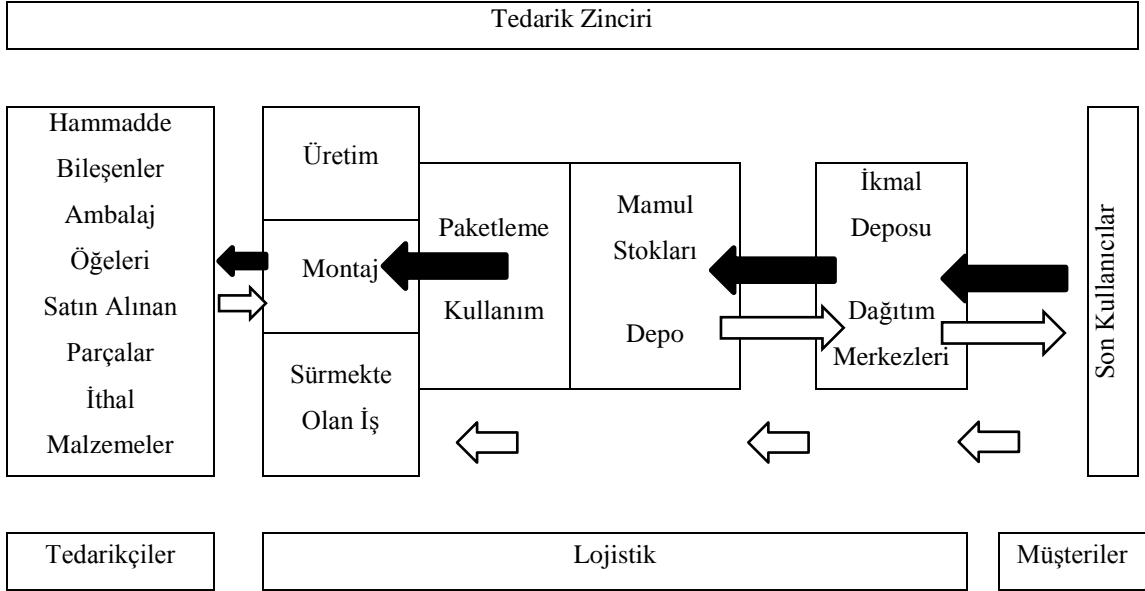
3.1.1 Lojistik ve Lojistik Yönetimi Kavramları

İnsanlık tarihi boyunca savaşlar, lojistik altyapısının gücü ve yeteneklerinin üstünlüğü ile kazanılmış ya da yoksunluğu ile kaybedilmiştir. Lojistiğin kritik rolü ilk olarak ordular tarafından anlaşılmıştır. Ordulardaki kullanımından elde edilen faydalar zaman içinde iş hayatında anlaşılmaya başlanmış ve lojistik yönetiminin rekabet avantajı oluşturmaktaki önemi kavranmıştır (Christopher 2005).

Kökene askeri alandaki uygulamalara ve araştırmalara dayanan lojistik; birliklerin mühimmat ve yakıtlarının doğru şekilde tedarik edilmesini garanti altına almak için kullanılan yöntemlerin incelenmesi olarak ortaya çıkmıştır ve amacı ordulara en etkin şekilde hareket etme ve savaşma imkanı sağlamaktır (Ghiani ve diğ. 2013). Askeri kökenden türemiş olmasına rağmen endüstriyel alan uygulamalarının ardından gelişen endüstriyel ve akademik araştırmalar sonucunda çeşitli tanımlar üretilmiştir (Öztürk 2011).

Lojistik; malzeme, parça ve bitmiş envanteri tedarik etme, taşıma ve depolama operasyonlarını, mevcut ve gelecek maliyet kârlılığının en üst düzeye çıkarılmasını sağlayacak şekilde stratejik olarak yönetme sürecidir. Lojistik, temel olarak, ürün ve bilgi akışı için tek bir plan oluşturmak isteyen bir planlama yaklaşımı ve çerçevedir. Tedarik zinciri yönetiminin dayandığı bu yaklaşım ve çerçeve, tedarikçiler ve müşteriler arasındaki süreçler arasında bağlantı kurmayı ve eşgüdüm sağlamayı hedefler (Christopher 2005). Dolayısıyla, lojistik tedarik zinciri yönetimi

yapısının bir parçası olarak görülür. Lojistik ve tedarik zinciri yönetimi terimleri eş anlamlı olarak kullanılırken, lojistik aslında tedarik zinciri yönetiminin bir alt kümesidir (Harrison ve Hoek 2008). İki kavram arasındaki bu ilişki Şekil 3.1’de gösterilmiştir.



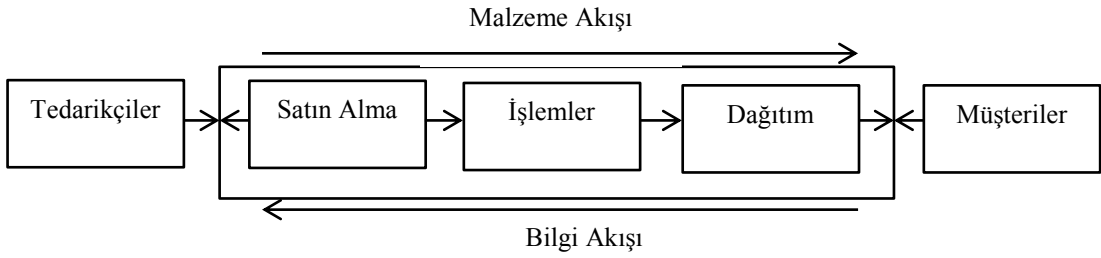
Şekil 3.1: Lojistik ve tedarik zinciri arasındaki ilişki (Rushton ve diğ. 2014).

Lojistik çalışanları için önde gelen profesyonel organizasyonlardan biri olan Lojistik Yönetimi Konseyi (The Council of Logistics Management - CLM)'nin tanımına göre:

Lojistik yönetimi; müşteri isteklerini karşılamak amacıyla hammadde üretim noktasından tüketim noktasına kadar malların, hizmetlerin ilgili bilginin etkin ve etkili akışı ve depolanmasının planlama, uygulama ve kontrol etme sürecidir (Lambert ve diğ. 1998).

Şekil 3.2’de de görüldüğü gibi lojistik yönetimi, müşteri ihtiyaçları doğrultusunda, pazardan başlayarak firma ve fonksiyonel faaliyetleri boyunca malzeme ve bilgi akışlarını kapsayan bir sistemdir (Christopher 2005). Malzeme akışı, malların dağıtım kanalları aracılığıyla tedarikçilerden mağazalara/müşterilere ulaşmasıdır. Bilgi akışı da, malzeme akışının doğru bir şekilde planlanıp kontrol edilebilmesi için müşteri talep ve satın alma bilgilerinin tedarikçiye, tedarikçiden de perakendeciye tedarik bilgilerinin aktarılmasıdır (Harrison ve Hoek, 2008). Bunun

yanı sıra bilgi akışı, tedarik zinciri yönetimi faaliyetlerinin yürütülmesinde zorunlu olan altyapı, araç-gereç ve kaynakları da içerir (Ghiani ve diğ. 2013).



Şekil 3.2: Lojistik yönetimi örneği (Christopher 2005).

3.1.2 Lojistiğin Amacı ve Önemi

Lojistik; bir şirketin varlığını devam ettirebilmesi için hizmet, kalite, fiyat gibi rekabet değişkenlerine karşı dayanıklı hale getirmeyi amaçlar (Hopbağlı 2009). Bu doğrultuda sermaye ve işletme maliyetini en aza indirmeyi ve müşteri hizmet kalitesini en üst seviyeye çıkarmayı hedefler. İşletme maliyetini düşürme amacı-doğrultusunda daha iyi depolama ve ulaşım seçenekleri kullanılarak taşıma ve depolama maliyetleri en aza indirgenmeye çalışılır. Hizmet kalitesi hedefinde de sipariş yanıt süresini en aza indirip ürünün kullanılabilir kalitede müşteriye ulaştırılması sağlanır (Kasilingam 1998). Bu nedenle hammadde kaynaklarından son müşteriye kadar tüm alanda lojistiğin avantajları görülmektedir (Güler 2006).

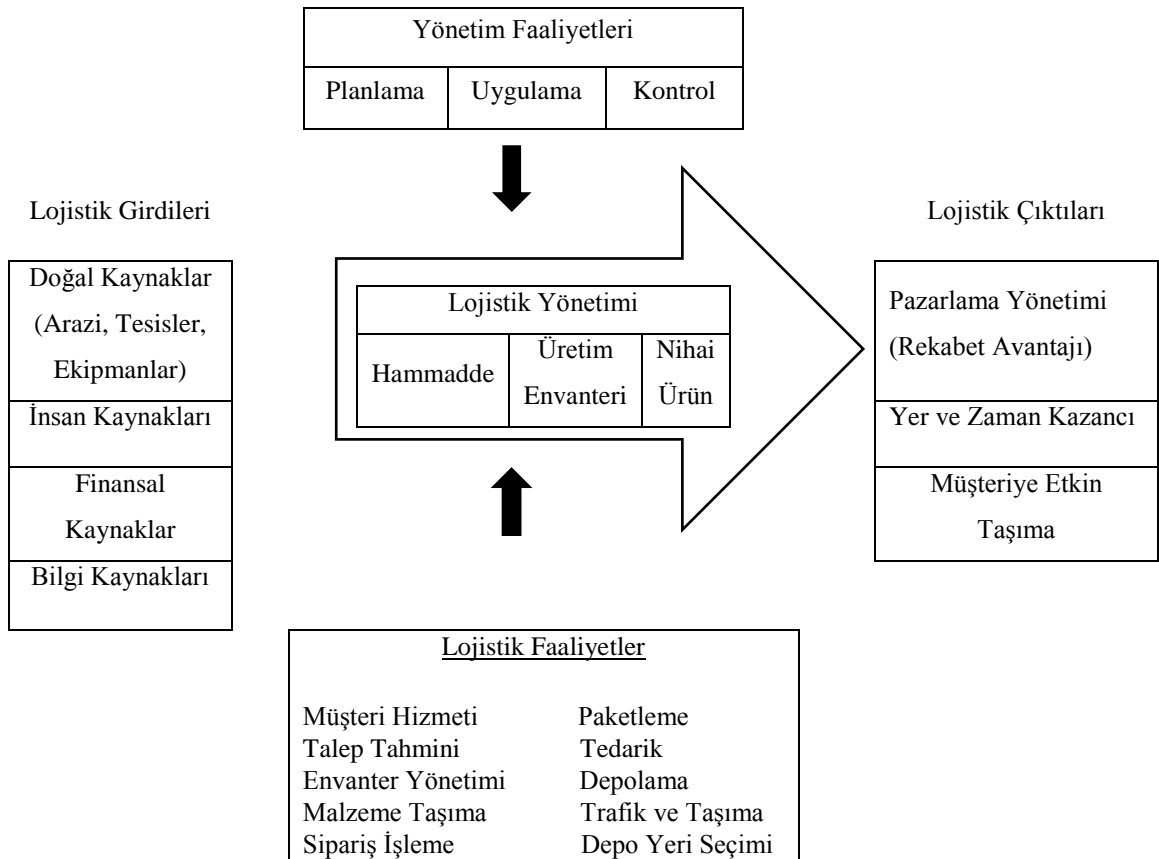
3.1.3 Lojistik Faaliyetler

Lojistik faaliyetler, bir ürünü başlangıç noktasından tüketim noktasına getirmek için akıştaki faaliyetler bütünüdür. Bu lojistik faaliyetler;

- Müşteri Hizmeti
- Talep Tahmini
- Envanter Yönetimi
- Lojistik İletişimi
- Malzeme Taşıma (Elleçleme)
- Sipariş İşleme

- Paketleme
- Bölümler ve Hizmet Desteği
- Tesis ve Depo Yeri Seçimi
- Tedarik
- İade malların Taşınması
- Depolama ve Stok
- Trafik ve Taşıma olarak sıralanabilir.

Lojistik sistemlerde çıktılar, rekabet üstünlüğü, zaman ve yer kazancı sağlayan hizmetler karışımıdır. Bu çıktılarının elde edilmesi de lojistik faaliyetlerin etkin bir şekilde gerçekleştirilmesi ile mümkündür. Lojistik çerçevesi içinde yer alan ve lojistik sistemlerde önemli bir yeri olan bu faaliyetler, lojistik sistem içerisindeki konumu ve diğer işlevler ile olan bağlantıları Şekil 3.3'te gösterilmiştir.



Şekil 3.3: Lojistik faaliyetler (Lambert ve diğ. 1998).

Depolama ve stok, toplam maliyetin en küçüklenmesinde önemli rollerden birini oynar. Çünkü depolama, malzemenin hammaddeden nihai ürüne dönüşme sürecinde, daha sonrasında da tüketicilere dağıtımına kadar olan süreçte ortaya çıkar. Stok yönetimi de arz-talebe göre gerekli ürün ve malzemenin yeterli düzeyde bulundurulmasıdır. Çünkü stok miktarı arttıkça nakit para akışı yavaşlar, üretilen mal bozulabilir veya depoda yer kaplayacağından dolayı fazladan depo maliyeti çıkarır. (internet 1). Tesis ve depo yeri seçimi sadece taşıma maliyetlerini düşürme açısından değil aynı zamanda müşteri hizmet kalitesini ve müşteriye geri dönüş hızı açısından önemli bir lojistik faaliyetidir. Müşteri hizmeti; ürünün, doğru müşteriye doğru zamanda ve yerde, minimum maliyetle ulaştırılmasıdır. Talep tahmini; tedarikçilerden malzemelerin ne kadar sipariş edilmesi gerektiği, üretilen ürünlerin ne kadarının taşınması veya tutulması ile ilgili faaliyetlerdir (Lambert ve diğ. 1998). Malzeme taşıma diğer bir adıyla da elleçleme olarak isimlendirilen lojistik faaliyeti; malzemenin tesis ve/veya depo içerisinde taşınması, istiflenmesi ve nakliye aracına taşınarak yüklemesi işlemlerini içermektedir (Özalp 2013). Malzeme taşıma faaliyeti; taşıma mesafesini, darboğazları, stok seviyelerini, hatalı kullanım ve hasar nedeniyle kaybı en aza indirmeyi içerdiğinden malzeme akışlarının iyi bir analizinin yapılmasıyla önemli ölçüde maliyet tasarrufu elde edilebilir (Lambert ve diğ. 1998).

3.1.4 Lojistik Süreçler

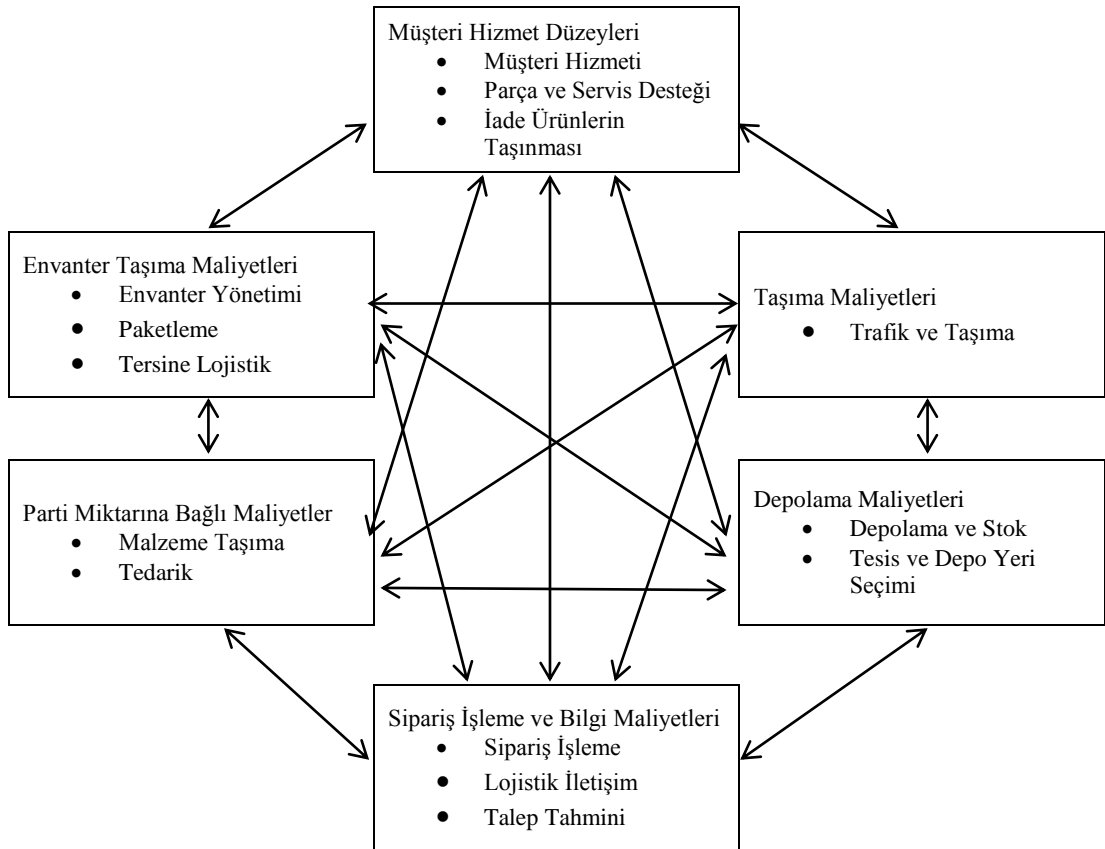
Lojistik faaliyetler üretim ve dağıtım süreçlerine bağlı olarak farklılık göstermektedir. Örnek olarak dağıtım sürecindeki ile tedarik sürecindeki nakliye ve depolama işlemleri birbirinden farklı anlayışlarla yerine getirilmektedir (Özalp 2013). Lojistik faaliyetlerin yerine getirilmesinde farklılık görülen bu süreçler tedarik lojistiği, tesis içi lojistik ve dağıtım lojistiği olarak 3 ana başlık altında toplanabilmektedir.

Tedarik lojistiği, sadece üretim öncesinde gerçekleşen, hammadde ve malzeme vb. bileşenlerin tedarik yönetimiyle ilgili faaliyetleri içerir. Tesis içi lojistik süreci, üretim tesislerinde gerçekleştirilir, ham maddelerin ve yardımcı malzemelerin nihai mallara dönüştürülmek üzere depodan veya hat üzerinden üretim alanlarına taşınmasını ve aynı zamanda yarı mamullerin ve nihai ürünlerin paketlenmesi ve

depolanması aşamalarını içerir. Dağıtım lojistik süreci ise nihai ürünlerin satış noktalarına veya müşterilere ulaştırılması ile ilgili olan faaliyetleri içerir (Ghiani ve diğ. 2013).

3.1.5 Lojistik Maliyetler

Lojistik maliyetler, lojistik sürecindeki faaliyetlerden dolayı ortaya çıkan maliyetlerdir. Müşteri hizmeti, taşıma, depolama, satın alma ve sipariş işleme gibi kategoriler temel lojistik maliyetleri arasında yer almaktadır (Lambert ve diğ. 1998). Şekil 3.4'te lojistik maliyetler ve bu maliyetlerle ilişkili lojistik faaliyetler gösterilmiştir.



Şekil 3.4: Lojistik yönetimi bileşenleri (Lambert ve diğ. 1998).

3.1.6 Kamusal Lojistik Ağ

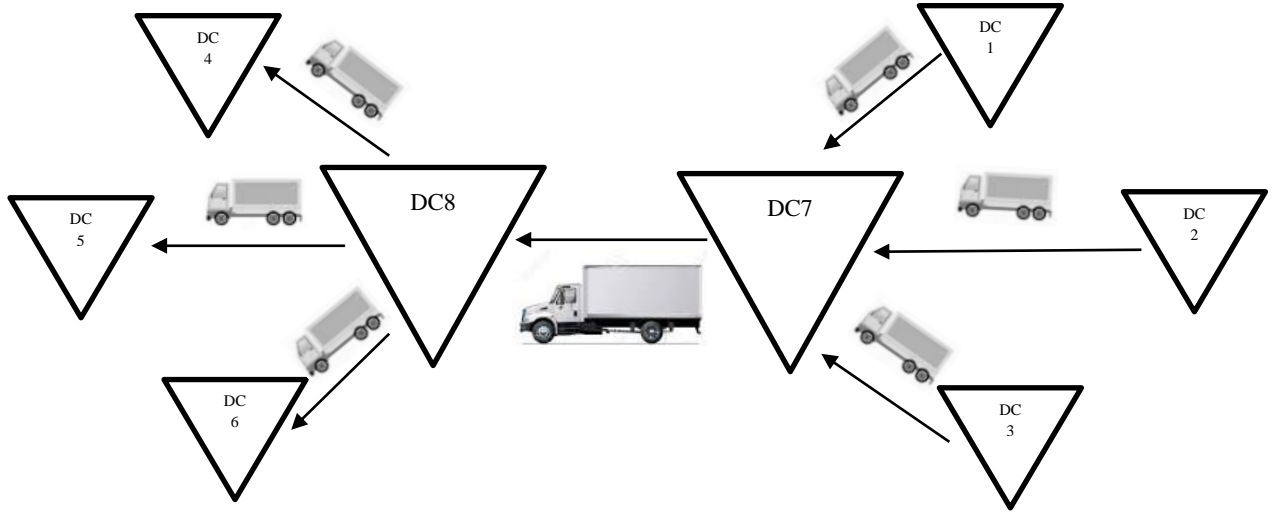
KLA, yüklerin taşınmasında OLA'ya alternatif olarak sunulmuş bir ağ çeşididir. KLA' yı oluşturanlar; dağıtım merkezleri (DM), kamyonlar ve paketlerdir. İnternet üzerinden transfer edilen paketler arasındaki yapıdan esinlenerek geliştirilen bu ağda, bir paket bir mağazadan büyükşehir bölgesi içinde (metropolitan area) bir alana yerleştirilmiş olan kamusal dağıtım merkezlerine gönderilir. Bu noktadan da paketlerin ilgili yere taşınma işlemi gerçekleştirilir (Xiang ve diğ. 2007). İnternet ve KLA arasındaki yapı benzerliği Tablo 3.1'de gösterilmiştir.

Tablo 3.1: KLA ve İnternet karşılaştırması (Kay 2016).

Kamusal Lojistik Ağ	İnternet
Taşınan Paketler	İletilen Paketler
Dağıtım Merkezleri	Yönlendiriciler
Kamyonlar	Kablo, fiber

KLA'daki dağıtım noktaları, internetteki yönlendiricilere karşılık gelmektedir. KLA'da bir paket, İnternettekine benzer bir şekilde genel dağıtım merkezlerine gönderilir ve bu noktalardan da ilgili yerlere/müşteriye teslimat için yönlendirilir. A ve B olmak üzere iki ayrı firmanın olduğunu varsayalım. OLA'da her firma kendi lojistik ağlarında taşıma işlemlerini gerçekleştirecektir. Firmaların sınırlı sayıda aktarma noktaları olduğundan dolayı, paketler varış noktasına gelene kadar gereksiz noktaları da dolaşabilmektedir. KLA'da ise farklı fonksiyonlar (depolama, transfer... vb.) farklı firmalar tarafından koordineli bir şekilde gerçekleştirilebilir. Buna örnek olarak başta verilen A firmasının kamyonlarıyla B firmasının dağıtım noktalarının koordineli olarak çalıştığı gösterilebilir. Bu sayede hem ekonomik açıdan büyük kazanç elde edilir hem de paketlerin müşteriye daha kısa sürede ulaşması sağlanır (Sittivijan 2015). Ayrıca bu durum sınırlı kapasiteye sahip küçük işletmelerin de paket taşımacılığına aktif olarak katılmalarına olanak sağlar. Şekil 3.5'te görüldüğü gibi farklı dağıtım noktalarından gelen ve farklı varış noktaları bulunan P1, P2 ve P3 isimli üç ayrı paket bulunmaktadır. P1 DM1'den gelip DM4'e, P2 DM2'den gelip DM5'e ve P3 de DM3'ten DM6'ya taşınacaktır. KLA sayesinde üç paket için de ortak olan DM7 – DM8 arasındaki ortak yol tek bir kamyon yüklemesi ile gerçekleştirilebilmektedir. Paketler DM7'den DM8'e kadar

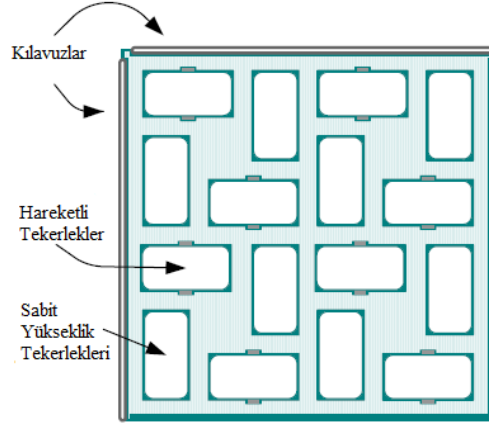
beraber taşınıp bu noktadan sonra dağıtım işlemleri gerçekleştirilir. Bu sayede kamyon tamamen doldurulacağı için maliyette bir düşüş sağlanır (Jain 2004).



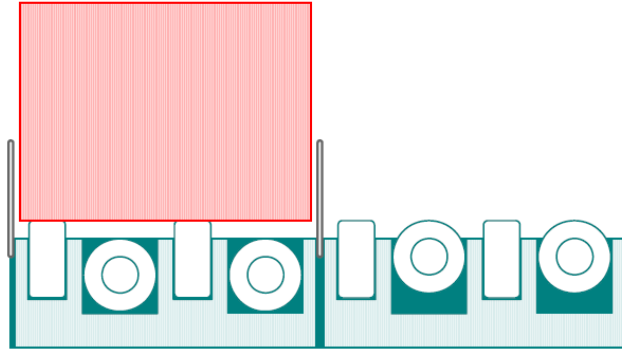
Şekil 3.5: KLA'da paket taşınması (Jain 2004).

3.1.7 Dağıtım Merkezleri

Dağıtım merkezleri, gerçekleştirilen operasyonların otomasyon sistemleri ile gerçekleştirilmesi sayesinde firmalar için maliyetlerin önemli ölçüde azaltılmasını sağlar bundan dolayı lojistik ağdaki en önemli bileşenlerdir (Datar 2011). Yükler, lojistik ağda transfer edilirken bir veya daha fazla DM'yi ziyaret edebileceğinden dolayı maliyetin minimuma indirilmesi için her bir DM'deki yükleme / boşaltma, sıralama ve depolama faaliyetlerinin olabildiğince otomatik hale gelmesi gerekmektedir (Kay 2004). Kay (2004), bu gereksinimlerin karşılanacağı bir DM tasarımı önermiştir. Önerilen sistem dikey olarak yükselip alçalan tekerlekli kare modüllerden oluşmaktadır. Şekil 3.6'da tek yönde hareket eden ve sabit yükseklikli tekerlekler ile diğer yönde hareket eden ve alçalıp yükselebilen tekerleklerden oluşan modüllerden birinin üstten görünümü gösterilmiştir. Kılavuzlar ise gerektiğinde yükselip alçalarak bir yükün hareketini sınırlamayı ve yönlendirmeyi sağlar (Bkz. Şekil 3.7).

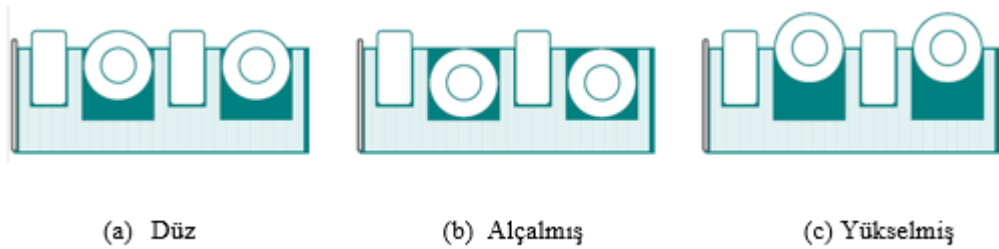


Şekil 3.6: Tek bir modülün üstten görünümü (Kay 2004).



Şekil 3.7: Yükselmiş durumdaki klavuzlar (Kay 2004).

Şekil 3.8’de (a) modül üzerinde bir yükün durduğu zaman her iki tip tekerleğin de aynı hizada olduğu durumu, (b) modül taşınacağı zaman hareketli tekerleklerin alçaldığı durumu, (c) modül diğer bir doğrultuya taşınacağı zaman hareketli tekerleklerin yükseldiği durumu göstermektedir. Bu tekerleklerin alçalıp yükselmesiyle yükler depoda yatay ve dikey doğrultuda hareket edebilmektedir.



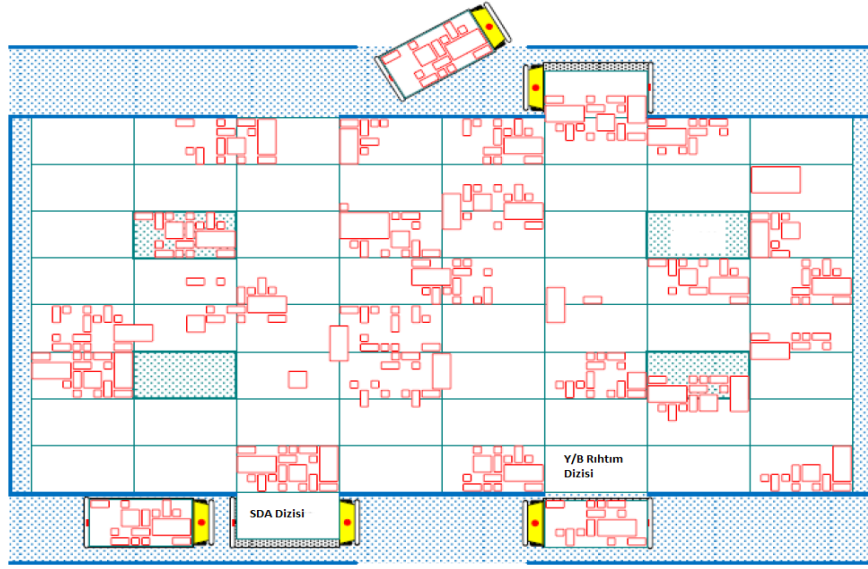
(a) Düz

(b) Alçalmış

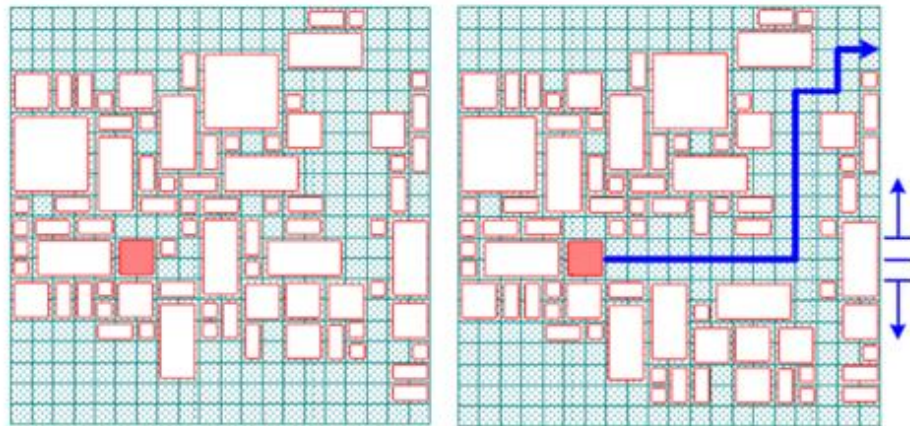
(c) Yükselmiş

Şekil 3.8: Yüklerin taşınması için alçalıp yükselen tekerlekler (Kay 2004).

Şekil 3.6’da gösterilen kare modüller bitişik olarak bir araya gelerek düzlemsel bir yüzey oluştururlar (Bkz. Şekil 3.9). Bu düzlemsel yüzeyde paketlerin yükleme / boşaltma işlemlerinin yapılması için rıhtımlar bulunur. Hedef noktasına varması gereken yüklerin yolunun açılması için depolama süreci boyunca, diğer yükler yoğun bir şekilde paketlenip aynı anda hareket edebilirler. Buna ilişkin örnek şekil 3.10’da gösterilmiştir. Koyu renkle belirtilen örnek paketin belirtilen noktaya varabilmesi için dolaşma yolu üzerindeki diğer paketlerin yoldan uzaklaştırılması gösterilmiştir.



Şekil 3.9: Rıhtımlı bir DM örneği (Kay 2004).



(a) Yük taşınmadan önceki depolama alanı (b) Yük için yolun temizlendiği depolama alanı

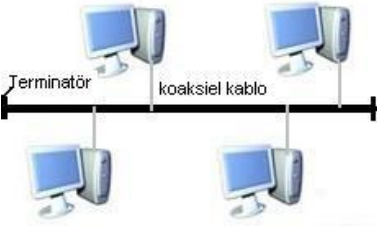
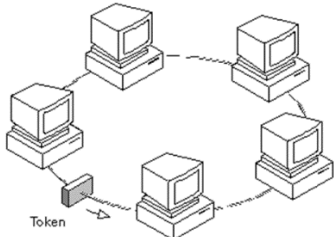
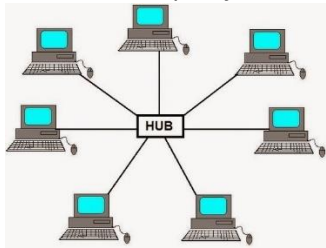
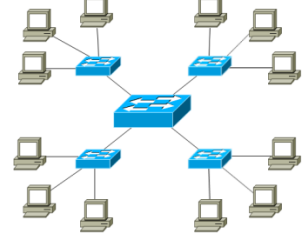
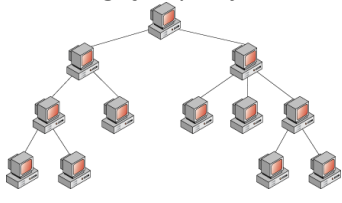
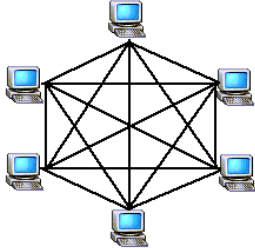
Şekil 3.10: DM içerisindeki bir yükün taşınması örneği (Kay 2004).

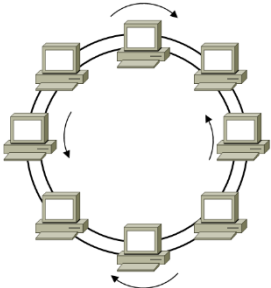
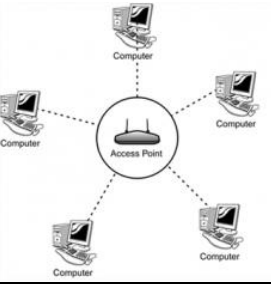
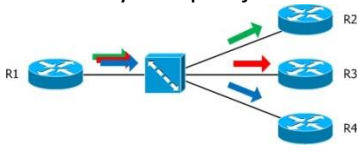
Tez çalışmasında ele alınan problem de birim boyutlu paketler üzerinde şekil 3.10 (b)'deki bir yol tanımlamaya ve otomatik DM geliştirmeye yöneliktir.

3.1.8 KLA ve OLA Arasındaki Farklar

OLA'nın en önemli özelliği tüm ağın tek bir firma tarafından kontrol edilmesidir. Ve bu ağdaki işlemlerin koordinasyonu için kullanılan teknolojilerin çoğu tescillidir. Uygun teknolojilerin ve özel tesislerin geliştirilmesi için gerekli olan geniş ölçekli operasyonlardan dolayı özel lojistik şirketlerin rekabetçi temel avantajı giriş için engeldir. Tek bir firma olması operasyon ölçeğini sınırlı hale getirmesinden dolayı dezavantaj yaratmaktadır. Sınırlı sayıda dağıtım merkezinin olmasından dolayı yükler hedef noktalarına varmadan önce gereksiz dolaşma yapabilmektedir. Bu dezavantajın bir sonucu olarak da ürün ve dağıtım arasındaki düzeni sağlamak için KLA kullanımının nasıl bir etki yaratabileceği sorusu ortaya çıkmıştır. KLA'nın en belirgin etkisi koordinasyon için tek bir firma gerekmemesi ve farklı operasyonların farklı firmalar tarafından gerçekleştirilme imkanı sağlamasıdır. Bu sayede firmalar ekonomik açıdan kazanç elde edebilirler (Kay ve Parlikad 2002). KLA ve OLA arasındaki farklar ağ topolojileri ile eşleştirilecek olursa OLA yıldız ağ topolojisine, KLA ise örgüsel ağ topolojisine benzer. Bu kavramların daha iyi anlaşılabilmesi ve KLA ve OLA arasındaki farkların iyi anlaşılabilmesi için Tablo 3.2'de ağ topolojileri ve açıklamaları gösterilmiştir.

Tablo 3.2: Ağ topolojileri (internet 5, internet 6).

<p>Yol Topolojisi</p> 	<p>Bütün makinelerin tek bir hat üzerinden iletişimi sağladığı topolojidir.</p> <p>Avantajları: Ağın kurulumu kolaydır. Ekonomiktir. Switch/hub gerektirmez.</p> <p>Dezavantajları: Ağa bağlanabilecek cihaz sayısı sınırlıdır. Omurga kablodaki bir sorun tüm ağı etkiler. Diğer topolojilere göre daha yavaştır.</p>
<p>Halka Topolojisi</p> 	<p>Halka biçiminde, tüm cihazların birbirine bağlandığı, verinin alıcı cihaza ulaşmaya kadar ağdaki tüm cihazları tek tek dolaştığı topoloji türüdür.</p> <p>Avantajları: Sunucuya ihtiyaç yoktur. Token kullanılarak daha büyük topolojiler yaratılabilir.</p> <p>Dezavantajları: Cihazlardan birinde oluşan sorun tüm ağı etkiler. Ekleme-çıkartma gibi işlemlerde tüm ağ etkilenir.</p>
<p>Yıldız Topolojisi</p> 	<p>Tüm cihazların merkezdeki ortak bir hub veya switch'e bağlandığı, en yaygın kullanıma sahip olan topolojidir.</p> <p>Avantajları: Ağdaki sorun tespiti ve ağın yönetimi kolaydır. Ağdaki cihazlardan birinde oluşacak sorun ağı etkilemez.</p> <p>Dezavantajları: Merkezdeki cihazda oluşacak sorun tüm ağı etkiler. Doğrusala göre daha fazla kablo gerektirir.</p>
<p>Gelişmiş Yıldız Topolojisi</p> 	<p>Yıldız topolojisinin genişletilmiş hali olup, merkezde bulunan hub veya switch'e yeni bir hub veya switch eklenmesi ile oluşturulur.</p> <p>Avantajları ve dezavantajları yıldız topolojisinde bahsedilenler ile aynıdır.</p>
<p>Ağaç Topolojisi</p> 	<p>Yıldız topolojisi ve yol topolojisinin bir arada kullanıldığı topoloji türüdür.</p> <p>Avantajları: Ağın genişletilmesi kolaydır. Dallardan birinde oluşacak sorun diğer cihazları etkilemez.</p> <p>Dezavantaj: Dallanma arttıkça ağın yönetimi zorlaşır. Kablolama işlemi zordur.</p>
<p>Örgüsel Topoloji</p> 	<p>Ağdaki her cihazın birbirine doğrudan bağlantılı olduğu ağ topolojisidir.</p> <p>Avantajları: Ağdaki bir cihaza bağlı oluşan sorun cihazın iletişimini kesmez. Cihazlar birbirine direk bağlı olduğu için veri iletim hızı yüksektir.</p> <p>Dezavantajları: Bağlantı sayısı çok ve maliyeti yüksektir. Karmaşık bir yapıya sahiptir.</p>

<p>Çift Halka Topolojisi</p> 	<p>Halka topolojisinden farkı birinci halkayı dıştan kuşatan ikinci bir halka bulunmasıdır. Bu sayede birbirine bağlı olan cihazlar arasında veri alışverişi sağlanır.</p>
<p>Hücresel Topoloji</p> 	<p>Her birinin kendi merkezi üzerinde birbirinden bağımsız düğümleri bulunan dairesel veya altıgen biçimindeki alanların oluşturduğu topoloji türüdür.</p> <p>Avantajı: Uzay boşluğu ve Dünya atmosferi hariç Herhangi bir taşıyıcı medya bulundurmazlar.</p> <p>Dezavantajı: Ortamda dolaşan sinyal izlenmeye açık bir durumda olması ve bunun getireceği güvenlik sorunlarıdır.</p>
<p>Yayın Topolojisi</p> 	<p>Gönderici cihazın veriyi ağa bıraktığı ve bu verinin diğer tüm cihazlara aynı anda iletilmesi kuralına dayanan topoloji yapısıdır.</p>

KLA ve OLA arasındaki farklar Tablo 3.3'te gösterilmiştir.

Tablo 3.3: KLA ve OLA karşılaştırması (Kay 2016).

Kamusal Lojistik Ağ	Özel Lojistik Ağ
Her bir kamyon ve DM farklı firmalar tarafından işletilebilir.	Paketin taşınmasını tek bir firma yürütür.
Her bir düğümün farklı düğümlerle bağlantılı olduğu örgüsel ağ yapısına benzer.	Her bir düğümün bir merkeze bağlı olduğu yıldız ağ yapısına benzer.
Merkezi olmayan kontrol	Merkezi kontrol
Modüler depolama nedeniyle değişen paket boyutlarından ortaya çıkan ölçek kayıpları	Geleneksel lojistik teknolojileri kullanıldığından ölçek ekonomileri

3.1.9 Eve Dağıtım Lojistik Ağı

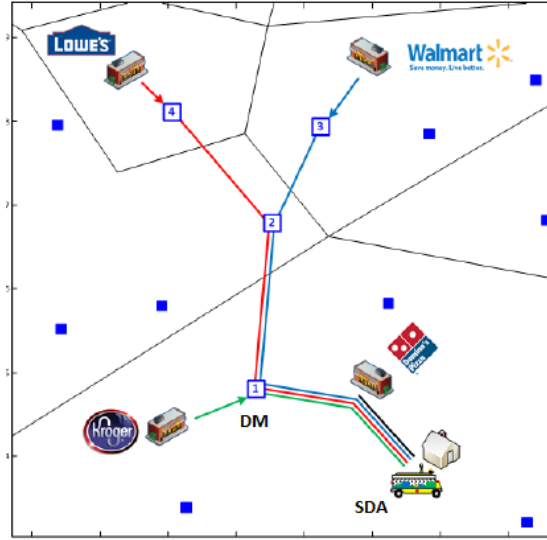
Eve dağıtım lojistik ağları (EDLA), yeniden kullanılabilir standarttaki sürücüsüz dağıtım araçları (SDA) ile yüklerin evlere dağıtımının yapılması ve bu sayede dağıtım maliyetlerini azaltmak için KLA temel alınarak geliştirilmiş lojistik ağlardır.

SDA, Google'ın geliştirdiği popüler sürücüsüz araçlardan farklı olarak alışveriş anlayışımızı çok önemli derecede değiştirebilecek bir sistemin parçasıdır. Küçük, pille çalışan, sürücüsüz bir otomobilin kargoya özel sürümü olan nakliye araçlarıdır. (internet 2). Dominos pizza zinciri bununla ilgili bir uygulama başlatmış ve İngiltere'de pizza siparişi veren müşterilere 'Domi-No-Drivers' (Şekil 3.11) ismi verilen iki tekerlekli sürücüsüz araçlarla teslimat yapmaya başlamıştır. Google ve amazon da paket ve malzeme taşımak için insansız hava araçlarıyla deneyler yapmış ancak Dominos'un kendi kendine çalışan motosikletleri otonom karayolu araçlarının ilk kullanımı olarak düşünülmektedir (internet 3).



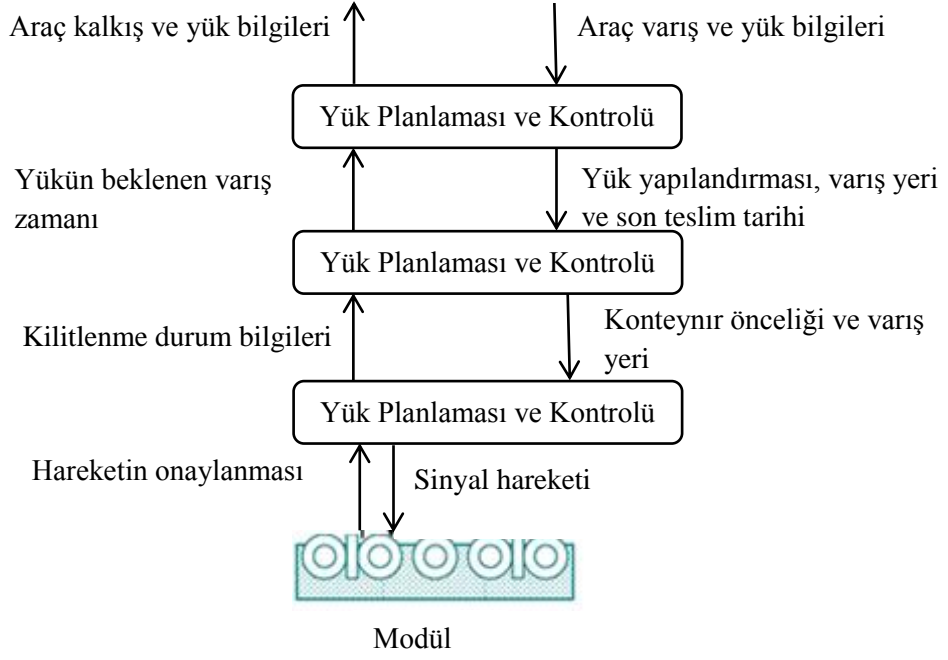
Şekil 3.11: Domi-No-Drivers.

EDLA'nda dağıtım; farklı firmalardan verilen siparişlerin tek tek müşteriye teslim edilmesi yerine şekil 3.12'de görüldüğü gibi sevkiyatların müşteri adresine en yakın noktadaki DM'de toplanıp sonrasında teslimatının yapılmasıyla gerçekleşmektedir (Kay 2013). Kay (2013)'ın tanımlamasına göre EDLA, sürücüsüz dağıtım araçları ve paket yükleme/boşaltma işlemlerinin yapıldığı tam otomatik depo sistemleri olmak üzere ayrılmaz 2 parçadan oluşmaktadır.



Şekil 3.12: EDLA örneği (Kay 2013).

Kay ve Parlikad (2002), Xiang ve diğ.(2007) Bansal (2004) çalışmalarında paket taşıyan araçların KLA'da nasıl bir işleyiş sürecine dair iş akışı sunmuşlardır. Bu tez çalışmasında ise KLA'yı oluşturan diğer bileşen üzerine yoğunlaşıp tam otomatik depo sistemleri için algoritma geliştirme amaçlanmıştır. Tam otomatik depo sistemi de başlık 3.1.7'de açıklanan dağıtım merkezleri yapısındadır. Yükleme/boşaltma, istifleme ve depo işlemlerinin otomatikleştirildiği depolardır. Yükselip alçalan tekerlekli modüllerden oluşur ve bu modül dizileri bir araya gelerek depodaki yüklerin ortogonal olarak taşınmasını sağlar. Bu depolar modüler depo olarak adlandırılmaktadır. Modüler deponun kontrolü için 3 katmanlı bir sistem kullanılır (Bkz. Şekil 3.13). Kontrolün ilk seviyesi yük planlamasıdır. Dağıtım aracı depoya vardığı zaman, içerisindeki farklı paketlerden oluşan yük aracın bilgileriyle beraber depoya boşaltılır. Yükleme bilgilerine (paketin hedef noktası, son teslim tarihi vb.) göre her bir paket için bir öncelik tanımlanır. Tanımlanan önceliklere ve kontrol algoritmasına göre bir yol planlanıp paketlerin hedef noktalarına hareketi gerçekleştirilir (Sittivijan 2013).



Şekil 3.13: 3 Katmanlı depo kontrol sistemi (Kay 2013).

4. Depo Yönetimi

Depo yönetimi sistemi, miktar ve depolama alanları ve özellikle de birbirleriyle olan ilişkilerinin yönetimi için tasarlanmış bir sistemdir. Depo yönetimi genellikle karmaşık depo ve dağıtım sistemlerinin kontrolü ve optimizasyonu anlamına gelir. Büyüklüklerin ve depolama yerlerinin yönetimi gibi bir envanter yönetiminin temel işlevselliğine ek olarak, nakliye araçlarının kontrolü ve planlaması, sistem durumunu kontrol etmek ve bir işletim ve en iyileme stratejisi seçmek için yöntemler ve araçları da içerir (Hompele ve Schmidt 2007). Bu bölümde depo ve depolama kavramları ile depo çeşitleri tanımlanmıştır. Depo ve depolamanın lojistik süreçler açısından önemine değinilmiştir.

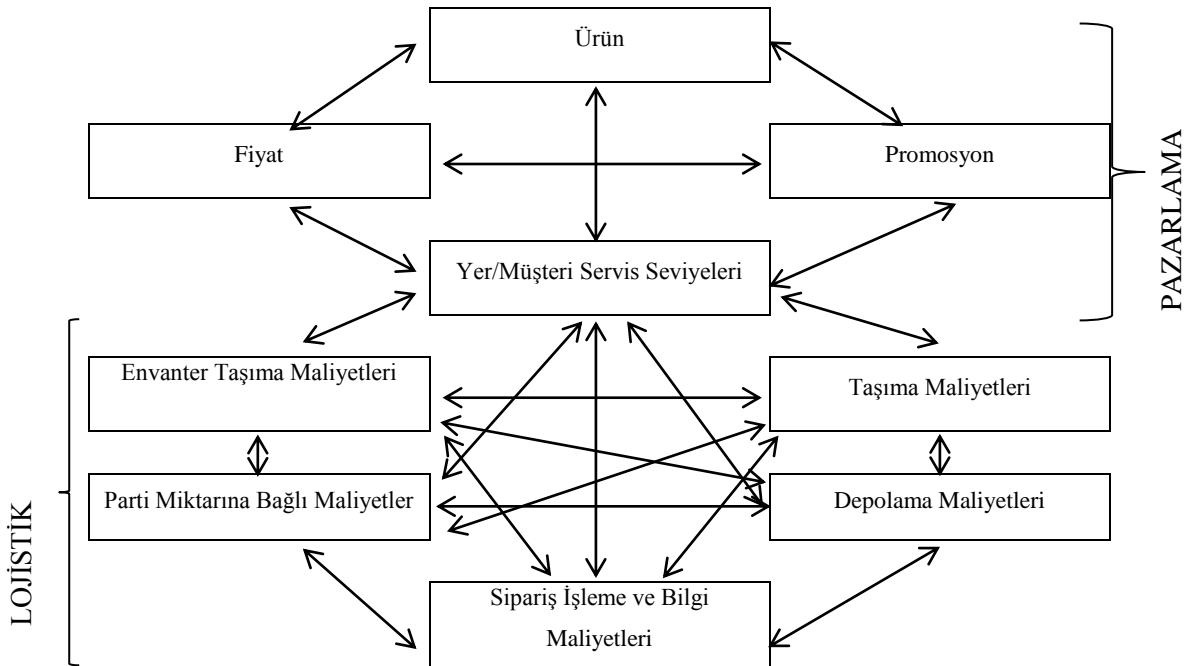
4.1 Depo ve Depolama

Depolama, ticari kazanç amacı bakımından çok eski tarihsel dönemlere kadar uzanmaktadır. Eski zamanlarda insanlar depolama kavramını acil durumlar için fazladan hayvan ve yiyecek tutmak olarak tanımlamışlardır. Uygarlıklar geliştikçe yerel depo tanımları ortaya çıkmış ve mallar nakliye, ticaret, üretim faaliyetleri ile bağlantılı olarak depolanmaya başlanmıştır. Bu gelişim devam edip, özellikle endüstri devrimi ile birlikte, üretimdeki olağan üstü artışlar ile beraber ulaşım yurtiçinden yurtdışına uzandığında depolar yerel depodan daha fazlası haline gelmiştir.

Depo; hammaddelerin, işlem görmekte olan (yarı mamul) veya bitmiş ürünlerin ihtiyaç anına kadar saklandığı (Kay 2015), ürünlerin başlangıç aşamasından müşteriye teslimine kadar olan süreçteki faaliyetler dizisinin gerçekleştirilmesinde kritik rol oynayan, tedarik zincirinin önemli bir bileşenidir (Öztürk 2011). Müşterinin ihtiyaç ve talepleri doğrultusunda, minimum maliyete ürünlerin muhafaza edildiği yerlerdir (Yener 2014). Depolar; tedarik zincirine, ürünün ihtiyaç olunan anda kullanılabilir olmasının depolama yolu ile sağlanması, ürünlerin verimli bir şekilde toplanması, sıralanması ve dağıtılması sağlanarak nakliyede ölçek ekonomisi sağlanması olmak üzere iki temel yolla katkıda

bulunmaktadır (Kay 2015). Yani depo genellikle stok tutmak veya depolamak gibi bir yer olarak görülse de aslında depo işlevselliği günümüz lojistik sisteminde envanter karma merkezi olarak görülebilir (Bowersox ve diğ. 2002). Hammadde ve yarı mamullerden nihai ürünlere kadar malların tedarik, üretim ve dağıtımının çeşitli aşamalarında yer alan depolar, yüksek müşteri hizmetleri seviyesinin sağlanması için büyük öneme sahiptir (Lambert ve diğ. 1998).

Depolama, lojistik süreçte maliyetlerin minimuma düşürülmesi, müşteri memnuniyetinin sağlanması açısından rekabet ortamında büyük öneme sahiptir (Şahin 2009). Depolama temel olarak depo-sevkiyat alanlarında gerçekleşen mal alımı, sipariş toplama, biriktirme ve nakliye gibi faaliyetler olmak üzere lojistik operasyonların birkaç yönünü içermektedir (Van den Berg ve Zijm 1999). Temel amacı, müşteri talepleri doğrultusunda düzenlenmiş büyük miktarlardaki malların depoya ve pazara olan hareketlerini kolaylaştırmaktır (Öztürk 2011). Depolama, ürünleri (hammadde, yarı mamul, nihai ürün) başlangıç ve tüketim noktası arasında saklayan ve bu ürünlerin yönetimi için durumuna, mahsulüne ilişkin bilgiler sağlayan bir lojistik sistemi parçası olarak tanımlanabilir. Şekil 4.1’te görüldüğü gibi, mümkün olan en düşük toplam maliyetle istenen seviyede müşteri hizmetinin sağlanmasında önemli bir rol oynamaktadır. Üretici ve müşteri arasındaki önemli bir bağlantıdır (Lambert ve diğ. 1998).

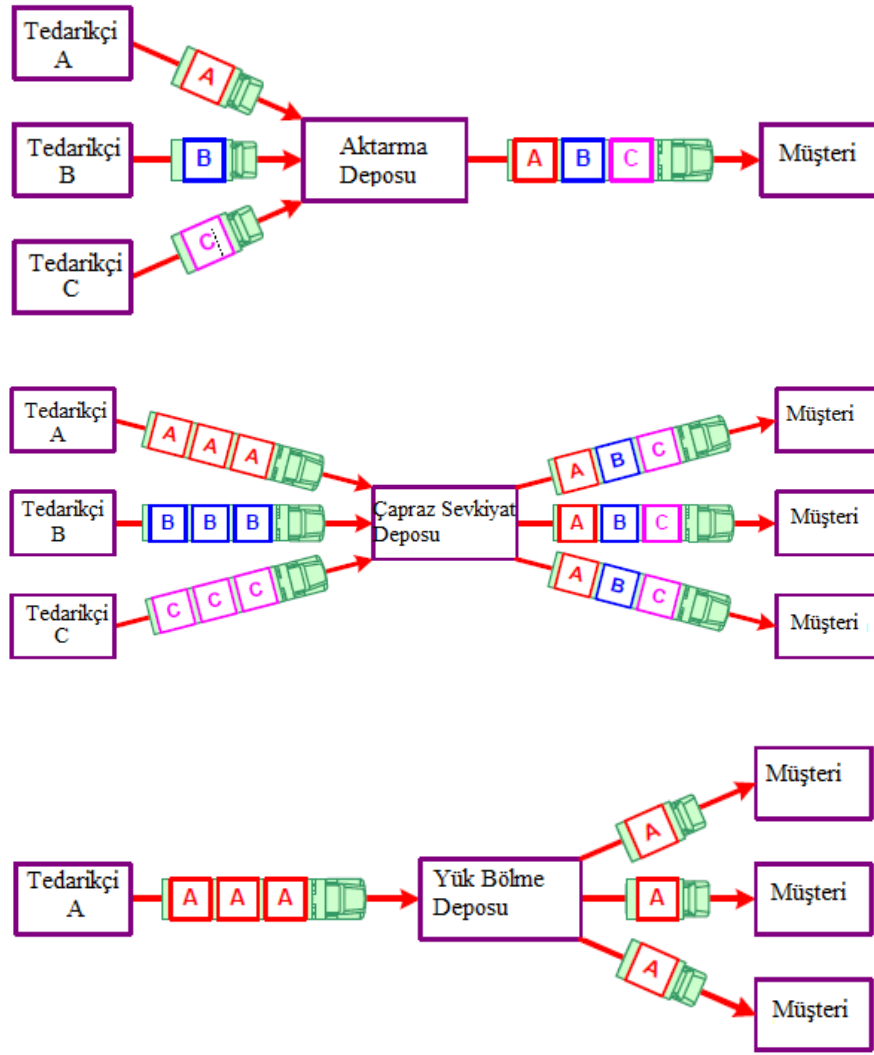


Şekil 4.1: Lojistik sistemdeki maliyetler (Lambert ve diğ. 1998.).

Depolamaya ihtiya duyulmasının sebepleri: (Lambert ve diğ. 1998)

- Nakliye giderlerinden tasarruf elde etmek
- Üretim ekonomilerinde kazanç elde etmek
- Tedarik kaynađını sürekli hale getirmek
- Firmanın müşteri servis politikalarını desteklemek
- Deđişen piyasa koşullarını uyum sağlamak (mevsimsellik, rekabet, talep dalgalanmaları... vb.)
- İstenen müşteri servis seviyesini en düşük toplam maliyetle elde etmek
- Her bir ürünün farklı siparişlerle müşteriye ulaştırılması yerine ürünlerin karıştırılmasını sağlamak gibi nedenler olarak sıralanabilir.

Depoların sadece envanter depolama görevleri yoktur. Arzın, müşteri talepleriyle daha iyi eşleştirilmesinde, nakliye masraflarının azaltılması ve daha iyi müşteri hizmetlerinin sunulması için ürünün çapraz sevkiyatının sağlanması gibi ekonomik açıdan pek çok alanda fayda sağlamaktadır (Bartholdi ve Hackman 2011). Transfer maliyetlerinde oldukça önemli bir etkiye sahiptirler ve hammadde/nihai ürün depoları, birleştirme (consolidation), çapraz sevkiyat, dökme gibi taşıma ekonomisini sağlamak için çeşitli amaçlarla kullanılmaktadır (Kay 2015). Şekil 4.2'de depoların fiziksel dağıtım sistemlerindeki kullanım alanlarından bazıları gösterilmiştir.



Şekil 4.2: Transfer maliyetini azaltmak için kullanılan depo çeşitleri (Kay 2015).

Birleştirme (consolidation) depoları farklı birçok yükü tek bir yük olarak birleştirmek amacıyla kullanılan depolardır. Tedarikçi ve depo arasındaki kısa mesafede küçük dağıtım kamyonları kullanılabilir, müşteriye transferdeki uzun mesafede ise römorklu tırlar kullanılabilir. Birleştirme (consolidation) depoları müşteriye daha düşük maliyetlerle tam kamyon yüklemesi (Full Truck Load) ile sevkiyat olanağı sağlar.

Çapraz sevkiyat tedarikçiden temin edilen malların depoya alınmadan tasnif edilerek müşterilerin ihtiyaçlarına göre sevk edilmesi işlemidir. Çapraz sevkiyatta, kamyonlarla gelen malzemeler çıkış kapısına yönlendirilerek bekletilmeden doğrudan doğruya ya da kısa bir süreliğine bekletilerek sevkiyatı gerçekleştirecek

kamyonlara yüklenir. Tam kamyon yükleme şeklinde sevkiyatın sağlanabilmesi amacıyla yük karması kullanır. Çapraz sevkiyat, lojistik süreçlerde tasarruf elde etmek ve rekabetçi avantaj kazanmak için uygulanabilecek önemli bir tedarik zinciri stratejisidir. Bu stratejide çapraz sevkiyat tesisleri geleneksel anlamda malzemelerin saklandığı depolar olarak değil transfer noktaları olarak görev alırlar. Bu tesisler malzeme depolamasını ortadan kaldırma amacına dönük olarak giren ürün akışının çıkan ürün akışı ile senkronize edildiği yerlerdir. Çapraz sevkiyat detaylandırılmış bir planlama ve koordinasyon gerektirdiği için her durumda uygun değildir.

Yük bölme (dökme, break-bulk) depo türünde depoya çok uzak mesafedeki tedarikçiden tam kamyon yükü şeklinde sevkiyat gelir ve burada küçük yüklere parçalanıp yakın uzaklıktaki her bir müşteriye dağıtılır. Belirli bir pazarda yapılacak küçük araç sevkiyatlarının ilk etapta bu pazarlara yakın bir depoya büyük parti sevkiyatları ile taşınması amaçlanır (Kay 2015).

4.2 Depolama Fonksiyonları

Her depo, parçası olduğu tedarik zincirinin özel ihtiyaçlarını karşılayacak şekilde tasarlanmalıdır. Bu özelliğe rağmen tüm depolar için ortak olan belli fonksiyonlar vardır. Lambert ve diğ. (1998)'ne göre depolama; taşıma, stoklama ve bilgi transferi olmak üzere 3 temel fonksiyondan oluşmaktadır.

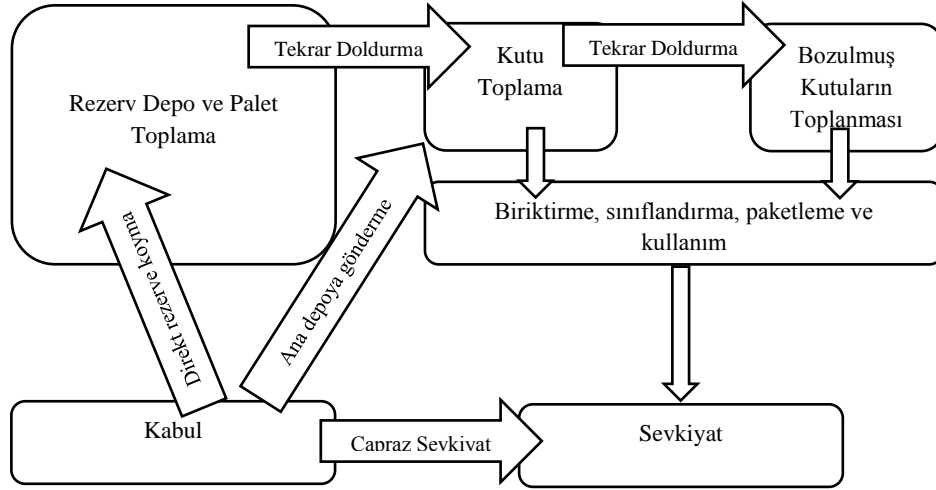
4.2.1 Taşıma Fonksiyonu

Firmalar, envanter dönüşlerini iyileştirmeye ve siparişleri üretimden son teslimata kadar hızlandırmaya odaklandığı için taşıma fonksiyonu şekil 4.3'te görüldüğü gibi bu 3 fonksiyon içerisinde en çok dikkat çeken fonksiyondur. Taşıma fonksiyonu,

- Mal kabul
- Taşıma / Yerleştirme
- Sipariş toplama / ayrıştırma
- Çapraz Sevkiyat

- Nakliye olmak üzere farklı iş süreçlerine bölünebilir.

Mal ve malzemeler büyük miktarlı gönderilerle depolara ulaşır ve buradaki ilk işlem boşaltmadır (Bowersox ve diğ. 2002). Mal kabul; gelen yüklerin boşaltılması, satın alma siparişlerinin kontrol edilmesi ve gelen mal kayıtlarının bilgisayar sistemine girilmesini içerir (Rushton ve diğ. 2014).

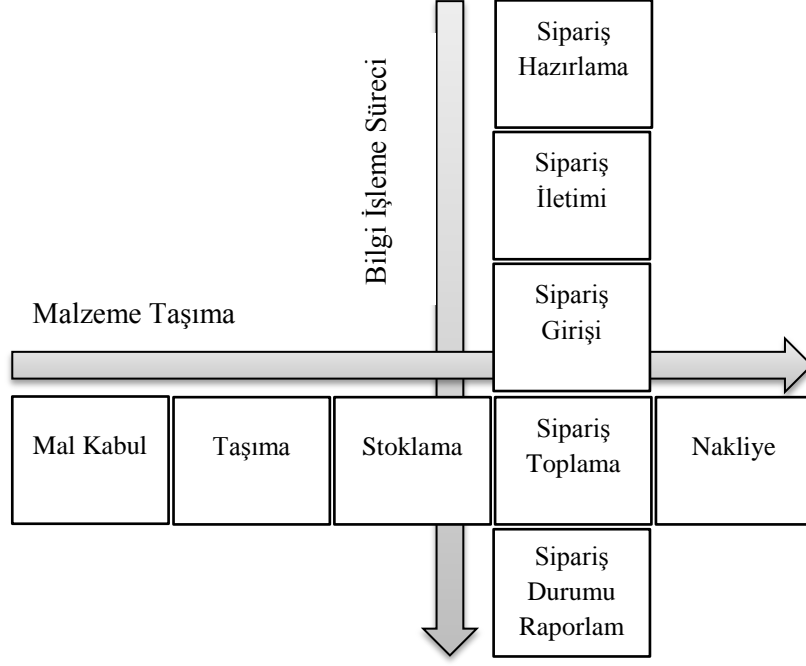


Şekil 4.3: Depo fonksiyonları (Şahin 2009).

Teslim alma işlemi, depoya gelen ürünlerin fiili olarak boşaltılmasını, depo stok kayıtlarının güncellenmesini, hasar kontrolü ve miktarda tutarsızlık olup olmadığının kontrolünü içermektedir (Lambert ve diğ. 1998). Malzemenin sınıflandırılması ve yeniden paketlenmesi de dahil olmak üzere, malların depolanmasına veya çapraz sevkiyatına hazırlık amacıyla bir deponun içine nakledilen malzemelerin boşaltılması, doğrulanması, denetlenmesi ve yeniden paketlenmesi işlemlerini kapsar (Kay 2015).

Transfer ve yerleştirme işlemi, ürünün depoya depolanması veya çapraz sevkiyat gibi durumlarda doğrudan nakliye alanına taşınması fiziksel hareketleri içerir (Kay 2015) (Lambert ve diğ. 1998).

Sipariş toplama, müşteri veya mağaza siparişlerine karşılık arzu edilen malzemelerin gruplanıp depodan çıkarılma sürecidir. Sipariş toplama, Şekil 4.4'te gösterildiği gibi depo ve sipariş işleminin kesişim noktasındadır (Kay 2015).



Şekil 4.4: Sipariş toplamanın depolama ve sipariş işlemedeki yeri

Çapraz sevkiyat, teslim alınan malların doğrudan sevkiyat noktasına aktarılması işlemidir.

4.2.2 Stoklama Fonksiyonu

Deponun temel işlevi malların belirli bir süre tüketicilerin taleplerinin karşılanması için gruplandırılması ve saklanmasıdır (Hopbağlı 2009). Depolamanın ikinci fonksiyonu olan stoklama, malların tüketiciler tarafından talep edildikleri zamanda onlara sunulmasıdır ve bu fonksiyon geçici veya yarı-kalıcı olarak uygulanabilir. Geçici depolama, yalnızca temel envanter yenilemesi için gerekli olan ürünün depolanmasını içerirken, yarı-kalıcı depolama envanterin normal yenilenme için gereken miktardan daha fazla depolanması kavramıdır. Daha önce de bahsedilen çapraz yerleştirmenin amacı, deponun yalnızca geçici stoklama işlevini kullanmaktır. Yarı-kalıcı depolama emniyet stok olarak da adlandırılır. Yarı-kalıcı depolamaya sebep olan yaygın koşullar mevsimsel talep dalgalanmaları, ileri alım, düzensiz talep gibi sıralanabilir (Lambert ve diğ. 1998).

4.2.3 Bilgi Transfer Fonksiyonu

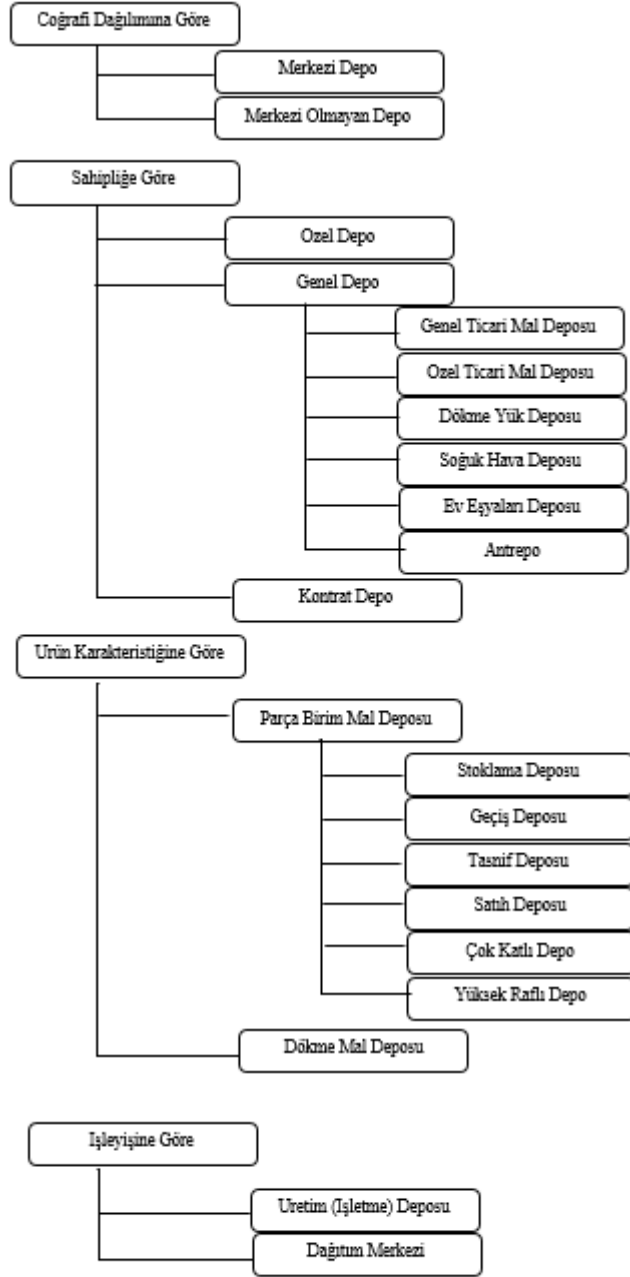
Depolamanın üçüncü önemli fonksiyonu olan bilgi transferi, taşıma ve depolama işlevleriyle eş zamanlı olarak gerçekleşir. Depo işlevlerinin başarılı bir şekilde işletilebilmesi için stok seviyeleri, üretim seviyeleri, stok tutma yerleri, müşteri bilgileri, gelen ve giden gönderiler, tesis alanı kullanımı ve personel ile ilgili bilgiler önemli bir yere sahiptir çünkü bilgi transfer fonksiyonu depolama faaliyetlerini yönetmeye çalıştığından dolayı her zaman doğru bilgiye ihtiyacı vardır.

Depodaki taşıma, stoklama ve bilgi transferi fonksiyonlarındaki herhangi bir verimsizliği ortadan kaldırmak önemlidir. Bu verimsizlikler; yedekli veya aşırı kullanım, boş alanların kötü kullanımı, aşırı bakım maliyetleri, eskimiş ekipman yüzünden oluşan amortismanlar, teslim alma ve nakliye liman koşulları, rutin işlemler için güncelliğini yitirmiş bilişim sistemlerinin kullanımı gibi sebeplerden kaynaklanabilir. Bilgi transfer fonksiyonunda da bu verimsizlikler evrak akışlarının azaltılmasına yönelik büro yönetimi otomasyon çalışmaları, elektronik veri değişimi ve barkod kullanan bilgisayarlı bilgi aktarımına gidilmesi ile ortadan kaldırılması sağlanır (Lambert ve diğ. 1998).

4.3 Depo Çeşitleri

Depolar ile ilgili bilimsel yazın incelendiğinde farklı açılara göre sınıflandırmanın olduğu ve pek çok depo türünün ortaya çıktığı görülmektedir. Ballou (2004) ile Lambert ve diğ. (1998) ürün şekline göre değişik hizmetler sunan genel depoları; genel ticari mal deposu, özel ticari mal deposu, dökme yük deposu, soğuk hava deposu, ev eşyaları deposu ve antrepo olarak alt sınıflara ayırmışlardır. Bowersox ve diğ. (2002), depoları sahipliğine göre özel, genel ve kontrat depolar olarak sınıflandırmıştır. Güler (2006), çalışmasında depoları coğrafi dağılımına göre merkezi ve merkezi olmayan depo olarak iki alt sınıfa ayırmıştır. Ghiani ve diğ. (2003) de, depolama kavramı kullanılıyorsa merkezi veya merkezi olmayan sistemden biri olması gerektiğini ileri sürmüştür. Merkezi depolamada, tek bir mal deposu tüm pazara hizmet ederken, merkezi olmayan depoda ise, piyasa, her biri farklı (daha küçük) bir depo tarafından sunulan farklı bölgelere ayrılır. Depolama alanlarının müşterilere çok daha yakın olması nedeniyle, merkezi olmayan depo,

teslimat sürelerinin azalmasına neden olur. Öte yandan, merkezi depolama, daha büyük ekonomi nedeniyle daha düşük tesis maliyetleri ile karakterize edilir. Bu sınıflandırmalardan yola çıkarak Şekil 4.5'teki gibi bir depo sınıflandırması gösterilebilir.



Şekil 4.5: Depo çeşitleri.

4.4 Modüler Depo Yönetimi

Bu bölümde modüler depo tanımlanmış ve bu sistem içerisindeki işleyiş ve bu işleyişe ait algoritmalar açıklanmıştır.

4.4.1 Modüler Depo Tanımı

Tam otomatik depolama sisteminin gerçekleştirilebilmesi için yükleme/boşaltma, ayırma ve depolama faaliyetleri açısından tam otomatik bir depolama sisteminin olması ve sistemin çeşitli paket boyutlarının herhangi bir yere herhangi bir zamanda taşınmasına izin verecek kadar esnek olması gerekmektedir. Bu gereksinimlerin karşılanabilmesi için Kay (2013), Bölüm 3.1.7’de de bahsedildiği gibi, KLA’da kullanılan DM’lerin özelleştirilmiş bir hali olan modüler depo kavramını tanımlamıştır. Modüler depo dikey olarak alçalıp yükselen kare modüllerden oluşan tam otomatik depo sistemidir.

4.4.2 Modüler Depo Kavramı ve Kısıtlamaları

Modüler depo $m \times n$ boyutlarında, birim kare modüllerden oluşan bir depo sistemidir (Bkz. Şekil 4.6). Her modülde ya bir nesne vardır ya da o modül boştur. Sistemde aktif ve aktif olmayan olmak üzere iki tür nesne bulunmaktadır. Her nesnenin kendine özgü bir öncelik değeri vardır. Aktif olmayan nesnelere farklı olarak aktif nesnelerin bir hedef noktaları ve başlangıç konumlarından hedef noktalarına varabilmeleri için tanımlanan özel güzergâhları bulunmaktadır. Sistemde normal koşullarda aktif olmayan nesnelerin hareket etmesi durumu yokken, eğer bu nesne aktif bir nesnenin tanımlanan güzergâhı üzerindeyse aktif nesnenin yolundan uzaklaştırılması için hareket ettirilir. Modüler depoda bir nesne ancak kendisinden daha düşük öncelikli nesnelere hareket ettirebilir.

16	5	1	4	
		11	6	
2		9		
3				25

Aktif Paket

Aktif Olmayan Paket

Boş Modül

Şekil Şekil 4.6: Modüler depo örneği

4.4.3 Temel İşleyiş

Modüler depo yönetimi için rota planlama, ana kontrol ve etiketleme süreci olmak üzere 3 ana bölümden oluşan bir sistem gerekmektedir. Ana kontrol, aktif ve aktif olmayan tüm nesnelerin hareketinin yönetildiği ve sistemde oluşabilecek kilitlenmelerin saptandığı süreçtir. Rota planlama; her aktif objenin, başlangıç konumundan tanımlı olan hedef noktasına varışı için gerekli olan yolu bulmada kullanılan süreçtir. Etiketleme süreci ise aktif bir nesnenin tanımlanan yolunun düşük öncelikli bir aktif nesne veya aktif olmayan nesne tarafından işgal edilmesi durumunda o nesnenin yoldan uzaklaştırılması için gerçekleştirilen süreçtir. Bölüm 3'te bu süreçlerle ilgili detaylara yer verilmiştir.

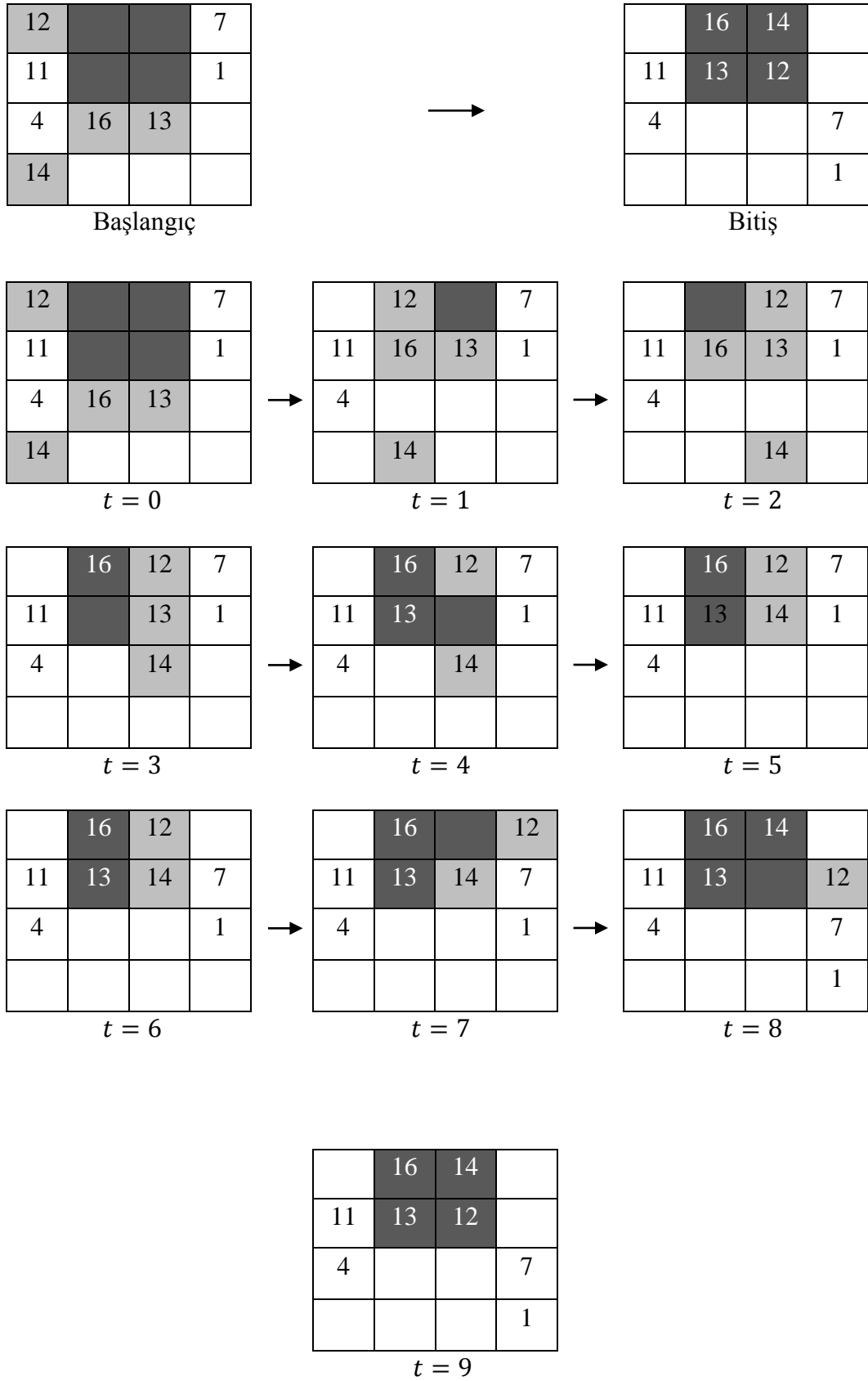
5. ÖNERİLEN ÇÖZÜM YAKLAŞIMI

Bu bölümde; çalışmada ele alınan problem kısaca tanımlanmış, incelenen bilimsel yazında yer alan algoritmalar, kullanılan veri kümeleri açıklanmış ve önerilen sezgisel algoritmalar test problemleri üzerinde koşturularak karşılaştırma sonuçları tablo ve grafiklerle gösterilmiştir.

5.1 Problemin Tanımı

Çalışma kapsamında modüler depo sistemi içerisindeki paketlerin hareketinin planlanması ile beraber çarpışma ve kilitlenmelerin önüne geçilmesi için A* temelli sezgisel bir yöntemin geliştirilmesidir. Sittivijan (2015) ve Datar (2011) tarafından önerilen algoritmalarla yola çıkarak A* temelli yeni bir sezgisel geliştirilmiştir.

Datar (2011) tarafından önerilen algoritma açgözlü bir yapıda olup sadece bir sonraki zaman dilimindeki hareketin planlandığı bir algoritmadır. Bu yöntemde ilerleyen aşama için rota planlaması yapılmamaktadır. Bir sonraki modülü arama süreci, etiketleme ve tüm paketlerin hareketinin kontrol edildiği ana kontrol sürecinden oluşmaktadır. Tanımlama, bilgi alımı, iletişim ve hareket adımları ana kontrol sürecinin parçalarını oluşturmaktadır. Bir sonraki modülü arama sürecinde öncelikle mevcut modülün tüm komşularına bakılır. Komşularından hedefe olan uzaklığı minimum olup önceliği mevcut nesneden daha düşük olan ve başka bir aktif nesne tarafından etiketlenmemiş olanları seçilir. Eğer birden fazla minimum değere sahip komşu varsa içlerinden en düşük öncelikli komşu modül bir sonraki zaman diliminde hareket edilecek modül olarak seçilir. Eğer seçilen modül başka düşük öncelikli bir nesne ile dolu ise etiketleme süreci başlatılır. Etiketleme süreci Sittivijan (2015)'nin çalışmasında önerilen algoritma ile aynı olup 5.2 başlığı altında detaylı bir şekilde açıklanmıştır. Datar (2011)'in algoritmasının çalışma süreci 4×4 boyutundaki bir örnek üzerinde incelenmiş ve $t = 0$ çözüm adımından $t = 9$ çözüm adımına kadar olan tüm adımları Şekil 5.1'de tek tek gösterilmiştir. Koyu gri renkler hedefleri, açık gri renkler aktif nesnelere ifade etmektedir.

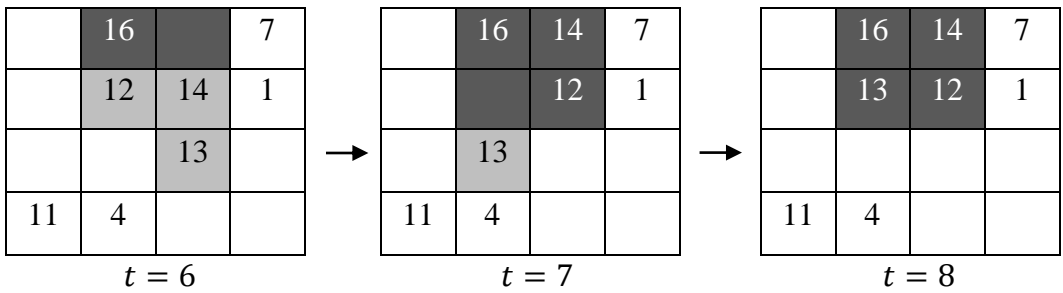
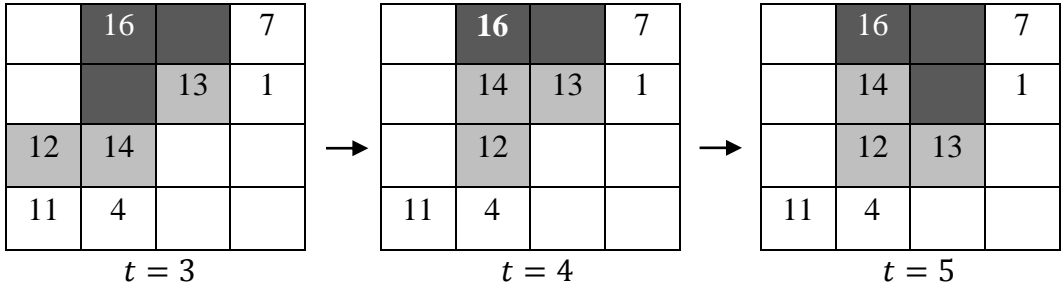
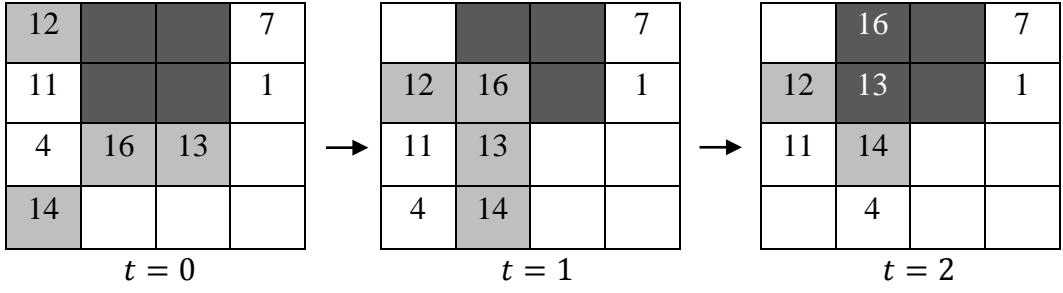
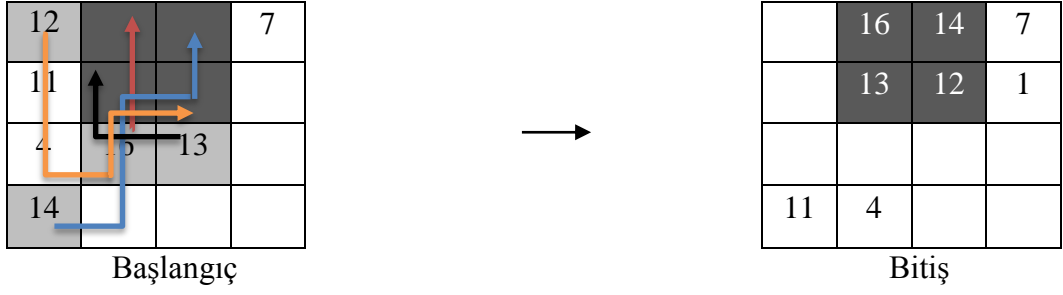


Şekil 5.1: ALG-B2 çözüm adımları

Sittivijan (2015)'in önerdiği algoritma, sezgisel temelli bir algoritma olup rota planlama, etiketleme ve ana kontrol süreçlerinden oluşur. Rota planlama süreci A* sezgiseline dayanılarak geliştirilen bir algoritmadır. A* algoritması statik bir ortamda kullanılan bir sezgisel olduğundan alt sınır hesaplamasında formül 5.1'de görüldüğü bir değişiklik yapılmıştır. Ayrıca alt sınır değeri hesaplanırken sadece bir sonraki aşama değil, k çözüm adım sayısı içerisindeki durumlarda göz önünde bulundurulmuştur. Önerilen yaklaşım, Sittivijan (2015)'nin algoritmasında geliştirmeler yapılarak gerçekleştirildiği için bu formülün açıklanmasına ilişkin bilgiler 5.2 başlığı altında detaylı bir şekilde açıklanmıştır.

$$LB_{(x,y)}^T = T_{(a,b)}^{(a_0,b_0)} + T_{(x,y)}^{(a,b)} + T_{(a_n,b_n)}^{(x,y)} \quad (5.1)$$

Sittivijan (2015), tarafından geliştirilen A* temelli sezgiselle göre, depoda bulunan tüm aktif nesnelere için sadece başlangıçta bir rota hesaplanmış ve nesnelere bu rotaya göre hareket ettirilmiştir. Ancak aktif nesnelere herhangi biri etiketleme sürecine dâhil edilmiş ve planlanan rotasından çıkmışsa onun için yeniden rota planlama süreci uygulanmıştır. Etiketleme süreci nesnelere önceliğine göre gerçekleştirilen bir süreç olup önerilen sezgiselle aynı şekilde çalışmaktadır. Ana kontrol süreci de aktif ve aktif olmayan nesnelere hareketlerinin denetlendiği, etiketleme ve rota planlama süreçlerinin işleyiş kontrollerinin yapıldığı kısımdır. Her zaman adımında ana kontrol süreci hedef noktasında olmayan her aktif nesnenin yüksek öncelikli başka bir nesne tarafından etiketlenip etiketlenmediğini ve planlanan rotasında olup olmadığını kontrol eder. Eğer etiketli değilse ve planlanan rotasında ise herhangi bir işlem yapılmaz. Ancak etiketli değil ama planlanan rotasında da değilse mevcut aktif nesne için yeniden rota planlaması gerçekleştirilir. Sittivijan (2015)'in algoritmasının çalışma süreci 4×4 boyutundaki bir örnek üzerinde incelenmiş ve $t = 0$ zaman adımından $t = 8$ zaman adımına kadar olan tüm adımları Şekil 5.2'de tek tek gösterilmiştir. Koyu gri renkler hedefleri, açık gri renkler aktif nesnelere ifade etmektedir. Her aktif nesne için başlangıçta tanımlanan yollar oklarla belirtilmiştir.



Şekil 5.2: ALG-B1 çözüm adımları

$t = 2$ zaman adımından $t = 3$ zaman adımına geçişte 14 numaralı paketin hareket edeceği modülde 13 numaralı paket bulunmaktadır. 14 numaralı paket 13'ten daha yüksek önceliği sahip olduğu için etiketle sürecini başlatır ve 13 numaralı paketi hareket rotasından uzaklaştırır. Etiketleme süreci boyunca 13 numaralı paket için yeniden yol işlemi yapılmaz. $t = 6$ zaman adımına gelindiğinde 13 numaralı paket etiketleme sürecinden ayrılır ve planlanan rotasında olmadığı için tekrar rota planlama uygulanıp hedefine ulaştırılması sağlanır.

5.2 Önerilen Yaklaşım

Önerilen sezgisel yaklaşım bölüm 4.4'te açıklandığı üzere A* temelli rota planlamanın yapıldığı rota planlama süreci, yüksek öncelikli nesnenin planlanan yolunda kendisinden daha düşük öncelikli veya aktif olmayan bir nesne ile karşılaşması durumunda uygulanacak olan etiketleme süreci ve tüm bu süreçlerin kontrol edildiği ana kontrol sürecinden oluşmaktadır.

5.2.1 Rota Planlama

Rota planlama, her aktif nesne için başlangıç konumundan hedef noktaya hareketindeki yolun bulunmasında kullanılmaktadır. Mevcut noktadan hedef noktaya olan yolun bulunmasında mevcut modülün etrafındaki ortogonal komşu modüller incelenir. Hareket edilecek bir sonraki modülün seçilmesinde A* temelli bir sezgisel yöntem kullanılmaktadır. Ama çalışmada ele alınan depo sistemi, eş zamanlı olarak paketlerin hareket edeceği bir sistem olduğundan dolayı statik bir çevre söz konusu değildir. A* algoritması ise statik bir çevrede kullanıldığından dolayı A* algoritması üzerinde geliştirmeler yapılarak kullanılmıştır.

A* algoritması başlangıç noktasından hedef noktaya olan en az maliyetli yolu bulmada kullanılan sezgisel bir algoritmadır. A* algoritmasında başlangıç durumundan hedef noktaya doğrudan bir maliyet hesaplanır. Algoritma, ilerleme yönünü bu maliyetin artıp azalmasına göre ayarlamaktadır. Böylece karmaşık ortamlarda dahi, algoritma $G(n)$ fonksiyonunu kullanarak hedeften uzaklaştığını anlayabilir.

$$F(n) = G(n) + H(n) \quad (5.2)$$

$F(n)$: Amaç fonksiyonu. Az olması o durumun daha az maliyetli olduğunu belirtir.

$G(n)$: Başlangıç noktasından hedef noktaya ulaşmak için gidilmesi gereken yolun maliyeti.

$H(n)$: Verilen bir noktadan hedef noktasına ulaşmak için gidilmesi gereken yolun sezgisel maliyeti.

Algoritmanın çalışması esnasında öncelikle mevcut düğümün komşu düğümleri ziyaret edilir. Ardından $F(n)$ değeri en düşük olan düğüm öncelikli olmak üzere hedef düğüm bulunana kadar ilerlenmeye devam edilir. Her bir adımda, bir önceki düğüm, gidilen düğümün ebeveyni olarak işaretlenir. Böylece hedef düğüme ulaşıldığında düğümlerin ebeveynleri takip edilerek başlangıç ve hedef düğümleri arasındaki yol elde edilir. Gidilmesi muhtemel düğümler, algoritma içerisinde açık liste isimli bir listede tutulmaktadır. Açık listede bulunan ve ziyaret edilen düğümler tekrar kontrol edilmemesi için kapalı listeye alınmaktadır. Kapalı listede yer alan bir düğüm, daha kısa bir yol bulunması durumunda tekrar açık listeye alınmaktadır. Açık liste içerisindeki düğümler, başlangıç düğümünden hedef düğüme kadar olan toplam maliyetlerine göre sıralıdır. Böylece öncelik sıralamasına sahip bir liste elde edilir. Hedef düğüme ulaşıldığında veya açık listede düğüm kalmayınca algoritma sonlandırılır. A* algoritmasının işleyişine yönelik açıklama Tablo 5.1'de gösterilmiştir.

Tablo 5.1: A* algoritması.

A* Algoritması
<p>Başlangıç açık ve kapalı liste oluştur $g, f=0$ Açık_liste={başlangıç_modülü} (f değerine göre sıralı tutulur.) While açık_liste boş olmadığı sürece Mevcut_modül=Açık_listenin ilk elemanı Açık_listedeki ilk elemanı sil If mevcut_modül=hedef_modül Break Else Mevcut_modülün komşularını bul For her komşu için Komşu.g=mevcut_modül.g + komşu ve mevcut_modül arasındaki uzaklık (h değeri “Manhattan” uzaklığı olarak hesaplanmıştır.) Komşu.h=komşu ve hedef modül arasındaki uzaklık Komşu.f=komşu.g+komşu.h If komşu açık_listede değil Açık_listeye ekle Komşu.ebeveyn=mevcut_modül Mevcut_modülü kapalı_listeye ekle</p>

Ele aldığımız problemde ise her bir nesne için böyle sabit değerler kabul etmemiz olası değildir. Çünkü depoda nesnelerin yeri zamanla değişmekte dolayısıyla statik değil dinamik bir ortam söz konusudur. Bu nedenle $F(n)$ fonksiyonu yerine $LB_{(x,y)}^T$ isimli yeni bir fonksiyon kullanılmıştır. $LB_{(x,y)}^T$ değeri en küçük olan modül, hareket ettirilecek modül olarak seçilir.

$$LB_{(x,y)}^T = T_{(a,b)}^{(a_0,b_0)} + T_{(x,y)}^{(a,b)} + T_{(a_n,b_n)}^{(x,y)} \quad (5.3)$$

Formül 5.3'e göre hareket ettirilecek nesnenin mevcut konumu (a, b) noktası ve nesnenin hedeflenen konumu (x, y) noktası varsayılmaktadır. İlk değişken, başlangıç noktasından (a_0, b_0) herhangi aracı modül (a, b) 'ye ulaşmak için olan dolaşma süresidir. İkinci değişken bir sonraki k adım boyunca komşu modül (x, y) 'ye gitmek için ağırlıklandırılmış tahmini dolanma süresidir. Sistemdeki nesnelerin yapılandırması bir zaman adımından diğerine değişebileceğinden dolayı formül 5.4'te tanımlandığı gibi her bir t çözüm adımı için ağırlıklandırılmış toplamları hesaplanır.

$$T_{(x,y)}^{(a,b)} = \sum_{t=1}^k w_t \times T_{(x,y),t}^{(a,b)} \quad (5.4)$$

$T_{(x,y),t}^{(a,b)}$ değeri hesaplanırken komşu modülün mevcut zaman adımından ($t = 1$) k zaman adımına kadarki ($t = k$) süreç boyunca doluluk boşluk durumu dikkate alınır. Çünkü sistemdeki nesnelerin durumu bir zaman adımından başka bir zaman adımına kadar büyük değişimler gösterebilir. Yalnızca geçerli zaman adımındaki durumu göz önüne alarak bir yol planlamak, aktif bir nesnenin mevcut konumu ile hedefi arasında etkili bir yol oluşturulmasını sağlamaz. Bu çalışmada rota planlama ufku $k = 3$ olarak alınmış ve her paket için rota planlamada mevcut zaman adımından itibaren 3 zaman adımı içerisindeki sistem durumu değerlendirilmiştir. $t = k$ zaman adımında eğer komşu modül boş ise mevcut modül komşu modüle bir zaman adımında ($T_{(x,y),t}^{(a,b)} = 1$), eğer komşu modül düşük öncelikli bir nesne ile doluyorsa o modüle geçiş bir veya daha fazla zaman adımında ($T_{(x,y),t}^{(a,b)} \geq 2$) gerçekleşir. Eğer komşu modül hedefine varmamış yüksek öncelikli bir nesne ile doluyorsa yüksek öncelikli nesne komşu modülden hareket edene kadar beklenir. Ama komşu modülde hedefine varmış yüksek öncelikli nesne varsa hareket ettirilemez ve bu değişken sonsuz değerini ($T_{(x,y),t}^{(a,b)} = \infty$) alır. 5.4 formülündeki w_t değeri keyfi seçilen değerlerdir ama $\sum_{t=1}^k w_t = 1$ ve $w_t > w_{t-1}$ koşullarını sağlaması gerekmektedir.

$T_{(a_n,b_n)}^{(x,y)}$ değişkeninin hesaplanmasında Manhattan uzaklığı yöntemi kullanılır. Komşu modülden hedef noktaya olan uzaklık değişkeni formül 5.5 ile hesaplanır.

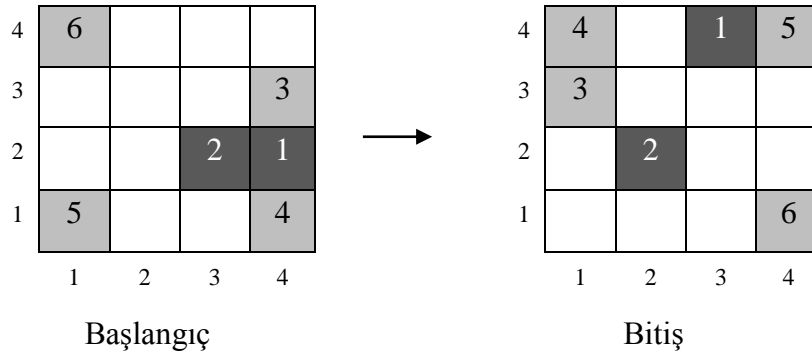
$$T_{(a_n,b_n)}^{(x,y)} = |x_{komşu} - x_{hedef}| + |y_{komşu} - y_{hedef}| \quad (5.5)$$

$LB_{(x,y)}^T$ değeri (a, b) noktasından (x, y) noktasına gitmek için tanımlanacak yolda kullanılan alt sınır değeridir. A* algoritmasında nasıl $F(n)$ değeri en küçük olan komşu seçiliyorsa burada da mevcut modülün ortogonal komşularından en küçük $LB_{(x,y)}^T$ değerine sahip komşu seçilmektedir.

Şekil 5.3'te 4×4 lük bir depo örneği gösterilmiştir. 3, 4, 5 ve 6 numaralı nesnelere aktif nesnelere ve bu örnek sistem için $LB_{(x,y)}^T$ değerlerinin nasıl

hesaplanacağı açıklanmıştır. Başlangıçta yapılan rota planlama ile aktif nesnelerin tanımlanan yol koordinatları sırasıyla şu şekilde belirlenmiştir:

- 6 : (1,4) → (2,4) → (3,4) → (4,4) → (4,3) → (4,2) → (4,1)
 5 : (1,1) → (2,1) → (3,1) → (3,2) → (3,3) → (3,4) → (4,4)
 4 : (4,1) → (4,2) → (4,3) → (3,3) → (2,3) → (1,3) → (1,4)
 3 : (4,3) → (3,3) → (2,3) → (1,3)



Şekil 5.3: 4 x 4 depo örneği

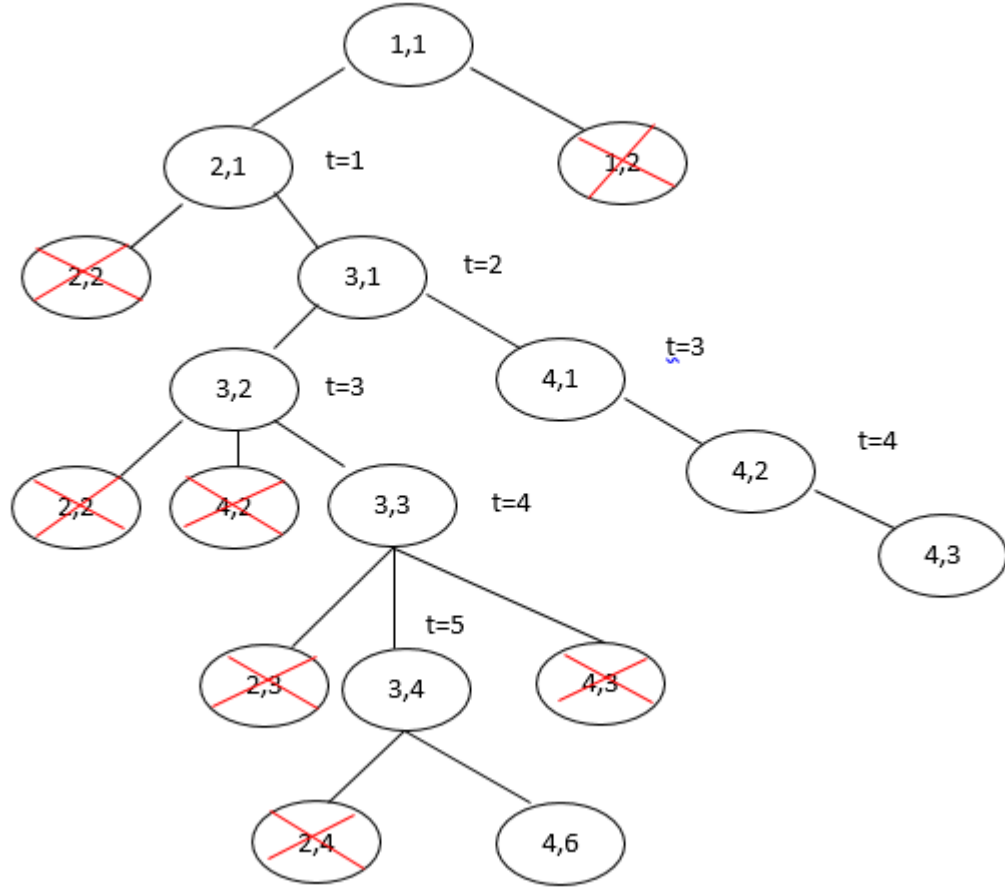
Tablo 5.2’de 5 numaralı nesne için (1,1) konumundan hedef noktası olan (4,4) konumuna olan yolun tanımlanmasında kullanılacak olan LB değerlerinin hesaplanmasına ilişkin matematiksel işlemler ayrıntılı olarak gösterilmiştir.

Tablo 5.2: LB hesabı için çözüm örneği.

5 numaralı örnek için LB değeri hesaplaması	
$LB_{(2,1)}^1 = 1+1+5 = 7$	$LB_{(4,2)}^4 = 12 + LB_{(4,2)}^{(4,1)} + 2 = 19.1813$
$LB_{(1,2)}^1 = 1+1+5 = 7$	$LB_{(4,2)}^{(4,1)} = 0.545 * 1 + 0.2727 * 1 + 0.1818 * 2$
$LB_{(2,2)}^2 = 7+1+4 = 12$	$LB_{(4,3)}^5 = 19.1813 + LB_{(4,3)}^{(4,2)} + 1 = 23.1808$
$LB_{(3,1)}^2 = 7+1+4 = 12$	$LB_{(4,3)}^{(4,2)} = 0.545 * 2 + 0.2727 * 1 + 0.1818$
$LB_{(3,2)}^3 = 12+1+3 = 16$	
$LB_{(4,1)}^3 = 12+1+3 = 16$	
<u>Açık Liste</u>	<u>Kapalı Liste</u>
(1,2), (2,1), (2,2), (3,1), (3,2), (4,2), (4,3)	(1,1), (2,1), (3,1), (4,1), (4,2), (3,2)
	t=1 t=2 t=3 t=4 t=3
	↓
	Daha düşük LB değeri olduğu için geri döner.

Tablo 5.3: Rota planlamada komşular arası dolaşma.

5 numaralı nesne için modüller arası dolaşımın düğümlerle gösterimi



Rota planlama sürecinin işleyişine yönelik açıklama Tablo 5.5'te gösterilmiştir.

Tablo 5.4: Rota planlama algoritması.

Rota Planlama

SET mevcut_modül=başlangıç_modülü, dolaşılacak listesi={mevcut_modül}, ebeveyn modül={}

While mevcut modül hedef modüle varmadığı sürece

Mevcut modülün komşularını tanımla

Komşuların daha önce dolaşılacak listesinde olup olmadığını kontrol et

If ziyaret edilmemiş komşular varsa

For gelecek k zaman adımında ziyaret edilmemiş olan tüm komşular

If komşu modül boş

$$T_{(x,y),t}^{(a,b)} = 1$$

Else If komşu modül düşük öncelikli nesne ile dolu

$$T_{(x,y),t}^{(a,b)} = 2$$

Else If komşu modül hedefine varmamış yüksek öncelikli nesne ile dolu

Yüksek önceliklinin hareket etmesinin bekle

Else If komşu modül hedefine varmamış yüksek öncelikli nesne ile dolu

If yüksek önceliklinin hedefe varma zamanı $\leq k-2$

$$T_{(x,y),t}^{(a,b)} = \infty$$

Else

Yüksek önceliklinin hareket etmesinin bekle

If minimum $LB_{(x,y)}^T < \infty$

Minimum $LB_{(x,y)}^T$ değerine sahip komşuyu hareket edilecek bir sonraki modül olarak seç

Güncelle mevcut modül, dolaşılacak listesi, ebeveyn modül

If ziyaret edilmemiş komşu yok

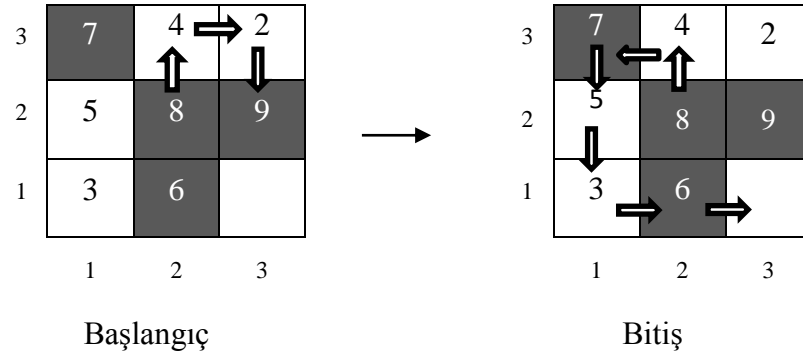
Mevcut nesnenin ebeveynini yeni mevcut aktif nesne olarak güncelle

Rota planlama sürecinin başına dön

Başlangıç modülünden hedef modüle olan yolu oluştur

5.2.2 Etiketleme Süreci

Yüksek öncelikli nesnenin planlanan yolunda ilerlerken, yol üzerinde kendisinden daha düşük öncelikli veya aktif olmayan bir nesne ile karşılaşarsa bu nesnelere tanımlanan yoldan uzaklaştırmak için kullanılan bir süreçtir.



Şekil 5.4: Etiketleme süreci örneği

Şekil 5.4-a'da 8 numaralı objemiz, (2,2) modülünden (2,3) modülüne hareket etmeye çalışmaktadır. (2,3) modülünde 4 önceliğine sahip aktif olmayan bir nesne bulunmaktadır. Bu nedenle 8 öncelikli nesne, 4 önceliğine sahip nesneyi etiketler. Etiketleme işleminde mevcut nesnenin önceliği olan 8, kalıtım önceliği olarak 4 önceliğine sahip nesneye devredilir. Bu sayede 4 önceliğine sahip nesne karşısına çıkan 5, 6 veya 7 öncelikli objeleri hareket ettirebilir. Çünkü bu nesne etiketleme işlemi süresince kalıtım önceliği olarak 8 değerini almıştır. 4 öncelikli nesne etiketlendikten sonra hareket edebileceği boş komşunun olup olmadığı incelenir. Komşuları (3,3) ve (1,3) modülleridir. 4 öncelikli nesne, en düşük öncelikli komşusu olan 2 önceliğine sahip nesneyi seçer. 4'ün kalıtım önceliği 2 öncelikli nesneye geçer ama 2 nesnesi yine 9 nesnesini etiketlemeye çalıştığında 9'un önceliği 8'den büyük olduğu için 9'u etiketleyemez ve 4 nesnesine geri döner. Burada ebeveyne geri dönme (backtracking) işlemi gerçekleştirilir. Etiketlenecek yeni nesne 7 olarak seçilir. $7 \rightarrow 5$ 'i, $5 \rightarrow 3$ 'ü, $3 \rightarrow 6$ 'yı etiketler ve en son boş modül olarak (3,1) modülü bulunur. Bu şekilde etiketleme süreci son bulur. En son durumda 6 (3,1)e, 3 (2,1)e taşınmış olur.

Etiketleme süreci aslında rota planlama ile benzerlik göstermektedir. Rota planlamadan farkı, hedef noktası için yol bulmak yerine tanımlı yol üzerindeki engelleri kaldırmak için boş bir modül bulmaya çalışır. Tablo 5.5'te etiketleme sürecinin işleyişine dair açıklama verilmiştir.

Tablo 5.5: Etiketleme süreci algoritması.

Etiketleme Süreci

```
SET mevcut_modül=başlangıç_modülü, dolaşıl原因 listesi={mevcut_modül}, ebeveyn_modül={}  
While mevcut_modül boş değilken  
    Mevcut_modülün komşularını tanımla  
    Komşuların daha önce dolaşıl原因 listesinde olup olmadığını kontrol et  
    If ziyaret edilmiş komşular varsa  
    For ziyaret edilmeyen tüm komşular  
        If komşu_modül boş  
            If komşu_modül etiketli değil  
                Boş_modül bulundu  
                Döngüden çık  
            Else If komşu_modül etiketli  
                If komşu_modülün kalıtım önceliği < aktif_nesnenin  
                önceliği  
                    Komşu_modüle ait etiketleme sürecini sıfırla  
                    Aktif_nesne tarafından komşu_modül için  
                    etiketleme sürecini başlat  
                    Boş_modül bulundu  
                    Döngüden çık  
                Else  
                    Komşu_modülü ziyaret edilen listesine ekle  
            Else If komşu_modül boş değil  
                If komşu_modül etiketli değil  
                    If komşu_modülün önceliği < aktif_nesnenin önceliği  
                    Komşu_modülden boş_modül buluncaya  
                    kadar etiketle sürecini tekrarla  
                    Else komşu_modülü dolaşıl原因 listesine ekle  
                If komşu_modül etiketli  
                    If komşu_modülün kalıtım önceliği < aktif_nesnenin  
                    önceliği  
                        Komşu_modüle ait etiketleme sürecini sıfırla  
                        Komşu_modülden boş_modül buluncaya  
                        kadar etiketle sürecini tekrarla  
                    Else komşu_modülü dolaşıl原因 listesine ekle  
            If aktif_nesnenin önceliğinden daha düşük öncelik/kalıtım öncelikli en az bir  
            modül varsa  
                If en düşük öncelikli komşu_modül etiketli değil  
                    Komşu_modül, bir sonraki zaman adımında hareket edilecek  
                    modül olarak seçilir  
                    Güncelle mevcut_modül, dolaşıl原因 listesi, ebeveyn_düğüm  
                If en düşük öncelikli komşu_modül etiketli  
                    Komşu_modüle ait etiketleme sürecini sıfırla  
                    Aktif_nesne tarafından komşu_modül için etiketleme sürecini  
                    başlat  
                    Komşu_modül, bir sonraki zaman adımında hareket edilecek  
                    modül olarak seçilir  
                    Güncelle mevcut_modül, dolaşıl原因 listesi, ebeveyn_düğüm  
            If aktif_nesnenin önceliğinden daha düşük öncelik/kalıtım öncelikli bir modül yoksa  
                Mevcut_nesnenin ebeveynini yeni mevcut aktif_nesne olarak güncelle  
                Etiketleme sürecinin başına dön  
            If ziyaret edilmiş komşu yok  
                Mevcut_nesnenin ebeveynini yeni mevcut aktif_nesne olarak güncelle  
                Etiketleme sürecinin başına dön  
Başlangıç_modülünden en son bulunan boş_modüle etiketleme sırasını oluştur
```

5.2.3 Ana Kontrol

Aktif ve aktif olmayan tüm nesnelere hareketi, kilitlenme durumları ana kontrol süreci tarafından kontrol edilir. Her zaman adımında ana kontrol süreci hedef noktasında olmayan her aktif nesneyi dikkate alır ve yüksek öncelikli başka bir nesne tarafından etiketlenip etiketlenmediğini kontrol eder. Eğer etiketli değilse değişen sistem ortamından dolayı mevcut aktif nesne için yeniden rota planlaması gerçekleştirir. Sittivijan (2015)'in çalışmasında uyguladığı yöntemden farkı da buradadır. Karşılaştırılan yöntemde ise eğer aktif nesne etiketli değil ise planlanan rotasında olup olmadığı kontrol edilir. Eğer aktif nesne planlanan rotasında değilse rota planlama süreci çağırılıp yeni bir yol tanımlanır, eğer planlanan rotasında ise ana kontrol süreci mevcut rotaya göre hareket edilecek olan modülü tanımlar.

Ana kontrol sürecinde karşılaşılan olası durumlar şu şekilde sıralanabilir:

- Eğer komşu modül boş ve başka bir nesne tarafından etiketli değilse mevcut aktif nesne o zaman dilimi içerisinde komşu modüle geçer.
- Eğer komşu modül boş ve başka bir nesne tarafından etiketli ise komşu modülün düşük öncelikli tarafından etiketlenip etiketlenmediği kontrol edilir. Eğer komşu modül düşük öncelikli nesne tarafından etiketliyse o nesneye ait etiketleme süreci sıfırlanır ve mevcut aktif nesne o zaman dilimi içerisinde komşu modüle geçer. Eğer komşu modül yüksek öncelikli nesne tarafından etiketliyse yüksek öncelikli nesne o modülden hareket edene kadar beklenir.
- Eğer komşu modül boş değil ve başka bir nesne tarafından etiketli ise yine komşu modülün düşük öncelikli tarafından etiketlenip etiketlenmediği kontrol edilir. Eğer komşu modül düşük öncelikli nesne tarafından etiketlendiyse o nesneye ait etiketleme süreci sıfırlanır ve mevcut nesne tarafından etiketleme işlemi yapılarak mevcut modülün komşu modüle hareketi sağlanır.
- Eğer komşu modül boş değil ve başka bir nesne tarafından etiketli değil ise komşu modüldeki nesnenin önceliğine bakılır. Eğer komşu modülün önceliği aktif nesnenin önceliğinden küçük ise bu modül için aktif nesne tarafından etiketleme süreci başlatılır ve etiketleme süreci

başarılı olursa aktif nesne komşu modüle geçiş yapar. Eğer komşu modülün önceliği aktif nesnenin önceliğinden büyük ise mevcut konumda beklenir.

Tablo 5.6’da ana kontrolün işleyişine dair açıklamaya yer verilmiştir.

Tablo 5.6: Ana kontrol süreci algoritması.

Ana Kontrol

Set zaman adımı=0

Tüm aktif nesnelere önceliğine göre sırala

While tüm aktif nesnelere hedef noktalarına varmadığı sürece

For her aktif nesne

 Rota planlama sürecini çalıştır

If rota planlama süreci başarısızsa

 Bekle

If komşu modül boş ve etiketli değil

 Komşu modüle hareket et

Else If komşu modül boş ve etiketli

If komşu modül düşük öncelikli nesne tarafından etiketli

 Komşu modüle ait etiketleme sürecini sıfırla

 Komşu modüle hareket et

Else If komşu modül yüksek öncelikli nesne tarafından etiketli

 Bekle

Else If mevcut nesne tarafından etiketliyse

 Komşu modüle hareket et

Else If komşu modül boş değil ve etiketli değil

If komşu modülün önceliği < mevcut aktif nesnenin önceliği

 Komşu modüle ait etiketleme sürecini sıfırla

 Mevcut nesne tarafından komşu modül için etiketleme sürecini başlat

If etiketleme süreci başarılı

 Komşu modüle hareket et

Else

 Bekle

Else

 Bekle

Else If komşu modül boş değil ve etiketli

If komşu modül düşük öncelikli nesne tarafından etiketli

 Komşu modüle ait etiketleme sürecini sıfırla

 Mevcut nesne tarafından komşu modül için etiketleme sürecini başlat

If etiketleme süreci başarılı

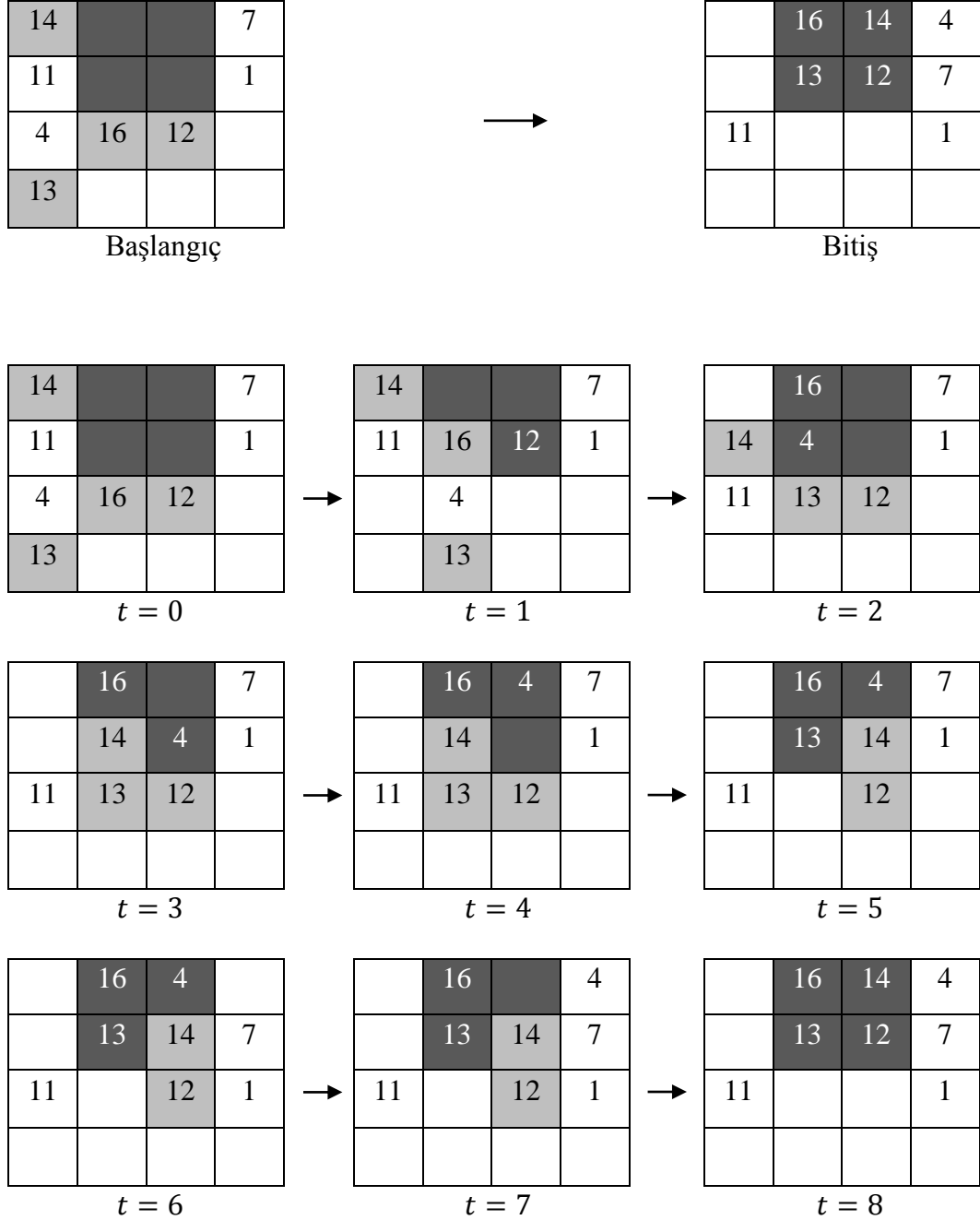
 Komşu modüle hareket et

Else

 Bekle

Güncelle zaman adımı=mevcut zaman adımı+1

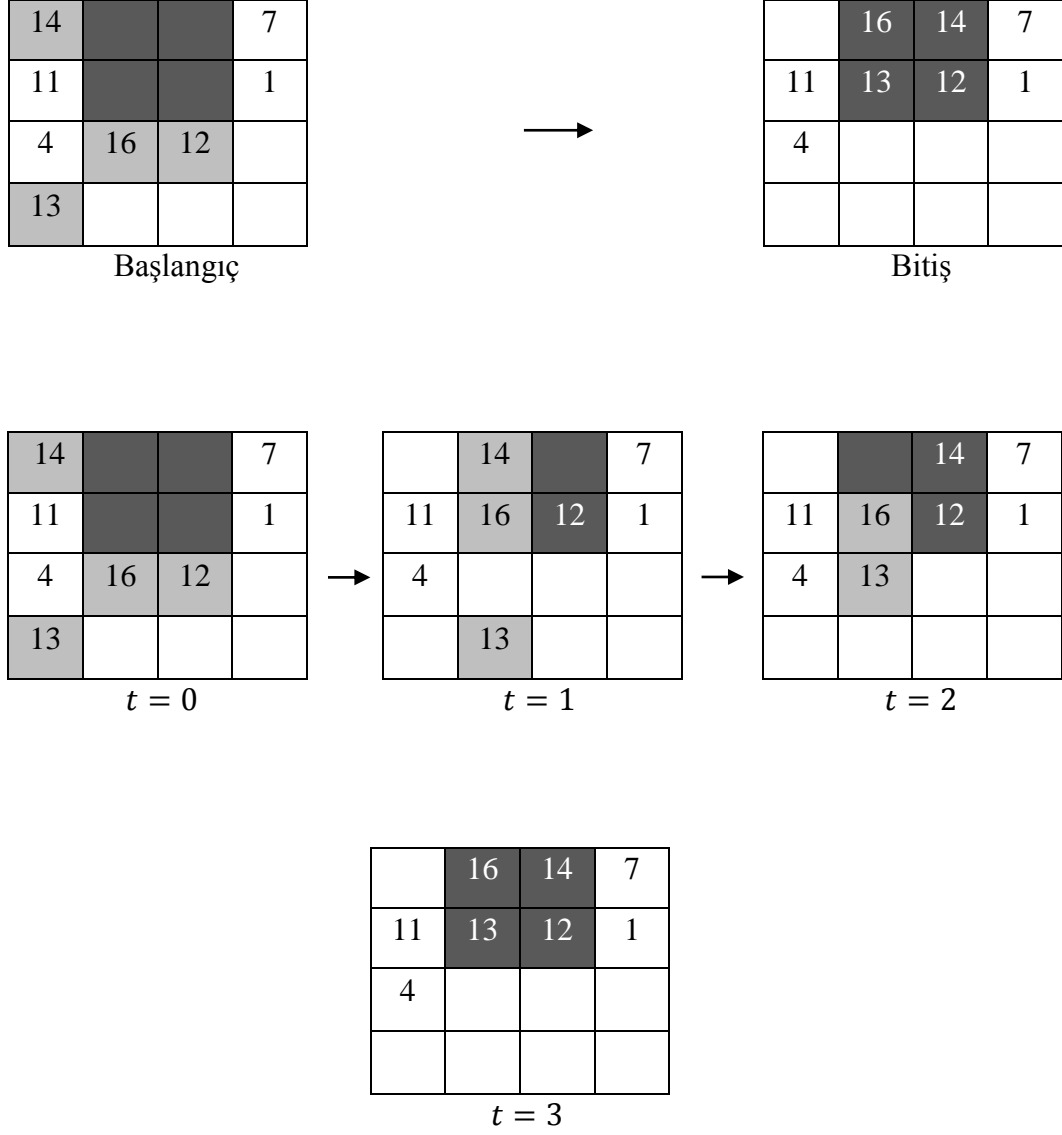
Önerilen sezgisel yaklaşımın çalışma süreci 4×4 boyutundaki bir örnek üzerinde incelenmiş ve $t = 0$ zaman adımından $t = 8$ zaman adımına kadar olan tüm adımları Şekil 5.5'te tek tek gösterilmiştir. Her zaman adımında etiketli olmayan tüm aktif nesnelere için buldukları noktadan tekrardan rota planlaması yapılmıştır. Koyu gri renkler hedefleri, açık gri renkler aktif nesnelere ifade etmektedir.



Şekil 5.5: Önerilen sezgisel yaklaşımın çözüm adımları

Önerilen sezgisel yaklaşımın temel halinde hem ana kontrol hem de etiketleme süreci kapsamında, etiketleme işlemine tabi tutulan nesnelere serbest bırakma işlemi (nesne daha önceden başka bir nesne tarafından etiketlenmişse bu sürecin iptal edilmesi) uygulanmaktadır. Önerilen yaklaşım üzerinde bir iyileştirme yapılmış ve bu işlem ana kontrol kısmından kaldırılıp sadece etiketleme süreci aşamasında uygulanmıştır. İyileştirilen bu algoritma ALG-P3 olarak isimlendirilmiştir.

Önerilen sezgisel yaklaşım üzerinde ikinci bir iyileştirme yapıлып çözüme daha kısa sürede ve daha az zaman adımıyla ulaşılması sağlanmıştır. İkinci iyileştirmenin yapıldığı bu algoritma ALG-P5 olarak isimlendirilmiştir. Algoritmanın ilk halinin rota planlama süreci içerisinde $LB_{(x,y)}^T$ değeri hesaplanırken her bir modülün k zaman adımı içerisindeki doluluk boşluk durumuna bakılmaktadır. Eğer k zaman adımı incelenen komşu modül, hedefine varacak yüksek öncelikli bir nesne ile dolacaksa o komşu modül için $LB_{(x,y)}^T = \infty$ olarak atanmaktadır. Ve aktif nesnenin o modüle geçişi engellenmektedir. Ama burada dikkat edilmesi gereken bir durum bulunmaktadır. Örneğin 14 numaralı nesnenin ilk hareketi incelendiğinde $k = 3$ alındığında $t = 0$, $t = 1$ ve $t = 2$ zaman adımlarındaki durumlar hesaplanır. 14 numaralı nesne $t = 0$ anından $t = 1$ zaman adımına geçişte sağ komşusuna geçebilecekken, komşu modüle $t = 2$ anında 16 numaralı öncelikli nesnenin varmasından dolayı bu modül için $LB_{(x,y)}^T = \infty$ olarak tanımlanır. İyileştirilen algoritmada ise yüksek önceliklinin hedefe varma durumu hesaplaması yapılırken sadece mevcut zaman adımı ve bir sonraki zaman adımı hedefine ulaşıyorsa $LB_{(x,y)}^T = \infty$ olarak atanır. Yani 14 öncelikli nesne için; $t = 0$ ve $t = 1$ zaman diliminde sağ komşu modülünde hedefine varan yüksek öncelikli nesne olmayacağı için komşu modüle geçiş olasılığı ortaya çıkar. Önerilen sezgiselin temel hali için çözülen örnek, iyileştirilmiş hali için de çözülmüş ve Şekil 5.6'da adımları gösterilmiştir. Önerilen algoritma hedef duruma 8 çözüm adımıyla ulaştığı görülürken iyileştirilmiş halinin 3 zaman adımıyla ulaştığı görülmüştür.



Şekil 5.6: ALG-P5 çözüm adımları

Önerilen sezgisel yaklaşımın üzerinde yapılan iyileştirmeler aynı şekilde temel alınan ALG-B1 üzerinde de yapılmış ve yeni algoritmalar çalışmada ALG-P2 ve ALG-P4 isimleri ile tanımlanmıştır.

5.3 Veri Kümeleri

Önerilen sistemin başarısının test edilmesi için 3 grup test kümesi için 23 adet test problemi üretilmiştir. 1. grup 4×4 boyutlarında bir adet %40, iki adet %50 yoğunluklu depo test problemlerini içermektedir. 2. grup 6×6 boyutlarında bir adet %40, bir adet %50, iki adet %60 ve bir adet %70 yoğunluklu depo test problemlerini

içermektedir. 3. grup 16×32 boyutlarında üç adet %20, birer adet %30 %40, %50, %60, %70, %80, %90, %95, %96, %97, %98, %99 yoğunluklu depo test problemlerinden oluşturulmuştur.

5.4 Önerilen Yaklaşımın Test Problemlerindeki Başarısı

Çalışma kapsamında önerilen algoritmaların daha iyi anlaşılabilmesi ve sonuçlarının incelenmesi amacıyla baz alınan Datar (2011) ve Sittivijan (2015)'in çalışmalarında kullandıkları algoritma ile A* temel alınarak geliştirilen algoritmalar çözüm süreleri ve çözüme ulaşmada geçen adım sayısı açısından karşılaştırılmıştır. Test edilen örnekler depo boyutlarına göre 3 gruba ayrılıp her grup için karşılaştırmalar farklı tablo ve grafikler ile gösterilmiştir.

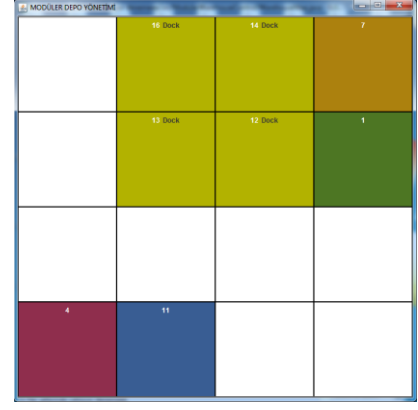
1. Grup: 1, 2 ve 3. nolu test problemleri
2. Grup: 4, 5, 6, 7 ve 8. nolu test problemleri
3. Grup: 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23. nolu test problemleri

Tüm algoritmalar Eclipse ortamında Java programlama dili kullanılarak kodlanmıştır. Karşılaştırmalar 4 GB RAM belleği olan ve 2.67 GHz işlemcili standart bir bilgisayarda gerçekleştirilmiştir. Elde edilen sonuçlar her algoritma için tablo ve grafiklerde gösterilmiş ve elde edilen verilerin değerlendirilmesi amacıyla bağıl ve mutlak sapmalar hesaplanmıştır. (20 numaralı örnek Alg-B1, Alg-P2 ve Alg-P4 sonuca ulaşamadığı için bu örnek değeri toplam sonucuna yansıtılmamıştır.) Mutlak sapma; ölçülen değer ile gerçek değer arasındaki farktır. Mutlak sapma Formül 3.6 kullanılarak hesaplanmıştır. Bağıl sapma; Formül 5.7'de gösterildiği gibi ölçülen değer ile gerçek sonucun farkının, gerçek sonuca bölümüdür. Çoğunlukla yüzdesel ifade edildiği için hata yüzdesi olarak da anılır (Internet 4).

Şekil 5.7, 5.8 ve 5.9'da 4×4 , 6×6 ve 16×32 boyutlu örneklere ait ekran görüntüleri verilmiştir.

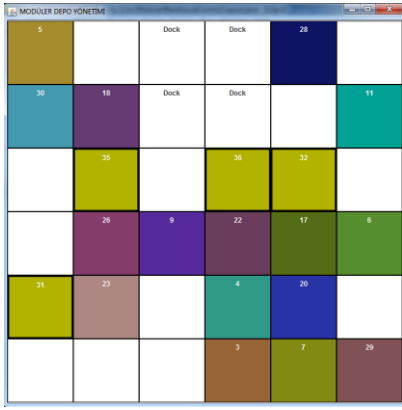


Başlangıç



Bitiş

Şekil 5.7: 4×4 boyutundaki problem örneği

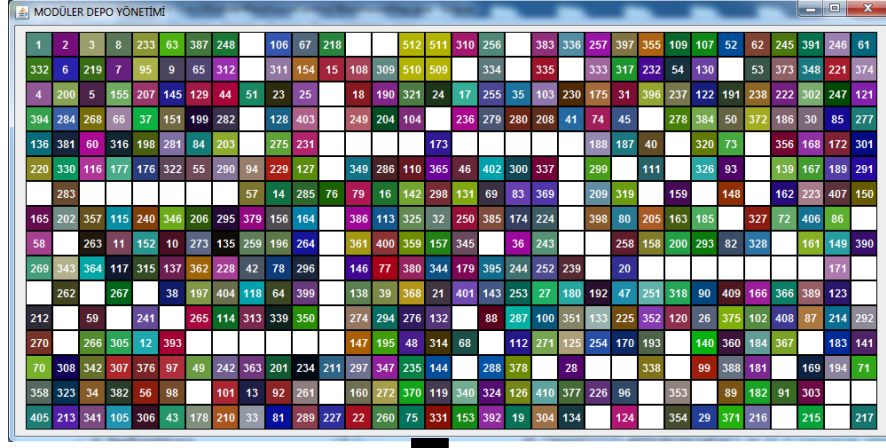


Başlangıç



Bitiş

Şekil 5.8: 6×6 boyutundaki problem örneği



Şekil 5.9: 16 × 32 boyutundaki problem örneği

$$\text{Mutlak Sapma} = |x_i - x_t| \quad (5.6)$$

$$\text{Bağıl Sapma} = \frac{|x_i - x_t|}{x_t} \times 100 \quad (5.7)$$

5.4.1 Tüm Algoritmaların Çalışma Süreleri ve Göreceli Karşılaştırmaları

Tablo 5.7’de incelenen tüm algoritmaların özet halinde birbirlerinden farkı gösterilmiştir.

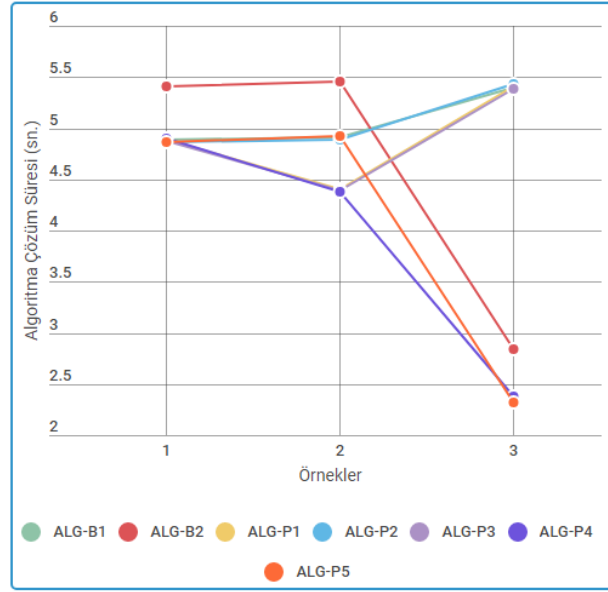
Tablo 5.7: İncelenen tüm algoritmaların karşılaştırması

ALG-B1	<ul style="list-style-type: none"> ❖ Temel alınan Sittivijan (2015)'in algoritması ❖ Sezgisel temelli bir yaklaşımdır. ❖ Her aktif nesne için başlangıçta A* temelli bir sezgisel ile rota planlaması gerçekleştirilir
ALG-B2	<ul style="list-style-type: none"> ❖ Datar (2011)'in çalışmasında önerdiği algoritma ❖ Açgözlü bir yaklaşımdır. ❖ Sadece bir sonraki zaman adımındaki hareketin planlandığı bir algoritmadır.
ALG-P1	<ul style="list-style-type: none"> ❖ ALG-B1'in her ortam değişikliğinde tekrar çalıştırıldığı algoritma (Önerilen Algoritma) ❖ Sezgisel temelli bir yaklaşımdır. ❖ Her aktif nesne için her zaman adımında A* temelli bir sezgisel ile tekrar rota planlaması gerçekleştirilir. ❖ ALG-B1 ve ALG-B2'ye göre daha kısa zamanda ve çözüm adım sayısında ulaşmaktadır.
ALG-P2	<ul style="list-style-type: none"> ❖ ALG-B1'in iyileştirilmiş halidir. Daha kısa sürede çözüme ulaşmaktadır. ❖ Etiketleme işlemine tabi tutulan nesnelere uygulanan serbest bırakma işlemi ana kontrol kısmından kaldırılıp sadece etiketleme süreci aşamasında uygulanmıştır.
ALG-P3	<ul style="list-style-type: none"> ❖ ALG-P1'in iyileştirilmiş halidir. Daha kısa sürede çözüme ulaşmaktadır. ❖ ALG-P2'de gerçekleşen iyileştirme gibi burada da serbest bırakma işlemi ana kontrol kısmından kaldırılıp sadece etiketleme süreci aşamasında uygulanmıştır
ALG-P4	<ul style="list-style-type: none"> ❖ ALG-P2'nin iyileştirilmiş halidir. ❖ Daha kısa sürede ve daha az çözüm adım sayısında sonuca ulaşmaktadır. ❖ $LB_{(x,y)}^T$ değerinin hesaplanmasında değişiklik yapılmıştır.
ALG-P5	<ul style="list-style-type: none"> ❖ ALG-P3'nin iyileştirilmiş halidir. ❖ Daha kısa sürede ve daha az çözüm adım sayısında sonuca ulaşmaktadır. ❖ $LB_{(x,y)}^T$ değerinin hesaplanmasında değişiklik yapılmıştır.

Tablo 5.8'de 4×4 boyutundaki test problemleri için tüm algoritmaların süreleri karşılaştırılmış, elde edilen sonuçlar Şekil 5.10'da gösterilmiştir. Ayrıca 1. grup test problemleri için tüm algoritmalar ALG-B1 ile göreceli olarak karşılaştırılıp elde edilen sonuçlar Tablo 5.9'da gösterilmiştir. Karşılaştırmalarda ALG-B1 baz performansı=1 olarak alınmıştır. Baz değerden küçük değerler daha iyi çözümlere, büyük değerler daha kötü çözümleri ifade etmektedir. Göreceli karşılaştırmalara ait sonuçlar Şekil 5.11'de gösterilmiştir. Tablo incelendiğinde önerilen algoritmaların daha kısa sürede çözüme ulaştığı görülmüştür.

Tablo 5.8: 1. Grup için tüm algoritmaların çalışma süreleri karşılaştırmaları.

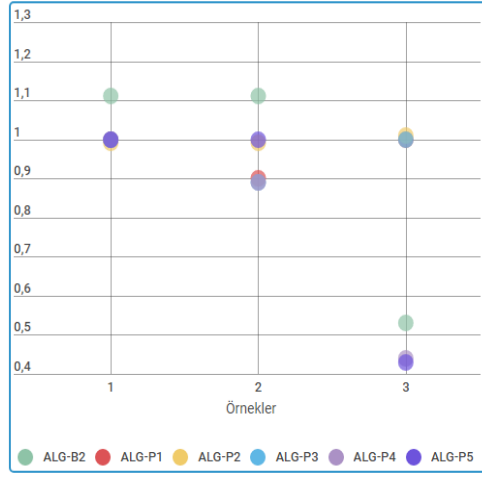
ALGORİTMA ÇÖZÜM SÜRELERİ (sn.)									
ÖRNEK	ALG-B1	ALG-B2	ALG-P1	ALG-P2	ALG-P3	ALG-P4	ALG-P5	Boyut	Yoğunluk
1	4,883	5,404	4,889	4,858	4,877	4,903	4,865	4*4	0,40
2	4,907	5,452	4,408	4,882	4,386	4,379	4,918	4*4	0,50
3	5,4	2,84	5,41	5,427	5,38	2,38	2,32	4*4	0,50
TOPLAM	15,19	13,696	14,707	15,167	14,643	11,662	12,103		



Şekil 5.10:1. Grup için tüm algoritmaların çalışma sürelerinin karşılaştırması

Tablo 5.9: 1. Grup için tüm algoritmaların göreceli karşılaştırmaları.

GÖRECELİ KARŞILAŞTIRMALAR								
ALG-B1	ALG-B2	ALG-P1	ALG-P2	ALG-P3	ALG-P4	ALG-P5	Boyut	Yoğunluk
1	1,11	1,00	0,99	1,00	1,00	1,00	4*4	0,40
1	1,11	0,90	0,99	0,89	0,89	1,00	4*4	0,50
1	0,53	1,00	1,01	1,00	0,44	0,43	4*4	0,50
1	0,90	0,97	1,00	0,96	0,77	0,80		

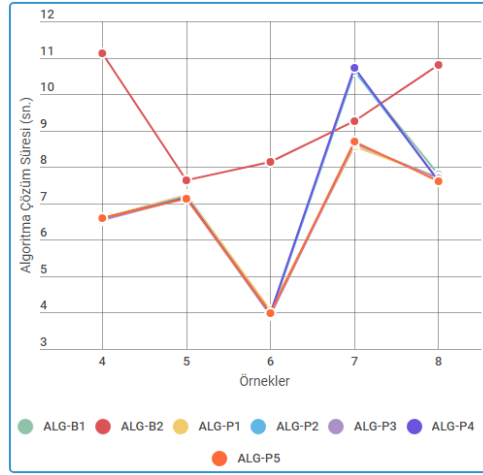


Şekil 5.11: 1. Grup için tüm algoritmaların ALG-B1'e göre göreceli karşılaştırması

Tablo 5.10'da 8×8 boyutundaki test problemleri için tüm algoritmaların süreleri karşılaştırılmış, elde edilen sonuçlar Şekil 5.12'de gösterilmiştir. Ayrıca 2. grup test problemleri için tüm algoritmalar ALG-B1 ile göreceli olacak şekilde karşılaştırılmış ve Tablo 5.11'de gösterilmiştir. Göreceli karşılaştırmalara ait sonuçlar Şekil 5.13'ta gösterilmiştir.

Tablo 5.10: 2. Grup için tüm algoritmaların çalışma süreleri karşılaştırmaları.

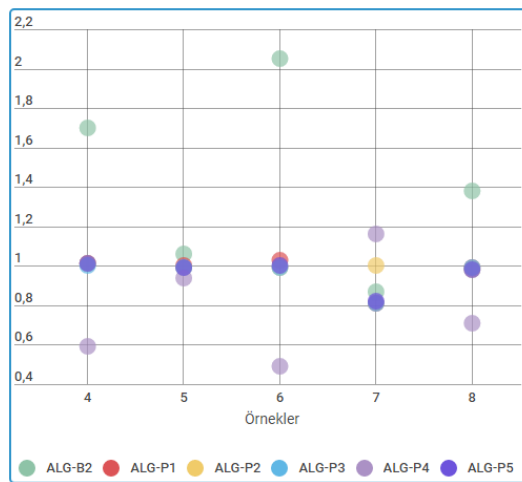
ÖRNEK	ALGORİTMA ÇÖZÜM SÜRELERİ (sn.)							Boyut	Yoğunluk
	ALG-B1	ALG-B2	ALG-P1	ALG-P2	ALG-P3	ALG-P4	ALG-P5		
4	6,535	11,111	6,599	6,577	6,551	6,607	6,599	6*6	0,40
5	7,216	7,635	7,232	7,173	7,146	7,146	7,126	6*6	0,50
6	3,969	8,14	4,071	3,945	3,942	3,979	3,972	6*6	0,60
7	10,654	9,263	8,577	10,652	8,64	10,713	8,697	6*6	0,60
8	7,798	10,79	7,726	7,621	7,687	7,615	7,613	6*6	0,70
TOPLAM	28,374	36,149	26,479	28,347	26,279	28,445	26,394		



Şekil 5.12: 2. Grup için tüm algoritmaların çalışma sürelerinin karşılaştırması

Tablo 5.11: 2. Grup için tüm algoritmaların ALG-B1'e göre göreceli karşılaştırmaları.

GÖRECELİ KARŞILAŞTIRMALAR									
ÖRNEK	ALG-B1	ALG-B2	ALG-P1	ALG-P2	ALG-P3	ALG-P4	ALG-P5	Boyut	Yoğunluk
4	1	1,70	1,01	1,01	1,00	1,01	1,01	6*6	0,40
5	1	1,06	1,00	0,99	0,99	0,99	0,99	6*6	0,50
6	1	2,05	1,03	0,99	0,99	1,00	1,00	6*6	0,60
7	1	0,87	0,81	1,00	0,81	1,01	0,82	6*6	0,60
8	1	1,38	0,99	0,98	0,99	0,98	0,98	6*6	0,70
TOPLAM	1	1,27	0,93	1,00	0,93	1,00	0,93		

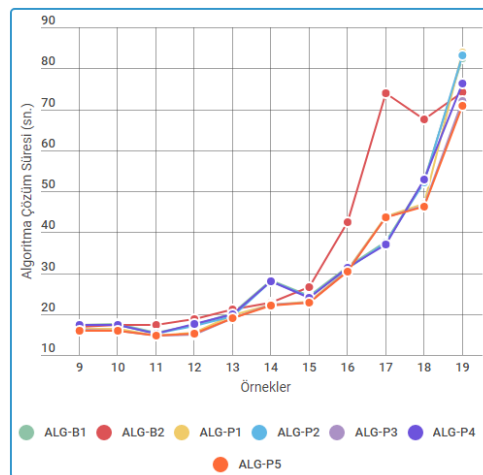


Şekil 5.13: 2. Grup için tüm algoritmaların ALG-B1'e göre göreceli karşılaştırmaları

Tablo 5.12’de 16×32 boyutundaki test problemleri için tüm algoritmaların süreleri karşılaştırılmış, elde edilen sonuçlar Şekil 5.14’te gösterilmiştir. Aynı şekilde 3. grup test problemleri için tüm algoritmalar ALG-B1 ile göreceli olacak şekilde karşılaştırılmış ve Tablo 5.13’te gösterilmiştir. Göreceli karşılaştırmalara ait sonuçlar Şekil 5.15’te gösterilmiştir. Tablodaki veriler incelendiğinde önerilen algoritmanın en iyileştirilmiş hali olan ALG-P5’in en iyi sonuca ulaştığı görülmüştür.

Tablo 5.12: 3. Grup için tüm algoritmaların çalışma süreleri ve göreceli karşılaştırmaları.

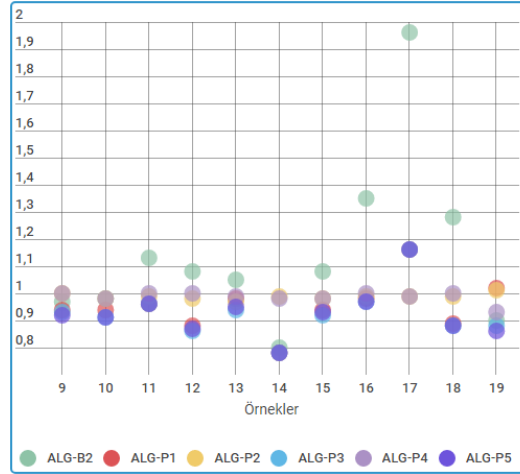
ÖRNEK	ALGORİTMA ÇÖZÜM SÜRELERİ (sn.)							Boyut	Yoğunluk
	ALG-B1	ALG-B2	ALG-P1	ALG-P2	ALG-P3	ALG-P4	ALG-P5		
9	17,355	16,875	16,327	17,32	16,105	17,303	15,95	16*32	0,20
10	17,557	17,259	16,478	17,266	16,016	17,259	16,012	16*32	0,20
11	15,354	17,323	14,715	15,243	14,666	15,287	14,7	16*32	0,20
12	17,423	18,835	15,333	17,05	14,994	17,502	15,18	16*32	0,30
13	20,073	21,039	19,597	19,438	18,937	19,9	19,04	16*32	0,40
14	28,325	22,797	22,232	28,064	22,21	27,893	22,052	16*32	0,50
15	24,543	26,569	22,951	24,14	22,702	23,943	22,76	16*32	0,60
16	31,5	42,49	30,945	31,132	30,53	31,403	30,426	16*32	0,70
17	37,732	73,991	43,833	37,38	43,626	37	43,6	16*32	0,80
18	52,658	67,6	46,992	52,206	46,372	52,8	46,224	16*32	0,90
19	82,327	74,194	83,755	83,038	72,056	76,26	70,747	16*32	0,95
20	-	138,412	122,928	-	120,902	-	120,445	16*32	0,96
21	-	-	-	-	-	-	-	16*32	0,97
22	-	136,428	-	-	-	-	-	16*32	0,98
23	-	-	-	-	-	-	-	16*32	0,99
TOPLAM	344,847	398,972	333,158	342,277	318,214	336,854	316,691		



Şekil 5.14: 3. Grup için tüm algoritmaların çalışma sürelerinin karşılaştırması

Tablo 5.13: 3. Grup için tüm algoritmaların çalışma süreleri ve göreceli karşılaştırmaları.

ÖRNEK	GÖRECELİ KARŞILAŞTIRMALAR							Boyut	Yoğunluk
	ALG-B1	ALG-B2	ALG-P1	ALG-P2	ALG-P3	ALG-P4	ALG-P5		
9	1	0,97	0,94	1,00	0,93	1,00	0,92	16*32	0,20
10	1	0,98	0,94	0,98	0,91	0,98	0,91	16*32	0,20
11	1	1,13	0,96	0,99	0,96	1,00	0,96	16*32	0,20
12	1	1,08	0,88	0,98	0,86	1,00	0,87	16*32	0,30
13	1	1,05	0,98	0,97	0,94	0,99	0,95	16*32	0,40
14	1	0,80	0,78	0,99	0,78	0,98	0,78	16*32	0,50
15	1	1,08	0,94	0,98	0,92	0,98	0,93	16*32	0,60
16	1	1,35	0,98	0,99	0,97	1,00	0,97	16*32	0,70
17	1	1,96	1,16	0,99	1,16	0,99	1,16	16*32	0,80
18	1	1,28	0,89	0,99	0,88	1,00	0,88	16*32	0,90
19	1	0,90	1,02	1,01	0,88	0,93	0,86	16*32	0,95
20	-	-	-	-	-	-	-	16*32	0,96
21	-	-	-	-	-	-	-	16*32	0,97
22	-	-	-	-	-	-	-	16*32	0,98
23	-	-	-	-	-	-	-	16*32	0,99
TOPLAM	1	1,16	0,97	0,99	0,92	0,98	0,92		



Şekil 5.15: 3. Grup için tüm algoritmaların ALG-B1'e göre göreceli karşılaştırması

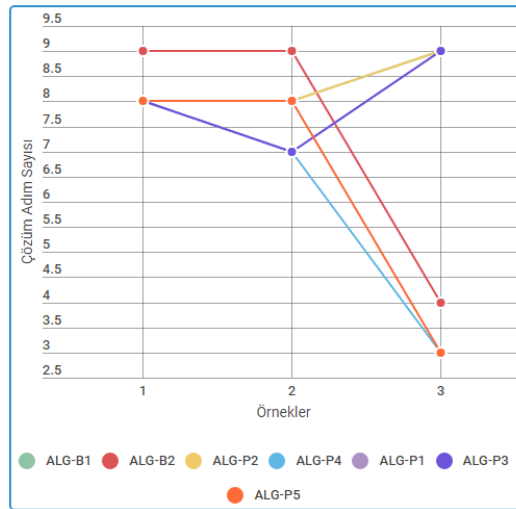
ALG-B1, ALG-P2, ALG-P4'ün 20,21,22 ve 23. örnekler, ALG-P1, ALG-P3, ALG-P5'in 21, 22 ve 23. örnekler için çözüme ulaşılmadığı görülmüştür. Bunun sebebi mevcut yapılandırmada, aktif paketlerin hedef noktalarına varış için bir yol tanımlanamaması ve kilitlemenin meydana gelmesinden dolayıdır. Paketler herhangi bir modüle hareket edememiş ve sistem kilitlemiştir.

5.4.2 Tüm Algoritmaların Çözüm Adım Sayılarının Karşılaştırması

Tablo 5.14’te 1. grup test problemleri için tüm algoritmalar paketlerin hedef noktalarına varması için gereken adım sayıları bakımından karşılaştırılmış, elde edilen sonuçlar Şekil 5.16’da gösterilmiştir. Aynı test problemleri için tüm algoritmalar, çözüm adım sayılarına göre ALG-B1 ile göreceli olacak şekilde karşılaştırılıp Tablo 5.15’te gösterilmiş ve sonuçlar Şekil 5.17’de gösterilmiştir.

Tablo 5.14: 1. Grup için tüm algoritmaların çözüm adım sayısı karşılaştırması.

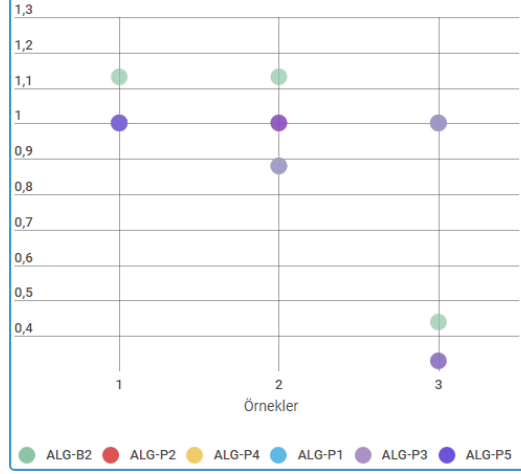
ÖRNEK	ÇÖZÜM ADIM SAYISI							Boyut	Yoğunluk
	ALG-B1	ALG-B2	ALG-P2	ALG-P4	ALG-P1	ALG-P3	ALG-P5		
1	8	9	8	8	8	8	8	4*4	0,40
2	8	9	8	7	7	7	8	4*4	0,50
3	9	4	9	3	9	9	3	4*4	0,50
TOPLAM	25	22	25	18	24	24	19		



Şekil 5.16: 1. Grup için tüm algoritmaların çözüm adım sayısının karşılaştırması

Tablo 5.15: 1. Grup için tüm algoritmaların ALG-B1’e göre göreceli çözüm adım sayısı karşılaştırması.

ÖRNEK	GÖRECELİ KARŞILAŞTIRMALAR							Boyut	Yoğunluk
	ALG-B1	ALG-B2	ALG-P2	ALG-P4	ALG-P1	ALG-P3	ALG-P5		
1	1	1,13	1,00	1,00	1,00	1,00	1,00	4*4	0,40
2	1	1,13	1,00	0,88	0,88	0,88	1,00	4*4	0,50
3	1	0,44	1,00	0,33	1,00	1,00	0,33	4*4	0,50
TOPLAM	1	0,88	1,00	0,72	0,96	0,96	0,76		

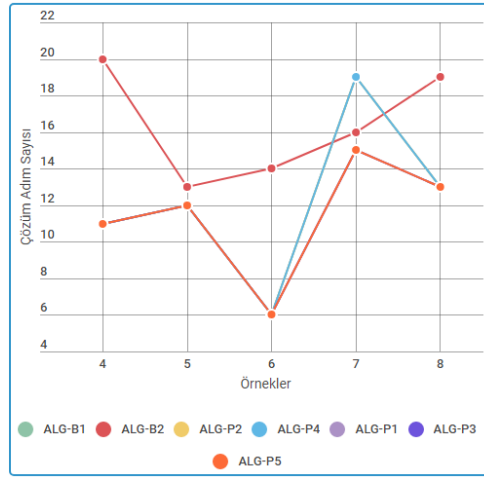


Şekil 5.17: 1. Grup için tüm algoritmaların ALG-B1'e göre göreceli karşılaştırması.

Tablo 5.16'da 2. grup test problemleri için tüm algoritmalar paketlerin hedef noktalarına varmada gereken adım sayıları bakımından karşılaştırılmış, elde edilen sonuçlar Şekil 5.18'de gösterilmiştir. Aynı test problemleri için tüm algoritmalar, çözüm adım sayılarına göre ALG-B1 ile göreceli olacak şekilde karşılaştırılıp Tablo 5.17'de gösterilmiş ve sonuçlar Şekil 5.19'da gösterilmiştir. Çözüm adım sayıları açısından da incelendiğinde önerilen algoritmaların ALG-B1 ve ALG-B2'den daha iyi sonuçlara ulaştığı gözlenmiştir.

Tablo 5.16: 2. Grup için tüm algoritmaların çözüm adım sayısı karşılaştırması.

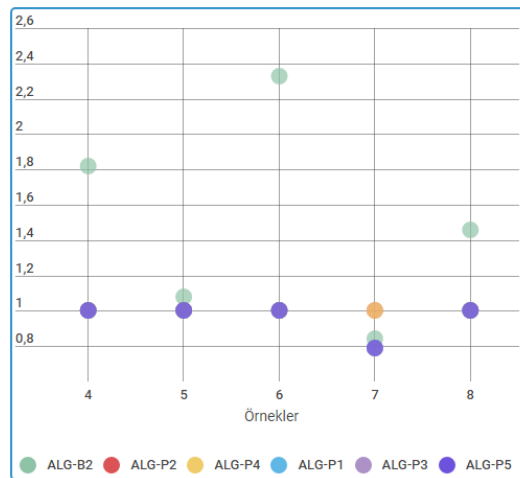
ÖRNEK	ÇÖZÜM ADIM SAYISI							Boyut	Yoğunluk
	ALG-B1	ALG-B2	ALG-P2	ALG-P4	ALG-P1	ALG-P3	ALG-P5		
4	11	20	11	11	11	11	11	6*6	0,40
5	12	13	12	12	12	12	12	6*6	0,50
6	6	14	6	6	6	6	6	6*6	0,60
7	19	16	19	19	15	15	15	6*6	0,60
8	13	19	13	13	13	13	13	6*6	0,70
TOPLAM	61	82	61	61	57	57	57		



Şekil 5.18: 2. Grup için tüm algoritmaların çözüm adım sayısının karşılaştırması

Tablo 5.17: 2. Grup için tüm algoritmaların ALG-B1'e göre göreceli çözüm adım sayısı karşılaştırması.

GÖRECELİ KARŞILAŞTIRMALAR									
ÖRNEK	ALG-B1	ALG-B2	ALG-P2	ALG-P4	ALG-P1	ALG-P3	ALG-P5	Boyut	Yoğunluk
4	1	1,82	1,00	1,00	1,00	1,00	1,00	6*6	0,40
5	1	1,08	1,00	1,00	1,00	1,00	1,00	6*6	0,50
6	1	2,33	1,00	1,00	1,00	1,00	1,00	6*6	0,60
7	1	0,84	1,00	1,00	0,79	0,79	0,79	6*6	0,60
8	1	1,46	1,00	1,00	1,00	1,00	1,00	6*6	0,70
TOPLAM	1	1,34	1,00	1,00	0,93	0,93	0,93		

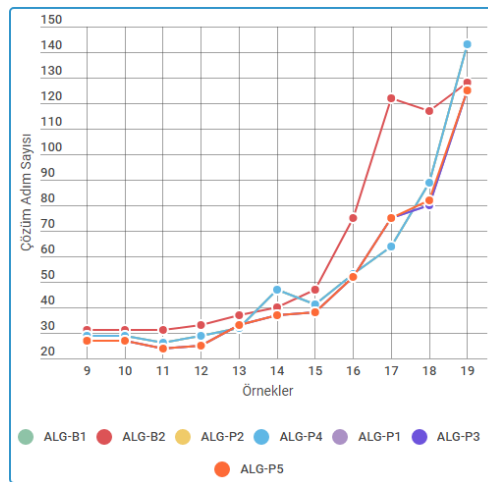


Şekil 5.19: 2. Grup için tüm algoritmaların ALG-B1'e göre göreceli karşılaştırması

Tablo 5.18’de 3. grup test problemleri için tüm algoritmalar paketlerin hedef noktalarına varmada gereken adım sayıları bakımından karşılaştırılmış, elde edilen sonuçlar Şekil 5.20’de gösterilmiştir. Aynı test problemleri için tüm algoritmalar, çözüm adım sayılarına göre ALG-B1 ile göreceli olacak şekilde Tablo 5.19’da gösterilmiş ve sonuçlar Şekil 5.21’de gösterilmiştir.

Tablo 5.18: 3. Grup için tüm algoritmaların çözüm adım sayısı karşılaştırması.

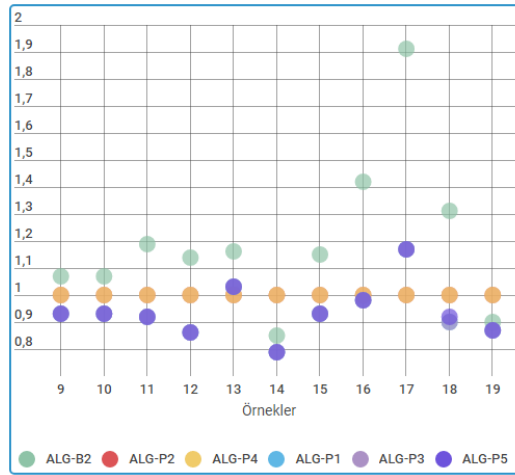
ÖRNEK	ÇÖZÜM ADIM SAYISI							Boyut	Yoğunluk
	ALG-B1	ALG-B2	ALG-P2	ALG-P4	ALG-P1	ALG-P3	ALG-P5		
9	29	31	29	29	27	27	27	16*32	0,20
10	29	31	29	29	27	27	27	16*32	0,20
11	26	31	26	26	24	24	24	16*32	0,20
12	29	33	29	29	25	25	25	16*32	0,30
13	32	37	32	32	33	33	33	16*32	0,40
14	47	40	47	47	37	37	37	16*32	0,50
15	41	47	41	41	38	38	38	16*32	0,60
16	53	75	53	53	52	52	52	16*32	0,70
17	64	122	64	64	75	75	75	16*32	0,80
18	89	117	89	89	80	80	82	16*32	0,90
19	143	128	143	143	125	125	125	16*32	0,95
20	-	232	-	-	205	205	208	16*32	0,96
21	-	-	-	-	-	-	-	16*32	0,97
22	-	-	-	-	-	-	-	16*32	0,98
23	-	-	-	-	-	-	-	16*32	0,99
TOPLAM	582	692	582	582	543	543	545		



Şekil 5.20: 3. Grup için tüm algoritmaların çözüm adım sayısının karşılaştırması

Tablo 5.19: 3. Grup için tüm algoritmaların ALG-B1'e göre göreceli çözüm adım sayısı karşılaştırması.

ÖRNEK	GÖRECELİ KARŞILAŞTIRMALAR							Boyut	Yoğunluk
	ALG-B1	ALG-B2	ALG-P2	ALG-P4	ALG-P1	ALG-P3	ALG-P5		
9	1	1,07	1,00	1,00	0,93	0,93	0,93	16*32	0,20
10	1	1,07	1,00	1,00	0,93	0,93	0,93	16*32	0,20
11	1	1,19	1,00	1,00	0,92	0,92	0,92	16*32	0,20
12	1	1,14	1,00	1,00	0,86	0,86	0,86	16*32	0,30
13	1	1,16	1,00	1,00	1,03	1,03	1,03	16*32	0,40
14	1	0,85	1,00	1,00	0,79	0,79	0,79	16*32	0,50
15	1	1,15	1,00	1,00	0,93	0,93	0,93	16*32	0,60
16	1	1,42	1,00	1,00	0,98	0,98	0,98	16*32	0,70
17	1	1,91	1,00	1,00	1,17	1,17	1,17	16*32	0,80
18	1	1,31	1,00	1,00	0,90	0,90	0,92	16*32	0,90
19	1	0,90	1,00	1,00	0,87	0,87	0,87	16*32	0,95
20	-	-	-	-	-	-	-	16*32	0,96
21	-	-	-	-	-	-	-	16*32	0,97
22	-	-	-	-	-	-	-	16*32	0,98
23	-	-	-	-	-	-	-	16*32	0,99
TOPLAM	1	1,19	1,00	1,00	0,93	0,93	0,94		



Şekil 5.21: 3. Grup için tüm algoritmaların ALG-B1'e göre göreceli karşılaştırması

Algoritmaların ALG-B1'e göre bağıl ve mutlak sapma karşılaştırmaları, çalışma süreleri ve çözüm adım sayıları bakımından detaylı karşılaştırmaları eklede sunulmuştur.

6. SONUÇ VE ÖNERİLER

Bu tez çalışmasında, modüler depo yönetimi konusu ele alınmış ve modüler depo içerisindeki birden çok birim boyutlu paketin eş zamanlı hareketi için A* temelli yeni bir sezgisel yöntem önerilmiştir.

Modüler depo yönetimi için önerilen yöntem üç aşamadan oluşmaktadır. Birinci aşama, her bir paketin başlangıç konumundan bitiş konumuna olan hareketinin gerçekleştirilmesi için kullanılan rota planlamadır. İkinci aşama, paket çarpışmalarını önlemek amacıyla kullanılan, paket önceliklerine dayalı olan etiketleme sürecidir. Son aşama da depoda bulunan tüm paketlerin hareketinin kontrol edildiği ana kontrol kısmıdır.

Önerilen yeni yöntem Sittivijan (2015)'un ve Datar (2011)'in çalışmalarında yer alan yöntemlerle karşılaştırılmıştır. Datar (2011) çalışmasında rota planlama kısmında sadece hedefe olan uzaklığa ve paketlerin önceliklerine bakarak bir sonraki hareket alanı seçmiştir. Sittivijan (2015) ise paketlerin hareketi için A* temelli bir sezgisel yöntem önermiştir. Çalışmamızda önerilen yöntemin Sittivijan (2005)'un çalışmasından farkı; var olan sezgiselde rota planlama en başta yapılıyor ve paketler planlanan rotalarının dışına çıkmadıkları sürece tekrar rota planlama işlemi uygulanmıyordu. Önerilen yöntemde ise her ortam değişikliğinde aktif paketler için rota planlama işlemi tekrar uygulanmaktadır. Ayrıca Sittivijan (2015)'un çalışmasında hem ana kontrol hem de etiketleme süreci kısmında, etiketleme işlemine tabi tutulan nesnelere serbest bırakma işlemi uygulanmaktadır. Önerilen yöntemde bu işlem ana kontrol kısmından kaldırılıp sadece etiketleme süreci aşamasında uygulanmış ve bu şekilde oldukça kaliteli sonuçlara ulaşılmıştır. Ayrıca rota planlama aşamasında LB değeri hesaplamasında yüksek öncelikli paketlerin hedefe varması durumundaki koşullara yönelik de bir iyileştirme yapılmış ve karşılaştırılan algorithmadan daha iyi sonuçlar elde edilmiştir. Bu işlemler sonucu elde edilen tüm çözüm süreleri karşılaştırılarak Yapılan testler sonucu, tezde önerilen sezgiselin ve bu sezgiselin iyileştirilmiş sürümlerinin temel alınan literatürdeki algorithmadan daha kısa sürede çözüme ulaştığı gözlemlenmiştir.

Test edilen örnekler için çözüme ulaşmada, toplam süreye bakıldığı zaman en iyi performansı ALG-P5'in sergilediği görülmüştür. ALG-B1 ve bu algoritmanın iyileştirilmiş versiyonları olan ALG-P2ve ALG-P4'ün, ALG-P1 ve bu algoritmanın iyileştirilmiş versiyonları olan ALG-P3ve ALG-P5'e göre daha az sayıda örnek için çözüme ulaşmıştır.

Bu tez çalışmasında üretilenler temel alınarak modüler depo yönetimi için farklı bakış açıları ile yeni çalışmalar geliştirilebilir. Rota planlama sürecinden yola çıkarak etiketleme sürecinde boş alan aranmasında A* veya benzeri temelli sezgisel algoritmalar kullanılabilir. Birim boyutlu paketler için önerilen algoritmadan esinlenerek çok boyutlu paketlerin eş zamanlı hareketi için de yeni bir yöntem geliştirilebilir.

7. KAYNAKLAR

Azarm, K. and Schmidt, G., “Conflict-free motion of multiple mobile robots based on decentralized motion planning and negotiation.”, *In Proceedings of the IEEE International Conference on Robotics and Automation*, 3526-3533, (1997).

Bansal, A., *Design of a Public Logistics Network*, (2004).

Bartholdi, J. J., & Hackman, S. T., *Warehouse & Distribution Science: Release 0.89*, Supply Chain and Logistics Institute, (2008).

Bauer, B., “The Manhattan Pair Distance Heuristic for the 15-Puzzle.”, (1994).

Bowersox, D. J., Closs, D. J. and Cooper, M. B., *Supply Chain Logistics Management*: McGraw-Hill/Irwin, (2002).

Internet - 4: Bulduk, İ., “Analitik Verilerin Değerlendirilmesi [online]”, (08 Şubat 2017), <https://www.slideshare.net/bulduki/analiz-sonularnn-istatistiksel-deerlendirilmesi>, (2013).

Christopher, M., *Logistics and Supply Chain Management*: Prentice Hall, (2005).

Internet - 5: Coşkun, C., “Network ve Topoloji Nedir? Topoloji Çeşitleri Nelerdir? [online]”, (20 Haziran 2017), <http://www.cenk.in/network-ve-topoloji-nedir-topoloji-cesitleri-nelerdir/>,

Datar, M. A., “Priority-based Control Algorithm for Movement of Packages in a Public Distribution Center.”, Master Thesis, North Carolina State University *Department of Industrial Engineering*, (2011).

Internet - 2: “Domino's launches world's first driverless pizza delivery vehicles [online]”, (10 Mayıs 2016), <https://www.dezeen.com/2015/04/01/dominos-launches-worlds-first-driverless-pizza-delivery-vehicles/>, (2015)

Flake, G. W. and Baum, E. B., “Rush Hour is PSPACE-complete, or “Why you should generously tip parking lot attendants””, *Theoretical Computer Science*, 270(1), 895-911, (2002).

Ghiani, G., Laporte, G., & Musmanno, R., *Introduction to Logistics Systems Planning and Control*, John Wiley & Sons, (2004).

Ghiani, G., Laporte, G. and Musmanno, R., *Introduction to Logistics Systems Management*: John Wiley & Sons, (2013).

Gue, K. R. and Kim, B. S., "Puzzle-based storage systems." *Naval Research Logistics (NRL)*, 54(5), 556-567, (2007).

Guo, Y. and Parker, L. E., "A distributed and optimal motion planning approach for multiple mobile robots.", *In Proceedings of the IEEE International Conference on Robotics and Automation*, Vol. 3, 2612-2619, (2002).

Güler, E., "Depo Yönetiminde Bilgi Teknolojileri Kullanımı", Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi *Fen Bilimleri Enstitüsü*, İstanbul, (2006).

Harrison, A. and van Hoek, R., *Logistics Management and Strateg*: Pearson Education, (2008).

Hauptman, A., Elyasaf, A., Sipper, M. and Karmon, A., "GP-Rush: using genetic programming to evolve solvers for the Rush Hour puzzle.", *In Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation*, 955-962, (2009).

Hearn, R. A. and Demaine, E. D., "PSPACE-completeness of sliding-block puzzles and other problems through the nondeterministic constraint logic model of computation.", *Theoretical Computer Science*, 343(1-2), 72-96, (2005).

Hopbaoglu, F., "Tedarik Zincirinde ve Lojistik Süreçlerde Depo Tasarımı ve Depo Yönetimi: Kozmetik Sektöründe Bir Uygulama", Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi *Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalı*, İstanbul, (2009).

Hopcroft, J. E., Schwartz, J. T. and Sharir, M., "On the Complexity of Motion Planning for Multiple Independent Objects; PSPACE-Hardness of the Warehouseman's Problem.", *The International Journal of Robotics Research*, 3(4), 76-88, (1984).

Jain, A., "Protocol Design for a Public Logistics Network.", Master Thesis, Department of Industrial and System Engineering, *North Carolina State University*, (2004).

Internet - 6: Karaca, Y., "Ağ Topolojileri Nelerdir? [online]", (20 Haziran 2017), <https://www.teknologweb.com/ag-topolojileri-nedir>, (2013).

Kasilingam, R. G., *Logistics and Transportation – Design and Planning*: Springer Science, (1998).

Kay, M. G. and Parlikad, A. N., “Material Flow Analysis for Public Logistics Networks.”, *Progress in 2002 Material Handling Research*, 205–218, Charlotte, NC, (2002).

Kay, M. G., “Protocol Design for a Public Logistic Network.”, *Progress in Material Handling Research*, 181-188, (2004).

Internet - 3: Kay, M. G., “Public Logistics Networks [online]”, (15 Nisan 2016), <http://www.ise.ncsu.edu/kay/pln/index.html>, (2008).

Kay, M. G., Home Delivery Logistics Networks using Driverless Delivery Vehicles. Working paper, Department of Industrial and System Engineering, North Carolina State University, (2013).

Kay, M. G., “Warehousing [online]”, (15 Nisan 2016), <http://www4.ncsu.edu/~kay/Warehousing.pdf>, (2015).

Kay, M. G., “Production System Design [online]”, (15 Kasım 2016), https://courses.ncsu.edu/ise453/common/Lec_Notes_for_Prod_Sys_Des.pdf, (2016).

Internet - 1: Kocar, C., “Temel Lojistik Faaliyetleri Nelerdir? [online]”, (06 Ocak 2017), <http://cihankocarlojistik.blogspot.com.tr/2015/10/temel-lojistik-faaliyetleri-nelerdir.html>, (2015).

Lambert, D. M., Stock, J. R., Ellram, L. M., *Fundamentals of Logistics Management*: Irwin McGraw-Hill, (1998).

LaValle, S. M. and Hutchinson, S. A., “Optimal motion planning for multiple robots having independent goals.”, *IEEE Transactions on Robotics and Automation*, 14(6), 912-925, (1998).

Leroy, S., Laumond, J. P. and Siméon, T., “Multiple path coordination for mobile robots: A geometric algorithm.”, *In IJCAI*, Vol. 99, 1118-1123, (1999).

Liu, S., Mao, L. and Yu, J., “Path planning based on ant colony algorithm and distributed local navigation for multi-robot systems.”, *In Proceedings of the 2006 IEEE International Conference on Mechatronics and Automation*, 1733-1738, (2006).

Lumelsky, V. J. and Harinarayan, K. R., “Decentralized motion planning for multiple mobile robots: The cocktail party model.”, *In Robot Colonies*, 121-135, (1997).

Özalp, M. M., “Loistik Yönetiminde Yalın Düşünce İle Sistem Tasarımı ve Bir Uygulama”, Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi *Fen Bilimleri Enstitüsü, Endüstri Mühendisliği Anabilim Dalı*, İstanbul, (2013).

Öztürk, A., “Etkin Depo Yönetimi ve Lojistik Depoların Etkin Depo Stratejileri üzerine Bir Araştırma”, Yüksek Lisans Tezi, İstanbul Ticaret Üniversitesi *Sosyal Bilimler Enstitüsü Uluslararası Ticaret Anabilim Dalı, Uluslararası Ticaret Anabilim Dalı*, İstanbul, (2011).

Ratner, D. and Warmuth, M., “The $(n-1)$ -puzzle and related relocation problems.”, *Journal of Symbolic Computation*, 10(2), 111-137, (1990).

Reinefeld, A., “Complete Solution of the Eight-Puzzle and the Benefit of Node Ordering in IDA*.”, *In Proceedings of the International Joint Conference on Artificial Intelligence*, 248-253, (1993).

Rushton, A., Croucher, P. and Baker, P., *The Handbook of Logistics & Distribution Management*: Kogan Page, (2014).

Sanchez, G. and Latombe, J. C., “Using a PRM planner to compare centralized and decoupled planning for multi-robot systems.”, *In Proceedings of the IEEE International Conference on Robotics and Automation.*, Vol. 2, 2112-2119, (2002).

Sarrafzadeh, M. and Maddila, S. R., “Discrete warehouse problem.”, *Theoretical Computer Science*, 140(2), 231-247, (1995).

Sharma, R. and Aloimonos, Y. “Coordinated motion planning: the warehouseman's problem with constraints on free space.”, *IEEE Transactions on Systems, Man, and Cybernetics*, 22(1), 130-141, (1992).

Sittivijan, P., “Modular Warehouse Control: Simultaneous Rectilinear Movement of Multiple Objects within Limited Free Space Environment.”, Doktora Tezi, North Carolina State University *Department of Industrial Engineering*, (2015).

Spitznagel, E. L., “A new look at the fifteen puzzle.”, *Mathematics Magazine*, 40(4), 171-174, (1967).

Švestka, P. and Overmars, M. H., “Coordinated path planning for multiple robots.”, *Robotics and Autonomous Systems*, 23(3), 125-152, (1998).

Şahin, Y., “Depo Operasyonlarının Planlanması için Genetik Algoritma Esaslı Bir Model”, Yüksek Lisans Tezi, Pamukkale Üniversitesi *Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalı*, Denizli, (2009).

Ten Hompel, M., and Schmidt, T., *Warehouse management: Automation and Organisation of Warehouse and Order Picking Systems*, Berlin: Springer, (2007).

Van Den Berg, J. P. and Zijm, W. H. M., “Models for warehouse management: Classification and examples.”, *International Journal of Production Economics*, 59(1), 519-528, (1999).

Yamashita, A., Arai, T., Ota, J. and Asama, H., “Motion planning of multiple mobile robots for cooperative manipulation and transportation.”, *IEEE Transactions on Robotics and Automation*, 19(2), 223-237, (2003).

Yener, F., “Veri Madenciliği ve Optimizasyon Teknikleri Kullanılarak Bir Depo Tasarımı: Perakende Sektöründe Uygulaması”, Yüksek Lisans Tezi, Sakarya Üniversitesi *Fen Bilimleri Enstitüsü*, Sakarya, (2014).

Yeung, D. Y. and Bekey, G., “A decentralized approach to the motion planning problem for multiple mobile robots.”, *In Proceedings of the IEEE International Conference on Robotics and Automation*, Vol. 4, 1779-1784, (1987).

Xiang, L., Kay, M. G. and Telford, J., *Public Logistics Network Protocol Design and Implementation*, (2007).

EKLER

8. EKLER

EK A.1 ALG-B1'e Göre Tüm Algoritmalar İçin Bağıl Sapma Değer Karşılaştırması

ALG-B1'E GÖRE BAĞIL SAPMALAR								
ÖRNEK	ALG-B2	ALG-P1	ALG-P2	ALG-P3	ALG-P4	ALG-P5	Boyut	Yoğunluk
1	10,67	0,12	0,51	0,12	0,41	0,37	4*4	0,40
2	11,11	10,17	0,51	10,62	10,76	0,22	4*4	0,50
3	47,41	0,19	0,50	0,37	55,93	57,04	4*4	0,50
TOPLAM	9,84	3,18	0,15	3,60	23,23	20,32		

ALG-B1'E GÖRE BAĞIL SAPMALAR								
ÖRNEK	ALG-B2	ALG-P1	ALG-P2	ALG-P3	ALG-P4	ALG-P5	Boyut	Yoğunluk
4	70,02	0,98	0,64	0,24	1,10	0,98	6*6	0,40
5	5,81	0,22	0,60	0,97	0,97	1,25	6*6	0,50
6	105,09	2,57	0,60	0,68	0,25	0,08	6*6	0,60
7	13,06	19,50	0,02	18,90	0,55	18,37	6*6	0,60
8	38,37	0,92	2,27	1,42	2,35	2,37	6*6	0,70
TOPLAM	27,40	6,68	0,10	7,38	0,25	6,98		

ALG-B1'E GÖRE BAĞIL SAPMALAR								
ÖRNEK	ALG-B2	ALG-P1	ALG-P2	ALG-P3	ALG-P4	ALG-P5	Boyut	Yoğunluk
9	2,77	5,92	0,20	7,20	0,30	8,10	16*32	0,20
10	1,70	6,15	1,66	8,78	1,70	8,80	16*32	0,20
11	12,82	4,16	0,72	4,48	0,44	4,26	16*32	0,20
12	8,10	12,00	2,14	13,94	0,45	12,87	16*32	0,30
13	4,81	2,37	3,16	5,66	0,86	5,15	16*32	0,40
14	19,52	21,51	0,92	21,59	1,53	22,15	16*32	0,50
15	8,25	6,49	1,64	7,50	2,44	7,26	16*32	0,60
16	34,89	1,76	1,17	3,08	0,31	3,41	16*32	0,70
17	96,10	16,17	0,93	15,62	1,13	15,55	16*32	0,80
18	28,38	10,76	0,86	11,94	0,27	12,22	16*32	0,90
19	9,88	1,73	0,86	12,48	7,37	14,07	16*32	0,95
20	-	-	-	-	-	-	16*32	0,96
21	-	-	-	-	-	-	16*32	0,97
22	-	-	-	-	-	-	16*32	0,98
23	-	-	-	-	-	-	16*32	0,99
TOPLAM	15,70	3,39	0,75	7,72	2,32	8,16		

**EK A.2 ALG-B1'e Göre Tüm Algoritmalar İçin Göreceli Mutlak Sapma
Değer Karşılaştırması**

ALG-B1'E GÖRE MUTLAK SAPMALAR								
ÖRNEK	ALG-B2	ALG-P1	ALG-P2	ALG-P3	ALG-P4	ALG-P5	Boyut	Yoğunluk
1	0,52	0,01	0,03	0,01	0,02	0,02	4*4	0,40
2	0,55	0,50	0,03	0,52	0,53	0,01	4*4	0,50
3	2,56	0,01	0,03	0,02	3,02	3,08	4*4	0,50
TOPLAM	1,494	0,483	0,023	0,547	3,528	3,087		

ALG-B1'E GÖRE MUTLAK SAPMALAR								
ÖRNEK	ALG-B2	ALG-P1	ALG-P2	ALG-P3	ALG-P4	ALG-P5	Boyut	Yoğunluk
4	4,58	0,06	0,04	0,02	0,07	0,06	6*6	0,40
5	0,42	0,02	0,04	0,07	0,07	0,09	6*6	0,50
6	4,17	0,10	0,02	0,03	0,01	0,00	6*6	0,60
7	1,39	2,08	0,00	2,01	0,06	1,96	6*6	0,60
8	2,99	0,07	0,18	0,11	0,18	0,19	6*6	0,70
TOPLAM	7,775	1,895	0,027	2,095	0,071	1,98		

ALG-B1'E GÖRE MUTLAK SAPMALAR								
ÖRNEK	ALG-B2	ALG-P1	ALG-P2	ALG-P3	ALG-P4	ALG-P5	Boyut	Yoğunluk
9	0,48	1,03	0,04	1,25	0,05	1,41	16*32	0,20
10	0,30	1,08	0,29	1,54	0,30	1,55	16*32	0,20
11	1,97	0,64	0,11	0,69	0,07	0,65	16*32	0,20
12	1,41	2,09	0,37	2,43	0,08	2,24	16*32	0,30
13	0,97	0,48	0,64	1,14	0,17	1,03	16*32	0,40
14	5,53	6,09	0,26	6,12	0,43	6,27	16*32	0,50
15	2,03	1,59	0,40	1,84	0,60	1,78	16*32	0,60
16	10,99	0,56	0,37	0,97	0,10	1,07	16*32	0,70
17	36,26	6,10	0,35	5,89	0,43	5,87	16*32	0,80
18	14,94	5,67	0,45	6,29	0,14	6,43	16*32	0,90
19	8,13	1,43	0,71	10,27	6,07	11,58	16*32	0,95
20	-	-	-	-	-	-	16*32	0,96
21	-	-	-	-	-	-	16*32	0,97
22	-	-	-	-	-	-	16*32	0,98
23	-	-	-	-	-	-	16*32	0,99
TOPLAM	54,125	11,689	2,57	26,633	7,993	28,156		

**EK A.3 ALG-B1, ALG-P2, ALG-P4 Algoritmalarının Çalışma Süreleri
ve ALG-B1'e Göre Göreceli Karşılaştırmaları**

ALGORİTMA ÇÖZÜM SÜRELERİ (sn.) GÖRECELİ KARŞILAŞTIRMALAR								
ÖRNEK	ALG-B1	ALG-P2	ALG-P4	ALG-B1	ALG-P2	ALG-P4	Boyut	Yoğunluk
1	4,883	4,858	4,903	1	0,99	1,00	4*4	0,40
2	4,907	4,882	4,379	1	0,99	0,89	4*4	0,50
3	5,4	5,427	2,38	1	1,01	0,44	4*4	0,50
TOPLAM	15,19	15,167	11,662	1	1,00	0,77		

ALGORİTMA ÇÖZÜM SÜRELERİ (sn.) GÖRECELİ KARŞILAŞTIRMALAR								
ÖRNEK	ALG-B1	ALG-P2	ALG-P4	ALG-B1	ALG-P2	ALG-P4	Boyut	Yoğunluk
4	6,535	6,577	6,607	1	1,01	1,01	6*6	0,40
5	7,216	7,173	7,146	1	0,99	0,99	6*6	0,50
6	3,969	3,945	3,979	1	0,99	1,00	6*6	0,60
7	10,654	10,652	10,713	1	1,00	1,01	6*6	0,60
8	7,798	7,621	7,615	1	0,98	0,98	6*6	0,70
TOPLAM	36,172	35,968	36,06	1	0,99	1,00		

ALGORİTMA ÇÖZÜM SÜRELERİ (sn.) GÖRECELİ KARŞILAŞTIRMALAR								
ÖRNEK	ALG-B1	ALG-P2	ALG-P4	ALG-B1	ALG-P2	ALG-P4	Boyut	Yoğunluk
9	17,355	17,32	17,303	1	1,00	1,00	16*32	0,20
10	17,557	17,266	17,259	1	0,98	0,98	16*32	0,20
11	15,354	15,243	15,287	1	0,99	1,00	16*32	0,20
12	17,423	17,05	17,502	1	0,98	1,00	16*32	0,30
13	20,073	19,438	19,9	1	0,97	0,99	16*32	0,40
14	28,325	28,064	27,893	1	0,99	0,98	16*32	0,50
15	24,543	24,14	23,943	1	0,98	0,98	16*32	0,60
16	31,5	31,132	31,403	1	0,99	1,00	16*32	0,70
17	37,732	37,38	37	1	0,99	0,99	16*32	0,80
18	52,658	52,206	52,8	1	0,99	1,00	16*32	0,90
19	83,755	82,327	83,038	1	0,98	0,99	16*32	0,95
20	-	-	-	-	-	-	16*32	0,96
21	-	-	-	-	-	-	16*32	0,97
22	-	-	-	-	-	-	16*32	0,98
23	-	-	-	-	-	-	16*32	0,99
TOPLAM	346,275	341,566	343,632	1	0,99	0,99		

**EK A.4 ALG-P1, ALG-P3, ALG-P5 Algoritmalarının Çalışma Süreleri
ve ALG-P1'e Göre Göreceli Karşılaştırmaları**

ALGORİTMA ÇÖZÜM SÜRELERİ (sn.) GÖRECELİ KARŞILAŞTIRMALAR								
ÖRNEK	ALG-P1	ALG-P3	ALG-P5	ALG-P1	ALG-P3	ALG-P5	Boyut	Yoğunluk
1	4,889	4,877	4,865	1	1,00	1,00	4*4	0,40
2	4,408	4,386	4,918	1	1,00	1,12	4*4	0,50
3	5,41	5,38	2,32	1	0,99	0,43	4*4	0,50
TOPLAM	14,707	14,643	12,103	1	1,00	0,82		

ALGORİTMA ÇÖZÜM SÜRELERİ (sn.) GÖRECELİ KARŞILAŞTIRMALAR								
ÖRNEK	ALG-P1	ALG-P3	ALG-P5	ALG-P1	ALG-P3	ALG-P5	Boyut	Yoğunluk
4	6,599	6,551	6,599	1	0,99	1,00	6*6	0,40
5	7,232	7,146	7,126	1	0,99	0,99	6*6	0,50
6	4,071	3,942	3,972	1	0,97	0,98	6*6	0,60
7	8,577	8,64	8,697	1	1,01	1,01	6*6	0,60
8	7,726	7,687	7,613	1	0,99	0,99	6*6	0,70
TOPLAM	34,205	33,966	34,007	1	0,99	0,99		

ALGORİTMA ÇÖZÜM SÜRELERİ (sn.) GÖRECELİ KARŞILAŞTIRMALAR								
ÖRNEK	ALG-P1	ALG-P3	ALG-P5	ALG-P1	ALG-P3	ALG-P5	Boyut	Yoğunluk
9	16,327	16,105	15,95	1	0,99	0,98	16*32	0,20
10	16,478	16,016	16,012	1	0,97	0,97	16*32	0,20
11	14,715	14,666	14,7	1	1,00	1,00	16*32	0,20
12	15,333	14,994	15,18	1	0,98	0,99	16*32	0,30
13	19,597	18,937	19,04	1	0,97	0,97	16*32	0,40
14	22,232	22,21	22,052	1	1,00	0,99	16*32	0,50
15	22,951	22,702	22,76	1	0,99	0,99	16*32	0,60
16	30,945	30,53	30,426	1	0,99	0,98	16*32	0,70
17	43,833	43,626	43,6	1	1,00	0,99	16*32	0,80
18	46,992	46,372	46,224	1	0,99	0,98	16*32	0,90
19	76,26	72,056	70,747	1	0,94	0,93	16*32	0,95
20	122,928	120,902	120,445	1	0,98	0,98	16*32	0,96
21	-	-	-	-	-	-	16*32	0,97
22	-	-	-	-	-	-	16*32	0,98
23	-	-	-	-	-	-	16*32	0,99
TOPLAM	448,591	439,116	437,136	1	0,98	0,97		

EK A.5 ALG-B1, ALG-P1, ALG-P3, ALG-P5 Algoritmalarının Çalışma Süreleri ve ALG-B1'e Göre Göreceli Karşılaştırmaları

ALGORİTMA ÇÖZÜM SÜRELERİ (sn.) GÖRECELİ KARŞILAŞTIRMALAR										
ÖRNEK	ALG-B1	ALG-P1	ALG-P3	ALG-P5	ALG-B1	ALG-P1	ALG-P3	ALG-P5	Boyut	Yoğunluk
1	4,883	4,889	4,877	4,865	1	1,00	1,00	1,00	4*4	0,40
2	4,907	4,408	4,386	4,918	1	0,90	0,89	1,00	4*4	0,50
3	5,4	5,41	5,38	2,32	1	1,00	1,00	0,43	4*4	0,50
TOPLAM	15,19	14,707	14,643	12,103	1	0,97	0,96	0,80		

ALGORİTMA ÇÖZÜM SÜRELERİ (sn.) GÖRECELİ KARŞILAŞTIRMALAR										
ÖRNEK	ALG-B1	ALG-P1	ALG-P3	ALG-P5	ALG-B1	ALG-P1	ALG-P3	ALG-P5	Boyut	Yoğunluk
4	6,535	6,599	6,551	6,599	1	1,01	1,00	1,01	6*6	0,40
5	7,216	7,232	7,146	7,126	1	1,00	0,99	0,99	6*6	0,50
6	3,969	4,071	3,942	3,972	1	1,03	0,99	1,00	6*6	0,60
7	10,654	8,577	8,64	8,697	1	0,81	0,81	0,82	6*6	0,60
8	7,798	7,726	7,687	7,613	1	0,99	0,99	0,98	6*6	0,70
TOPLAM	36,172	34,205	33,966	34,007	1	0,95	0,94	0,94		

ALGORİTMA ÇÖZÜM SÜRELERİ (sn.) GÖRECELİ KARŞILAŞTIRMALAR										
ÖRNEK	ALG-B1	ALG-P1	ALG-P3	ALG-P5	ALG-B1	ALG-P1	ALG-P3	ALG-P5	Boyut	Yoğunluk
9	17,355	16,327	16,105	15,95	1	0,94	0,93	0,92	16*32	0,20
10	17,557	16,478	16,016	16,012	1	0,94	0,91	0,91	16*32	0,20
11	15,354	14,715	14,666	14,7	1	0,96	0,96	0,96	16*32	0,20
12	17,423	15,333	14,994	15,18	1	0,88	0,86	0,87	16*32	0,30
13	20,073	19,597	18,937	19,04	1	0,98	0,94	0,95	16*32	0,40
14	28,325	22,232	22,21	22,052	1	0,78	0,78	0,78	16*32	0,50
15	24,543	22,951	22,702	22,76	1	0,94	0,92	0,93	16*32	0,60
16	31,5	30,945	30,53	30,426	1	0,98	0,97	0,97	16*32	0,70
17	37,732	43,833	43,626	43,6	1	1,16	1,16	1,16	16*32	0,80
18	52,658	46,992	46,372	46,224	1	0,89	0,88	0,88	16*32	0,90
19	82,327	76,26	72,056	70,747	1	0,93	0,88	0,86	16*32	0,95
20	-	122,928	120,902	120,445	-	-	-	-	16*32	0,96
21	-	-	-	-	-	-	-	-	16*32	0,97
22	-	-	-	-	-	-	-	-	16*32	0,98
23	-	-	-	-	-	-	-	-	16*32	0,99
TOPLAM	344,847	325,663	318,214	316,691	1	0,94	0,92	0,92		

EK A.6 Rotalama Ufkundaki Adım Sayısına (k) Göre Çözüm Adım Sayılarının Karşılaştırılması

ÇÖZÜM ADIM SAYISI									
	ÖRNEK	ALG-B1	ALG-P2	ALG-P4	ALG-P1	ALG-P3	ALG-P5	Boyut	Yoğunluk
k=3	1	8	8	8	8	8	8	4*4	0,40
	2	8	8	7	7	7	8	4*4	0,50
	3	9	9	3	9	9	3	4*4	0,50
	TOPLAM	25	25	18	24	24	19		
k=4	ÖRNEK	ALG-B1	ALG-P2	ALG-P4	ALG-P1	ALG-P3	ALG-P5	Boyut	Yoğunluk
	1	8	8	8	8	8	8	4*4	0,40
	2	8	8	7	7	7	8	4*4	0,50
	3	9	9	3	9	9	3	4*4	0,50
	TOPLAM	25	25	18	24	24	19		
k=5	ÖRNEK	ALG-B1	ALG-P2	ALG-P4	ALG-P1	ALG-P3	ALG-P5	Boyut	Yoğunluk
	1	8	8	8	8	8	8	4*4	0,40
	2	8	8	8	7	7	7	4*4	0,50
	3	9	9	3	9	9	3	4*4	0,50
	TOPLAM	25	25	19	24	24	18		

ÇÖZÜM ADIM SAYISI									
	ÖRNEK	ALG-B1	ALG-P2	ALG-P4	ALG-P1	ALG-P3	ALG-P5	Boyut	Yoğunluk
k=3	3	6	6	6	6	6	6	8*8	0,60
	4	13	13	13	13	13	13	8*8	0,70
	5	11	11	11	11	11	11	8*8	0,40
	6	12	12	12	12	12	12	8*8	0,50
	7	19	19	19	15	15	15	8*8	0,60
	TOPLAM	61	61	61	57	57	57		
		ÖRNEK	ALG-B1	ALG-P2	ALG-P4	ALG-P1	ALG-P3	ALG-P5	Boyut
k=4	3	6	6	6	6	6	6	8*8	0,60
	4	13	13	13	13	13	13	8*8	0,70
	5	11	11	11	11	11	11	8*8	0,40
	6	12	12	12	12	12	12	8*8	0,50
	7	19	19	19	15	15	15	8*8	0,60
	TOPLAM	61	61	61	57	57	57		
k=5	ÖRNEK	ALG-B1	ALG-P2	ALG-P4	ALG-P1	ALG-P3	ALG-P5	Boyut	Yoğunluk
	3	6	6	6	6	6	6	8*8	0,60
	4	13	13	13	13	13	12	8*8	0,70
	5	11	11	11	11	11	11	8*8	0,40
	6	12	12	12	12	12	13	8*8	0,50
	7	19	19	19	15	15	15	8*8	0,60
	TOPLAM	61	61	61	57	57	57		

ÇÖZÜM ADIM SAYISI

	ÖRNEK	ALG-B1	ALG-P2	ALG-P4	ALG-P1	ALG-P3	ALG-P5	Boyut	Yoğunluk
k=3	9	29	29	29	27	27	27	16*32	0,20
	10	29	29	29	27	27	27	16*32	0,20
	11	26	26	26	24	24	24	16*32	0,20
	12	29	29	29	25	25	25	16*32	0,30
	13	32	32	32	33	33	33	16*32	0,40
	14	47	47	47	37	37	37	16*32	0,50
	15	41	41	41	38	38	38	16*32	0,60
	16	53	53	53	52	52	52	16*32	0,70
	17	64	64	64	75	75	75	16*32	0,80
	18	89	89	89	80	80	82	16*32	0,90
	19	143	143	143	125	125	125	16*32	0,95
	20	-	-	-	205	205	208	16*32	0,96
	21	-	-	-	-	-	-	16*32	0,97
	22	-	-	-	-	-	-	16*32	0,98
23	-	-	-	-	-	-	16*32	0,99	
	TOPLAM	582	582	582	543	543	545		
k=4	ÖRNEK	ALG-B1	ALG-P2	ALG-P4	ALG-P1	ALG-P3	ALG-P5	Boyut	Yoğunluk
	9	29	29	29	27	27	27	16*32	0,20
	10	29	29	29	27	27	27	16*32	0,20
	11	26	26	26	23	23	23	16*32	0,20
	12	29	29	29	25	25	25	16*32	0,30
	13	29	29	29	33	33	33	16*32	0,40
	14	40	40	40	34	34	34	16*32	0,50
	15	41	41	41	39	39	39	16*32	0,60
	16	53	53	53	52	52	52	16*32	0,70
	17	64	64	64	75	75	75	16*32	0,80
	18	89	89	89	80	80	82	16*32	0,90
	19	143	143	143	125	125	125	16*32	0,95
	20	-	-	-	205	205	208	16*32	0,96
	21	-	-	-	-	-	-	16*32	0,97
22	-	-	-	-	-	-	16*32	0,98	
23	-	-	-	-	-	-	16*32	0,99	
	TOPLAM	572	572	572	540	540	542		
k=5	ÖRNEK	ALG-B1	ALG-P2	ALG-P4	ALG-P1	ALG-P3	ALG-P5	Boyut	Yoğunluk
	9	29	29	29	27	27	27	16*32	0,20
	10	29	29	29	27	27	27	16*32	0,20
	11	26	26	26	23	23	23	16*32	0,20
	12	29	29	29	25	25	25	16*32	0,30
	13	29	29	29	33	33	33	16*32	0,40
	14	40	40	40	34	34	34	16*32	0,50
	15	41	41	41	39	39	39	16*32	0,60
	16	53	53	53	52	52	52	16*32	0,70
	17	64	64	64	75	75	75	16*32	0,80
	18	89	89	89	80	80	82	16*32	0,90
	19	143	143	143	125	125	125	16*32	0,95
	20	-	-	-	205	205	202	16*32	0,96
	21	-	-	-	-	-	-	16*32	0,97
22	-	-	-	-	-	-	16*32	0,98	
23	-	-	-	-	-	-	16*32	0,99	
	TOPLAM	572	572	572	540	540	542		

EK A.7 Çözüm Adım Sayılarının ALG-B1'e Göre Göreceli Karşılaştırılması

GÖRECELİ KARŞILAŞTIRMALAR									
	ÖRNEK	ALG-B1	ALG-P2	ALG-P4	ALG-P1	ALG-P3	ALG-P5	Boyut	Yoğunluk
k=3	1	1,00	1,00	1,00	1,00	1,00	1,00	4*4	0,40
	2	1,00	1,00	0,88	0,88	0,88	1,00	4*4	0,50
	3	1,00	1,00	0,33	1,00	1,00	0,33	4*4	0,50
	TOPLAM	1,00	1,00	0,72	0,96	0,96	0,76		
k=4	ÖRNEK	ALG-B1	ALG-P2	ALG-P4	ALG-P1	ALG-P3	ALG-P5	Boyut	Yoğunluk
	1	1,00	1,00	1,00	1,00	1,00	1,00	4*4	0,40
	2	1,00	1,00	0,88	0,88	0,88	1,00	4*4	0,50
	3	1,00	1,00	0,33	1,00	1,00	0,33	4*4	0,50
TOPLAM	1,00	1,00	0,72	0,96	0,96	0,76			
k=5	ÖRNEK	ALG-B1	ALG-P2	ALG-P4	ALG-P1	ALG-P3	ALG-P5	Boyut	Yoğunluk
	1	1,00	1,00	1,00	1,00	1,00	1,00	4*4	0,40
	2	1,00	1,00	1,00	0,88	0,88	0,88	4*4	0,50
	3	1,00	1,00	0,33	1,00	1,00	0,33	4*4	0,50
TOPLAM	1,00	1,00	0,76	0,96	0,96	0,72			

GÖRECELİ KARŞILAŞTIRMALAR									
	ÖRNEK	ALG-B1	ALG-P2	ALG-P4	ALG-P1	ALG-P3	ALG-P5	Boyut	Yoğunluk
k=3	3	1,00	1,00	1,00	1,00	1,00	1,00	8*8	0,60
	4	1,00	1,00	1,00	1,00	1,00	1,00	8*8	0,70
	5	1,00	1,00	1,00	1,00	1,00	1,00	8*8	0,40
	6	1,00	1,00	1,00	1,00	1,00	1,00	8*8	0,50
	7	1,00	1,00	1,00	0,79	0,79	0,79	8*8	0,60
	TOPLAM	1,00	1,00	1,00	0,93	0,93	0,93		
k=4	ÖRNEK	ALG-B1	ALG-P2	ALG-P4	ALG-P1	ALG-P3	ALG-P5	Boyut	Yoğunluk
	3	1,00	1,00	1,00	1,00	1,00	1,00	8*8	0,60
	4	1,00	1,00	1,00	1,00	1,00	1,00	8*8	0,70
	5	1,00	1,00	1,00	1,00	1,00	1,00	8*8	0,40
	6	1,00	1,00	1,00	1,00	1,00	1,00	8*8	0,50
	7	1,00	1,00	1,00	0,79	0,79	0,79	8*8	0,60
TOPLAM	1,00	1,00	1,00	0,93	0,93	0,93			
k=5	ÖRNEK	ALG-B1	ALG-P2	ALG-P4	ALG-P1	ALG-P3	ALG-P5	Boyut	Yoğunluk
	3	1,00	1,00	1,00	1,00	1,00	1,00	8*8	0,60
	4	1,00	1,00	1,00	1,00	1,00	0,92	8*8	0,70
	5	1,00	1,00	1,00	1,00	1,00	1,00	8*8	0,40
	6	1,00	1,00	1,00	1,00	1,00	1,08	8*8	0,50
	7	1,00	1,00	1,00	0,79	0,79	0,79	8*8	0,60
TOPLAM	1,00	1,00	1,00	0,93	0,93	0,93			

GÖRECELİ KARŞILAŞTIRMALAR									
ÖRNEK	ALG-B1	ALG-P2	ALG-P4	ALG-P1	ALG-P3	ALG-P5	Boyut	Yoğunluk	
k=3	9	1,00	1,00	1,00	0,93	0,93	0,93	16*32	0,20
	10	1,00	1,00	1,00	0,93	0,93	0,93	16*32	0,20
	11	1,00	1,00	1,00	0,92	0,92	0,92	16*32	0,20
	12	1,00	1,00	1,00	0,86	0,86	0,86	16*32	0,30
	13	1,00	1,00	1,00	1,03	1,03	1,03	16*32	0,40
	14	1,00	1,00	1,00	0,79	0,79	0,79	16*32	0,50
	15	1,00	1,00	1,00	0,93	0,93	0,93	16*32	0,60
	16	1,00	1,00	1,00	0,98	0,98	0,98	16*32	0,70
	17	1,00	1,00	1,00	1,17	1,17	1,17	16*32	0,80
	18	1,00	1,00	1,00	0,90	0,90	0,92	16*32	0,90
	19	1,00	1,00	1,00	0,87	0,87	0,87	16*32	0,95
	20	-	-	-	-	-	-	16*32	0,96
	21	-	-	-	-	-	-	16*32	0,97
	22	-	-	-	-	-	-	16*32	0,98
23	-	-	-	-	-	-	16*32	0,99	
TOPLAM	1,00	1,00	1,00	0,93	0,93	0,94			
k=4	ÖRNEK	ALG-B1	ALG-P2	ALG-P4	ALG-P1	ALG-P3	ALG-P5	Boyut	Yoğunluk
	9	1,00	1,00	1,00	0,93	0,93	0,93	16*32	0,20
	10	1,00	1,00	1,00	0,93	0,93	0,93	16*32	0,20
	11	1,00	1,00	1,00	0,88	0,88	0,88	16*32	0,20
	12	1,00	1,00	1,00	0,86	0,86	0,86	16*32	0,30
	13	1,00	1,00	1,00	1,14	1,14	1,14	16*32	0,40
	14	1,00	1,00	1,00	0,85	0,85	0,85	16*32	0,50
	15	1,00	1,00	1,00	0,95	0,95	0,95	16*32	0,60
	16	1,00	1,00	1,00	0,98	0,98	0,98	16*32	0,70
	17	1,00	1,00	1,00	1,17	1,17	1,17	16*32	0,80
	18	1,00	1,00	1,00	0,90	0,90	0,92	16*32	0,90
	19	1,00	1,00	1,00	0,87	0,87	0,87	16*32	0,95
	20	-	-	-	-	-	-	16*32	0,96
	21	-	-	-	-	-	-	16*32	0,97
22	-	-	-	-	-	-	16*32	0,98	
23	-	-	-	-	-	-	16*32	0,99	
TOPLAM	1,00	1,00	1,00	0,94	0,94	0,95			
k=5	ÖRNEK	ALG-B1	ALG-P2	ALG-P4	ALG-P1	ALG-P3	ALG-P5	Boyut	Yoğunluk
	9	1,00	1,00	1,00	0,93	0,93	0,93	16*32	0,20
	10	1,00	1,00	1,00	0,93	0,93	0,93	16*32	0,20
	11	1,00	1,00	1,00	0,88	0,88	0,88	16*32	0,20
	12	1,00	1,00	1,00	0,86	0,86	0,86	16*32	0,30
	13	1,00	1,00	1,00	1,14	1,14	1,14	16*32	0,40
	14	1,00	1,00	1,00	0,85	0,85	0,85	16*32	0,50
	15	1,00	1,00	1,00	0,95	0,95	0,95	16*32	0,60
	16	1,00	1,00	1,00	0,98	0,98	0,98	16*32	0,70
	17	1,00	1,00	1,00	1,17	1,17	1,17	16*32	0,80
	18	1,00	1,00	1,00	0,90	0,90	0,92	16*32	0,90
	19	1,00	1,00	1,00	0,87	0,87	0,87	16*32	0,95
	20	-	-	-	-	-	-	16*32	0,96
	21	-	-	-	-	-	-	16*32	0,97
22	-	-	-	-	-	-	16*32	0,98	
23	-	-	-	-	-	-	16*32	0,99	
TOPLAM	1,00	1,00	1,00	0,94	0,94	0,95			

EK A.8 Rotalama Ufkundaki Adım Sayısına (k) Göre Çözüm Sürelerinin Karşılaştırılması

		ÇÖZÜM ADIM SAYISI								
		ÖRNEK	ALG-B1	ALG-P2	ALG-P4	ALG-P1	ALG-P3	ALG-P5	Boyut	Yoğunluk
k=3	1	4,883	4,858	4,903	4,889	4,877	4,865	4*4	0,40	
	2	4,907	4,882	4,379	4,408	4,386	4,918	4*4	0,50	
	3	5,4	5,427	2,38	5,41	5,38	2,32	4*4	0,50	
	TOPLAM	15,19	15,167	11,662	14,707	14,643	12,103			
k=4	ÖRNEK	ALG-B1	ALG-P2	ALG-P4	ALG-P1	ALG-P3	ALG-P5	Boyut	Yoğunluk	
	1	4,88	4,83	4,84	4,86	4,85	4,86	4*4	0,40	
	2	4,85	4,88	4,36	4,34	4,33	4,9	4*4	0,50	
	3	5,38	5,34	2,29	5,35	5,36	2,34	4*4	0,50	
TOPLAM	15,11	15,05	11,49	14,55	14,54	12,1				
k=5	ÖRNEK	ALG-B1	ALG-P2	ALG-P4	ALG-P1	ALG-P3	ALG-P5	Boyut	Yoğunluk	
	1	4,84	4,84	4,87	4,88	4,88	4,9	4*4	0,40	
	2	4,87	4,83	4,86	4,37	4,33	4,36	4*4	0,50	
	3	5,41	5,37	2,3	5,37	5,37	2,3	4*4	0,50	
TOPLAM	15,12	15,04	12,03	14,62	14,58	11,56				

		ÇÖZÜM ADIM SAYISI								
		ÖRNEK	ALG-B1	ALG-P2	ALG-P4	ALG-P1	ALG-P3	ALG-P5	Boyut	Yoğunluk
k=3	4	6,535	6,577	6,607	6,599	6,551	6,599	6*6	0,40	
	5	7,216	7,173	7,145	7,232	7,146	7,126	6*6	0,50	
	6	3,969	3,945	3,979	4,071	3,942	3,972	6*6	0,60	
	7	10,654	10,652	10,713	8,577	8,64	8,697	6*6	0,60	
	8	7,798	7,621	7,615	7,726	7,687	7,613	6*6	0,70	
	TOPLAM	36,172	35,968	36,059	34,205	33,966	34,007			
k=4	ÖRNEK	ALG-B1	ALG-P2	ALG-P4	ALG-P1	ALG-P3	ALG-P5	Boyut	Yoğunluk	
	4	6,52	6,48	6,6	6,59	6,52	6,61	6*6	0,40	
	5	7,05	7,01	7,07	7,1	7,06	7,06	6*6	0,50	
	6	3,91	3,89	3,95	3,92	3,92	3,94	6*6	0,60	
	7	10,75	10,73	10,61	8,65	8,64	8,65	6*6	0,60	
	8	7,6	7,5	7,59	7,64	7,63	7,61	6*6	0,70	
TOPLAM	35,83	35,61	35,82	33,9	33,77	33,87				
k=5	ÖRNEK	ALG-B1	ALG-P2	ALG-P4	ALG-P1	ALG-P3	ALG-P5	Boyut	Yoğunluk	
	4	6,54	6,54	6,6	6,54	6,56	6,59	6*6	0,40	
	5	7,14	7,09	7,06	7,09	7,06	7,07	6*6	0,50	
	6	3,91	3,9	3,95	3,9	3,91	3,9	6*6	0,60	
	7	10,66	10,26	10,62	8,662	8,62	8,16	6*6	0,60	
	8	7,57	7,57	7,56	7,62	7,56	7,67	6*6	0,70	
TOPLAM	35,82	35,36	35,79	33,812	33,71	33,39				

ÇÖZÜM ADIM SAYISI									
ÖRNEK	ALG-B1	ALG-P2	ALG-P4	ALG-P1	ALG-P3	ALG-P5	Boyut	Yoğunluk	
k=3	9	17,355	17,32	17,303	16,327	16,105	15,95	16*32	0,20
	10	17,557	17,266	17,259	16,478	16,016	16,012	16*32	0,20
	11	15,354	15,243	15,287	14,715	14,666	14,7	16*32	0,20
	12	17,423	17,05	17,502	15,333	14,994	15,18	16*32	0,30
	13	20,073	19,438	19,9	19,597	18,937	19,04	16*32	0,40
	14	28,325	28,064	27,893	22,232	22,21	22,052	16*32	0,50
	15	24,543	24,14	23,943	22,951	22,702	22,76	16*32	0,60
	16	31,5	31,132	31,403	30,945	30,53	30,426	16*32	0,70
	17	37,732	37,38	37	43,833	43,626	43,6	16*32	0,80
	18	52,658	52,206	52,8	46,992	46,372	46,224	16*32	0,90
	19	83,755	82,327	83,038	76,26	72,056	70,747	16*32	0,95
	20	-	-	-	122,928	120,902	120,445	16*32	0,96
	21	-	-	-	-	-	-	16*32	0,97
	22	-	-	-	-	-	-	16*32	0,98
23	-	-	-	-	-	-	16*32	0,99	
TOPLAM	346,275	341,566	343,632	325,663	318,214	316,691			
k=4	ÖRNEK	ALG-B1	ALG-P2	ALG-P4	ALG-P1	ALG-P3	ALG-P5	Boyut	Yoğunluk
	9	17,02	17,062	17,475	16,552	16,312	16,352	16*32	0,20
	10	17,31	17,244	17,454	16,612	16,58	16,692	16*32	0,20
	11	15,582	15,632	16,122	14,592	14,842	14,672	16*32	0,20
	12	17,88	17,89	18,256	15,832	15,764	15,754	16*32	0,30
	13	17,63	17,442	18,052	17,792	17,7	17,594	16*32	0,40
	14	23,54	23,812	24,372	20,772	20,97	21,165	16*32	0,50
	15	24,242	24,022	25,24	23,924	24,162	23,934	16*32	0,60
	16	31,052	31,022	31,782	31,336	31,214	31,332	16*32	0,70
	17	37,264	37,322	38,59	44,356	44,51	44,654	16*32	0,80
	18	52,364	52,266	53,628	47,252	47,71	47,454	16*32	0,90
	19	82,777	82,988	82,754	72,421	72,367	73,32	16*32	0,95
	20	-	-	-	121,978	121,273	121,803	16*32	0,96
	21	-	-	-	-	-	-	16*32	0,97
22	-	-	-	-	-	-	16*32	0,98	
23	-	-	-	-	-	-	16*32	0,99	
TOPLAM	336,661	336,702	343,725	321,441	322,131	322,923			
k=5	ÖRNEK	ALG-B1	ALG-P2	ALG-P4	ALG-P1	ALG-P3	ALG-P5	Boyut	Yoğunluk
	9	17,042	16,972	17,652	16,654	16,512	16,312	16*32	0,20
	10	17,232	17,06	17,75	16,526	16,69	16,774	16*32	0,20
	11	15,632	15,722	16,222	14,812	14,69	14,512	16*32	0,20
	12	17,85	17,762	18,322	15,712	15,84	15,694	16*32	0,30
	13	17,574	17,552	18,214	17,742	17,754	17,58	16*32	0,40
	14	23,542	23,6	24,342	20,953	21,142	20,97	16*32	0,50
	15	24,272	23,857	24,842	24,04	23,984	24,194	16*32	0,60
	16	31,003	31,172	32,034	31,16	31,26	31,796	16*32	0,70
	17	37,34	37,253	38,342	44,284	44,563	44,554	16*32	0,80
	18	52,254	51,642	53,582	47,244	47,304	47,672	16*32	0,90
	19	85,024	82,984	84,574	71,831	72,433	71,9	16*32	0,95
	20	-	-	-	121,975	122,962	122,71	16*32	0,96
	21	-	-	-	-	-	-	16*32	0,97
22	-	-	-	-	-	-	16*32	0,98	
23	-	-	-	-	-	-	16*32	0,99	
TOPLAM	338,765	335,576	345,876	320,958	322,172	321,958			

EK A.9 Çözüm Sürelerinin ALG-B1'e Göre Göreceli Karşılaştırılması

GÖRECELİ KARŞILAŞTIRMALAR									
	ÖRNEK	ALG-B1	ALG-P2	ALG-P4	ALG-P1	ALG-P3	ALG-P5	Boyut	Yoğunluk
k=3	1	1,00	0,99	1,00	1,00	1,00	1,00	4*4	0,40
	2	1,00	0,99	0,89	0,90	0,89	1,00	4*4	0,50
	3	1,00	1,01	0,44	1,00	1,00	0,43	4*4	0,50
	TOPLAM	1,00	1,00	0,77	0,97	0,96	0,80		
k=4	ÖRNEK	ALG-B1	ALG-P2	ALG-P4	ALG-P1	ALG-P3	ALG-P5	Boyut	Yoğunluk
	1	1,00	0,99	0,99	1,00	0,99	1,00	4*4	0,40
	2	1,00	1,01	0,90	0,89	0,89	1,01	4*4	0,50
	3	1,00	0,99	0,43	0,99	1,00	0,43	4*4	0,50
TOPLAM	1,00	1,00	0,76	0,96	0,96	0,80			
k=5	ÖRNEK	ALG-B1	ALG-P2	ALG-P4	ALG-P1	ALG-P3	ALG-P5	Boyut	Yoğunluk
	1	1,00	1,00	1,01	1,01	1,01	1,01	4*4	0,40
	2	1,00	0,99	1,00	0,90	0,89	0,90	4*4	0,50
	3	1,00	0,99	0,43	0,99	0,99	0,43	4*4	0,50
TOPLAM	1,00	0,99	0,80	0,97	0,96	0,76			

GÖRECELİ KARŞILAŞTIRMALAR									
	ÖRNEK	ALG-B1	ALG-P2	ALG-P4	ALG-P1	ALG-P3	ALG-P5	Boyut	Yoğunluk
k=3	4	1,00	1,01	1,01	1,01	1,00	1,01	6*6	0,40
	5	1,00	0,99	0,99	1,00	0,99	0,99	6*6	0,50
	6	1,00	0,99	1,00	1,03	0,99	1,00	6*6	0,60
	7	1,00	1,00	1,01	0,81	0,81	0,82	6*6	0,60
	8	1,00	0,98	0,98	0,99	0,99	0,98	6*6	0,70
TOPLAM	1,00	0,99	1,00	0,95	0,94	0,94			
k=4	ÖRNEK	ALG-B1	ALG-P2	ALG-P4	ALG-P1	ALG-P3	ALG-P5	Boyut	Yoğunluk
	4	1,00	0,99	1,01	1,01	1,00	1,01	6*6	0,40
	5	1,00	0,99	1,00	1,01	1,00	1,00	6*6	0,50
	6	1,00	0,99	1,01	1,00	1,00	1,01	6*6	0,60
	7	1,00	1,00	0,99	0,80	0,80	0,80	6*6	0,60
8	1,00	0,99	1,00	1,01	1,00	1,00	6*6	0,70	
TOPLAM	1,00	0,99	1,00	0,95	0,94	0,95			
k=5	ÖRNEK	ALG-B1	ALG-P2	ALG-P4	ALG-P1	ALG-P3	ALG-P5	Boyut	Yoğunluk
	4	1,00	1,00	1,01	1,00	1,00	1,01	6*6	0,40
	5	1,00	0,99	0,99	0,99	0,99	0,99	6*6	0,50
	6	1,00	1,00	1,01	1,00	1,00	1,00	6*6	0,60
	7	1,00	0,96	1,00	0,81	0,81	0,77	6*6	0,60
8	1,00	1,00	1,00	1,01	1,00	1,01	6*6	0,70	
TOPLAM	1,00	0,99	1,00	0,94	0,94	0,93			

GÖRECELİ KARŞILAŞTIRMALAR									
ÖRNEK	ALG-B1	ALG-P2	ALG-P4	ALG-P1	ALG-P3	ALG-P5	Boyut	Yoğunluk	
k=3	9	1,00	1,00	1,00	0,94	0,93	0,92	16*32	0,20
	10	1,00	0,98	0,98	0,94	0,91	0,91	16*32	0,20
	11	1,00	0,99	1,00	0,96	0,96	0,96	16*32	0,20
	12	1,00	0,98	1,00	0,88	0,86	0,87	16*32	0,30
	13	1,00	0,97	0,99	0,98	0,94	0,95	16*32	0,40
	14	1,00	0,99	0,98	0,78	0,78	0,78	16*32	0,50
	15	1,00	0,98	0,98	0,94	0,92	0,93	16*32	0,60
	16	1,00	0,99	1,00	0,98	0,97	0,97	16*32	0,70
	17	1,00	0,99	0,99	1,16	1,16	1,16	16*32	0,80
	18	1,00	0,99	1,00	0,89	0,88	0,88	16*32	0,90
	19	1,00	0,98	0,99	0,91	0,86	0,84	16*32	0,95
	20	-	-	-	-	-	-	16*32	0,96
	21	-	-	-	-	-	-	16*32	0,97
	22	-	-	-	-	-	-	16*32	0,98
23	-	-	-	-	-	-	16*32	0,99	
TOPLAM	1,00	0,99	0,99	0,94	0,92	0,91			
k=4	ÖRNEK	ALG-B1	ALG-P2	ALG-P4	ALG-P1	ALG-P3	ALG-P5	Boyut	Yoğunluk
	9	1,00	1,00	1,03	0,97	0,96	0,96	16*32	0,20
	10	1,00	1,00	1,01	0,96	0,96	0,96	16*32	0,20
	11	1,00	1,00	1,03	0,94	0,95	0,94	16*32	0,20
	12	1,00	1,00	1,02	0,89	0,88	0,88	16*32	0,30
	13	1,00	0,99	1,02	1,01	1,00	1,00	16*32	0,40
	14	1,00	1,01	1,04	0,88	0,89	0,90	16*32	0,50
	15	1,00	0,99	1,04	0,99	1,00	0,99	16*32	0,60
	16	1,00	1,00	1,02	1,01	1,01	1,01	16*32	0,70
	17	1,00	1,00	1,04	1,19	1,19	1,20	16*32	0,80
	18	1,00	1,00	1,02	0,90	0,91	0,91	16*32	0,90
	19	1,00	1,00	1,00	0,87	0,87	0,89	16*32	0,95
	20	-	-	-	-	-	-	16*32	0,96
	21	-	-	-	-	-	-	16*32	0,97
22	-	-	-	-	-	-	16*32	0,98	
23	-	-	-	-	-	-	16*32	0,99	
TOPLAM	1,00	1,00	1,02	0,95	0,96	0,96			
k=5	ÖRNEK	ALG-B1	ALG-P2	ALG-P4	ALG-P1	ALG-P3	ALG-P5	Boyut	Yoğunluk
	9	1,00	1,00	1,04	0,98	0,97	0,96	16*32	0,20
	10	1,00	0,99	1,03	0,96	0,97	0,97	16*32	0,20
	11	1,00	1,01	1,04	0,95	0,94	0,93	16*32	0,20
	12	1,00	1,00	1,03	0,88	0,89	0,88	16*32	0,30
	13	1,00	1,00	1,04	1,01	1,01	1,00	16*32	0,40
	14	1,00	1,00	1,03	0,89	0,90	0,89	16*32	0,50
	15	1,00	0,98	1,02	0,99	0,99	1,00	16*32	0,60
	16	1,00	1,01	1,03	1,01	1,01	1,03	16*32	0,70
	17	1,00	1,00	1,03	1,19	1,19	1,19	16*32	0,80
	18	1,00	0,99	1,03	0,90	0,91	0,91	16*32	0,90
	19	1,00	0,98	0,99	0,84	0,85	0,85	16*32	0,95
	20	-	-	-	-	-	-	16*32	0,96
	21	-	-	-	-	-	-	16*32	0,97
22	-	-	-	-	-	-	16*32	0,98	
23	-	-	-	-	-	-	16*32	0,99	
TOPLAM	1,00	0,99	1,02	0,95	0,95	0,95			

9. ÖZGEÇMİŞ

1992 yılında İzmir'in Konak ilçesinde doğan Elif Gülfidan Dayıoğlu ilköğretim ve lise eğitimini İzmir'de tamamlamıştır. Akademik eğitimini 2010 – 2014 yılları arasında Pamukkale Üniversitesi Bilgisayar Mühendisliği bölümünde tamamlamış, 2014'te Pamukkale Üniversitesi Bilgisayar Mühendisliği Anabilim dalında yüksek lisans eğitimine başlamıştır. 2014 yılından itibaren Mersin Üniversitesi'nde araştırma görevlisi olarak çalışmaya devam etmektedir.

Adı Soyadı : Elif Gülfidan DAYIOĞLU

Doğum Yeri ve Tarihi : Konak / İZMİR - 30.07.1992

Lisans Üniversite : Pamukkale Üniversitesi

Elektronik Posta : edayioğlu@mersin.edu.tr

İletişim Adresi : Akdeniz Mahallesi Alparslan Türkeş Bulvarı
Güzeldağ Sitesi G Blok No:303 Erdemli /
MERSİN