T.C.
ULUDAĞ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

DAĞITIM - TOPLAMALI ARAÇ ROTALAMA PROBLEMİNİN İKİ BOYUTLU YÜKLEME KISITI ALTINDA MODELLENMESİ VE ÇÖZÜMÜ

Figen KAS

Prof. Dr. Erdal EMEL
(Danışman)

YÜKSEK LİSANS TEZİ
ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI

BURSA, 2017

TEZ ONAYI

Figen KAS tarafından hazırlanan "Dağıtım - Toplamalı Araç Rotalama Probleminin İki Boyutlu Yükleme Kısıtı Altında Modellenmesi Ve Çözümü" adlı tez çalışması aşağıdaki jüri tarafından oy birliği/oy çokluğu ile Uludağ Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalı'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Danışman	: Prof. Dr. Erdal EMEL
Başkan	: Prof. Dr. Erdal EMEL
Üye	: Doç.Dr.Gürkan ÖZTÜRK
Üye	:Yrd.Doç.Dr.Mehmet AKANSEL

Yukarıdaki sonucu onaylarım

Prof. Dr. Ali BAYRAM

Enstitü Müdürü

19/4/2017

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

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

beyan ederim.

Figen KAS

ÖZET

Yüksek Lisans Tezi

DAĞITIM - TOPLAMALI ARAÇ ROTALAMA PROBLEMİNİN İKİ BOYUTLU YÜKLEME KISITI ALTINDA MODELLENMESİ VE ÇÖZÜMÜ

Figen KAS

Uludağ Üniversitesi
Fen Bilimleri Enstitüsü
Endüstri Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. Erdal EMEL

Lojistik yönetiminde önemli bir yere sahip olan araç rotalama problemi bir çok farklı kısıt altında incelenmektedir. Son yıllarda önem kazanan yükleme kısıtı ve rota boyunca dağıtım ve toplama isteklerinin aynı zamanda karşılandığı problem tipi, araç rotalama problemi litareteratüründe sıklıkla karşımıza çıkmaktadır. Yapısı gereği NP-Zor olan rotalama problemin bu eklentiler sonrası zorluk derecesi daha da artmaktadır. Bu çalışmanın amacı iki boyutlu yükleme kısıtı altında dağıtım toplamalı araç rotalama problemini matematiksel olarak modellemek ve kesin çözüm yaklaşımı geliştirmektir. Ayrıca rota boyunca yüklerin araç içinde yeniden yerleştirmeye maruz kalmadan taşınmasına yönelik olarak da, son giren ilk çıkar kısıtı (LIFO-Last In First Out) dikkate alınmıştır. Problemin modellenmesi için karışık tamsayılı matematiksel bir model önerilmiştir. Ancak LIFO kısıtının uygulanması, elde edilen tamsayı çözümlerden LIFO koşuluna uygun olanları kontrol eden ardıl bir kontrol algoritması üzerinden gerçekleştirilmiştir. Önerilen matematiksel model ve ardıl algoritma Mosel dilinde kodlanmış ve kesme düzlemi kullanan doğrusal programlama tabanlı dal sınır algoritması desteği ile çözülmüştür. Çözüm süresinin kısa olduğu küçük boyutlu problemler ile önerilen yaklaşımın optimal çözüm becerisi kanıtlanmıştır.

Anahtar Kelimeler: Dağıtım toplamalı araç rotalama problemi, iki boyutlu yükleme, LIFO, tamsayılı programlama.

2017, vii + 85 sayfa.i

ABSTRACT

MSc Thesis

MODELING AND SOLUTION OF PICKUP- DELIVERY VEHICLE ROUTING PROBLEM WITH TWO DIMENSIONAL LOADING CONSTRAINTS

Figen KAS

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

Supervisor: Prof. Dr. Erdal EMEL

In logistics research, the vehicle routing problem has been extensively studied for a variety of constraints. Among many others, container loading problem under sequential pickup and delivery requests along the route is a frequently encountered problem of logistics management. While the routing problem is NP-difficult by its structure, it becomes even more difficult to model these problems with these type of additional requirements. The purpose of this study is to develop a precise model and a solution approach for the vehicle routing problem with pickup-delivery under two dimensional loading constraints. It is also assumed that the loading must obey a last in-first-out rule (LIFO) which prohibits relocating loads within the vehicle. For the exact solution of the problem, a mixed integer mathematical model is proposed. However, for the complete implementation of the LIFO constraint, an efficient posterior control procedure is developed to check for the feasibility of candidate integer solutions. The proposed MILP model is solved by a solver package with an additional algorithmic programming capability in Mosel language. The solution approach with small size problems proved to be optimal.

Keywords: Integer programming, LIFO, pickup-delivery vehicle routing problem, two dimensional loading.

2017, vii + 85 pages.

TEŞEKKÜR

Yüksek Lisans eğitimim boyunca ve bu çalışmanın gerçekleştirilmesinde; değerli bilgi ve tecrübeleriyle beni yönlendiren, sabır ve hoşgörüyle çalışmamı bilimsel temeller ışığında yürütmemi sağlayan ve desteğini benden esirgemeyen değerli ve saygı değer hocam Prof. Dr. Erdal EMEL'e, bölüm içerisinde bana katkı sağlayan tüm hocalarıma, beni bu günlere getiren aileme ve Halil İbrahim'e sonsuz şükran ve teşekkürlerimi sunarım.

Figen KAS

İÇİNDEKİLER

Sayfa

ÖZET i
ABSTRACT ii
TEŞEKKÜR iii
KISALTMALAR DİZİNİ v
ŞEKİLLER DİZİNİ vi
ÇİZELGELER DİZİNİ vii
1. GİRİŞ 1
2. KAYNAK ÖZETLERİ 3
2.1. Araç Rotalama Problemi 3
2.2. Dağıtım Toplamalı Araç Rotalama Problemi 7
2.3. Yükleme Kısıtlı Araç Rotalama Problemi 11
2.4. Son Giren İlk Çıkar (LIFO-Last In First Out) Kısıtlı Araç Rotalama Problemi 18
2.5. İki ve Üç Boyutlu Yükleme Kısıtı Altında Dağıtım Toplamalı Araç Rotalama Problemi 22
3. MATERYAL VE YÖNTEM 25
3.1. Materyal 26
3.2. Problemin Tanımı 31
3.3. Matematiksel Model 33
3.4. Çözüm Yöntemi 36
4. BULGULAR 49
4.1. Uygulama 49
4.1.1. Uygulama örneği 49
4.1.2. Karşılaştırmalı sonuçlar 57
4.2. Bulgular 61
5. TARTIŞMA VE SONUÇ 62
KAYNAKLAR 64
EKLER 70
EK 1 Örnek Problemin FICO Xpress Programında Yazımı 71
EK 2 Örnek Problemin FICO Xpress Programında Çözümü 77
EK 3 Noktalara Ait Maliyet Verileri 81
ÖZGEÇMİŞ 85

KISALTMALAR DİZİNİ

Kısaltmalar	Açıklama
NP	Non-deterministic Polynomial-time (Deterministik olmayan polinom zaman)
VRP	Vehicle Routing Problem (Araç rotalama problemi)
TPS	Travelling Salesman Problem (Gezgin Satıcı Problemi)
PD	Pickup-delivery (Toplama-dağıtım)
CVRP	Capacitated Vehicle Routing Problem (Kapasite kısıtlı araç rotalama problemi)
VRPTW	Vehicle Routing Problem Time Windows (Zaman pencereli araç rotalama problemi)
VRPPD	Vehicle Routing Problem with Pickup and Delivery (Dağıtım toplamalı araç rotalama problemi)
VRPB	The Vehicle Routing Problem With Backhauls (Geri toplamalı araç rotalama problemi)
PPVRP	The Pallet Packing Vehicle Routing Problem (Palet yüklemeli araç rotalama problemleri)
LIFO	Last In First Out (Son giren ilk çıkar)
FIFO	First In First Out (İlk giren ilk çıkar)
2DL	Two Dimensional Loading (İki boyutlu yükleme)
3DL	Three Dimensional Loading (Üç boyutlu yükleme)
2L-CVRP	Two Dimensional Capacitated Vehicle Routing Problem (İki boyutlu kapasite kısıtlı araç rotalama problemi)
3L-CVRP	Three Dimensional Capacitated Vehicle Routing Problem (Üç boyutlu kapasite kısıtlı araç rotalama problemi)
2L-CVRPDP	Two Dimensional Capacitated Vehicle Routing Problem With Pickup Delivery (İki boyutlu kapasite kısıtlı dağıtım toplamalı araç rotalama problemi)
3L-VRPD	Three Dimensional Vehicle Routing Problem With Pickup Delivery (Üç boyutlu kapasite kısıtlı dağıtım toplamalı araç rotalama problemi)
TSPPD	The Travelling Salesman Problem With Pickup Delivery (Dağıtım toplamalı gezgin satıcı problemi)
GRASP	A Greedy Randomized Adaptive Search Procedure
ELS	Evolutionary Local Search (Evrimsel yerel arama)
MIP	Mixed Integer Programming (Karışık tam sayı programlama)
IP	Integer Programming (Tam sayı programlama)
BB	Branch and bound (Dal sınır)
BC	Branch and cut (Dal kesme)
CG	Column generation (Sütun Oluşturma)
DP	Dynamic Programming (Dinamik Programlama)

ŞEKİLLER DİZİNİ

Sayfa

Şekil 2.1. Araç rotalama problem türleri (Toth ve Vigo 2002) 4
Şekil 2.2. Önce dağıtım sonra toplamalı araç rotalama problemi 8
Şekil 2.3. Eş zamanlı dağıtım toplamalı araç rotalama problemi 8
Şekil 2.4. Karışık dağıtım ve toplamalı araç rotalama problemi 9
Şekil 3.1. Yükleme kısıtı altında rotalama problemine bir örnek (Packer 3d 2017) 25
Şekil 3.2. LIFO kısıtı ve öncelik ilişkisi gösterimi 38
Şekil 3.3. Öncelik ilişkileri ihlal edilmiş rota 39
Şekil 3.4. LIFO kısıtına uygun rota örneği 40
Şekil 3.5. LIFO kısıtına uygun yükleme planı 41
Şekil 3.6. LIFO kısıtının ihlal edildiği rota örneği... 42
Şekil 3.7. LIFO kısıtının ihlal edildiği yükleme planı 42
Şekil 3.8. Farklı toplama nokta sayıları için rota kombinasyonları 43
Şekil 3.9. Algoritma akış diyagramı 46
Şekil 3.10. Kontrol algoritması sözde kodu... 48
Şekil 4.1. Örnek problem rota ve yükleme planı 55
Şekil 4.2. Örnek problem % fark grafiği.. 56
Şekil 4.3. Örnek problem amaç fonksiyonu değişimi.. 56
Şekil 4.4. Örnek problem beş toplama noktası % fark grafiği 57
Şekil 4.5. Örnek problem beş toplama noktası amaç fonksiyonu değişimi 58
Şekil 4.6. Örnek problem altı toplama noktası % fark grafiği 58
Şekil 4.7. Örnek problem altı toplama noktası amaç fonksiyonu değişimi 59
Şekil 4.8. Örnek problem sekiz toplama noktası % fark grafiği 59
Şekil 4.9. Örnek problem sekiz toplama noktası amaç fonksiyonu değişimi 60
Şekil 4.10. Örnek problem dokuz toplama noktası % fark grafiği 60
Şekil 4.11. Örnek problem dokuz toplama noktası amaç fonksiyonu değişimi 61

ÇİZELGELER DİZİNİ

Sayfa

Çizelge 2.1. Literatürde dağıtım toplamalı araç rotalama problemine ilişkin yapılan çalışmalar 10
Çizelge 2.2. Literatürde yükleme kısıtını ele alan çalışmalar 13
Çizelge 2.3. Literatürde yükleme kısıtını ele alan çalışmalar (devam) 14
Çizelge 2.4. Literatürde yükleme kısıtını ele alan çalışmalar (devam) 15
Çizelge 2.5. Literatürde LIFO kısıtını ele alan çalışmalar. 19
Çizelge 2.6. Literatürde iki ve üç boyutlu yerleştirme kısıtını ele alan çalışmalar 24
Çizelge 4.1. Noktalara ait talep miktarları (kg) 50
Çizelge 4.2. Örnek problem için araç verileri 50
Çizelge 4.3. Örnek problem için maliyet verileri 50
Çizelge 4.4. Farklı toplama noktaları için problem sonuçları 57
Çizelge EK 3.1. Beş toplama noktası maliyet verileri 81
Çizelge EK 3.2. Beş toplama noktası talep verileri 81
Çizelge EK 3.3. Altı toplama noktası maliyet verileri 82
Çizelge EK 3.4. Altı toplama noktası talep verileri 82
Çizelge EK 3.5. Sekiz toplama noktası maliyet verileri 83
Çizelge EK 3.6. Sekiz toplama noktası talep verileri 83
Çizelge EK 3.7. Dokuz toplama noktası maliyet verileri 84
Çizelge EK 3.8. Dokuz toplama noktası talep verileri 84

1. GİRİŞ

Lojistik faaliyetlerinin yönetimi, günümüzde giderek artan tedarik zinciri ilişkileri içinde taleplerin karşılanmasında önemli rol oynamaktadır. Taleplerin vaktinde karşılanmasının yanı sıra en düşük maliyetle gerçekleştirilmesi lojistik yönetiminin temel amacıdır. Bu noktada, artan rekabet koşulları lojistik şirketlerini kârlılık arayışında yeni çözümlere zorlamakta ve mühendislik alanlarında farklı uygulamalarla etkin çözümler üretilmesine yol açılmaktadır. Geçmişi 1950'li yıllara dayanan araç rotalama problemleri, lojistik yönetiminde en uygun maliyet veren taşıma yönteminin belirlenmesinde yoğun olarak araştırılan bir optimizasyon alanı olagelmiştir. Problem sektör ve ürün bazında çeşitlilik göstermekte ve günümüzde farklı birçok tür ve kısıt altında incelenmektedir. Bilimsel alanda kullanılan çeşitli optimizasyon yöntemleri ve bilgisayar teknolojilerinin sağladığı destek sayesinde gelişen yazılım programları bu problemin çözümünde kuruluşlara önemli katkı sağlamaktadır.

Araç rotalama problemleri genellikle merkezi bir depodan başlayan rota üzerinde yer alan müşterilerin taleplerinin karşılanması problemi olarak tanımlanır. Ancak dinamik müşteri talepleri ile birlikte problemde yapısal değişiklikler meydana gelebilmektedir. Firmaların tedarikçileri ile birlikte yürüttükleri tam zamanında üretim süreçlerinde yarımamul tedariği sırasında boş kasa ya da hatalı ürünleri geri çevirmeleri buna örnek gösterilebilir. Benzer bir alanda, örneğin kargo şirketlerinin adrese ulaştırma ve adresten gönderi alma hizmetleri veya gıda sektöründe depozitolu ürün ya da son kullanma tarihi geçmiş ürünlerin toplanması (örneğin; günlük süt üreten firmaların dolu süt şişelerini marketlere ulaştırırken boş şişeleri toplamaları) ya da geri dönüşüm çalışmalarının yapıldığı sektörler, aynı rota boyunca dağıtım ve toplama işlemlerinin birlikte yapıldığı yeni bir yapı ortaya çıkarmıştır. Bu yeni problem tipi, literatürde 'dağıtım-toplamalı araç rotalama problemi' adı altında incelenmektedir. Tüm bu işlemler esnasında ürünlerin sağlam ve güvenli bir şekilde teslim edilmesi müşteri memnuniyeti açısından oldukça önemlidir. Yüklerin uygun rotada müşterilere ulaştırılması sırasında ürünlerin araç içerisine uygun biçimde yüklenerek araç hacminin en iyi şekilde kullanılması yükleme probleminin bir gerekliliği olarak ortaya çıkmaktadır. Uygulamada yükleme ve dağıtım-toplamalı rotalama tipindeki bu iki problem sıklıkla birlikte ele alınmayı gerektirir. Söz konusu problem yapısal olarak zor ve karmaşıktır. Literatürde bu tümleşik probleme yönelik az sayıda çalışma bulunmaktadır. Yükleme kısıtı altında araç rotalama probleminin tarihsel gelişimi ve bugünkü süreçte mevcut durumu Pollaris ve ark. (2015) tarafından geniş bir şekilde ele alınmıştır. Ayrıca Iori ve Martello (2010) yine yükleme kısıtı altında rotalama problemlerini etkin bir şekilde incelemiştir. Tüm bu çalışmalar ve literatüre bakıldığında yükleme kısıtı altında dağıtım-toplamalı araç rotalama problemine matematiksel model odaklı kesin çözüm yaklaşımı sunulan bir çalışmaya ise rastlanmamıştır. Literatürde bulunan çalışmalar probleme sıklıkla sezgisel yöntemler ile çözüm getirebilmiştir.

Bu çalışmanın amacı; bir taşıma aracı ile rota boyunca toplanacak ve dağıtılacak yüklerin taşıma hacmine uygun bir biçimde yüklenmesi ve boşaltılması esnasında maliyet en küçüklemesini sağlayacak en uygun rotanın bulunmasıdır. Bu amacın gerçekleştirilmesi için geliştirilecek kesin çözüm yaklaşımı ile gerçek hayatta sık karşılaşılan bu probleme matematiksel modelleme ve kesin çözüm tekniği açısından katkı sağlanması hedeflenmektedir.

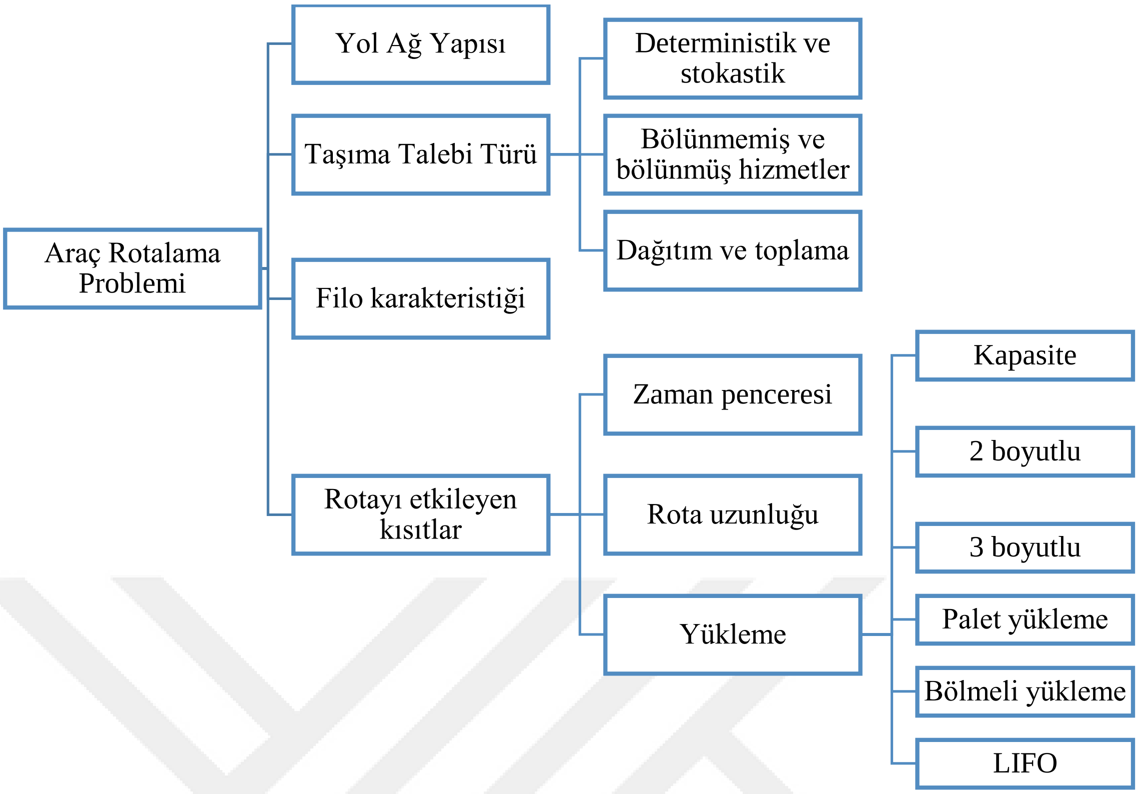
2. KAYNAK ÖZETLERİ

2.1. Araç Rotalama Problemi

Lojistik alanındaki literatürde oldukça geniş yer alan araç rotalama problemi, en basit tanımıyla; bir merkezde bulunan araçların, talepleri bilinen müşteri kümesine hizmet sağlayıp, tekrar merkeze dönmesini sağlayacak en kısa rotaların bulunması problemidir (Gencer ve Yaşa 2013). Problem ilk olarak Dantzig ve Ramser (1959) tarafından çalışılmıştır. Çalışmada benzin istasyonlarına benzin teslimatı ile ilgili bir uygulama ele alınmış ve matematiksel programlama modeli üzerinde çalışılmıştır. Değişen koşullar altında araç rotalama problemi bir çok farklı yapı ve özellikte ele alınmıştır. Günümüze değin problemin çözüm yaklaşımı üzerine yapılan çalışmalar farklı matematiksel ve sezgisel çözüm yöntemlerinin gelişmesini sağlamıştır. Klasik araç rotalama problemi varsayımları şöyle sıralanabilir:

- Her araç rotaya tek bir noktadan başlar ve rota aynı noktada sona erer,
- Müşteri talepleri karşılanmalıdır,
- Her müşteri yalnızca bir kez ziyaret edilebilir,
- Modele ait diğer tüm kısıtlar sağlanmalıdır.

Önceden bilinen yol ağı bir graf üzerinde düğüm noktaları ve ayrıtlar yardımıyla tanımlanmaktadır. Müşteri ve merkez depo düğüm noktaları, düğüm noktalarının birbirine bağlandığı yollar ise ayrıtlar yardımıyla gösterilmektedir. i ve j noktaları göstermek üzere, ayrıtlar (i, j) şeklinde ifade edilmektedir. Bir nokta çifti arasındaki (i, j) ayrıtının belirli bir c_{ij} maliyeti vardır. Bu maliyete bağlı olarak problem çözümünde araç sabit maliyeti ve rota maliyetini içeren maliyet en küçüklemesi, toplam rota zamanının en küçüklenmesi, tüm müşterilere hizmet vermek için gerekli olan araç sayısının en küçüklenmesi vb. şeklinde farklı amaç fonksiyonları kullanılabilir. Araç rotalama probleminin birçok alanda kullanılması farklı kısıtlar altında incelenmesine ve yeni türlerinin oluşmasına olanak sağlamıştır. Araç rotalama probleminin temel türleri Şekil 2.1`de görülmektedir.



Şekil 2.1. Araç rotalama problem türleri (Toth ve Vigo 2002)

1. Yol ağ yapısı:

Yol ağ yapısına göre araç rotalama problemi, i ve j noktaları arasındaki mesafelerin gidiş-geliş yönüne bağlı olarak eşit olup olmamasına göre simetrik ve asimetrik araç rotalama problemi olarak ikiye ayrılmaktadır. Simetrik araç rotalama probleminde; i ve j noktaları arasındaki mesafeler i'den j'ye ve j'den i'ye gidişte birbirine eşittir. Yani maliyet matrisinde iki nokta arasında gidiş ve geliş maliyetleri (mesafesi) eşit olduğu bu durumda matris simetriktir. Asimetrik araç rotalama probleminde ise i noktasından j noktasına gidiş mesafesi ile j noktasında i'ye olan uzaklığın farklı olduğu problem türüdür. Bu durumda maliyet (mesafe) matrisi simetrik değildir (Düzakın ve Demircioğlu 2009).

2. Taşıma talebi türü:

- Deterministik ve stokastik araç rotalama: stokastik araç rotalama problemi, klasik araç rotalama probleminin, problem elemanlarından bir ya da birkaçının rassal olduğu durumlarda karşılaşılan bir problem çeşididir (Koç 2012). Müşteri, talep ya da servis süresi problem içinde stokastik olarak tanımlanabilir. Dinamik araç rotalama probleminde ise probleme ait öğelerin rota boyunca belirli olduğu problem türdür.

- Bölünmemiş ve bölünmüş hizmetler: bölümlenmemiş hizmet verilen problemlerde, tek bir araç, tek bir servis operasyonunu gerçekleştirmektedir. Fakat araç kapasitesinin aşıldığı ya da bölünmüş küçük hizmetlerin maliyet açısından kazanç sağladığı bazı durumlarda hizmet bölünebilir (Toth ve Vigo 2002).
- Dağıtma ve toplama: Dağıtım işlemi, merkezi bir depodan yüklenen ürünlerin ilgili müşterilere dağıtılmasını; toplama işlemi ise atık vb. maddelerin müşterilerden rota boyunca toplanarak merkez bir depoya götürülmesi şeklinde rotalama problemlerinde yer almaktadır. Bunun yanısıra bu iki işlemin rota boyunca birlikte yapıldığı durumlar söz konusudur. Dağıtım-toplamalı araç rotalama problemlerine ilişkin detaylı araştırma Bölüm 2.2.'de verilmiştir.

3. Rotayı etkileyen kısıtlar:

Araç rotalama probleminin uygulamalarında rotanın belirlenmesinde bazı önemli kısıtlar probleme dahil edilmektedir. Bu kısıtlar gerçek hayat problemlerine etkili çözüm üretmek için oldukça önemlidir. Bu kısıtlar aşağıda verilmiştir.

- Zaman penceresi kısıtı: Çoğu rotalama probleminde bulunan seyahat, servis, bekleme gibi zaman penceresi içerisinde bulunan bazı kısıtlar çizelgeleme ile ilişkili kısıtlardır (Toth ve Vigo 2002). Bu kısıt altında ele alınan problemlerde her müşterinin hizmet alması gereken belirli zaman aralıkları vardır. Aracın müşterileri ziyaret ettiği süre boyunca, bu zaman aralıklarına uygun olarak hizmet verilmeli ve bu duruma uygun bir rota oluşturulmalıdır.
- Rota uzunluğu kısıtı: Bu kısıt en büyük rota uzunluğunu baştan tayin ederek gezilecek noktaları kısıtlamaktadır. Belirlenen rota uzunluğu aşıldığında yeni bir rota oluşturulmalıdır.
- Yükleme kısıtı: Yükleme kısıtı, aracın ağırlık ve hacim kapasitesi olarak ikiye ayrılmaktadır. Ağırlık kısıtı, rota boyunca araçta bulunan yüklerin toplam ağırlık kapasitesini aşmasını engellemektedir. Hacim kapasitesi ise araçta bulunan toplam yük hacminin araç alanı ya da hacminden daha küçük olmasını

sağlamaktadır. Bu tür problemler çok boyutlu yükleme problemleri olarak isimlendirilmektedir (Toth ve Vigo 2002). Yükleme problemlerinde araç kapasitesinin alan olarak göz önünde bulundurulduğu problemler iki boyutlu yükleme problemleri; hacim olarak göz önünde bulundurulduğu problemler ise üç boyutlu yükleme problemleri olarak adlandırılmaktadır. Benzer şekilde araç kapasitesinin önceden belirli alanlara ayrıldığı durumlar bölmeli/yığın yükleme problemleri olarak isimlendirilmektedir. İki boyutlu probleme benzer biçimde, dikdörtgen palet alanına dikdörtgen kutuların yüklenmesini ele alan problemler palet yükleme problemi olarak karşımıza çıkmaktadır. Bahsedilen yükleme kısıtlarına ek olarak; LIFO yükeleme kısıtı en son ziyaret edilen noktanın ürününün ilk önce indirilmesini ifade etmektedir. Kısıtın amacı, indirme sırasında araç içerisinde meydana gelebilecek gereksiz taşımaların ve bu sebepten kaynaklanacak beklemelerin azaltılmasını sağlamaktır. Bu tez çalışması iki ve üç boyutlu yükleme ve LIFO kısıtı altında araç rotalama problemi için literatür araştırması yapılmıştır. Bu doğrultuda Bölüm 2.3. ve Bölüm 2.4. incelenebilir.

4. Filo karakteristiği:

Rotalama probleminde benzer özelliklere sahip araçların aynı depoda bulundukları varsayılmaktadır. Ancak bazı durumlarda araçların farklı depolarda bulunmaları ya da hız, maliyet, kapasite vb. farklı özelliklere sahip olmaları söz konusudur. Ayrıca kamyon ve tırların otonom ya da otonom olmayan çeşitlerinin olması probleme farklı bir yaklaşım getirmektedir. Filo karakteristiği açısından araç rotalama problemi üç grupta toplanabilir:

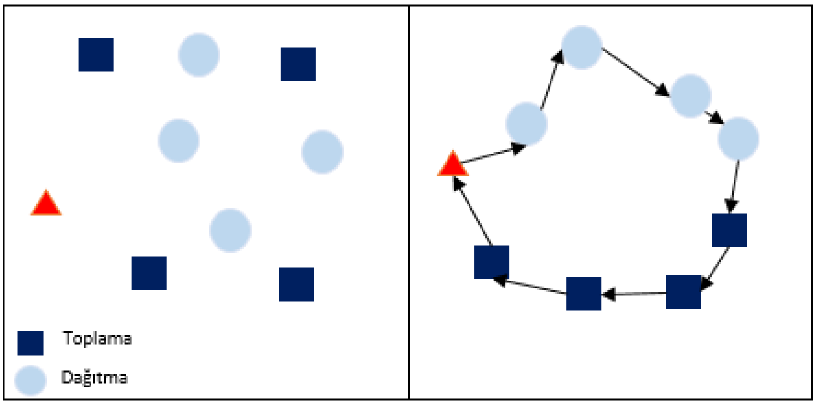
- Çok depolu araç rotalama problemi; araçlar aynı özelliklere sahiptir fakat rota başlangıç ve bitiş depoları farklıdır (Toth ve Vigo 2002).
- Heterojen ya da karışık filo araç rotalama problemi; araç rotalama probleminde dağıtım yapan araçların belirli bir kapasitesinin olması durumudur. Karma kapasiteli araç rotalama probleminde her bir aracın birbirinden farklı bir kapasitesi olabilir (Düzakın ve Demircioğlu 2009).
- Kamyon ve tır rotalama problemi; yalnızca kamyonların rotalandığı problemler ile kamyon ve tırların rotalandığı problemler olarak karşımıza çıkmaktadır(Toth ve Vigo 2002).

Bu tez çalışmasında çok boyutlu yükleme kısıtı altında dağıtım toplamalı araç rotalama problemi üzerinde durulacaktır. Bu nedenle; dağıtım toplamalı araç rotalama problemi, yükleme problemi ve LIFO kısıtına yönelik yapılan literatür araştırması çalışmanın devamında sunulmuştur.

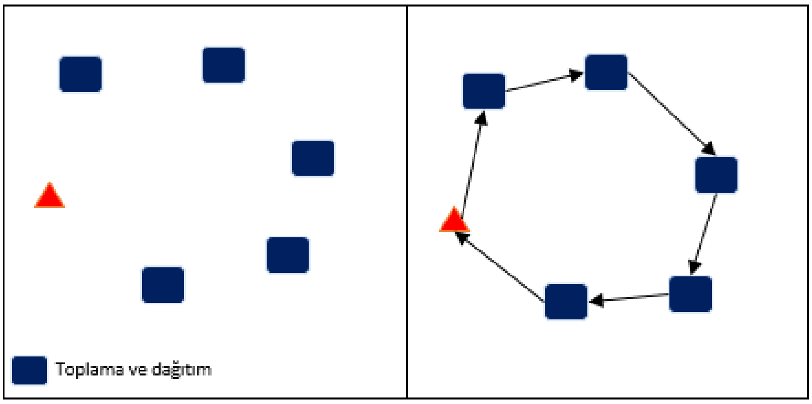
2.2. Dağıtım Toplamalı Araç Rotalama Problemi

Lojistik faaliyetlerinin gerçekleştirilmesinde araçların müşterilere ürün taşımasının yanı sıra bazı müşteri noktalarından ürün toplamaları da gerekmektedir. Normalde müşteri depolarında bulunan ürünlerin doğrudan başka müşterilere taşınmadığı varsayılmaktadır. Diğer bir deyişle, tüm ürünler bir merkez depodan dağıtılır ya da bir merkez depoda toplanır (Nagy ve Salhi 2005). Fakat uygulamada bu durum farklılık göstermektedir. Dağıtım-toplamalı araç rotalama probleminde, araçların çıkış noktası depodur. Rotalama probleminde olduğu gibi araçlar, rota boyunca dağıtım-toplama şeklindeki tüm müşteri taleplerini karşılar ve en son depoya dönerler. Her müşteriye yalnızca bir araç hizmet verebilir. Rota boyunca dağıtılan ve toplanan yük miktarı toplam araç kapasitesini geçemez. Rota boyunca maliyet minimizasyonu amaçlanır (Côté ve ark. 2012b). Dağıtım toplamalı araç rotalama problemleri üç başlık altında incelenebilir:

Önce dağıtım sonra toplamalı araç rotalama problemi (VRP with delivery-first, pickup-second): Bu alanda yapılan birçok çalışma; müşterilerin mal alan ve mal gönderen şeklinde ayrıldığını, buna ek olarak araçların yalnızca tüm yüklerini dağıttıktan sonra yükleme yapabileceklerini varsaymaktadır (Şekil 2.1). Bunun bir nedeni araç içinde dağıtılacak ve toplanacak malların yeniden yerleştirilmelerinin zor olmasıdır. Bu varsayım uygulamada kolaylık sağlamaktadır (Côté ve ark. 2012b). Bu problem ilk olarak Deif ve Bodin (1984) tarafından çalışılmıştır. Bahsedilen problem tipi bu tez çalışması kapsamında kullanılmamaktadır.



Şekil 2.2. Önce dağıtım sonra toplamalı araç rotalama problemi

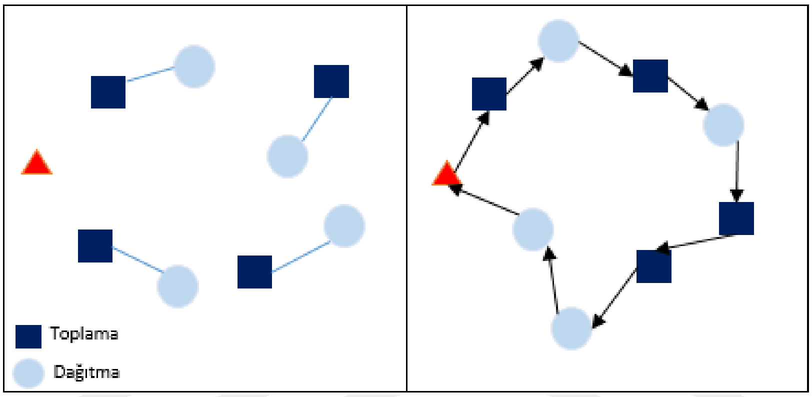


Şekil 2.3. Eş zamanlı dağıtım toplamalı araç rotalama problemi

Eşzamanlı dağıtım toplamalı araç rotalama problemi (VRP with simultaneous pickup and delivery): Bu problem tipinde müşteriler eşzamanlı olarak mal alıp gönderebilir. Eşzamanlı olarak mal gönderme işlemi, rota boyunca her müşteriye uğrandığında dağıtılacak olan malın bırakılıp, alınacak olanın toplanmasıdır (Şekil 2.3). Her müşteriye yalnızca bir kez gidilebilir. Problemi ilk defa ele alan Min (1989) olmuştur. Min, bir depo, iki araç ve 22 müşteriden oluşan problem için müşterileri sınıflandırmış ve her grup için gezgin satıcı problemini çalışmıştır. Olursuz ayrıtlarlar cezalandırılmış ve gezgin satıcı problemi tekrar çözülmüştür. Halse (1992) de problemin araç rotalama versiyonu için çalışmalarda bulunmuştur (Nagy ve Salhi 2005).

Karışık dağıtım ve toplamalı araç rotalama problemi (VRP with mixed pickup and delivery): Araçlar tüm müşterileri, rota boyunca ziyaret edebilir. Literatüre bakıldığında

(Çizelge 2.1) bu problem tipinin ele alındığı çalışmaların sayısı azdır. Bu yaklaşım ilk olarak Golden ve ark. (1985) tarafından ele alınmıştır. Casco ve ark. (1988) toplama noktaları için dağıtım rotasında henüz dağıtılmış yüklerin yerleştirme maliyetini dikkate alan bir yük tabanlı yerleştirme prosedürü geliştirmiştir. Mosheiov (1994) dağıtım toplamalı gezgin satıcı problemi konusunda araştırma yapmıştır. Anily ve Mosheiov (1994) , minimum kapsama ağacı algoritması üreterek dağıtım toplamalı gezgin satıcı problemi için bir çözüm metodu geliştirmiştir. Salhi ve Nagy (1999) ise Casco ve ark. (1988)'nın yerleştirme metodunu genişletmiştir (Nagy ve Salhi 2005).



Şekil 2.4. Karışık dağıtım ve toplamalı araç rotalama problemi

Ruland ve Rodin (1997), gezgin satıcı problemini dağıtım toplamalı gezgin satıcı problemi olarak değiştirerek dal-kesme algoritmasını kullanmış ve probleme yeni bir çözüm yaklaşımı geliştirmiştir. Gezgin satıcı probleminin Dantzig-Fulkerson-Johnson modelinde olduğu gibi, dağıtım toplamalı problem için de tam sayı programlama modeli oluşturulmuştur. Dağıtım toplamalı rotalama probleminin gezgin satıcı problemi ile birlikte ele alındığı diğer bir çalışma Gendreau ve ark. (1999)'a aittir. Dolu şişelerin bırakılması ve boş süt şişelerinin toplanmasını ele alan gezgin satıcı probleminin çözümünde yasaklı arama algoritması kullanılmıştır. Probleme yeni bir yaklaşım Côté ve ark. (2012b) tarafından getirilmiştir. Yatay düzlemde birden çok bölüme ayrılmış (stack) araç rota boyunca dağıtım ve toplama işlemlerini gerçekleştirmektedir.

Çizelge 2.1. Literatürde dağıtım toplamalı araç rotalama problemine ilişkin yapılan çalışmalar

Çalışma	**Yazar-Yıl**	**Problem tipi**						**Model / Çözüm**		
		TSP	**VRP**	**PD**	**2DL**	**3DL**	**LIFO**	**MIP/IP**	**BB/BC/CG**	**Sezgisel**
The vehicle routing problem with backhauling: Two approaches	(Golden ve ark. 1985)		X	X						X
The traveling salesman problem with delivery and backhauls	(Anily ve Mosheiov 1994)	X		X						X
The pickup and delivery problem: Faces and branch-and-cut algorithm	(Ruland ve Rodin 1997)	X		X				X	X	
Heuristics for the traveling salesman problem with pickup and delivery	(Gendreau ve ark. 1999)	X		X						X
A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling	(Salhi ve Nagy 1999)		X	X						X
Scheduling transportation of live animals to avoid the spread of diseases	(Sigurd ve ark. 2004)		X	X					CG DP	
Models and branch-and-cut algorithms for pickup and delivery problems with time windows.	(Ropke ve ark. 2007)		X	X				X	X	
The double travelling salesman problem with multiple stacks formulation and heuristic solution approaches	(Petersen ve Madsen 2009)	X		X			X	X		X
Large neighborhood search for the pickup and delivery traveling salesman problem with multiplestacks	(Côté ve ark. 2012b)	X		X			X	X		X
A novel hybrid method on VRP with pickup and delivery	(Ning ve ark. 2016)		X	X				X		X
Adaptive large neighborhood search for the pickup and delivery problem with time windows, profits, and reserved requests	(Li ve ark. 2016)		X	X				X		X

Bu sırada yükleme ve boşaltma işlemleri ilk giren-son çıkar (LIFO, last in-first out) prensibine uygun yapılmaktadır. Rotalama probleminin NP-zor yapıda olması aynı zamanda probleme dağıtım-toplama, yükleme ve LIFO gibi ek kısıtların getirilmesi nedenleriyle çözüm aşamasında komşuluk arama sezgisel yöntemi kullanılmıştır. LIFO kısıtı araç rotalama problemlerinde önemli bir yere sahiptir. Bu konudaki çalışmalar Bölüm 2.4.'te sunulmuştur. Benzer kısıtlar altındaki probleme Petersen ve Madsen (2009) yine komşuluk arama sezgiseli ile çözüm getirmiştir. Aynı çalışmada dağıtım toplamalı araç rotalama probleminin farklı bir versiyonu olan zaman penceresi kısıtı ele alınmıştır. Sigurd ve ark. (2004), Ropke ve ark. (2007) ile Li ve ark. (2016) tarafından yapılan çalışmalarda problem, zaman kısıtı altında incelenmiştir. Dağıtım toplamalı araç rotalama problemine Ning ve ark. (2016) tarafından genetik algoritma kullanılarak yeni bir çözüm yaklaşımı geliştirilmiştir. Literatür araştırmalarında, bu probleme kesin çözüm yaklaşımı geliştirildiğine dair bir bilgi edinilememiş olup, problemin çözümü için sezgisel yöntemlerin kullanıldığı görülmüştür. Literatürdeki bu eksiği dikkate alarak yapılan bu tez çalışmasında, ele alınan araç rotalama problemi, kesin çözüm yaklaşımı geliştirmek amacıyla yükleme kısıtı altında dağıtım-toplamalı araç rotalama problemi olarak seçilmiştir.

Yükleme kısıtlı araç rotalama probleminin gelişimi ile ilgili bilgiler Bölüm 2.3'de verilmiştir.

2.3. Yükleme Kısıtlı Araç Rotalama Problemi

Rotalama problemlerinde, araçta bulunan yükler için hacimlerinden bağımsız genel ifadeler kullanılmaktadır. Araç kapasitesi sadece ağırlık kısıtı ile tanımlanmaktadır. Fakat taşımacılık faaliyetlerinde ürünlerin araç içerisinde serbestçe hareket etmesi çeşitli sorunlara yol açmaktadır. Bu nedenle yükleme problemi bu sorunların ortadan kaldırılması için oldukça önemlidir. Yükleme problemi, bir dizi yükün bir veya birden çok konteynere en uygun biçimde yüklenmesi şeklinde tanımlanabilir. Araç yükleme veya depolama işlemlerinde, yüksek kapasite kullanımı elde edebilmek için yükleme problemi oldukça faydalıdır. Uygulamada problemin çözümüne ilişkin çeşitli yazılımlar kullanılmaktadır (örneğin; http://www.packer3d.com/). Problem, yapısı itibari ile "Kesme Problemi (Cutting Stock Problem)"ne çok yakındır. Paketleme probleminde üç boyutlu işlemler ele alınırken kesme probleminde bir veya iki boyutlu işlemler ele alınmaktadır. Her iki problemde de, belirli bir dış boyut içine daha küçük boyutlu şekilleri yerleştirme söz konusudur (Türkay 2002). Yükleme problemleri alt kısıtlara ayrılarak geniş bir biçimde sınıflandırılabilir. Yükleme problemi kısıtları:

- Çok boyutlu yükleme problemleri,

- Kargo ile ilgili kısıtlar; tam sevkiyat kısıtları, tahsis kısıtları, pozisyon kısıtları,
- Konteyner ile ilgili kısıtlar; ağırlık kısıtı, ağırlık dağılımı kısıtları,
- Ürün ile ilgili kısıtlar; yükleme öncelikleri, ortogonalite (diklik) kısıtı, oryantasyon (yönelim) kısıtları, istifleme kısıtları,
- Yük ile ilgili kısıtlar; kararlılık kısıtları (dikey-yatay) şeklinde tanımlanabilir (Pollaris ve ark. 2015).

Yükleme problemi karşımıza Chen ve ark. (1991)'nın iki boyutlu (two-dimensional loading-2DL) palet yükleme problemi ile çıkmaktadır. Çalışmada kutuların, sayısı önceden bilinen ve standart uzunluklara sahip paletlerin üzerine toplam palet sayısını en küçükleyecek ve araç boy-uzunluk kapasitesini aşmayacak şekilde yüklenmesini sağlayan matematiksel model önerilmiştir. Yine Chen ve ark. (1995) tarafından kullanılmayan alanı en küçükleyecek üç boyutlu (three-dimensional loading-3DL) genel bir konteyner yükleme modeli önerilmiştir. Yükleme problemi NP-zor yapıda olduğundan dolayı problemin çözümünde sezgisel algoritmalar kullanılmaktadır. Bortfeldt ve Wäscher (2013) yükleme problemi için önerilen çözüm yaklaşımlarını kapsayan bir çalışma ortaya koymuştur.

Yükleme ve rotalama problemlerinin birleşmesi bu tez çalışmasında ele alınan tümleşik problemin ikinci kısmını oluşturmaktadır. Her iki problem de NP-zor yapıda olan optimizasyon problemleridir. Bu problemlerin birleştirilmesi oldukça zor olmasına rağmen, elde edilebilirse birleştirilmiş problemin çözümü tercih edilecektir (Pollaris ve ark. 2015). Yükleme kısıtlı araç rotalama problemleri temel olarak şu kategorilerde tanımlanabilir: İki boyutlu kapasite kısıtlı araç rotalama problemi (2L-CVRP), üç boyutlu kapasite kısıtlı araç rotalama problemi (3L-CVRP), çok bölmeli araç rotalama problemleri, palet yüklemeli araç rotalama problemleri (PPVRP), LIFO/FIFO (first in-first out / ilk giren-ilk çıkar) kısıtı altında dağıtım toplamalı gezgin satıcı problemi (TSPPD) ve yükleme kısıtı altında dağıtım toplamalı araç rotalama problemi (Iori ve Martello 2010). İki ve üç boyutlu araç rotalama problemleri Çizelge 2.2`de görüldüğü gibi literatürde geniş yer tutmaktadır.

Çizelge 2.2. Literatürde yükleme kısıtını ele alan çalışmalar

Çalışma	Yazar-Yıl	Problem tipi						Model / Çözüm		
		TSP	**VRP**	**PD**	**2DL**	**3DL**	**LIFO**	**MIP/IP**	**BB/BC/CG**	**Sezgisel**
Vehicle routing problem with packing constraints	(Türkay ve Emel 2003)		X			X		X		
A tabu search algorithm for a routing and container loading problem	(Gendreau ve ark. 2006)		X			X	X			X
Logistics optimization: Vehicle routing with loading constraints	(Aprile ve ark. 2007)		X			X				X
An exact approach for the vehicle routing problem with two-dimensional loading constraints	(Iori ve ark. 2007)		X		X			X	X	
A tabu search heuristic for the vehicle routing problem with two-dimensional loading constraints	(Gendreau ve ark. 2008)		X		X					X
Ant colony optimization for the two-dimensional loading vehicle routing problem	(Fuellerer ve ark. 2009)		X		X					X
An integrated approach to the vehicle routing and container loading problems	(Moura ve Oliveira 2009)		X			X	X	X		X
Metaheuristics for vehicle routing problems with three-dimensional loading constraints	(Fuellerer ve ark. 2010)		X			X	X			X
Simulated annealing for the vehicle routing problem with two-dimensional loading constraints	(Leung ve ark. 2010)		X		X					X
Extended guided tabu search and a new packing algorithm for the two-dimensional loading vehicle routing problem	(Leung ve ark. 2011)		X		X					X
A relaxation method for the three-dimensional loading capacitated vehicle routing problem	(Ren ve ark. 2011)		X			X	X			X
A multi-start evolutionary local search for the two-dimensional loading capacitated vehicle routing problem	(Duhamel ve ark. 2011)		X		X					X

Çizelge 2.3. Literatürde yükleme kısıtını ele alan çalışmalar (devam)

Çalışma	Yazar-Yıl	Problem tipi						Model / Çözüm		
		TSP	VRP	PD	2DL	3DL	LIFO	MIP/IP	BB/BC/CG	Sezgisel
Pheromone-based heuristic column generation for vehicle routing problems with black box feasibility	(Massen ve ark. 2012)		X			X	X			X
A hybrid algorithm for the capacitated vehicle routing problem with three-dimensional loading constraints	Bortfeldt 2012)		X			X	X			X
A tabu search algorithm for the capacitated vehicle routing problem with three-dimensional loading constraints	(Wisniewski ve ark. 2012)		X			X	X			X
A two-stage tabu search algorithm with enhanced packing heuristics for the 3L-CVRP and M3L-CVRP	(Zhu ve ark. 2012)		X			X	X			X
A hybrid genetic algorithm for the vehicle routing problem with three-dimensional loading constraints	(Miao ve ark. 2012)		X			X	X			X
A GRASP- ELS for the vehicle routing problem with three-dimensional loading constraints	(Lacomme ve ark. 2013)		X			X				X
Artificial bee colony algorithm for two-dimensional loading capacitated vehicle routing problem	(Bin ve ark. 2013)		X		X					X
A hybrid approach for the vehicle routing problem with three-dimensional loading constraints	(Ruan ve ark. 2013)		X			X	X	X		X
Packing first, routing second a heuristic for the vehicle routing and loading problem	(Bortfeldt ve Homberger 2013)		X			X	X			X
Local search techniques for a routing-packing problem	(Ceschia ve ark. 2013)		X			X	X			X
An optimization model for the vehicle routing problem with practical three-dimensional loading constraints	(Junqueira ve ark. 2013)		X			X		X		
An exact algorithm for the two-dimensional orthogonal packing problem with unloading constraints	(Côté ve ark. 2014)		X		X				X	X
A biased-randomized algorithm for the two-dimensional vehicle routing problem with and without item rotations	(Dominguez ve ark. 2014)		X		X					X

Çizelge 2.4. Literatürde yükleme kısıtını ele alan çalışmalar (devam)

Çalışma	Yazar-Yıl	Problem tipi						Model / Çözüm		
		TSP	VRP	PD	2DL	3DL	LIFO	MIP/IP	BB/BC/CG	Sezgisel
An effective tabu search approach with improved loading algorithms for the 3L-CVRP	(Tao ve Wang 2015)		X			X	X			X
An evolutionary local search for the capacitated vehicle routing problem minimizing fuel conssumption under three dimensional loading constraints	(Zhang ve ark. 2015)		X			X	X			X
Heuristic algorithms for a three-dimensional loading capacitated vehicle routing problem in a carrier	(Junqueira ve Morabito 2015)		X			X				X
Avariable neighborhood search for the capacitated vehicle routing problem with two-dimensional loading constraints	(Wei ve ark. 2015)		X			X				X
A branch-and-cut approach for the vehicle routing problem with loading constraints	(Hokama ve ark. 2016)		X		X	X			X	X

2L-CVRP tipindeki problemlerde yüklerin ve aracın yükleme alanı yalnızca en ve boy uzunluğu olarak kabul edilmektedir. Bu problem gerçek hayatta yüklerin ağırlık, kırılganlık ya da büyük boyutlar nedeniyle birbirleri üstüne yüklenemediği durumlarda ortaya çıkmaktadır (Strodl ve ark. 2010). 2L-CVRP için sezgisel örnek Iori ve ark. (2007) tarafından geliştirilmiştir. Çalışmada 35 müşteriye kadar optimal problem çözümü veren dal-kesme algoritması kullanılmıştır. Dal-kesme algoritması kullanılan bir diğer çalışmayı Côté ve ark. (2014) ortaya koymuştur. Gendreau ve ark. (2008) probleme yasak arama sezgisel algoritması ile çözüm getirmiştir. Ayrıca Leung ve ark. (2011) yasak arama yöntemini genişleterek problem için yeni bir paketleme sezgiseli geliştirmiştir. Fuellerer ve ark. (2009) ortaya koyduğu çalışmada problemin yükleme kısmı farklı sezgisel yöntemlerle ele alınırken genel çözümü için karınca kolonisi optimizasyon yöntemi kullanılmıştır. Aynı şekilde Bin ve ark. (2013) yaptığı çalışmada yükleme problemi için farklı sezgisel yöntemler kullanılırken genel çözüm için yapay arı kolonisi algoritması kullanılmıştır. Problemin çözümünde kullanılan diğer bir sezgisel yöntem tavlama benzetimi sezgiselidir. Leung ve ark. (2010) tarafından önerilen yöntemde algoritmayı hızlandıran etkili veri yapıları ve paketleme sezgiselleri içeren yeni bir yaklaşım kullanılmıştır. Duhamel ve ark. (2011) tarafından geliştirilen diğer yöntemler ise GRASP (a greedy randomized adaptive search procedure) ve ELS (evolutionary local search) algoritmalarıdır. Çalışmada basit bir yükleme kısıtı için elde edilen sonuç kaynak kısıtlı proje çizelgeleme problemine (resource constrained project scheduling problem) dönüştürülmüştür. Artık, çözüm algoritması oluşan bu gevşetilmiş problem ile çözüm arayarak en son bulunan çözüm 2L-CVRP problem çözümüne dönüştürülmektedir. 2L-CVRP problemlerinde az rastlanan, ürünlerin 90 derece döndürülmesi varsayımı Dominguez ve ark. (2014) tarafından ele alınmıştır. Wei ve ark. (2015) rota ve yükleme problemleri için iki farklı sezgisel yöntem önermiştir. Rotalama problemi için komşu arama algoritması, yükleme problemi için ise ufuk çizgisi (skyline) algoritması kullanılmıştır. iki ve üç boyutlu problemler için bir müşterinin yükü indirilirken diğer müşterinin yükünün hareket etirilmemesi yani boşaltma kısıtı altında dal-kes algoritması Hokama ve ark. (2016) tarafında önerilmiştir.

Üç boyutlu yükleme probleminde iki boyutlu probleme ek olarak yüklerin ve yükleme alanının yüksekliği göz önünde bulundurulur. Yükler, konteyner içinde üst üste yerleştirilebilir. Bu durum; probleme kırılganlık, dikey durağanlık vb. kısıtların eklenmesini gerektirebilmektedir. Üç boyutlu yükleme kısıtı altında rotalama problemine kapsamlı matematiksel model Türkay ve Emel (2003) tarafından geliştirilmiştir. Problemin NP-zor yapıda olmasından dolayı literatürde sezgisel çözüm yöntemleri oldukça fazladır. Gendreau ve ark. (2006) ilk olarak 3L-CVRP problemine yasak arama algoritması kullanarak sezgisel çözüm yaklaşımı geliştirmiştir. Ele alınan problem kırılganlık, LIFO, dikey döndürme (yüklerin en-boy düzleminde 90 derece döndürülmesi) varsayımlarını içermektedir. Fuellerer ve ark. (2010) benzer varsayımlar altında aynı problemi karınca kolonisi optimizasyonu yöntemi ile ele almıştır. 3L-CVRP problemine aynı varsayımlar altında yasak arama algoritması kullanarak çözüm getiren diğer bir çalışma Wisniewski ve ark. (2012) aittir. Zhu ve ark. (2012) homojen araçlardan oluşan filo için çözüm yöntemi olarak iki aşamalı yasak arama algoritmasını kullanmıştır. Tao ve Wang (2015) yasak arama algoritmasını etkili bir biçimde geliştirerek literatürde bulunan diğer çözüm yöntemleri ile karşılaştırmıştır. Yasak arama algoritması kullanarak probleme çözüm getiren çalışmalardan bir diğeri Bortfeldt (2012)'e aittir. Aprile ve ark. (2007) paketleme kısıtları içeren 3L-CVRP modeline tavlama benzetimi yöntemi ile çözüm getirmiştir. 3L-CVRP probleminin zaman penceresi kısıtı altında incelendiği ilk çalışma Moura ve Oliveira (2009) aittir. Çalışmada, üç boyutlu yükleme ve zaman pencereli araç rotalama problemleri için iki ayrı çözüm yöntemi kullanılmıştır. İlk olarak tümleşik problemin ele alındığı matematiksel model ile sonuç elde edilmiştir. Problem çözümünde ikinci olarak ardışık aday listesi yöntemi kullanılmıştır. Aynı şekilde Bortfeldt ve Homberger (2013) de zaman pencereli 3L-CVR problemini ele almıştır. Çalışmada iki aşamalı yeni bir sezgisel algoritma kullanılmıştır. İlk olarak müşteri istekleri doğrultusunda üç boyutlu yükleme problemi çözülmüş, ikinci aşamada ise yükleme sonucuna uygun rota oluşturulmuştur. Ceschia ve ark. (2013) da problemi zaman penceresi kısıtı altında incelemiştir. Problemin çözümü için yerel arama sezgisel algoritması kullanılmıştır. Zhang ve ark. (2015) minimum yakıt tüketimi altında evrimsel yerel arama sezgisel yöntemini kullanmıştır. Ren ve ark. (2011) 3L-CVRP problemini, kapasite kısıtlı araç rotalama problemi ve yükleme problemi olarak ele almıştır. Yükleme probleminin çözümü, rotalama probleminden ayrı olarak ağaç arama (tree search) sezgisel yöntemi kullanılarak elde edilmiştir. Ayrıca 3L-CVRP probleminin çözümü için gevşetilmiş model kullanılmıştır. Bu sayede araç kullanımı artmış, maliyet azalmıştır. Lacomme ve ark. (2013) 3L-CVRP çözümünde GRASP- ELS sezgiselini kullanmıştır. Probleme farklı bir çözüm yaklaşımını Massen ve ark. (2012) tarafından getirilmiştir. Problemin çözümünde sütun oluşturma (column generation) yöntemi kullanılmıştır. Ayrıca rota doğruluğu kara kutu fonksiyonu ile kontrol edilmektedir. 3L-CVRP çözümünde yükleme problemi için yasak arama algoritması kullanılan diğer bir çalışma Miao ve ark. (2012) aittir. Çalışmada rotalama problemi için genetik algoritma kullanılmıştır. İki yöntem entegre edilerek hibrid bir algoritma oluşturulmuştur. 3L-CVR probleminin çözümü için Junqueira ve ark. (2013) ve Ruan ve ark. (2013) tarafından matematiksel model kullanılmıştır. Ruan ve ark. yaptığı çalışmanın devamında rotalama probleminin çözümünde bal arısı algoritması, yükleme probleminin çözümünde ise altı tip sezgisel algoritma olarak adlandırdıkları sezgisel algoritma kullanılmıştır. Junqueira ve Morabito (2015) da problemi sezgisel yaklaşımla ele almıştır.

2L-CVRP ve 3L-CVR problemlerinde, bir merkez depodan dolu olarak çıkan aracın müşterilere uygun rotada dağıtım yapması gerekir. Fakat gerçek hayatta bu varsayımların dışında kalan durumlarla karşılaşılmaktadır. Bunlardan en önemlisi Bölüm 2.1.'de bahsedilen rota boyunca dağıtım ve toplama işlemlerinin yapıldığı durumdur. Bahsedilen dağıtım toplamalı araç rotalama problemi, 2L veya 3L-CVRP problemine entegre edilerek rota boyunca yüklerin dağıtım ve toplamasının yapıldığı aynı zamanda konteyner içerisine optimal şekilde yerleşimlerinin sağlandığı oldukça zor bir problem olarak ortaya çıkmaktadır. Ayrıca problem, farklı bir çok kısıt içerebilir. Bu kısıtların en önemlisi son giren ilk çıkar (LIFO, last in-first out) kısıtıdır. Ürün dağıtımı sırasında zaman, iş gücü, ekipman ve maliyet kaybının azaltılması için kullanılması gerekmektedir. Bu kısıt belirli bir merkez depodan yüklenen yüklerin sıra bağımlı olarak uygun rota boyunca herhangi bir yer değişikliğine maruz bırakılmadan müşterilere ulaştırılmasını hedefler. Bu açıdan direkt olarak rotayı etkilemektedir. LIFO kısıtı, yüklerin rota boyunca yüklenip boşaltılmasının eş zamanlı olarak yapıldığı durumunda problemi daha zor hale dönüştürmektedir. Araç boyutunun iki veya üç boyutlu olması problemin zorluğunu artırmakta ve LIFO kısıtının sağlanmasını önemli ölçüde etkilemektedir. Bu nedenle bu çalışmada 2L-CVRPDP-LIFO problemi üzerinde durulacaktır. Bölüm 2.4, literatürde LIFO kısıtı içeren çalışmaları ve kullanılan çözüm yaklaşımları sunulmuştur.

2.4. Son Giren İlk Çıkar (LIFO-Last In First Out) Kısıtlı Araç Rotalama Problemi

Lojistik literatüründe; sıralı yükleme, arkadan yükleme ya da son giren-ilk çıkar kısıtı olarak yer alan LIFO kısıtı araç rotalama problemlerinde son yıllarda giderek önem kazanmıştır. Uygulamada konteyner içerisinde yüklerin yüklenmesi ve boşaltılması sırasında meydana gelen gereksiz taşıma ya da yeniden yerleştirme faaliyetlerinin engellemek amacıyla tanımlanan LIFO kısıtı; rotada bir sonraki müşteri tarafından talep edilen yük ile aracın boşaltma kapısı arasında diğer yüklerin bulunmaması şeklinde tanımlanabilir (Iori ve Martello 2010). LIFO kısıtı özellikle dağıtım toplamalı ve yığın yükleme problemlerinde yer almaktadır. Bu kısıta ilişkin literatürde yer alan çalışmalar Çizelge 2.3'te görülmektedir.

Çizelge 2.5. Literatürde LIFO kısıtını ele alan çalışmalar.

Çalışma	Yazar-Yıl	Problem tipi						Model / Çözüm		
		TSP	**VRP**	**PD**	**2DL**	**3DL**	**LIFO**	**MIP/ IP**	**BB/BC/CG**	**Sezgisel**
An additive branch-and-bound algorithm for the pickup and delivery traveling salesman problem with LIFO or FIFO loading	(Carrabs ve ark. 2007a)	X		X			X		X	
Variable neighborhood search for the pickup and delivery traveling salesman problem with LIFO loading	(Carrabs ve ark. 2007b)	X		X			X			X
The double traveling salesman problem with multiple stacks: a variable neighbor hood search approach	(Felipe ve ark. 2009)	X		X			X	X		X
The double travelling salesman problem with multiple stacks formulation and heuristic solution approaches	(Petersen ve Madsen 2009)	X		X			X	X		X
A branch-and-cut algorithm for the pickup and delivery traveling salesman problem with LIFO loading	(Cordeau ve ark. 2010)	X		X			X	X	X	X
An exact method for the double TSP with multiple stacks	(Lusby ve ark. 2010)	X		X			X			X
Research on the algorithm for 3L-CVRP with considering the utilization rate of vehicles	(Ma ve ark. 2011)		X			X	X			X
Multiple pickup and delivery TSP with LIFO and distance constraints: A VNS approach	(Gao ve ark. 2011)	X		X			X			X
A tabu search algorithm for the capacitated vehicle routing problem with three-dimensional loading constraints	(Wisniewski ve ark. 2012)		X			X	X			X
A branch-and-cut algorithm for the pickup and delivery traveling salesman problem with multiple stacks	(Côté ve ark. 2012a)	X		X			X	X	X	X
Multiple pickup and delivery traveling salesman problem with last-in-first-out loading and distance constraints	(Cheang ve ark. 2012)	X		X			X			X
The traveling purchaser problem with multiples stacks and deliveries: A branch-and-cut approach	(Batista-Galván ve ark. 2013)	X		X			X	X	X	X

Çizelge 2.3. Literatürde LIFO kısıtını ele alan çalışmalar (devam)

Çalışma	Yazar-Yıl	Problem tipi						Model / Çözüm		
		TSP	**VRP**	**PD**	**2DL**	**3DL**	**LIFO**	**MIP/ IP**	**BB/ BC/CG**	**Sezgisel**
A hybrid approach for the vehicle routing problem with three-dimensional loading constraints	(Ruan ve ark. 2013)		X	X		X	X	X		X
A population-based metaheuristic for the pickup and delivery problem with time windows and LIFO loading	(Cherkesly ve ark. 2015)		X	X			X			X
Solving a long-distance routing problem using ant colony optimization	(Royo ve ark. 2015)		X	X			X	X		X
The multiple vehicle pickup and delivery problem with LIFO constraints	(Benavent ve ark. 2015)		X	X			X	X	X	X
A dynamic programming based local search approach for the double traveling salesman problem with multiple stacks	(Urrutia ve ark. 2015)	X		X			X			X
The vehicle routing problem with simultaneous pickups and deliveries and two-dimensional loading constraints	(Zachariadis ve ark. 2016)		X	X	X		X			X
Branch-price-and-cut algorithms for the pickup and delivery problem with time windows and multiple stacks	(Cherkesly ve ark. 2016)		X	X			X	X	X	X
New formulation and branch-and-cut algorithm for the pickup and delivery traveling salesman problem with multiple stacks	(Sampaio ve Urrutia 2017)	X		X			X	X	X	
The pickup and delivery problem with time windows and handling operations	(Veenstra ve ark. 2017)		X	X			X		X	

LIFO kısıtının araç rotalama problemi literatüründe bulunan ilk örneği Ladany ve Mehrez (1984)'e aittir. Çalışmada dağıtım toplamalı gezgin satıcı problemi LIFO kısıtı altında ele alınmıştır. Yine aynı problem Pacheco (1997) tarafından ele alınmış çözüm önerisi olarak komşu arama sezgisel yöntemi kullanılmıştır. Carrabs ve ark. (2007a) bahsedilen probleme ilk giren-ilk çıkar (FIFO, First In-First Out) kısıtını ekleyerek, problemin çözümünde dal-sınır algoritması kullanmıştır. Carrabs ve ark. (2007b) yine aynı problem için değişken komşu arama algoritması kullanarak probleme çözüm getirmiştir. Probleme dal-sınır algoritması ile çözüm getirilen diğer bir çalışma Cordeau ve ark. (2010)‘ a aittir. Çalışmada gezgin satıcı problemine yönelik üç farklı matematiksel model önerilmiştir. Ayrıca LIFO kısıtını sağlamaya yönelik yeni eşitsizlikler üretilmiştir. Dal-kesme algoritmasında öncelikle ayrıştırma prosedürü uygulanmış; ardından LIFO kısıtının sağlanması için yasak arama algoritması kullanılmıştır. Ayrıca LIFO kısıtı altında dağıtım toplamalı gezgin satıcı problemine birden fazla araç varsayımı ve mesafe kısıtı altında değişken komşu arama çözüm yöntemini Gao ve ark. (2011) geliştirmiştir. Probleme aynı varsayımlar altında metasezgisel olarak iki aşamalı çözüm getiren diğer bir çalışma Cheang ve ark. (2012)‘ a aittir. Çalışmada ilk olarak kullanılan araç sayısının enküçüklenmesi, ikinci aşamada ise toplam mesafenin enküçüklenmesi sağlanmaktadır. Yine aynı problem için Benavent ve ark. (2015) karşık tam sayılı matematiksel model önermiştir. Çalışmanın devamında çözüme dal-kesme algoritması ve sezgisel algoritma ile ulaşılmıştır. Problemi zaman penceresi kısıtı altında Cherkesly ve ark. (2015) ele almıştır. Probleme sezgisel yöntemler kullanılarak çözüm geliştirilmiştir. Zaman penceresi kısıtını dikkate alan diğer bir çalışma ise Veenstra ve ark. (2017) aittir. Bu iki problemin ortak noktası zaman penceresi kısıtının haricinde konteynerin bölümlere ayrılmış olduğu varsayılarak yığın (katman) şeklinde yükleme yapılmasıdır. Katman şeklinde önceden ayrılmış bölmeler içeren konteyner yükleme problemleri rotalama probleminin sık karşılaşılan bir türüdür. Konteyner boyunca uzanan doğrusal yığın yapısı ürünlerin rota sırasına uygun olarak yüklenmesine olanak sağlamaktadır. Aynı zamanda yüklerin boşaltılması durumunda zorunlu olarak yığında önde bulunan yük boşaltılmalıdır. Aksi halde gereksiz yükleme ve boşaltma işlemi yapılır. Yığın yapısı kendi doğası gereği LIFO kısıtı ile paralellik göstermektedir. Felipe ve ark. (2009)`nın yaptıkları çalışma; toplam rota uzunluğunu en küçükleyecek şekilde, yüklerin eşit kapasiteli çoklu yığınlara toplama hattı olarak belirlenmiş ağ boyunca önce yüklenmesi, ardından dağıtım hattı olarak belirlenmiş ağ boyunca yükleme sırasına uygun olarak, yani LIFO prensibini göz önünde bulundurarak, boşaltılmasını amaçlamaktadır. Aynı problem tanımı altında Petersen ve Madsen (2009) probleme metasezgisel yöntemler kullanarak çözüm geliştirmiştir. Urrutia ve ark. (2015) probleme dinamik programlama yaklaşımıyla çözüm geliştirmiştir. Problemin, yüklerin paletler şeklinde taşındığı toplam taşıma maliyetinin en küçüklenmesini amaçlayan varyasyonu Lusby ve ark. (2010) tarafından ele alınmıştır. Dağıtım ve toplama noktalarının aynı rota üzerinde ziyaret edildiği problem türüne örnek olarak Côté ve ark. (2012a) ait çalışma verilebilir. Çalışmada sınırlı kapasiteye sahip yığınlar için rota boyunca maliyet en küçüklemesi sağlayacak şekilde tüm noktalar ziyaret edilmelidir. Çalışmada probleme ilişkin matematiksel model yer almaktadır. Problemin çözümünde dal-kesme algoritması kullanılmıştır. Aynı probleme dal-kesme algoritması ile çözüm getirilen diğer bir çalıma Sampaio ve Urrutia (2017) aittir. Çalışmada problem için karışık tamsayılı model önerilmiş ve çözüm için dal-kesme algoritması kullanılmıştır. Batista-Galván ve ark. (2013), birbirinden oldukça uzak dağıtım ve toplama noktaları için kullanılan gezgin alıcı problemini çoklu yığın ve LIFO kısıtı altında ele alarak probleme, benzer şekilde dal-kesme algoritması ile çözüm yaklaşımı geliştirmişlerdir. Cherkesly ve ark. (2016) probleme zaman penceresi kısıtını ekleyip dal-fiyat ve kesme algoritması kullanarak ağır ya da taşınması tehlikeli yükler için farklı bir çalışma ortaya koymuştur. LIFO kısıtının araç rotalama problemleri içerisinde yer aldığı çalışmalar şu şekilde örneklendirilebilir. Ma ve ark. (2011) üç boyutlu kapasite kısıtlı konteyner yükleme problemini LIFO kısıtı altında ele almıştır. Çalışmanın amacı toplam rota maliyetini en küçükleyecek ve araç kullanım oranını enbüyükleyecek uygun rotanın bulunmasıdır. Problemin çözümünde yerel arama ve yasak arama algoritmaları kullanılmıştır. Aynı problem Wisniewski ve ark. (2012) tarafından benzer şekilde ele alınmıştır. Rotalama ve yükleme problemleri ayrı incelenmiştir. Ruan ve ark. (2013) benzer probleme bal arası optimizasyon yöntemi ile çözüm getirmiştir. Royo ve ark. (2015) uzun mesafeli rotalama problemini, küçük yükler ve çeşitli zaman kısıtları altında LIFO prensibine uygun olarak en iyi şekilde çözmeyi amaçlamıştır. Problemin çözümü için matematiksel model önerilmiştir. Fakat önerilen model problemin gerçek boyutlarda çözümünde yetersiz kalmıştır. Bu nedenle karınca kolonisi optimizasyon yöntemi kullanılmıştır.

2.5. İki ve Üç Boyutlu Yükleme Kısıtı Altında Dağıtım Toplamalı Araç Rotalama Problemi

Dağıtım-toplamalı araç rotalama problemine iki ve üç boyutlu yükleme kısıtının entegre edildiği çalışmalar Çizelge 2.4`de görülmektedir. Chang ve Liao (2008) tarafından önerilen dağıtım-toplamalı araç rotalama problemi ve istifleme planlaması problemi üç boyutlu yerleştirme kısıtının bahsedildiği ilk çalışmadır. Bu çalışmada karışık dağıtım-toplamalı en kısa yol problemi ile istifleme planı için uygun sonuç bulunması hedeflenmiştir. Problem iki bölümde ele alınmıştır. İlk olarak karışık dağıtım-toplamalı en kısa yol probleminin çözümü için matematiksel model önerilmiştir. Fakat problem yapısı NP-Zor olduğundan uygun çözüm yöntemi olarak sezgisel algoritmalar kullanılmıştır. İstifleme problemi için geliştirilen matematiksel model ise ara şehirlerde yükleme boşaltmaların en küçüklenmesini amaçlamaktadır. Problemin NP-Tam yapısından dolayı üç boyutlu yerleştirme probleminin çözüm yöntemi için sezgisel algoritmalar kullanılmıştır.

Bartók ve Imreh (2011) 3L-VRPD probleminin ağırlık kısıtlı ilk örneğini sergilemişlerdir. Problemin çözümü için önerilen matematiksel modelde sadece parametreler ve bazı koşullar sözlü olarak tanımlanmıştır. İlgili problem sezgisel yöntemlerle çözülmüştür. Önerilen sezgisel algoritma; başlangıç, rotalama, basit lokal arama, paketleme ve gelişmiş lokal arama aşamalarından oluşan entegre bir yöntemdir. Önerilen algoritma, farklı ağırlık ve boyuta sahip araçlar için test edilmiştir. Männel ve Bortfeldt (2015a) tarafından ele alınan çalışmada dağıtım-toplamalı araç rotalama problemi için dikdörtgen yükleme alanına sahip araca, dikdörtgen yükler (kutular) yeniden yerleştirmeye izin vermeyecek şekilde yerleştirilerek optimal çözümün bulunması hedeflenmiştir. Önerilen çözüm yöntemi iki sezgisel algoritma içermektedir. İlk olarak tek boyutlu dağıtım-toplama problemi için rotalama probleminin çözümü komşu arama algoritması kullanılarak bulunmuştur. Ardından ağaç arama algoritması ile paketleme/yükleme kısıtı sağlanmıştır. Çalışmada, paketler yüklendikten sonra ve boşaltılmadan önce hiç bir şekilde yer değişikliği yapılamaz varsayımı yer almaktadır. Bu haliyle LIFO kısıtı sağlanmaktadır. LIFO kısıtına ek olarak yeniden yükleme yasağı olarak adlandırılan (reloading ban) kısıt çalışmada yer almaktadır. Bu kısıt rota boyunca değişen yükleme planında A müşterisine ait kutuların mevcut yerlerinin bu müşterinin dağıtım noktasına gelene kadar hiçbir şekilde değişmemesini garanti eder. Ayrıca bu kısıta ek olarak birbirinden bağımsız kısmi rota (independent partial routes) kısıtı da modele eklenerek ürünlerin toplama noktalarından alındıktan sonra dağıtım noktalarına götürülene kadar rotanın farklı bir bölümündeki farklı müşterinin kutularıyla aynı anda taşınmasına engel olur (Männel ve Bortfeldt 2015b).

Çizelge 2.6. Literatürde iki ve üç boyutlu yerleştirme kısıtını ele alan çalışmalar

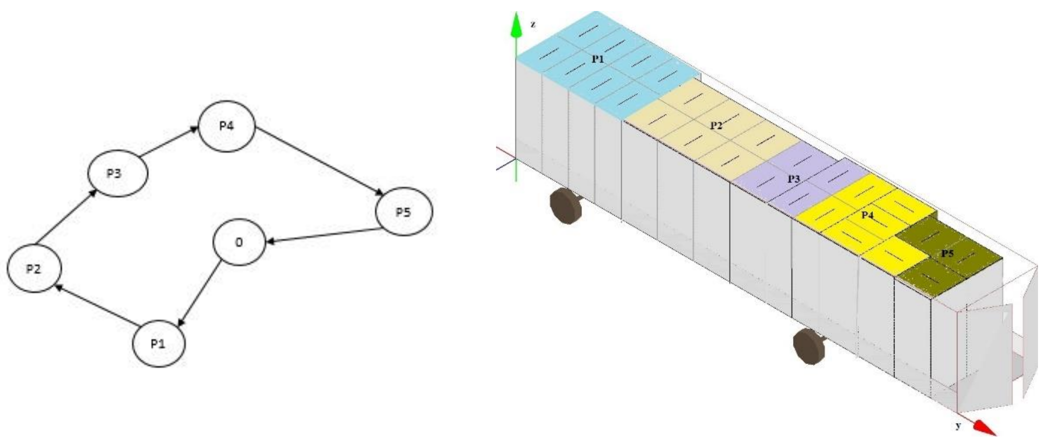
Çalışma	Yazar-Yıl	Problem tipi						Model / Çözüm		
		TSP	**VRP**	**PD**	**2DL**	**3DL**	**LIFO**	**MIP/ IP**	**BB/BC/CG**	**Sezgisel**
Path finding with stowage planning consideration in a mixed pickup–delivery and specified-node network	(Chang ve Liao 2008)		X	X		X	X	X		X
Pickup and Delivery Vehicle Routing with Multidimensional Loading Constraints	(Bartók ve Imreh 2011)		X	X		X				X
Designing vehicle routes for a mix of different request types, under time windows and loading constraints	(Zachariadis ve ark. 2013)		X	X		X				X
A Hybrid Algorithm for the Vehicle Routing Problem with Pickup and Delivery and 3D Loading Constraints	(Männel ve Bortfeldt 2015a)		X	X		X	X			X
Solving the Pickup and Delivery Problem with 3D Loading Constraints and Reloading Ban	(Männel ve Bortfeldt 2015b)		X	X		X	X			X
The Vehicle Routing Problem with Simultaneous Pick-ups and Deliveries and Two-Dimensional Loading Constraints	(Zachariadis ve ark. 2016)		X	X	X		X			X

Problemin iki boyutlu hali Zachariadis ve ark. (2016) tarafından ele alınmıştır. Çalışmada dağıtım ve toplama işlemleri eş zamanlı olarak yapılmaktadır. Bu yüzden yapılan her yüklemeden sonra rotada bulunan her ark için yük kapasitesi kontrol edilmelidir. Rota boyunca ürünlerin LIFO prensibine uygun olarak dağıtılması gerekmektedir. Zachariadis ve ark. (2013) ait olan diğer bir çalışmada dağıtım toplamalı rotlama problemi zaman penceresi kısıtı altında incelenmiştir. Bu çalışmada üç boyutlu kutular alanı belli bir konteynır içerisine iki boyutlu paletlerin üzerinde yerleştirilmektedir. Bu yaklaşımlar altında çalışma ilk niteliği taşımaktadır.

Literatürde bulunan yükleme kısıtı altında dağıtım toplamalı araç rotalama problemlerine bakıldığında problemin yapısından dolayı sezgisel algoritmalar kullanılmıştır. Sezgisel algoritmalar çözüm süresinde önemli iyileştirme sağlamalarına rağmen kesin çözümü garanti etmezler. Bu nedenle iki boyutlu dağıtım toplamalı araç rotalama problemine LIFO kısıtı altında çözüm getirecek matematiksel model ve model oluşturulurken kullanılan materyal ve yöntemler ilerleyen bölümlerde sunulmuştur.

3. MATERYAL VE YÖNTEM

Rotalama ve yükleme problemlerinin tümleşik bir yapıda ele alınması lojistik faaliyetlerinin optimizasyonu açısından önemlidir. Rotalama probleminde yükleme kısıtının dikkate alınması, ürünlerin araca yüklendikten sonra herhangi bir hasara uğramadan taşınmalarını sağlar. Yükleme kısıtı altında rotalama problemleri farklı varsayımlar altında karşımıza çıkmaktadır. Merkezi bir depodan boş halde çıkan ve rota boyunca yük toplayarak tekrar merkez depoya dönen araç rotalama problem tipi için örnek rota ve yükleme planı Şekil 3.1'de verilmiştir. Şekilde rota boyunca gidilen noktalara ait yüklerin sırayla araca yüklendiği görülmektedir. Bu tipin dışında merkezi depodan yüklü olarak çıkan aracın rota boyunca yükleri uygun sırada dağıttığı ya da toplama ve dağıtma işlemlerinin aynı rotada birlikte yapıldığı örnekler de mevcuttur. Toplama ve dağıtma işlemlerinin rota boyunca gerçekleştirildiği problem türü gerçek hayatta geniş uygulama alanına sahiptir. Literatüre bakıldığında dağıtım toplamalı araç rotalama ve yükleme problemlerini ayrı olarak ele alan bir çok çalışma karşımıza çıkmaktadır. Fakat tümleşik probleme ilişkin örnek sayısı oldukça azdır. Ayrıca literatürde tümleşik problemin çözümü için kullanılan herhangi bir kesin çözüm yaklaşımı bulunamamıştır. Elde edilen bu bilgiler doğrultusunda bu tez çalışmasının amacı; iki boyutlu yükleme kısıtı altında dağıtım toplamalı araç rotalama problemine kesin çözüm yaklaşımı geliştirilmesidir.



Şekil 3.1. Yükleme kısıtı altında rotalama problemine bir örnek (Packer 3d 2017)

Dağıtım toplamalı araç rotalama problemi araç rotalama probleminin doğasından dolayı NP-Zor problem sınıfında yer almaktadır. Ayrıca probleme yükleme ve LIFO kısıtı eklendiğinde daha da karmaşık bir yapı ortaya çıkmaktadır. Bu durumda tümleşik problem iki alt problem ve LIFO kısıtından oluşmaktadır. Literatüre bakıldığında probleme ilişkin varsayımlarda bazı esneklikler yapıldığı görülmektedir. Örneğin; LIFO kısıtı ihlal edilmiş, problem iki parçaya bölünmüş, rotalama problemi çözülmüş, ardından yükleme planı oluşturulmuş ve LIFO kontrolü yapılmış vb. Fakat LIFO kısıtının ihlal edilip edilmediği problem için önemli bir ayrımdır. LIFO kısıtı, yükleme kısıtı altında dağıtım toplamalı araç rotalama problemine uygulandığında rotayı doğrudan etkilemektedir. Bu kısıt uygulamada yüklü ürünlerin herhangi bir hasara uğramadan teslim edilmesi ve araç içerisinde ürünlerin tekrar yerleştirilmesinden kaynaklanan gereksiz zaman kayıplarının önlenmesi açısından oldukça önemlidir. Bölüm 2.4`te LIFO kısıtının araç rotalama problemlerinde kullanıldığı çalışmalar incelenmiştir. LIFO kısıtın mantıksal ifadelerle tanımlanarak matematiksel bir kısıt haline dönüştürülmesinin oldukça zor olduğu görülmekle beraber literatürde bazı örnekler mevcuttur ((Felipe ve ark. 2009), (Petersen ve Madsen 2009), (Cordeau ve ark. 2010), (Côté ve ark. 2012a), (Royo ve ark. 2015), (Benavent ve ark. 2015)). Ancak bu matematiksel modellerin yer aldığı problemlerin çözümünde sezgisel algoritmalar kullanılmıştır.

Yükleme kısıtı altında dağıtım toplamalı araç rotalama problemi için oluşturulacak bir matematiksel model ile bu modelde yeterli LIFO kısıtı yazmadan gerekli LIFO koşullarının sağlandığı kontrol edilerek problemin kesin çözümüne ulaşılabilir. Çalışmanın devamında model oluşturmada kullanılan kaynaklar, problem tanımı, oluşturulan matematiksel model ve uygulama örneği verilmiştir.

3.1. Materyal

Ele alınan tümleşik problem dağıtım-toplamalı araç rotalama problemi ve çok boyutlu yükleme kısıtı altında araç rotalama problemlerinden oluşmaktadır. Bu iki problem için literatürde matematiksel model örneklerine ve kesin çözüm yöntemlerine rastlamak mümkündür.

Dağıtım-toplamalı araç rotalama problemi ile ilgili olarak yapılan literatür araştırmasına Bölüm 2.1'de ulaşılabilir. Problem, rotalama probleminin doğası gereği NP-Zor yapıdadır. Probleme ilişkin olarak literatürde bulunan temel ve vektörel gösterimler aşağıda verilmiştir.

Cordeau (2006) tarafından ele alınan çalışmada, maliyet minimizasyonunu hedefleyen; kapasite, zaman penceresi ve öncelik kısıtlarını içeren matematiksel bir model önerilmiştir. Dağıtım-toplamalı araç rotalama problemine ilişkin notasyon ve kısıtlar şu şekildedir: n, hizmet verilen kullanıcı sayısını göstermektedir. $G = (N, A)$ tamamlanmış bir graf, $N = P \cup D \cup \{0,2n + 1\}$, $P = \{1, \dots, n\}$ ve $D = \{n + 1, \dots ,2n\}$ olarak tanımlanmıştır. P ve D alt kümeleri toplama ve dağıtım noktalarını içerir. 0 ve $2n + 1$ noktaları başlangıç ve hedef depo noktalarıdır. Her i kullanıcısı i başlangıç noktası ve $n + i$ hedef noktası ile ilişkilidir. K, araç kümesini göstermektedir. Her $k \in K$ olarak ifade edilmektedir. Her ayrıt $(i, j) \in A$ ve $k \in K$ olmak üzere, eğer k aracı i noktasından j noktasına gidiyorsa $x_{ij}^k = 1$'dir. Tanımlamalar doğrultusunda dağıtım-toplamalı araç rotalama problemine ait temel kısıtlar şu şekildedir:

$$\sum_{k \in K} \sum_{j \in N} x_{ij}^k = 1 \qquad \forall i \in P \tag{3.1}$$

$$\sum_{j \in N} x_{ij}^k - \sum_{j \in N} x_{(n+i)j}^k = 0 \qquad \forall i \in P, k \in K \tag{3.2}$$

$$\sum_{j \in N} x_{0j}^k = 1 \qquad \forall k \in K \tag{3.3}$$

$$\sum_{j \in N} x_{ji}^k - \sum_{j \in N} x_{ij}^k = 0 \qquad \forall i \in P \cup D,\ k \in K \tag{3.4}$$

$$\sum_{i \in N} x_{i(2n+1)}^k = 1 \qquad \forall k \in K \tag{3.5}$$

$(3.1) - (3.2)$ kısıtlar her isteğin yalnızca bir defa karşılanacağını ve başlangıç ve hedef noktalarının aynı araçla ziyaret edilmesini sağlamaktadır. $(3.3) - (3.5)$ kısıtlar her k

aracı için rotanın merkez depoda başlayıp hedef depoda bitmesini garantilemektedir (Cordeau 2006). Çalışmada toplama ve dağıtım noktaları için farklı küme tanımları yapılmış ve modellemede bu ayrıma dikkat edilmiştir. Benzer şekilde Ropke ve ark. (2007) ve Cordeau ve ark. (2010) çalışmalarında dağıtım-toplamalı araç rotalama problemi için benzer matematiksel modeli kullanmışlardır. Bu tez çalışmasında iki boyutlu yükleme kısıtı altında dağıtım-toplamalı araç rotalama problemi için oluşturulan matematiksel modelde rotalama kısıtları benzer küme tanımları kullanılarak oluşturulmuştur.

Yükleme kısıtı, ilk çalışmalarda yalnıca konteyner yükleme problemi olarak, rotalamadan bağımsız ele alınmıştır. Ardından yüklerin merkez depoda araca yüklenerek rota boyunca dağıtılması varsayımı kullanılmıştır. Yüklerin merkez depodan dağıtılması problemine Junqueira ve ark. (2013) tarafından yapılan çalışma örnek olarak verilebilir. Bu çalışmada üç boyutlu yükleme kısıtı altında araç rotalama problemine bir iyileştirme modeli sunulmuştur. Rota boyunca maliyet en küçüklemesi amaçlanmaktadır. Çalışmada üç boyutlu yüklemenin ele alınmasından dolayı probleme dikey kararlılık, kutuların yük taşıma dayanımı ve bir arada tutulma gerekliliği kısıtları eklenmektedir. $G = (N, A)$ için N depo noktası (node 1) ve müşterileri kapsayan noktalar kümesi $(node2, \dots, n)$ ile A (k,l) ayrıt kümesidir. V, (L, W, H) boyutlarına sahip araçların kümesidir. c_{kl}, k noktasından l noktasına gitmenin maliyeti, $(k, l) \in A$, ve $k \in N\backslash\{1\}$ olmak üzere D_k, k müşterisine ait negatif olmayan talebi göstermektedir. Her k $(k \in N\backslash\{1\})$ müşterisi için i $(i \in M)$ kutu tipi olmak üzere b_{ik} kutu sayısı; l_i uzunluk, w_i en ve h_i boyuna sahip i tipindeki kutuları göstermektedir. (Ayrıca $\sum_{i \in M} l_i\, w_i\, h_i\, b_{ik} = D_k,\ k \in N\backslash\{1\}$ ve $\sum_{k \in N} b_{ik} = b_i,\ i \in M$ olmak üzere b_i, i tipindeki toplam kutu sayısını göstermektedir.) $t \in N$ olmak üzere k noktasından l noktasına kaçıncı aşamada gidildiğini göstermektedir. Kutular araç içerisine ön-sol-alt köşeden başlanarak yerleştirilmektedir ve bu nokta koordinat sisteminin merkezi olarak kabul edilmektedir. (x, y, z) araca yerleştirilen bir kutunun olası koordinatlarıdır. Olası x, y, z eksenleri aracın $X = \{0,1,2, \dots, L - min_i(l_i)\}$, $Y = \{0,1,2, \dots, W - min_i(w_i)\}$ ve $Z = \{0,1,2, \dots, H - min_i(h_i)\}$ kümelerine aittir. X, Y, Z kümeleri;

$$X = \left\{ x : x = \sum_{i \in M} \varepsilon_i\, l_i\,,\ \ 0 \leq x \leq L - min_i(l_i),\ \ 0 \leq \varepsilon_i \leq b_i \text{ ve tamsayı},\ \ i \in M \right\}$$

$Y = \{y : y = \sum_{i \in M} \varepsilon_i \, w_i \,, \;\; 0 \le y \le W - min_i(w_i), \;\; 0 \le \varepsilon_i \le b_i \text{ ve tamsayı}, \;\; i \in M\}$

$$Z = \left\{ z : z = \sum_{i \in M} \varepsilon_i \, h_i \,, \;\; 0 \le z \le H - min_i(h_i), \;\; 0 \le \varepsilon_i \le b_i \text{ ve tamsayı}, \;\; i \in M \right\}$$

Olmak üzere;

$X_i = \{x \in X : 0 \le x \le L - l_i\}, \;\; i \in M$

$Y_i = \{y \in Y : 0 \le y \le W - w_i\}, \;\; i \in M$

$Z_i = \{z \in Z : 0 \le z \le H - h_i\}, \;\; i \in M$

kümeleri X, Y ve Z'nin alt kümeleridir. Bu kümeler, modelde her kutunun tam olarak araç içerisinde kalmasını sağlamaktadır.

Karar değişkenleri;

$$d_{kl}^{tv} = \begin{cases} 1, & v \text{ aracı } t. \text{ aşamada } k \text{ noktasından } l \text{ noktasına direkt giderse;} \\ 0, & \text{aksi halde.} \end{cases}$$

$$a_{xyz}^{iktv} = \begin{cases} 1, & t. \text{ aşamada ziyaret edilen } k \text{ müşterisinin } i \text{ tipindeki} \\ & \text{kutusu } v \text{ aracının ön-sol-alt köşesi } (x, y, z) \text{ pozisyonunda ise} \\ 0, & \text{aksi halde.} \end{cases}$$

Önerilen matematiksel model aşağıdaki gibidir:

$$\min \sum_{k \in N} \sum_{l \in N} \sum_{t \in N} \sum_{v \in V} c_{kl} \, d_{kl}^{tv} \tag{3.6}$$

$$\sum_{l \in N} \sum_{t \in N} \sum_{v \in V} d_{kl}^{tv} = 1 \qquad k \in N \backslash \{1\} \tag{3.7}$$

$$\sum_{l \in N} \sum_{t \in N \backslash \{1\}} \sum_{v \in V} t \, d_{kl}^{tv} - \sum_{p \in N} \sum_{t \in N} \sum_{v \in V} t \, d_{pk}^{tv} = 1 \qquad k \in N \backslash \{1\} \tag{3.8}$$

$$\sum_{l \in N \backslash \{1\}} d_{1l}^{1v} \le 1 \qquad v \in V \tag{3.9}$$

$$\sum_{l \in N} d_{kl}^{(t+1)v} - \sum_{p \in N} d_{pk}^{tv} = 0 \qquad k \in N \backslash \{1\}, t \in N \backslash \{n\}, v \in V \tag{3.10}$$

$$\sum_{i\in M}\sum_{k\in N\setminus\{1\}}\sum_{l\in N}\sum_{t\in N\setminus\{1\}} l_i\, w_i\, h_i\, b_{ik}\, d_{kl}^{tv} \leq L\, W\, H \qquad v\in V \tag{3.11}$$

$$\sum_{i\in M}\sum_{x\in X_i}\sum_{y\in Y_i}\sum_{z\in Z_i} a_{xyz}^{iktv} = \sum_{i\in M}\sum_{l\in N} b_{ik}\, d_{lk}^{tv} \qquad k\in N\setminus\{1\}, t\in N\setminus\{n\}, v\in V \tag{3.12}$$

$$\sum_{i\in M}\sum_{k\in N\setminus\{1\}}\sum_{t\in N\setminus\{n\}}\sum_{\{x\in X_i:x'-l_i+1\leq x\leq x'\}}\sum_{\{y\in Y_i:y'-w_i+1\leq y\leq y'\}}\sum_{\{z\in Z_i:z'-h_i+1\leq z\leq z'\}} a_{xyz}^{iktv} \leq 1 \qquad x'\in X, y'\in Y, z'\in Z, v\in V \tag{3.13}$$

$$\sum_{t\in N\setminus\{n\}}\sum_{v\in V}\sum_{x\in X_i}\sum_{y\in Y_i}\sum_{z\in Z_i} a_{xyz}^{iktv} = b_{ik} \qquad i\in M, k\in N\setminus\{1\} \tag{3.14}$$

Bu formülasyonda, (3.6) amaç fonksiyonunun hedefi müşterileri ziyaret edecek araçların toplam maliyetlerini (ya da toplam uzaklığı) minimize etmektir. Kısıt (3.7) müşterilerin en az bir kez ziyaret edilmesini sağlamaktadır. Kısıt (3.8) her turu birbirine bağlamayı sağlamaktadır. Kısıt (3.9) ilk aşamada her aracın depodan en fazla bir kez çıkmasını sağlamaktadır. Kısıt (3.10), eğer v aracı p müşterisinden k müşterisine t. aşamada gittiyse $(t+1)$. aşamada aynı aracın k müşterisinden başka bir l müşterisine gitmesini sağlanmaktadır. Kısıt (3.11) ise her aracın kendisine ayrılan kapasiteyi aşmamasını sağlanmaktadır. Kısıt (3.12) rotalama ve yükleme çiftinin bağlantısını sağlamaktadır. Örneğin; eğer v aracı, t. aşamada l müşterisinden sonra k müşterisini ziyaret ederse, k müşterisi tarafından istenen kutular o aşamada boşaltılmalıdır. Kısıt (3.13), kutuların örtüşmemesini sağlamaktadır; örneğin, v aracında (x', y', z') noktalarını (kutunun üst, arka ve sağ yüzündeki noktalar hariç olmak üzere) içeren bir kutu varsa yalnızca bu kutu bu noktaları içerebilir (Beasley 1985, Junqueira ve ark. 2012). Kısıt (3.14), talebin karşılanmasını sağlamaktadır (Junqueira ve ark. 2013).

Çalışmada önerilen matematiksel model yukarıda belirtilen kısıtlara ek olarak dikey kararlılık, kutuların yük taşıma dayanımı ve bir arada tutulma gerekliliği için bir takım kısıtlar daha içermektedir. Ancak çalışmanın sonuç kısmına bakıldığında üç boyutlu yükleme modeli için kutu yerleşimlerinde fiziksel olarak gerçekleşmesi mümkün olmayan durumların meydana geldiği görülmektedir. Örneğin, bazı kutuların araç tabanı ile hiçbir teması olmadan, havada kaldığı durumlar söz konusu olmaktadır. Bu sorun, aslında problem için kurulan böylesine karmaşık bir modelin dahi kapsayıcı olmayabileceğini göstermektedir.

Üç boyutlu modellemede söz konusu olumsuz durumlarla karşılaşmamak üzere, bu tez çalışmasında yükleme problemi iki boyutlu olarak ele alınmıştır.

3.2. Problemin Tanımı

Bu çalışmanın amacı; bilinen müşteri taleplerinin maliyet en küçüklemesi sağlanarak karşılanmasına yönelik kesin çözüm yaklaşımı geliştirmektir. Ele alınan tümleşik problem; konteyner yükeleme ve dağıtım toplamalı araç rotalama problemlerinden oluşmaktadır.

Model, bir graf üzerinden şöyle tanımlanabilir; $G = (V, A)$'nin bütün bir graf olduğu varsayımı altında, $V = \{0,1, \dots, n, n+1, \dots 2n, 2n+1\}$ tüm noktaların kümesi, A ise (i, j) ayrıt çiftlerini birbirine bağlayan ayrıt setini göstermektedir. N toplam nokta sayısı olmak üzere her biri i yükleme ve $n+i$ boşaltma noktasıyla ilişkili olan n adet toplama isteği vardır. $V+= \{1,2, \dots 2n+1\}$ pozitif noktalar kümesini tanımlar. Her $(i, j) \in A$ olmak üzere negatif olmayan c_{ij} maliyeti i noktasından j noktasına gidiş maliyetini göstermektedir. 0 ve $2n+1$ merkez depoyu göstermektedir. Her i noktasından kendine gidiş maliyeti $c_{ii} = 0$'dır. P, toplama noktaları kümesi, D dağıtım noktaları kümesi, $S = V \setminus \{0,2n+1\}$ depo noktaları hariç tüm noktaların kümesini göstermektedir. $Q_{ağırlık}$ aracın ağırlık kapasitesini, Q_{alan} aracın alan kapasitesini göstermektedir. Z, amaç fonksiyonu değerini göstermektedir.

Problem varsayımları;

- Rota bir merkez depodan başlar ve aynı depoda sona erer.
- İlk başta araç boştur.
- Araç yalnızca bir rota oluşturabilir.
- Bir müşteriye yalnızca bir araç tarafından hizmet verilebilir ve bu araç müşteri taleplerinin tümünü karşılar.

- Araç depodan ayrıldıktan sonra herhangi bir dağıtım noktasına uğramadan önce mutlaka en az bir toplama noktasına uğramalıdır.
- Rota boyunca araç ağırlık ve alan kapasitesi aşılamaz.
- Müşteri talebi deterministiktir.
- Müşteri talepleri standart paletlere önceden yüklenmiş şekilde taşınmaktadır.
- Her müşteri yalnızca bir palet kullanabilir.
- Müşterilerin paletlere konteyner yüksekliğini aşmayacak şekilde yükleme yaptığı varsayılmaktadır.
- Palet boyutları tam sayıdır.
- Paletler araca sol ön köşeden başlanarak yerleştirilir.
- Palet eni ve uzunluğu sırasıyla x ve y eksenleri ile paralel olamalıdır.
- Yükleme ve boşaltma işlemleri aracın arkasından araç uzunluğu boyunca yapılmaktadır.
- Paletler araca LIFO prensibine uygun olarak yerleştirilmelidir. Örneğin sıradaki nokta dağıtım noktası ise bu noktaya ait paletin kapı ile arasında herhangi bir noktaya ait palet bulunamaz.
- Araç içi hareketler yasaklanmıştır. Palet yüklendikten sonra ve indirilmeden önce her hangi bir şekilde yeri değiştirilemez.

Problem parametreleri;

$O = \{1, \ldots ,2n + 1\}$ ziyaret edilen müşterinin hangi sırada ziyaret edildiğini gösteren sıra kümesi ($b \in O$).

f_{iv} F palet tiplerinin sayısı olmak üzere ($v \in F$), i müşterisine ait v tipinden kaç adet palet olduğunu gösterir. Ele alınan problemde her müşteri için bir adet, tek tip palet bulunmaktadır.

$INDIS_i$ $i \in P$ olmak üzere i toplama noktası ile ilişkilendirilen dağıtım noktası kümesidir. Örneğin; $INDIS_{(2)} = n + 2$.

BOY, EN konteynerin boyu ve eni.

$x \in X,\ y \in Y$ konteynerin eni ve boyu doğrultusunda palet sol alt köşesinin koordinatları.

X_x, Y_y koordinatlara özel parametreler.

k, K başlangıç depo noktası hariç nokta sayısı.

p_i i toplama noktasına ait toplama talep miktarı.

d_j j dağıtım noktasına ait dağıtım talep miktarı.

H_i i toplama noktasına ait yük alanı.

3.3. Matematiksel Model

Literatürde araç rotalama problemine ait farklı kısıtlar ve varsayımlar altında bir çok matematiksel model bulunmaktadır. Bu çalışmada ele alınan tümleşik problem için önerilen matematiksel model aşağıda verilmiştir.

İkili Değişkenler;

$$R_{ijb} = \begin{cases} 1, i \text{ noktasından } j \text{ noktasına } b. \text{ sırada gidiliyorsa} \\ 0, \text{aksi halde} \end{cases}$$

$$a_{ivbxy} = \begin{cases} 1, i \text{ müşterisine ait } v \text{ tipi palet } x, y \text{ koordinatlarına yüklenirse} \\ 0, \text{aksi halde} \end{cases}$$

Tam Sayı Değişkenler;

u_i = Alt tur engelleme değişkeni

L_j = j noktasından çıktıktan sonra kamyonun toplam yükü

Amaç Fonksiyonu:

$$\min \sum_{i \in V} \sum_{j \in V} \sum_{b \in O} c_{ij} R_{ijb} + \sum_{j \in V} \sum_{v \in F} \sum_{b \in O} \sum_{x \in X} \sum_{y \in Y} a_{jvbxy} \ (X_x + Y_y) \quad (3.15)$$

Kısıtlar:

$$\sum_{j \in P} R_{0j1} = 1 \quad (3.16)$$

$$\sum_{j \in V+} \sum_{b \in O} R_{ijb} = 1 \qquad i \in S, i \neq j, b > 1 \quad (3.17)$$

$$\sum_{i\neq q} R_{iqb} - \sum_{j\neq q} R_{qj(b+1)} = 0 \quad q \in S, b \in O\backslash\{2n+1\}, i \in V\backslash\{2n+1\}, j \in V\backslash\{0\} \quad (3.18)$$

$$\sum_{i\in D} R_{i(2n+1)(2n+1)} = 1 \qquad (3.19)$$

$$R_{ijb} = 0 \qquad i \in P, \quad j \in D, j \neq indis[i], \quad b \in O \qquad (3.20)$$

$$u_i - u_j + k\, R_{ijb} \leq k - 1, \qquad i > 0, j > 0, i \neq j, b \in O \qquad (3.21)$$

$$L_i \geq 0 + p_i - d_i - M\,(1 - R_{0i1}) \qquad i \in P \qquad (3.22)$$

$$L_j \geq L_i + p_j - d_j - M\left(1 - R_{ijb}\right) \qquad i, j \in S, b \in O\backslash\{1\}, i \neq j \qquad (3.23)$$

$$p_j \leq L_j \qquad j \in P \qquad (3.24)$$

$$0 \leq L_{(n+j)} \leq Q_{ağırlık} - p_j \qquad j \in P \qquad (3.25)$$

$$L_{(2n+1)} = 0 \qquad (3.26)$$

$$L_0 = 0 \qquad (3.27)$$

$$R_{0j1} \leq 1 \qquad j \in P \qquad (3.28)$$

$$\sum_{b\in O} R_{ijb} \leq 1 \qquad i \in S \;\; j \in V_+, i \neq j, b > 1 \qquad (3.29)$$

$$R_{i(2n+1)(2n+1)} \leq 1 \qquad i \in D \qquad (3.30)$$

$$\sum_{j} R_{0j1} = 1 \qquad j \in P \qquad (3.31)$$

$$\sum_{i\in S}\sum_{j\in S} R_{ijb} = 1 \qquad i \neq j, b \in O\backslash\{1, 2n+1\} \qquad (3.32)$$

$$R_{j(2n+1)(2n+1)} \leq 1 \qquad j \in D \qquad (3.33)$$

$$R_{j(n+j)2} \leq R_{0j1} \qquad j \in P \qquad (3.34)$$

$$R_{jq2} \leq R_{0j1} \qquad j \in P, \;\; q \in P, \;\; j \neq q \qquad (3.35)$$

$$R_{j(n+j)(b+1)} - m\, R_{ijb} \geq 0 \qquad i\, \in S, j \in P,\;\; i \neq j, b\; \in O \backslash \{2n+1\}, b > 2 \qquad (3.36)$$

$$\sum_{i \in V} \sum_{j \in V_+} \sum_{b \in O} \sum_{v \in F} H_i\; f_{iv}\, R_{ijb} \leq Q_{alan} \qquad i < 2n+1, i \neq j \qquad (3.37)$$

$$\sum_{\{x \leq W - minw_i\}} \sum_{\{y \leq L - minl_i\}} \sum_{\{v \in F\}} a_{ivbxy} = \sum_{j \in V \backslash \{2n+1\}} \sum_{v \in F} R_{jib}\, f_{iv}$$

$$i \in V_+, b \in O, i \neq j \qquad (3.38)$$

$$\sum_{\{i \in V\}} \sum_{\{b \in O\}} \sum_{\{xx - w_i + 1 \leq x \leq xx\}} \sum_{\{yy - l_i + 1 \leq y \leq yy\}} \sum_{\{v \in F\}} a_{ivbxy} \leq 1$$

$$xx \leq W - w_i, yy \leq L - l_i \qquad (3.39)$$

$$\sum_{b \in O} \sum_{x \leq W - w_i} \sum_{y \leq L - l_i} a_{ivbxy} = f_{iv} \qquad \forall i,\;\; \forall v \qquad (3.40)$$

$$R_{ijb}\; M + X_x\; a_{iv(b-1)xy} \leq X_{x''}\; a_{jvbx''y''} + M$$

$$i \in P, j \in P, i \neq j, b \in O, b > 1, x \in X, x'' \in X, x \neq x'', y \in Y, y'' \in Y, y \neq y'', v \in F \qquad (3.41)$$

$$R_{ijb}\; M + Y_y\; a_{iv(b-1)xy} \leq Y_{y''}\; a_{jbx''y''} + M$$

$$i \in P, j \in P, i \neq j, b \in O, b > 1, x \in X, x'' \in X, x \neq x'', y \in Y, y'' \in Y, y \neq y'', v \in F \qquad (3.42)$$

Kısıt (3.15) maliyeti minimize edecek amaç fonksiyonu değerini göstermektedir. Kısıt (3.16), aracın başlangıç noktasından çıktıktan sonra mutlaka bir toplama noktasına gidilmesini sağlamaktadır. Kısıt (3.17), bir noktaya yalnızca bir kere gidilmesini ve noktadan bir kere çıkılmasını sağlamaktadır. Kısıt (3.18), rota boyunca dengeyi sağlamaktadır. Kısıt (3.19), rotada en son depoya gidilmesini sağlamaktadır. Bunu sağlamak için sanal bir depo noktası kullanılmıştır. Kısıt (3.20), rota boyunca i toplama noktasından çıkıldıktan sonra $n + i$ dağıtım noktasından farklı bir dağıtım noktasına gidilmesini önlemektedir. Bu sayede i toplama noktasına ait yük $n + i$ dağıtım noktasına gelmeden önce diğer yükler ile hiçbir etkileşimde bulunmadan indirilebilir. Aksi durumda $n + i$ dağıtım noktasından farklı bir dağıtım noktasına gidildiğinde bu noktaya ait yükün indirilmesi için araç içerisinde yüklerin yeniden yerleştirmesi söz konusu olabilir. Kısıt (3.21), rota boyunca alt tur oluşumunu engelleyen akış kısıtıdır. (3.22) −

(3.27) kısıtları, rota boyunca aracın toplam yükünü hesaplayarak kapasitesinin aşılmamasını sağlamaktadır (Dethloff 2001, Guy Desaulniers 2002, Tasan ve Gen 2012). $(3.28)-(3.33)$ kısıtları sıralama kısıtlarıdır. Bu kısıtlar merkez depo noktasından başlanarak ziyaret edilen noktaların yalnızca biz kez ziyaret edilmelerini ve $b.$ sırada yalnızca bir noktanın ziyaret edilmesini sağlamaktadır. $(3.34)-(3.36)$ kısıtları, rota boyunca $n+j$ dağıtım noktasının j toplama noktasından önce ziyaret edilmesini önlemektedir. İlk sırada ziyaret edilen j toplama noktasının ardından $n+j$ dağıtım noktası ziyaret edilebilir (3.34) ya da j toplama noktasının ardından kendinden farklı bir toplama noktası ziyaret edilebilir (3.35). Kısıt (3.36) ise $b.$ sırada ziyaret edilen bir toplama noktasının dağıtım noktasına en erken $(b+1).$ sırada gidilebileceğini göstermektedir. $(3.37)-(3.40)$ kısıtları rotaya uygun bir şekilde yüklerin araca yüklenmelerini sağlamaktadır. Kısıt (3.37), toplam palet alanının araç alanından küçük olmasını sağlamaktadır. Kısıt (3.38), kısıt rotalama ve yükleme çiftinin bağlantısını sağlamaktadır. Kısıt (3.39), yükleme alanı içerisinde paletlerin herhangi bir kesişme yaşamadan yüklenmesini sağlamaktadır. Kısıt (3.40), talebin karşılanmasını sağlamaktadır. Kısıt $(3.41)-(3.42)$, ardışık olarak gidilen i ve j toplama noktalarına ait paletlerin araç içerisinde LIFO kısıtına uygun olarak yerleştirilmesini sağlamaktadır. Kısıt $(3.41)-(3.42)$, $R_{ijb}=1, (i,j \in P)$ ise; i noktasına ait paletin (x,y) sol alt köşe koordinatlarının j noktasına ait paletin sol alt köşe koordinatlarından küçük ya da eşit olmasını sağlamaktadır. Bu iki kısıt en son gidilen toplama noktasınına ait paletin daha önce gidilen noktalara ait paletlere göre aracın boşaltma doğrultusunda en önde olmasını sağlayarak LIFO kısıtına katkı sağlamaktadır.

3.4. Çözüm Yöntemi

Problem çözümü için önerilen matematiksel modelde LIFO prensibini kontrol eden bir kısıt bulunmamaktadır. Kısıt $(3.16)-(3.20)$, toplama ve dağıtım kümeleri ile ilişkilendirilmiştir. Bu kısıtlar LIFO kısıtı için gerekli olmakla beraber yeterli olmamaktadır. Ayrıca noktalar arası öncelik ilişkilerinin sağlanması için de yeni kısıtlar üretilmiştir $(3.28)-(3.36)$. Aşağıda öncelik kısıtları ile LIFO kısıtına yönelik yapılan çalışmalar ve LIFO kısıtının sağlanması amacıyla oluşturulan algoritma verilmiştir.

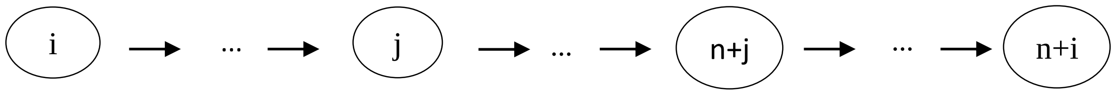
Dağıtım toplamalı araç rotalama probleminde rota, öncül-ardıl ve LIFO kısıtlarını sağlamalıdır. LIFO kısıtı altında araç rotalama problemi literatürüne bakıldığında bu iki kısıtın dikkate alındığı bir çok çalışma ile karşılaşılmaktadır (Bölüm 2.4). Bu çalışmalara örnek olarak; Cordeau (2006), Carrabs ve ark. (2007a), Cordeau ve ark. (2010), Côté ve ark. (2012a) ait çalışmalar verilebilir. Aşağıda verilen bilgiler bu çalışmalardan yola çıkılarak derlenmiştir.

Öncül-ardıl kısıtı; herhangi bir toplama noktasından gelen talebin bu noktanın dağıtım noktası ziyaret edilmeden karşılanmasını gerektirmektedir. LIFO kısıtı ise bir yük yığını dikkate alınarak aşağıdaki şekilde tanımlanabilir:

a) bir toplama noktası ziyaret edildiğinde alınan yük yığının üstüne yerleştirilir;
b) bir dağıtım noktasının ziyaret edilmesi, yalnızca o noktaya ait teslimat yükü yığının üstünde ise gerçekleştirilebilir (Cordeau ve ark. 2010).

LIFO kısıtı aynı zamanda yığın yükleme ve çok boyutlu yükleme problemlerinde de karşımıza çıkmaktadır. LIFO kısıtı araca yüklenmiş ürünlerin araç içinde taşınması ve boşaltılması sırasında herhangi bir olumsuzlukla karşılaşılmaması için önemlidir. Bu nedenle çalışmada ele alınan tümleşik probleme LIFO kısıtı eklenmiştir.

LIFO kısıtı altında dağıtım toplamalı gezgin satıcı problemlerinde, yükün yükleme ve boşaltılmasının LIFO isteğine uygun olarak yapılmasını gerektirmektedir. Örneğin; eğer i toplama noktası yüklemesi j toplama noktası yüklemesinden önce ise, j dağıtım noktası teslimatı mutlaka i dağıtım noktası teslimatından önce olmalıdır (Şekil 3.2). LIFO kısıtı altında gezgin satıcı probleminin araçla yük dağıtımı uygulamalarında yük için tek bir giriş ve çıkış noktası vardır ve bu nedenle yükün tekrar yerleştirilmesine izin verilmez. Bu durum; güvenlik, ürün fiziki durumları gerektirdiğinde (örneğin; ağırlık, kırılganlık, boyut) ya da müşteri lokasyonlarında servis süresinin azaltılması koşullarında karşımıza çıkabilmektedir (Carrabs ve ark. 2007a).



Şekil 3.2. LIFO kısıtı ve öncelik ilişkisi gösterimi

Yukarıda belirtildiği gibi LIFO kısıtı en son gidilmiş toplama noktasına ait yükün yığının üstünde ya da kamyonun yükleme-boşaltma doğrultusunda en önde olduğunu varsaymaktadır. Bu durumda hiçbir yer değiştirmeye gerek kalmadan en önde olan yük boşaltılabilir. LIFO kıstına geçmeden önce toplama ve dağıtma noktaları arasında öncelik ilişkilerine değinilecektir.

***Önerme* 1**

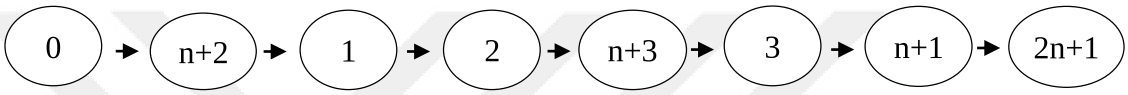
$n+i$ dağıtım noktası talebinin karşılanabilmesi için mutlaka i toplama noktasına gidilmiş olmalıdır.

İlişkili i ve $n+i$ noktaları için i noktasından alınan yükün $n+i$ noktasında boşaltılmalı öngörüsü geçerlidir. Bu nedenle i noktasına mutlaka $n+i$ noktasından önce gidilmelidir. Bu koşul montaj hattı problemlerinde ürünlerin öncelik ilişkilerine bağlı olarak makinelere atanmaları ile benzerlik göstermektedir. n iş sayısı, işler $\{J_1, J_2, \dots, J_n\}$, performans süreleri $p_1, p_2, \dots, p_n$ ve öncelik ilişkileri seti bulunmaktadır. Eğer J_u J_v'den önce ise alt simgeler $u < v$ şeklinde ifade edilir. X_{ij} karar değişkeni, J_j işi i $(i = 1,2, \dots, k)$ istasyonuna atanırsa 1 değerini alır. Öncelik ilişkisi şu şekilde ifade edilir:

$$X_{hv} \leq \sum_{i=1}^{h} X_{iu} \qquad h = 1,2, \dots, k; \quad (u, v) \in R \tag{3.43}$$

$R = \{(u, v) | \, J_u \text{ işi } J_v \text{ işinden hemen öncedir}\}$; öncelik ilişkileri matrisidir (Lynwood A. Johnson 1974). Kısıt (3.1), $u < v$ öncelik ilişkisine sahip iki iş için u işi v işinden önce atanmalıdır şeklinde açıklanabilir. Bu durum ilişkili toplama ve dağıtım noktaları için de geçerli olacaktır.

Kanıt Öncelik ilişkisi açısından, merkez depodan ilk olarak gidilecek nokta mutlaka bir j toplama noktası olmalıdır. j toplama noktasından sonra ikinci sırada gidilebilecek noktalar; $n+j$ dağıtım noktasına ya da j noktasından farklı bir q toplama noktasıdır (Kısıt (3.34) – (3.35)). Rota boyunca öncelik ilişkisini sağlayacak diğer bir diğer kısıt (3.36) gereğince, i noktasından j toplama noktasına $b.$ sırada gidiliyorsa, $n+j$ noktasına en erken $b+1.$ sırada gidilmelidir. Şekil 3.3 te, $n+2$ dağıtım noktasına birinci sırada gidilmiştir ve bu durum öncelik ilişkisine ters düşmektedir. Aynı şekilde $n+3$ dağıtım noktasına da ilişkili olduğu toplama noktasından önce gidilmiştir. Bu ihlalaller öncelik ilişkileri kısıtları ile ortadan kaldırılmaktadır.



Şekil 3.3. Öncelik ilişkileri ihlal edilmiş rota

LIFO kısıtının sağlanması rota boyunca gidilen ayrıt çiftlerinin olurlu olmasına bağlıdır. Buna göre; pp ayrıtı bir (i,j) ayrıtı olmak üzere iki toplama noktasını bağlar, pd ayrıtı bir $(i,n+i)$ ayrıtı olmak üzere bir toplama noktasını bir dağıtım noktasına bağlar ve dd ayrıtı bir $(n+i,n+j)$ ayrıtı olmak üzere iki dağıtım noktasını birbirine bağlar (Cordeau ve ark. 2010).

Cordeau ve ark. (2010)'na ait çalışmada ele alınan uyumsuz öncül ve ardıl eşitsizlikleri, hamburger eşitsizlikleri ve uyumsuz yol eşitsizliklerinden yola çıkılarak LIFO kısıtı aşağıdaki *Önerme* 2 ile yorumlanmıştır.

***Önerme* 2**

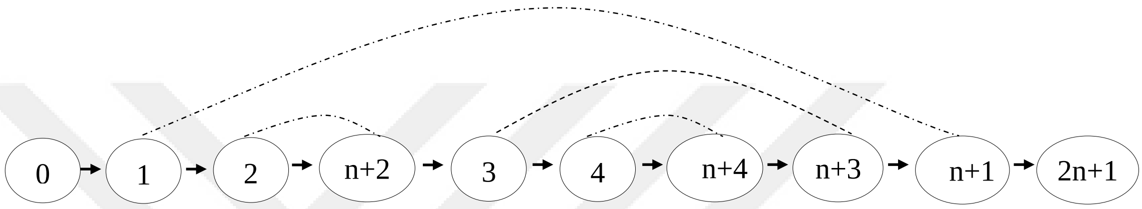
Herhangi bir rotanın LIFO kısıtını sağlaması için bir ayrıt oluşturan nokta çiftleri aşağıdaki şartları sağlamalıdır:

a) Herhangi bir i toplama noktasından sonra, rotada daha önce gidilmemiş herhangi bir j toplama noktasına gidilebilir.
b) Herhangi bir i toplama noktasından sonra, i toplama noktasının bağlı olduğu $n+i$ dağıtım noktasına gidilebilir.

c) Herhangi bir $n+i$ dağıtım noktasından sonra, (i toplama noktası hariç) rotada daha önce gidilmemiş herhangi bir j toplama noktasına gidilebilir.

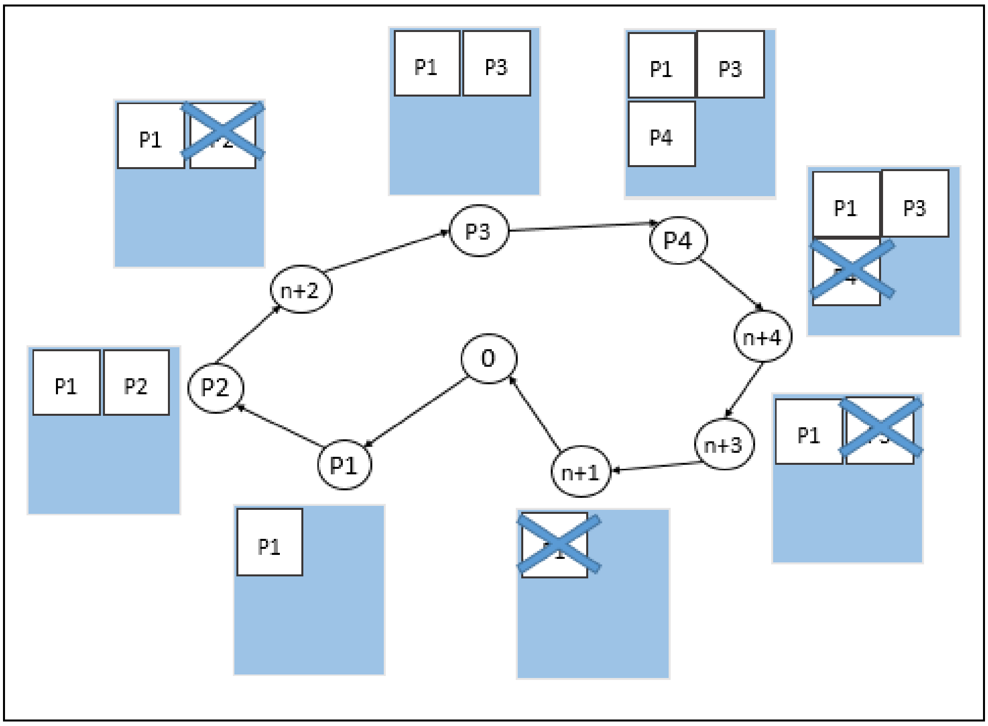
d) Herhangi bir $n+i$ dağıtım noktasından sonra, rotada i toplama noktasından önce ziyaret edilmiş - uygun - bir j toplama noktasının $n+j$ dağıtım noktasına gidilebilir.

LIFO kısıtına uygun rota ve yerleşim planı Şekil 3.4`de görülmektedir.



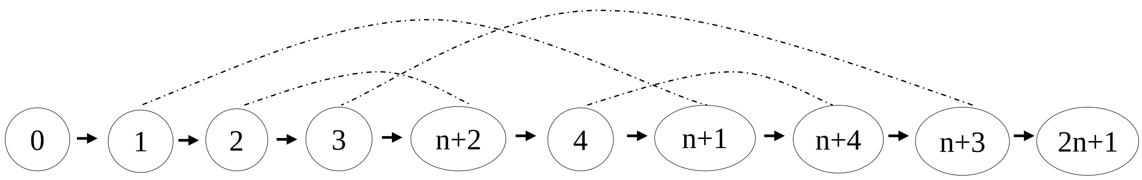
Şekil 3.4. LIFO kısıtına uygun rota örneği

Kanıt İlk üç maddeye uygun rota (Şekil 3.4.) ve yükleme planı (Şekil 3.5.) yukarıda verilmiştir. Fakat bazı durumlarda bu dört şartın sağlanması olurlu bir rota için yeterli olamayabilir. Şekil 3.6' da görüldüğü gibi 3. toplama noktasından sonra $n+2$ dağıtım noktasına, 4. toplama noktasından sonra $n+1$ dağıtım noktasına gidilmiş ve LIFO kısıtı ihlal edilmiştir. Kısıt (3.20), i bir toplama noktası olmak üzere kendi dağıtım noktası hariç diğer tüm dağıtım noktalarına gidişi engellemektedir. i noktasında araca yüklenen yük yığının en üstünde ya da kapıya en yakın sırada olacaktır. Bu nedenle herhangi bir yerdeğiştirme işlemi yapmamak için ya en son gidilen toplama noktasından farklı bir toplama nokasına, ya da bu noktaya ait dağıtım noktasına gidilmelidir.

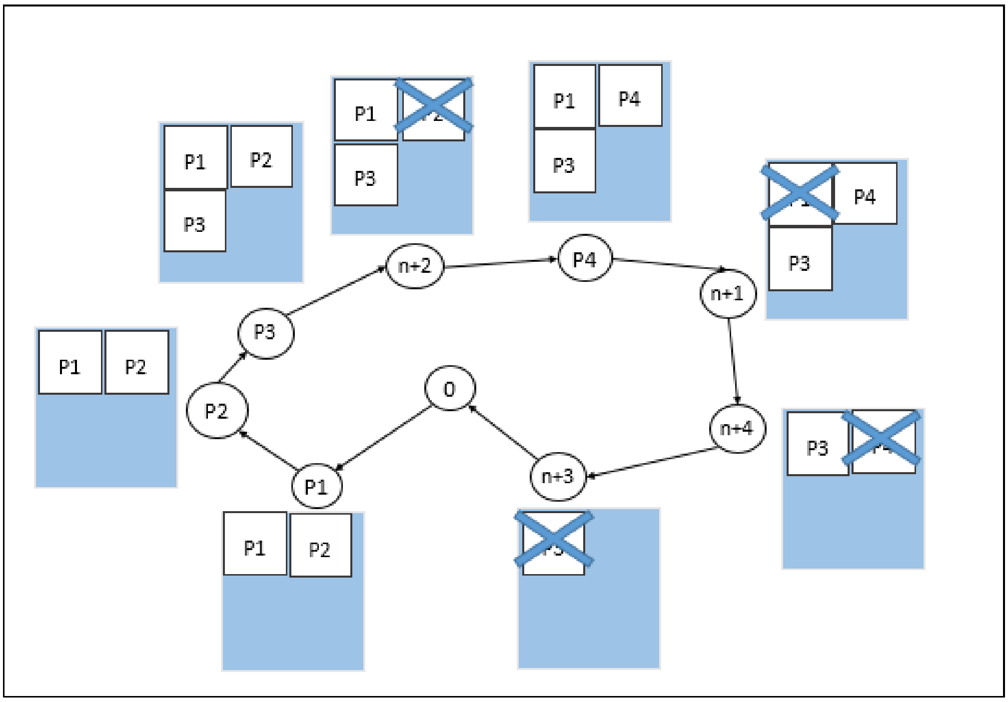


Şekil 3.5. LIFO kısıtına uygun yükleme planı

Şekil 3.7 ‘de probleme ait LIFO kısıtının ihlal edildiği yükleme planı verilmiştir. LIFO kısıtı yüklerin rota boyunca yer değiştirme işlemlerinin önlenmesi için oldukça önemlidir. O merkez depodan rotaya başlayan araç, uygun rota oluşturacak şekilde müşterilerin toplama taleplerini karşılamaya başlamıştır. Yükleme işlemi aracın sol ön köşesinden başlamak kaydıyla yapılmaktadır. P3 noktasından sonra gidilebilecek noktalar kümesi $Y \setminus \{P1, P2, P3\} \cup (n+3)$ ‘tür. Fakat çözümde görüldüğü gibi LIFO kısıtı çiğnenmiştir. Bu durumda $n+2$ noktasına gelindiğinde $P2$ noktasına ait paletin önünde herhangi bir yük olmadığı için herhangi bir yeniden yerleştirme durumu söz konusu değildir, fakat bu durum LIFO prensibine aykırıdır. Aynı şekilde $P4$ noktasından sonra $n+1$ noktasına gidilmesi LIFO prensibine aykırı bir durumdur. Şekil 3.7’ de görüldüğü gibi bu durumda $P3$ noktasına ait palet için yeniden yerleştirme gerekir.

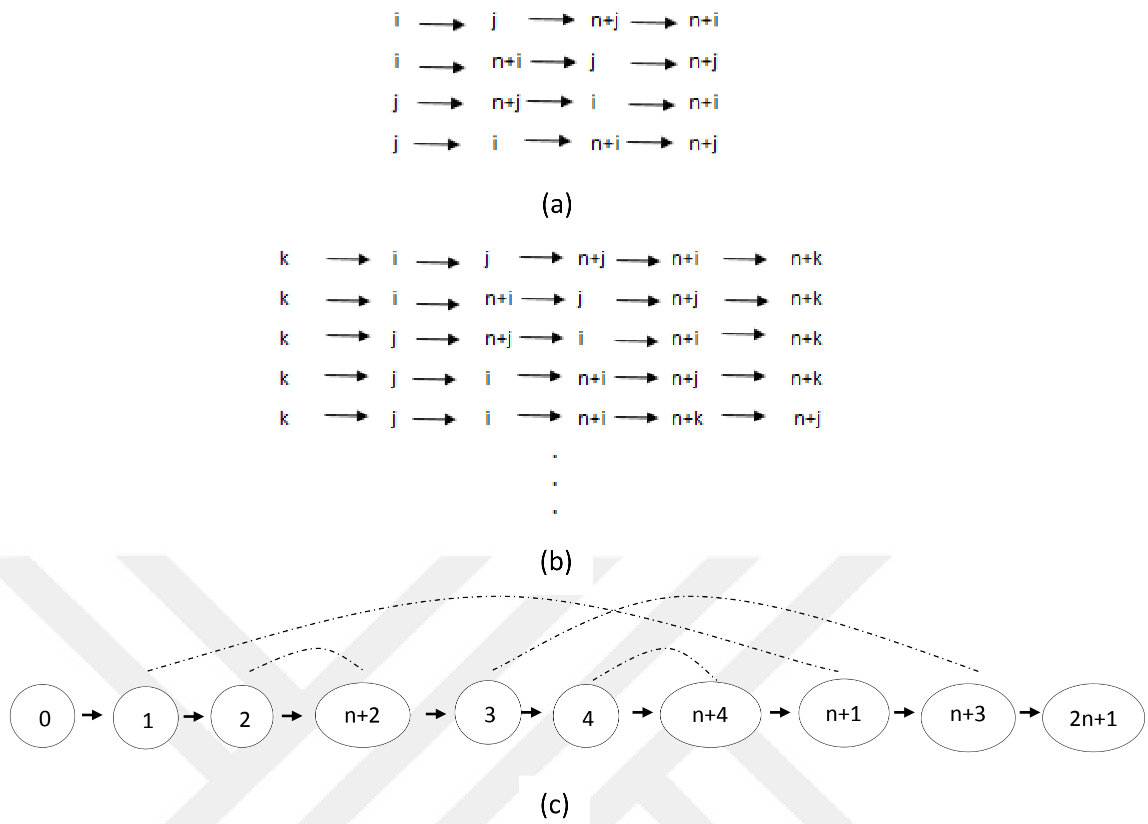


Şekil 3.6. LIFO kısıtının ihlal edildiği rota örneği



Şekil 3.7. LIFO kısıtının ihlal edildiği yükleme planı

Nokta sayısının arttığı problemlerde yukarıda belirtilen şartları sağladığı halde LIFO kısıtına uygun olmayan rota örnekleri ortaya çıkmaktadır (Şekil 3.8). i, j, k toplama noktaları olmak üzere $(3.20), (3.28) - (3.36)$ kısıtları göz önünde bulundurulduğunda oluşabilcek rota kombinasyonları aşağıda verilmiştir.



Şekil 3.8. Farklı toplama nokta sayıları için rota kombinasyonları (a) İki toplama noktası için rota kombinasyonları, (b) Üç toplama noktası için rota kombinasyonları, (c) Dört toplama noktasına ait LIFO kısıtına uygun olmayan rota örneği

Şekil 3.8 (a)'da toplama noktası sayısı iki olduğu durumda LIFO kısıtını ihlal eden herhangi bir rota oluşmamaktadır. Şekil 3.8 (b) ve (c)'de nokta sayısı sırasıyla üç ve dörde çıkarıldığında LIFO kısıtını ihlal eden rota kombinasyonuna rastlanabilmektedir. Ancak bu durum, matematiksel modelin kombinatorik yapısı nedeniyle karar değişkeninin almaması gereken değerin önceden bilinmesi ile engellenebilir. Gidilen nokta bir dağıtım noktası ise rota değişkeni bu toplama noktasına bir önceki hariç hangi sırada gelindiğini bilememektedir. Bu sebeple gidilen dağıtım noktası için LIFO kısıtı kontrol edilemez. Kısıtın sağlanması için oluşacak tüm rota kombinasyonlarında LIFO'yu engelleyen ayrıt çiftlerinin oluşumuna engel olmak gerekir. Ancak tüm kombinasyonların önceden belirlenmesi ve uygunsuz ayrıt çiftlerine karşı önceden kısıt geliştirerek önlem alınması matematiksel modelin aşırı derecede büyümesine sebep olacaktır. Literatüre bakıldığında LIFO kısıtını sağlamak amacıyla geliştirilen matematiksel ifadelerin çözüm aşamasında yerini sezgisel yöntemlere bıraktığı görülmektedir. Yukarıda LIFO kısıtını sağlamaya yönelik olarak verilen şartlar ve kısıtlar bir noktanın öncül ve ardıllarının kontrol edilmesini sağlamaktadır. Ancak gidilen bir toplama noktasına ait dağıtım noktasına rota boyunca hangi sırada gidileceği baştan kontrol edilememektedir. Şekil 3.8 (c)'de $n + 4$ dağıtım noktasının ardından gelebilecek noktalar kümesi; rotada daha önce gidilmemiş bir toplama noktası ya da $n + 3$ dağıtım noktası olabilir. Fakat 1. toplama noktasına 3. toplama noktasından daha önce gidilmiş olmasına rağmen, $n + 3$ dağıtım noktasının $n + 1$ dağıtım noktasından önce ziyaret edilmesi gerekmektedir. Şüphesiz LIFO kısıtsız probleme ait matematiksel modelin kombinasyonlarında bu çözüm mevcuttur. Fakat en iyi çözüm olmaması nedeniyle reddedilmiştir.

***Önerme* 3**

Olurlu tam sayı çözümü olan her problemde LIFO kısıtını sağlayan en az bir tam sayı çözüm mevcuttur.

Kanıt Dal sınır ve dal kesme algoritması, tamsayılı matematiksel programlama modellerinin kesin çözümünde kullanılan bir yöntemdir. Bu yöntem belirlenen amaca ulaşmak için tüm kombinasyonları kontrol etmektedir. Bu çalışmada ele alınan problemlerde de oluşturulacak tüm rota kombinasyonları arasında LIFO kısıtını sağlayan en az bir tam sayı çözüm bulunmaktadır. Örneğin üç toplama noktası için LIFO kısıtını sağlayan en basit rota $0 \to p1 \to d1 \to p2 \to d2 \to p3 \to d3 \to 2n + 1$ şeklindedir. Bu çözüm LIFO kısıtı altında iki boyutlu dağıtım-toplamalı araç rotalama problemi için olurlu bir çözümdür. Algoritma boyunca LIFO kısıtının sağlandığı farklı rota kombinasyonları bulunabilir.

Bu çalışmada, dal sınır algoritması boyunca LIFO kısıtına uygun rotayı bulmaya yönelik bir kontrol algoritması önerilmiştir. Bu algoritma çözüm sonrası kısıt kontrol prosedürü (post solution constraint checking procedure) olarak isimlendirilebilir. Önerilen algoritma matematiksel model ile eş zamanlı olarak çalışmaktadır. Problemin çözümünde kullanılan FICO Xpress Optimization Suite ortamında matematiksel model ve tümleşik algoritma yazımına elverişli Mosel dili ile problem çözümünde kolaylık sağlamaktadır. Önerilen algoritma tam sayı programlama modelinin çözümünden elde edilen rotanın kesme ekleme prosedürü içerisine kullanılmasını sağlamaktadır. Kesme düzlemi yöntemleri, tam sayı çözümlerin dış bükey çözüm uzayına yeni kısıtlar ekleyerek doğrusal gevşetilmiş çözümünün olurlu tam sayı çözüme yaklaşmasını ve gevşetilmiş çözüm tarafından üretilen sınır değerinin iyileştirilmesini sağlamaktadır (Anonim 2009), (https://examples.xpress.fico.com/example.pl?id=mosel_solv_2 2013).

Bir akış diyagramı ve uygulama adımları üzerinden bu tezde önerilen LIFO kısıt kontrol algoritması tanımlanmıştır. Bu algoritmanın Şekil 3.9'da verilen akış diyagramı aşağıdaki adımlarla açıklanabilir.

Adım 1 : Gevşetilmiş matematiksel modelin BC algoritmasına göre aday çözüm veren düğüm için modeli çöz.

Adım 1a : Rota değişkeni R_{ijb}; $\sum_{b \in O} R_{ijb} = K, \ \forall i, j$ iken $R_{ijb} = 1, \ \forall b$ koşulunu sağlamıyorsa Adım 1'e dön. (Bu özel durum, BC algoritmasından alınan gevşetilmiş çözümün $R_{ijb} \in \mathbb{R}$ karar değişkeni kullanımına bağlı olarak $1 - tolerance \leq R_{ijb} \leq 1 + tolerance$ şartına uymayan yuvarlama hatası olarak ortaya çıkabilir.)

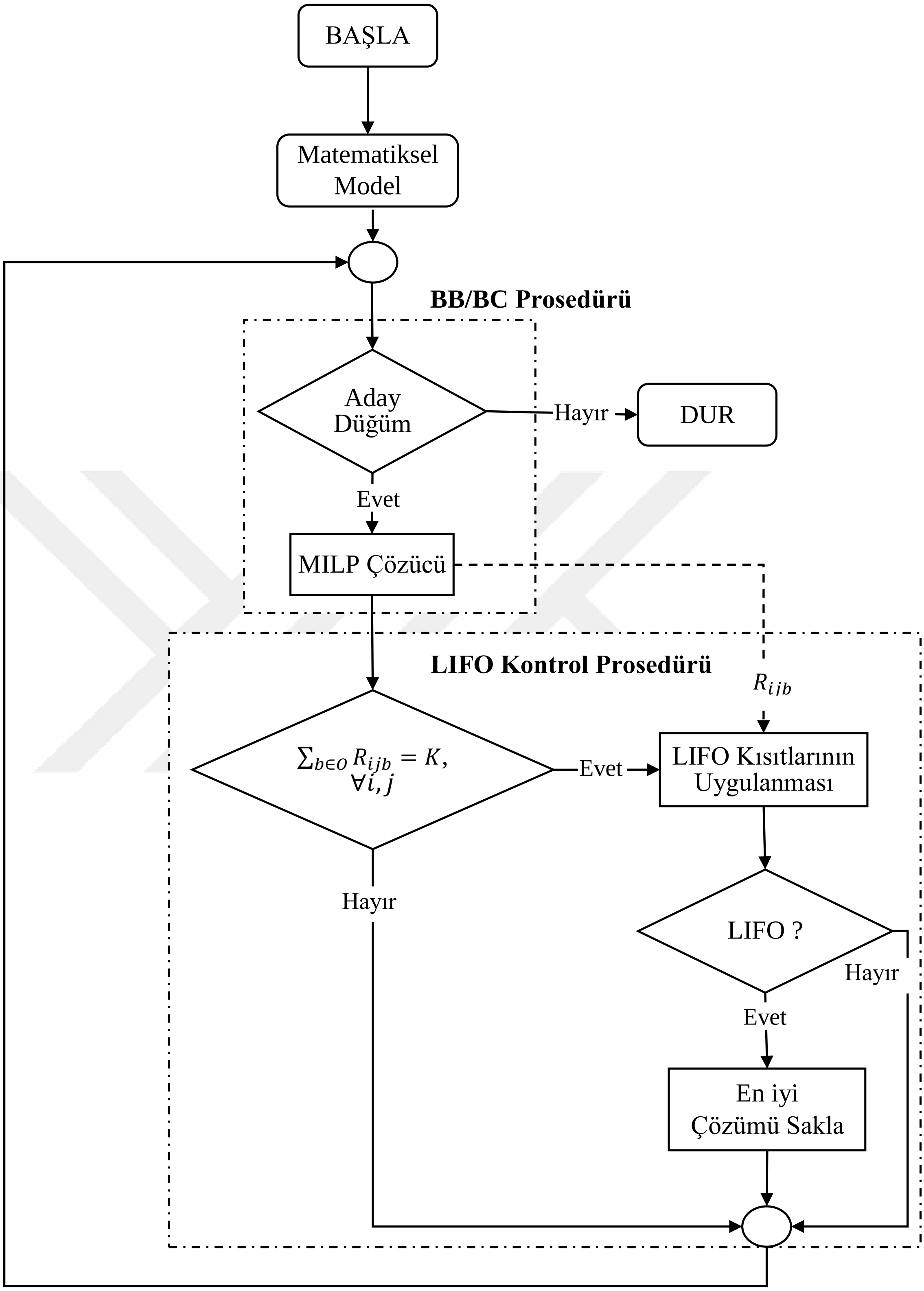
Adım 2 : $R_{ijb} = 1$ için b. sırada gidilecek nokta $p = j$ 'dir.

Adım 3 : p noktası için kontrol yapılır. p dağıtım noktası ise Adım 2'ye dön. p toplama noktası ise bulunan tamsayı çözümde ilişkili olduğu dağıtım noktasına kaçıncı sırada gidildiğini bul (h).

Adım 4 : Kontrol edilen tam sayı çözümde, $R_{pj(b+1)}$, b. sırada gidilen p noktasından $(b+1)$. sırada gidilecek noktayı bul. $p1 = j$'dir.

Adım 5 : $p1$ noktası için kontrol yapılır. $p1$ dağıtım noktası ise Adım 2'ye dön. $p1$ toplama noktası ise bulunan tamsayı çözümde ilişkili olduğu dağıtım noktasına kaçıncı sırada gidildiğini bul $(h1)$.

Adım 6 : $p, p1$ toplama noktası ve $h \leq h1$ ise çözüm LIFO kısıtını sağlamamaktadır; Adım 1'e dön. $h > h1$ ise çözüm LIFO kısıtını sağlamaktadır; Adım 2'ye dön.



Şekil 3.9. Algoritma akış diyagramı

Algoritma, matematiksel modelin gevşetilmiş çözümünün bulunmasıyla başlar ve dallanma boyunca çözüm üretir. Bulunan her çözüm için tamsayı R_{ijb} rota değişken kontrolü yapılır. Bunun amacı yalnızca K adet tamsayı çözüm bulunduğunda algoritmanın devam etmesini sağlamaktır. K depo noktası hariç toplam nokta sayısıdır. Bir tam sayı çözümde K kadar tam sayı değişken 1 değerini almalıdır. Tam sayı çözüm sayısı K değerine eşit değilse algoritma yeni bir çözüm üretmek için başa döner. Eğer K adet tam sayı çözüm bulunmuşsa kontrol prosedürü olarak adlandırılan aşamaya geçilir. Kontrol prosedürü bulunan tam sayı çözümün LIFO'ya uygunluğunu araştırmaktadır. Bunu yaparken ilk sırada gidilen noktadan başlayarak sırayla tüm noktaları dolaşır ve arka arkaya gidilen iki toplama noktası çifti için kontrol yapar. Bunun nedeni bu iki toplama noktasının dağıtım noktalarının öncelik ilişkilerinin kontrol edilmesidir. Örneğin; $i, b.$ sırada gidilen toplama noktası ve j, $(b+1).$ sırada gidilen toplama noktası olsun. $(i \rightarrow j)$ olduğunda, araç bu iki noktaya ardışık olarak gitmektedir. LIFO kısıtı gereği rotada ilk önce $n+j$ dağıtım noktasına, ardından da $n+i$ dağıtım noktasına gidilmelidir $(\dots n+j \rightarrow \cdots \rightarrow n+i \dots)$. LIFO kısıtı ancak ve ancak bu koşul ile sağlanabilir. Bu nedenle $n+j,\ n+i$ dağıtım noktalarına kaçıncı sırada gidildiği belirlenmelidir. Elde edilen tam sayı değişkenler kullanılarak rotada bu iki dağıtım noktasına gitme sırası bulunur. $(n+i).$ dağıtım noktasına $(h).$ sırada, $(n+j).$ dağıtım noktasına ise $(h').$ sırada gidiliyorsa $h > h'$ olmalıdır. Eğer $(i \rightarrow j)$ olduğu durumda $(\dots n+j \rightarrow \cdots \rightarrow n+i \dots)$ değil ise, yani $h' > h$ ise algoritma yeni bir tam sayı çözüm bulmak için Adım 1'e döner. Fakat koşul sağlandıysa $(h > h')$; tam sayı çözüm için oluşan rotada kalan noktaları algoritma kontrole devam eder. Herhangi bir uygunsuzluk bulunmadığı takdirde LIFO kısıtını sağlayan tam sayı çözüm kabul edilir ve Dal-Sınır ve Dal-Kesme algoritmasında tam sayı çözümler bitene kadar aynı kontrol algoritması adımları tekrar edilir. Algoritmaya ait sözde kod Şekil 3. 10'da verilmiştir.

Bölüm 4'te yukarıda bulunan matematiksel model ve kontrol algoritması için örnek bir problem verilmiştir. Örnek problem algoritma adımları boyunca incelenmiştir

```
1: Matematiksel modeli çöz, aday çözüm için;
2:     For b in O, i,j in V do
3:           If R_ijb = 1  Then
4:                 icount = 1 + icount
5:           Else Adım 1`e dön
6:           End If
7:     End For
8:     If icount = K Then
9:           For b in O Do
10:                For i,j in V Do
11:                      If R_ijb = 1 Then
12:                            p ← j
13:                      End If
14:                End For
15:                If p in P Then
16:                      For b′ in O b′ > b, k in V Do
17:                            If R_k(indis_p)b′ = 1 Then
18:                                  h ← b′
19:                      End If
20:                      End For
21:                End If
22:                For x in X, y in Y Do
23:                      If a_pbxy = 1 Then
24:                            Yaz,p noktasının konumu:x,y,
                               Yük miktarı:L(p)
25:                      End If
26:                End For
27:                For j in V, j <> p Do
28:                      If R_pj(b+1) = 1 Then
29:                            p1 ← j
30:                      End If
31:                End For
32:                If p1 in P Then
33:                      For b″ in O, b″ > b + 1, h1 <> H, k in V Do
34:                            If R_k(indis_p1)b″ = 1 Then
35:                                  h1 ← b″
36:                            End If
37:                      End For
38:                End If
39:                If H ≤ H1 and p1 in P and p1 in P Then
40:                      Yaz, LIFO sağlanmıyor!, Adım 1`e dön
41:                Break
42:                Else
43:                      Yaz, LIFO kısıtını sağlayan çözüm bulundu!,
                         Adım 1`e dön
44:                End If
45:          End For
46:    Tüm çözümler kontrol edildiyse
47:    End If
48:    DUR
```

Şekil 3.10. Kontrol algoritması sözde kodu

4. BULGULAR

Bu tez çalışmasında ele alınan optimizasyon probleminin çözümünde FICO Xpress Optimization Suite programı kullanılmıştır (http://www.fico.com/en/products/fico-xpress-optimization-suite 2017). FICO Xpress, büyük ve karmaşık optimizasyon problemlerinin modellemesi ve çözülmesine olanak tanıyan matematiksel modelleme ve optimizasyon paket programıdır. İçerisinde, Mosel olarak adlandırılan doğrusal ve karışık tamsayı programlama modellerinin tanımlanabildiği bir programlama dili üzerinden kod geliştirme ortamı ile doğrusal ve doğrusal olmayan modeller için çözüm algoritmaları bulunmaktadır. Model oluşturmak için hazırlanmış Mosel dili, cebirsel ve algoritmik tanımlamalarda kolaylık sağlamaktadır. Ayrıca bu yapı ile doğrusal ve doğrusal olmayan matematiksel modellerin çözümü yanında, kullanıcı tanımlı ek algoritmalar birlikte kullanılabilmektedir. Programın merkezinde doğrusal programlama, ikinci dereceden programlama ve karışık tamsayı programlama problemlerine çözüm metodu arayan ve geliştiren Xpress-Optimizer yapısı bulunmaktadır. Ayrıca bu yapı sayesinde zor problemlerin paralel olarak çözülmesi sağlanabilmekte ve matematiksel model ayrıştırma yöntemleri kullanılmaktadır. Bu çalışmada 64 bit Xpress 8.0 versiyonu kullanılmıştır. Matematiksel model ve kontrol algoritması Xpress-Mosel ile yazılmıştır. Programın matematiksel model ve algoritmaları birlikte çalıştırma özelliğinden faydanılarak Bölüm 3.3’te verilen matematiksel model ve Bölüm 3.4’te verilen kontrol algoritması birlikte kodlanmıştır. Kontrol algoritmasının başlangıç aşamasında Xpress’in içinde bulunan kesme üretme prosedürü kullanılmıştır. Problemin modellenmiş hali EK 1’de, çözümü ise EK 2’de verilmiştir.

4.1. Uygulama

4.1.1. Uygulama örneği

Model, örnek olarak ele alınan 7 toplama noktası için aşağıdaki şekilde uygulanmıştır. Merkez depodan harekete başlayan boş araç, rota boyunca yedi toplama noktasına uğrayarak bu noktaların toplama taleplerini karşılayacak ve ilgili dağıtım noktalarına teslimatı yapacaktır. Yükler, sabit en ve uzunluğa $(1 \times 1 \text{ m})$ sahip, paletler ile taşınmakadır. Paletlere önceden yükleme yapıldığı varsayılmaktadır. Palet alanı 1 m^2'dir. Uygulama örneğinde tek tip palet kullanıldığı için $(v = 1)$ Xpress-Mosel dili ile yazılmış moldelde palet inidisine yer verilmemiştir. Çizelge 4.1`de toplama ve dağıtım noktalarının talep miktarları, Çizelge 4.2`de araç ile ilgili bilgiler ve Çizelge 4.3`e maliyet verileri verilmiştir. Çalışmada aracın sabit maliyeti göz ardı edilmiştir.

Çizelge 4.1. Noktalara ait talep miktarları (kg)

Talepler			
Nokta	**Toplama Talebi**	**Nokta**	**DağıtımTalebi**
1	20	8	20
2	30	9	30
3	20	10	20
4	10	11	10
5	10	12	10
6	20	13	20
7	10	14	10

Çizelge 4.2. Örnek problem için araç verileri

Araç	**L (m)**	**W (m)**	**Alan (m²)**	**Ağırlık (kg)**
1	6	2	12	100

Çizelge 4.3. Örnek problem için maliyet verileri

								j								
i		**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
	0	-	70	112	82	56	128	122	76	86	64	48	202	144	89	24	-
	1	70	-	120	107	104	81	105	95	115	86	115	102	99	76	150	70
	2	112	120	-	149	136	133	110	134	124	144	138	125	122	154	160	112
	3	82	107	149	-	129	116	113	90	114	104	139	126	123	100	46	82
	4	56	104	136	129	-	115	102	99	76	81	105	95	115	86	180	56
	5	128	81	133	116	115	-	128	201	110	90	82	31	60	98	160	128
	6	122	105	110	113	102	128	-	37	90	45	88	205	129	67	70	122
	7	76	95	134	90	99	201	37	-	22	178	200	150	28	50	150	76
	8	86	115	124	114	76	110	90	22	-	48	100	81	44	84	240	86
	9	64	86	144	104	81	90	45	178	48	-	182	42	144	40	140	64
	10	48	115	138	139	105	82	88	200	100	182	-	99	93	130	230	48
	11	202	102	125	126	95	31	205	150	81	42	99	-	129	180	200	202
	12	144	99	122	123	115	60	129	28	44	144	93	129	-	144	120	144
	13	89	76	154	100	86	98	67	50	84	40	130	180	144	-	100	89
	14	24	150	160	46	180	160	70	150	240	140	230	200	120	100	-	24
	15	-	70	112	82	56	128	122	76	86	64	48	202	144	89	24	-

Ele alınan örnekte (toplama noktası sayısı) $n = 7$`dir. Bu toplama noktaları ile ilişkili toplam yedi adet dağıtım noktası bulunmaktadır. Her i toplama noktası $n + i$ dağıtım noktası ile ilişkilendirilmektedir.

Çözüm aşamasında öncelikle matematiksel model yardımıyla gevşetilmiş çözüm bulunur. Ardından FICO Xpress programı içerisinde bulunan tree cut generation (ağaç kesim üretimi) algoritması içerisine kodlanmış olan kontrol algoritması dallanma boyunca üretilen tamsayı çözümlerin LIFO kısıtına uygunluğunu araştırır.

Aşağıda LIFO kısıtına uymayan bir örnek çözüm Bölüm 3.4`te verilen algoritma adımları boyunca incelenmiştir. Öncelikle matematiksel modelin ilk çözümünden bir gevşetilmiş çözüm bulunur. Gevşetilmiş çözüm bulunduktan sonra dallanmaya geçilir.

Adım 1 : Gevşetilmiş matematiksel modelin BC algoritmasına göre göre aday çözüm veren düğüm için modeli çöz.

Adım 1a : Rota değişkeni R_{ijb}; $\sum_{b \in O} R_{ijb} = K, \ \forall i,j$ iken $R_{ijb} = 1, \ \forall b$ koşulunu sağlamıyorsa Adım 1'e dön. (Bu özel durum, BC algoritmasından alınan gevşetilmiş çözümün $R_{ijb} \in \mathbb{R}$ karar değişkeni kullanımına bağlı olarak $(1 - tolerance \leq R_{ijb} \leq 1 + tolerance)$ şartına uymayan yuvarlama olarak ortaya çıkabilir.)

Adım 2 : $R_{ijb} = 1$ için b. sırada gidilecek nokta $p = j$`dir.

b, 1`den başlamak üzere sırayla tüm rotada gidilen noktalar bulunur. $b$. sırada gidilen nokta $p$`dir. Örneğin; ele alınan uygulama örneğinde $K = 15$`tir. Adım 1a`da verilen kontrol yapılır. $K = 15$ olduğu durum için, LIFO kısıtı kontrol edilecektir.

$R_{071} = 1, \ b = 1$. sırada gidilen nokta; 7. toplama noktasıdır.

Adım 3 : p noktası için kontrol yapılır. p dağıtım noktası ise Adım 2`ye dön. p toplama noktası ise bulunan tamsayı çözümde ilişkili olduğu dağıtım noktasına kaçıncı sırada gidildiğini bul (h).

$b = 1$. sırada gidilen nokta; 7. toplama noktasının ilişkili olduğu dağıtım noktası $n + 7$. dağıtım noktası yani 14. noktadır. 14. dağıtım noktasına $h = 6$. sırada gidilmiştir.

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

Adım 4 : Kontrol edilen tam sayı çözümde, $R_{pj(b+1)}$, b. sırada gidilen p noktasından $(b+1)$. sırada gidilecek noktayı bul. $p1 = j$'dir.

$b + 1 = 2$. sırada gidilen nokta; 2. toplama noktasıdır. Şu ana kadar elde edilen rota:

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

Adım 5 : $p1$ noktası için kontrol yapılır. $p1$ dağıtım noktası ise Adım 2`ye dön. $p1$ toplama noktası ise bulunan tamsayı çözümde ilişkili olduğu dağıtım noktasına kaçıncı sırada gidildiğini bul $(h1)$.

$b + 1 = 2$. sırada gidilen nokta; 2. toplama noktasının ilişkili olduğu dağıtım noktası $n + 2$. dağıtım noktası yani 9. dağıtım noktasıdır. 9. dağıtım noktasına $h1 = 11$. sırada gidilmiştir. Algoritmada ardışık olarak gidilen toplama noktaları için LIFO uygunluk kontrolü yapılmaktadır. LIFO kısıtı gereği i`den $j$`ye gidiliyorsa $n + j$`ye $n + i$`den önce gidilmelidir.

Adım 6 : $p, p1$ toplama noktası ve $h \leq h1$ ise çözüm LIFO kısıtını sağlamamaktadır; Adım 1'e dön. $h > h1$ ise çözüm LIFO kısıtını sağlamaktadır; Adım 2`ye dön.

Model dal-sınır algoritması en iyi çözümüne ulaşana kadar bu adımları tekrarlamaktadır.

i	~~7~~	~~1~~				~~14~~					~~9~~				
b	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**

$b, b + 1$ sıralarında gidilen iki nokta p ve $p1$ için bu koşul sağlanamamaktadır. Bulunan tamsayı çözüm LIFO kısıtına uygun değildir. Algoritma bu aşamada yeni bir aday çözüm bulmak için başa döner.

Yukarıda verilen örnek çözümde LIFO kısıtının ihlal edildiği görülmektedir. Yedi toplama noktası için verilen problem verileri kullanılarak çalıştırılan modele ait algoritma adımları ve sonuçlar aşağıda verilmiştir.

Gevşetilmiş amaç fonksiyonu değeri: 1 045,90

Adım 1 : Gevşetilmiş matematiksel modelin BC algoritmasına göre aday çözüm veren düğüm için modeli çöz.

Adım 1a : Rota değişkeni R_{ijb}; $\sum_{b \in O} R_{ijb} = K,\ \forall i, j$ iken $R_{ijb} = 1,\ \forall b$ koşulunu sağlamıyorsa Adım 1`e dön. (Bu özel durum, BC algoritmasından alınan gevşetilmiş çözümün $R_{ijb} \in \mathbb{R}$ karar değişkeni kullanımına bağlı olarak $1 - tolerance \leq R_{ijb} \leq 1 + tolerance$ şartına uymayan yuvarlama hatası olarak ortaya çıkabilir.)

Adım 2 : $R_{ijb} = 1$ için b. sırada gidilecek nokta $p = j$ 'dir.

b, 1`den başlama üzere sırayla tüm rotada gidilen noktalar bulunur. $b$. sırada gidilen nokta $p$`dir. Örneğin; ele alınan uygulama örneğinde $K = 15$`tir. Adım 1a `te verilen kontrol yapılır. $K = 15$ olduğu durum için, LIFO kısıtı kontrol edilecektir.

$b = 1$. sırada gidilen nokta; 1. toplama noktasıdır.

Adım 3 : p noktası için kontrol yapılır. p dağıtım noktası ise Adım 2`ye dön. p toplama noktası ise bulunan tamsayı çözümde ilişkili olduğu dağıtım noktasına kaçıncı sırada gidildiğini bul (h). (Bu adımda p1 noktasında yüklenen paletin x, y koordinatları ve aracın bu noktadan çıktığı andaki toplam yükü hesaplanmaktadır.)

$b = 1$. sırada gidilen nokta; 1. toplama noktasının ilişkili olduğu dağıtım noktası $n + 1$. dağıtım noktası yani 8. noktadır. 8. dağıtım noktasına $h = 10$. sırada gidilmiştir.

1. sırada gidilen 1. toplama noktasına ait paletin koordinatları ; $x{:}\,0, y{:}\,0$`dır. Ayrıca aracın bu noktadan ayrıldıktan sonraki yükü $L_1 = 20`ir$.

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

Adım 4 : Kontrol edilen tam sayı çözümde, $R_{pj(b+1)}$, b. sırada gidilen p noktasından $(b + 1)$. sırada gidilecek noktayı bul. $p1 = j$`dir.

$b + 1 = 2$. sırada gidilen nokta; 2. toplama noktasıdır. Şu ana kadar elde edilen rota:

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

Adım 5 : $p1$ noktası için kontrol yapılır. $p1$ dağıtım noktası ise Adım 2`ye dön. $p1$ toplama noktası ise bulunan tamsayı çözümde ilişkili olduğu dağıtım noktasına kaçıncı sırada gidildiğini bul $(h1)$. (Bu adımda $p1$ noktasında yüklenen paletin x, y koordinatları ve aracın bu noktadan çıktığı andaki toplam yükü hesaplanmaktadır.)

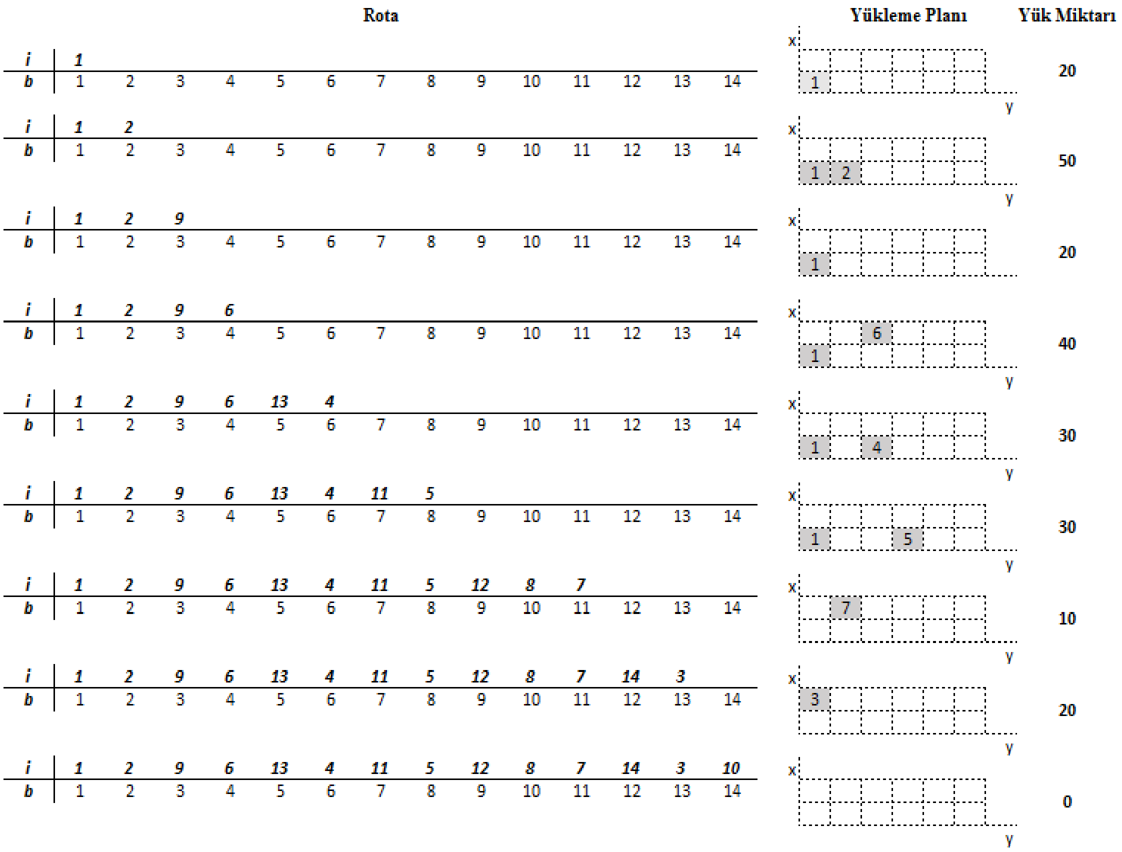
$b + 1 = 2$. sırada gidilen nokta; 2. toplama noktasının ilişkili olduğu dağıtım noktası $n + 2$. dağıtım noktası yani 9. dağıtım noktasıdır. 9. dağıtım noktasına $h1 = 3$. sırada gidilmiştir. Algoritmada ardışık olarak gidilen toplama noktaları için LIFO uygunluk kontrolü yapılmaktadır. LIFO kısıtı gereği i`den $j$`ye gidiliyorsa $n + j$`ye $n + i$`den önce gidilmelidir.

2. sırada gidilen 2. toplama noktasına ait paletin koordinatları; x: 0, y: 3`tür. Ayrıca aracın bu noktadan ayrıldıktan sonraki yükü $L_2 = 50$ ’ dir.

Adım 6 : $p, p1$ toplama noktası ve $h \leq h1$ ise çözüm LIFO kısıtını sağlamamaktadır; Adım 1’e dön. $h > h1$ ise çözüm LIFO kısıtını sağlamaktadır; Adım 2`ye dön

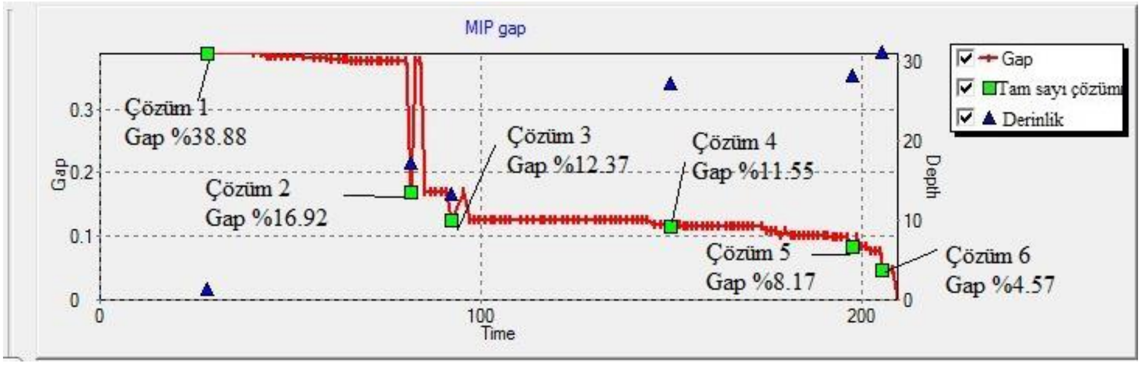
i	1	2	9							8					
b	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**

Kontrol edilen tamsayı çözümde ilk iki sırada gidilen noktalar için LIFO kısıtı sağlanmaktadır. Algoritma LIFO kısıtına aykırı herhangi bir dizilişe rastlamadığı sürece bu tamsayı çözüm için tüm rotayı kontrol eder. İncelenen tamsayı çözümün diğer noktalarıda kontrol edilecektir. Bu tamsayı çözüm örnek problem için LIFO kısıtını sağlayan en iyi çözümdür. $Z = 1\,179$’dur. Çözüme ait rota ve yükleme planı Şekil 4.1’de verilmiştir. Şekilde sırayla gidilen tüm toplama noktaları için yükleme planları görülmektedir.



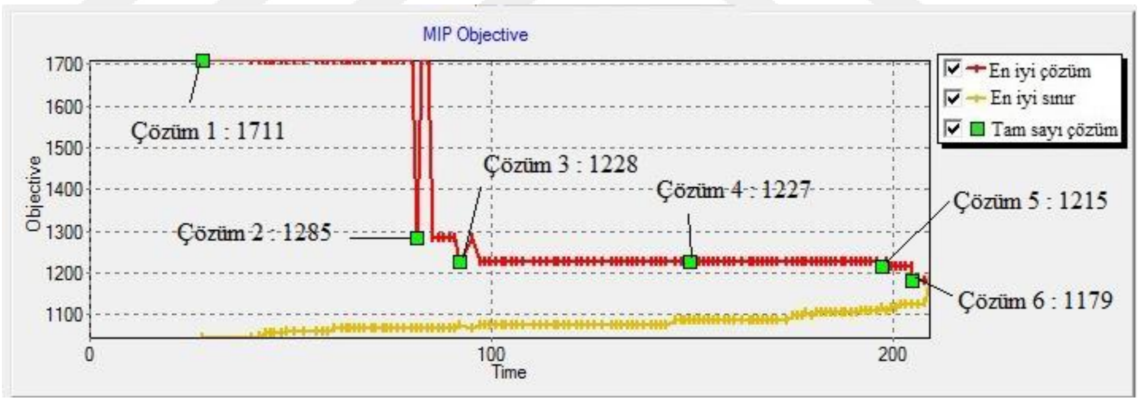
Şekil 4.1. Örnek problem rota ve yükleme planı

Örnek problem için altı tam sayı çözüm bulunmuştur. Bulunan çözümlerden yalnızca en iyi sonuç LIFO kısıtına uygundur.



Şekil 4.2. Örnek problem % fark grafiği

Dal sınır algoritması boyunca o ana kadar bulunmuş en iyi aday çözüm ve her düğümde elde edilen gevşetilmiş çözüm arasındaki % fark (aday çözüm − gevşetilmiş çözüm = gap) değeridir. Şekil 4.2`te örnek probleme ait % fark değerinin zamana göre değişimi görülmektedir. Şekil 4.3`te ise mevcut en iyi tamsayı çözümün en iyi sınır değeri ile bağlantısını göstermektedir.



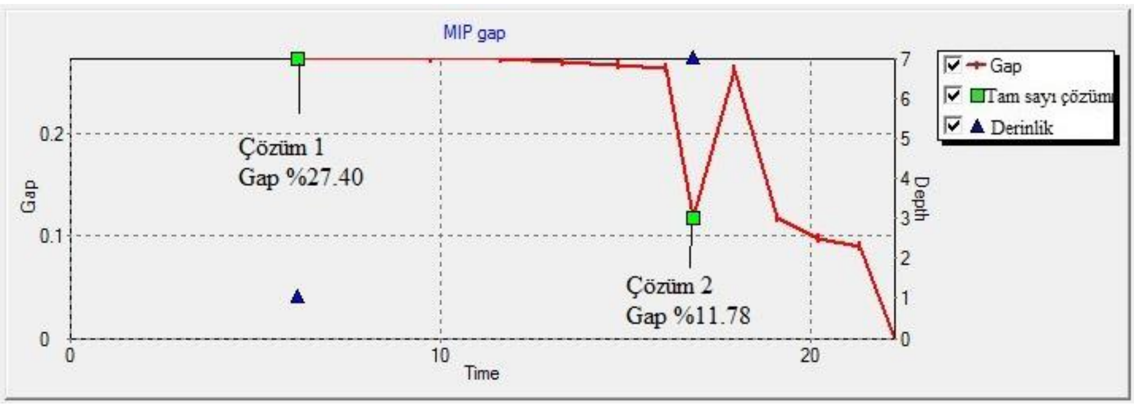
Şekil 4.3. Örnek problem amaç fonksiyonu değişimi

4.1.2. Karşılaştırmalı sonuçlar

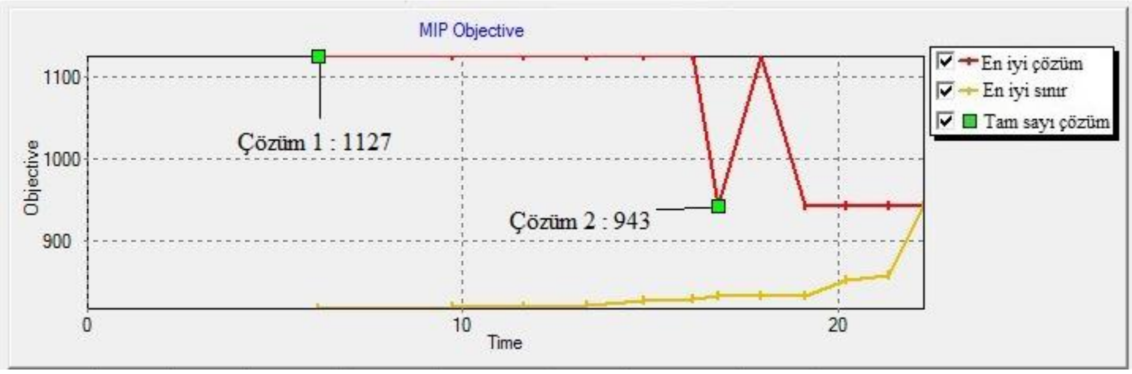
Dağıtım toplamalı araç rotalama probleminin iki boyutlu yükleme ve LIFO kısıtı altında çözmüne yönelik olarak önerilen karışık tam sayılı matematiksel model ve kontrol prosedürü farklı toplama noktaları için test edilmiştir. Çizelge 4.4`te farklı toplama noktaları için amaç fonksiyonu değerileri, problem çözüm süreleri ve LIFO kısıtını sağlayan en iyi tam sayı çözüm için gap oranları verilmiştir. Toplama noktası sayısı dokuzdan fazla olduğu durumda bellek yetersizliği nedeniyle çözüme ulaşılamamıştır. Çizelge 4.4`te verilen toplama noktaları için araç alan ve yük kapasitesi ile palet alanı aynıdır. Noktalara ait maliyet ve talep verileri EK 3`te verilmiştir.

Çizelge 4.4. Farklı toplama noktaları için problem sonuçları

Nokta Sayısı (Toplama Noktası)	Amaç Fonksiyonu	Süre (sn)	GAP (%)	Satır Sayısı	Sütun Sayısı
5	943	22	%11,78	103 667	4 225
6	1 052	86	%10.47	186 263	6 189
7	1 179	204	%4,57	303 873	8 641
8	1 215	627	%9.05	462 665	11 629
9	1 301	4 231	%9.92	668 807	15 201

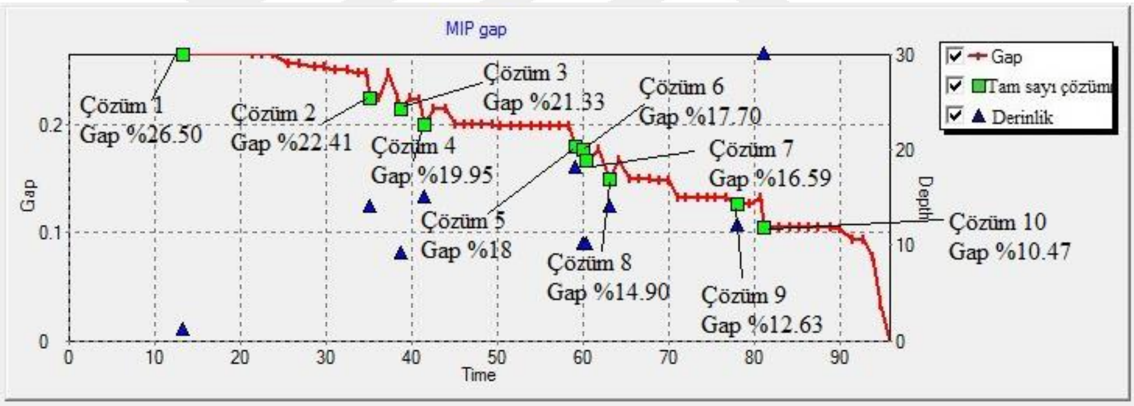


Şekil 4.4. Örnek problem beş toplama noktası % fark grafiği

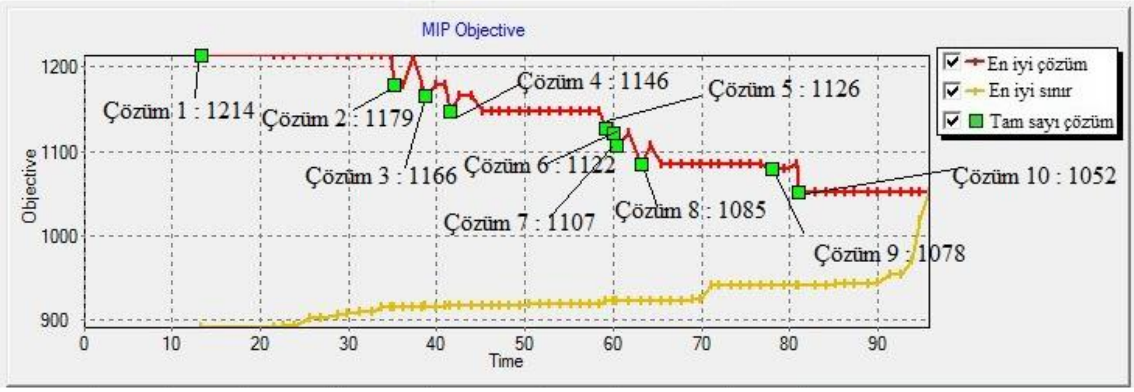


Şekil 4.5. Örnek problem beş toplama noktası amaç fonksiyonu değişimi

Dal sınır algoritması boyunca beş toplama noktası için iki adet tam sayı çözüm bulunmuştur. Bu tam sayı çözümlerden yalnızca Çözüm 2 LIFO kısıtını sağlamaktadır. Ayrıca bu çözüm en iyi çözümdür.

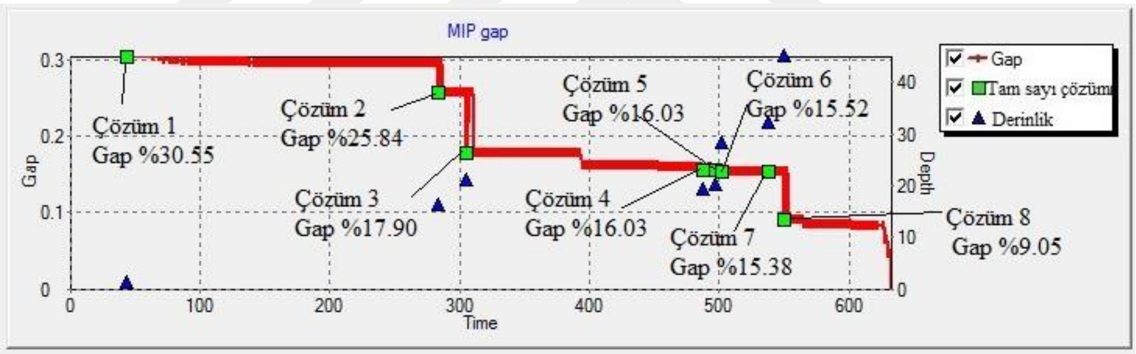


Şekil 4.6. Örnek problem altı toplama noktası % fark grafiği

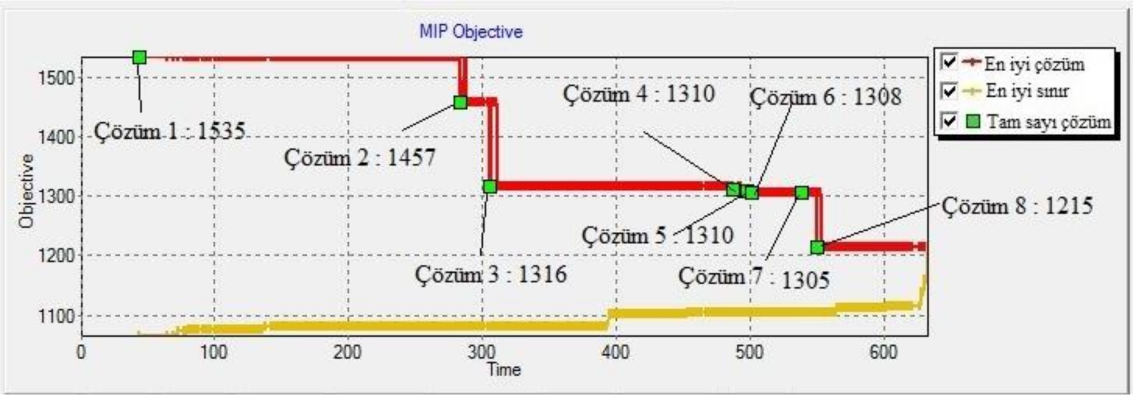


Şekil 4.7. Örnek problem altı toplama noktası amaç fonksiyonu değişimi

Dal sınır algoritması boyunca altı toplama noktası için on adet tam sayı çözüm bulunmuştur. Bu tam sayı çözümlerden Çözüm 7 ve Çözüm 10 LIFO kısıtını sağlamaktadır. Ayrıca Çözüm 10 çözüm en iyi çözümdür.

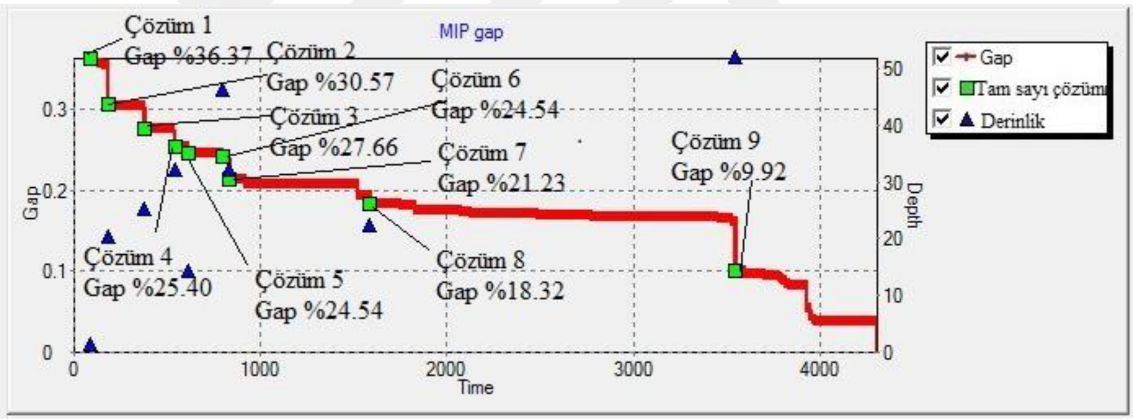


Şekil 4.8. Örnek problem sekiz toplama noktası % fark grafiği

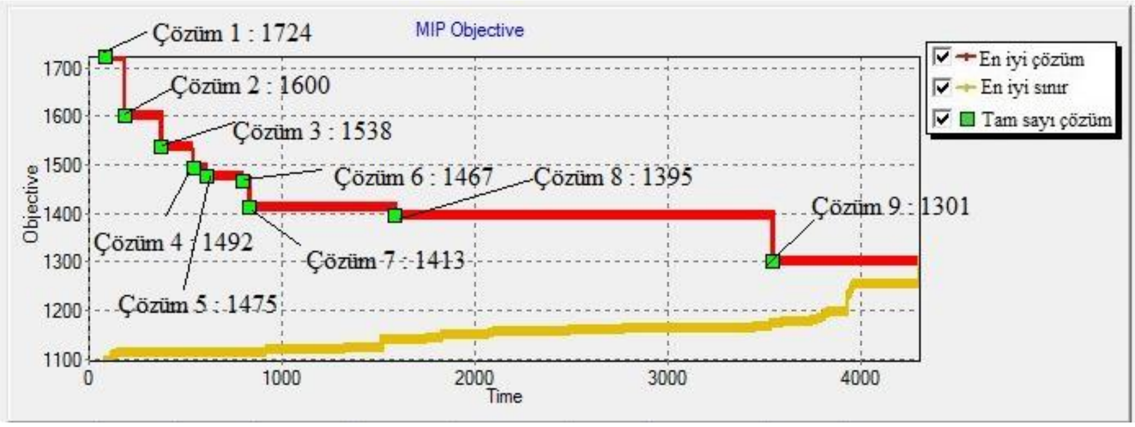


Şekil 4.9. Örnek problem sekiz toplama noktası amaç fonksiyonu değişimi

Dal sınır algoritması boyunca sekiz toplama noktası için sekiz adet tam sayı çözüm bulunmuştur. Bu tam sayı çözümlerden Çözüm 7 ve Çözüm 8 LIFO kısıtını sağlamaktadır. Ayrıca Çözüm 8 en iyi çözümdür.



Şekil 4.10. Örnek problem dokuz toplama noktası % fark grafiği



Şekil 4.11. Örnek problem dokuz toplama noktası amaç fonksiyonu değişimi

Dal sınır algoritması boyunca dokuz toplama noktası için dokuz adet tam sayı çözüm bulunmuştur. Bu tam sayı çözümlerden Çözüm 3, Çözüm 5 ve Çözüm 9 LIFO kısıtını sağlamaktadır. Ayrıca Çözüm 9 en iyi çözümdür.

4.2. Bulgular

Dağıtım toplamalı araç rotalama probleminin iki boyutlu yükleme kısıtı altında çözülmesine yönelik olarak bu tez çalışmasında önerilen kesin çözüm yöntemi, farklı toplama noktaları için test edilmiştir. Ortaya çıkan sonuçlarda toplama noktası sayısı arttıkça çözüm süresinin uzadığı görülmektedir. Araç rotalama problemi NP-Zor yapıdadır. Nokta sayısının artması problemin karmaşıklığını artırmaktadır. Bu nedenle toplama noktası sayısı artıkça problemin çözüm süresi de artış göstermektedir.

Matematiksel model ve kontrol algoritması farklı sayıda toplama noktası içeren problemler için test edilmiştir. Toplama noktası sayısının dokuzu aştığı durumlarda kullanılan belleğin yetersiz kalması nedeniyle çözüm elde edilememiştir. Nokta sayısının artmasıyla önerilen matematiksel model kısıtlarına bağlı olarak üretilen kombinasyon sayısı gittikçe artmaktadır.

5. TARTIŞMA VE SONUÇ

Bu tez çalışması kapsamında araç rotalama problemi, yükleme kısıtı ve dağıtım toplamalı araç rotalama problemi tümleşik olarak ele alınmıştır. Tümleşik probleme yönelik olarak geliştirilen matematiksel model, LIFO kısıtının kontrol edildiği bir kontrol prosedürü ile birlikte kullanılmıştır.

Problem, dağıtım toplamalı araç rotalama ve yükleme problemlerini içerdiğinden NP-Zor yapıdadır. Problemin incelenmesi ve zorluğunun anlaşılması için öncelike araç rotalama problemi ve türleri incelenmiştir. Ardından tümleşik probleme yönelik olarak dağıtım toplamalı araç rotalama problemi ve yükleme kısıtı altında araç rotalama problemi ayrıntılı olarak incelenmiştir. Bu aşamada tümleşik probleme etki eden önemli bir faktör LIFO kısıtıdır. Lojistik faaliyetlerinde ürünlerin hasar görmesini önlemek, daha hızlı ve kaliteli hizmet sağlayabilmek için LIFO kısıtı kuşkusuz önem arz etmektedir. Araç rotalama problemleri içerisinde LIFO kısıtına yönelik olarak yapılan çalışmalar da bu problem kapsamında incelenmiştir. Yapılan literatür araştırması sonucunda problemin zorluk derecesinden kaynaklı olarak iki boyutlu yükleme kısıtı altında dağıtım toplamalı araç rotalama problemine yönelik kesin çözüm yaklaşımına rastlanmamıştır. Bu doğrultuda iki boyutlu yükleme kısıtı altında araç rotalama problemi karışık tam sayılı programlama yöntemi kullanılarak modellenmiştir. Fakat oluşturulan matematiksel model tam anlamıyla LIFO gerek ve yeter şartlarını sağlayamamaktadır. Bu nedenle FICO Xpress programının matematiksel model ve algoritmaları tümleşik bir yapı içerisinde çözmesinden yararlanılarak, LIFO kısıtını problem çözüm aşamasında kontrol eden bir "kontrol algoritması" geliştirilmiştir. Ortaya konan bu çözüm yöntemi sayesinde problemin kesin çözümüne ulaşılmıştır. Model farklı veri setleri üzerinde uygulanmış ve doğruluğu test edilmiştir. Problemin çözümünde dal-sınır ve dal-kesme yaklaşımı ile çalışan Xpress Optimization Suite programı kullanılmıştır.

Kullanılan yöntem diğer sezgisel yöntemlerle karşılaştırıldığında çok büyük boyuttaki problemlerin çözümünde yetersiz kalabilmektedir. Fakat daha önce iki boyutlu yükleme ve dağıtım toplamalı araç rotalama problemine yönelik kesin çözüm yaklaşımı geliştiren bir çalışmaya rastlanamamıştır. Bu çalışma bu tür zor problemlerin modellenmesi ve çözümü için motivasyon kaynağı olabilir. Ek olarak yükleme probleminin karmaşıklığı ve araç içerisinde rota boyunca yük kontrolünün doğru bir şekilde yapılması amacıyla yükleme kısıtı iki boyutlu olarak alınmıştır. Paletlerin araç içerisine yerleştirilmesinde araç hacminin en iyi şeklide kullanılması için matematiksel model amaç fonksiyonu ve yükleme kısıtlarında iyileştirmeler yapılabilir. Ayırca rota boyunca bir noktadan toplama ve bir noktaya dağıtım işlemlerinin yapıldığı varsayılmaktadır. İleriki çalışmalarda üç boyutlu yükleme kısıtı, eş zamanlı dağıtım toplamlı ya da bölünmüş toplama/dağıtım taleplerinin bulunduğu yeni örnekler çalışılabilir.

KAYNAKLAR

Anonim, 2009. FICO Xpress Optimization Suite Xpress-Mosel User Guide. https://www.google.com.tr/url?sa=t&rct=j&q=&esrc=s&source=web&cd=7&ved=0ahUKEwj689LE4LDSAhXqKJoKHS86DOoQFghRMAY&url=https%3A%2F%2Fwww.researchgate.net%2Ffile.PostFileLoader.html%3Fid%3D55d330545cd9e336688b45f6%26assetKey%3DAS%253A273834266038272%25401442298584272&usg=AFQjCNGDHHX8vpiompGa88XyLTY8UsSMoQ&sig2=ABz0ubyEeITTQN0Jb0TMyg&bvm=bv.148073327,d.bGs (Erişim Tarihi: 27.02.2017).

Aprile, D., Egeblad, J., Garavelli, A., Lisi, S., Pisinger, D. 2007. Logistics optimization: vehicle routing with loading constraints. ICPR-19th International Conference on Production Research, 29 July- 2 August 2007, Valparaiso, Chile.

Bartók, T., Imreh, C. 2011. Pickup and delivery vehicle routing with multidimensional loading constraints. *Acta Cybernetica,* 20(1): 17-33.

Batista-Galván, M., Riera-Ledesma, J., Salazar-González, J. J. 2013. The traveling purchaser problem, with multiple stacks and deliveries: A branch-and-cut approach. *Computers & Operations Research,* 40(8): 2103-2115.

Beasley, J. 1985. An exact two-dimensional non-guillotine cutting tree search procedure.*Operations Research,* 33(1): 49-64.

Benavent, E., Landete, M., Mota, E., Tirado, G. 2015. The multiple vehicle pickup and delivery problem with LIFO constraints. *European Journal Of Operational Research,* 243(3): 752-762.

Bin, W., Hong, C., Zhi-yong, C. 2013. Artificial bee colony algorithm for two-dimensional loading capacitated vehicle routing problem. In: International Conference on Management Science and Engineering (ICMSE), IEEE, 406-412.

Bortfeldt, A. 2012. A hybrid algorithm for the capacitated vehicle routing problem with three-dimensional loading constraints. *Computers & Operations Research*, 39(9): 2248-2257.

Bortfeldt, A., Homberger, J. 2013. Packing first, routing second—a heuristic for the vehicle routing and loading problem. *Computers & Operations Research,* 40(3): 873-885.

Bortfeldt, A., Wäscher, G. 2013. Constraints in container loading–A state-of-the-art review. *European Journal of Operational Research,* 229(1): 1-20.

Carrabs, F., Cerulli, R., Cordeau, J.-F. 2007a. An additive branch-and-bound algorithm for the pickup and delivery traveling salesman problem with LIFO or FIFO loading. *INFOR: Information Systems and Operational Research,* 45(4): 223-238.

Carrabs, F., Cordeau, J.-F., Laporte, G. 2007b. Variable neighborhood search for the pickup and delivery traveling salesman problem with LIFO loading. *INFORMS Journal on Computing,* 19(4): 618-632.

Casco, D., Golden, B., Wasil, E. 1988. Vehicle Routing With Backhauls: Models, Algorithms, and Case Studies. Vehicle Routing : Methods and Studies. Studies in Management Science and Systems. *Publication of: Dalctraf,* 16.

Ceschia, S., Schaerf, A., Stützle, T. 2013. Local search techniques for a routing-packing problem. *Computers & Industrial Engineering,* 66(4): 1138-1149.

Chang, T.-S., Liao, Y.-F. 2008. Path finding with stowage planning consideration in a mixed pickup–delivery and specified-node network. *Transportation Research Part E: Logistics and Transportation Review,* 44(6): 970-985.

Cheang, B., Gao, X., Lim, A., Qin, H., Zhu, W. 2012. Multiple pickup and delivery traveling salesman problem with last-in-first-out loading and distance constraints. *European Journal Of Operational Research,* 223(1): 60-75.

Chen, C., Lee, S.-M., Shen, Q. 1995. An analytical model for the container loading problem. *European Journal Of Operational Research,* 80(1): 68-76.

Chen, C., Sarin, S., Ram, B. 1991. The pallet packing problem for non-uniform box sizes. *The International Journal Of Production Research,* 29(10): 1963-1968.

Cherkesly, M., Desaulniers, G., Irnich, S., Laporte, G. 2016. Branch-price-and-cut algorithms for the pickup and delivery problem with time windows and multiple stacks. *European Journal of Operational Research*, 250(3): 782-793.

Cherkesly, M., Desaulniers, G., Laporte, G. 2015. A population-based metaheuristic for the pickup and delivery problem with time windows and LIFO loading. *Computers & Operations Research*, 6223-35.

Cordeau, J.-F. 2006. A branch-and-cut algorithm for the dial-a-ride problem. *Operations Research,* 54(3): 573-586.

Cordeau, J. F., Iori, M., Laporte, G., Salazar González, J. J. 2010. A branch-and-cut algorithm for the pickup and delivery traveling salesman problem with LIFO loading. *Networks,* 55(1): 46-59.

Côté, J.-F., Gendreau, M., Potvin, J.-Y. 2014. An exact algorithm for the two-dimensional orthogonal packing problem with unloading constraints. *Operations Research,* 62(5): 1126-1141.

Côté, J. F., Archetti, C., Speranza, M. G., Gendreau, M., Potvin, J. Y. 2012a. A branch-and-cut algorithm for the pickup and delivery traveling salesman problem with multiple stacks. *Networks,* 60(4): 212-226.

Côté, J. F., Gendreau, M., Potvin, J. Y. 2012b. Large neighborhood search for the pickup and delivery traveling salesman problem with multiple stacks. *Networks,* 60(1): 19-30.

Dantzig, G. B., Ramser, J. H. 1959. The truck dispatching problem. *Management Science,* 6(1): 80-91.

Dethloff, J. 2001. Vehicle routing and reverse logistics: the vehicle routing problem with simultaneous delivery and pick-up. *OR-Spektrum,* 23(1): 79-96.

Dominguez, O., Juan, A. A., Faulin, J. 2014. A biased-randomized algorithm for the two-dimensional vehicle routing problem with and without item rotations. *International Transactions in Operational Research,* 21(3): 375-398.

Duhamel, C., Lacomme, P., Quilliot, A., Toussaint, H. 2011. A multi-start evolutionary local search for the two-dimensional loading capacitated vehicle routing problem. *Computers & Operations Research,* 38(3): 617-640.

Düzakın, E., Demircioğlu, M. 2009. Araç Rotalama Problemleri ve Çözüm Yöntemleri. *Çukurova Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi,* 13(1).

Felipe, Á., Ortuño, M. T., Tirado, G. 2009. The double traveling salesman problem with multiple stacks: A variable neighborhood search approach. *Computers & Operations Research,* 36(11): 2983-2993.

Fuellerer, G., Doerner, K. F., Hartl, R. F., Iori, M. 2009. Ant colony optimization for the two-dimensional loading vehicle routing problem. *Computers & Operations Research,* 36(3): 655-673.

Fuellerer, G., Doerner, K. F., Hartl, R. F., Iori, M. 2010. Metaheuristics for vehicle routing problems with three-dimensional loading constraints. *European Journal of Operational Research,* 201(3): 751-759.

Gao, X., Lim, A., Qin, H., Zhu, W. 2011. Multiple pickup and delivery TSP with LIFO and distance constraints: a VNS approach. International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, 2011, Syracuse, NY.

Gencer, C., Yaşa, Ö. 2013. Ulaştirma Komutanliği Ring Seferlerinin Eş Zamanli Dağitim Toplama Karar Destek Sistemi. *Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi,* 22(3).

Gendreau, M., Iori, M., Laporte, G., Martello, S. 2006. A tabu search algorithm for a routing and container loading problem. *Transportation Science,* 40(3): 342-350.

Gendreau, M., Iori, M., Laporte, G., Martello, S. 2008. A Tabu search heuristic for the vehicle routing problem with two-dimensional loading constraints. *Networks,* 51(1): 4-18.

Golden, B.L., Baker, E.K., Alfaro, J.L., Schaffer, J.R., 1985. The Vehicle Routing Problem with Backhauling: Two Approaches. 1985, University of Maryland, College Park.

Gendreau, M., Laporte, G., Vigo, D. 1999. Heuristics for the traveling salesman problem with pickup and delivery. *Computers & Operations Research,* 26(7): 699-714.

Guy Desaulniers, b. D., Andreas Erdmann, Marius M. Solomon, Francois Soumis. 2002. VRP with Pickup and Delivery: The Vehicle Routing Problem, Editörler: Toth, P. Vigo, D., Bologna, Italy: Society of Industrial and Applied Mathematics. pp. 225-242.

Hokama, P., Miyazawa, F. K., Xavier, E. C. 2016. A branch-and-cut approach for the vehicle routing problem with loading constraints. *Expert Systems with Applications,* 471-13.

Iori, M., Martello, S. 2010. Routing problems with loading constraints. *Top,* 18(1): 4-27.

Iori, M., Salazar-González, J.-J., Vigo, D. 2007. An exact approach for the vehicle routing problem with two-dimensional loading constraints. *Transportation Science,* 41(2): 253-264.

Junqueira, L., Morabito, R. 2015. Heuristic algorithms for a three-dimensional loading capacitated vehicle routing problem in a carrier. *Computers & Industrial Engineering,* 88110-130.

Junqueira, L., Morabito, R., Yamashita, D. S. 2012. Three-dimensional container loading models with cargo stability and load bearing constraints. *Computers & Operations Research,* 39(1): 74-85.

Junqueira, L., Oliveira, J. F., Carravilla, M. A., Morabito, R. 2013. An optimization model for the vehicle routing problem with practical three-dimensional loading constraints. *International Transactions in Operational Research,* 20(5): 645-666.

Koç, Ç. 2012. Zaman bağımlı araç rotalama problemi. Yüksel Lisans Tezi, Selçuk Üniversitesi Fen Bilimleri Enstitüsü, Konya.

Lacomme, P., Toussaint, H., Duhamel, C. 2013. A GRASP× ELS for the vehicle routing problem with basic three-dimensional loading constraints. *Engineering Applications of Artificial Intelligence,* 26(8): 1795-1810.

Ladany, S. P., Mehrez, A. 1984. Optimal routing of a single vehicle with loading and unloading constraints. *Transportation Planning and Technology,* 8(4): 301-306.

Leung, S. C., Zheng, J., Zhang, D., Zhou, X. 2010. Simulated annealing for the vehicle routing problem with two-dimensional loading constraints. *Flexible Services And Manufacturing Journal,* 22(1-2): 61-82.

Leung, S. C., Zhou, X., Zhang, D., Zheng, J. 2011. Extended guided tabu search and a new packing algorithm for the two-dimensional loading vehicle routing problem. *Computers & Operations Research,* 38(1): 205-215.

Li, Y., Chen, H., Prins, C. 2016. Adaptive large neighborhood search for the pickup and delivery problem with time windows, profits, and reserved requests. *European Journal of Operational Research,* 252(1): 27-38.

Lusby, R. M., Larsen, J., Ehrgott, M., Ryan, D. 2010. An exact method for the double TSP with multiple stacks. *International Transactions in Operational Research,* 17(5): 637-652.

Lynwood A. Johnson, D. C. M. 1974. Operations Research in Production Planning, Scheduling, and Inventory Control. New York, 544.

Ma, H.-w., Zhu, W., Xu, S. 2011. Research on the algorithm for 3L-CVRP with considering the utilization rate of vehicles: Intelligent computing and information science, Editör: Chen, R., Chongqing, China, 621-629.

Männel, D., Bortfeldt, A. 2015a. A Hybrid Algorithm for the Vehicle Routing Problem with Pickup and Delivery and 3D Loading Constraints. Univ., Faculty of Economics and Management, 2015, Magdeburg, Germany.

Männel, D., Bortfeldt, A. 2015b. Solving the Pickup and Delivery Problem with 3D Loading Constraints and Reloading Ban. Univ., Faculty of Economics and Management, 2015, Magdeburg, Germany.

Massen, F., Deville, Y., Van Hentenryck, P. 2012. Pheromone-based heuristic column generation for vehicle routing problems with black box feasibility: Integration of AI and OR Techniques in Contraint Programming for Combinatorial Optimzation Problems, Editörler: : Beldiceanu N, Jussien N, Pinson É, Heidelberg, Berlin, 260-274.

Miao, L., Ruan, Q., Woghiren, K., Ruo, Q. 2012. A hybrid genetic algorithm for the vehicle routing problem with three-dimensional loading constraints. *RAIRO-Operations Research,* 46(1): 63-82.

Min, H. 1989. The multiple vehicle routing problem with simultaneous delivery and pick-up points. *Transportation Research Part A: General,* 23(5): 377-386.

Mosheiov, G. 1994. The travelling salesman problem with pick-up and delivery. *European Journal of Operational Research*, 79(2): 299-310.

Moura, A., Oliveira, J. F. 2009. An integrated approach to the vehicle routing and container loading problems. *OR spectrum,* 31(4): 775-800.

Nagy, G., Salhi, S. d. 2005. Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries. *European journal of operational research,* 162(1): 126-141.

Ning, T., Guo, C., Chen, R., Jin, H. 2016. A Novel Hybrid Method on VRP with Pickup and Delivery. *The Open Cybernetics & Systemics Journal*, 10(1): 56-60.

Pacheco, J. A. 1997. Heurístico para los problemas de rutas con carga y descarga en sistemas LIFO, *Statistica and Operations Research Transactions,* 21, 153-175.

Packer 3d, 2017. http://www.packer3d.com/ (Erişim Tarihi: 27.02.2017).

Petersen, H. L., Madsen, O. B. 2009. The double travelling salesman problem with multiple stacks–Formulation and heuristic solution approaches. *European Journal of Operational Research,* 198(1): 139-147.

Pollaris, H., Braekers, K., Caris, A., Janssens, G. K., Limbourg, S. 2015. Vehicle routing problems with loading constraints: state-of-the-art and future directions. *OR Spectrum*, 37(2): 297-330.

Ren, J., Tian, Y., Sawaragi, T. 2011. A relaxation method for the three-dimensional loading capacitated vehicle routing problem. System Integration (SII), 2011 IEEE/SICE International Symposium on System Integration, 2011, IEEE.

Ropke, S., Cordeau, J. F., Laporte, G. 2007. Models and branch-and-cut algorithms for pickup and delivery problems with time windows. *Networks,* 49(4): 258-272.

Royo, B., Sicilia, J.-A., Oliveros, M.-J., Larrodé, E. 2015. Solving a Long-Distance Routing Problem using Ant Colony Optimization. *Applied Mathematics,* 9(2L): 415-421.

Ruan, Q., Zhang, Z., Miao, L., Shen, H. 2013. A hybrid approach for the vehicle routing problem with three-dimensional loading constraints. *Computers & Operations Research,* 40(6): 1579-1589.

Ruland, K., Rodin, E. 1997. The pickup and delivery problem: Faces and branch-and-cut algorithm. *Computers & Mathematics with Applications,* 33(12): 1-13.

Salhi, S., Nagy, G. 1999. A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling. *Journal of the Operational Research Society,* 50(10): 1034-1042.

Sampaio, A. H., Urrutia, S. 2017. New formulation and branch-and-cut algorithm for the pickup and delivery traveling salesman problem with multiple stacks. *International Transactions in Operational Research,* 24(2): 77-98.

Sigurd, M., Pisinger, D., Sig, M. 2004. Scheduling transportation of live animals to avoid the spread of diseases. *Transportation Science,* 38(2): 197-209.

Strodl, J., Doerner, K. F., Tricoire, F., Hartl, R. F. 2010. On index structures in hybrid metaheuristics for routing problems with hard feasibility checks: an application to the 2-

dimensional loading vehicle routing problem: Hybrid Metaheuristics, Editörler: Blesa, M., Blum, C., Raidl, G., Roli, A.,Sampels, M., Heidelberg, Berlin, 160-173.

Tao, Y., Wang, F. 2015. An effective tabu search approach with improved loading algorithms for the 3L-CVRP. *Computers & Operations Research*, 55:127-140.

Tasan, A. S., Gen, M. 2012. A genetic algorithm based approach to vehicle routing problem with simultaneous pick-up and deliveries. *Computers & Industrial Engineering*, 62(3): 755-761.

Toth, P., Vigo, D. 2002. The Vehicle Routing Problem. Bologna, Italy, 463.

Türkay, A. 2002. Yükleme Kısıtı Altında Taşıt Rotalama Problemi. *Yüksek Lisans Tezi*, Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalı, Uludağ Üniversitesi, Bursa.

Türkay, A., Emel, E. 2003. Vehicle routing problem with packing constraints. Presentation on the 5th Euro/Informs Joint International Meeting, 2003, July.

Urrutia, S., Milanés, A., Løkketangen, A. 2015. A dynamic programming based local search approach for the double traveling salesman problem with multiple stacks. *International Transactions in Operational Research*, 22(1): 61-75.

Veenstra, M., Cherkesly, M., Desaulniers, G., Laporte, G. 2017. The pickup and delivery problem with time windows and handling operations. *Computers & Operations Research*, 77127-140.

Wei, L., Zhang, Z., Zhang, D., Lim, A. 2015. A variable neighborhood search for the capacitated vehicle routing problem with two-dimensional loading constraints. *European Journal of Operational Research*, 243(3): 798-814.

Wisniewski, M. A., Ritt, M., Buriol, L. S. 2012. A tabu search algorithm for the capacitated vehicle routing problem with three-dimensional loading constraints. XLIII Simposio Brasileiro de Pesquısa Operacional, 2011, Ubatuba, Sao Paulo.

Zachariadis, E. E., Tarantilis, C. D., Kiranoudis, C. T. 2013. Designing vehicle routes for a mix of different request types, under time windows and loading constraints. *European Journal of Operational Research*, 229(2): 303-317.

Zachariadis, E. E., Tarantilis, C. D., Kiranoudis, C. T. 2016. The vehicle routing problem with simultaneous pick-ups and deliveries and two-dimensional loading constraints. *European Journal of Operational Research*, 251(2): 369-386.

Zhang, Z., Wei, L., Lim, A. 2015. An evolutionary local search for the capacitated vehicle routing problem minimizing fuel consumption under three-dimensional loading constraints. *Transportation Research Part B: Methodological*, 82:20-35.

Zhu, W., Qin, H., Lim, A., Wang, L. 2012. A two-stage tabu search algorithm with enhanced packing heuristics for the 3L-CVRP and M3L-CVRP. *Computers & Operations Research*, 39(9): 2178-2195.

EKLER

EK 1 Örnek Problemin FICO Xpress Programında Yazımı

EK 2 Örnek Problemin FICO Xpress Programında Çözümü

EK 3 Noktalara Ait Maliyet Verileri

EK 1 Örnek Problemin FICO Xpress Programında Yazımı

```
model ikiboyutludagitimtoplamalirotalamaproblemi
uses "mmxprs","mmsystem","mmodbc"
parameters
   K=15
   WEIGHT=100                 !ağırlık kısıtı
   ALAN=12                    !hacim kısıtı
   EN=2                       !m
   BOY=6                      !m
   M=1000
   m=0.001
   l=1                        !kutu boyu
   w=1                        !kutu eni
   ALG = 6                    ! Algorithm choice [default settings: 0]
   CUTDEPTH = 1000            ! Maximum tree depth for cut generation
   EPS = 1e-6
end-parameters
forward   function cb_hata:boolean
forward procedure tree_cut_gen
declarations
   LIFOSatisfied:boolean
   NODE =0..K                 !depo dahil tüm noktalar
   SIRA = 1..K                !sıra indisi
   KUTUTIP=1                  !kutu tipi
   X=0..2
   XX=0..2
   Y=0..6
   YY=0..6
   PICKUP=1..7                !toplama noktaları
   DELIVERY=8..14             !dağıtım noktaları
   S=1..14
   Pickup: set of string
   Delivery: set of string
   INDIS: array (PICKUP) of integer  !ilgili toplama noktasına ait dağıtım noktası
   COST: dynamic array(NODE,NODE) of integer        !maliyet
   XDATA:dynamic array(X) of integer
   YDATA:dynamic array(Y) of integer
   P:dynamic array (NODE) of integer                !toplama yük miktarı;
   D:dynamic array (NODE) of integer                !dağıtım yük miktarı;
   V:dynamic array (NODE) of integer                !Kutuların hacimleri;
   F:dynamic array (NODE) of integer      !noktalara ait belirli bir tipteki kutu sayısı
   R: dynamic array(NODE,NODE,SIRA) of mpvar        !Rota değişkeni
   U:array(NODE) of mpvar                           !alt tur değişkeni
   L:array(NODE) of mpvar                           !i noktasına ait yük değişkeni
   a:array(NODE,SIRA,X,Y) of mpvar                  !koordinatlara atama değişkeni
   starttime:real
end-declarations
initializations from "mmsheet.excel:veri.xls"
       COST as "noindex;[7.2$B2:Q17]"
end-initializations
INDIS::[8,9,10,11,12,13,14]          !ilgili toplama noktasına ait dağıtım noktası
P::[0,20,30,20,10,10,20,10,0,0,0,0,0,0,0,0] !toplama yük miktarı;
D::[0,0,0,0,0,0,0,0,20,30,20,10,10,20,10,0] !dağıtım yük miktarı;
V::[1,1,2,1,3,1,4,1,5,1,6,1,7,1]            !Kutuların hacimleri;
F::[0,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0]
XDATA::[0,1,2]
YDATA::[0,1,2,3,4,5,6]
forall   (i in NODE, j in NODE,b in SIRA ) create(R(i,j,b))
forall  (i in NODE, j in NODE,b in SIRA)do
   R(i,j,b) is_binary
end-do
forall(i in NODE,b in SIRA, x in X, y in Y ) create(a(i,b,x,y))
```

```
forall (i in NODE,b in SIRA, x in X, y in Y )do
   a(i,b,x,y) is_binary
end-do

!amaç fonksiyonu
Maliyet:=sum(i in NODE,j in NODE,b in SIRA)COST(i,j)* R(i,j,b)+sum(j in NODE,b in SIRA,x
in X,y in Y)(YDATA(y)+XDATA(x))*a(j,b,x,y)

!rotalama kısıtları
sum(j in PICKUP)  R(0,j,1) = 1
forall(i in S)    sum(j in 1..15|i<>j, b in SIRA|b>1) R(i,j,b) = 1
forall(q in S,b in SIRA|b<15) sum(i in 0..14|i<>q)R(i,q,b) - sum(j in 1..15|q<>j)
R(q,j,b+1) = 0
sum(i in DELIVERY)   R(i,15,15)=1
forall(i in PICKUP,j in DELIVERY|j<>INDIS(i),b in SIRA)    R(i,j,b)=0

!alt tur kısıtı
forall(i in NODE|i>0,j in NODE|i<>j and j>0,b in SIRA)U(i)-U(j)+K*R(i,j,b)<=K-1

!Rotada gidilen noktanın yük miktarı ve kapasite kısıtları;
forall(i in PICKUP)  L(i) >= 0+P(i)-D(i)-M*(1-R(0,i,1))
forall(i in S, j in S|i<>j, b in SIRA|b>1) L(j) >= L(i)+P(j)-D(j)-M*(1-R(i,j,b))
forall(j in PICKUP) P(j) <= L(j)
forall(j in DELIVERY)  L(j) >= 0
forall(j in PICKUP) L(INDIS(j)) <= WEIGHT - P(j)
L(15)=0
L(0)=0

!Bir şehir yalnızca bir sıraya atanabilir;
forall(j in PICKUP) R(0,j,1)<=1
forall(i in S, j in NODE|j>0 and i<>j)sum(b in SIRA|b>1)   R(i,j,b)<=1
forall(i in DELIVERY) R(i,15,15)<=1

!Bir sıraya yalnızca bir şehir atanabilir;
sum(j in PICKUP)  R(0,j,1)=1
forall(b in SIRA|b>1 and b<15) sum(i in S, j in S|i<>j) R(i,j,b)=1
forall(j in DELIVERY) R(j,15,15)<=1

!Sıralama kısıtları;
forall(j in PICKUP)R(j,INDIS(j),2)<=R(0,j,1)
forall(j in PICKUP,q in PICKUP|j<>q)R(j,q,2)<=R(0,j,1)
forall(j in PICKUP,i in S|i<>j,b in SIRA|b>2 and b<15)R(j,INDIS(j),b+1)-m*R(i,j,b)>=0

!Yükleme kısıtları;
sum(i in NODE|i<K,j in NODE|j>0 and i<>j,b in SIRA) D(i)*F(i)*R(i,j,b)  <= ALAN
forall(i in NODE|i>0,b in SIRA)              sum(x in 0..EN-1,y in 0..BOY-1)a(i,b,x,y) =
sum(j in NODE|j<K and i<>j)R(j,i,b)*F(i)
forall(xx in 0..EN-1,yy in 0..BOY-1) sum(i in NODE, b in SIRA, x in 0..xx|x>=xx, y in
0.. yy|y>=yy) a(i,b,x,y)<=1
forall(i in NODE) sum(b in SIRA, x in 0..EN-1,y in 0..BOY-1)a(i,b,x,y) = F(i)
forall(i in PICKUP, j in PICKUP|i<>j, b in 2..K,x in X,xx in X|x<>xx,y in Y,yy in
Y|y<>yy) R(i,j,b)*M+XDATA(x)*a(i,b-1,x,y)<=XDATA(xx)*a(j,b,xx,yy)+M
forall(i in PICKUP, j in PICKUP|i<>j, b in 2..K,x in X,xx in X|x<>xx,y in Y,yy in
Y|y<>yy) R(i,j,b)*M+YDATA(y)*a(i,b-1,x,y)<=YDATA(yy)*a(j,b,xx,yy)+M

 ! Uncomment to get detailed MIP output
 setparam("XPRS_VERBOSE", true)
 ! All cost data are integer, we therefore only need to search for integer  solutions
 setparam("XPRS_MIPADDCUTOFF", -0.999)
! Set Mosel comparison tolerance to a sufficiently small value
 setparam("ZEROTOL", EPS/100)
 writeln("**************  ALG=",ALG,  "***************")
 starttime:=gettime
 SEVERALROUNDS:=false
```

```
 TOPONLY:=false
 case ALG of
  1: setparam("XPRS_CUTSTRATEGY", 0)   ! No cuts
  2: setparam("XPRS_PRESOLVE", 0)      ! No presolve
  3: tree_cut_gen                      ! User branch-and-cut + automatic cuts
  4: do                                ! User branch-and-cut (several rounds),
      tree_cut_gen                     ! no automatic cuts
      setparam("XPRS_CUTSTRATEGY", 0)
      SEVERALROUNDS:=true
     end-do
  5: do                                ! User cut-and-branch (several rounds)
      tree_cut_gen                     ! + automatic cuts
      SEVERALROUNDS:=true
      TOPONLY:=true
     end-do
  6: do                                ! User branch-and-cut (several rounds)
      tree_cut_gen                     ! + automatic cuts
      SEVERALROUNDS:=true
     end-do
 end-case

!Find Solution
   minimize(Maliyet)
!Solution printing
         writeln("Maliyet: ",getobjval)
         writeln("i  ","X  ","Y  ")
forall(i in NODE, b in SIRA, x in X, y in Y)
   if(getsol(a(i,b,x,y))>0) then
      writeln(i,"  ",x,"  ",y)
   end-if
   writeln("CIKIS  ","VARIS ","SIRA  ", "YUK")

   forall(i in 1..K)
      if(getsol(R(0,i,1))>0) then
            writeln("  ",0, "     ",i,"(P)","  ", 1,"     ",getsol(L(0)))
            p:=i
            while (p<>K) do
         forall(j in 1..K, h in 2..K)
            if(getsol(R(p,j,h))>0) then
                  if (j in PICKUP) then
                     writeln("  ",p, "     ",j,"(P)","  ", h,"     ",getsol(L(p)))
                  else
                     writeln("  ",p, "     ",j,"(D)","  ", h,"     ",getsol(L(p)))
                  end-if
               p:=j
            end-if
         end-do
      end-if
   B:=1
   while (B<15) do
      forall(i in NODE)do
         forall(j in PICKUP)do
            if getsol(R(i,j,B))>0 then
               writeln("S",B," Pic: ",j)
               p1:=j
            end-if
         end-do
      end-do
      forall(h in SIRA|h>B)do
         forall(k in NODE)do
            if getsol(R(k,INDIS(p1),h))>0 then
               writeln("S",h," Del: ",INDIS(p1))
               H:=h
            end-if
```

```
            end-do
        end-do
        forall(i in NODE)do
           forall(j in PICKUP)do
              if getsol(R(i,j,B+1))>0 then
                 writeln("S",B+1," Pic: ",j)
                 p2:=j
              end-if
           end-do
        end-do
        forall(h1 in SIRA|h1>B+1)do
           forall(k in NODE)do
              if getsol(R(k,INDIS(p2),h1))>0 then
                 writeln("S ",h1," Del: ",INDIS(p2))
                 H1:=h1
              end-if
           end-do
        end-do
        forall(j in PICKUP|j<>p)do
           if getsol(R(j,p2,B+1))>0 then
              if H<H1 then
                  writeln("rota yanlis","    ", INDIS(p2),". node    ", H1,". siraya
                 atanamaz")
              end-if
           end-if
        end-do
        B+=2
  end-do
 function cb_hata:boolean
 declarations
    ncut:integer                                     ! Counter for cuts
    cut: array(range) of linctr               ! Cuts
    cutid: array(range) of integer            ! Cut type identification
    type: array(range) of integer             ! Cut constraint type
    objval,ds: real
    vH: integer
    vH1:integer
    vp:integer
    vp1:integer
    icount: integer
 end-declarations
 depth:=getparam("XPRS_NODEDEPTH")
 LIFOSatisfied := true
 icount := 0
  if((TOPONLY and depth<1) or (not TOPONLY and depth<=CUTDEPTH)) then
    forall(b in SIRA|b<K+1)do
       forall(i in NODE)do
          forall(j in NODE)do
             if abs(getsol(R(i,j,b))-1.) < 0.000001 AND isintegral(R(i,j,b)) then
                icount += 1
             else
                returned := false
             end-if
          end-do
       end-do
    end-do
 !Bulunan tamsayı çözüm sayısı rota K değerine eşit ise çözümü sına
    if icount = K then
       writeln("Integer Variable Count: ", icount)
       writeln("Integer solution found. LIFO conditions will be checked.")
       ncut:=0
       forall(b in SIRA|b<K)do
 ! b. sırada gidilecek noktanın numarasını bul
          forall(i in NODE)do
```

```
            forall(j in NODE)do
               if abs(getsol(R(i,j,b))-1.) < 0.001 and isintegral(R(i,j,b)) then
                  vp:=j
               end-if
            end-do
         end-do
! Gidilecek vp noktası PICKUP ise onun DELIVERY noktasına hangi sırada gidileceğini bul
         if (vp in PICKUP) then
            forall(h in SIRA|h>b and h<K+1)do
               forall(k in NODE)do
                  if abs(getsol(R(k,INDIS(vp),h))-1.) < 0.001 and
                  isintegral(R(k,INDIS(vp),h)) then
                     writeln(b, ". Pickup: ", vp," Delivery: ",INDIS(vp), " @ B: ", h)
                     vH:=h
                  end-if
               end-do
            end-do
            forall(x in X, y in Y)do
               if abs(getsol(a(vp,b,x,y))-1.)< 0.001 then
                  writeln("   Noktanin konumu: "," x: ",x," y: ",y," Yuk miktari:
                  ",getsol(L(vp)))
               end-if
             end-do
         end-if
! b. sıradaki noktadan b+1. sıradakine gittikten sonra gelinen noktanın numarasını bul
         forall(j in NODE|j<>vp)do
            if abs(getsol(R(vp,j,b+1))-1.) < 0.001 and isintegral(R(vp,j,b+1)) then
               vp1:=j
            end-if
         end-do
! B+1. sırada gelinen nokta PICKUP ise
         if (vp1 in PICKUP and vp1<>vp) then
            forall(h1 in SIRA|h1>b+1 and h1<K+1 and h1<>vH)do
               forall(k in NODE)do
                  if abs(getsol(R(k,INDIS(vp1),h1))-1.) < 0.001 and
                  isintegral(R(k,INDIS(vp1),h1)) then
                     vH1:=h1
                  end-if
               end-do
            end-do
         end-if
         if(vH <= vH1 and vp1 in PICKUP and vp in PICKUP) then
            writeln("!!*** LIFO conditions are not satisfied.", depth," :depth "," node:
            ", getparam("XPRS_NODES"), "ObjVal:", getparam("XPRS_LPOBJVAL"))
            LIFOSatisfied:=false
            break
         end-if
      end-do
      if LIFOSatisfied then
         writeln("EUREKA: Solution found satisfying LIFO constraint.", depth," :depth
         ","node: ", getparam("XPRS_NODES"), " ObjVal: ", getparam("XPRS_LPOBJVAL"))
      end-if
      returned := false
      if SEVERALROUNDS then
         returned:=true                   ! Repeat until no new cuts generated
      end-if
   end-if
 end-if
end-function
! ****Optimizer settings for using the cut manager****
procedure tree_cut_gen
    setparam("XPRS_PRESOLVE", 0)         ! Switch presolve off
    setparam("XPRS_ROOTPRESOLVE", 0)     ! Switch B&B root presolve off
    setparam("XPRS_EXTRAROWS", 5000)     ! Reserve extra rows in matrix
```

```
    setparam("XPRS_CUTSTRATEGY", 0)
    setcallback(XPRS_CB_CM, "cb_hata")  ! Set the cut-manager callback function
end-procedure
end-model
```

EK 2 Örnek Problemin FICO Xpress Programında Çözümü

```
**************  ALG=6***************
Reading Problem \xprs_4ee0cb0
Problem Statistics
      303873 (   5000 spare) rows
        8641 (      0 spare) structural columns
      790619 (      0 spare) non-zero elements
Global Statistics
        8610 entities        0 sets        0 set members
Minimizing MILP \xprs_4ee0cb0
Original problem has:
    303873 rows         8641 cols       790619 elements      8610 globals
Symmetric problem: generators: 168, support set: 180 (partial detection)
 Number of orbits: 12, largest orbit: 15 (largest clique: 15)
Will try to keep branch and bound tree memory usage below 4.9Gb
Starting concurrent solve with dual, primal and barrier (2 threads)

                           Concurrent-Solve,   5s
              Dual                       Primal                   Barrier
    objective   sum inf        objective   dual inf        p.obj.     d.obj.
 p   .2426036   .0000000 |  D  1258.2849  10.830264 |  B -1.215E+08   .0000000
 p  804.25677   .0000000 |  D  1509.0982   .0000000 |  B -1.047E+08 -3233.7850
 P  974.99847   .0000000 |  D  1153.5052   .0000000 |  B -1.001E+08  365.87274
 P  1030.5986   .0000000 |  D  1082.5993   .0000000 |  B -34143531.  12014.973
 P  1038.4253   .0000000 |  D  1046.5916   .0000000 |  B -4275357.1  10725.468
----- interrupted ------ | ------- optimal -------- | ----- interrupted ------
Concurrent statistics:
      Dual: 1430 simplex iterations, 4.50s
    Primal: 3287 simplex iterations, 5.13s
   Barrier: 5 barrier and 0 simplex iterations, 5.82s
            Barrier used 2 threads 2 cores, L1\L2 cache: 32K\3072K
            Barrier used AVX support
Optimal solution found

   Its         Obj Value      S   Ninf  Nneg        Sum Inf  Time
  3287       1045.743012      P      0     0        .000000     8
Dual solved problem
  3287 simplex iterations in 8s
```

```
Final objective                          : 1.045743012056538e+03
  Max primal violation      (abs / rel) : 9.978e-07 / 9.978e-07
  Max dual violation        (abs / rel) : 1.705e-13 / 8.731e-14
  Max complementarity viol. (abs / rel) :       0.0 /       0.0
All values within tolerances

Starting root cutting & heuristics

 Its Type    BestSoln    BestBound   Sols    Add    Del     Gap     GInf   Time
Heuristic search started
Heuristic search stopped
M         1711.000000  1045.743012      1                 38.88%       0     28
Will try to keep branch and bound tree memory usage below 4.9Gb

Starting tree search.
Deterministic mode with up to 4 running threads and up to 8 tasks.

    Node     BestSoln    BestBound   Sols Active  Depth     Gap     GInf   Time
       1  1711.000000  1045.904811      1      2      1  38.87%      113     41
       2  1711.000000  1054.232039      1      2      3  38.39%      189     43
       3  1711.000000  1054.831765      1      3      3  38.35%       87     43
       4  1711.000000  1054.831765      1      4      4  38.35%       56     45
       5  1711.000000  1055.624014      1      5      4  38.30%      162     46
       6  1711.000000  1055.624014      1      6      5  38.30%      162     47
       7  1711.000000  1055.624014      1      8      5  38.30%       57     47
       8  1711.000000  1057.315015      1      8      6  38.20%       35     49
       9  1711.000000  1057.315015      1      9      6  38.20%      155     50
      10  1711.000000  1057.315015      1      9      7  38.20%       70     50
      20  1711.000000  1065.915694      1     18     12  37.70%       48     65
      32  1711.000000  1067.514564      1     22     13  37.61%       45     68
      42  1711.000000  1067.514564      1     24     13  37.61%       31     73
Will try to keep branch and bound tree memory usage below 4.3Gb
      52  1711.000000  1067.514564      1     31     14  37.61%      151     78
      62  1711.000000  1067.514564      1     32     19  37.61%       46     80
a     72  1285.000000  1067.514564      2     37     17  16.92%        0     82
      76  1285.000000  1067.514564      2     37     10  16.92%       30     82
      87  1285.000000  1067.514564      2     33     18  16.92%       38     84
     103  1285.000000  1067.514564      2     38     19  16.92%       32     85
```

```
Will try to keep branch and bound tree memory usage below 3.8Gb
B&B tree size: 8.0Mb total

    Node     BestSoln    BestBound   Sols Active  Depth     Gap     GInf   Time
a    120  1228.000000  1076.077002      3     43     13  12.37%        0     93
     203  1228.000000  1076.077002      3     19     19  12.37%      147    112
     303  1228.000000  1085.255167      3     61     16  11.62%       81    146
a    310  1227.000000  1085.255167      4     59     27  11.55%        0    151
     405  1227.000000  1085.255167      4     47     17  11.55%       45    163
     505  1227.000000  1095.263516      4     50     13  10.74%       33    176
     611  1227.000000  1103.323716      4     22     29  10.08%       38    187
a    700  1215.000000  1115.717473      5     40     28   8.17%        0    199
     713  1215.000000  1115.717473      5     37     40   8.17%      119    200
a    782  1179.000000  1125.031614      6     17     31   4.58%        0    206
     815  1179.000000  1125.031614      6     11     30   4.58%       41    207
 *** Search completed ***     Time:   210 Nodes:        865
Number of integer feasible solutions found is 6
Best integer solution found is  1179.000000
Best bound is  1179.000000
Maliyet: 1179
i  X  Y
1  0  0
2  0  1
3  1  0
4  0  2
5  0  3
6  1  2
7  1  1
CIKIS  VARIS SIRA  YUK
  0     1(P)   1     0
  1     2(P)   2     20
  2     9(D)   3     50
  9     6(P)   4     20
  6     13(D)   5     40
  13     4(P)   6     20
  4     11(D)   7     30
  11     5(P)   8     20
  5     12(D)   9     30
  12     8(D)   10     20
```

```
  8      7(P)   11      0
  7      14(D)   12      10
  14      3(P)   13      0
  3      10(D)   14      20
  10      15(D)   15      0
S1 Pic: 1
S10 Del: 8
S2 Pic: 2
S 3 Del: 9
S10 Del: 8
S4 Pic: 6
S 5 Del: 13
S10 Del: 8
S6 Pic: 4
S 7 Del: 11
S10 Del: 8
S8 Pic: 5
S 9 Del: 12
S10 Del: 8
S11 Pic: 7
S12 Del: 14
S13 Pic: 3
S14 Del: 10
```

EK 3 Noktalara Ait Maliyet Verileri

Çizelge EK 3.1. Beş toplama noktası maliyet verileri

i \ *j*	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
0	0	70	112	82	56	128	122	76	86	64	48	0
1	70	-	120	107	104	81	105	95	115	86	115	70
2	112	120	-	149	136	133	110	134	124	144	138	112
3	82	107	149	-	129	116	113	90	114	104	139	82
4	56	104	136	129	-	115	102	99	76	81	105	56
5	128	81	133	116	115	-	128	201	110	90	82	128
6	122	105	110	113	102	128	-	37	90	45	88	122
7	76	95	134	90	99	201	37	-	22	178	200	76
8	86	115	124	114	76	110	90	22	-	48	100	86
9	64	86	144	104	81	90	45	178	48	-	182	64
10	48	115	138	139	105	82	88	200	100	182	-	48
11	-	70	112	82	56	128	122	76	86	64	48	-

Çizelge EK 3.2. Beş toplama noktası talep verileri

Talepler			
Nokta	**Toplama Talebi**	**Nokta**	**Dağıtım Talebi**
1	20	6	20
2	10	7	10
3	20	8	20
4	15	9	15
5	15	10	15

Çizelge EK 3.3. Altı toplama noktası maliyet verileri

							j								
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**
	0	0	70	112	82	56	128	122	76	86	64	48	202	144	0
	1	70	-	120	107	104	81	105	95	115	86	115	102	99	70
	2	112	120	-	149	136	133	110	134	124	144	138	125	122	112
	3	82	107	149	-	129	116	113	90	114	104	139	126	123	82
	4	56	104	136	129	-	115	102	99	76	81	105	95	115	56
	5	128	81	133	116	115	-	128	201	110	90	82	31	60	128
i	**6**	122	105	110	113	102	128	-	37	90	45	88	205	129	122
	7	76	95	134	90	99	201	37	-	22	178	200	150	28	76
	8	86	115	124	114	76	110	90	22	-	48	100	81	44	86
	9	64	86	144	104	81	90	45	178	48	-	182	42	144	64
	10	48	115	138	139	105	82	88	200	100	182	-	99	93	48
	11	202	102	125	126	95	31	205	150	81	42	99	-	129	180
	12	144	99	122	123	115	60	129	28	44	144	93	129	-	144
	13	-	70	112	82	56	128	122	76	86	64	48	180	144	-

Çizelge EK 3.4. Altı toplama noktası talep verileri

Talepler			
Nokta	**Toplama Talebi**	**Nokta**	**Dağıtım Talebi**
1	20	7	20
2	30	8	30
3	20	9	20
4	10	10	10
5	10	11	10
6	20	12	20

Çizelge EK 3.5. Sekiz toplama noktası maliyet verileri

									j										
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**
	0	-	70	112	82	56	128	122	76	86	64	48	202	144	89	24	27	111	0
	1	70	-	120	107	104	81	105	95	115	86	115	102	99	76	150	45	146	70
	2	112	120	-	149	136	133	110	134	124	144	138	125	122	154	160	154	68	112
	3	82	107	149	-	129	116	113	90	114	104	139	126	123	100	46	198	42	82
	4	56	104	136	129	-	115	102	99	76	81	105	95	115	86	180	201	157	56
	5	128	81	133	116	115	-	128	201	110	90	82	31	60	98	160	45	129	128
	6	122	105	110	113	102	128	-	37	90	45	88	205	129	67	70	76	41	122
	7	76	95	134	90	99	201	37	-	22	178	200	150	28	50	150	49	38	76
i	**8**	86	115	124	114	76	110	90	22	-	48	100	81	44	84	240	112	139	86
	9	64	86	144	104	81	90	45	178	48	-	182	42	144	40	140	45	56	64
	10	48	115	138	139	105	82	88	200	100	182	-	99	93	130	230	168	200	48
	11	202	102	125	126	95	31	205	150	81	42	99	-	129	180	200	210	87	202
	12	144	99	122	123	115	60	129	28	44	144	93	129	-	144	120	125	40	144
	13	89	76	154	100	86	98	67	50	84	40	130	180	144	-	100	44	72	89
	14	24	150	160	46	180	160	70	150	240	140	230	200	120	100	-	87	112	24
	15	27	45	154	198	201	45	76	49	112	45	168	210	125	44	87	-	17	27
	16	111	146	68	42	157	129	41	38	139	56	200	87	40	72	112	17	-	111
	17	0	70	112	82	56	128	122	76	86	64	48	202	144	89	24	27	111	-

Çizelge EK 3.6. Sekiz toplama noktası talep verileri

Talepler			
Nokta	**Toplama Talebi**	**Nokta**	**Dağıtım Talebi**
1	20	9	20
2	30	10	30
3	20	11	20
4	10	12	10
5	10	13	10
6	20	14	20
7	10	15	10
8	30	16	30

Çizelge EK 3.7. Dokuz toplama noktası maliyet verileri

j

i		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
	0	0	70	112	82	56	128	122	76	86	64	48	202	144	89	24	27	111	10	206	0
	1	70	0	120	107	104	81	105	95	115	86	115	102	99	76	150	45	146	145	187	70
	2	112	120	0	149	136	133	110	134	124	144	138	125	122	154	160	154	68	189	45	112
	3	82	107	149	0	129	116	113	90	114	104	139	126	123	100	46	198	42	45	91	82
	4	56	104	136	129	0	115	102	99	76	81	105	95	115	86	180	201	157	61	65	56
	5	128	81	133	116	115	0	128	201	110	90	82	31	60	98	160	45	129	47	171	128
	6	122	105	110	113	102	128	0	37	90	45	88	205	129	67	70	76	41	122	80	122
	7	76	95	134	90	99	201	37	0	22	178	200	150	28	50	150	49	38	204	125	76
	8	86	115	124	114	76	110	90	22	0	48	100	81	44	84	240	112	139	169	24	86
i	**9**	64	86	144	104	81	90	45	178	48	0	182	42	144	40	140	45	56	82	44	64
	10	48	115	138	139	105	82	88	200	100	182	0	99	93	130	230	168	200	176	83	48
	11	202	102	125	126	95	31	205	150	81	42	99	0	129	180	200	210	87	192	17	202
	12	144	99	122	123	115	60	129	28	44	144	93	129	0	144	120	125	40	81	215	144
	13	89	76	154	100	86	98	67	50	84	40	130	180	144	0	100	44	72	23	110	89
	14	24	150	160	46	180	160	70	150	240	140	230	200	120	100	0	87	112	79	68	24
	15	27	45	154	198	201	45	76	49	112	45	168	210	125	44	87	0	17	16	77	27
	16	111	146	68	42	157	129	41	38	139	56	200	87	40	72	112	17	0	40	113	111
	17	10	145	189	45	61	47	122	204	169	82	176	192	81	23	79	16	40	0	97	16
	18	206	187	45	91	65	171	80	125	24	44	83	17	215	110	68	77	113	97	0	35
	19	0	70	112	82	56	128	122	76	86	64	48	202	144	89	24	27	111	16	35	0

Çizelge EK 3.8. Dokuz toplama noktası talep verileri

Talepler			
Nokta	**Toplama Talebi**	**Nokta**	**Dağıtım Talebi**
1	20	10	20
2	30	11	30
3	20	12	20
4	10	13	10
5	10	14	10
6	20	15	20
7	10	16	10
8	30	17	30
9	20	18	20

ÖZGEÇMİŞ

Adı Soyadı	: Figen KAS
Doğum Yeri ve Tarihi	: Nilüfer / 15.06.1992
Yabancı Dili	: İngilizce
Eğitim Durumu (Kurum ve Yıl)	
Lise	: Şahinler Anadolu Lisesi-2010
Lisans	: Dokuz Eylül Üniversitesi End. Müh.-2014
Yüksek Lisans	: Uludağ Üniversitesi End. Müh.-2017
Çalıştığı Kurum/Kurumlar ve Yıl	: Işıksoy Tekstil A.Ş.-2016
İletişim (e-posta)	: figenkas@gmail.com
Yayınları	: