

**ARAÇ ROTALAMA PROBLEMLERİ İÇİN
MATEMATİKSEL MODELLER VE ÇÖZÜM YÖNTEMLERİ**

Melis ALPASLAN

Yüksek Lisans Tezi

Endüstri Mühendisliği Anabilim Dalı

Mayıs-2015

**Bu tez çalışması Anadolu Üniversitesi Bilimsel Araştırma Projeleri
Komisyonu Başkanlığı tarafından desteklenmiştir. Proje No: 1402F052**

JÜRİ VE ENSTİTÜ ONAYI

Melis Alpaslan'ın “**Araç Rotalama Problemleri için Matematiksel Modeller ve Çözüm Yöntemleri**” başlıklı **Endüstri Mühendisliği** Anabilim Dalındaki, Yüksek Lisans Tezi 29.04.2015 tarihinde, aşağıdaki jüri tarafından Anadolu Üniversitesi Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliğinin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir.

	Adı-Soyadı	İmza
Üye (Tez Danışmanı) :	Prof. Dr. REFAİL KASIMBEYLİ
Üye	: Yrd. Doç. Dr. ZEHRA KAMIŞLI ÖZTÜRK
Üye	: Doç. Dr. AYDIN SİPAHİOĞLU

Anadolu Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun
tarih ve sayılı kararıyla onaylanmıştır.

Enstitü Müdürü

ÖZET

Yüksek Lisans Tezi

ARAÇ ROTALAMA PROBLEMLERİ İÇİN MATEMATİKSEL MODELLER VE ÇÖZÜM YÖNTEMLERİ

Melis ALPASLAN

Anadolu Üniversitesi

Fen Bilimleri Enstitüsü

Endüstri Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. Refail KASIMBEYLİ

2015, 81 sayfa

Bu çalışmada, klasik araç rotalama problemi, açık uçlu araç rotalama problemi ve bölünmüş talepli araç rotalama problemleri incelenmiş ve bu problemlere yönelik yeni karma tamsayı tek amaçlı ve çok amaçlı modeller geliştirilmiştir. Araç filo türü olarak heterojen filo ele alınmıştır. Öncelikle tek amaçlı modellerde toplam rota maliyeti (kilometre başına oluşan maliyetler) ve araçların depodan çıkma maliyetlerinin en küçüklenmesi hedeflenirken, çok amaçlı modellerde ise bu amaçlara ek olarak, heterojen filolu araç rotalama problemlerinde farklı türlerde araçlar bulunduğundan, literatürde daha önceden ele alınmayan, kullanılan araç türü en küçüklenmesi amaçlanmıştır. Çok amaçlı matematiksel modelin çözümünde ağırlıklandırılmış toplam yöntemi kullanılmıştır. Geliştirilen modeller, öncelikle küçük boyutlu problemler için GAMS paket programı ile çözdürülmüş, büyük boyutlu problemler için ise yasaklı arama algoritması çalışılmıştır. Algoritma, literatürdeki test problemleri üzerinde denenmiş ve elde edilen hesaplamalı sonuçlar kıyaslamalı bir şekilde sunulmuştur.

Anahtar Kelimeler: Modelleme, Araç Rotalama, Ağırlıklandırılmış Toplam, Yasaklı Arama

ABSTRACT

Master of Science Thesis

MATHEMATICAL MODELS AND SOLUTION METHODS FOR VEHICLE ROUTING PROBLEMS

Melis ALPASLAN

Anadolu University

Graduate School of Sciences

Industrial Engineering Program

Advisor: Prof. Dr. Refail KASIMBEYLİ

2015, 81 pages

In this study, classical vehicle routing problem, open vehicle routing problem and split delivery vehicle routing problem are examined and for the regarding problems, new mixed integer single objective and multi-objective mathematical models are developed. In the context of this thesis, heterogeneous fleet vehicles are considered. While in single objective models total routing cost (cost per kilometer) and fixed costs for leaving depot minimization is aimed, in the multi objective mathematical models, because of the availability of different types of vehicles in the heterogenous fleet vehicle routing problems, vehicle type minimization is also aimed which is not considered in literature previously. Weighted sum scalarization method is applied for the multi-objective mathematical models. New models, for the small size of problems are solved in GAMS package software, for the large sized problems, tabu search algorithm is applied. This algorithm is applied to the test problems in the literature and computational results are presented with comparison.

Keywords: Modeling, Vehicle Routing, Weighted Sum, Tabu Search

TEŐEKKÖR

Tezimin hazırlanması sırasında vermiŐ olduđu destek ve yönlendirmelerinden dolayı danışman hocam Sayın Prof. Dr. Refail KASIMBEYLİ' ye ve her zaman yanımda olan aileme teşekkür ederim.

Melis ALPASLAN

Mayıs, 2015

İÇİNDEKİLER

Sayfa

ÖZET.....	i
ABSTRACT.....	ii
TEŞEKKÜR.....	iii
İÇİNDEKİLER	iv
ŞEKİLLER DİZİNİ.....	vii
ÇİZELGELER DİZİNİ	viii
1 GİRİŞ.....	1
2 GEZGİN SATICI PROBLEMİ	4
3 ARAÇ ROTALAMA PROBLEMİ	6
3.1 Araç Rotalama Problemi Türleri.....	7
3.1.1 Klasik araç rotalama problemi (KARP)	7
3.1.2 Heterojen filolu araç rotalama problemi (HFARP)	8
3.1.3 Açık uçlu araç rotalama problemi (AUARP)	9
3.1.4 Heterojen filolu açık uçlu araç rotalama problemi (HFAUARP)....	9
3.1.5 Zaman pencereli araç rotalama problemi	10
3.1.6 Önce dağıt sonra topla araç rotalama problemi	10
3.1.7 Eş zamanlı topla-dağıt araç rotalama problemi	10
3.1.8 Mesafe kısıtlı araç rotalama problemi	11
3.1.9 Dinamik araç rotalama problemleri.....	11
3.1.10 Stokastik araç rotalama problemi.....	11
3.1.11 Çoklu depoya sahip araç rotalama problemi.....	11
3.1.12 Bölünebilir talepli araç rotalama problemi(BTARP).....	11
3.2 Araç Rotalama Problemlerinin Uygulama Alanları.....	12
4 LİTERATÜRDE İNCELENEN MATEMATİKSEL MODELLER.....	13
5 TEZ KAPSAMINDA GELİŞTİRİLEN MATEMATİKSEL MODELLER....	17
5.1 Heterojen Filolu Araç Rotalama Problemi için Geliştirilen Matematiksel Modeller.....	17
5.1.1 HFARP için geliştirilen tek amaçlı matematiksel model	17
5.1.2 HFARP için geliştirilen çok amaçlı matematiksel model	19

5.2	Heterojen Filolu Açık Uçlu Araç Rotalama Problemi için Geliştirilen Matematiksel Modeller	21
5.2.1	HFAUARP için geliştirilen tek amaçlı matematiksel model	21
5.2.2	HFAUARP için geliştirilen çok amaçlı matematiksel model.....	23
5.3	Bölünebilir Talepli Araç Rotalama Problemi için Geliştirilen Matematiksel Modeller	25
5.3.1	BTARP için geliştirilen tek amaçlı matematiksel model	25
5.3.2	BTARP için geliştirilen çok amaçlı matematiksel model	27
6	ARAÇ ROTALAMA PROBLEMİ İÇİN LİTERATÜRDE ÇALIŞILAN KESİN ÇÖZÜM YÖNTEMLERİ.....	30
6.1	Dal Sınır Yöntemi	30
6.2	Kesme Düzlemi Yöntemi.....	30
6.3	Dinamik Programlama	30
6.4	Araç Rotalama Problemi İçin Çalışılan Sezgisel Çözüm Yöntemleri	31
6.4.1	Yapıcı sezgiseller.....	31
6.4.1.1	Clarke ve Wright kazanım algoritması	31
6.4.1.2	En yakın komşuluk yöntemi	32
6.4.1.3	Mole ve Jameson sıralı ekleme sezgiseli	32
6.4.1.4	Christofides, Mingozi ve Toth sıralı ekleme sezgiseli	33
6.4.2	İki aşamalı yöntemler	33
6.4.2.1	Süpürme algoritması	34
6.4.2.2	Fisher ve Jaikumar algoritması	34
6.4.2.3	Bramel ve Simchi-Levi algoritması	35
6.4.2.4	Kesilmiş Dal-Sınır Yöntemi	35
6.4.2.5	Taçyaprağı algoritmaları	35
6.4.2.6	Önce rotala sonra kümele metotları	35
6.4.3	İyileştirici sezgiseller.....	36
7	ARAÇ ROTALAMA PROBLEMİ İÇİN LİTERATÜRDE ÇALIŞILAN METASEZGİSEL YÖNTEMLER.....	37
7.1	Tavlama Benzetimi (TB)	37
7.2	Genetik Algoritma (GA)	38
7.3	Uyarlanabilir Hafıza Prosedürü	40
7.4	Yapay Sinir Ağları (YSA)	40

7.5	Karınca Kolonisi Algoritması (KKA).....	40
7.6	Yasaklı Arama Algoritması (YA).....	42
7.6.1	İki erken (two early) yasaklı arama algoritması.....	44
7.6.2	Osman yasaklı arama algoritması.....	44
7.6.3	Yasaklı rota algoritması.....	44
7.6.4	Taillard algoritması	45
7.6.5	Xu and Kelly algoritması.....	45
7.6.6	Barbarasoğlu ve Özgür algoritması	46
8	GELİŞTİRİLEN MODELLER İÇİN ÇIKAN GAMS SONUÇLARI VE ANALİZİ.....	47
8.1	HFARP için Çıkan GAMS Sonuçları	48
8.2	HFAUARP için Çıkan GAMS Sonuçları	50
8.3	BTARP için Çıkan GAMS Sonuçları	51
9	UYGULANAN YASAKLI ARAMA ALGORİTMASI VE ALGORİTMA SONUCUNDA ELDE EDİLEN SONUÇLARIN ANALİZİ	54
9.1	Uygulanan Yasaklı Arama Yöntemi	54
9.2	Uygulanan Yasaklı Arama Algoritması Sonuçları ve Analizi.....	57
9.2.1	HFARP için uygulanan yasaklı arama algoritması sonuçları ve analizi.....	59
9.2.2	HFAUARP için uygulanan yasaklı arama algoritması sonuçları ve analizi.....	62
9.2.3	BTARP için uygulanan yasaklı arama algoritması sonuçları ve analizi.....	65
10	SONUÇ VE ÖNERİLER	67
11	KAYNAKÇA.....	68

ŞEKİLLER DİZİNİ

Şekil 7.1. Karıncaların bir engelle karşılaştıkları durumda izledikleri yol [33]....	42
Şekil 9.1. İki farklı rotada bulunan müşterilerin swap (yer deęiştirme) hareketi .	55
Şekil 9.2. Yasaklı arama algoritması.....	56

ÇİZELGELER DİZİNİ

8.1. HFARP VE HFAUARP için kullanılan şehir verileri	48
8.2. HFARP VE HFAUARP için kullanılan araç verileri.....	48
8.3. HFARP için geliştirilen tek amaçlı modelin GAMS sonucu	49
8.4. HFARP için geliştirilen çok amaçlı modelin GAMS sonucu	49
8.5. HFAURP için geliştirilen tek amaçlı modelin GAMS sonucu	50
8.6. HFAUARP için geliştirilen çok amaçlı modelin GAMS sonucu	51
8.7. Heterojen filolu BTARP için kullanılan şehir verileri	52
8.8. Heterojen filolu BTARP için kullanılan araç verileri	52
8.9. Heterojen filolu BTARP için geliştirilen tek amaçlı modelin GAMS sonucu	52
8.10. Heterojen filolu BTARP için geliştirilen çok amaçlı modelin GAMS sonucu	53
9.1. Yasaklı arama algoritmasında kullanılan veriler.....	58
9.2. HFARP'nin başlangıç çözümü en yakın komşuluk yöntemi ile oluşturulan algoritma sonuçları.....	60
9.3. HFARP'nin başlangıç çözümü rassal olarak oluşturulan algoritma sonuçları	61
9.4. HFAUARP'nin başlangıç çözümü en yakın komşuluk yöntemi ile oluşturulan algoritma sonuçları.....	62
9.5. HFAUARP'nin başlangıç çözümü rassal olarak oluşturulan algoritma sonuçları	64
9.6. Heterojen filolu BTARP'nin başlangıç çözümü en yakın komşuluk yöntemi ile oluşturulan algoritma sonuçları.....	66

1 GİRİŞ

Araç rotalama problemi (ARP) lojistik, dağıtım problemleri ve ulaştırma kapsamında büyük öneme sahiptir. Çizelgeleme, lojistik, stok yönetimi, üretim planlama gibi alanlarda karşılaşılan en önemli kombinatoriyal eniyileme problemlerinden bir tanesidir. ARP ilk olarak Dantzig ve Ramser [1] tarafından kamyon dağıtım problemi olarak ele alınmıştır. Problemin ortaya çıkmasından sonra bu probleme yönelik birçok farklı çalışma yapılmıştır. Klasik ARP’de amaç, bir ya da birden fazla depodan belirli sayıda müşteriye belirli kısıtlar altında dağıtım veya toplama yapmaktır. Rotalar depodan başlar ve müşterilere dağıtım ya da toplama işlemi yapıldıktan sonra araçlar tekrar depoya dönerler. Rotayı araç kapasitesini aşmadan tamamlamak gerekmektedir.

Bu tez kapsamında farklı araç rotalama problemleri incelenmiştir. Yapılan çalışmada, literatürde yer alan klasik araç rotalama problemi, açık uçlu araç rotalama problemi ve bölünmüş talepli araç rotalama problemi türlerine yönelik incelemeler ve çalışmalar yapılmıştır. Filo türü olarak heterojen filo ele alınmıştır. Heterojen filoda farklı türlerde araçlar bulunmaktadır. Filodaki tüm türlerin kullanılması firmalar için oldukça yüksek maliyetlidir. Çünkü, farklı türlerdeki (farklı kapasitedeki) araçların maliyetleri birbirlerinden farklıdır. Bu maliyetler, bakım ve amortisman maliyetleri, eğer araç başka bir yerden kiralanıyorsa kiralama maliyetleri, depoda farklı türde araçların tutulma maliyetleri vb. maliyetleri içermektedir. Firmalar gerçek hayat problemlerinde, taleplerini karşılayacakları şehirlere ne kadar az tür araç gönderirlerse, toplam maliyetler ve araç maliyetleri de o kadar azalacaktır. Çünkü firmanın ihtiyacı olabilecek tüm türlere sahip olması pek mümkün değildir. Mutlaka başka bir yerden araç kiralama durumu söz konusu olabilmektedir. Bu yüzden araçların tür maliyetleri önemlidir. İncelenen bu problemlere yönelik tek amaçlı ve çok amaçlı modeller geliştirilmiştir. Tek amaçlı modelde toplam rota maliyeti (kilometre başına oluşan maliyetler) en küçüklenmesi ve araçların depodan çıkma maliyetleri en küçüklenirken, çok amaçlı modellerde ise bu amaçlara ek olarak kullanılan araç türünün de en küçüklenmesi hedeflenmiştir. Böylece literatürde gözlemlenen kullanılan araç türü en küçüklenmesi açığının kapatılması amaçlanmıştır.

Literatürde bu problemlere yönelik farklı çözüm yöntemleri çalışılmıştır. Klasik heterojen filolu ARP için Taillard [2] sezgisel sütun oluşturma tekniğini, Gendreau ve ark. [3] yasaklı algoritmasını, Renaud ve Boctor [4] süpürme algoritmasına dayalı yeni bir sezgisel, Tarantilis ve ark. [5] kabul edilebilir eşik değerli metasezgisel bir algoritma, Li ve ark. [6] kayıttan kayıta seyahat algoritmasını (record to record travel algorithm), Brandao [7] deterministik yasaklı arama algoritmasını, Imran ve ark. [8] ise değişken komşuluk tabanlı bir sezgisel algoritma çalışmışlardır. Belfiore ve ark. [9] yayılan arama algoritmasını bir gerçek hayat problemine yönelik uygulamışlardır. Euchı ve Chabchoub [10] hibrit yasaklı arama algoritması, Liu [11] ise genetik algoritma tabanlı hibrit popülasyon sezgiseli çalışmıştır.

Açık uçlu ARP için literatürde probleme yönelik birçok metasezgisel algoritma çalışılmıştır. Brandao [12] yasaklı arama algoritması üzerinde, Yu ve ark. [13] hibrit olarak genetik algoritma ve yasaklı arama algoritması üzerinde çalışmışlardır. Li ve ark. [14] , heterojen filolu açık uçlu araç rotalama problemi üzerinde yasaklı arama algoritmasıyla uyarlanabilir hafıza tabanlı programlama yöntemini, Fleszar ve ark. [15] açık uçlu probleme yönelik etkili bir değişken komşuluk arama yöntemi geliştirmişlerdir. Salari ve ark. [16] tamsayı doğrusal programlamaya dayalı olarak geliştirilmiş bir sezgisel yöntem, MirHassani ve Abolghasemi [17] parçacık sürü eniyileme algoritmasını çalışmışlardır.

Bölünmüş talepli ARP için literatürde farklı çalışmalar bulunmaktadır. Dror ve Trudeau [18] kazanım algoritmasını, Archetti ve ark. [19] yasaklı arama algoritmasını, Jin ve ark. [20] sütun oluşturma tekniğini, Bolduc ve ark. [21] yasaklı arama tekniğini, Silva ve ark. [22] probleme yönelik ötelenmiş yerel arama sezgiselini kullanmışlardır.

Bu çalışmada ise çözüm yöntemi olarak yasaklı arama algoritması çalışılmıştır. Yasaklı arama algoritmasının seçilme nedeni hem ARP problemlerinde çok fazla çalışılan bir algoritma olması, hem de yasaklı arama algoritmasının en önemli özelliklerinden kuvvetlendirme ve çeşitlendirme aşamalarının algoritma içerisinde etkin olarak kullanılabilmesidir.

Hesaplama teorisine göre, deterministik olmayan polinom zamanlı problemler sınıfında yer alan problemler karmaşıklıkça çözüm süresi üssel

olarak artmaktadır. NP-zor problemler ise, NP sınıfındaki problemler kadar zordur fakat polinom (P) zamanlı problemler sınıfına indirgenebilir. ARP, NP-zor problemler sınıfındadır. ARP, NP-zor problem sınıfında yer aldığından şehir sayısı 10'dan fazla olduğu zaman standart yazılımlar ile problem çözülememektedir. Bu yüzden büyük test problemlerine yönelik yasaklı arama algoritması çalışılmıştır. Yasaklı arama algoritması çok amaçlı modele de uyarlanmıştır. Literatürde çok amaçlı probleme yönelik bir yasaklı arama algoritması da bulunmadığından probleme yönelik yeni bir bakış açısı kazandırıldığı düşünülmektedir.

2 GEZGİN SATICI PROBLEMİ

Gezgin satıcı problemi (GSP) literatürde en çok bilinen ve çalışılan kombinatoryal eniyileme problemlerden bir tanesidir. Gezgin satıcı probleminde satıcı bulunduğu başlangıç şehriden başlayarak kalan $n-1$ adet şehri ziyaret eder ve başlangıç konumuna geri döner. Bu yüzden arama uzayının boyutu $n!$ kadardır. Her i ve j şehri arasındaki mesafeler bir uzaklık matrisi ile gösterilir. Her şehre mutlaka ve sadece bir kere uğranılmalıdır. Problemden amaç bu kısıtları göz önüne alarak en kısa veya en küçük maliyetli turu bulmaktır. ARP ise m -GSP olarak literatürde yer almaktadır [23]. GSP’de tek satıcı olmasına rağmen ARP’de birden fazla gezgin (araç) bulunmaktadır. GSP’ nin temel matematiksel modeli şu şekildedir [24] :

$G = \{V, A\}$: bir serim olsun.

$V = \{1, \dots, n\}$: düğüm kümesi

A : ayrıt kümesi

1 noktası depoyu göstermektedir.

Parametre

c_{ij} : i şehriden j şehrine gitme maliyeti

$c_{ii} = 0$ kabul edilir.

Karar Değişkeni

$x_{ij} : \begin{cases} 1, & \text{eğer } i \text{ şehriden } j \text{ şehrine gidiliyorsa} \\ 0, & \text{diğer durumda} \end{cases}$

$$\text{En küçük } z = \sum_i \sum_{j \neq i} c_{ij} x_{ij} \quad (2.1)$$

Kısıtlar:

$$\sum_{j=1}^n x_{ij} = 1 \quad \forall i, i \neq j \quad (2.2)$$

$$\sum_{i=1}^n x_{ij} = 1 \quad \forall j, j \neq i \quad (2.3)$$

$$\sum_{i,j \in S} x_{ij} \leq |S| - 1, \quad S \subset V \quad 2 < |S| < n - 1 \quad (2.4)$$

$$x_{ij} \in \{0,1\} \tag{2.5}$$

Kısıt (2.1) amaç fonksiyonudur ve toplam rota maliyetinin en küçüklenmesini hedeflemektedir. Kısıt (2.2) her şehirden mutlaka bir yere gidilmelidir ifadesini belirtir. Kısıt (2.3)'e göre her şehre mutlaka bir yerden gelinmelidir. Kısıt (2.4) alttır engelleme kısıtıdır.

3 ARAÇ ROTALAMA PROBLEMİ

Araç rotalama problemi (ARP) ilk olarak Dantzig ve Ramser tarafından 1959 yılında çalışılmıştır [1]. Problem, kamyon dağıtım problemi olarak analiz edilmiştir. En genel haliyle, ARP’de bir yol ağı bulunmaktadır ve bu ağda araçlar ile müşterilerden mal alımı ya da mal toplanması işlemleri yapılmaktadır. Klasik araç rotalama probleminde, verilen n tane şehrin belirli talep miktarları vardır. Depo noktasının talebi 0 olarak kabul edilir. Amaç fonksiyonu temel olarak rota maliyeti en küçüklenmesi şeklinde tanımlanır. Klasik ARP’nin temel kısıtları şunlardır: Her şehre mutlaka ve sadece bir kere uğranılmalıdır. Her müşteri noktası k araç rotasından sadece bir tanesinde mutlaka yer almalıdır. Müşterilerin talepleri mutlaka karşılanmak zorundadır. Her rota, mutlaka depodan başlar ve depoda sona erer. Aracın rotasında bulunan şehirlerin talepleri toplamı, o aracın kapasitesini aşamaz.

Gerçek hayat uygulamalarında ARP'nin sahip olduğu farklı kısıtlar da bulunmaktadır. Bu kısıtlar: müşteriler ile ilgili kısıtlar (stokastik talepli müşteriler ve deterministik talepli müşteriler), ürünlerin dağıtımının yapılması ya da toplama işleminin yapılması (hem dağıtım hem toplama işlemi de yapılabilir) ve sürücüler ile ilgili kısıtlar (çalışma saatleri ve yasal kısıtlamalar gibi).

ARP’de dikkate alınan temel amaçlar:

- Toplam rota maliyetini en küçüklemek.
- Araçların maliyetlerini (sabit maliyetler ya da değişken maliyetler) en küçüklemek.
- Talebi karşılanamayan müşteriler varsa bu müşterilerden gelecek olan toplam ceza maliyetlerini en küçüklemek.
- Toplam rota zamanını en küçüklemek.
- Araç sayısını en küçüklemek.
- Rotaların sürelerini en küçüklemek gibi amaçlardır.

3.1 Araç Rotalama Problemi Türleri

3.1.1 Klasik araç rotalama problemi (KARP)

Araç rotalama probleminin en çok çalışılan türlerinden biridir. Rotalar depoda başlar ve depoda sona erer. Aynı kapasiteye sahip araçlar (homojen filo) depoda bulunmaktadır. Müşteri talepleri deterministiktir. Her şehre mutlaka ve sadece bir araç uğramalıdır ve şehrin talebini tamamıyla karşılamalıdır. KARP şu şekilde açıklanabilir:

$G = \{V, A\}$: bir serim olsun.

$V = \{0, \dots, n\}$: düğüm kümesi

A : ayrıt kümesi

0 noktası depoyu göstermektedir. Bazı formülasyonlarda $n+1$ düğümü depo olarak gösterilmektedir. Kalan düğümler müşteri noktalarıdır.

Parametreler

c_{ij} : i şehrinde j şehrine gitme maliyeti

d_{ij} : i şehri ile j şehri arasındaki mesafe

$c_{ii} = 0$ kabul edilir.

Karar Değişkeni

$x_{ij} = \begin{cases} 1, & \text{eğer } i \text{ şehrinde } j \text{ şehrine gidiliyorsa} \\ 0, & \text{diğer durumda} \end{cases}$

Eğer G yönlü bir serim ise, c maliyet matrisi asimetriktir yani $c_{ij} \neq c_{ji}$.

Bu durumda probleme asimetrik klasik araç rotalama problemi (AKARP) denir.

Eğer $c_{ij} = c_{ji}$ olursa probleme simetrik klasik araç rotalama problemi (SKARP)

denir. Bu problemde A kümesinin yerini yönsüz kenar kümesi olan E kümesi alır.

Laporte [25] tarafından yapılan formülasyon izleyen şekilde verilebilir:

$$\text{En küçük } z = \sum_{i \neq j} d_{ij} x_{ij} \quad (3.1)$$

Kısıtlar:

$$\sum_j x_{ij} = 1 \quad , \quad \forall i \in V \quad (3.2)$$

$$\sum_i x_{ij} = 1 \quad , \quad \forall j \in V \quad (3.3)$$

$$\sum_{i,j \in S} x_{ij} \geq |S| - v(S) \quad , \quad \{S: S \in V \setminus \{1\}, |S| \geq 2\} \quad (3.4)$$

$$x_{ij} \in \{0,1\} \quad \forall i,j \in E ; i \neq j \quad (3.5)$$

Amaç fonksiyonu (3.1): toplam mesafeyi en küçükmektir. Kısıt (3.2)'ye göre her şehirden mutlaka bir başka şehre gidilmelidir. (3.3)'e göre her şehre mutlaka bir yerden gelinmelidir. (3.2) ve (3.3) kısıtları araç akış kısıtlarıdır. (3.4) alttut engelleme kısıtıdır.

3.1.2 Heterojen filolu araç rotalama problemi (HFARP)

Literatürde, heterojen filolu araç rotalama problemine ilk olarak Kirby tarafından 1959 yılında tek sayfalık bir çalışma ile değinilmiştir [26]. Heterojen filolu araç rotalama probleminde filoda farklı türde (farklı kapasitede) araçlar bulunmaktadır. Her türe ait araç sayısı sınırsız ya da sınırlı sayıda olabilir. Baldacci ve ark. [27] bu problemi, araçların sabit maliyetlerinin olup olmamasına, rotalama maliyetlerinin gidilen yere bağlı olup olmamasına ve filo boyutunun sınırlı ya da sınırsız olmasına göre sınıflandırmışlardır.

Filonun sınırsız olduğu durumda heterojen filoyu temel alarak, literatürde bu konu üzerinde yapılan detaylı ilk çalışma filo büyüklüğü ve karma araç rotalama problemi olarak Golden ve ark. [28] tarafından yapılmıştır. Bu çalışmada araç rotalamayı üç farklı sınıfta incelemişlerdir:

Klasik Araç Rotalama Problemi: Kapasiteleri aynı olan ve önceden sayısı bilinen araçlar bulunmaktadır. Araçların sabit maliyetleri göz önüne alınmaksızın sadece kilometre başına sahip olunan maliyetlere dikkat çekilir. Okul servislerini rotalama, gazete dağıtımı, farklı endüstrilerdeki kamyon dağıtım problemleri vb. örnekler verilmiştir.

Filo Büyüklüğü Problemi: Filo boyutunun müşteri taleplerine yeterli olmadığı durumda verilecek olan önemli karar kaç tane aracın satın alınacağı ya da

kiralanacağı durumudur [28]. Bu problemde sabit araç maliyetleri ve araçların kilometre başına değişken maliyetleri göz önüne alınmalıdır.

Filo Büyüklüğü ve Karışımı Problemi: Bu problemde tüm araçlar aynı maliyete sahiptir ve karma araçlardan oluşan filonun optimize edilmesi amaçlanmaktadır.

3.1.3 Açık uçlu araç rotalama problemi (AUARP)

Klasik ARP'den farkı, rotaların depodan başlaması fakat bir müşteri noktasında sona ermesidir. Bu problem türüne literatürde yapılan ilk tanımlama Schrage [29] tarafından gerçekleştirilmiştir. Havayolu kargo rotalama problemi için bir uygulama yapılmıştır. Daha sonra probleme yönelik ilgi Sariklis ve Powell [30] tarafından yapılan çalışma ile hız kazanmıştır. AUARP'nin klasik ARP'den farkı, rotaların depodan başlayıp bir müşteri noktasında sona ermesidir. Tur depoda tamamlanmadığından açık uçlu olarak tanımlanmıştır. Araçların depoya dönmemesi genelde firmaların kendi araç filosundaki araçlarının hepsinin kendisine ait olmaması (dışarıdan araç kiralama durumu olduğunda) ya da, araçların müşteri taleplerini karşılamada yetersiz olduğu durumlarda olur. Eğer araçlar başka bir firmadan kiralandıysa bu kiralanan araçların firmanın kendi deposuna dönme zorunluluğu bulunmamaktadır. Firma, toplam rota maliyeti en küçüklenmesi yanında müşteri taleplerine tamamıyla yetebilmek amacıyla kiralaması gereken minimum araç sayısını da elde etmeye çalışacaktır. Kilometre başına kat edilen mesafede araçların sahip olduğu maliyetlerin yüksek olduğu durumlarda, araçların depoya dönmemesi maliyetleri de önemli ölçüde azaltacaktır. [30].

3.1.4 Heterojen filolu açık uçlu araç rotalama problemi (HFAUARP)

Açık uçlu araç rotalama probleminin önemli bir türüdür. Endüstri alanında ve servis problemlerinde oldukça kullanılmaktadır. Problem literatüre Li ve ark. [14] tarafından yapılan çalışma ile kazandırılmıştır. Bu çalışmada düzenlenmiş yasaklı arama algoritması ile multi start adaptive (çoklu uyarlanabilir) bellek programlama metasezgiseli önerilmiştir. Bu algoritmadaki amaçları arama

hafızasını kullanarak daha etkin sonuçlar üretmektir. Adaptif bellek programlama (ABP) Taillard ve ark. [31] tarafından oluşturulmuştur. ABP ile, kombinatoriyal eniyileme problemlerinin daha iyi çözümleri, algoritmaların (yasaklı arama, genetik algoritma vb.) iterasyonları boyunca elde edilen diğer iyi çözümlerin farklı bileşenlerinin birleştirilmesi ile elde edilebilir mantığına dayanmaktadır. HFAUARP'de farklı türlerde araçlar ve bu araçların belirli sayıları bulunmaktadır [32]. Araçlar depoya dönmemektedir. Rotalarını herhangi bir müşteri noktasında tamamlarlar.

3.1.5 Zaman pencereci araç rotalama problemi

Bu problemde müşteri talepleri belirli bir zaman penceresi $[a_i, b_i]$ içerisinde karşılanmalıdır. Burada a_i : i . müşteriye servisin başlaması gereken zamanı, b_i : i . müşteriden en geç çıkma zamanı olarak belirtilmektedir. Araçlar depodan ayrıldıktan sonra, t_{ij} (i şehirden j şehrine gitmek için harcanan zaman) kadarlık bir seyahat süresinden sonra müşteri noktasına uğrayarak s_i kadarlık bir sürede müşteriye hizmet verirler.

3.1.6 Önce dağıt sonra topla araç rotalama problemi

Bu problemde müşteri kümeleri iki ayrı altkümeye ayırır. İlk altküme dağıtım yapılacak müşteriler, diğer altküme ise geri toplama yapılacak müşteri grubudur. Eğer rotada, her iki tür müşteri de bulunuyorsa, öncelik dağıtım yapılacak müşterilere verilmektedir.

3.1.7 Eş zamanlı topla-dağıt araç rotalama problemi

Her müşterinin d_i ve p_i olmak üzere iki tane parametresi bulunmaktadır. d_i : dağıtım yapılacak ürünlerin miktarlarını, p_i : toplama yapılacak ürün miktarlarını belirtmektedir. Bu ürünler homojendir. Bazı durumlarda tek bir talep miktarı olarak, $d_i = d_i - p_i$ şeklinde de gösterilebilir [33]. Her müşteri noktası için, dağıtımın toplamadan önce yapılması varsayılmaktadır.

3.1.8 Mesafe kısıtlı araç rotalama problemi

Rotalara atanan her bir araç için gidebilecekleri maksimum mesafe kısıtı bulunmaktadır. Araçlar gidebilecekleri bu mesafeyi aşamaz.

3.1.9 Dinamik araç rotalama problemleri

Süreç içerisinde birden meydana gelen değişimleri içerir. Bu değişimler; müşteri taleplerindeki değişim, farklı talep noktalarının oluşması, aracın gideceği yolun durumundaki değişim (yolun kapanması ya da tadilat durumu) vb.

3.1.10 Stokastik araç rotalama problemi

Deterministik ARP'nin bir uzantısıdır fakat bu problemdeki bazı yapılar rassaldır. Problem, stokastik müşterili, stokastik talepli ya da stokastik zamanlı olabilir. Deterministik durumda sağlanan ve geçerli olan durumlar bu problemde sağlanamadığından bu problemi çözmek daha zordur.

3.1.11 Çoklu depoya sahip araç rotalama problemi

Problemde birden çok depo bulunmaktadır. Araçlar bu depolardan hareket etmektedir. Depoların ve müşterilerin lokasyonları önceden bilinmektedir. Araçlar hangi depodan çıkıyorsa o depoya geri dönmek zorundadır.

3.1.12 Bölünebilir talepli araç rotalama problemi (BTARP) :

Müşterilerin talebi birden fazla aracın uğraması ile karşılanmaktadır [18]. Bölünmüş talebe sahip problemde her müşterinin talebinin bölünmesi zorunluluğu yoktur. Mutlaka en az bir müşterinin talebi bölünüyorsa bu problem altında incelenebilir.

BTARP, ilk olarak Dror ve Trudeau tarafından 1989 yılında çalışılmıştır [18]. Bu problemde müşteri talebi, birden fazla araç ile karşılanabilir. Probleme

yönelik kazanım algoritmasını analiz ederek bir sezgisel algoritma önermişlerdir. Dror ve ark. [34] tarafından, probleme yönelik geçerli eşitsizlikler oluşturulmuştur. Problem tamsayılı doğrusal programlama olarak formüle edilmiştir.

3.2 Araç Rotalama Problemlerinin Uygulama Alanları

Gerçek hayat problemlerinde en fazla karşılaşılan kombinatoriyal eniyileme problemlerinden bir tanesidir. Uygulama alanı oldukça fazladır. Benzin veya mazot dağıtımı, ürünlerin sevkiyatı, süt dağıtımı ve toplanması, ana depodan ya da depolardan ürünlerin dağıtılması, posta hizmetleri, havayolu ile yolcu ve ürün taşınması vb. [35] örnekler verilebilir.

4 LİTERATÜRDE İNCELENEN MATEMATİKSEL MODELLER

Bu bölümde literatürde problemlere yönelik incelenen matematiksel modellerden birkaçı verilmiştir.

Gheysens ve ark. [36] , HFARP'yi şu şekilde modellemişlerdir:

Kümeler:

i, j : şehirlerin kümesi $i, j = 0, \dots, n$

T : araç türü sayısı

Parametreler:

Q_k : k . türdeki aracın kapasitesi ($Q_1 < Q_2 < \dots < Q_T$)

f_k : k . aracın sabit maliyeti (depodan çıkma maliyeti) ($f_1 < f_2 < \dots < f_T$)

d_j : j . müşterinin talebi

c_{ij} : i müşterisinden j müşterisine gitme maliyeti ($c_{ij} = c_{ji}$)

Karar Değişkenleri:

x_{ij}^k : $\begin{cases} 1, & \text{eğer } k \text{ türündeki araç } i \text{ şehirden } j \text{ şehrine gidersen} \\ 0, & \text{diğer durumda} \end{cases}$

y_{ij} : i şehirden j şehrine giden akış miktarı

Amaç Fonksiyonu:

$$\text{En küçük } z = \sum_{k=1}^T f_k \sum_{j=1}^n x_{0j}^k + \sum_{k=1}^T \sum_{i=0}^n \sum_{j=0}^n c_{ij} x_{ijk} \quad (4.1)$$

Kısıtlar:

$$\sum_{k=1}^T \sum_{i=0}^n x_{ij}^k = 1 \quad j = 1, \dots, n \quad (4.2)$$

$$\sum_{i=0}^n x_{ij}^k - \sum_{l=0}^n x_{jl}^k = 0 \quad j = 0, \dots, n; k = 1, \dots, T \quad (4.3)$$

$$\sum_{i=0}^n y_{ij} - \sum_{l=0}^n y_{jl} = d_j \quad j = 1, \dots, n \quad (4.4)$$

$$y_{0j} \leq \sum_{k=1}^T Q_k x_{0j}^k \quad j = 1, \dots, n \quad (4.5)$$

$$y_{ij} \leq M \sum_{k=1}^T x_{ij}^k \quad i \neq j = 0, \dots, n \quad (4.6)$$

$$y_{ij} \geq 0 \quad i \neq j = 0, \dots, n \quad (4.7)$$

$$x_{ij}^k \in \{0,1\} \quad i \neq j = 0, \dots, n \quad k = 1, \dots, T \quad (4.8)$$

Amaç fonksiyonu (4.1)'de amaç, toplam rota maliyetini ve aracın sabit maliyetlerini en küçüklemektir. $\sum_{j=1}^n x_{0j}^k$ ifadesi filoda yer alan toplam k adet aracı göstermektedir. Kısıt (4.2)'ye göre her j şehrine (depo hariç) mutlaka bir araç bir yerden gelmelidir. Kısıt (4.3) araç akış kısıtıdır. Eğer k aracı i şehirden j şehrine geldiyse, mutlaka j şehirden çıkıp bir l şehrine gitmelidir. Kısıt (4.4), talebin karşılanması kısıtıdır. Kısıt (4.5)' e göre y_{0j} rotadaki toplam yükü ifade ettiğinden, bu yük miktarı rotadaki araç kapasitesini aşamaz. (4.6)'ya göre eğer i 'den j 'ye hiçbir araç gitmezse, bu iki şehir arasında dağıtım yapılmamıştır demektir. M çok büyük bir sayıyı ifade etmektedir. Eğer $\sum_{k=1}^T x_{ij}^k = 1$ olursa bu kısıt gereksiz olur.

Golden ve ark. [28] problemi şu şekilde modellemiştir:

Kümeler:

i, j : şehirlerin kümesi $i, j = 0, \dots, n$

T : araç türü sayısı

Parametreler:

Q_k : k . türdeki aracın kapasitesi ($Q_1 < Q_2 < \dots < Q_T$)

f_k : k . aracın sabit maliyeti (depodan çıkma maliyeti) ($f_1 < f_2 < \dots < f_T$)

d_j : j . müşterinin talebi

c_{ij} : i müşterisinden j müşterisine gitme maliyeti ($c_{ij} = c_{ji}$)

r_i : i . müşterinin ürün akış miktarı

Karar Değişkeni:

$x_{ij}^k: \begin{cases} 1, & \text{eğer } k \text{ türündeki araç } i \text{ şehirden } j \text{ şehrine giderse} \\ 0, & \text{diğer durumda} \end{cases}$

$$\text{En küçük } z = \sum_{k=1}^T f_k \sum_{j=1}^n x_{0j}^k + \sum_{k=1}^T \sum_{i=0}^n \sum_{j=0}^n c_{ij} x_{ijk} \quad (4.9)$$

Kısıtlar:

$$\sum_{k=1}^T \sum_{i=0}^n x_{ij}^k = 1 \quad j = 1, \dots, n \quad (4.10)$$

$$\sum_{i=0}^n x_{ip}^k - \sum_{j=0}^n x_{pj}^k = 0 \quad k = 1, \dots, T, \quad p = 1, \dots, n \quad (4.11)$$

$$r_0 = 0 \quad (4.12)$$

$$r_j - r_i \geq (d_j + a_T) \sum_{k=1}^T x_{ij}^k - a_T \quad i = 0, \dots, n, \quad j = 1, \dots, n \quad (4.13)$$

$$r_j \leq \sum_{k=1}^T \sum_{i=0}^n a_k x_{ij}^k \quad (4.14)$$

$$x_{ij}^k \in \{0,1\} \quad \forall i, j, k \quad j = 1, \dots, n \quad (4.15)$$

Amaç fonksiyonunda amaç, toplam rota maliyetini ve aracın sabit maliyetlerini en küçükmektir. Kısıt (4.10)'a göre her j şehrine (depo hariç) mutlaka bir araç bir yerden gelmelidir. Kısıt (4.11) araç akış kısıtıdır. Kısıt (4.12) deponun ürün akış miktarının 0 olduğunu belirtmektedir. (4.13), i depo dahil, j depo hariç olmak üzere r_i ile talepler arasındaki bağlantıyı göstermektedir. Kısıt (4.14)'e göre bir müşterinin birikimli talebi, o müşteriye servis yapan aracın kapasitesi ile sınırlıdır. (4.13) ve (4.14) alttır engelleme kısıtı olarak da çalışmaktadır. Benzer formülasyonlar da literatürde bulunmaktadır [37].

Dror ve ark. [34] tarafından yapılan BTARP için yapılan formülasyon:

$G=(N,A)$: bir serim

$N=\{0, \dots, n\}$: müşteri noktaları (şehirler)

$A=\{(i, j): i, j \in N, i \neq j\}$: ayrıt kümesi

0: depo noktası

m : depodaki araç sayısı ve bu sayı $[\bar{m}, \underline{m}]$ aralığındadır ve $1 \leq \bar{m} \leq \underline{m} \leq n$.

Araçların kapasiteleri aynı ya da farklı olabilir.

Parametreler:

Q_v : v . aracın kapasitesi

q_i : i . şehrin talebi , $q_i \leq \max_v \{Q_v\}$

c_{ij} : i şehirden j şehrine gitme maliyeti

Karar Değişkenleri:

x_{ijv} : $\begin{cases} 1, \text{ eğer } v. \text{ araç } i \text{ şehirden } j \text{ şehrine gittiyse} \\ 0, \text{ diğer durumda} \end{cases}$

y_{iv} : v . aracın i . şehrin talebini karşıladığı oran

$$\text{Enküçük } z = \sum_{i=0}^n \sum_{j=0}^n \sum_{v=1}^{\bar{m}} c_{ij} x_{ijv} \quad (4.16)$$

Kısıtlar:

$$\sum_{i=0}^n x_{ikv} - \sum_{j=0}^n x_{kjh} = 0 \quad k = 0, \dots, n; v = 1, \dots, \bar{m} \quad (4.17)$$

$$\sum_{v=1}^{\bar{m}} y_{ij} = 1 \quad i = 1, \dots, n \quad (4.18)$$

$$\sum_{i=1}^n q_i y_{iv} \leq Q_v \quad v = 1, \dots, \bar{m} \quad (4.19)$$

$$\sum_{j=0}^n x_{ijv} \geq y_{iv} \quad i = 1, \dots, n ; v = 1, \dots, \bar{m} \quad (4.20)$$

$$\sum_{i,j \in S} x_{ij} \leq |S| - 1 \quad S \subset N \setminus \{0\}; 1 \leq |S| \leq n - 1 \quad (4.21)$$

$$\sum_{v=1}^{\bar{m}} \sum_{j=0}^n x_{ijv} \geq \sum_{v=1}^{\bar{m}} y_{iv} = 1 \quad (4.22)$$

$$x_{ijv} \in \{0,1\} \quad i, j = 0, \dots, n ; v = 1, \dots, \bar{m} \quad (4.23)$$

$$0 \leq y_{iv} \leq 1 \quad i = 1, \dots, n; v = 1, \dots, \bar{m} \quad (4.24)$$

Amaç fonksiyonu (4.16), toplam rota maliyetini en küçüklemeyi hedeflemektedir.

Kısıt (4.17), araç akış kısıtıdır. Kısıt (4.18), şehrin talebinin karşılanan oranları

toplamının mutlaka 1'e eşit olması gerektiğini ifade etmektedir. Kısıt (4.19),

şehrin talebinin, araç tarafından karşılanan oran ile çarpımının asla kapasiteyi

aşamayacağını belirtmektedir. Kısıt (4.20), y_{iv} oranının en fazla 1 olabileceğini

belirtmektedir. Kısıt (4.21), alttur engelleme kısıtıdır. Kısıt (4.22), bir şehre birden

fazla uğrama olabileceğini ifade etmektedir.

5 TEZ KAPSAMINDA GELİŞTİRİLEN MATEMATİKSEL MODELLER

Bu bölümde, heterojen filolu araç rotalama problemi, heterojen filolu açık uçlu araç rotalama problemi ve heterojen filolu bölünebilir talepli araç rotalama problemi için geliştirilen tek amaçlı ve çok amaçlı matematiksel modeller verilmiştir.

5.1 Heterojen Filolu Araç Rotalama Problemi için Geliştirilen Matematiksel Modeller

5.1.1 HFARP için geliştirilen tek amaçlı matematiksel model

Depo noktası 1 olarak alınmıştır.

Kümeler

i, j : şehirlerin kümesi $i, j = 1, \dots, n$

t : farklı türdeki araçların kümesi $t = 1, \dots, T$

v : farklı türlere ait araç sayısı $v = 1, \dots, A_t$

Parametreler

Q_t : t . tür aracın kapasitesi

c_{ij} : i şehirden j şehrine gidilen mesafe

A_t : t . türe ait araç sayısı

d_i : i . şehrin talebi

M_{tv} : t . türün v . aracının kullanılmasının maliyeti

p_t : t . türün kilometre başına oluşan rota maliyeti

Karar Değişkenleri

$x_{ijtv} = \begin{cases} 1, & t. \text{ türün } v. \text{ aracı } i \text{ şehirden } j \text{ şehrine giderse} \\ 0, & \text{diğer durumda} \end{cases}$

$f_{tv} = \begin{cases} 1, & t. \text{ türün } v. \text{ aracı kullanılıyorsa} \\ 0, & \text{diğer durumda} \end{cases}$

u_i ve u_j altturda kullanılan pozitif değişken olarak tanımlıdır.

Amaç fonksiyonu:

$$\text{En küçük } z = \sum_{i=1}^n \sum_{j=1}^n \sum_{t=1}^T \sum_{v=1}^{A_t} c_{ij} p_t x_{ijtv} + \sum_{t=1}^T \sum_{v=1}^{A_t} M_{tv} f_{tv} \quad (5.1)$$

Kısıtlar:

$$\sum_{i=1}^n \sum_{j \neq i}^n \sum_{v=1}^{A_t} d_i x_{ijtv} \leq Q_t \quad \forall t \quad (5.2)$$

$$\sum_{j=1}^n \sum_{v=1}^{A_t} x_{1jtv} \leq A_t \quad \forall t \quad (5.3)$$

$$\sum_{i \neq k}^n x_{iktv} - \sum_{j \neq k}^n x_{kjtv} = 0 \quad \forall k = 1, \dots, n, \forall t, \forall v \quad (5.4)$$

$$\sum_{i=1}^n \sum_{t=1}^T \sum_{v=1}^{A_t} x_{ijtv} = 1 \quad \forall j = 1, \dots, n, j \neq i \quad (5.5)$$

$$u_i - u_j + n \sum_{t=1}^T \sum_{v=1}^{A_t} x_{ijtv} \leq n - 1 \quad \forall i, j = 2, \dots, n, i \neq j \quad (5.6)$$

$$\sum_{i=1}^n x_{i1tv} = f_{tv} \quad \forall t, \forall v \quad (5.7)$$

$$\sum_{i=1}^n \sum_{j=1}^n x_{ijtv} \leq M * f_{tv} \quad \forall t, \forall v \quad (5.8)$$

$$x_{ijtv}, f_{tv} \in \{0,1\} \quad i, j = 1, \dots, n ; t = 1, \dots, T; v = 1, \dots, A_t \quad (5.9)$$

Amaç fonksiyonu olan kısıt (5.1), toplam rota maliyetinin ve araçların depodan çıkma maliyetlerinin toplamının en küçüklenmesini ifade etmektedir. Kısıt (5.2) ise kapasite kısıtını belirtmektedir. Bu kısıta göre, aracın (t . türün v . aracı) rotasında bulunan şehirlerin talepleri toplamı o aracın kapasitesini aşamaz. Kısıt (5.3), her t tür araç için, depodan o türe ait araç sayısından fazla araç çıkamayacağını belirtmektedir. Çünkü bütün türlerin bütün araçları depodan çıkarsa, araçlar gidecekleri yerleri önceden bildiği için yükleme ona göre yapılır ve araç rotalamının önemi kalmaz. Kısıt (5.4) literatürde de var olan araç akış kısıtıdır. Eğer araç i şehirden k şehrine gittiyse o k şehirden de bir başka j ($j \neq i$) şehrine gitmelidir. Kısıt (5.5)'e göre, her j şehrine mutlaka herhangi bir i şehirden

gelinmelidir. Kısıt (5.6) alttur engelleme kısıtıdır. Miller ve ark. [38] tarafından geliştirilmiştir. Kısıt (5.7)'ye göre eğer t . türün v . aracı kullanılıyorsa $f_{tv} = 1$ olur ve bu araç depodan çıkıp kullanıldığı için mutlaka depoya dönmelidir. Kısıt (5.8), x karar değişkeni ile f karar değişkeni arasındaki bağlantıyı sağlayan kısıttır. M büyük bir sayıdır (kıyaslamalı olarak).

5.1.2 HFARP için geliştirilen çok amaçlı matematiksel model

Depo noktası 1 olarak alınmıştır.

Kümeler

i, j : şehirlerin kümesi $i, j = 1, \dots, n$

t : farklı türdeki araçların kümesi $t = 1, \dots, T$

v : farklı türlere ait araç sayısı $v = 1, \dots, A_t$

Parametreler

Q_t : t . tür aracın kapasitesi

c_{ij} : i şehirden j şehrine gidilen mesafe

A_t : t . türe ait araç sayısı

d_i : i . şehrin talebi

M_{tv} : t . türün v . aracının kullanılmasının maliyeti

p_t : t . türün kilometre başına oluşan rota maliyeti

u_i ve u_j altturda kullanılan pozitif değişken olarak tanımlıdır.

Karar Değişkenleri

x_{ijtv} : $\begin{cases} 1, & t. \text{ türün } v. \text{ aracı } i \text{ şehirden } j \text{ şehrine giderse} \\ 0, & \text{diğer durumda} \end{cases}$

f_{tv} : $\begin{cases} 1, & t. \text{ türün } v. \text{ aracı kullanılıyorsa} \\ 0, & \text{diğer durumda} \end{cases}$

h_t : $\begin{cases} 1, & t. \text{ tür araç kullanılıyorsa} \\ 0, & \text{diğer durumda} \end{cases}$

Amaç fonksiyonları:

$$\text{En küçük } z_1 = \sum_{i=1}^n \sum_{j=1}^n \sum_{t=1}^T \sum_{v=1}^{A_t} c_{ij} p_t x_{ijtv} + \sum_{t=1}^T \sum_{v=1}^{A_t} M_{tv} f_{tv} \quad (5.10)$$

$$\text{En küçük } z_2 = \sum_{t=1}^T h_t \quad (5.11)$$

Kısıtlar:

$$\sum_{i=1}^n \sum_{j \neq i}^n \sum_{v=1}^{A_t} d_i x_{ijtv} \leq Q_t \quad \forall t \quad (5.12)$$

$$\sum_{j=1}^n \sum_{v=1}^{A_t} x_{1jtv} \leq A_t \quad \forall t \quad (5.13)$$

$$\sum_{i \neq k}^n x_{ikt v} - \sum_{j \neq k}^n x_{kjt v} = 0 \quad \forall k = 1, \dots, n, \forall t, \forall v \quad (5.14)$$

$$\sum_{i=1}^n \sum_{t=1}^T \sum_{v=1}^{A_t} x_{ijtv} = 1 \quad \forall j = 1, \dots, n, j \neq i \quad (5.15)$$

$$u_i - u_j + n \sum_{t=1}^T \sum_{v=1}^{A_t} x_{ijtv} \leq n - 1 \quad \forall i, j = 2, \dots, n, i \neq j \quad (5.16)$$

$$\sum_{i=1}^n x_{i1tv} = f_{tv} \quad \forall t, \forall v \quad (5.17)$$

$$\sum_{i=1}^n \sum_{j=1}^n x_{ijtv} \leq M * f_{tv} \quad \forall t, \forall v \quad (5.18)$$

$$\sum_v^{A_t} f_{tv} \leq M * h_t \quad \forall t \quad (5.19)$$

$$x_{ijtv}, f_{tv}, h_t \in \{0,1\} \quad i, j = 1, \dots, n ; t = 1, \dots, T; v = 1, \dots, A_t \quad (5.20)$$

İlk amaç fonksiyonu olan kısıt (5.10), toplam rota maliyetinin ve araçların depodan çıkma maliyetlerinin toplamının en küçüklenmesini ifade etmektedir. İkinci amaç fonksiyonu (5.11) ise, kullanılan araç türlerini en küçüklemeyi amaçlamaktadır. Kısıt (5.12) ise kapasite kısıtını belirtmektedir. Bu kısıta göre, aracın (t . türün v . aracı) rotasında bulunan şehirlerin talepleri toplamı o aracın kapasitesini aşamaz. Kısıt (5.13), her t tür araç için, depodan o türe ait araç sayısından fazla araç çıkamayacağını belirtmektedir. Çünkü bütün türlerin bütün araçları depodan çıkarsa, araçlar gidecekleri yerleri önceden bildiği için yükleme ona göre yapılır ve

araç rotalamasının önemi kalmaz. Kısıt (5.14) literatürde de var olan araç akış kısıtıdır. Eğer araç i şehirden k şehrine gittiyse o k şehirden de bir başka j ($j \neq i$) şehrine gitmelidir. Kısıt (5.15)'e göre, her j şehrine mutlaka herhangi bir i şehirden gelmelidir. Kısıt (5.16) alttur engelleme kısıtıdır [38]. Kısıt (5.17)'ye göre eğer t . türün v . aracı kullanılıyorsa $f_{tv} = 1$ olur ve bu araç depodan çıkıp kullanıldığı için mutlaka depoya dönmelidir. Kısıt (5.18), x karar değişkeni ile f karar değişkeni arasındaki bağlantıyı sağlayan kısıttır. M büyük bir sayıdır (kıyaslamalı olarak). Kısıt (5.19), f karar değişkeni ile h karar değişkeni arasında bağlantıyı sağlayan kısıttır.

5.2 Heterojen Filolu Açık Uçlu Araç Rotalama Problemi için Geliştirilen Matematiksel Modeller

5.2.1 HFAUARP için geliştirilen tek amaçlı matematiksel model

Depo noktası 1 olarak alınmıştır.

Kümeler

i, j : şehirlerin kümesi $i, j = 1, \dots, n$

t : farklı türdeki araçların kümesi $t = 1, \dots, T$

v : farklı türlere ait araç sayısı $v = 1, \dots, A_t$

Parametreler

Q_t : t . tür aracın kapasitesi

c_{ij} : i şehirden j şehrine gidilen mesafe

A_t : t . türe ait araç sayısı

d_i : i . şehrin talebi

M_{tv} : t . türün v . aracının kullanılmasının maliyeti

p_t : t . türün kilometre başına oluşan rota maliyeti

u_i ve u_j altturda kullanılan pozitif değişken olarak tanımlıdır.

Karar Değişkenleri

$$x_{ijtv} = \begin{cases} 1, & t. \text{ türün } v. \text{ aracı } i \text{ şehirden } j \text{ şehrine giderse} \\ 0, & \text{diğer durumda} \end{cases}$$

$$f_{tv} : \begin{cases} 1, & t. \text{ türün } v. \text{ aracı kullanılıyorsa} \\ 0, & \text{diğer durumda} \end{cases}$$

$$y_{itv} : \begin{cases} 1, & t. \text{ türün } v. \text{ aracı } i. \text{ şehre uğrarsa} \\ 0, & \text{diğer durumda} \end{cases}$$

Amaç fonksiyonu:

$$\text{En küçük } z = \sum_{i=1}^n \sum_{j=1}^n \sum_{t=1}^T \sum_{v=1}^{A_t} c_{ij} p_t x_{ijtv} + \sum_{t=1}^T \sum_{v=1}^{A_t} M_{tv} f_{tv} \quad (5.21)$$

Kısıtlar

$$\sum_{i=1}^n \sum_{v=1}^{A_t} d_i y_{itv} \leq Q_t \quad \forall t \quad (5.22)$$

$$\sum_{j=1}^n \sum_{v=1}^{A_t} x_{1jtv} \leq A_t \quad \forall t \quad (5.23)$$

$$\sum_{t=1}^T \sum_{v=1}^{A_t} y_{itv} = 1 \quad i = 1, \dots, n \quad (5.24)$$

$$\sum_{i \neq j}^n x_{ijtv} = y_{jtv} \quad j = 1, \dots, n, \forall t, \forall v \quad (5.25)$$

$$\sum_{j=1}^n x_{ijtv} \leq y_{itv} \quad i = 1, \dots, n, \forall t, \forall v, i \neq j \quad (5.26)$$

$$u_i - u_j + n \sum_t \sum_v x_{ijtv} \leq n - 1 \quad \forall i, j = 2, \dots, n, i \neq j \quad (5.27)$$

$$\sum_i \sum_{j \neq i}^n x_{ijtv} \leq M * f_{tv} \quad \forall t, \forall v \quad (5.28)$$

$$x_{ijtv}, f_{tv}, y_{itv} \in \{0,1\} \quad i, j = 1, \dots, n ; t = 1, \dots, T; v = 1, \dots, A_t \quad (5.29)$$

Amaç fonksiyonu olan (5.21), toplam rota maliyetinin ve araçların depodan çıkma maliyetlerinin toplamının en küçüklenmesini ifade etmektedir. Kısıt (5.22) kapasite kısıtını ifade etmektedir. Kısıt (5.23), her t tür araç için, depodan o türe ait araç sayısından fazla araç çıkamayacağını belirtmektedir. Kısıt (5.24), her müşteri noktasına sadece bir kez mutlaka bir araç tarafından uğranılması gerektiğini belirtmektedir. Kısıt (5.25)'e göre, eğer j . şehre t . türün v . aracı uğradıysa y_{jtv} karar değişkeni 1 olur ve bu j şehrine mutlaka başka bir i şehirden gelinmiştir. Kısıt

(5.26)'ya göre, eğer i . şehre t . türün v . aracı uğradıysa $y_{itv} = 1$ olur. Bu i şehirden herhangi başka bir j şehrine gidilebilir ya da gidilmez. Bu kısıt modelin açık uçlu olmasını sağlayan kısıttır. Kısıt (5.27) alttır engelleme kısıtıdır. Kısıt (5.28), x karar değişkeni ile f karar değişkeni arasındaki bağlantıyı sağlar. M büyük bir sayıdır.

5.2.2 HFAUARP için geliştirilen çok amaçlı matematiksel model

Depo noktası 1 olarak alınmıştır.

Kümeler

i, j : şehirlerin kümesi $i, j = 1, \dots, n$

t : farklı türdeki araçların kümesi $t = 1, \dots, T$

v : farklı türlere ait araç sayısı $v = 1, \dots, A_t$

Parametreler

Q_t : t . tür aracın kapasitesi

c_{ij} : i şehirden j şehrine gidilen mesafe

A_t : t . türe ait araç sayısı

d_i : i . şehrin talebi

M_{tv} : t . türün v . aracının kullanılmasının maliyeti

p_t : t . türün kilometre başına oluşan rota maliyeti

u_i ve u_j altturda kullanılan pozitif değişken olarak tanımlıdır.

Karar Değişkenleri

$$x_{ijtv} : \begin{cases} 1, & t. \text{ türün } v. \text{ aracı } i \text{ şehirden } j \text{ şehrine giderse} \\ 0, & \text{diğer durumda} \end{cases}$$

$$f_{tv} : \begin{cases} 1, & t. \text{ türün } v. \text{ aracı kullanılıyorsa} \\ 0, & \text{diğer durumda} \end{cases}$$

$$y_{itv} : \begin{cases} 1, & t. \text{ türün } v. \text{ aracı } i. \text{ şehre uğrarsa} \\ 0, & \text{diğer durumda} \end{cases}$$

$$h_t : \begin{cases} 1, & t. \text{ tür araç kullanılıyorsa} \\ 0, & \text{diğer durumda} \end{cases}$$

Amaç fonksiyonları:

$$\text{En küçük } z_1 = \sum_i^n \sum_j^n \sum_t^T \sum_v^{A_t} c_{ij} p_t x_{ijtv} + \sum_t^T \sum_v^{A_t} M_{tv} f_{tv} \quad (5.30)$$

$$\text{En küçük } z_2 = \sum_t^T h_t \quad (5.31)$$

Kısıtlar

$$\sum_i^n \sum_v^{A_t} d_i y_{itv} \leq Q_t \quad \forall t \quad (5.32)$$

$$\sum_{j=1}^n \sum_v^{A_t} x_{1jtv} \leq A_t \quad \forall t \quad (5.33)$$

$$\sum_t^T \sum_v^{A_t} y_{itv} = 1 \quad \forall i, i = 1, \dots, n \quad (5.34)$$

$$\sum_{i \neq j}^n x_{ijtv} = y_{jtv} \quad \forall j, j = 1, \dots, n, \forall t, \forall v, j \neq i \quad (5.35)$$

$$\sum_{j=1}^n x_{ijtv} \leq y_{itv} \quad \forall i, i = 1, \dots, n, \forall t, \forall v, i \neq j \quad (5.36)$$

$$u_i - u_j + n \sum_t^T \sum_v^{A_t} x_{ijtv} \leq n - 1 \quad \forall i, j = 2, \dots, n, i \neq j \quad (5.37)$$

$$\sum_i^n \sum_{j \neq i}^n x_{ijtv} \leq M * f_{tv} \quad \forall t, \forall v \quad (5.38)$$

$$\sum_v^{A_t} f_{tv} \leq M * h_t \quad \forall t \quad (5.39)$$

$$x_{ijtv}, f_{tv}, y_{itv}, h_t \in \{0,1\} \quad i, j = 1, \dots, n ; t = 1, \dots, T; v = 1, \dots, A_t \quad (5.40)$$

Birinci amaç olan (5.30), toplam rota maliyetinin ve araçların depodan çıkma maliyetlerinin toplamının en küçüklenmesini ifade etmektedir. İkinci amaç olan (5.31) ise, kullanılan araç türlerini en küçüklemeyi amaçlamaktadır. Kısıt (5.32) kapasite kısıtını ifade etmektedir. Kısıt (5.33), her t tür araç için, depodan o türe ait araç sayısından fazla araç çıkamayacağını belirtmektedir. Kısıt (5.34), her müşteri noktasına sadece bir kez mutlaka bir araç tarafından uğranılması gerektiğini

belirtmektedir. Kısıt (5.35)'e göre, eğer j . şehre t . türün v . aracı uğradıysa y_{jtv} karar değişkeni 1 olur ve bu j şehrine mutlaka başka bir i şehirden gelinmiştir. Kısıt (5.36)'ya göre, eğer i . şehre t . türün v . aracı uğradıysa $y_{itv} = 1$ olur. Bu i şehirden herhangi başka bir j şehrine gidilebilir ya da gidilmeyebilir. Bu kısıt modelin açık uçlu olmasını sağlayan kısıttır. Kısıt (5.37) alttır engelleme kısıttır. Kısıt (5.38), x ile f karar değişkeni arasındaki bağlantıyı sağlar. M büyük bir sayıdır. Kısıt (5.39) ise f ile h arasındaki bağlantıyı sağlayan kısıttır.

5.3 Bölünebilir Talepli Araç Rotalama Problemi için Geliştirilen Matematiksel Modeller

5.3.1 BTARP için geliştirilen tek amaçlı matematiksel model

Depo noktası 1 olarak alınmıştır.

Kümeler

i, j : şehirlerin kümesi $i, j = 1, \dots, n$

t : farklı türdeki araçların kümesi $t = 1, \dots, T$

v : farklı türlere ait araç sayısı $v = 1, \dots, A_t$

Parametreler

Q_t : t . tür aracın kapasitesi

c_{ij} : i şehirden j şehrine gidilen mesafe

A_t : t . türe ait araç sayısı

d_i : i . şehrin talebi

M_{tv} : t . türün v . aracının kullanılmasının maliyeti

p_t : t . türün kilometre başına oluşan rota maliyeti

u_i ve u_j alttırdaki kullanılan pozitif değişken olarak tanımlıdır.

Karar Değişkenleri

x_{ijtv} : $\begin{cases} 1, & t. \text{ türün } v. \text{ aracı } i \text{ şehirden } j \text{ şehrine giderse} \\ 0, & \text{diğer durumda} \end{cases}$

f_{tv} : $\begin{cases} 1, & t. \text{ türün } v. \text{ aracı kullanılıyorsa} \\ 0, & \text{diğer durumda} \end{cases}$

y_{itv} : t . türün v . aracının, i şehrinin talebini karşıladığı oran

Amaç fonksiyonu:

$$\text{En küçük } z = \sum_i^n \sum_j^n \sum_t^T \sum_v^{A_t} c_{ij} p_t x_{ijtv} + \sum_t^T \sum_v^{A_t} M_{tv} f_{tv} \quad (5.41)$$

Kısıtlar:

$$\sum_i^n x_{iktv} - \sum_j^n x_{kjtv} = 0 \quad k = 1, \dots, n, \forall t, \forall v \quad (5.42)$$

$$\sum_{j=1}^n \sum_v^{A_t} x_{1jtv} \leq A_t \quad \forall t \quad (5.43)$$

$$\sum_{i=1}^n d_i y_{itv} \leq Q_t \quad \forall t, \forall v \quad (5.44)$$

$$y_{itv} \leq \sum_j^n x_{ijtv} \quad i = 1, \dots, n, \forall t, \forall v \quad (5.45)$$

$$\sum_{t=1}^T \sum_{v=1}^{A_t} y_{itv} = 1 \quad i = 1, \dots, n \quad (5.46)$$

$$u_i - u_j + n \sum_t^T \sum_v^{A_t} x_{ijtv} \leq n - 1 \quad i, j = 2, \dots, n, i \neq j \quad (5.47)$$

$$\sum_j^n \sum_t^T \sum_v^{A_t} x_{ijtv} \geq 1 \quad i = 1, \dots, n, i \neq j \quad (5.48)$$

$$\sum_i^n \sum_j^n x_{ijtv} \leq M * f_{tv} \quad \forall t, \forall v \quad (5.49)$$

$$0 \leq y_{itv} \leq 1 \quad \forall i, \forall t, \forall v \quad (5.50)$$

$$x_{ijtv}, f_{tv}, y_{itv}, h_t \in \{0,1\} \quad i, j = 1, \dots, n ; t = 1, \dots, T; v = 1, \dots, A_t \quad (5.51)$$

Amaç fonksiyonu olan (5.41), toplam rota maliyetinin ve araçların depodan çıkma maliyetlerinin toplamının en küçüklenmesini ifade etmektedir. Kısıt (5.42), araç akış kısıtıdır. Kısıt (5.43), her t tür araç için, depodan o türe ait araç sayısından fazla araç çıkamayacağını belirtmektedir. Kısıt (5.44)'de y_{itv} t . türün v . aracının i şehrinin talebinin ne kadarlık kısmını karşıladığını belirten bir oranı ifade eder. Her tür ve her araç için bakıldığında taleplerin oran ile çarpımı araç kapasitesini aşamaz. Kısıt (5.45)'de x_{ijtv} toplamaları 1 olursa y_{itv} oranı mutlaka 1 den küçük ya da eşit

olmalıdır. Oranların toplamı mutlaka 1 olmalıdır. Kısıt (5.46)'de ise, her i şehri için (depo hariç), araçlar tarafından karşılanan şehirlerin taleplerinin oranlarının toplamı mutlaka 1 olmak zorundadır. Kısıt (5.47), alttur engelleme kısıtıdır. Kısıt (5.48)'e göre her şehre mutlaka bir kez uğranılmalıdır. Bir şehrin talebi bölünüyorsa, o şehre farklı araçlar uğrayabilir. Kısıt (5.49), x ile f karar değişkeni arasındaki bağlantıyı sağlar. M büyük bir sayıdır. Kısıt (5.50), y_{itv} oranının 0 ile 1 arasında olduğunu belirtmektedir.

5.3.2 BTARP için geliştirilen çok amaçlı matematiksel model

Depo noktası 1 olarak alınmıştır.

Kümeler

i, j : şehirlerin kümesi $i, j = 1, \dots, n$

t : farklı türdeki araçların kümesi $t = 1, \dots, T$

v : farklı türlere ait araç sayısı $v = 1, \dots, A_t$

Parametreler

Q_t : t . tür aracın kapasitesi

c_{ij} : i şehirden j şehrine gitme maliyeti (mesafe)

A_t : t . türe ait araç sayısı

d_i : i . şehrin talebi

M_{tv} : t . türün v . aracının kullanılmasının maliyeti

p_t : t . türün kilometre başına oluşan rota maliyeti

u_i ve u_j altturda kullanılan pozitif değişken olarak tanımlıdır.

Karar Değişkenleri

$$x_{ijtv} : \begin{cases} 1, & t. \text{ türün } v. \text{ aracı } i \text{ şehirden } j \text{ şehrine giderse} \\ 0, & \text{diğer durumda} \end{cases}$$

$$f_{tv} : \begin{cases} 1, & t. \text{ türün } v. \text{ aracı kullanılıyorsa} \\ 0, & \text{diğer durumda} \end{cases}$$

$$h_t : \begin{cases} 1, & t. \text{ tür araç kullanılıyorsa} \\ 0, & \text{diğer durumda} \end{cases}$$

y_{itv} : t . türün v . aracının, i şehrinin talebini karşıladığı oranı ifade etmektedir.

Amaç fonksiyonları:

$$\text{En küçük } z_1 = \sum_i^n \sum_j^n \sum_t^T \sum_v^{A_t} c_{ij} p_t x_{ijtv} + \sum_t^T \sum_v^{A_t} M_{tv} f_{tv} \quad (5.52)$$

$$\text{En küçük } z_2 = \sum_t^T h_t \quad (5.53)$$

$$\sum_i^n x_{iktv} - \sum_j^n x_{kjtv} = 0 \quad k = 1, \dots, n, \forall t, \forall v \quad (5.54)$$

$$\sum_{j=1}^n \sum_v^{A_t} x_{1jtv} \leq A_t \quad \forall t \quad (5.55)$$

$$\sum_{i=1}^n d_i y_{itv} \leq Q_t \quad \forall t, \forall v \quad (5.56)$$

$$y_{itv} \leq \sum_j^n x_{ijtv} \quad i = 1, \dots, n, \forall t, \forall v \quad (5.57)$$

$$\sum_{t=1}^T \sum_{v=1}^{A_t} y_{itv} = 1 \quad i = 1, \dots, n \quad (5.58)$$

$$u_i - u_j + (n) \sum_t^T \sum_v^{A_t} x_{ijtv} \leq n - 1 \quad i, j = 1, \dots, n, i \neq j \quad (5.59)$$

$$\sum_j^n \sum_t^T \sum_v^{A_t} x_{ijtv} \geq 1 \quad i = 1, \dots, n, i \neq j \quad (5.60)$$

$$\sum_i^n \sum_j^n x_{ijtv} \leq M * f_{tv} \quad \forall t, \forall v \quad (5.61)$$

$$0 \leq y_{itv} \leq 1 \quad \forall i, \forall t, \forall v \quad (5.62)$$

$$\sum_v^{A_t} f_{tv} \leq M * h_t \quad \forall t \quad (5.63)$$

$$x_{ijtv}, f_{tv}, y_{itv}, h_t \in \{0,1\} \quad i, j = 1, \dots, N ; t = 1, \dots, T; v = 1, \dots, A_t \quad (5.64)$$

Amaç fonksiyonu olan (5.52), toplam rota maliyetinin ve araçların depodan çıkma maliyetlerinin toplamının en küçüklenmesini ifade etmektedir. İkinci amaçta (5.53) ise kullanılan araç türü en küçüklenmesi amaçlanmıştır. Kısıt (5.54), araç akış

kısıttır. Kısıt (5.55), her t tür araç için, depodan o türe ait araç sayısından fazla araç çıkamayacağını belirtmektedir. Kısıt (5.56)'de y_{itv} t . türün v . aracının i şehrinin talebinin ne kadarlık kısmını karşıladığını belirten bir oranı ifade eder. Her tür ve her araç için bakıldığında taleplerin oran ile çarpımı araç kapasitesini aşamaz. Kısıt (5.57)'de x_{ijtv} toplamları 1 olursa y_{itv} oranı mutlaka 1 den küçük ya da eşit olmalıdır. Oranların toplamı mutlaka 1 olmalıdır. Kısıt (5.58)'de ise, her i şehri için (depo hariç), araçlar tarafından karşılanan şehirlerin taleplerinin oranlarının toplamı mutlaka 1 olmak zorundadır. Kısıt (5.59), alttur engelleme kısıttır. Kısıt (5.60)'a göre her şehre mutlaka bir kez uğranılmalıdır. Bir şehrin talebi bölünüyorsa, o şehre farklı araçlar uğrayabilir. Kısıt (5.61), x ile f karar değişkeni arasındaki bağlantıyı sağlar. M büyük bir sayıdır. Kısıt (5.62), y_{itv} oranının 0 ile 1 arasında olduğunu belirtmektedir. Kısıt (5.63), f karar değişkeni ile h karar değişkeni arasındaki bağlantıyı büyük M sayısı ile (kıyaslamalı olarak) sağlamaktadır.

6 ARAÇ ROTALAMA PROBLEMİ İÇİN LİTERATÜRDE ÇALIŞILAN KESİN ÇÖZÜM YÖNTEMLERİ

6.1 Dal Sınır Yöntemi

Tamsayılı programlama problemlerinin çözümünde kullanılmaktadır. Böl ve yönet ilkesine dayalı bir sayım yöntemidir. Büyük problemler daha küçük problemlere bölünmektedir. Bölme aşamasında, tüm uygun çözümler daha küçük alt kümelere ayrılır. Dallandırma adım sayısını azaltmak için sınırlandırma yöntemi kullanılır. Dallandırma ile oluşturulan problemlerin çözümlerinin alt ve üst sınır değerleri belirlenir. Tüm alt problemler sınırlandırılırsa algoritma sona erer. Eğer altkümenin sınırı, altkümenin eniyi çözümü asla kapsayamayacağını gösteriyorsa bu altküme çıkartılır. En iyi alt sınır, problemin çözümü olur. Bu yöntemde problem için eniyi çözüm aranırken, problemin bütün aşamaları sistematik olarak gözden geçirilmelidir.

6.2 Kesme Düzlemi Yöntemi

Dal sınır algoritmasına alternatif olarak geliştirilen bir yöntemdir. Tamsayılı doğrusal programlama probleminde gerekiyorsa, öncelikle sınırların tamsayı olması sağlanır ve bu durumda sınırların değiştirilmesi söz konusudur. Problemin daha sonra eniyi çözüm tablosu bulunur. Eniyi çözüm tamsayı ise durulur. Problemin çözümü bulunmuştur.

6.3 Dinamik Programlama

Böl ve yönet yönteminde alt problemler birbirinden bağımsızdır. Birbirinden bağımsız olmadığı durumda dinamik programlama kullanılır. Dinamik programlama her bir alt problemi bir kez çözerek bu çözümleri bir tabloda tutar. Aynı alt problem ile tekrar karşılaşıldığında problemi yeniden çözmeyerek, tablodan bu problemin değerini okur.

6.4 Araç Rotalama Problemi İçin Çalışılan Sezgisel Çözüm Yöntemleri

ARP için birçok sezgisel yaklaşım önerilmiştir. Klasik ARP sezgisel yöntemleri üç ana kategoride incelenebilir [33].

- **Yapıcı Sezgiseller:** Çözüm maliyetini dikkate alarak uygun bir çözüm oluştururlar. Bu sezgisellerde iyileştirme aşaması yoktur.
- **İki Aşamalı Sezgiseller**
- **İyileştirici sezgiseller:** Burada amaç herhangi bir uygun çözümü, araç rotaları arasında ya da rotanın kendi içerisinde müşteri noktalarının değişimi ya da ark değişimleri yaparak iyileştirmeye çalışmaktır.

6.4.1 Yapıcı sezgiseller

ARP çözümlerini oluşturmak için farklı teknikler bulunmaktadır. Bu yöntemlerden olan kazanım yöntemi ve en yakın komşuluk yöntemi ile rotalar birleştirilir. Daha sonra ekleme maliyetleri dikkate alınarak müşteriler araç rotalarına adım adım eklenir [33].

6.4.1.1 Clarke ve Wright kazanım algoritması

Kazanım algoritması [39] ARP' de yaygın olarak bilinen bir yöntemdir. Yöntem, her iki müşteri arasındaki kazanımı (maliyet) hesaplar. Maliyet kazanımlarının hesaplanmasından sonra iki müşteri noktası arasına başka bir müşteri noktası eklenir. Bu algoritma araç sayısının karar değişkeni olduğu problemlere uygulanır.

Adım 1: Kazanım hesaplanması

c_{i0} : i şehrinde depoya olan mesafe

c_{0j} : depodan j şehrine olan mesafe

s_{ij} : i ve j nin farklı turlarda değil aynı turda olmasından kazanılan kazanım değeri olmak üzere,

$$s_{ij} = c_{i0} + c_{0j} - c_{ij} \quad i, j = 1, \dots, n \quad i \neq j$$

formülasyonu ile kazanım hesaplanır.

i ve j şehirleri ayrı rotalarda iken kazanım algoritmasının uygulanması ile i ve j aynı rotaya dahil edilir.

Kazanım algoritmasının paralel ve seri olmak üzere iki versiyonu vardır. Seri versiyonunda her seferinde sadece bir rota oluşturulur iken paralel versiyonda her seferinde birden fazla rota oluşturulabilir. Kazanım algoritmasında öncelikle tüm müşteri çiftlerinin kazanımı hesaplanır. Daha sonra bu kazanımlar azalan sıraya göre sıralanırlar. Listenin en başında bulunan müşteri çiftinden başlanılarak her seferinde bir nokta çifti ele alınır. Kazanım listesinin en başından başlayarak $(0,j)$ ve $(i,0)$ kenarlarını içeren bir rota olup olmadığı kontrol edilerek uygun bir şekilde birleştirilmesi durumuna bakılır [33]. Eğer uygunsuzsa, $(0,j)$ ve $(i,0)$ bağlantıları silinerek yerine (i,j) bağlantısı eklenir ve böylece bir rota birleştirilir. Rotaların birleşiminde araç kapasitesi göz önüne alınmalıdır ve kapasite ihlali gerçekleştirilemez.

6.4.1.2 En yakın komşuluk yöntemi

Bir müşteri noktasından başlanılarak, o müşteri noktasına en yakın komşu olan şehir, araç kapasite koşullarına uygunluk durumlarına bakılarak eklenir. En çok çalışılan yöntemlerden bir tanesidir.

6.4.1.3 Mole ve Jameson sıralı ekleme sezgiseli

Bu sezgiselde α ve μ olmak üzere rota oluşturmada kullanılan bir tane parametre bulunmaktadır. Algoritmanın adımları şu şekildedir:

Adım 1) (Başlangıç rota tanımlama): Eğer tüm noktalar herhangi bir rotada yer alıyorsa durulur. Yer almıyorsa, i rotalanmayan bir nokta olmak üzere $(0, i, 0)$ rotası oluşturulur.

Adım 2) (Bir sonraki nokta seçimi): 1. Adımda oluşturulan bu rotaya eklenebilecek rotalanmamış tüm noktaların ekleme maliyetleri hesaplanır. Eğer hiçbir ekleme uygun çözüm oluşturmazsa 1. Adıma geri dönülür.

Adım 3) (Rota eniyileme): 3-opt uygulanarak güncel rota optimize edilir ve 1. Adıma dönülür [33]. 3-opt hareketinde turda yer alan 3 tane kenar silinir ve

serimdeki kalan olası yollar yeniden birleştirilir. Yeniden birleştirilme durumu her alternatif için hesaplanır ve en iyi olan seçilir.

6.4.1.4 Christofides, Mingozzi ve Toth sıralı ekleme sezgiseli

Christofides, Mingozzi ve Toth sıralı ekleme sezgiseli [40], Mole ve Jameson yöntemine göre daha karmaşık bir yöntemdir. Diğer yöntemlere göre daha iyi çözümler bulunmuştur. Algoritmanın adımları izleyen şekilde verilmiştir:

Adım 1: İlk rota indeksi $k=1$ olacak şekilde adlandırılır.

Adım 2: Rotalanmamış herhangi bir nokta seçilerek k başlangıç rotasına tanımlanır. Her rotalanmamış i noktası için bir maliyet hesaplanır.

Adım 3: k rotasına uygun bir şekilde eklenebilecek rotalanmamış müşteriler arasından en düşük maliyetli olanı seçilir. Müşteri noktası bu k rotasına eklenir. Daha sonra 3-opt algoritması ile k rotası optimize edilir. K rotasına daha fazla müşteri noktası eklenemeyinceye kadar bu adım tekrar edilir.

Adım 4: Eğer tüm noktalar bir rotaya atandıysa durulur, atanmadıysa k birer artırılarak 1. Adıma geçilir [33].

6.4.2 İki aşamalı yöntemler

İki aşamalı sezgiseller iki ayrı sınıfa ayrılırlar. Birinci sınıfta önce kümele sonra rotala metotları, ikinci sınıfta ise önce rotala sonra kümele metotları yer almaktadır. Önce kümele sonra rotala metotlarında öncelikle müşteri noktaları ile uygun kümeler oluşturulur daha sonra oluşturulan bu kümeler için araç rotalama uygulanır. Bu metotlara örnek olarak süpürme algoritması ve Fisher ve Jaikumar algoritmasının [37] da kullandığı genelleştirilmiş atama metodu verilebilir. Önce rotala sonra kümele metotlarında ise, öncelikle bütün müşteri noktaları rotalanır ve daha sonra uygun araç rotalarına bölünür.

6.4.2.1 Süpürme algoritması

Önce kümele sonra rotala metotlarından bir tanesidir. Gillett ve Miller tarafından 1974 yılında önerilmiştir [41]. ARP'nin düzlemsel örneklerine uygulanır. Depo başlangıç noktası olarak kabul edilir ve depodan başlayarak müşteri noktalarına rassal olarak ulaşır. Başlangıçta bir i müşteri noktasına gidilerek araç kapasitesi dolana kadar müşteri noktaları bu rotaya eklenir. Kapasite dolunca araç depoya geri döner. Tüm araçlar kullanılıncaya kadar algoritma devam eder. Oluşturulan bu araç rotaları daha sonra GSP ile optimize edilir. Daha sonra her bir küme için gezgin satıcı problemi çözülerek araç rotaları oluşturulur. Süpürme algoritmasının adımları şu şekildedir [33]:

Adım 1) Rota başlatma adımıdır. Kullanılmamış k aracı seçilir.

Adım 2) Rota oluşturma adımıdır. En küçük açığa sahip olan rotalanmayan bir müşteri noktasından başlanılarak, aracın kapasitesi dolana kadar ya da maksimum rota uzunluğu aşılmamasına kadar müşteri noktaları eklenir. Her eklemenden sonra 3-opt prosedürü yapılabilir. Rotalanmamış müşteri noktaları varsa 1. adıma dönlür.

Adım 3) Rota eniyilemesi yapılır. GSP ile oluşturulan her rota iyileştirilir.

6.4.2.2 Fisher ve Jaikumar algoritması

Önce kümele sonra rotala metotlarındandır. Küme oluşturmak için genelleştirilmiş atama yöntemi kullanılır [42].

Adım 1) Merkez seçme adımıdır. Her k kümesini tanımlamak için bir çekirdek (merkez) müşteri noktası seçilir.

Adım 2) Müşterilerin çekirdeklere yerleştirilme adımıdır. Her k kümesine her bir i müşterisini ekleme maliyeti hesaplanır.

Adım 3) Genelleştirilmiş atama adımıdır. Müşteri ağırlıkları, araç kapasitesi ve maliyetler ile GAP çözülür.

Adım 4) GSP adımıdır. Genelleştirilmiş atama problemi çözümüne bağlı olarak ilgili her küme için GSP çözülür.

6.4.2.3 Bramel ve Simchi-Levi algoritması

Bramel ve Simchi-Levi [43] tarafından geliştirilmiştir. Bu algoritmada çekirdekler, kapasiteli lokasyon probleminin çözülmesi ile hesaplanır ve kalan noktalar ikinci adımda yavaşça rotalarına dahil edilir. Öncelikle K tane merkez yerleştirilmeli, daha sonra kapasite kısıtını aşmadan n müşteri noktasından mesafeyi en küçükleyecek şekilde en yakın merkezlerine yerleştirilir. Minimum ekleme maliyeti olan rota çekirdeklerine her bir adımda müşteriler atanır ve böylece araç rotaları oluşturulmuş olur.

6.4.2.4 Kesilmiş Dal-Sınır Yöntemi

Bu algoritma 1979 yılında Christofides ve ark. tarafından önerilmiştir [40]. Arama ağacının araç rotaları bazında birden fazla seviyesi bulunmaktadır ve her bir seviye uygun araç rotaları içermektedir. Ağacın her bir seviyesinde sadece 1 tane dallanma vardır. Kısıtlandırılmış ağaç, birkaç umut verici rotanın her seviyede tutulması ile oluşturulabilir.

6.4.2.5 Taçyaprağı algoritmaları

Süpürme algoritmasının doğal bir uzantısıdır. Taç yaprakları şeklinde birden fazla rota oluşturulur ve küme ayrışımı problemi çözülerek sonuç rota seçimi yapılır [33].

6.4.2.6 Önce rotala sonra kümele metotları

İlk aşamada, yan kısıtlar dikkate alınmadan büyük bir GSP turu oluşturulur. Daha sonra bu tur ikinci aşamada uygun araç rotalarına ayrıştırılır. Beasley [44], Haimovich ve Rinnooy Kan [45], Bertsimas ve Simchi-Levi [46] bu metotlar ile ilgili çalışmalar yapmıştır.

6.4.3 İyileştirici sezgiseller

Bu sezgiseller bir kerede her rotayı ayrı ayrı ele alarak ya da birden fazla rotayı ele alarak kullanır. Tek rota iyileştirici sezgiseller ve çoklu rota iyileştirici sezgiseller çalışılmıştır. Yerel arama algoritmaları genellikle diğer sezgiseller tarafından oluşturulan başlangıç çözümlerini iyileştirmek amacıyla kullanılır. Verilen bir çözümden başlanılarak, yerel arama yapılır. Bu yöntemde daha iyi maliyetli çözümler elde edebilmek için müşterilerin yerleri değiştirilebilir ya da farklı değişim hareketleri yapılır. Eğer iyileştirilmiş bir çözüm bulunursa, elde edilen bu çözüm güncel çözüm olur ve süreç bu şekilde devam eder. Eğer değilse, yerel minimum çözüm bulunmuş olur. Komşuluk yapısında oldukça çeşitlilik bulunmaktadır. Bu yüzden bu komşuluklar iç rota komşuluklarına ayrıştırılabilir. En sık kullanılan komşuluk yapısı, ilk olarak Lin [47] tarafından çalışılan λ -opt sezgiselidir. Bu sezgiselde, eldeki çözümdeki λ adet kenar kaldırılır ve kalan λ adet kenar ile yer değiştirilir. Çözümdeki tüm komşulukları bulmak için gereken hesaplama zamanı n^λ kadardır. Uygulamalarda genellikle 1-opt ve 3-opt kullanılmaktadır [48].

7 ARAÇ ROTALAMA PROBLEMİ İÇİN LİTERATÜRDE ÇALIŞILAN METASEZGİSEL YÖNTEMLER

Çeşitli metasezgisel yöntemler ARP'nin çözümü için uygulanmıştır. Klasik sezgisellere göre, çözüm uzayında daha geniş bir bölgede arama yaparak daha az oranda yerel eniyiye takılırlar. Klasik sezgiseller genelde kısa sürede bir çözüm bulmaya yönelik yöntemlerdir.

Metasezgisel yöntemler genel olarak üç sınıfta incelenebilir [48]:

- Yerel aramanın kullanıldığı; tavlama benzetimi ve yasaklı arama.
- Popülasyon türü aramanın kullanıldığı; genetik algoritma ve uyarlanabilir hafıza prosedürü.
- Öğrenme mekanizmalarının kullanıldığı, sinir ağları ve karınca kolonisi algoritması.

Yerel arama temelli algoritmalar, bir x_t çözümünün $N(x_t)$ komşuluğunda, durdurma kriteri sağlanana kadar, iteratif olacak şekilde, x_t çözümünden başlayarak, her t iterasyonda x_{t+1} çözümüne geçerek tüm çözüm uzayını ararlar [48]. Yerel arama algoritmalarında komşuluk yapısını belirleyecek olan kural oldukça önemlidir. Popülasyon temelli arama yapan algoritmalarda, çözüm popülasyonunun farklı nesillerinde çalışılır. Öğrenme mekanizmaları temelli algoritmalarda ise, karıncaların doğal yaşamdaki davranışlarından esinlenildiği karınca kolonisi algoritmasında olduğu gibi ya da insan beyninin bilgi işleme tekniğinden esinlenen sinir ağları algoritmaları geri bildirim mekanizması ile çözümleri düzenler.

7.1 Tavlama Benzetimi (TB)

TB, katıların fiziksel tavlama süreci ile olan benzerliklerinden ortaya çıkmıştır. Kirkpatrick ve ark. tarafından 1983 yılında ilk defa çalışılmıştır [49]. Stokastik bir arama sürecidir. Katı madde, sıvı durumundaki bütün parçacıkları rasgele hareket edinceye kadar ısıtılır. Daha sonra parçacıklar donmuş bir hale gelen kadar sıcaklık yavaş yavaş azaltılır. Bu sırada sistem enerjisi en düşük seviyededir.

Algoritmanın adımları şu şekildedir:

Adım 1) Başlangıç çözümünün oluşturulması ve bunun mevcut çözüm olarak tutulması.

Adım 2) Başlangıç sıcaklığının tanımlanması ile mevcut sistem sıcaklığı başlangıç sıcaklığına eşitlenir.

Adım 3) Rasgele seçilmiş bir hareket ile eldeki çözümden komşu çözüm üretilir.

Adım 4) Oluşturulan komşu çözüm ile mevcut çözümün arasındaki fark (Δ) hesaplanır. Eğer $\Delta > 0$ ise 3.adıma dönlür. Değilse, yeni çözüm bu komşu çözüm olarak kabul edilir.

Adım 5) Sıcaklık güncellenir ve durma kriterinde algoritma sona erer.

TB' de aramaya yeteri kadar yüksek bir sıcaklık değeri ile başlamak çok önemlidir. Amaç fonksiyonunda, Δ kadarlık bir yükselmeye yol açan hareketin kabul edilme olasılığını veren fonksiyona kabul fonksiyonu denir. Sıcaklık arttıkça amaç fonksiyonunda artışa neden olacak olan hareketlerin olasılığı da artmaktadır. Sıcaklık azaldıkça bu olasılık da azalmaktadır.

7.2 Genetik Algoritma (GA)

Popülasyon tabanlı bir metasezgisel algoritmadır. 1975 yılında Holland [50] tarafından ilk defa çalışılmıştır. Tek bir noktadan değil, noktalar kümesinden algoritma başlamaktadır. Doğal seçilime dayalı arama prosedürüdür. İteratif olarak, yeni bir popülasyon oluşturulur ve güncel popülasyon ile yeni popülasyon yer değiştirir. GA, evrimsel programlamanın en çok kullanılan alanıdır. Gen, birey, popülasyon, fitness (uygunluk), çaprazlama ve mutasyon, GA içerisindeki önemli yapılarıdır. Genler, olası çözümlerin temelini oluştururken, bireyler genlerinde bulunan özellikleri taşırlar. Her bir birey olası bir çözümdür. Kromozom olarak da adlandırılabilirler. Popülasyon, olası çözümlerden oluşan bir topluluğu belirtir. Uygunluk değeri ise, bireyin çevre şartlarına karşı ne kadar dayanıklı olduğunu gösteren bir değerdir. Çözümün kalitesini belirler ve bir uygunluk fonksiyonu ile hesaplanır. Genetik algoritmalar paralel çalışırlar yani aynı anda arama uzayının birden fazla noktasında arama yapabilirler. Problemlerin çözümünü genetik kodlar

ile aramaktadır. GA'da kullanılan çaprazlama operatörü ve mutasyon operatörü önemli rol oynamaktadır. Çaprazlama operatörü, iki çözüm arasında uygulanırken, mutasyon operatörü çeşitliliği sağlamak için bireylerin değiştirilmesini sağlamaktadır [23]. Çaprazlamada, çözümlerin farklı bölümleri birleştirilerek daha iyi çözümlerin elde edilmesi amaçlanır. Çaprazlama yapılırken, düşük uyumluluk değerine sahip bireylerin ebeveyn olma ihtimali daha düşüktür. Bu yüzden bir süre sonra popülasyon dışında kalırlar. Bu duruma elitizm denir. Elitizm ile giderek daha iyi nesiller oluşturulmaya çalışılmaktadır. Mutasyonda ise amaç çeşitliliği sağlamaktır. Seçilen tek bir bireye gerçekleştirilen ufak değişikliklerdir. Böylece ebeveynlerde olmayan bazı özellikler çocuklara aktarılabilir. Bireylerin gösterimi, uygunluk fonksiyonu ve genetik operatörlerin doğruluğu algoritmada çok önemli bir yere sahiptir. GA'daki ebeveynler sistematik olarak yavrular ile değiştirilir. GA'da birey sayısı, bireylerin seçim yöntemi, çaprazlama ve mutasyon olasılığı gibi parametreler bulunmaktadır. Birey seçim yöntemleri; rulet tekeri seçim yöntemi, sıralama seçim yöntemi ve turnuva seçim yöntemidir.

Algoritmanın genel adımları şu şekildedir:

Adım 1) Başlangıç popülasyonu oluşturulur.

Adım 2) Popülasyondaki her bir bireyin uygunluk değeri hesaplanır.

Adım 3) Popülasyondaki bireyler, birey seçim yöntemine göre çaprazlama yapılmak için seçilir.

Adım 4) Seçilen bu bireyler, çaprazlama olasılığına göre çaprazlanır ve yeni bireyler oluşturulur.

Adım 5) Yeni bireylere mutasyon uygulanır (mutasyon olasılığına göre).

Adım 6) Oluşturulan bu yeni bireyler popülasyonun mevcut bireyleridir.

GA'lar arama uzayının çok büyük ve karmaşık olduğu problemlerde oldukça kullanışlı bir yöntemdir. Problemin bütünsel en iyi çözümünü bulmak için garanti vermezler ama kabul edilebilir iyi çözümleri makul bir süre içerisinde bulabilirler [51].

7.3 Uyarlanabilir Hafıza Prosedürü

1995 yılında Rochat ve Taillard [52] tarafından ileri sürülmüştür ve metasezgisel alanına oldukça katkısı olmuştur. Uyarlanabilir hafıza, iyi çözümlerin tutulduğu bir havuzdur. Bu havuzda bulunan en kötü çözümler, daha iyi çözümler ile yer değiştirilir. Bu prosedürde, yeni bir çözüm oluşturabilmek için, havuzdan farklı çözümler seçilir ve bu çözümler yeniden birleştirilir. ARP kapsamında bu yöntemde, araç rotaları bu çözümlerden çıkartılarak yeni çözüm için bir temel olarak kullanılır. Bu çıkartılma durumu, daha önceden seçilen rotalar ile belirlenen rotaların çakışmaması haline kadar devam eder. Eğer daha fazla çıkartılma yapılamıyorsa, arama prosedürü, seçilen rotalar ve bazı seçilmeyen müşteri noktalarından oluşan parça çözümlerden başlatılır.

7.4 Yapay Sinir Ağları (YSA)

İnsan beyninin çalışma sisteminin benzetimi sonucunda ortaya çıkan bir algoritmadır. İnsan beyninden esinlenerek, öğrenme süreci matematiksel olarak modellenmiştir. Deneyimden yararlanır. YSA'da öğrenme sırasında giriş ve çıkış bilgileri verilir ve kurallar konulur. Hesaplama, öğrenmeden sonra paralel olarak gerçekleşir. YSA'nın hatalara toleransı bulunmaktadır. Bu algoritmalar genellikle sınıflandırma, tahminleme ve modelleme uygulamalarında kullanılmaktadır.

7.5 Karınca Kolonisi Algoritması (KKA)

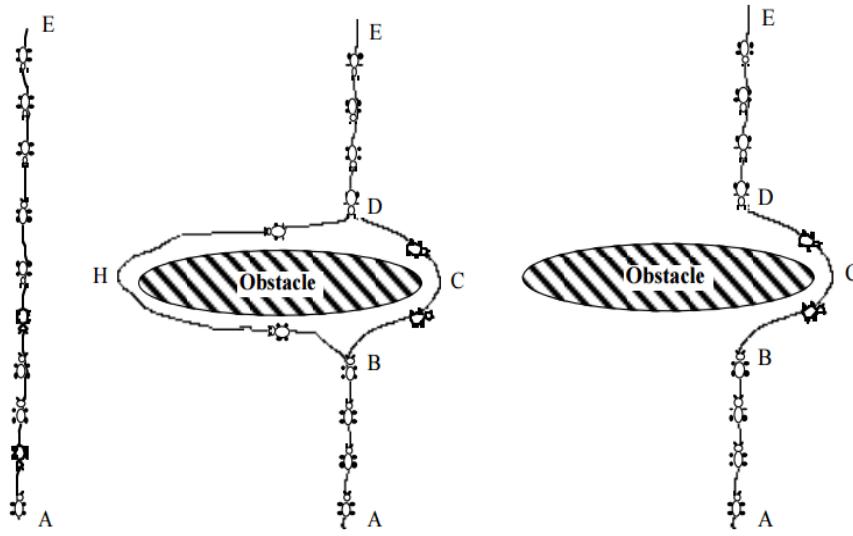
Dorigo ve ark. [53] tarafından önerilmiştir. Yaptıkları çalışmada, karınca kolonilerinin davranışlarını matematiksel modele dayandırarak bu algoritmayı ilk kez GSP üzerinde denemişlerdir. Olumlu sonuçlar almalarından sonra, algoritma farklı araştırmacılar tarafından yaygın şekilde kullanılmıştır. Bu algoritma, karıncaların gerçek davranışları üzerinden ilham almıştır. Gerçek karıncaların yiyecek arama hareketine odaklanılmıştır. Karıncalar yuvaları ile ulaşmak istedikleri besin arasındaki mesafeyi en küçükleme özelliğine sahiptirler. Yem araştırırken birbirleriyle haberleşmek için karıncalar kullandıkları yola feromon adı

verilen bir madde bırakırlar. Bu madde uçucu ve kokusu olan bir maddedir. Feromonun daha fazla olduğu yol karıncalar tarafından daha büyük bir olasılık ile seçilir. Feromonun bu yoğunluğu algoritmanın güçlendirme süreci olarak adlandırılır ve yiyecek miktarına bağlıdır. Feromon uçucu olduğundan zaman ilerledikçe buharlaşma süreci gerçekleşir. Feromon kuvvetlendirme oranı parametresi ile düğümler arasındaki feromon miktarlarının önem dereceleri belirlenir. Feromon buharlaşma oranı parametresi ile, her iterasyon sonucunda düğümlerin arasında bulunan feromonların ne kadarının buharlaşacağı belirlenir. Algoritmadaki sezgisellik kuvvetlendirme oranı ile de müşteri düğümleri arasındaki mesafenin önem derecesi belirlenir [54].

Algoritma şu şekildedir:

- Bütün karıncalar için yolların üretilmesi adıdır. Karıncalar şehirlere rasgele olacak şekilde yerleştirilir.
- Her karınca sezgisellik kuvvetlendirme oranı ve feromon kuvvetlendirme oranı parametreleri doğrultusunda şehirler arasında dolaşarak rotaları oluşturur.
- Yolların uzunluklarının hesaplanması adıdır. Her bir karıncanın rota mesafesi hesaplanır.
- Yollara bırakılan feromon miktarının güncellenmesi adıdır. Karıncaların rota mesafelerine göre feromon değerleri güncellenir.
- Bulunulan ana kadarki en kısa yolun tutulması adıdır.
- Bu adımlar durdurma kriterine kadar devam eder.

Karıncaların bir engelle karşılaştıklarında izledikleri yol Şekil 7.1’de gösterilmiştir.



Şekil 7.1. Karıncaların bir engelle karşılaştıkları durumda izledikleri yol [33]

7.6 Yasaklı Arama Algoritması (YA)

Yasaklı arama algoritması ilk defa Glover tarafından 1986 yılında incelenmiştir [55]. 1990 yıllarında yasaklı arama algoritması eniyileme problemlerinin çözümünde önem kazanmıştır. YA, yerel arama sezgiseli kılavuzluğunda, yerel eniyilemenin ötesinde bir arama uzayını keşfetmek amacı ile kullanılan tek çözüm temelli bir metasezgisel yöntemdir. Yerel eniyiden kaçmak amacıyla, tüm komşu çözümlerde iyileşme sağlanamadığında, iyileşmeyen çözümleri de kabul eder. Yerel aramada olduğu gibi, daha iyi bir komşuluk bulunduğu anda eldeki çözüm değişir. Yerel eniyiye ulaşıldığında ise, güncel çözümden daha kötü olan bir aday çözüm seçimi ile arama devam eder. Komşuluktaki en iyi çözüm, eldeki çözümü iyileştirmese de yeni güncel çözüm olarak seçilir [23]. Daha önceden gidilen komşuluklara tekrar gidilmesini önlemek amacıyla (çevrim oluşmasını engellemek için) yasaklı arama algoritması daha önceden ziyaret edilen komşulukları çıkartır. YA, yakınlarda uygulanan hareketleri ya da çözümlerin hafızasını yasak listesi (YL) ile dikkate alır. YL, kısa dönem hafızalı çalışmaktadır ve her bir iterasyonda bu hafıza güncellenir. Yasak listesindeki amaç, daha önceden yapılan hareketlerin ya da oluşturulan çözümlerin

tekrar oluşmasını engellemektir. Çünkü gidilen tüm çözümleri hafızada tutmak hem zaman hem de alan kaybına neden olur. YL, belirli sayıda yasak hareketi tutar ve genelde yasak listesinde hareketlerin özellikleri tutulmaktadır. YL, YA algoritmasının performansı açısından oldukça kritik bir parametredir. Listenin uzunluğunun az olması, çevrim oluşmasına neden olabilir. Listenin uzunluğunun artması aramanın çeşitlenmesine yardımcı olacaktır. Yasak listesinin uzunluğu statik, dinamik ya da uyarlanabilir olabilir [23]. Statik olduğunda, problemin ve komşuluğun büyüklüğüne bağlı olur. YL parametresinin ne olması gerektiği hakkında eniyi bir değer söylenemez. Bu yüzden YL boyutu farklı örnekler için çeşitlendirilerek uygulanır. Dinamik olduğunda, arama boyunca listenin uzunluğu değişir. Uyarlanabilir olduğunda ise, yasak listesinin boyutu arama hafızasına göre güncellenir. YA algoritmasının en kritik özelliklerinden biri yasak yıkma (aspirasyon) kriteri denilen durumdur. Yasak yıkma kriteri şu durumu sağlamaktadır: Eğer bir hareket eldeki çözümden daha iyi çözüme götürüyorsa yani iyi bir hareket ise ama yasak listesinde yer alıyorsa, yasak yıkma kriteri sayesinde bu hareket yasak listesinden çıkartılır ve hareketin yapılmasına izin verilir. Böylece daha önceden yasaklanmış harekete izin verilerek arama uzayı çeşitlendirilir ve yasaklı arama algoritmasının etkinliği artar. Böylelikle kabul edilebilir komşu çözümler, yasaklı olmayan çözümler ya da yasak yıkma kriterini sağlayan çözümler olur. YA içerisinde önemli olan diğer konular, kuvvetlendirme ve çeşitlendirme aşamalarıdır:

Güçlendirme (orta dönem hafıza): Amaç, elit çözümlerin özelliklerini tutarak, aramayı bu özelliklere sahip çözümlerin olduğu bölgelerde yapmaktır. Arama boyunca bulunan elit çözümleri tutar. Elit çözümlerin özelliklerine öncelik verme amacı vardır. Yakın zamanlı belleği kullanır. Arama bu özellikler doğrultusunda yanlıdır ve kuvvetlidir [23].

Çeşitlendirme (uzun dönem hafıza): Arama uzayının gidilmemiş bölgelerini araştırır. Oluşturulan çözümdeki elit çözümlerin özelliklerinden vazgeçilerek, arama uzayının gidilmemiş diğer bölgelerine doğru aramanın kaymasını sağlamaktadır.

YA algoritması, literatürde farklı yazarlar tarafından ARP'ye yönelik olarak uygulanmıştır. Bunlardan bazıları şunlardır [33]:

7.6.1 İki erken (two early) yasaklı arama algoritması

Willard [56] tarafından çalışılmıştır. Bu algoritmada çözüm öncelikle büyük bir tura dönüştürülür ve uygun çözümlere, güncel çözümden 3-opt hareketli ile ulaşılabilecek şekilde komşuluk yapısı oluşturulur. Bir sonraki çözüm yasaklı olmayan en iyi hareket ile hesaplanır. Genelde komşuluk yapısında swap (yer değiştirme) hareketi kullanılır ve iki farklı rotada bulunan müşterilerin yerleri uygunluk ve kapasite koşulları çerçevesinde değiştirilir. Her iterasyonda eniyi yasaklı olmayan uygun hareket seçilir.

7.6.2 Osman yasaklı arama algoritması

Osman [57], komşuluk yapısını λ -değişim üretme mekanizması (λ -interchange generation mechanism) ile yeniden tanımlamıştır. $\lambda=1$ olarak alınır. Algoritmanın en iyi kabul edilebilir versiyonunda, tüm komşuluklar aranır ve en iyi yasaklı olmayan uygun hareket seçilir. Diğer versiyon olan ilk en iyi kabul edilebilir versiyonda ise, eğer varsa, ilk olarak iyileşen kabul edilebilir hareket seçilir. Yoksa, en iyi kabul edilebilir hareket uygulanır.

7.6.3 Yasaklı rota algoritması

Gendreau ve ark. [58] tarafından geliştirilmiştir. Komşuluk yapısı tüm çözümler tarafından tanımlanır. Bu tanımlamaya göre tüm çözümler, eldeki çözümden, rotasındaki bir müşterinin kaldırılarak başka bir rotaya eklenmesi ile oluşur. Rotaya ekleme müşterinin p en yakın komşuluk yapısını içeren GENI (Generalized Insertion Procedure) ile gerçekleşir. GENI, GSP problemi için Gendreau ve ark. [59] tarafından geliştirilmiştir. Böylece herhangi var olan bir rota azaltılabilir ya da yeni bir rota eklenebilir. Algoritmanın farklı diğer bir yanı ise, arama sürecinde, kapasite ya da maksimum rota uzunluğu kısıtlarına göre uygun olmayan (infeasible) çözümlerin de incelenmesidir. Bu yüzden, amaç fonksiyonunda kapasite aşımından kaynaklanan ya da süre aşımından dolayı 2 adet ceza terimi bulunmaktadır. Her bir terim, kendini ayarlayabilen bir parametre ile

ağırlıklandırılmıştır. Buna göre; eğer önceki 10 adet çözüm uygun çözüm ise, her parametre 2'ye bölünür. Eğer uygun çözüm değilse, her parametre 2 ile çarpılır. Böylece, algorithmada uygun ve uygun olmayan çözümlerin yer alması, yerel minimuma takılma olasılığını azaltır. Arama sürecinin belirli noktalarında, yasaklı rota, henüz yeni eklenen müşteri noktaları bulunan rotaları yeniden iyileştirir. Bu prosedür, Unstringing Stringing (US) prosedürü ile gerçekleşir. US prosedürü, Gendreau ve ark. [59] tarafından GSP'de eniyileme sonrası amacı ile geliştirilmiştir. Yasaklı rota, yasak listesi kullanmaz. Yasak listesi yerine, rassal yasaklı etiketleri kullanır. Bir müşteri noktası ne zaman r rotasından s rotasına, t iterasyonunda hareket ettiyse, o müşteri noktasının r rotasına yeniden eklenmesi $t+\theta$ iterasyonuna kadar yasaktır. θ : tamsayılı rassal bir parametredir ve [5,10] aralığından seçilir. Çeşitlendirme stratejisinde, sıklıkla hareket ettirilen müşterilere bir ceza verilir. Böylece amaç, daha az hareket eden müşterilerin hareket etme olasılıklarının artırılmasıdır [58].

7.6.4 Taillard algoritması

Taillard [60] tarafından 1993 yılında geliştirilmiştir. Algorithmada rassal yasaklı süreleri ve sürekli çeşitlilik kullanılır. Komşuluk yapısı yasaklı rotadan farklıdır. Standart müşteri ekleme ve müşteri değişimleri kullanılır (swap, insert vb.). Problem, alt problemlere ayrıştırılır. Her bir alt problem paralel olarak birbirinden bağımsız bir şekilde çözülür. Alt problemin sınırları dinamik olarak yeniden tanımlanır. Yasaklı rotaya göre uygulanışı daha kolaydır.

7.6.5 Xu and Kelly algoritması

Bu algorithmada daha karmaşık bir komşuluk yapısı önermişlerdir [61]. İki rota arasındaki müşteri değişimi olarak swap hareketini ve yerel rota iyileştirmelerini kullanmışlardır. Araç kapasiteleri de göz önüne alınarak, müşteri noktalarının rotadan çıkartılma ya da rotaya eklenme maliyetlerini hesaplamak için bazı yaklaşımlar geliştirilmiştir. Rotayı yeniden eniyilemek, 3-opt hareketi ile

sağlanmaktadır. En iyi çözümlerin tutulduğu havuz hafızada tutulur ve arama yeni parametreler ile yeniden başlatılır.

7.6.6 Barbarasoğlu ve Özgür algoritması

Algoritma yasaklı aramadaki çeşitlendirme stratejisini kullanmadan uygulanmıştır [33]. Sadece uygun çözümler hesaplanarak komşu çözümler λ -değişim mekanizması ile oluşturulur. Rota iyileştirmesi 1-opt ile gerçekleştirilir.

8 GELİŞTİRİLEN MODELLER İÇİN ÇIKAN GAMS SONUÇLARI VE ANALİZİ

Geliştirilen tek amaçlı ve çok amaçlı matematiksel modeller, Golden ve ark. [28] tarafından geliştirilen test problemlerinden problem 13'ün ilk 10 şehir verisi üzerinde GAMS paket programı ile denenmiştir. CPLEX çözücüsü kullanılmıştır. Bu veriler literatürde Golden [28] tarafından oluşturulup Taillard [2] tarafından heterojen filoya sahip ARP için çözdürülmüştür. Çok amaçlı model çözümü için, en çok kullanılan skalerleştirme yöntemlerinden biri olan, ağırlıklandırılmış toplam yöntemi kullanılmıştır. Skalerleştirme, çok amaçlı problemin etkin çözümlerinin bir alt kümesini bulmak için, eklenen yeni değişkenler ve parametreler ile, çok amaçlı modele bağlı olan, tek amaçlı bir problemdir. Ağırlıklandırılmış toplam yönteminde amaçlara farklı ağırlıklar verilmektedir. Böylece hangi amacın ağırlığı fazlaysa önem derecesi de artmaktadır. Bu yöntemde verilen ağırlıklar parametredir. Ağırlıkların değişmesi, konveks problemler için güçlü etkin ve zayıf etkin çözümlerin hesaplanmasını sağlamaktadır.

HFARP ve HFAUARP için kullanılan şehir verileri Çizelge 8.1'de, araç verileri ise Çizelge 8.2'de gösterilmektedir. Şehirlerin koordinatları ve talep miktarları Golden [28] tarafından oluşturulan problem 13'e ait olan ilk 10 şehrin verileridir. Araç verileri de problem 13'e aittir. Tür enküçüklenmesi için tür maliyetleri ek olarak ilave edilmiştir.

Çizelge 8.1. HFARP VE HFAUARP için kullanılan şehir verileri

	Şehir koordinatları		Talep miktarları
	x	y	
1	30	40	0
2	37	52	7
3	49	49	30
4	52	64	16
5	20	26	9
6	40	30	21
7	21	47	15
8	17	63	19
9	31	62	23
10	52	33	11

Çizelge 8.2. HFARP VE HFAUARP için kullanılan araç verileri

Tür	Araç kapasitesi	Araç sayıları	Rota maliyeti	Depodan çıkış maliyeti	Tür maliyeti
1	20	4	1	20	20
2	30	2	1,1	35	35
3	40	4	1,2	50	50
4	70	4	1,7	120	60
5	120	2	2,5	225	75
6	200	1	3,2	400	90

8.1 HFARP için Çıkan GAMS Sonuçları

HFARP için elde edilen tek amaçlı modelin sonuçları Çizelge 8.3'te, çok amaçlı modelin sonuçları ise Çizelge 8.4'te gösterilmektedir. Çizelge 8.3'te tek amaçlı model eniyi çözümde 4 tür, toplamda 4 araç kullanarak toplam maliyeti 480.63 olarak hesaplamıştır.

Çizelge 8.3. HFARP için geliştirilen tek amaçlı modelin GAMS sonucu

$z = 480,63$

Rotalar	Kullanılan araçlar
1-8-1	1.tür 1. araç
1-7-5-1	2.tür 1.araç
1-9-4-1	3.tür 1. araç
1-2-3-10-6-1	4.tür 1. araç

Çizelge 8.4'te görüldüğü gibi, w_1 ağırlığı (toplam rota ve depodan çıkış maliyetini etkileyen ağırlık) 1, w_2 ağırlığı (kullanılan araç türünü etkileyen ağırlık) 9 iken, yani ikinci amaç birinci amaçtan 9 kat daha fazla önemli olduğunda, model kullanılan araç türü maliyetlerini en küçükmek için tek tür araç seçmiştir. 6. tür 1. araç ile rota tamamlanmıştır. Diğer taraftan, w_1 ağırlığı 9; w_2 ağırlığı 1 olduğunda ise, birinci amaç 9 kat daha önemli olduğundan, toplam rota ve depo maliyetlerini en küçükmek için 4 tür araç kullanılmış ve toplam maliyet 480,63 bulunmuştur. Tek amaçlıda bulunan eniyi değer ile aynıdır. Çizelge 8.4'te ağırlıklar değiştiğinde, modelin hangi araçları ve hangi yolları seçtiği detaylı olarak görülmektedir. Örneğin; w_1 ağırlığı 7, w_2 ağırlığı 3 olduğunda 4 tür araç, w_1 ağırlığı 6, w_2 ağırlığı 4 olduğunda ise, 3 tür araç kullanılmıştır.

Çizelge 8.4. HFARP için geliştirilen çok amaçlı modelin GAMS sonucu

w_1	w_2	z_1	z_2	Rotalar	Kullanılan araçlar
1	9	583,91	1	1-7-8-9-2-4-3-10-6-5-1	6. tür 1. araç
2	8	583,91	1	1-5-6-10-3-4-2-9-8-7-1	6. tür 1. araç
3	7	583,91	1	1-5-6-10-3-4-2-9-8-7-1	6. tür 1. araç
4	6	494,55	3	1-10-1	1.tür 1. araç
				1-5-6-1	2.tür 1.araç
				1-2-3-4-9-8-7-1	5.tür 1. araç
5	5	494,55	3	1-10-1	1.tür 1. araç
				1-6-5-1	2.tür 1.araç
				1-7-8-9-4-3-2-1	5.tür 1. araç
6	4	494,55	3	1-10-1	1.tür 1. araç
				1-5-6-1	2.tür 1.araç
				1-7-8-9-4-3-2-1	5.tür 1. araç
7	3	494,55	4	1-8-1	1.tür 1. araç
				1-5-7-1	2.tür 1.araç
				1-4-9-1	3.tür 1. araç
				1-2-3-10-6-1	4.tür 1. araç
8	2	480,63	4	1-8-1	1.tür 1. araç
				1-5-7-1	2.tür 1.araç
				1-4-9-1	3.tür 1. araç
				1-2-3-10-6-1	4.tür 1. araç
9	1	480,63	4	1-8-1	1.tür 1. araç
				1-7-5-1	2.tür 1.araç
				1-9-4-1	3.tür 1. araç
				1-2-3-10-6-1	4.tür 1. araç

8.2 HFAUARP için Çıkan GAMS Sonuçları

Çizelge 8.5 ve Çizelge 8.6 HFAUARP için çıkan tek ve çok amaçlı modellerin GAMS sonucunu göstermektedir. Çizelge 8.5'e göre eniyi çözüm tek amaçlı model için 187,66 olarak bulunmuştur.

Çizelge 8.5. HFAURP için geliştirilen tek amaçlı modelin GAMS sonucu

$$z = 187.66$$

Rotalar	Kullanılan araçlar
1-5	1. tür 1. araç
1-10	1. tür 2. araç
1-2-9	2. tür 1. araç
1-7-8	3. tür 1. araç
1-3-4	4. tür 1. araç
1-6	4. tür 2. araç

Çizelge 8.6'da görüldüğü gibi, w_1 ağırlığı (toplam rota ve depodan çıkış maliyetini etkileyen ağırlık) 1, w_2 ağırlığı (kullanılan araç türünü etkileyen ağırlık) 9 iken, ikinci amaç birinci amaçtan 9 kat daha fazla önemli iken model kullanılan araç türü maliyetlerini en küçüklemek için tek tür araç seçmiştir. 6. tür 1 araç ile rota tamamlanmıştır. w_1 ağırlığı 9, w_2 ağırlığı 1 olduğunda ise, birinci amaç 9 kat daha önemli olduğundan, toplam rota ve depo maliyetlerini en küçüklemek için 4 tür araç kullanılmış ve toplam maliyet 187.66 bulunmuştur. Tek amaçlıda bulunan eniyi değer ile aynıdır. w_1 ağırlığı 8, w_2 ağırlığı 2 olduğunda 4 tür araç; w_1 ağırlığı 5, w_2 ağırlığı 5 olduğunda ise yine 4 tür araç seçilmiştir. Farklı ağırlıklar ile amaçların nasıl değişkenlik gösterdiği görülmektedir.

Çizelge 8.6. HFAUARP için geliştirilen çok amaçlı modelin GAMS sonucu

w1	w2	z1	z2	Rotalar	Kullanılan araçlar
1	9	438.86	1	1-7-8-9-2-43-10-6-5	6. tür 1 araç
2	8	266.26	2	1-5	3. tür 1. araç
				1-2-9	3. tür 2. araç
				1-7-8	5. tür 1. araç
				1-6-10-3-4	5. tür 2. araç
3	7	266.26	2	1-2-9	3. tür 1. araç
				1-5	3. tür 2. araç
				1-7-8	5. tür 1. araç
				1-6-10-3-4	5. tür 2. araç
4	6	187.66	4	1-10	1. tür 1. araç
				1-5	1. tür 2. araç
				1-2-9	2. tür 2. araç
				1-7-8	3. tür 1. araç
				1-3-4	4. tür 1. araç
5	5	187.66	4	1-6	4. tür 2. araç
				1-10	1. tür 1. araç
				1-5	1. tür 2. araç
				1-2-9	2. tür 2. araç
				1-7-8	3. tür 1. araç
6	4	187.66	4	1-3-4	4. tür 1. araç
				1-6	4. tür 2. araç
				1-5	1. tür 1. araç
				1-10	1. tür 2. araç
				1-2-9	2. tür 1. araç
7	3	187.66	4	1-7-8	3. tür 1. araç
				1-6-3-4	4. tür 1. araç
				1-5	1. tür 1. araç
				1-10	1. tür 2. araç
				1-2-9	2. tür 1. araç
8	2	187.66	4	1-7-8	3. tür 1. araç
				1-3-4	4. tür 1. araç
				1-6	4. tür 2. araç
				1-5	1. tür 1. araç
				1-10	1. tür 2. araç
9	1	187.66	4	1-2-9	2. tür 1. araç
				1-7-8	3. tür 1. araç
				1-3-4	4. tür 1. araç
				1-6	4. tür 2. araç
				1-5	1. tür 1. araç

8.3 BTARP için Çıkan GAMS Sonuçları

Çizelge 8.7’de heterojen filolu BTARP’nin GAMS analizi için kullanılan şehir verileri, Çizelge 8.8’de ise heterojen filolu BTARP’nin GAMS analizi için kullanılan araç verileri görülmektedir. Veriler küçük boyutlu bölünebilir talepli problemi analiz etmek amacıyla oluşturulmuştur.

Çizelge 8.7. Heterojen filolu BTARP için kullanılan şehir verileri

	Şehir koordinatları		Talep miktarları
	x	y	
1	145	215	0
2	151	264	10
3	159	261	8
4	130	254	12
5	128	252	14

Çizelge 8.8. Heterojen filolu BTARP için kullanılan araç verileri

Tür	Araç kapasitesi	Araç sayıları	Rota maliyeti	Depodan çıkış maliyeti	Tür maliyeti
1	10	1	1	20	100
2	15	2	1,1	35	150
3	20	3	1,2	50	200

Çizelge 8.9, Heterojen filolu BTARP için geliştirilen tek amaçlı modelin GAMS sonucunu, Çizelge 8.10 ise çok amaçlı modelin sonucunu göstermektedir. Çizelge 8.9'a göre, tek amaçlı modelin eniyi sonucu 3 tür ve toplamda 3 tane araç kullanılarak bulunmuştur. 4. şehrin talebi bölünmüştür. 1. türün 1. aracı, 4. şehrin talebinin 9,996'lık kısmını karşılamıştır. 3. türün 1. aracı ise, 4. şehrin bölünmeden sonra kalan talebini karşılamıştır.

Çizelge 8.9. Heterojen filolu BTARP için geliştirilen tek amaçlı modelin GAMS sonucu

$$z = 770,60$$

Rotalar	Kullanılan araçlar
1-4(9,996)-1	1.tür 1. araç
1-5-1	2.tür 1. araç
1-4(2,004)-2-3-1	3.tür 1. araç

Çizelge 8.10. Heterojen filolu BTARP için geliştirilen çok amaçlı modelin GAMS sonucu

w1	w2	z1	z2	Rotalar	Kullanılan araçlar
1	9	926,40	1	1-4-1	3.tür 1. araç
				1-2-3-1	3.tür 2. araç
				1-5-1	3.tür 3. araç
2	8	926,40	1	1-2-3-1	3.tür 1. araç
				1-4-1	3.tür 2. araç
				1-5-1	3.tür 3. araç
3	7	926,40	1	1-4-1	3.tür 1. araç
				1-5-1	3.tür 2. araç
				1-2-3-1	3.tür 3. araç
4	6	926,40	1	1-4-1	3.tür 1. araç
				1-2-3-1	3.tür 2. araç
				1-5-1	3.tür 3. araç
5	5	812,40	2	1-5(9,996)-1	1.tür 1. araç
				1-2-3-1	3.tür 1. araç
				1-5(4,004)-4-1	3.tür 2. araç
6	4	812,40	2	1-5(9,996)-1	1.tür 1. araç
				1-2-3-1	3.tür 1. araç
				1-5(4,004)-4-1	3.tür 2. araç
7	3	812,40	2	1-5(9,996)-1	1.tür 1. araç
				1-5(4,004)-4-1	3.tür 1. araç
				1-3-2-1	3.tür 2. araç
8	2	770,60	3	1-4(2,004)-1	1.tür 1. araç
				1-5-1	2.tür 1. araç
				1-3-2-4(9,996)-1	3.tür 1. araç
9	1	770,60	3	1-4(9,996)-1	1.tür 1. araç
				1-5-1	2.tür 1. araç
				1-4(2,004)-2-3-1	3.tür 1. araç

Çizelge 8.10’da görüldüğü gibi, w_1 ağırlığı (toplam rota ve depodan çıkış maliyetini etkileyen ağırlık) 1, w_2 ağırlığı (kullanılan araç türünü etkileyen ağırlık) 9 iken, ikinci amaç birinci amaçtan 9 kat daha önemli olduğunda, model kullanılan araç türünü en küçüklemek için tek tür araç seçmiştir. 3. tür 3 araç ile rota tamamlanmıştır. w_1 ağırlığı 9, w_2 ağırlığı 1 olduğunda ise, birinci amaç 9 kat daha önemli olduğundan, toplam rota ve depo maliyetlerini en küçüklemek için 3 tür araç toplamda 3 araç kullanılmış ve toplam maliyet 770,60 olarak bulunmuştur. Tek amaçlıda bulunan eniyi değer ile aynıdır. 4. şehrin talebi bölünmüştür.

9 UYGULANAN YASAKLI ARAMA ALGORİTMASI VE ALGORİTMA SONUCUNDA ELDE EDİLEN SONUÇLARIN ANALİZİ

9.1 Uygulanan Yasaklı Arama Yöntemi

Tez kapsamında kullanılan yasaklı arama algoritmasında başlangıç çözümü, HFARP ve HFAUARP için hem en yakın komşuluk sezgiseli ile hem de rassal olarak, BTARP için ise en yakın komşuluk sezgiseli ile oluşturulmuştur.

Başlangıç çözümünde uygulanan en yakın komşuluk sezgiseli izleyen şekilde çalışmaktadır:

Adım 1) Depo noktasından başlanarak, kapasite kısıtları göz önüne alınarak, depoya en yakın şehir rotaya eklenir.

Adım 2) Rotaya eklenen şehre en yakın olan şehir yine kapasite kısıtları altında kontrol edilerek rotaya eklenir.

Adım 3) Kısıtlar sağlanmaması durumunda rota tamamlanır ve yeni bir rota oluşturulur.

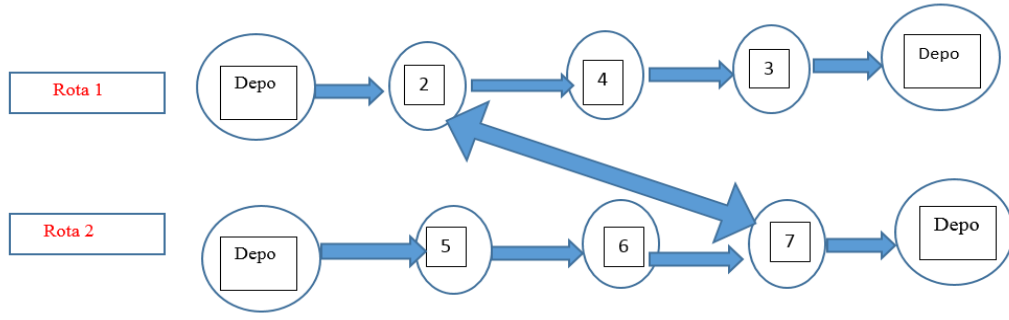
Başlangıç çözümü için kullanılan rassallık adımları izleyen şekildedir:

Adım 1) Problemdaki şehir sayısı n olmak üzere depo noktası 1'den n 'ye kadar olacak şekilde n 'lik bir dizi rasgele oluşturulur. Oluşturulan bu dizi daha sonra kullanılmak üzere hafızada tutulur.

Adım 2) Başlangıç çözümüne gelmeden, oluşturulan bu dizideki ilk eleman okunur ve şehirler listesinde hangi indekse denk geliyorsa o şehir kapasite kısıtlarına göre rotaya eklenir. Örneğin, dizideki birinci eleman 3. şehir ise, şehirlerin bulunduğu dosyaya gidilir ve 3. şehrin sahip olduğu parametreler okunarak rotaya başlanır.

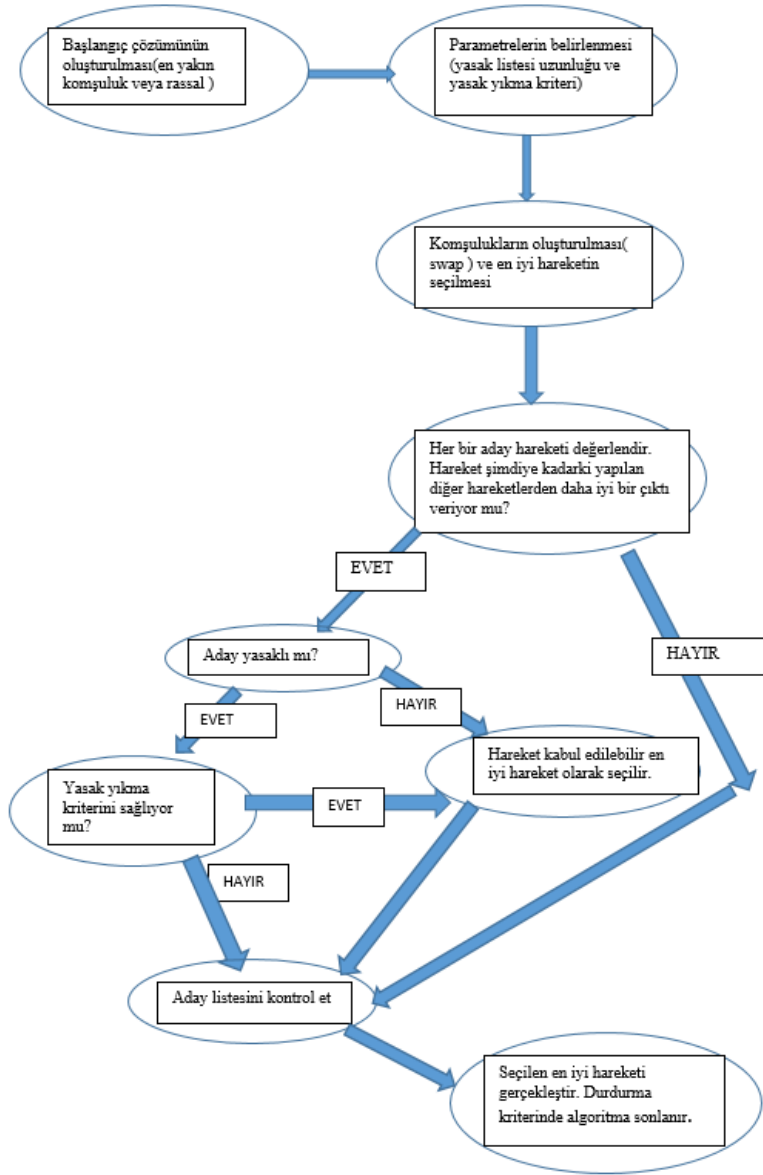
Adım 3) Kapasite kısıtları altında algoritma devam eder. Kapasite yeterli olmadığından yeni bir rota başlatılır.

Komşuluk yapısı olarak literatürde de ele alınan [52], müşterilerin yer değiştirilmesi hareketi (swap) kullanılmıştır. Yer değiştirme hareketine göre, farklı rotalarda bulunan iki müşterinin yeri değiştirilir. Bu hareket Şekil 9.1'de gösterilmektedir.



Şekil 9.1. İki farklı rotada bulunan müşterilerin swap (yer değiştirme) hareketi

Yasak listesinde, yer değiştirme hareketi uygulanan şehirler tutulmaktadır. Örneğin; farklı rotalarda bulunan 1. ve 4. şehirlerin yerleri değiştirildiyse bu hareket yapıldığı için artık yasaklı durumuna geldiğinden yasak listesine (1,4) olarak eklenir. Bu şehirler daha sonra belirli bir iterasyon sonucunda listeden çıkartılıp tekrar kullanılabilir. Aspirasyon kriteri (yasak yıkma kriteri) sağlanmış olur. Böylece daha önceden yasaklanmış harekete izin verilerek arama uzayı çeşitlendirilir ve yasaklı arama algoritmasının etkinliği artar. Böylelikle kabul edilebilir komşu çözümler yasaklı olmayan çözümler ya da yasak yıkma kriterini sağlayan çözümler olur. Algoritma çeşitlendirme ve kuvvetlendirme adımları ile durdurma kriterine kadar devam eder. Algoritmanın akış şeması Şekil 9.2’de gösterilmektedir.



Şekil 9.2. Yasaklı arama algoritması

Çok amaçlı modele uygulanan yasaklı arama algoritması adımları izleyen şekildedir:

Adım 1) Araç türü en küçüklenmesi hedeflendiğinden, öncelikle tür bazında toplam kapasiteler hesaplanır.

Adım 2) Eğer bir tane tür tüm şehirlere gidebiliyorsa o tür seçilir ve yasaklı arama algoritması adımları uygulanır.

Adım 3) Türün kapasitesi yetmiyorsa, türler arası geçişlerde ceza maliyeti (araç tür maliyeti) olduğundan, algoritma en küçük maliyetli türlere geçiş yaparak aramaya devam eder. Böylece fazla türlerin kullanılması bu ceza maliyetleri ile engellenmiş olur.

Bölünebilir talepli model için uygulanan yasaklı arama algoritması adımları izleyen şekildedir:

Adım 1) Öncelikle araçlar verilen kısıtlar altında rotalarına şehirleri ekleyerek devam eder.

Adım 2) Kapasitenin şehrin talebini karşılamadığı bir noktaya gelindiğinde araç kapasitesinden, rotadaki o ana kadar bulunan şehirlerin talepleri toplamı çıkartılır.

Adım 3) Hesaplanan bu değer, araca eklenecek olan şehrin talebine bölünür. Eğer çıkan sonuç, aracın bölme işlemi yapması için verilen bir eşik değerinden büyük ise bölmeye izin verilir.

Adım 4) Eşik değerinden küçük ise, şehrin talebi bölünmez ve rota tamamlanır. Bu uygulamada bölme eşik değeri parametresi 0.2 ve 0.4 olarak alınmıştır.

9.2 Uygulanan Yasaklı Arama Algoritması Sonuçları ve Analizi

Yasaklı arama algoritması C# programlama dili kullanılarak yazılmıştır ve intel core i7 işlemcili 16 GB RAM'i olan bilgisayarda koşturulmuştur. Her bir koşum ortalama 3 dakika sürmüştür. Farklı parametreler ve şehir sayılar için algoritmanın durdurma kriteri 1000 iterasyon olarak belirlenmiştir. Yasak listesi uzunluğu 7 veya 10 olarak alınmış, yasak yıkma kriteri ise 5 veya 10 iterasyon şeklinde değiştirilerek çıkan sonuçlar analiz edilmiştir. Kullanılan test verilerinde şehir sayıları 50, 75 ve 100'dür.

Başlangıç çözümü rassal olarak oluşturulan veri setleri 10 kez koşturulmuş ve yasaklı arama algoritmasının sonucunda bu 10 koşum sonucunda oluşan en iyi çözümler analizde verilmiştir. Kullanılan test problemleri Golden [28] ve Taillard [2] tarafından oluşturulan verilerdir. Çok amaçlı modellere yönelik araç sayıları bu çalışma kapsamında oluşturulmuştur. Böylece tür en küçüklenmesi

istendiğinde hangi türün kaç aracının kullanılacağı gözlemlenmek istenmiştir. Kullanılan test verilerinin özellikleri Çizelge 9.1’de verilmiştir. Bu verilerde şehir sayısı 50, 75 ve 100 olarak alınmıştır. Şehirlerin koordinatları ve talepleri bellidir. Araçların kilometre başına sahip oldukları maliyetler ve depodan çıkış maliyetleri bulunmaktadır. Tek amaçlıya yönelik kullanılan araç sayıları literatürden alınmıştır. Çok amaçlıya yönelik araç sayıları ise 10 olarak alınmış ve hangi türün kaç adet aracının seçileceği gözlemlenmek istenmiştir. Ayrıca, araç tür maliyetleri de eklenerek, tür enküçüklenmesi problemine yönelik bir çalışma yapılması hedeflenmiştir.

Çizelge 9.1. Yasaklı arama algoritmasında kullanılan veriler

Problem no	Şehir sayısı	Kilometre başı maliyet	Depodan çıkış maliyeti	Tek amaçlıda araç sayısı	Çok amaçlı için araç sayısı	Araç tür maliyeti
13	50	1. tür 1 2. tür 1,1 3. tür 1,2 4. tür 1,7 5. tür 2,5 6. tür 3,2	1. tür 20 2. tür 35 3. tür 50 4. tür 120 5. tür 225 6. tür 400	1. tür 4 2. tür 2 3. tür 4 4. tür 4 5. tür 2 6. tür 2	1. tür 10 2. tür 10 3. tür 10 4. tür 10 5. tür 10 6. tür 10	1. tür 10 2. tür 15 3. tür 20 4. tür 35 5. tür 60 6. tür 100
14	50	1. tür 1 2. tür 1,1 3. tür 1,4	1. tür 1000 2. tür 1500 3. tür 3500	1. tür 4 2. tür 2 3. tür 1	1. tür 10 2. tür 10 3. tür 10	1. tür 100 2. tür 150 3. tür 290
15	50	1. tür 1 2. tür 1,6 3. tür 2	1. tür 100 2. tür 250 3. tür 450	1. tür 5 2. tür 3 3. tür 2	1. tür 10 2. tür 10 3. tür 10	1. tür 10 2. tür 25 3. tür 45
16	50	1. tür 1 2. tür 1,6 3. tür 2,1	1. tür 100 2. tür 200 3. tür 400	1. tür 3 2. tür 4 3. tür 3	1. tür 10 2. tür 10 3. tür 10	1. tür 10 2. tür 20 3. tür 40
17	75	1. tür 1 2. tür 1,2 3. tür 1,5 4. tür 1,8	1. tür 25 2. tür 80 3. tür 150 4. tür 320	1. tür 6 2. tür 4 3. tür 2 4. tür 1	1. tür 10 2. tür 10 3. tür 10 4. tür 10	1. tür 10 2. tür 22 3. tür 40 4. tür 70
18	75	1. tür 1 2. tür 1,3 3. tür 1,9 4. tür 2,4 5. tür 2,9 6. tür 3,2	1. tür 10 2. tür 35 3. tür 100 4. tür 180 5. tür 400 6. tür 800	1. tür 4 2. tür 4 3. tür 2 4. tür 2 5. tür 1 6. tür 2	1. tür 10 2. tür 10 3. tür 10 4. tür 10 5. tür 10 6. tür 10	1. tür 10 2. tür 25 3. tür 50 4. tür 75 5. tür 125 6. tür 200
19	100	1. tür 1 2. tür 1,4 3. tür 1,7	1. tür 500 2. tür 1200 3. tür 2100	1. tür 4 2. tür 3 3. tür 3	1. tür 10 2. tür 10 3. tür 10	1. tür 10 2. tür 20 3. tür 30
20	100	1. tür 1 2. tür 1,7 3. tür 2	1. tür 100 2. tür 300 3. tür 500	1. tür 7 2. tür 5 3. tür 3	1. tür 10 2. tür 10 3. tür 10	1. tür 10 2. tür 23 3. tür 35

9.2.1 HFARP için uygulanan yasaklı arama algoritması sonuçları ve analizi

HFARP'nin başlangıç çözümü en yakın komşuluk sezgiseli ile ortaya çıkan algoritma sonuçları Çizelge 9.2'de gösterilmektedir. Buna göre, problem 18'de görüldüğü gibi, tek amaçlı modelde toplam maliyetler 8251,13 iken, tüm araçlar (6 tür toplamda 15 araç) kullanılmaktadır. Çok amaçlı model sonucuna bakıldığında, toplam maliyet 7535,78 olarak bulunmuştur ve tür en küçüklenmesi yapılarak 4. tür 10 araç ile rotalar tamamlanmıştır. Problem 19'da ise tek amaçlı modelde toplam maliyet 12635,68 olarak hesaplanmış, toplamda 3 tür kullanılmıştır. Çok amaçlı modelin sonucunda ise, toplam maliyette 76,77 kadarlık bir artış olmuş ama 2. tür 8 araç ile rotalar tamamlanmıştır. Diğer iki tür kullanılmamıştır. Problem 15'e bakıldığında, tek amaçlı modelin toplam maliyeti tüm türler kullanılarak 4109,82 olarak bulunmuş, çok amaçlı modelde ise 2. tür 9 araç ile toplam maliyet 4411,77 olarak hesaplanmıştır. Yasak listesi ve yasak yıkma kriterleri değiştirilerek bu koşulların analizi nasıl etkilediği gözlemlenmek istenmiştir. Literatürde elde edilen sonuçlara bakıldığında, sadece toplam rota maliyetlerinin hesaplatıldığı, depo maliyetleri ve tür maliyetlerinin hesaplanmadığı gözlemlenmiştir. Örneğin problem 13 için bulunan sonuç toplam rota maliyeti olarak literatürde 1517,84 olarak bulunmuştur ve toplamda 17 tane araç kullanılmıştır. Araçların depodan çıkma maliyetleri ve tür maliyetleri bu maliyet içinde göz önüne alınmamıştır. Problem 19'da toplam rota maliyeti literatürde 1117,51 olarak bulunmuş ve 10 adet araç kullanılmıştır. Tez kapsamında uygulanan yasaklı arama algoritması ile 9 adet araç kullanılmıştır. Maliyetler arasında oluşan fark, depodan çıkma maliyetleri ve tür maliyetlerinden kaynaklanmaktadır.

Çizelge 9.2. HFARP'nin başlangıç çözümü en yakın komşuluk yöntemi ile oluşturulan algoritma sonuçları

Problem no	n (şehir sayısı)	Tabu Listesi uzunluğu	Tabu yıkma	Toplam rota maliyeti	TEK AMAÇ			ÇOK AMAÇ		Literatürde elde edilen sonuç (rota maliyeti)	Kullanılan araç sayısı
					Depodan çıkma maliyeti	Rota+depo + tür maliyeti	Kullanılan araç sayısı	En yakın komşuluk sezgiseli			
								Toplam maliyet	Kullanılan araç sayısı		
13	50	7	5	2392,84	4392,84	4962,84	15	5596,35	6. tür 6 araç ile	1517,84	17
13	50	7	10	2392,84	4392,84	4962,84	15	5596,35	6. tür 6 araç ile		
13	50	10	5	2392,84	4392,84	4962,84	15	5596,35	6. tür 6 araç ile		
13	50	10	10	2392,84	4392,84	4962,84	15	5596,35	6. tür 6 araç ile	607,53	7
14	50	7	5	1126,23	11626,23	11717,23	7(hepsi)	15277,12	3. tür 4 araç		
14	50	7	10	1126,23	11626,23	11717,23	7(hepsi)	15277,12	3. tür 4 araç		
14	50	10	5	1126,23	11626,23	11717,23	7(hepsi)	15277,12	3. tür 4 araç	1015,29	9
14	50	10	10	1126,23	11626,23	11717,23	7(hepsi)	15277,12	3. tür 4 araç		
15	50	7	5	1744,82	3894,82	4109,82	10(hepsi)	4411,77	2. tür 9 araç		
15	50	7	10	1744,82	3894,82	4109,82	10(hepsi)	4411,77	2. tür 9 araç	1144,94	9
15	50	10	5	1744,82	3894,82	4109,82	10(hepsi)	4411,77	2. tür 9 araç		
15	50	10	10	1744,82	3894,82	4109,82	10(hepsi)	4411,77	2. tür 9 araç		
16	50	7	5	2437,63	4737,63	4967,63	10(hepsi)	5014,15	3. tür 6 araç	1061,96	11
16	50	7	10	2437,63	4737,63	4967,63	10(hepsi)	5014,15	3. tür 6 araç		
16	50	10	5	2437,63	4737,63	4967,63	10(hepsi)	5014,15	3. tür 6 araç		
16	50	10	10	2437,63	4737,63	4967,63	10(hepsi)	5014,15	3. tür 6 araç	1823,58	14
17	75	7	5	2193,81	3283,81	3581,81	13(hepsi)	3819,1	3. tür 8 araç		
17	75	7	10	2193,81	3283,81	3581,81	13(hepsi)	3819,1	3. tür 8 araç		
17	75	10	5	2193,81	3283,81	3581,81	13(hepsi)	3819,1	3. tür 8 araç	1117,51	10
17	75	10	10	2193,81	3283,81	3581,81	13(hepsi)	3819,1	3. tür 8 araç		
18	75	7	5	4656,13	7366,13	8251,13	15(hepsi)	7535,78	4. tür 10 araç		
18	75	7	10	4656,13	7366,13	8251,13	15(hepsi)	7535,78	4. tür 10 araç	1534,17	13
18	75	10	5	4656,13	7366,13	8251,13	15(hepsi)	7535,78	4. tür 10 araç		
18	75	10	10	4656,13	7366,13	8251,13	15(hepsi)	7535,78	4. tür 10 araç		
19	100	7	5	2675,68	12475,68	12635,68	9	12712,45	2. tür 8 araç	1534,17	13
19	100	7	10	2675,68	12475,68	12635,68	9	12712,45	2. tür 8 araç		
19	100	10	5	2675,68	12475,68	12635,68	9	12712,45	2. tür 8 araç		
19	100	10	10	2675,68	12475,68	12635,68	9	12712,45	2. tür 8 araç	1534,17	13
20	100	7	5	2992,42	6692,42	6982,42	15(hepsi)	8497,79	3. tür 8 araç		
20	100	7	10	2992,42	6692,42	6982,42	15(hepsi)	8497,79	3. tür 8 araç		
20	100	10	5	2992,42	6692,42	6982,42	15(hepsi)	8497,79	3. tür 8 araç	1534,17	13
20	100	10	10	2992,42	6692,42	6982,42	15(hepsi)	8497,79	3. tür 8 araç		

HFARP'nin başlangıç çözümü rassal olarak oluşturulan yasaklı arama algoritması sonuçları ise Çizelge 9.3'te gösterilmektedir. Problem 15'de görüldüğü gibi, tek amaçlıda elde edilen toplam maliyet 4108,74 olup tüm türler kullanılarak 10 araç ile rotalar tamamlanmıştır. Çok amaçlıda ise, toplamda 4141,99 maliyet ile 2. tür 9 araç rotaları tamamlamaya yetmiştir. Hem kullanılan türlerde bir azalma, hem de araç sayısında bir azalma meydana gelmiştir. Toplam maliyetteki artış ise sadece 33,25 kadar olmuştur. Problem 18'de ise, yasak listesi uzunluğu 7 ve yasak yıkma kriteri 5 olduğunda, çok amaçlı modeldeki meydana gelen toplam maliyet iyileşmesi %5 kadar olmuştur. Ayrıca tek amaçlıda 6 tür araç kullanılmıştır. Çok amaçlıda ise tek tür 10 araç ile rotalar tamamlanmıştır. Problem 19'da ise YL uzunluğu 7, yasaklı yıkma kriteri 10 olduğunda, tek amaçlıda toplam maliyet 13523,82 olarak hesaplanmıştır. Tüm türler kullanılmıştır. Çok amaçlıda ise, sadece 2. tür 8 araç kullanılmış olup, toplam maliyet %4 azalmıştır. Problem 15 için,

literatürde bulunan toplam rota maliyeti sonucu 1015,29'dur. Toplamda tüm türler ve 10 adet araç kullanılmıştır. Tez kapsamında uygulanan yasaklı arama algoritması sonucunda ise yaklaşık %20'lik bir fark ile sadece 2. tür 9 araç ile rotalar tamamlanmıştır.

Çizelge 9.3. HFARP'nin başlangıç çözümü rassal olarak oluşturulan algoritma sonuçları

Problem no	n şehir sayısı	Tabu Liste	Tabu yıkma	TEK AMAÇ				ÇOK AMAÇ		Literatürde elde edilen sonuç (rota maliyeti)	Kullanılan araç sayısı
				Toplam rota maliyeti	Depodan çıkma maliyeti	Rota+depo+tür maliyeti	Kullanılan araç sayısı	Rota+depo+tür maliyeti	Kullanılan araç sayısı		
13	50	7	5	2638,89	4113,95	5226,86	18(hepsi)	5390,14	6. tür 6 araç	1517,84	17
13	50	7	10	2498,23	4150,051	5438,5	15	6127,04	6. tür 6 araç		
13	50	10	5	2463,43	4805,56	5322,39	18(hepsi)	6085,09	6. tür 6 araç		
13	50	10	10	2553,99	4695,7	4979,07	15	5859,03	6. tür 6 araç		
14	50	7	5	1080,13	11496,89	11655,46	7(hepsi)	15133,31	3. tür 4 araç	607,53	7
14	50	7	10	1171,14	11496,89	11747,49	7(hepsi)	15168,62	3. tür 4 araç		
14	50	10	5	1183,71	11623,53	11780,2	7(hepsi)	15392,41	3. tür 4 araç		
14	50	10	10	1140,47	11608,95	11755,49	7(hepsi)	15125,57	3. tür 4 araç		
15	50	7	5	1320,54	3899,4	4108,74	10(hepsi)	4141,99	2. tür 9 araç	1015,29	10
15	50	7	10	1913,69	4250,85	4374,32	10(hepsi)	4149,24	2. tür 9 araç		
15	50	10	5	1604,82	3756,64	3945,91	10(hepsi)	4242,02	2. tür 9 araç		
15	50	10	10	1718,36	3861	4175,87	10(hepsi)	4224,99	2. tür 9 araç		
16	50	7	5	2145,77	4131,44	5042,91	10(hepsi)	5146,22	3. tür 6 araç	1144,94	9
16	50	7	10	2324,48	4225,02	4651,12	10(hepsi)	5028,29	3. tür 6 araç		
16	50	10	5	2316,39	4287,09	4471,94	10(hepsi)	5521,87	3. tür 6 araç		
16	50	10	10	1955,84	4285,35	4887,57	10(hepsi)	4845,65	3. tür 6 araç		
17	75	7	5	2232,34	3191,78	3668,52	13(hepsi)	4018,59	3. tür 8 araç	1061,96	11
17	75	7	10	2072,6	3792,16	3170,15	13(hepsi)	3566,41	3. tür 8 araç		
17	75	10	5	2074,65	3233,62	3722,46	13(hepsi)	3694,85	3. tür 8 araç		
17	75	10	10	2126,89	3385,14	3749,49	13(hepsi)	3772,36	3. tür 8 araç		
18	75	7	5	3682,8	6092,88	7012,94	14	6661,09	4.tür 10 araç	1823,58	14
18	75	7	10	3853,87	6114,4	8142,29	15(hepsi)	6977,01	4.tür 10 araç		
18	75	10	5	3828,24	6682,71	7304,11	14	6080,45	4.tür 10 araç		
18	75	10	10	3778,06	7238,48	7370,3	14	6405,93	4.tür 10 araç		
19	100	7	5	3266,4	13330,5	14452,39	9	13337,45	2. tür 8 araç	1117,51	10
19	100	7	10	3984,5	16593,56	13523,82	9	12986,54	2. tür 8 araç		
19	100	10	5	3259,7	12260,7	14516,23	9	15678,2	2. tür 8 araç		
19	100	10	10	3100,48	13345,2	15692,56	9	16994,23	2. tür 8 araç		
20	100	7	5	3328,67	7652,25	7140,26	15(hepsi)	8527,07	3. tür 8 araç	1534,17	13
20	100	7	10	3787,43	7860,61	7267,18	15(hepsi)	8396,23	3. tür 8 araç		
20	100	10	5	3463,85	7839,67	7816,72	15(hepsi)	7918,24	3. tür 8 araç		
20	100	10	10	3759,85	7434,14	7329,74	15(hepsi)	8531,93	3. tür 8 araç		

9.2.2 HFAUARP için uygulanan yasaklı arama algoritması sonuçları ve analizi

HFAURP'nin başlangıç çözümü en yakın komşuluk sezgiseli ile ortaya çıkan algoritma sonuçları Çizelge 9.4'te gösterilmektedir. Problem 18'de görüldüğü gibi, tek amaçlı sonucunda toplamda 15 araç ve 6 tür kullanılmasına rağmen, çok amaçlıda 4. tür 10 araç ile tür enküçüklenmesi yapılmıştır. Ayrıca toplam maliyette % 22,3 kadarlık bir iyileşme meydana gelmiştir. Diğer test problemlerinde de yine çok amaçlı modellerde, tek tür araç kullanılmıştır. Problem 17'de görüldüğü gibi tek amaçlı model sonucunda toplam maliyet 3125,69 olarak bulunmuştur ve tüm türler kullanılmıştır. Çok amaçlı model sonucuna bakıldığında ise, toplam maliyet 3652,77 olarak hesaplanmış ve uygulanan yasaklı arama yöntemi ile 3. tür 8 araç ile rotalar tamamlanmıştır.

Çizelge 9.4. HFAUARP'nin başlangıç çözümü en yakın komşuluk yöntemi ile oluşturulan algoritma sonuçları

Problem no	n (şehir sayısı)	Tabu listesi uzunluğu	Tabu yıkma	TEK AMAÇ				ÇOK AMAÇ	
				En yakın komşuluk				En yakın komşuluk	
				Toplam rota maliyeti	Depodan çıkma maliyeti	Rota+depo + tür maliyeti	Kullanılan araç sayısı	Rota+depo+ tür maliyeti	Kullanılan araç sayısı
13	50	7	5	1811,93	3816,98	4381,93	14	6376,82	6. tür 6 araç
13	50	7	10	1811,93	3816,98	4386,98	14	6376,82	6. tür 6 araç
13	50	10	5	1811,93	3811,93	4381,93	14	6376,82	6. tür 6 araç
13	50	10	10	1816,98	3816,98	4386,98	14	6376,82	6. tür 6 araç
14	50	7	5	1119,85	11619,85	12609,85	7(hepsi)	10864,35	1. tür 9 araç
14	50	7	10	1119,85	11619,85	12609,85	7(hepsi)	10864,35	1. tür 9 araç
14	50	10	5	1119,85	11619,85	12609,85	7(hepsi)	10864,35	1. tür 9 araç
14	50	10	10	1119,85	11619,85	12609,85	7(hepsi)	10864,35	1. tür 9 araç
15	50	7	5	1513,93	3663,93	3878,93	10(hepsi)	10054,3	1. tür 9 araç
15	50	7	10	1513,93	3663,93	3878,93	10(hepsi)	10054,3	1. tür 9 araç
15	50	10	5	1513,93	3663,93	3878,93	10(hepsi)	10054,3	1. tür 9 araç
15	50	10	10	1513,93	3663,93	3878,93	10(hepsi)	10054,3	1. tür 9 araç
16	50	7	5	1943,47	4243,47	4473,47	10(hepsi)	4845,49	3. tür 6 araç
16	50	7	10	1943,47	4243,47	4473,47	10(hepsi)	4845,49	3. tür 6 araç
16	50	10	5	1943,47	4243,47	4473,47	10(hepsi)	4845,49	3. tür 6 araç
16	50	10	10	1943,47	4243,47	4473,47	10(hepsi)	4845,49	3. tür 6 araç
17	75	7	5	1737,69	2827,69	3125,69	13(hepsi)	3652,77	3. tür 8 araç
17	75	7	10	1737,69	2827,69	3125,69	13(hepsi)	3652,77	3. tür 8 araç
17	75	10	5	1737,69	2827,69	3125,69	13(hepsi)	3652,77	3. tür 8 araç
17	75	10	10	1737,69	2827,69	3125,69	13(hepsi)	3652,77	3. tür 8 araç
18	75	7	5	4208,44	6918,44	7803,44	15(hepsi)	6062,63	4.tür 10 araç
18	75	7	10	4208,44	6918,44	7803,44	15(hepsi)	6062,63	4.tür 10 araç
18	75	10	5	4208,44	6918,44	7803,44	15(hepsi)	6062,63	4.tür 10 araç
18	75	10	10	4208,44	6918,44	7803,44	15(hepsi)	6062,63	4.tür 10 araç
19	100	7	5	2314,38	12114,38	12274,38	10(hepsi)	12405,72	2. tür 8 araç
19	100	7	10	2314,38	12114,38	12274,38	10(hepsi)	12405,72	2. tür 8 araç
19	100	10	5	2314,38	12114,38	12274,38	10(hepsi)	12405,72	2. tür 8 araç
19	100	10	10	2314,38	12114,38	12274,38	10(hepsi)	12405,72	2. tür 8 araç
20	100	7	5	2792,57	6492,57	6782,57	15(hepsi)	8497,79	3. tür 8 araç
20	100	7	10	2792,57	6492,57	6782,57	15(hepsi)	8497,79	3. tür 8 araç
20	100	10	5	2792,57	6492,57	6782,57	15(hepsi)	8497,79	3. tür 8 araç
20	100	10	10	2792,57	6492,57	6782,57	15(hepsi)	8497,79	3. tür 8 araç

HFAURP'nin başlangıç çözümü rassal olarak oluşturulan algoritma sonuçları Çizelge 9.5'te gösterilmektedir. Problem 14'de yasak listesi uzunluğu 10 ve yasak yıkma kriteri 10 olduğunda, 3 tür toplam 7 araç ile tek amaçlı model sonucu elde edilmiştir. Çok amaçlı sonucuna göre ise 3. tür 4 araç kullanılmış ve toplam maliyette yaklaşık % 0,5'lik bir iyileşme elde edilmiştir. Problem 16'da ise, yasak liste uzunluğu 10, yasak yıkma kriteri 5 iken, tek amaçlı modelde tüm türler ve toplamda 10 araç kullanılmıştır. Çok amaçlı sonucuna göre ise, 3. tür 6 araç ile toplam maliyette % 1,2'lik bir iyileştirme gerçekleştirilmiştir. Problem 17'de yasak liste uzunluğu 7, yasak yıkma kriteri 10 iterasyon iken, çok amaçlı modelde toplam maliyette % 1,95'lik bir iyileştirme yapılmıştır ve rotalar 3. tür 8 araç ile tamamlanmıştır. Problem 19'da ise yasak liste uzunluğu 10, yasak yıkma kriteri 10 iken, çok amaçlı modelde toplam maliyette yaklaşık %1,5 kadarlık bir iyileşme gerçekleşmiştir. Çok amaçlı modellerde, tek tür aracın kullanılarak rotaların tamamlandığı gözlemlenmiştir.

Çizelge 9.5. HFAUARP'nin başlangıç çözümü rassal olarak oluşturulan algoritma sonuçları

Problem no	n (şehir sayısı)	Tabu listesi uzunluğu	Tabu yıkma	TEK AMAÇ				ÇOK AMAÇ		Kullanılan araç sayısı
				Rassal				Rassal		
				Toplam rota maliyeti	Depodan çıkma maliyeti	Rota+depo+tür maliyeti	Kullanılan araç sayısı	Rota+depo+tür maliyeti	Kullanılan araç sayısı	
13	50	7	5	1900,52	3715,42	4100,25	14	6211,45	6. tür 6 araç	
13	50	7	10	1877,55	3644,12	4215,89	14	6544,47	6. tür 6 araç	
13	50	10	5	1956,64	3847,56	4390,47	14	6632,15	6. tür 6 araç	
13	50	10	10	1846,52	3755,41	4412,69	16	6542,12	6. tür 6 araç	
14	50	7	5	920,3	11354,61	15463,88	7(hepsi)	16250,75	3. tür 4 araç	
14	50	7	10	952,14	12356,42	16213,24	7(hepsi)	16350,2	3. tür 4 araç	
14	50	10	5	964,53	11731,99	15478,23	7(hepsi)	16272,54	3. tür 4 araç	
14	50	10	10	985,62	12541,26	16952,3	7(hepsi)	16897,56	3. tür 4 araç	
15	50	7	5	1481,85	3923,19	3988,45	10(hepsi)	9956,47	1. tür 9 araç	
15	50	7	10	1512,12	3838,56	3995,64	10(hepsi)	10056,23	1. tür 9 araç	
15	50	10	5	1601,56	3434,89	3654,21	10(hepsi)	9854,12	1. tür 9 araç	
15	50	10	10	1615,22	3654,12	3875,46	10(hepsi)	9932,41	1. tür 9 araç	
16	50	7	5	1855,46	4566,21	4852,77	10(hepsi)	4912,55	3. tür 6 araç	
16	50	7	10	1896,32	4356,89	4744,46	10(hepsi)	4753,21	3. tür 6 araç	
16	50	10	5	1944,56	4415,66	4715,23	10(hepsi)	4655,12	3. tür 6 araç	
16	50	10	10	2003,41	4502,13	4879,56	10(hepsi)	4899,52	3. tür 6 araç	
17	75	7	5	1985,46	2954,12	3478,56	13(hepsi)	3678,99	3. tür 8 araç	
17	75	7	10	1857,63	3042,54	3654,61	13(hepsi)	3245,61	3. tür 8 araç	
17	75	10	5	1756,44	3124,56	3541,72	13(hepsi)	3755,17	3. tür 8 araç	
17	75	10	10	1895,46	3247,16	3655,77	13(hepsi)	3547,89	3. tür 8 araç	
18	75	7	5	3709,02	7012,56	7653,45	15(hepsi)	7456,33	4.tür 10 araç	
18	75	7	10	3894,56	7125,44	7664,52	15(hepsi)	7895,41	4.tür 10 araç	
18	75	10	5	3605,78	7263,99	7569,44	15(hepsi)	7754,13	4.tür 10 araç	
18	75	10	10	3724,12	7965,41	8004,56	15(hepsi)	7965,52	4.tür 10 araç	
19	100	7	5	2899,41	14564,78	16542,33	10(hepsi)	17841,22	2. tür 8 araç	
19	100	7	10	2645,79	13564,89	15478,99	10(hepsi)	16984,53	2. tür 8 araç	
19	100	10	5	2654,36	12758,96	14563,01	10(hepsi)	16547,99	2. tür 8 araç	
19	100	10	10	2755,41	13564,9	15879,22	10(hepsi)	15642,33	2. tür 8 araç	
20	100	7	5	3586,02	6410,98	6961,08	15(hepsi)	7918,54	3. tür 8 araç	
20	100	7	10	3654,2	6563,4	6742,13	15(hepsi)	7654,13	3. tür 8 araç	
20	100	10	5	3659,84	6452,19	6830,15	15(hepsi)	7326,58	3. tür 8 araç	
20	100	10	10	3470,23	6678,95	6955,41	15(hepsi)	7456,23	3. tür 8 araç	

9.2.3 BTARP için uygulanan yasaklı arama algoritması sonuçları ve analizi

Heterojen filolu BTARP'nin başlangıç çözümü en yakın komşuluk sezgiseli ile ortaya çıkan algoritma sonuçları Çizelge 9.6'da gösterilmektedir. Problem 15'te görüldüğü gibi, tek amaçlıda toplam maliyet tüm türler kullanılarak 3750,95 olarak hesaplanmıştır. Çok amaçlıda ise, tek tür kullanılarak 3536,62 olarak bulunmuştur. Maliyette %5,7'lik bir iyileşme sağlanmıştır. Problem 17'de ise yasak listesi uzunluğu 7, yasak yıkma kriteri 5 olduğunda, çok amaçlı modelin toplam maliyeti, tek amaçlı modelin toplam maliyetine göre % 7,4'lük bir iyileşme elde edilmiştir. Problem 19'da ise, çok amaçlı sonuçta hem toplam maliyetler azalmış hem de 2. tür 8 araç ile rotalar tamamlanmıştır. Tek amaçlıda tüm türler kullanılmasına rağmen, çok amaçlı model tek tür araç ile rotasını tamamlayabilmiştir. Çıkan sonuçlar, tür enküçüklenmesinin önemini vurgulamak ile birlikte, farklı türlerdeki araçların kullanılmasının toplam maliyeti nasıl etkilediğini göstermektedir. Ayrıca, BTARP için seçilen bölme eşik değeri parametresi değıştikçe farklı sonuçlar elde edilmiştir. Örneğin, Problem 13'te, bölme eşik değeri parametresi 0.2 iken, tek amaçlı modelde toplam maliyet 4248,33, bölme eşik değeri parametresi 0.4 olduğunda ise, toplam maliyet 4236,18 olarak hesaplanmıştır. Tek amaçlı modelde bu problem için tüm türler kullanılmasına rağmen, çok amaçlı model sonucunda 5. tür 9 araç ile rotalar tamamlanmıştır.

Çizelge 9.6. Heterojen filolu BTARP'nin başlangıç çözümü en yakın komşuluk yöntemi ile oluşturulan algoritma sonuçları

Problem no	n (şehir sayısı)	Tabu listesi uzunluğu	Tabu yıkma	Bölme eşik değeri	TEK AMAÇ				ÇOK AMAÇ		Kullanılan araç sayısı
					En yakın komşuluk sezgiseli				En yakın komşuluk sezgiseli		
					Toplam rota maliyeti	Depodan çıkma maliyeti	Rota+depo + tür maliyeti	Kullanılan araç sayısı	Rota+depo+ tür maliyeti		
13	50	7	5	0,2	2058,33	3738,33	4248,33	18(hepsi)	4539,33	5. tür 9 araç	
13	50	7	10	0,2	2058,33	3738,33	4248,33	18(hepsi)	4539,33	5. tür 9 araç	
13	50	7	5	0,4	2046,18	3726,18	4236,18	18(hepsi)	4514,64	5. tür 9 araç	
13	50	7	10	0,4	2046,18	3726,18	4236,18	18(hepsi)	4514,64	5. tür 9 araç	
14	50	7	5	0,2	774,91	11274,91	12264,91	7(hepsi)	9879,73	1. tür 9 araç	
14	50	7	10	0,2	774,91	11274,91	12264,91	7(hepsi)	9879,73	1. tür 9 araç	
14	50	7	5	0,4	765,03	11265,03	12255,03	7(hepsi)	9869,85	1. tür 9 araç	
14	50	7	10	0,4	765,03	11265,03	12255,03	7(hepsi)	9869,85	1. tür 9 araç	
15	50	7	5	0,2	1385,95	3535,95	3750,95	10(hepsi)	3536,62	2. tür 8 araç	
15	50	7	10	0,2	1385,95	3535,95	3750,95	10(hepsi)	3536,62	2. tür 8 araç	
15	50	7	5	0,4	1385,95	3535,95	3750,95	10(hepsi)	3538,62	2. tür 8 araç	
15	50	7	10	0,4	1385,95	3535,95	3750,95	10(hepsi)	3538,62	2. tür 8 araç	
16	50	7	5	0,2	1517,47	3817,47	4047,47	10(hepsi)	4317,46	3. tür 5 araç	
16	50	7	10	0,2	1517,47	3817,47	4047,47	10(hepsi)	4317,46	3. tür 5 araç	
16	50	7	5	0,4	1458,54	3758,54	3988,54	10(hepsi)	4305,2	3. tür 5 araç	
16	50	7	10	0,4	1458,54	3758,54	3988,54	10(hepsi)	4305,2	3. tür 5 araç	
17	75	7	5	0,2	1462,35	2552,35	2850,35	13(hepsi)	2635,93	3. tür 7 araç	
17	75	7	10	0,2	1462,35	2552,35	2850,35	13(hepsi)	2635,93	3. tür 7 araç	
17	75	7	5	0,4	1451,2	2541,2	2839,2	13(hepsi)	2612,02	3. tür 7 araç	
17	75	7	10	0,4	1451,2	2541,2	2839,2	13(hepsi)	2612,02	3. tür 7 araç	
18	75	7	5	0,2	2466,54	4406,54	5121,54	15(hepsi)	5053,58	4. tür 10 araç	
18	75	7	10	0,2	2466,54	4406,54	5121,54	15(hepsi)	5053,58	4. tür 10 araç	
18	75	7	5	0,4	2439,54	4379,54	5094,1	15(hepsi)	4986,7	4. tür 10 araç	
18	75	7	10	0,4	2439,54	4379,54	5094,1	15(hepsi)	4986,7	4. tür 10 araç	
19	100	7	5	0,2	1532	11332	11492	10(hepsi)	11405,7	2. tür 8 araç	
19	100	7	10	0,2	1532	11332	11492	10(hepsi)	11405,7	2. tür 8 araç	
19	100	7	5	0,4	1524,76	11324,76	11484,76	10(hepsi)	11411	2. tür 8 araç	
19	100	7	10	0,4	1524,76	11324,76	11484,76	10(hepsi)	11411	2. tür 8 araç	
20	100	7	5	0,2	2151,26	5351,26	5606,26	15(hepsi)	6630,8	3. tür 8 araç	
20	100	7	10	0,2	2151,26	5351,26	5606,26	15(hepsi)	6630,8	3. tür 8 araç	
20	100	7	5	0,4	2086,51	5286,51	5541,51	15(hepsi)	6652,7	3. tür 8 araç	
20	100	7	10	0,4	2086,51	5286,51	5541,51	15(hepsi)	6652,7	3. tür 8 araç	

10 SONUÇ VE ÖNERİLER

ARP, gerçek hayat problemlerinde sıkça karşılaşılan bir problemdir. Literatürde probleme yönelik çok sayıda çalışma bulunmaktadır. Bu çalışmada klasik ARP, AUARP ve BTARP üzerinde çalışılmış olup filo türü olarak heterojen filo seçilmiştir. Literatürde, heterojen filolu bu problemler için kullanılan araç türü en küçüklenmesine yönelik bir çalışmaya rastlanmaması bu tez çalışmasının motivasyonu olmuştur. Çünkü, gerçek hayat uygulamalarında bir firmanın, müşterilerinin taleplerini karşılaması için tüm türlere sahip olması her zaman mümkün olmamaktadır. Dışarıdan kiralanan araçların maliyetleri, farklı türlerdeki araçların kullanılmasından kaynaklanan amortisman ve bakım maliyetleri, farklı türleri depoda tutma maliyetleri firmalar için oldukça yüksek maliyetlere neden olur. Bu yüzden kullanılan araç türü en küçüklenmesi uygulamalarda büyük önem taşır.

Literatürde incelenen bu üç probleme yönelik öncelikle karma tamsayılı tek amaçlı ve çok amaçlı matematiksel modeller geliştirilmiştir. Küçük boyutlu problemler için GAMS paket programı kullanılmıştır. Çok amaçlıda, amaçlara verilen ağırlıkların değişmesi ile amaçların etkin olarak değiştikleri gözlemlenmiştir. Çok amaçlı küçük boyutlu problemlerin çözümünde ağırlıklandırılmış toplam yöntemi kullanılmıştır. Büyük boyutlu problemler için ise literatürde de ARP'nin çözüm yöntemi olarak çok kullanılan yasaklı arama algoritması kullanılmıştır. Tek nokta temelli arama yapan bu algoritma, etkin bir şekilde kodlanarak, tek amaçlı ve çok amaçlı problemlerin sonuçları kıyaslamalı olarak verilmiştir. Çok amaçlı modellerde, tek amaçlı modellere kıyasla algoritma sonucunda daha az tür araç kullanılarak rotaların tamamlandığı analiz edilmiştir.

Gelecek çalışmalar için, popülasyon temelli metasezgisel algoritmalarından genetik algoritma veya farklı metasezgisel yöntemler çalışılarak, bu sonuçlar kendi aralarında kıyaslanabilir. Araç türü en küçüklenmesinin yanı sıra modellere farklı amaçlar da eklenebilir.

11 KAYNAKÇA

- [1] Dantzig, G.B. ve Ramser, J. H., The Truck Dispatching Problem, *Management Science*, 6, 1, 80-91, 1959.
- [2] Taillard, E., A heuristic column generation method for the heterogeneous fleet vehicle routing problem, *Operations Research*, 33, 1, 1-14, 1999.
- [3] Gendreau, M., Laporte, G., Musaraganyi, C. ve Taillard, E. D., A tabu search heuristic for the heterogeneous fleet vehicle routing problem, *Computers & Operations Research*, 26, 1153-1173, 1999.
- [4] Renaud, J. ve Boctor, F. F., A sweep-based algorithm for the fleet size and mix vehicle routing problem, *European Journal of Operations Research*, 140, 618-628, 2002.
- [5] Trantilis, C., Kiranoudis, C., ve Vassiliadis, V., A threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem, *European Journal of Operational Research*, 152, 148-158, 2004.
- [6] Li, F., Golden, B. ve Wasil, E., A record-to-record travel algorithm for solving the heterogeneous fleet vehicle routing problem, *Computers and Operations Research*, 34, 2734-2742, 2007.
- [7] Brandao, J., A deterministic tabu search algorithm for the fleet size and mix vehicle routing problem, *European Journal of Operational Research*, 195, 716-728, 2009.
- [8] Imran, A., Salhi, S., ve Wassan, N. A., A variable neighborhood based algorithm for the heterogeneous fleet vehicle routing problem, *European Journal of Operational Research*, 197, 509-518, 2009.
- [9] Belfiore, P. ve Yoshizaki, H. T., A scatter search for a real life heterogeneous fleet vehicle routing problem with time windows and split deliveries in Brazil, *European Journal of Operational Research*, 199, 750-758, 2009.
- [10] Euchi, J. ve Chabchoub, H., A hybrid tabu search to solve the heterogeneous fixed fleet vehicle routing problem, *Logistics Research*, 2, 3-11, 2010.

- [11] Liu, S., A hybrid population heuristic for the heterogeneous vehicle routing problems, *Transportation Research Part E*, 54, 67-78, 2013.
- [12] Brandao, J., A tabu search algorithm for the open vehicle routing problem, *European Journal of Operational Research*, 157, 552-564, 2004.
- [13] Yu, S., Ding, C. ve Zhu, K., A hybrid GA-TS algorithm for open vehicle routing optimization of coal mines material, *Expert Systems with Applications*, 38, 10568-10573, 2011.
- [14] Li, X., Leung, S. C. ve Tian, P., A Multi Start Adaptive Memory-based Tabu Search Algorithm for the Heterogeneous Fixed Fleet Open Vehicle Routing Problem, *Expert Systems with Applications*, 39, 365-374, 2012.
- [15] Fleszar, K., Ibrahim, H. O. ve Hindi, K. S., A variable neighborhood search algorithm for the open vehicle routing problem, *European Journal of Operational Research*, 195, 803-809, 2009.
- [16] Salari, M., Toth, P. ve Tramontani, A., An ILP improvement procedure for the open vehicle routing problem, *Computers & Operations Research*, 37, 2106-2120, 2010.
- [17] MirHassani, S. A. ve Abolghasemi, N., A particle swarm optimization algorithm for open vehicle routing problem, *Expert Systems with Applications*, 38, 11547-11551, 2011.
- [18] Dror, M. ve Trudeau, P., Savings by Split Delivery Routing, *Transportation Science*, 23, 2, 141-145, 1989.
- [19] Archetti, C., Speranza, M. G. ve Hertz, A., A tabu search algorithm for the split delivery vehicle routing problem, *Transportation Science*, 40, 1, 64-73, 2006.
- [20] Jin, M., Liu, K. ve Ekşioğlu, B., A column generation approach for the split delivery vehicle routing problem, *Operations Research Letters*, 36, 265-270, 2008.
- [21] Bolduc, M. C., Laporte, G., Renaud, J. ve Boctor, F. F., A tabu search heuristic for the split delivery vehicle routing problem with production and

- demand calendars, *European Journal of Operational Research*, 202, 122-130, 2010.
- [22] Silva, M. M., Subramanian, A. ve Ochi, L. S., An iterated local search heuristic for the split delivery vehicle routing problem, *Computers & Operations Research*, 53, 234-249, 2015.
- [23] E.-G. Talbi, *Metaheuristics: From Design to Implementation*, John Wiley & Sons, 2009.
- [24] Laporte, G., The Travelling Salesman Problem: An overview of exact and approximate algorithms, *European Journal of Operational Research*, 59, 231-247, 1992.
- [25] Laporte, G., The Vehicle Routing Problem: An overview of exact and approximate algorithms, *European Journal of Operational Research*, 59, 345-358, 1992.
- [26] Kirby, D., Is your fleet the right size?, *Operational Research Quarterly*, 10, 252, 1959.
- [27] Baldacci, R., Maria, B. ve Vigo, D., *Routing a Heterogeneous Fleet of Vehicles*, *The Vehicle Routing Problem*, Cesena, Springer, 2008.
- [28] Golden, B., Assad, A., Levy, L. ve Gheysen, F., The fleet size and mix vehicle routing problem, *Computers and Operations Research*, 11, 49-66, 1984.
- [29] Schrage, L., Formulation and structure of more complex/realistic routing and scheduling problems, *Networks*, 11, 229-232, 1981.
- [30] Sariklis, D. ve Powell, S., A heuristic method for the open vehicle routing problem, *Operational Research Society*, 51, 564-573, 2000.
- [31] Taillard, E. D., Gambardella, L. M., Gendreau, M. ve Potvin, J.Y., Adaptive memory programming: A unified view of metaheuristics, *European Journal of Operational Research*, 135, 1-16, 2001.
- [32] Yousefikhoshbakht, M., Didehvar, F. ve Rahmati, F., Solving the heterogeneous fixed fleet open vehicle routing problem by a combined

- metaheuristic algorithm, *International Journal of Production Research*, 52, 9, 2565-2575, 2014.
- [33] Toth, P. ve Vigo, D., *The Vehicle Routing Problem*, Bologna, Siam, 2002.
- [34] Dror, M., Laporte, G. ve Trudeau, P., Vehicle routing with split deliveries, *Discrete Applied Mathematics*, 50, 239-254, 1994.
- [35] Düzakın, E. ve Demircioğlu, M., Araç Rotalama Problemleri ve Çözüm Yöntemleri, *Çukurova Üniversitesi İİBF Dergisi*, 13, 1, 68-87, 2009.
- [36] Gheysens, F., Golden, B. ve Assad, A., A comparison of techniques for solving the fleet size and mix vehicle routing problem, *OR Spectrum*, 6, 207-216, 1984.
- [37] Fisher, M. ve Jaikumar, R., A generalized assignment heuristic for the vehicle routing problem, *Networks*, 11, 109-124, 1981.
- [38] Miller, C. E., Tucker, A. W. ve Zemlin, R. A., Integer programming formulation of traveling salesman problems, *Journal of the ACM*, 7, 4, 326-329, 1960.
- [39] Clarke, G. ve Wright, J., Scheduling of vehicles from a central depot to a number of delivery points, *Operations Research*, 12, 568-581, 1964.
- [40] Christofides, N., Mingozzi, A. ve Toth, P., *The vehicle routing problem*, *Combinatorial Optimization*, Chichester, Wiley, 1979, 315-338.
- [41] Gillett, B. ve Miller, L., A heuristic algorithm for the vehicle dispatching problem, *Operations Research*, 22, 340-349, 1974.
- [42] Laporte, G. ve Semet, F., Classical Heuristics for the Vehicle Routing Problem, *Les Cahiers du GERAD*, 54, 1999.
- [43] Bramel, J. ve Simchi-Levi, D., A location based heuristic for general routing problems, *Operations Research*, 4, 43, 649-660, 1995.
- [44] Beasley, J., Route-first cluster-second methods for vehicle routing, *Omega*, 11, 403-408, 1983.
- [45] Haimovich, M. ve Rinnooy Kan, A., Bounds and Heuristics for capacitated routing problems, *Mathematics of Operations Research*, 10, 527-542, 1985.

- [46] Bertsimas, D. J. ve Simchi-Levi, D., A new generation of vehicle routing research: Robust algorithms addressing uncertainty, *Operations Research*, 44, 286-304, 1996.
- [47] Lin, S., Computer solutions of the travelling salesman problem, *Bell System Technical Journal*, 44, 2245-2269, 1965.
- [48] Cordeau, J. F., Laporte, G., Savelsbergh, M. W. ve D. Vigo, Vehicle Routing, *Handbook in OR & MS*, Elsevier B.V, 367-428, 2007.
- [49] Kirkpatrick, S., Gelatt, C. ve Vecchi, M., Optimization by simulated annealing, *Science*, 220, 4598, 671-680, 1983.
- [50] Holland, J. H., *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*, Ann Arbor, MIT Press Cambridge, 1992.
- [51] Beasley, D., Bull, D. ve Martin, R., An Overview of Genetic Algorithms: Part 1, Fundamentals, *University Computing*, 15, 2, 58-69, 1993.
- [52] Taillard, E. D. ve Rochat, Y., Probabilistic diversification and intensification in local search for vehicle routing, *Journal of Heuristics*, 1, 147-167, 1995.
- [53] Colorni, A., Dorigo, M. ve Vittorio, M., Distributed optimization by ant colonies, *European Conference on Artificial Life*, Amsterdam, 1991.
- [54] Dikmen, H., Dikmen, H., Elbir, A., Ekşi, Z. ve Çelik, F., Gezgin Satıcı Probleminin Karınca Kolonisi ve Genetik Algoritmalarla Eniyilemesi ve Karşılaştırılması, *Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 1, 18, 8-13, 2014.
- [55] Glover, F., Future Paths for Integer Programming and Links to Artificial Intelligence, *Computers and Operations Research*, 13, 533-549, 1986.
- [56] Willard, J., *Vehicle routing using r-optimal tabu search*, London, 1989.
- [57] Osman, İ. H., Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem, *Annals of Operations Research*, 41, 421-451, 1993.
- [58] Gendreau, M., Hertz, A. ve Laporte, G., A tabu search heuristic for the vehicle routing problem, *Management Science*, 40, 10, 1276-1290, 1994.

- [59] Gendreau, M., Hertz, A. ve Laporte, G., New Insertion and PostOptimization Procedures for The Travelling Salesman Problem, *Operations Research*, 40, 6, 1086-1094, 1992.
- [60] Taillard, E., Parallel iterative search methods for vehicle routing problems, *Networks*, 23, 661-673, 1993.
- [61] Xu, J. ve Kelly, J. P., A network flow-based tabu search heuristic for the vehicle routing problem, *Transportation Science*, 30, 379-393, 1996.