

**T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

ARAÇ ROTALAMA PROBLEMİ

FEYYAZ CELALETİN KURUL

**YÜKSEK LİSANS TEZİ
MATEMATİK MÜHENDİSLİĞİ ANABİLİM DALI
MATEMATİK MÜHENDİSLİĞİ PROGRAMI**

**DANIŞMAN
YRD. DOÇ. DR. BEYZA AHLATCIOĞLU ÖZKÖK**

İSTANBUL, 2013

T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ARAÇ ROTALAMA PROBLEMİ

Feyyaz Celalettin KURUL tarafından hazırlanan tez çalışması 20/09/2013 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Matematik Mühendisliği Anabilim Dalı'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Tez Danışmanı

Yrd. Doç. Dr. Beyza AHLATCIOĞLU ÖZKÖK

Yıldız Teknik Üniversitesi

Jüri Üyeleri

Yrd. Doç. Dr. Beyza AHLATCIOĞLU ÖZKÖK

Yıldız Teknik Üniversitesi

Doç. Dr.İbrahim EMİROĞLU

Yıldız Teknik Üniversitesi

Prof. Dr. Fatma TİRYAKİ

Yıldız Teknik Üniversitesi

ÖNSÖZ

Küreselleşen dünya ortamında şirketlerin yalnızca babadan kalma yöntemlerle yönetilemeyeceği, mevcut yapıların teknolojik araçlarla da desteklenmesi gerektiği aşîkardır. Geçmişten günümüze kadar birçok firmanın yaşadığı sıkıntılardan biri yüksek lojistik maliyetleridir. Yaklaşık elli beş yıldır üzerinde çalışılan araç rotalama problemleri firmaların karşılaştığı bu sıkıntıyı çözebilmek için kullanılmaktadır. Bu çalışmada dağıtım sistemlerindeki araç rotalama problemleri incelenmiş ve çözüm yaklaşımları gözden geçirilmiştir. Kapasite kısıtlı araç rotalama problemi için bir tamsayılı matematiksel model kullanılarak bir dizi görüşmeler sonucunda reel verileri elde edilen bir dağıtım firmasında uygulama yapılmıştır.

Yüksek lisans eğitimimde ve tez çalışmam boyunca engin fikir ve becerilerini paylaşan, bilimsel hayatıma yön veren, cesaret verici konuşmaları ve motive edici tutumuyla desteğini hiç bir zaman esirgemeyen değerli hocam ve tez danışmanım Yrd. Doç. Dr. Beyza AHLATCIOĞLU ÖZKÖK' e en içten duygularıyla teşekkür ederim.

Çalışmalarım esnasında birçok konuda desteğini esirgemeyen hocam Yrd. Doç. Dr. Hale GONCE KÖÇKEN' e teşekkür ederim.

Yüksek lisans eğitimim sürecinde tanıdığım, zamanla daha çok ısındığım ve bir ağabey gibi gördüğüm Sertan ALKAN' a mutlu ve mutsuz her anımda beni dinlediği, yönlendirdiği ve yeri geldiğinde destek verdiği için çok ama çok teşekkür ederim.

Bugünlere gelmemi sağlayan maddi ve manevi desteklerini esirgemeyen anne ve babama, görüş ve düşünceleri ile bana yardımda bulunan arkadaşlarıma teşekkür ederim.

Son olarak tezimin uygulama bölümüne verdikleri katkıdan dolayı başta Naci YILDIRIM olmak üzere tüm Yıldırım Yumurta çalışanlarına teşekkürü borç bilirim.

Ağustos, 2013

Feyyaz Celalettin KURUL

İÇİNDEKİLER

	Sayfa
SİMGE LİSTESİ.....	vii
KISALTMA LİSTESİ.....	viii
ŞEKİL LİSTESİ.....	ix
ÇİZELGE LİSTESİ	x
ÖZET	xi
ABSTRACT.....	xii
BÖLÜM 1	
GİRİŞ.....	1
1.1 Literatür Özeti	1
1.2 Tezin Amacı	4
1.3 Hipotez	5
BÖLÜM 2	
GEZGİN SATICI VE ARAÇ ROTALAMA PROBLEMİ	6
2.1 Gezgin Satıcı Problemi	6
2.2 Araç Rotalama Problemi	7
2.2.1 Araç Rotalama Problemi Temel Bileşenleri	12
2.2.2 Araç Rotalama Problemi Eniyileme Ölçütleri	13
2.2.3 Araç Rotalama Prensipleri	13
2.2.4 Araç Rotalama Problemi Uygulama Alanları	14
2.2.5 Araç Rotalama Problem Çeşitleri.....	15
2.2.5.1 Kapasite Kısıtlı Araç Rotalama Problemi (KARP)	17
2.2.5.2 Mesafe-Kapasite Kısıtlı Araç Rotalama Problemi (MKARP)	18
2.2.5.3 Geri Toplamalı Araç Rotalama Problemi (GTARP)	18
2.2.5.4 Zaman Pencereci Araç Rotalama Problemi (ZPARP).....	21
2.2.5.5 Dağıtımli Toplamalı Araç Rotalama Problemi (DTARP).....	24

2.2.5.6 Zaman Pencereli Geri Toplamalı Araç Rotalama Problemi (ZPGTARP)	27
2.2.5.7 Zaman Pencereli Dağıtımli toplamalı Araç Rotalama Problemi (ZPDTARP)	27
2.2.5.8 Diğer Araç Rotalama Problemi Türleri	27
BÖLÜM 3	
ARAÇ ROTALAMA PROBLEMİ ÇÖZÜM YAKLAŞIMLARI.....	29
3.1 Kesin Çözüm Yöntemleri	29
3.1.1 Dal ve Sınır Algoritması (Branch and Bound Algorithm)	30
3.1.2 Kesme Düzlemi Algoritması (Cutting Plane Algorithm).....	31
3.1.3 Dinamik Programlama (Dynamic Programming).....	31
3.2 Klasik Sezgisel Yöntemler	32
3.2.1 Rota Kurucu Sezgiseller (Route Building Heuristics)	32
3.2.2 İki Aşamalı Sezgiseller (Two-Phase Heuristics)	33
3.2.2.1 Süpürme Algoritması (Sweep Algorithm)	34
3.2.2.2 Fisher ve Jaikumar Algoritması (Fisher and Jaikumar Algorithm) 34	
3.2.2.3 Taç Yaprığı Algoritması (Petal Algorithm)	35
3.2.2.4 Önce Rotala Sonra Grupla Algoritması (Route-First, Cluster-Second Algorithm)	36
3.2.3 Tur Geliştirici Sezgiseller (Route Improvement Heuristics).....	36
3.4 Meta Sezgiseller Yöntemler	37
3.4.1 Tavlama Benzetimi (Simulated Annealing).....	38
3.4.2 Tabu Arama (Tabu Search)	39
3.4.3 Genetik Algoritma (Genetic Algorithm).....	40
3.4.4 Karınca Koloni Optimizasyonu (Ant Colony Optimization).....	41
BÖLÜM 4	
UYGULAMA	43
4.1 Uygulamanın Önemi	43
4.2 Uygulamanın Özellikleri	43
4.3 Uygulamada Kullanılacak Tamsayıli Lineer Matematiksel Model	44
4.4 Verilerin Toplanması ve Düzenlenmesi	46
4.5 Mevcut ve Elde Edilen Araç Rotaların Karşılaştırması	47
BÖLÜM 5	
SONUÇ VE ÖNERİLER	60
KAYNAKLAR	62
EK-A	
PAZARTESİ & CUMA GÜNLERİ UZAKLIK MATRİSİ	68

EK-B

SALI GÜNÜ UZAKLIK MATRİSİ 69

EK-C

ÇARŞAMBA & CUMARTESİ GÜNLERİ UZAKLIK MATRİSİ 71

EK-D

PERŞEMBE GÜNÜ UZAKLIK MATRİSİ 79

EK-E

CUMA GÜNÜ UZAKLIK MATRİSİ 81

ÖZGEÇMİŞ 83

SİMGE LİSTESİ

$G(V, A)$	Yönlü şebeke
$G(V, E)$	Yönsüz şebeke
V	Düğüm kümesi
A	Ark kümesi
E	Kenar kümesi
i	Düğüm noktası indisi
j	Düğüm noktası indisi
N	Düğüm sayısı
M	Araç sayısı
q_i	i düğümünün talebi
D_{ij}	i ' den j ' ye uzaklık
C	Araç kapasitesi
(i, j)	i ' den j ' ye tanımlı ark
k	Araç indisi

KISALTMA LİSTESİ

ADARP	Ayrık Dağıtımli Araç Rotalama Problemi
ADTARP	Aynı Anda Dağıtımli Toplamalı Araç Rotalama Problemi
ARP	Araç Rotalama Problemi
AUARP	Açık Uçlu Araç Rotalama Problemi
ÇDARP	Çok Depolu Araç Rotalama Problemi
DTARP	Dağıtımli Toplamalı Araç Rotalama Problemi
GA	Genetik Algoritma
GAP	Genelleştirilmiş Atama Problemi
GSP	Gezgin Satıcı Problemi
GTARP	Geri Toplamalı Araç Rotalama Problemi
KARP	Kapasite Kısıtlı Araç Rotalama Problemi
KDTARP	Karışık Dağıtımli Toplamalı Araç Rotalama Problemi
KKO	Karınca Koloni Optimizasyonu
m	metre
MARP	Mesafe Kısıtlı Araç Rotalama Problemi
MKARP	Mesafe-Kapasite Kısıtlı Araç Rotalama Problemi
PARP	Periyodik Araç Rotalama Problemi
SARP	Stokastik Araç Rotalama Problemi
TA	Tabu Arama
TB	Tavlama Benzetimi
ZPARP	Zaman Pencere li Araç Rotalama Problemi
ZPDARP	Zaman Pencere li Dağıtımli Toplamalı Araç Rotalama Problemi
ZPGTARP	Zaman Pencere li Geri Toplamalı Araç Rotalama Problemi

ŞEKİL LİSTESİ

	Sayfa
Şekil 2. 1	Alt turlar içeren gezgin satıcı problemi için uygun olmayan çözüm (Ahuja vd. [24]) 7
Şekil 2. 2	Araç rotalama problemi (Machado vd. [29])..... 8
Şekil 2. 3	ARP temel sınıfları (Toth ve Vigo [4]) 16
Şekil 2. 4	Dağıtımli toplamalı araç rotalama problem çeşitleri (Zachariadis vd. [50]) 25
Şekil 4. 1	Pazartesi&Cuma günleri mevcut rota 48
Şekil 4. 2	Pazartesi&Cuma günleri elde edilen rota..... 48
Şekil 4. 3	Salı günü mevcut rota 50
Şekil 4. 4	Salı günü İzmit bölgesi mevcut rota 50
Şekil 4. 5	Salı günü elde edilen rota 50
Şekil 4. 6	Salı günü İzmit bölgesi elde edilen rota..... 51
Şekil 4. 7	Perşembe günü mevcut rota 52
Şekil 4. 8	Perşembe günü elde edilen rota 52
Şekil 4. 9	Cuma günü mevcut rota 54
Şekil 4. 10	Cuma günü elde edilen rota 54
Şekil 4. 11	Çarşamba&Cumartesi günleri 1. aracın mevcut rotası 57
Şekil 4. 12	Çarşamba&Cumartesi günleri 1. aracın elde edilen rotası..... 58
Şekil 4. 13	Çarşamba&Cumartesi günleri 2. aracın mevcut rotası 59
Şekil 4. 14	Çarşamba&Cumartesi günleri 2. aracın elde edilen rotası..... 59

ÇİZELGE LİSTESİ

	Sayfa
Çizelge 4. 1	Pazartesi&Cuma günleri müşteri bilgileri..... 47
Çizelge 4. 2	Pazartesi&Cuma günleri rota karşılaştırması.....47
Çizelge 4. 3	Salı günü müşteri bilgileri..... 49
Çizelge 4. 4	Salı günü rota karşılaştırması.....49
Çizelge 4. 5	Perşembe günü müşteri bilgileri.....51
Çizelge 4. 6	Perşembe günü rota karşılaştırması..... 51
Çizelge 4. 7	Cuma günü müşteri bilgileri.....53
Çizelge 4. 8	Cuma günü rota karşılaştırması..... .53
Çizelge 4. 9	Çarşamba&Cumartesi günleri müşteri bilgileri.....55
Çizelge 4. 9	Çarşamba&Cumartesi günleri müşteri bilgileri (devam)..... 56
Çizelge 4. 10	Çarşamba&Cumartesi günleri rota karşılaştırması..... 56

ARAÇ ROTALAMA PROBLEMİ

Feyyaz Celalettin KURUL

Matematik Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

Tez Danışmanı: Yrd. Doç. Dr. Beyza AHLATCIOĞLU ÖZKÖK

Günümüz lojistik yönetiminde en önemli kararlardan biri dağıtım noktalarındaki talepleri karşılayacak şekilde araçların minimum maliyetler ile rotalanmasıdır. Bu amaç doğrultusunda araç rotalama problemleri üzerine uzun yıllardır birçok çalışma yapılmış ve elde edilen çözüm yöntemleri birçok şirket tarafından kullanılmaktadır. Araç rotalama problemi belli bir müşteri kümesine hizmet etmek için mevcut depo veya depolardan, bir araç filosu için optimal rotaların bulunması problemidir.

Tezimizde kısaca araç rotalama probleminin temelini oluşturan gezgin satıcı problemine değindikten sonra araç rotalama problemi, türleri ve çözüm yöntemleri ile ilgili bilgi verilmiştir. Araç rotalama problemi türlerinden biri olan kapasite kısıtlı araç rotalama problemi için tamsayılı lineer matematiksel programlama modeli kurulmuş ve GAMS 24.1.3 programında yazılmıştır. Model Yıldırım Yumurtacılıktan elde edilen güncel veriler üzerine uygulanmış ve sonuçlar şirketin mevcut rotaları ile karşılaştırılmıştır.

Anahtar Kelimeler: Araç rotalama problemi, Tamsayılı Lineer Programlama, Matematiksel Programlama

VEHICLE ROUTING PROBLEM

Feyyaz Celalettin KURUL

Department of Mathematic Engineering
MSc. Thesis

Adviser: Assist. Prof. Dr. Beyza AHLATCIOĞLU ÖZKÖK

In today's logistics management, one of the most important decisions is that vehicles are routed to meet the demands of distribution points with minimum costs. For this purpose, many studies have been done on the vehicle routing problems for many years, and the resulting solution methods used by many companies. Vehicle routing problem, in order to serve a particular customer set from available depot or depots, is a problem to finding the optimal routes for a vehicle fleet.

At our thesis, after briefly mentioned about traveling salesman problem, which form the basis of vehicle routing problem, we provided information about vehicle routing problem, its types and its solution methods. Integer linear mathematical programming model was established for the capacitated vehicle routing problem, which is one type of vehicle routing problem, and the model was written on GAMS 24.1.3 program. The model applied on real data which was obtained from Yıldırım Yumurta and the results were compared with the company's existing routes.

Keywords: Vehicle routing problem, Integer Linear Programming, Mathematical Programming

1.1 Literatür Özeti

Araç rotalama problemi (ARP) alanında teorik araştırma ve pratik uygulamalar 1959 yılında Dantzig ve Ramser [1] tarafından yayımlanan “The Truck Dispatching Problem” adlı makale ile başlamıştır. Yazarlar bu çalışmalarında benzin istasyonlarına benzin dağıtım problemi üzerinde durmuşlar ve problemin çözümü için ilk matematiksel programlama modelini kurmuşlardır (Golden vd. [2]). ARP hakkında basılan ilk kitap Eilon vd. [3] tarafından 1971 yılında yayımlanmıştır. 1971 den beri ARP ile ilgili çok sayıda kitap yayımlanmıştır. Bunlardan bazıları Crainic ve Laporte (1998), Golden ve Assad (1988) ve Toth ve Vigo [4] dir (Marikanis ve Migdalas [5]). Literatürde bibliyografya ve taksonomi sınıflandırmaları ile makaleleri inceleyen çalışmalar mevcuttur. Bazı inceleme yazıları Christofides vd. (1979), Magnanti (1981), Bodin ve Golden (1981), Lenstra ve Rinnoy Kan (1981), Bodin vd. (1983), Christofides (1985), Assad (1988), Desrochers vd. (1990), Laporte [6], Fisher (1995), Laporte ve Osman [7], Laporte vd. (2000), Cordeau vd. [8], Marinakis ve Migdalas (2002), Cordeau vd. [9], Marikanis ve Migdalas [5], Golden vd. [2], Ekşioğlu vd. [10] ve Potvin [11] dir.

Laporte ve Osman [7] yayımlamış oldukları makalede dört klasik rotalama problemi olan Gezgin Satıcı Problemi, Araç Rotalama Problemi, Çinli Postacı Problemi ve Kırsal Postacı Problemine ait 500 adet makaleyi problem tipleri, çözüm yöntemleri vb. başlıklar altında incelemişler ve alfabetik olarak sunmuşlardır.

Cordeau vd. [8] araç rotalama problemi için önemli bir kaç klasik ve modern sezgisel yöntemi özetlemişler ve doğruluk, hız, basitlik ve esneklik kriterlerini kullanarak yöntemleri karşılaştırmışlardır.

Cordeau vd. [9] makalelerinde araç rotalama problemi için son yıllarda önerilen en iyi meta sezgisellerin bazılarını incelemişlerdir. Bunlar yerel arama, popülasyon arama ve öğrenme mekanizmalarına dayanmaktadır.

Marikanis ve Migdalas [5] klasik, zaman pencereli ve diğer çeşitleri olmak üzere ARP'lerin kesin, sezgisel ve meta sezgisel yöntemler ile yapılmış çalışmaları üzerine açıklamalı bir kaynakça sunmuşlardır.

Golden vd. [2] ARP'lerin çözümleri için meta sezgiselleri kullanan çalışmaların kategorize edilmiş bir bibliyografisini sunmuşlardır. Bu bibliyografya ARP'lerin ve meta sezgisellerin birçok türünü içermektedir. Yazarlar bu çalışmalarında çalışma sayfaları, bitirme ve doktora tezlerine ulaşmak zor olduğundan yalnızca dergi ve konferanslarda yayımlanan makaleleri incelemişlerdir.

Ekşioğlu vd. [10] ARP'yi türlerine göre sınıflandırdıkları 1021 dergi makalesi, 381 bildiri, 61 teknik rapor, 5 kitap ve 26 kitap bölümünden oluşan toplam 1494 adet çalışmayı içeren bir literatür incelemesi sunmuşlardır.

Potvin [11] ARP'lerin evrimsel algoritmalarının uygulamaları üzerine bir literatür araştırması yapmıştır. Bu araştırma klasik kapasiteli ARP'ler ve çeşitlerine uygulanan genetik algoritmalar, evrim stratejileri ve parçacık sürüsü optimizasyonları üzerine çalışmalar incelemektedir.

Christofides vd. [12] q -rotaları ve minimum yayımlı k -dereceli merkezzel ağaç (shortest spanning k -degree centre tree, k -DCT) dan hesaplanan alt sınırları içeren araç rotalama probleminin kesin çözümleri için ağaç arama algoritmaları sunmuşlardır. Sayısal sonuçlar literatürden türetilen problemler için sunulmuş ve sonuçlar q -rotalarından türetilen sınırların k -DCT den türetilenlerden daha üstün olduğunu ve 25 müşteriye kadar olan araç rotalama problemlerinin tam olarak çözülebildiğini göstermişlerdir.

Laporte vd. [13] asimetrik mesafe kısıtlı araç rotalama problemi için kesin çözüm algoritması geliştirmişlerdir. Problem, alt problemlerin bazı kısıtlarmalar içeren atama problemleri ile modifiye edildiği bir dal ve kesme ağacı tarafından çözülmüştür. Düğüm sayısı 100' e kadar olan problemler için sayısal sonuçlar raporlanmıştır.

Hadjiconstantinou vd. [14] q -yolları ve minimum- k yollarının hesaplanmasını ele alan orijinal problemin iki gevşetmesinin birleşiminden elde edilen alt sınırları kullanan araç rotalama problemini çözmek için kesin bir algoritma sunmuşlardır.

Eryavuz ve Gencer [15] araç rotalama problem çözüm yöntemlerini kullanarak Balıkesir Ordudonatım Okulu personel servis araçlarının toplam güzergah mesafesini minimize etmeye çalışmışlardır. Problem çözümünde sezgisel çözüm metotlarından tasarruf (saving) ve rassal tasarruf (randomized savings) algoritmaları ve ARP 328 yazılımını kullanmışlardır. Elde edilen çözümler tur geliştirme algoritmalarıyla (2-opt ve Or-opt) daha iyi hale getirilmeye çalışılmış; elde edilen sonuçlar kendi aralarında ve mevcut durum ile karşılaştırılmıştır.

Toth ve Vigo [16] araç rotalama probleminin temel versiyonunun çözümü için önerilmiş dal ve sınır yaklaşımına dayalı kesin algoritmaları gözden geçirmişlerdir. Kriter örnekler kullanılarak farklı gevşetme ve algoritmaların performanlarını karşılaştırmışlardır.

Ralphs vd. [17] kapasite kısıtlı araç rotalama probleminin kapasite kısıtları için bir dekompozisyon tabanlı ayırma yöntemi açıklamışlardır.

Lysgaard vd. [18] kapasite kısıtlı araç rotalama problemleri için yeni bir dal ve kesme algoritması sunmuşlardır. Algoritma çeşitli kesme düzlemleri kullanmaktadır. Sayısal sonuçlar ile çok sayıda örnek için önerdikleri yeni algoritmalarının rekabetçi olduğunu göstermişlerdir.

Başkaya ve Öztürk [19] bir ekmek fabrikasının 5 satış şubesine ekmek dağıtım problemini ele almışlar ve problemi dal-kesme yöntemi ile çözmüşlerdir.

Fukasawa vd. [20] kapasite kısıtlı araç rotalama problemi için en iyi iki kesin algoritma olan dal ve kesme ile Lagrange gevşetmesi/sütun üretmenin birleşimi olan bir algoritma sunmuşlardır. Ortaya atılan dal ve kesme ve fiyat algoritması 135 düğüme kadar olan literatürdeki tüm örnekler için optimal sonuç üretmektedir.

Baldacci vd. [21] kapasite kısıtlı araç rotalama problemi için kapasite ve grup (clique) eşitsizlikleri ile ilgili ek kesmeler ile küme bölümlleme formülasyonuna dayanan yeni bir kesin algoritma sunmuşlardır.

Güvez vd. [22] gerçek bir uygulama olan Kırıkkale ili tıbbi atık toplama problemi üzerinde aracın kullanmış olduğu en uygun rotayı belirlememek ve önerilen rotanın maliyetini minimize etmek için 0-1 tamsayılı programlama modeli kullanmışlardır. Çalışma sonunda aracın izlemiş olduğu rota ile modelin önermiş olduğu rotayı karşılaştırmışlar ve elde edilen sonuçların mevcut rotadan %20.63 daha iyi olduğunu elde etmişlerdir.

1.2 Tezin Amacı

Tezimizde temeli gezgin satıcı problemine dayanan araç rotalama problemleri ve çözüm yöntemleri hakkında genel bilgi verilmiş, reel bir veri üzerinde kapasite kısıtlı araç rotalama problemi ele alınmış ve çözülmüştür.

Çalışmanın giriş bölümünden sonra yer alan Bölüm 2' de kısaca gezgin satıcı problemi anlatılmış, araç rotalama problemlerinin tanımı, genel bilgiler ve çeşitleri ayrıntılı olarak incelenmiştir. Temel sınıfları hakkında modeller sunulmuştur.

Bölüm 3 içerisinde araç rotalama problemleri çözüm teknikleri olan kesin, sezgisel ve meta sezgisel yöntemler hakkında genel bilgiler verilmiştir. Bu yöntemler içerisinde ARP için sıklıkla kullanılan dal ve sınır, kesme düzlemi, dinamik programlama kesin çözüm yöntemleri, Tasarruf, Süpürme, Fisher ve Jaikumar, Taç Yaprağı ve Önce Rotala Sonra Grupla algoritmaları olan sezgiseller ve meta sezgisellerde de Tabu Arama, Tavlama Benzetimi, Karınca Kolonisi Optimizasyonu ve Genetik Algoritma yöntemleri açıklanmıştır.

Uygulama kısmının anlatıldığı Bölüm 4 ise, problem tanımlandıktan sonra uygulama da kullanılacak kesin çözüm yöntemi modellenmiştir. Son bölümde ise sonuç ve değerlendirme yapılmıştır. Ayrıca, gelecek çalışmalara yönelik öneriler sunulmuştur.

1.3 Hipotez

"Araç Rotalama Problemi" isimli tezimizde uygulama olarak İstanbul'da faaliyet gösteren bir gıda dağıtım firmasının günlük müşteri taleplerine göre araç rotalarının belirlenmesi ve maliyetlerin minimizasyonu amaçlanmıştır. Yapılan bu çalışmada kapasite kısıtlı araç rotalama problemine ait tamsayılı lineer matematiksel model ile çözüm aranmıştır. Mevcut rotalar ile elde edilen rotalar karşılaştırılarak iyileştirme oranları incelenmiştir.

GEZGİN SATICI VE ARAÇ ROTALAMA PROBLEMİ

2.1 Gezgin Satıcı Problemi

Gezgin Satıcı Problemi (GSP) 18. yüzyılda İrlandalı matematikçi Sir William Rowan Hamilton ve İngiliz matematikçi Thomas Penygton Kirkman tarafından incelenmiştir (Matai vd. [23]). GSP, bir satıcının bulunduğu şehirden başlayarak belirli bir listedeki tüm şehirleri bir kez ziyaret ederek tekrar başladığı şehre dönmesi ve bu süreçte kat ettiği mesafenin minimizasyonu ile ilgilenir. Problem yapısında her şehrin diğerine olan mesafesinin bilindiği varsayılır. Ziyaret edilmek üzere verilmiş n adet şehir için mümkün toplam rota sayısı GSP' nin uygun çözümler kümesini oluşturur ve toplam rota sayısı $(n - 1)!/2$ ile hesaplanır. Genel olarak GSP simetrik, asimetrik ve çoklu olarak sınıflandırılabilir (Matai vd. [23]). GSP için literatürde birçok matematiksel model sunulmuştur.

$G = (V, A)$ bir şebeke olsun. Burada $V = \{1, 2, \dots, n\}$ şehirleri (düğümleri) ve A arkların kümesini belirtmektedir. $C = (c_{ij})$ i şehirden j şehrine olan uzaklık veya seyahat maliyetlerini, x_{ij} i şehirden j şehrine satıcının seyahat edip etmediğini gösteren 0-1 değişkenlerini göstermektedir. Ayrıca satıcının düğüm 1 de seyahat edebileceği $n - 1$ alternatif şehre sahip olduğu varsayılır. Kaynak düğüm keyfi olarak seçilebilir ve diğer düğümlerin her birine bir birim teslimat yapılması gereklidir (Ahuja vd. [24]).

Bu bilgiler ışığında GSP' nin en temel lineer tamsayılı matematiksel modeli aşağıdaki gibidir (Laporte [25]);

$$\text{Min} \sum_{i \neq j} c_{ij} x_{ij} \quad (2.1)$$

$$\sum_{1 \leq j \leq n} x_{ij} = 1, \quad i = 1, 2, \dots, n, \quad (2.2)$$

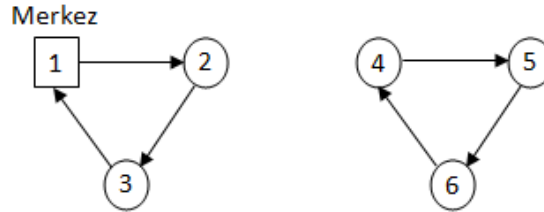
$$\sum_{1 \leq i \leq n} x_{ij} = 1, \quad j = 1, 2, \dots, n, \quad (2.3)$$

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

$$x_{ij} \in \{0,1\}, \quad i, j = 1, 2, \dots, n, \quad i \neq j. \quad (2.5)$$

Modelde amaç fonksiyonu optimal tur maliyetini tanımlamaktadır. Kısıt (2.2) her bir düğüme bir kez girilebileceğini, (2.3) her düğümden bir kez çıkılabileceğini gösteren kısıtlardır. (2.4) alt tur eleme kısıdını ifade etmektedir ve n düğümden daha az düğüm içeren turların oluşumunu engellemektedir.

Şekil 2.1, 6 adet düğüme sahip bir GSP' de (2.4) alt tur eleme kısıdı kullanılmadan (2.2) ve (2.3) kısıtları için uygun olmayan çözüm örneğini göstermektedir (Ahuja vd. [24]).



Şekil 2. 1 Alt turlar içeren gezgin satıcı problemi için uygun olmayan çözüm (Ahuja vd. [24])

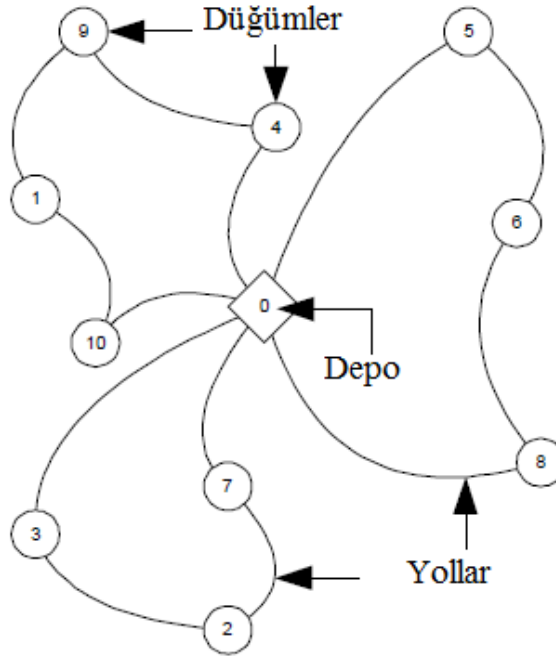
2.2 Araç Rotalama Problemi

Araç rotalama problemi (ARP) en zorlu kombinatoryal optimizasyon problemlerinden birisidir. İlk defa Dantzig ve Ramser [1] tarafından 1959 yılında formüle edilmesinden bu yana ARP yöneylem araştırması ile ilgilenen akademisyenler tarafından oldukça fazla ilgi çekmiştir. ARP' ye olan ilginin başlıca sebebi pratik olarak uygulanabilirliğidir.

ARP gezgin satıcı problemine (GSP) benzerliği ile bilinmektedir. GSP de amaç bir düğümden başlayarak ve tekrar aynı düğüme geri dönerek, tüm müşterileri (düğümleri) bir kez ziyaret eden bir satıcı için en kısa seyahati bulmaktır. ARP, GSP' nin tek tip ve sınırlı kapasiteye sahip birden fazla aracın kullanıldığı ve bazı eklenmiş kısıtlar ile geliştirilmiş haline verilen isimdir (Floudas ve Pardalos [26]).

Araç rotalama problemindeki esas amaç bir araç filosu için minimum maliyetli rota kümesi oluşturmaktır. Her bir rota tek bir depoda başlar, talepleri bilinen bir müşteri kümesine hizmet verildikten sonra yine aynı noktada son bulur. Her müşteriye sadece bir araç tahsis edilmeli ve bir araç ile ilişkilendirilen müşterilerin toplam talebi o aracın kapasitesini geçmemelidir. ARP de kullanılan herhangi bir aracın kapasitesinin tüm müşteri taleplerinin en büyüğünden daha büyük olduğu varsayılır, böylece her bir araca bir veya birden fazla müşterinin atanması sağlanır (Balakrishnan [27]).

Diğer bir ifade ile araç rotalama problemi bir dizi yan kısıtlar eşliğinde dağınık coğrafi yapıya sahip şehir veya müşteriler için bir veya birden çok depoda bulunan araç filosuna ait rotaların belirlenmesini içeren problem sınıflarının tümüne verilen genel bir addır (Rego [28]). Şekil 2.2' de ARP modelinin grafiksel olarak görülebileceği bir örnek sunulmuştur ve düğümler müşterileri ifade etmektedir (Machado vd. [29]).



Şekil 2. 2 Araç rotalama problemi (Machado vd. [29])

Klasik araç rotalama probleminde mevcut depolarda bulunan araçlar ile farklı noktalardaki müşterilerin talepleri karşılanmaya çalışılır. ARP de amaç minimum zaman, yol ve maliyet ile müşteri taleplerini karşılayabilecek rotayı belirlemektir (Özkan [30]). Araç rotalama üzerine literatürde bulunan birçok çalışma aşağıdaki genel varsayımlara sahiptir (Caccetta ve Hill [31]).

1. Tek tip ticari bir mal, talepleri önceden bilinen müşterilere, tek bir depodan dağıtılır.
2. Her müşteri talebi yalnızca bir araç tarafından karşılanır.
3. Tüm araçlar aynı kapasiteye sahiptir ve tek tur yaparlar.
4. Her bir araç tarafından kat edilen mesafe belirli bir sınırı aşamaz.
5. Her müşteri belirli bir zaman aralığında ziyaret edilmelidir.
6. Amaç mevcut araçlar tarafından gidilen toplam yolu minimize etmektir.

4 ve 5 kısıtları göz ardı edildiğinde kapasite kısıtlı araç rotalama problemi olarak adlandırılan en çok çalışılan ARP ortaya çıkmaktadır.

Problem boyutu (düğüm sayısı) ile birlikte problem çözümü için gerekli olan işlem adımlarının üstsöl olarak arttığı NP-zor (Non-deterministic Polynomial-time hard) problem sınıfına giren ARP, yaygın olarak ele alınan bir tamsayılı programlama problemidir (Şeker [32]). Günümüze kadar tüm ARP uygulamalarında optimal sonucu elde edebilecek tek bir kesin çözüm yolu bulunmamaktadır. Karşılaşılabilecek problemlerde sonuca ulaşabilmek için kesin çözümlerin yanı sıra yaklaşık çözümlerin elde edilebileceği sezgisel ve meta sezgisel yöntemler geliştirilmiştir.

Klasik araç rotalama probleminde müşteriler, her bir müşteri için gerekli olan servis zamanı ve müşteriler arası seyahat süresi önceden bilinmektedir. Klasik ARP aşağıdaki gibi tanımlanabilir (Yeun vd. [33], Laporte [6]):

$G = (V, A)$ bir şebeke olsun. 1 düğümü depoyu geri kalanlar müşterileri temsil etmek üzere $V = \{1, \dots, n\}$ düğüm kümesidir ve $A = \{(i, j) : i, j \in V, i \neq j\}$ ark kümesini ifade eder. Kimi zaman depo $n + 1$ ile de ilişkilendirilebilir (Toth ve Vigo [4]). Negatif olmayan maliyet/uzaklık matrisi c_{ij} veya seyahat zaman matrisi t_{ij} her bir $(i, j) \in A$

arkı ile ilişkilendirilir (Cordeau vd. [8]). Genel olarak (i, i) arkına izin verilmez ve $c_{ii} = +\infty$ değerini alır. $\forall (i, j) \in A$ için $c_{ij} = c_{ji}$ olduğunda problem simetrik ARP olarak, aksi durumda asimetrik ARP olarak isimlendirilir (Toth ve Vigo [4]). Simetrik ARP yönsüz $G = (V, E)$ şebekesi üzerinde tanımlıdır ve $E = \{(i, j): i, j \in V, i < j\}$ bir kenar kümesidir (Cordeau vd. [8]). Birçok pratik uygulama da maliyet matrisi üçgen eşitsizliğini sağlamaktadır. Yani $\forall i, j, p \in V$ için $c_{ip} + c_{pj} \geq c_{ij}$ dir (Toth ve Vigo [4]). Ayrıca, depoda mevcut k adet aracın bulundurulduğu varsayılır ve $k_l \leq k \leq k_u$ sınır koşullarına sahiptir. $k_l = k_u$ olduğunda k nın sabit olduğu, $k_l = 1$, $k_u = n - 1$ olduğunda ise k nın serbest olduğu söylenir. Genellikle k sabit alınmadığında, araç kullanımını sabit bir f maliyeti ile ilişkilendirmek mantıklıdır (Yeun vd. [33], Laporte [6]). Her müşteri negatif olmayan d_i talebine ve her bir araç C kapasitesine sahiptir (Cordeau vd. [8]).

Klasik ARP minimum maliyetli k adet rotayı bulmayla ilgilenir ve rotaların maliyetleri rotaların içerdiği arkların maliyetleri toplamı ile ifade edilir. Şöyle ki (Toth ve Vigo [4]);

- her bir rota depoyu içermelidir;
- her bir müşteri (düğüm) $(i \in V \setminus \{1\})$ kesin olarak bir rota içerisinde ziyaret edilmelidir;
- bir rota içerisinde ziyaret edilen düğümlerin talepleri toplamı rotayı kullanan aracın kapasitesi C 'yi geçmemelidir.

Klasik ARP için kurulacak modelde amaç ürün dağıtımı (toplamı) için müşteri noktalarından geçen minimum maliyetli rotaların kümesini bulmaktır. $x_{ij}^k (i, j)$ arkı üzerinde i şehirden j şehrine k aracının sevkiyat yapıp yapmadığını gösteren 0-1 değişkenidir. $y_{ij} (i, j)$ arkı üzerinde araç seyahatlerinin olup olmadığını belirtir. Model (Ahuja vd. [24]);

$$\text{Min} \sum_{1 \leq k \leq K} \sum_{(i,j) \in A} c_{ij} x_{ij}^k \quad (2.6)$$

$$\sum_{1 \leq k \leq K} x_{ij}^k = y_{ij}, \quad \forall i, j \in V, \quad (2.7)$$

$$\sum_{1 \leq j \leq n} y_{ij} = 1, \quad \forall i \in V \setminus \{1\}, \quad (2.8)$$

$$\sum_{1 \leq i \leq n} y_{ij} = 1, \quad \forall j \in V \setminus \{1\}, \quad (2.9)$$

$$\sum_{1 \leq j \leq n} y_{1j} = K, \quad (2.10)$$

$$\sum_{1 \leq i \leq n} y_{i1} = K, \quad (2.11)$$

$$\sum_{2 \leq i \leq n} \sum_{1 \leq j \leq n} d_i x_{ij}^k \leq C, \quad k = 1, 2, \dots, n, \quad (2.12)$$

$$\sum_{i \in Q} \sum_{j \in Q} y_{ij} \leq |Q| - 1, \quad \forall Q \subset V \setminus \{1\}, \quad (2.13)$$

$$y_{ij} \in \{0, 1\}, \quad \forall (i, j) \in A, \quad (2.14)$$

$$x_{ij}^k \in \{0, 1\}, \quad \forall (i, j) \in A \text{ ve } k = 1, 2, \dots, K. \quad (2.15)$$

$A' = \{(i, j): y_{ij} = 1\}$ olarak tanımlansın. (2.6) amaç fonksiyonu araçların geçtiği tüm arkların mesafeleri toplamını göstermektedir. Her arktan sadece bir aracın geçmesini sağlamak için (2.7) kısıdı oluşturulmuştur. (2.8) ve (2.9) kısıtları ile V nin her bir düğümüne sadece bir giriş ve çıkış arkının kullanılması sağlanmaktadır. (2.10) ve (2.11) kısıtlarının oluşturulma sebebi $\{1\}$ düğüm noktası olan depoya giren ve çıkan araç sayısının belirlenmesidir. Kısıt (2.12) ile her bir araç kapasitesinin geçeceği düğümlerdeki taleplerin toplamına eşit olması veya toplamlarından büyük olması sağlanmaktadır. Kısıt (2.13) çözümün $\{2, 3, \dots, n\}$ düğümlerini kullanan hiç bir alt turu içermemesini sağlamaktadır (diğer bir deyişle bu düğümler herhangi bir alt tur içermez). Aksi takdirde A' arkları Q kümesine dahil olan bazı turlar içerecek ve çözüm (2.13) kısıdını ihlal edeceklerdir. Çünkü (2.13) kısıdının sol tarafı en az $|Q|$ olmalıdır. Bu sebeple (2.13) kısıtları alt tur kırma kısıtlarıdır (Ahuja vd. [24]).

2.2.1 Araç Rotalama Problemi Temel Bileşenleri

ARP temel bileşenleri talep yapısı, malzeme tipi, dağıtım/toplama noktaları ve araç filosu olmak üzere dört başlık altında incelenmektedir (Çalışkan [34]).

Talep Yapısı: ARP' de statik ve dinamik olmak üzere iki tür talep mevcuttur. Statik talep, tur öncesi taleplerin bilindiği durumdur. Dinamik talep ise bazı düğümlerdeki taleplerin tur öncesinde bilindiği bazılarının tur esnasında öğrenildiği talep tipidir.

Malzeme Tipi: ARP' de malzeme tipi göz önüne alındığında birçok çeşit malzeme ile karşılaşmaktadır. Taşınan malzeme, türüne göre basit paketler veya karma paketler olarak düşünülebilir. Gıda maddeleri, gazete dağıtım, çöp toplama vs. basit paketler sınıfına girmektedir. Öğrenci servisleri güvenlik, etkinlik, eşitlik gibi ilave bazı amaçlardan dolayı karma yapıli malzeme türüne örnek olarak verilebilmektedir. Ayrıca tehlikeli madde vb. taşıyan araçların rotalarının belirlenmesinde coğrafi özellikler büyük önem teşkil etmektedir.

Dağıtım/Toplama Noktaları: ARP' de genellikle depo toplama noktası, müşterilerde dağıtım noktası olarak bilinmektedir. Örnek olarak tüketim malzemelerinin fabrikalardan toptancılara dağıtımını düşünülebilir. Depo genellikle rotanın başladığı ve bittiği noktadır. Problem depo adedine göre tek depolu ve çok depolu olarak ikiye ayrılmaktadır. Çok depolu problemlerde araçlar çıktıkları depolara geri dönüyor ise depolar birbirinden bağımsız tek depolu ARP olarak ele alınabilmektedir. Aracın bir depodan çıkıp başka bir depoya vardığı problemlerde depolar bir bütün olarak ele alınmaktadır. Servis araç rotası belirleme problemleri gibi bazı ARP' ler de dağıtım ve toplama noktaları aynıdır.

Araç Filosu: ARP' ler homojen ve heterojen olmak üzere iki farklı tür filoya sahip olabilir. ARP' ler de çoğunlukla kapasiteleri bilinen ve homojen (aynı kapasitede) yapıda araçlar kullanılmaktadır. Heterojen filo yapısında ise kullanılan araçlar farklı kapasitelere sahiptirler. Heterojen araçlara sahip problemlerde hangi aracın hangi rotaya hizmet vereceğini belirlemek için ilave kısıtlar gerekmektedir. Araçların kapasiteleri dışında hız, yakıt tüketimi, malzemeye uygunluğunu gibi özellikleri de mevcuttur. Bu özelliklerin rotalama kararlarına doğrudan etkisi yoktur.

2.2.2 Araç Rotalama Problemi Eniyileme Ölçütleri

Her ne kadar klasik araç rotalama probleminde amaç fonksiyonu toplam rota uzunluğunun minimizasyonu olarak tanımlanmış olsa da literatürdeki matematiksel modellerde amaç fonksiyonu için farklı optimizasyon ölçütleri de tanımlanmıştır. Bunlardan en yaygın olanları;

Rota Sayısı: Her bir rota içerisinde bir aracın kullanıldığı düşünüldüğünde rota sayısı ile birlikte araç sayısının da azaltılabileceği böylelikle yatırım maliyetlerinin minimize edilebileceği görülmektedir.

Yük Dengesi: Her rota boyunca her araç tarafından taşınan ürün miktarındaki dengeyi ifade etmektedir.

Uzunluk Dengesi: Rota uzunluklarındaki çeşitliliğin azaltılmasıdır (Bowerman vd. [35]).

Rota Süresi: Genel olarak rota boyunca seyahat, yükleme/boşaltma ve bekleme sürelerinin toplamını ifade etmektedir.

Müşteri Memnuniyeti: Ürünlerin müşterilere zamanında teslimatı ile ilgili ölçüttür. Bazı durumlarda parçalı (eksik) teslimattan doğan cezaların en aza indirgenmesidir (Tüfekçier [36]).

2.2.3 Araç Rotalama Prensipleri

Literatürde birçok araç rotalama problemi ve çözümleri için geliştirilmiş birçok kesin ve sezgisel algoritmalar bulunmaktadır. Ortaya atılan algoritmaların bir çoğu reel uygulamalarda en optimal sonucu vermemektedir. Bu sebeple araştırmacılar ve akademisyenler en etkin ve yararlı sonucu verecek algoritmalar üzerine çalışmalara devam etmektedir.

Araştırmacılar tarafından araştırmalar ve uygulamalar esnasında daha başarılı uygulanabilir rotaların oluşturulabilmesi için sekiz ilke göz önünde bulundurulmaktadır:

1. Toplam gidilen mesafenin kısılması için noktalar arası en kısa olanlar seçilmelidir.

2. Benzer noktalardaki dağıtımlar birleştirilerek farklı günlerdeki dağıtımlar engellenmelidir. Böylelikle yakın tarihler içerisinde aynı noktaya tekrar gidilmesinin önüne geçilmiş olunmaktadır.
3. Rota başlangıç noktası olarak depodan mümkün en uzak nokta seçilmelidir.
4. Yapılacak olan rotalamaların şekli, gözyaşı şeklinde olmalıdır. Bu sayede uzak noktalara ulaşımında kazanç elde edilmiş olur.
5. Toplam maliyeti azaltmak ve avantaj elde edebilmek için en yüksek kapasiteye sahip araçlar seçilmelidir.
6. Mümkün olabildiğince dağıtım ve toplama işlemleri aynı araç ile yapılmalıdır. Böylelikle toplam maliyet ve zamandan tasarruf edilir.
7. Mevcut rota dışı noktalara ulaşımında küçük araçlar kullanılmalıdır.
8. Gerekli durumlarda dağıtım ve toplama zamanları tekrar kararlaştırılarak zaman tasarrufuna gidilmelidir (Şeker [32]).

2.2.4 Araç Rotalama Problemi Uygulama Alanları

Genel olarak ARP bir şebeke içerisindeki belirli noktalar arasında mal ve hizmet dağıtımı ile ilgilenmektedir. Dolayısıyla gerçek hayat problemlerinin birçoğunda uygulama alanı mevcuttur. Bunlardan bazıları (Ekşioğlu vd. [10], Düzakın ve Demircioğlu [37]);

- Ürün ve hizmetlerin bir veya daha fazla sayıdaki depodan, çeşitli müşteri yerlerine dağıtımı,
- Üretim planlaması ve hammadde, yarı mamul ve mamullerin fabrikalar arası taşınması,
- Restoran ve kafelere içecek dağıtımı,
- ATM makinelerine para dağıtımı ve zamanlaması,
- Benzin ve mazot dağıtımı,
- Restoranlardan atık yağların toplanması,

- Ev aletleri tamir servisi ve teslimatı,
- İnternet tabanlı ev şirketlerinin ürün dağıtımı,
- Süt toplama ve stok yönetimi,
- Evlerden hayır bağış toplama,
- Taşınabilir tuvalet teslimatı,
- Hapishane ve mahkemeler arası mahkum taşıma,
- Tıbbi ofislerden laboratuarlara idrar örneklerinin taşınması,
- Çöp toplama ve aktarma,
- Depolardan perakendecilere toptan dağıtım,
- Posta teslimat kamyonunun yönlendirilmesi,
- Havayolu şirketleri ile yolcu ve ürün taşınması,
- DVD film kiralama.

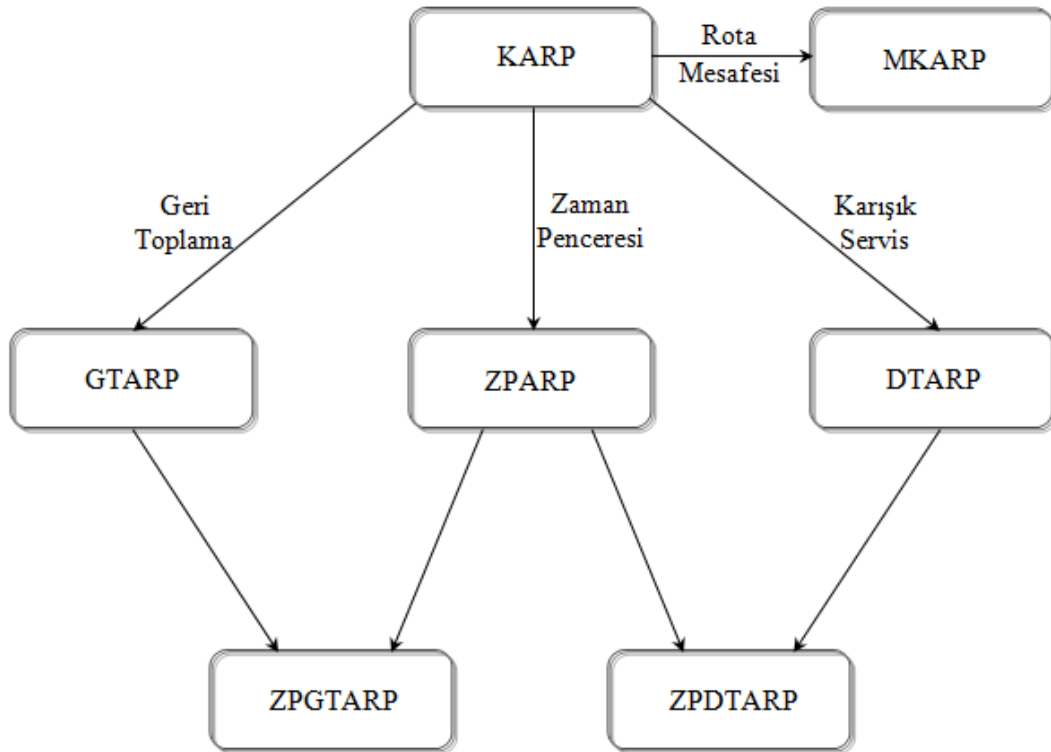
2.2.5 Araç Rotalama Problem Çeşitleri

Araç rotalama problemleri coğrafik olarak dağılmış müşteri kümesine bir veya daha fazla merkezi depodan dağıtım veya toplama rotalarının tasarımını içerir (Crainic ve Laporte [38]). ARP ile ilgili karmaşık gerçek hayat uygulamalarında çözüm bulmak çok zor veya imkansız olduğundan akademik çalışmalar genellikle önemli kısıtları içeren basitleştirilmiş problem tipleri ile ilgili olmaktadır. Günümüze kadar birçok ARP üzerinde incelemeler yapılmış ve farklı yaklaşımlar ele alınmıştır (Yurtkuran [39]). ARP ile ilgili incelemeler arttıkça faktör, kısıt ve amaçlara bağlı olarak farklı türde problemler ortaya çıkmaktadır (Crainic ve Laporte [38]).

Problem toplama, dağıtım veya her ikisinin birleşimini mi içermektedir? Toplama ve dağıtım arasında bir öncelik var mıdır? Dağıtım tek bir depodan mı yoksa birden çok depodan mı yapılmaktadır? Kaç adet araç mevcuttur? Sayı sabit midir veya bir karar değişkenini mi temsil eder? Araç filosu homojen veya heterojen midir? Araçların kapasite, hız ve işletme maliyeti nedir? Sürücü çalışma koşulları nelerdir? Ödeme

yapısı, normal iş günü uzunluğu ve fazla mesai koşulları nedir? Gün içerisinde birden fazla tura izin verilmekte midir? Talep statik veya dinamik bir yapıya mı sahiptir? Her bir müşteri planlama döneminde ne sıklıkla veya ne zaman gezilmelidir? Belirli bir günde müşterilerin belirli bir zaman penceresi içerisinde gezilmesi gerekli midir? tarzı bazı sorular ile ARP türleri belirlenmeye çalışılır (Crainic ve Laporte [38]).

Temel ARP yapıları Kapasite Kısıtlı Araç Rotalama Problemi (Capacitated Vehicle Routing Problem-KARP), Mesafe-Kapasite Kısıtlı Araç Rotalama Problemi (Distance-Constrained Capacitated Vehicle Routing Problem-MKARP), Geri Toplamalı Araç Rotalama Problemi (Vehicle Routing Problem with Backhaul-GTARP), Zaman Pencere Araç Rotalama Problemi (Vehicle Routing Problem with Time Window-ZPARP), Dağıtım Toplamalı Araç Rotalama Problemi (Vehicle Routing Problem with Pickup and Delivery-DTARP), Zaman Pencere Geri Toplamalı Araç Rotalama Problemi (Vehicle Routing Problem with Backhaul and Time Window-ZPGTARP), Zaman Pencere Dağıtım Toplamalı Araç Rotalama Problemi (Vehicle Routing Problem with Pickup and Delivery and Time Window-ZPDTARP) Şekil 2.3' de belirtilmiştir.



Şekil 2. 3 ARP temel sınıfları (Toth ve Vigo [4])

2.2.5.1 Kapasite Kısıtlı Araç Rotalama Problemi (KARP)

Kapasite kısıtlı araç rotalama problemi (KARP) ARP' nin en temel modelidir. KARP' de talepler deterministik, bölünmemiş ve önceden bilinmektedir. Tüm müşteri talepleri karşılanmakta, araçlar özdeş ve tek bir depoda yer almaktadır. KARP' de amaç tüm müşterilere hizmet vermek için gerekli toplam maliyeti (rota sayısı ve/veya rotaların uzunluk veya seyahat süresi) minimize etmektir (Augerat vd. [40], Toth ve Vigo [16]).

KARP yönlü bir şebeke $G = (V, A)$ üzerinde tanımlansın. Burada $V = \{0, 1, \dots, n\}$ düğüm kümesini, $\forall i \in V \setminus \{0\}$ düğümü negatif olmayan d_i talepli müşterileri ve $\{0\}$ düğümü de depoyu ifade etmektedir. $A = \{(i, j): i, j \in V, i \neq j\}$ arkları seyahat maliyeti c_{ij} ile ilişkilendirilir. Depoda C kapasitesine sahip özdeş K adet araçtan oluşan sabit bir filo mevcuttur. KARP toplam seyahat maliyeti minimize edilmiş K rotalarının kümesinin belirlenmesi olarak ifade edilir. x_{ijk} ($k = 1, \dots, K$) (i, j) arkı üzerinde i şehrinden j şehrine k aracının sevkiyat yapıp yapmadığını gösteren 0-1 değişkenidir. y_{ik} k aracı tarafından i müşterisine servis yapıp yapılmadığını gösterir.

Model:

$$\text{Min} \sum_{i \in V} \sum_{j \in V} c_{ij} \sum_{k=1}^K x_{ijk} \quad (2.16)$$

$$\sum_{k=1}^K y_{ik} = 1, \quad \forall i \in V \setminus \{0\}, \quad (2.17)$$

$$\sum_{k=1}^K y_{0k} = K, \quad (2.18)$$

$$\sum_{j \in V} x_{ijk} = \sum_{j \in V} x_{jik} = y_{ik}, \quad \forall i \in V, \quad k = 1, \dots, K, \quad (2.19)$$

$$\sum_{i \in V} d_i y_{ik} \leq C, \quad \forall k = 1, \dots, K, \quad (2.20)$$

$$\sum_{i \in S} \sum_{j \in S} x_{ijk} \leq |S| - 1, \quad \forall S \subseteq V \setminus \{0\}, \quad |S| \geq 2, \quad k = 1, \dots, K, \quad (2.21)$$

$$y_{ik} \in \{0,1\} \quad \forall i \in V, \quad k = 1, \dots, K, \quad (2.22)$$

$$x_{ijk} \in \{0,1\} \quad \forall i, j \in V, \quad k = 1, \dots, K. \quad (2.23)$$

(2.17)-(2.19) kısıtları sırasıyla her bir müşterinin bir kez ziyaret edileceğini, depodan K adet aracın ayrılacağını ve düğümlerin giren çıkan dengesini gösterir. (2.20) kısıdı her bir k aracı için kapasite kısıdı iken (2.21) kısıdı alttır eleme kısıdıdır (Toth ve Vigo [4]).

2.2.5.2 Mesafe-Kapasite Kısıtlı Araç Rotalama Problemi (MKARP)

Mesafe kısıtlı araç rotalama problemi (MARP) her bir rota için kapasite kısıdı yerine maksimum uzunluk (veya zaman) kısıdının kullanıldığı ARP türüdür. MARP' de t_{ij} (veya t_e) $\forall (i, j) \in A$ arkında (veya $e \in E$ kenarında) negatif olmayan uzunluğu ifade etmektedir. Her bir rotanın toplam ark uzunluğu maksimum rota uzunluğu L 'yi geçmemelidir. Eğer araçlar farklı ise maksimum rota uzunlukları T_k , $k = 1, \dots, K$ ile gösterilmektedir. Ark uzunlukları seyahat zamanlarını ifade ederken her bir i müşterisi için s_i , i düğümünde aracın servis için harcadığı zamanı göstermektedir. Alternatif olarak servis zamanları arkların seyahat zamanlarına eklenebilir. $\forall (i, j)$ arki için $t_{ij} = t'_{ij} + s_i/2 + s_j/2$ dir. Burada t'_{ij} (i, j) arkının orijinal seyahat zamanıdır.

Genel olarak maliyet ve uzunluk matrisleri birbirleri ile çakışmaktadır. Yani $\forall (i, j) \in A$ arki için $c_{ij} = t_{ij}$ (veya $\forall e \in E$ için $c_e = t_e$) dir. Arkların seyahat sürelerine servis zamanları eklendiğinde problemin amacı toplam rotanın veya rota sürelerinin minimize edilmesidir.

Araç kapasitelerinin ve maksimum mesafe kısıtlarının birlikte bulunduğu ARP' ler mesafe-kapasite kısıtlı araç rotalama problemi olarak adlandırılır (Toth ve Vigo [4]).

2.2.5.3 Geri Toplamalı Araç Rotalama Problemi (GTARP)

GTARP hem dağıtım hem toplama noktalarını içeren araç rotalama problemidir. GTARP sırasıyla müşterilerin dağıtım ve toplama taleplerini karşılayacak optimal rotaları bulmayı amaçlamaktadır. GTARP modellenirken aşağıdaki senaryo takip edilir. İlk olarak her bir araç depodan yola çıkar ve dağıtım talepleri karşılanır. Araç yükü tamamen boşaltıldıktan sonra depoya geri teslim edilecekleri toplamak üzere araç

müşterilere toplama işlemi için hizmet vermektedir (Zachariadis ve Kiranoudis [41]). GTARP ile ilgili ilk tamsayı lineer programlama modeli Goetschalckx ve Jacobs-Blecha (1989) tarafından sunulmuştur (Toth ve Vigo [4]).

GTARP tam yönsüz $G = (V, A)$ şebekesi üzerinde tanımlı, $V = \{0\} \cup L \cup B$ düğüm kümesi ve $A = \{(i, j) : i, j \in V\}$ ark kümesi olsun. $\forall i \in V$ için $c_{ii} = +\infty$ ve $i \neq j$ iken $\forall i, j \in V$ için $c_{ij} = c_{ji}$ olacak şekilde $\forall (i, j)$ arkı negatif olmayan bir c_{ij} maliyeti (uzaklığı) ile ilişkilendirilir. Alt kümeler $L = \{1, 2, \dots, n\}$ ve $B = \{n + 1, n + 2, \dots, n + m\}$ sırasıyla dağıtım ve toplama yapılacak müşterileri, 0 düğümü de depoyu ifade eder. Toplam müşteri sayısı N ile gösterilir ve depoda Q kapasitesine sahip K adet özdeş araç mevcuttur. $\forall i \in L \cup B$ müşterisi dağıtılacak ($i \in L$) veya toplanacak ($i \in B$) belirli bir q_i talebine sahiptir. Araç sayısı $K \geq \max\{K_L, K_B\}$ ile ifade edilir. K_L ve K_B sırasıyla dağıtım ve toplama yapılacak müşterilere hizmet etmek için gerekli en az sayıdaki araç sayısıdır (Brandao [42]).

GTARP de en az maliyetli K araç rotasını bulmak için aşağıdaki kısıtlar ele alınır (Toth ve Vigo [4]):

- Her tur depoyu ziyaret etmelidir,
- $\forall j \in V \setminus \{0\}$ düğümleri tam olarak bir kez ziyaret edilmelidir,
- bir rotaya ait dağıtım ve toplama düğümlerindeki talep ayrı olarak araç kapasitesi Q 'u geçmemelidir,
- aksi söylenmedikçe tüm rotalarda dağıtım düğümleri toplama düğümlerine göre önceliklidir.

Farklı türde (heterojen) araç filosuna sahip GTARP için aşağıdaki karma tamsayılı modeli verilmiştir (Tavakkoli-Moghaddam vd. [43]).

C_{ijk} : k aracının i düğümünden j düğümüne gitmesi için gerekli maliyet

n : dağıtım düğümleri sayısı

m : toplama düğümleri sayısı

M : araç sayısı

Q_k : k aracının kapasitesi

f_i : i düğümündeki dağıtım talebi

d_i : i düğümündeki toplama talebi

x_{ijk} : k aracının i düğümünden j düğümüne hareket edip etmediğini belirten $\{0,1\}$ değişkeni (gidiyorsa değeri 1, aksi takdirde 0)

$$\text{Min } z = \sum_{i=0,\dots,n+m} \sum_{j=0,\dots,n+m} \sum_{k=1,\dots,M} C_{ijk} x_{ijk} \quad (2.24)$$

$$\sum_{j=0,\dots,n+m} \sum_{k=1,\dots,M} x_{ijk} = 1, \quad i = 1, \dots, n+m, \quad i \neq j, \quad (2.25)$$

$$\sum_{i=0,\dots,n+m} \sum_{k=1,\dots,M} x_{ijk} = 1, \quad j = 1, \dots, n+m, \quad j \neq i, \quad (2.26)$$

$$\sum_{j=0,\dots,n+m} \sum_{k=1,\dots,M} x_{0jk} = M, \quad (2.27)$$

$$\sum_{i=0,\dots,n+m} \sum_{k=1,\dots,M} x_{i0k} = M, \quad (2.28)$$

$$\sum_{i=1,\dots,n} \sum_{j=1,\dots,n+m} d_i x_{ijk} \leq Q_k, \quad k = 1, \dots, M, \quad j = 0, 1, \dots, n+m, \quad (2.29)$$

$$\sum_{i=n+1,\dots,n+m} \sum_{j=1,\dots,n+m} f_i x_{ijk} \leq Q_k, \quad k = 1, \dots, M, \quad j = 0, 1, \dots, n+m, \quad (2.30)$$

$$\sum_{i=0,1,\dots,n+m} x_{ijk} - \sum_{l=0,1,\dots,n+m} x_{jlk} = 0, \quad j = 1, \dots, n+m, \quad k = 1, \dots, M, \quad (2.31)$$

$$\sum_{j=0,1,\dots,n+m} x_{ijk} - \sum_{l=0,1,\dots,n+m} x_{lik} = 0, \quad i = 0, 1, \dots, n+m, \quad k = 1, \dots, M, \quad (2.32)$$

$$\sum x_{ijk} \leq |S| - 1, \quad S \subseteq \{2, 3, \dots, n+m\}, \quad (2.33)$$

$$\sum_{i=n+1,\dots,n+m} \sum_{j=0,\dots,n} \sum_{k=1,\dots,M} x_{ijk} = 0, \quad (2.34)$$

$$x_{ijk} = \{0,1\}, \quad i = 0,1, \dots, n + m, \quad k = 1, \dots, M. \quad (2.35)$$

Sunulan modelde amaç fonksiyonu (2.24) ile seyahat maliyetini minimize etmek amaçlanmıştır. (2.25) ve (2.26) kısıtları depo hariç tüm düğümlerden sadece bir rotanın geçmesini sağlamaktadır. (2.27) ve (2.28) kısıtları depoya giren ve depodan çıkan araç sayısının M olmasını sağlamaktadır. (2.29) ve (2.30) kısıtları sırasıyla toplama ve dağıtım yapılan noktalardaki taleplerin araç kapasitesi Q_k' 'ı geçemeyeceğini göstermektedir. Rotanın devamlılığı (2.31) ve (2.32) kısıtları tarafından sağlanmaktadır. Bir araç sadece bir rotaya hizmet etmektedir. Alt tur oluşumunu engellemek için (2.33) kısıdı yazılmıştır. Dağıtım düğümlerine toplama düğümlerinden önce uğranılması (2.34) kısıdı ile gerçekleştirilmiştir.

2.2.5.4 Zaman Pencere Arac Rotalama Problemi (ZPARP)

$\forall i \in V$ müşterisindeki hizmetin verilen belirli bir $[a_i, b_i]$ zaman aralığında gerçekleştiği zaman pencere arac rotalama problemi klasik ARP' nin önemli bir çeşididir (Cordeau vd. [44]). Zaman pencere kısıdı kesin (hard) ve yumuşak (soft) olmak üzere ikiye ayrılır. Kesin zaman pencere ARP' de bir araç a_i zamanından önce belirlenen noktaya varabilir ve bekleyebilir fakat araç belirlenen b_i vaktinden sonra müşteriye varırsa teslimata izin verilmemektedir. Yumuşak zaman pencere ARP' de kesin zaman pencereliden farklı olarak araç b_i vaktinden sonra müşteriye vardığında amaç içerisinde bir cezaya sebep olmaktadır (Dondo ve Cerda [45]). ZPARP' nin dağıtım yönetiminde birçok uygulaması mevcuttur. İçecek ve gıda dağıtımı, gazete dağıtımı, ticari ve endüstriyel atık toplama yaygın örneklerdir (Cordeau vd. [44]).

Tek araçlı bozulabilir gıda ürünleri için kurulmuş ZPARP matematiksel olarak aşağıdaki şekilde ifade edilebilir (Azi vd. [46]).

A ark kümesine sahip tam yönlü bir şebeke içerisinde $N = \{1,2, \dots, n\}$ düğümler kümesine depodan bozulabilir gıda ürünleri taşıyan Q kapasitesine sahip tek bir araca sahip olunsun. Her bir $(i,j) \in A$ arki ile uzaklık d_{ij} ve seyahat zamanı t_{ij} ilişkilendirilsin. $i \in N$ müşterileri q_i talebi, servis (veya bekleme) zamanı s_i ve $[a_i, b_i]$ (a_i servise başlamak için en erken zamanı, b_i en geç zamanı gösterir) zaman penceresi ile

nitelendirilsin. Bu sebeple araç i müşterisine a_i zamanından önce varırsa beklemelidir. Araç bir günde $K = \{1, \dots, k\}$ rotaya sahiptir. Burada her bir rota depoda başlar ve depoda son bulur. Bu rotaların bazıları boş olabilir. Genel kayıp olmadan rotalar $1, 2, \dots, k$ sırasıyla hizmet verir. Depo 0 veya $n + 1$ ile gösterilir. Depo bir arkın başlangıç ve bitiş düğümü ise $s_0 = s_{n+1} = 0$, $q_0 = q_{n+1} = 0$, $a_0 = a_{n+1} = 0$, $b_0 = b_{n+1} = \infty$ dir. N^+ sembolü $N \cup \{0, n + 1\}$ için A^+ sembolü de $A \cup \{(0, n + 1)\}$ için kullanılır. Burada $(0, n + 1)$ arkı $d_{0,n+1} = 0$ mesafesine ve $t_{0,n+1} = 0$ seyahat zamanına sahip yapay arktır. Bozulabilir ürün taşındığından dolayı bir rota içerisindeki tüm müşterilere belirli bir zaman içerisinde hizmet sunulmalıdır. Son şart rota başlama zamanına bir t_{max} eklenerek tanımlanır. Ayrıca araç yüklemesi için bir hazırlanma süresi σ^r her bir $r \in K$ rotası ile ilişkilendirilir. Amaç kapasite, zaman penceresi ve zaman sınırı kısıtları sağlanırken tüm müşterilere servis edecek toplam seyahat uzunluğunu minimize etmektir. Keyfi büyük bir sabit M kullanarak problem aşağıdaki gibi formüle edilebilir:

x_{ij}^r : $(i, j) \in A^+$ arkı r rotasında ise 1, aksi durumda 0 dır; r rotası boş ise $x_{0,n+1}^r$ değeri 1 olur,

y_i^r : i müşterisi r rotasında ise 1, aksi durumda 0,

t_i^r : r rotasındaki i müşterisinde servise başlama zamanı,

t_0^r : r rotasının başlama zamanı,

t_{n+1}^r : r rotasının bitiş zamanı,

$$\text{Min} \sum_{r \in K} \sum_{(i,j) \in A} d_{ij} x_{ij}^r \quad (2.36)$$

$$\sum_{j \in N^+} x_{ij}^r = y_i^r, \quad i \in N, \quad r \in K, \quad (2.37)$$

$$\sum_{r \in K} y_i^r = 1, \quad i \in N, \quad (2.38)$$

$$\sum_{i \in N^+} x_{ih}^r - \sum_{j \in N^+} x_{hj}^r = 0, \quad h \in N, \quad r \in K, \quad (2.39)$$

$$\sum_{i \in N^+} x_{0i}^r = 1, \quad r \in K, \quad (2.40)$$

$$\sum_{i \in N^+} x_{i(n+1)}^r = 1, \quad r \in K, \quad (2.41)$$

$$\sum_{i \in N} q_i y_i^r \leq Q, \quad r \in K, \quad (2.42)$$

$$t_i^r + s_i + t_{ij} - M(1 - x_{ij}^r) \leq t_j^r, \quad (i, j) \in A^+, \quad r \in K, \quad (2.43)$$

$$a_i y_i^r \leq t_i^r \leq b_i y_i^r, \quad i \in N, \quad r \in K, \quad (2.44)$$

$$t_0^1 \geq \sigma^1, \quad (2.45)$$

$$t_{n+1}^r + \sigma^{r+1} \leq t_0^{r+1}, \quad r = 1, \dots, k-1, \quad (2.46)$$

$$\sigma^r = \beta \sum_{i \in N} s_i y_i^r, \quad r \in K, \quad (2.47)$$

$$t_i^r \leq t_0^r + t_{max}, \quad i \in N, \quad r \in K, \quad (2.48)$$

$$x_{ij}^r = \{0,1\}, \quad (i,j) \in A^+, \quad r \in K, \quad (2.49)$$

$$y_i^r = \{0,1\}, \quad i \in N, \quad r \in K. \quad (2.50)$$

Formül içerisinde kısıt (2.38) her müşterinin sadece bir kez ziyaret edilebileceğini gösterir. (2.39), (2.40) ve (2.41) denklemleri araç yolu olarak tanımlanan akışı koruma kısıtlarıdır. Kısıt (2.42) bir rota üzerindeki talep toplamının araç kapasitesini aşamayacağını belirten kısıttır. (2.43)-(2.46) denklemleri zaman çizelgesinin olabilirliğini garantiye alır. (2.47) kısıdı bir rota içerisindeki tüm müşterilerin β parametresi ile çapılmış servis zamanlarının toplamı olan araç hazırlanma zamanını tanımlar. Son olarak, (2.48) bir müşteriye servis için zaman sınır kısıdına karşılık gelir. r rotasında i müşterisi yer almadığında (2.44) denkleminin t_i^r değişkenlerini 0 yapmaya çalıştığı dikkate alınmalıdır. Sonuç olarak, (2.48) denklemi bu durumda otomatik olarak sağlanır.

Farklı model kurulumlarını incelemek için, çok depolu heterojen filoya sahip ZPARP hakkında Dondo ve Cerda' nın [45] yapmış olduğu çalışma, Tan vd. [47] yayımladıkları makaleleri ve Toth ve Vigo' nun [4] çıkardıkları "The vehicle routing problem" isimli kitaptaki zaman kısıtlı model baz alınabilir. Literatürde bunlar dışında başka modellerde mevcuttur.

2.2.5.5 Dağıtımli Toplamalı Araç Rotalama Problemi (DTARP)

Araç rotalama probleminin temel versiyonlarından olan dağıtımli toplamalı araç rotalama probleminde (DTARP) $\forall i$ müşteri d_i dağıtım ve p_i toplama miktarları ile ilişkilendirilir. Bazı durumlarda müşteriler için tek bir talep miktarı $d'_i = d_i - p_i$ olarak kullanılabilir. Böylece miktar negatif olabilir. $\forall i$ müşteri için O_i dağıtım talebinin merkezi olan düğümü, D_i toplama talebinin gideceği yer olan düğümü gösterir (Toth ve Vigo [4]).

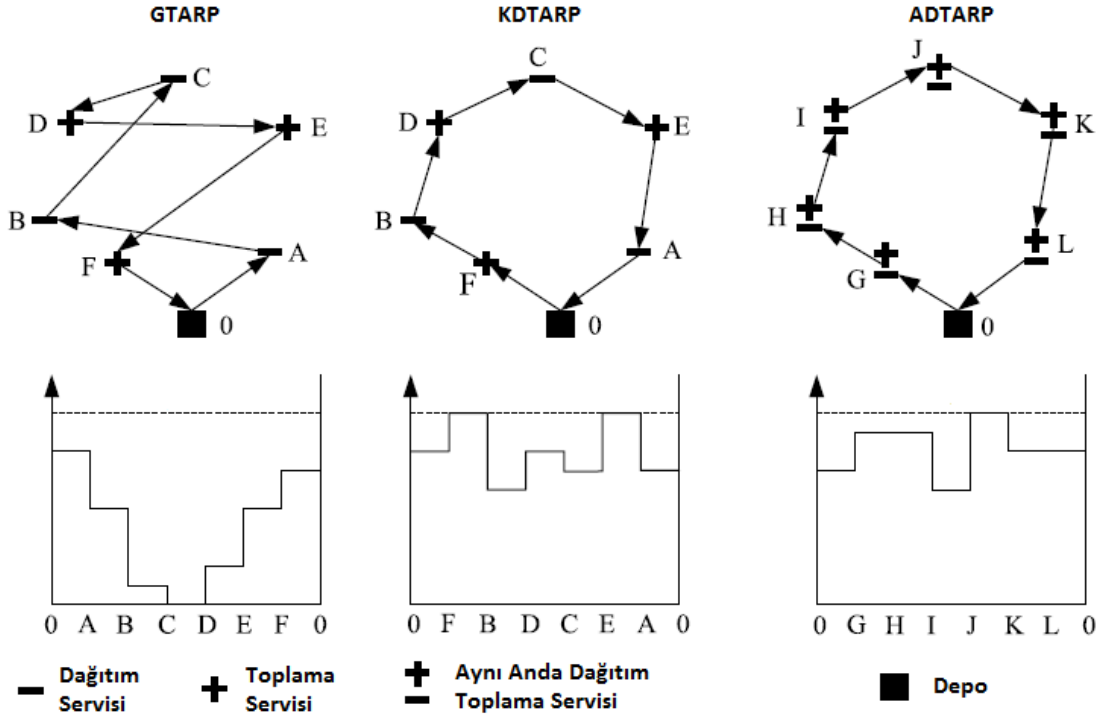
Her müşteri düğümünde dağıtım işlemi toplama işleminden önce yapılmaktadır. Bu sebeple aracın mevcut yükü belirlenen bir düğümüne varmadan önce başlangıç yükü eksi hali hazırdaki dağıtım taleplerinin tümü artı hali hazırdaki toplama talepleri olarak ifade edilmektedir (Toth ve Vigo [4]).

Minimum maliyetli rota birleşimini bulmak için DTARP kısıtları (Toth ve Vigo [4]):

- her rota depoyu ziyaret etmelidir,
- her müşteri noktası tam olarak bir kez ziyaret edilmelidir,
- rota boyunca aracın mevcut yükü negatif olamaz ve asla araç kapasitesini geçmemelidir,
- aynı rota içerisinde depodan farklı O_i müşteriye D_i müşterisinden önce hizmet verilmelidir.

Birçok DTARP çeşidinden biri olan aynı anda dağıtımli toplamalı araç rotalama problemi (ADTARP) için Dethloff (2001), Nagy ve Salhi [48] ve Tang Montané ve Galvão [49] tarafından matematiksel modeller önerilmiştir. Diğer bazı DTARP çeşitleri geri toplamalı araç rotalama problemi (GTARP) ve karışık dağıtımli toplamalı araç rotalama problemidir (KDTARP). GTARP de dağıtım işlemleri bitmeden toplam işlemi yapılmaz

iken, KDTARP de dğğümlere birden çok ziyaret koşulu ve sıra koşulu olmadan bazı dğğümler de sadece dağıtım bazılarında da sadece toplama işlemi yapılır. Şekil 2.4' de dağıtımli toplamalı araç rotalama problem çeşitleri gösterilmiştir (Zachariadis vd. [50]).



Şekil 2. 4 Dağıtımli toplamalı araç rotalama problem çeşitleri (Zachariadis vd. [50])

Maksimum mesafe kısıtlı DTARP matematiksel modeli aşağıdaki şekilde ifade edilebilir (Tang Montané ve Galvão [49]):

V : Müşteri kümesi

V_0 : Depo dahil müşteri kümesi $V_0 = V \cup \{0\}$

n : toplam müşteri sayısı $n = |V|$

c_{ij} : i ve j müşterileri arası uzaklık

p_j : j müşterisinin toplama talebi, $j = 1, \dots, n$

d_j : j müşterisinin dağıtım talebi, $j = 1, \dots, n$

Q : araç kapasitesi

MD : her bir k rotası için izin verilen maksimum uzunluk

$\bar{k}$: maksimum araç sayısı

x_{ij}^k : eğer (i, j) arkı k aracının rotasına ait ise 1, değil ise 0

y_{ij} : müşterilerdeki toplama taleplerinin i müşterisine (i müşterisi dahil) kadar rotalanan ve (i, j) arkında taşınan miktarı

z_{ij} : i müşterisinden sonra müşterilere dağıtılacak talebin rotalanan ve (i, j) arkında taşınan miktar

$$\text{Min} \sum_{k=1}^{\bar{k}} \sum_{i=0}^n \sum_{j=0}^n c_{ij} x_{ij}^k, \quad (2.51)$$

$$\sum_{i=0}^n \sum_{k=1}^{\bar{k}} x_{ij}^k = 1, \quad j = 1, \dots, n, \quad (2.52)$$

$$\sum_{i=0}^n x_{ij}^k - \sum_{i=0}^n x_{ji}^k = 0, \quad j = 0, \dots, n, \quad k = 0, \dots, \bar{k}, \quad (2.53)$$

$$\sum_{j=1}^n x_{0j}^k \leq 1, \quad k = 1, \dots, \bar{k}, \quad (2.54)$$

$$\sum_{i=0}^n \sum_{j=0}^n c_{ij} x_{ij}^k \leq MD, \quad k = 1, \dots, \bar{k}, \quad (2.55)$$

$$\sum_{i=0}^n y_{ji} - \sum_{i=0}^n y_{ij} = p_j, \quad \forall j \neq 0, \quad (2.56)$$

$$\sum_{i=0}^n z_{ij} - \sum_{i=0}^n z_{ji} = d_j, \quad \forall j \neq 0, \quad (2.57)$$

$$y_{ij} + z_{ij} \leq Q \sum_{k=1}^{\bar{k}} x_{ij}^k, \quad i, j = 0, \dots, n, \quad (2.58)$$

$$x_{ij}^k \in \{0,1\}, \quad i, j = 0, \dots, n, \quad (2.59)$$

$$y_{ij} \geq 0, \quad i, j = 0, \dots, n, \quad (2.60)$$

$$z_{ij} \geq 0, \quad i, j = 0, \dots, n. \quad (2.61)$$

Amaç fonksiyonu ile toplam seyahat mesafesi minimize edilmeye çalışılmaktadır. Kısıt (2.52) ile her müşterinin tam olarak bir araç tarafından ziyaret edilmesi sağlanır. Kısıt (2.53) her müşteriye aynı aracın varış ve çıkışını garanti eder. Kısıt (2.54) ile en fazla $\bar{k}$ aracın kullanılabileceği belirtilir. (2.55) maksimum uzaklık kısıdıdır. (2.56) ve (2.57) denklemleri sırasıyla toplama ve dağıtım için akış denklemleridir. Tüm müşteriler için her iki talebinde karşılanmasını garanti eder. Kısıt (2.58) toplama ve dağıtım taleplerinin sadece çözüm içerisinde kullanılan arklar ile taşınabileceğini belirler. Ayrıca rotanın herhangi bir bölümünde bir araç tarafından taşınan toplam yükün bir üst sınıra sahip olduğunu gösterir.

2.2.5.6 Zaman Pencereleli Geri Toplamalı Araç Rotalama Problemi (ZPGTARP)

Belirli bir zaman penceresinde geri toplama işleminin yapıldığı araç rotalama problemidir. ZPARP ve GTARP' nin bir arada bulunduğu ARP türüdür.

2.2.5.7 Zaman Pencereleli Dağıtımlı Toplamalı Araç Rotalama Problemi (ZPDARP)

ZPDARP zaman penceresi koşulu altında dağıtım toplama işleminin yapıldığı ARP türüdür. ZPARP ve DTARP' nin birleştirilmiş halidir.

2.2.5.8 Diğer Araç Rotalama Problemi Türleri

Stokastik Araç Rotalama Problemi (SARP): Ziyaret edilecek müşteri kümesi, müşteri talepleri veya seyahat zamanları bilinen bir olasılık dağılımından gelen rassal değişkenler olarak modellenebilen araç rotalama problemidir (Baykoç ve İşleyen [51]).

Ayrık Dağıtımlı Araç Rotalama Problemi (ADARP): Farklı araçların aynı müşteriye hizmet verebildiği araç rotalama problemidir (Neo [52]). ADARP de müşteri talepleri araç kapasitelerinden fazla olabilir (Kemer [53]).

Periyodik Araç Rotalama Problemi (PARP): Aynı zamanda verilen bir periyot içerisinde müşteri dağıtımlarının düzenlenmesi olarak tanımlanır. Periyodun her bir günü için araç

rotaları toplam gezilecek mesafeyi minimize edecek şekilde kurulmalıdır (Alonso vd. [54]).

Çok Depolu Araç Rotalama Problemi (ÇDARP): Birden çok depoya sahip olan araç rotalama problemleridir. Depo başına hizmet görecektir müşteriler tespit edilir ve rotalar oluşturulur. Bir depodan çıkan araçlar tekrar aynı depoya geri dönüyor ise problem depo başına ayrı ayrı klasik araç rotalama problemi olarak tanımlanabilir (Keskintürk [55]).

Açık Uçlu Araç Rotalama Problemi (AUARP): Klasik ARP den farklı olarak araç rota üzerindeki müşteri taleplerini karşıladıktan sonra depoya geri dönmemektedir (Li vd. [56]).

ARAÇ ROTALAMA PROBLEMİ ÇÖZÜM YAKLAŞIMLARI

Araç rotalama probleminin Dantzig ve Ramser [1] tarafından 1959 yılında sunulmasından bu yana araştırmacılar tarafından klasik ARP ve çeşitleriyle ilgili birçok çözüm yöntemi sunulmuştur. NP-zor sınıfında bulunan ARP' lerde problem boyutu büyüdükçe optimal çözümü elde etmek güçleşmekte ve çözüm süresi uzamaktadır. Bu kapsamda kesin çözümlerin yanı sıra yaklaşık çözüme daha kısa sürede ulaşan sezgisel yöntemler sunulmuştur. Genel olarak ARP çözüm yöntemleri optimal sonuca ulaşp ulaşmamalarına göre kesin çözüm yöntemleri ve sezgisel yöntemler olmak üzere ikiye ayrılmaktadır. 1990' lı yıllardan günümüze kadar üzerinde çalışılan meta sezgiseller ile birlikte sezgisel yöntemler de klasik ve meta sezgiseller olarak ikiye ayrılmıştır.

3.1 Kesin Çözüm Yöntemleri

GSP için geliştirilmiş kesin çözüm yöntemleri ARP' de de kullanılmaktadır. Kesin çözüm yöntemleri matematiksel programlama tabanlı yöntemlerdir. Kesin yöntemler ile ARP için optimum sonuçlar bulunabilmektedir. Kesin çözüm yöntemleri literatürde farklı şekillerde sınıflandırılmıştır. Örnek bir sınıflandırma Laporte [6] dan incelenebilir. Tamsayılı model olarak kurulan ARP' yi çözmek için kullanılan bu yöntemlerden bizim burada vereceklerimiz en bilinenlerinden dal ve sınır, kesme düzlemi ve dinamik programlama yöntemleridir. Bunların yanı sıra dal ve kesme algoritması, dal ve değer algoritması, sütun oluşturma, ağaç arama gibi farklı metotlarda literatürde mevcuttur.

3.1.1 Dal ve Sınır Algoritması (Branch and Bound Algorithm)

Dal ve sınır algoritması problem için optimal çözümü ararken problem ile ilgili tüm aşamaları sistemli bir şekilde gözden geçirmektedir. Uygulamada birçok tamsayılı programlama problemi bu algoritma ile çözülebilmektedir. Algoritma öncelikle olası problem çözümlerinin kümesini daha küçük alt kümelere ayırır ve her bir alt küme (alt problem) için alt ve üst sınır değerleri belirler. Sonrasında, adım adım alt problemler çözüm kümesinden elenerek optimal çözüme ulaşılır. Algoritma dallandırma (branching), sınırlama (bounding) ve bağlama (fathoming) adımlarından oluşmaktadır.

Dallandırma: Büyük boyutlu problemler için çözüme ulaşmak zor olduğundan, problem alt problemlere ayrıştırılır (dallandırılır) ve bu dallandırma alt problemler bağlanıncaya kadar yapılır. Alt problemler oluşturulurken hangi değişkenin dallandırılması gerektiğine kesir değerinin ekonomik önemine göre karar verilir. En fazla ekonomik öneme sahip kesir değerli değişken dallandırılır. Dallandırma başka bir deyişle problemin uygun çözümler kümesinin daha küçük alt kümelere ayrıştırılmasıdır. Uygun çözümler kümesindeki her bir çözüme aday çözüm adı verilir.

Sınırlama: Problem içerisinde dallandırma adım sayısının azaltılması için sınırlama işlemi kullanılır. Sınırlama ile genellikle dallandırma sürecinde oluşturulan her bir alt problemin optimum çözümünün alabileceği en küçük veya en büyük değer belirlenir. Burada en küçük değer alt sınırı, en büyük değer ise üst sınırı oluşturur. Çoğu durumda, problem çözümünü zorlaştıran kısıtlayıcılardan birisi çıkarılarak problemin gevşetmesi (relaxation) elde edilir. Tamsayılı programlama problem çözümlerini en çok zora sokan kısıtlayıcılar, karar değişkenlerinin tamsayılı olması istenen kısıtlayıcılardır. Tüm alt problemler sınırlandığında, dal ve sınır algoritması sona erer. En iyi alt sınırda tamsayılı doğrusal programlama probleminin çözümü olur.

Bağlama: Bir alt problem;

- alt problemin uygun çözüme sahip olmaması,
- alt problemin tüm değişkenlerinin tamsayı değerli bir optimal çözüm vermesi,
- alt problemin optimal çözüm değerinin maksimizasyon problemlerinde diğer alt problemin çözüm değerinden küçük olması, durumlarında bağlanabilir (Öztürk [57]).

3.1.2 Kesme Düzlemi Algoritması (Cutting Plane Algorithm)

Doğrusal programlama problemleri için ilk kez Gomory tarafından 1958 yılında tamsayılı çözümleri sağlayacak hesaplama yöntemi geliştirilmiştir. Bu sebeple kesme düzlemi algoritmasına Gomory kesme düzlemi de denmektedir. Bu yöntem ile sonlu sayıda işlemten sonra optimal bir tamsayılı çözüm elde edilir. Doğrusal programlama probleminin simpleks yöntemiyle bulunan optimal sürekli çözümü başlangıç noktası olarak ele alınır. Eğer çözüm tamsayılı değil ise doğrusal programlama problemine kesim düzlemi kuralına göre elde edilmiş bir doğrusal kısıtlama dahil edilir.

Kesim düzlemi prosesinde ilk olarak simpleks yöntem ile oluşan optimal çözüm değerleri arasından en büyük olan kesir değerli karar değişkeni seçilir. Sonrasında bu değişken satırında bulunan değişken katsayıları tamsayı ve kesirli olarak ayrılır ve tamsayı olanlar denklemin sağ tarafında toplanır. Sağ tarafta yer alan tamsayı değişkenler atılır ve sadece kesirli elemanlar bırakılır. Böylece denklem eşitsizlik halini almış olur. Burada denklemin sol tarafı sağ tarafına göre büyük veya eşit olacaktır. Bu işlemlerden sonra ek kısıtlayıcı elde edilmiş olunur ve denklemin standart hale getirilmesi için bu ek kısıtlayıcıya yeni bir değişken eklenir. Bu ek kısıtlayıcı ile beraber simpleks çözüm adımlarına geçilir (Öztürk [57]).

3.1.3 Dinamik Programlama (Dynamic Programming)

Bir dizi karar verme işlemini optimize eden dinamik programlama bir matematik işlemler bütünüdür. Dinamik programlama çözüm yaklaşımı temelde problemi veya problemin bir kısmını parçalara böler ve parça çözümlerini bir sonraki alt parçanın girdisi olarak kullanır. Her bir alt problem için optimal bir çözüm bulunur. Bu çözümler problemin genel çözümüne eklenerek nihai çözüme ulaşılmaktadır. n sayıda karar verme aşamalarına sahip bir problem, n sayıda ve her biri tek bir karar değişkenine sahip, problemlere bölünür. Hesaplama süresi, bir problem içindeki değişkenler sayısınca üstel olarak büyürken, alt problemler sayısınca doğrusal olarak büyür (Çetin [58]).

3.2 Klasik Sezgisel Yöntemler

Klasik sezgisel yöntemler ARP' ler için turların yapımı ve geliştirilmesinde çözüm uzayını kısmen tarayan, kabul edilebilir süreler içerisinde optimum çözüme yakın iyi çözümler üretebilen yöntemlerdir. Klasik sezgiseller genel olarak üç kategoride sınıflandırılabilir.

- Rota kurucu (yapıcı) sezgiseller
- İki aşamalı sezgiseller
- Rota geliştirici (iyileştirici) sezgiseller

3.2.1 Rota Kurucu Sezgiseller (Route Building Heuristics)

Rota kurucu sezgisellerde, ARP çözümlerinin oluşturulması için bir tasarruf kriteri kullanarak mevcut rotaların birleştirilmesi ve bir ekleme maliyeti kullanarak araç rotalarına düğümlerin atanmasından oluşan iki ana teknik kullanılır. Bu sezgiseller çözüm maliyetini göz önünde tutarlarken aşamalı olarak arkların birbirine eklenmesi sonucu uygun bir çözüm rotası kurarlar fakat kendi başlarına bir gelişim aşaması içermemektedirler (Toth ve Vigo [4]). Bu sezgisellerin en bilineni Tasarruf algoritmasıdır.

Tasarruf Algoritması (Saving Algorithm)

Tasarruf sezgiseli ilk olarak Clarke ve Wright (1964) tarafından ARP çözümü için önerilmiştir. Bu yüzden Clarke ve Wright algoritması olarak da bilinmektedir. Algoritmanın birçok çeşidi ve iyileştirmesi önerilmiştir. Birçok farklı ARP türü için tasarruf algoritması önerilmişse de genelde KARP üzerine çalışmalar yapılmıştır (Ropke [59]). $(1, \dots, i, 1)$ ve $(1, j, \dots, 1)$ rotalarını tek bir $(1, \dots, i, j, \dots, 1)$ rotasında birleştirmek mümkün olduğunda bir s_{ij} uzaklık tasarrufu oluşturulur. Algoritma araç sayısının karar değişkeni olduğu problemler için geçerlidir, ayrıca yönlü ve yönsüz problemlerde eşit derecede çalışmaktadır. Fakat Vigo [60] çalışmasında potansiyel rota birleşimlerinin yarıya düşmesine rağmen yönlü durumlarda algoritmanın davranışının kötüleşebileceğini göstermiştir. Algoritmanın paralel ve sıralı olmak üzere iki versiyonu mevcuttur (Toth ve Vigo [4]).

Adım 1. $i, j = 2, \dots, n$ ve $i \neq j$ için $s_{ij} = c_{i1} + c_{1j} - c_{ij}$ tasarrufu hesaplanır. $i, j = 2, \dots, n$ için $n - 1$ adet $(1, i, 1)$ araç rotası oluşturulur.

Adım 2. Tasarruf azalan biçimde sıralanır.

Paralel versiyon (en uygun birleşme)

Adım 3. Sırasıyla $(i, 1)$ ve $(1, j)$ arklarını içeren iki araç rotası ele alınsın. Eğer $s_{ij} > 0$ ise $(i, 1)$ ve $(1, j)$ arklarının silinmesi ve (i, j) arkının oluşturulması ile bu rotalar geçici olarak birleştirilir. Ortaya çıkan rota uygun ise birleşme uygulanır. Daha fazla iyileşme mümkün olmayana kadar bu işleme devam edilir ve sonlandırılır (Laporte [6]).

Sıralı versiyon (yol genişletme)

Adım 3. Sırayla her bir $(1, i, \dots, j, 1)$ rotası ele alınsın. Mevcut rota ile başka bir ark (veya kenar) $(k, 1)$ veya ark (veya kenar) $(1, l)$ ' nin birleşiminin uygun olduğu bir s_{ki} veya s_{jl} tasarrufu tanımlanır. Birleşim uygulanır ve ortaya çıkan rotalar için süreç tekrarlanır. Eğer uygun bir birleşim mevcut değilse sonraki rota ele alınır ve aynı işlemler yeniden uygulanır. Uygun birleştirme kalmayana kadar devam edilir ve durdurulur (Toth ve Vigo [4]).

Bu prosedür $O(n^2 \log n)$ sürede gerçekleşir fakat bu karmaşıklık uygun data yapılarının kullanılmasıyla azaltılabilir. Aynı zaman da Gaskel (1967) , Yellow (1970) ve Paessens (1988) bu yöntemin bir dizi çeşidini önermişlerdir. Clarke ve Wright algoritması dolaylı olarak sabit araç maliyetini ve araç boyutunu göz ardı etmektedir. Araç maliyetleri f her bir c_{1j} ($j = 2, \dots, n$) için sabit maliyet eklenerek hesaba katılabilir. Tasarruf negatif olsa bile sabit araç sayılı çözümler gerekli sayıda rotaya ulaşana kadar adım 3 tekrar edilerek elde edilebilir (Laporte [6]).

3.2.2 İki Aşamalı Sezgiseller (Two-Phase Heuristics)

İki aşamalı sezgiseller, ARP çözüm sürecini iki alt probleme ayrıştırmaya dayanır (Cordeau vd. [44]):

- *Kümeleme:* her bir rotaya karşılık gelen alt kümelere müşteri ayrımının tanımlanması,
- *Rotalama:* her bir rota üzerinde müşteri sıralarının tanımlanması.

İki fazlı sezgiseller önce kümele sonra rotala metotları ve önce rotala sonra kümele metotları olmak üzere iki sınıfa ayrılır. İlk durumda düğümler uygun kümeler halinde

düzenlenir ve her biri için bir araç rotası oluşturulur. İkinci durumda ilk olarak tüm düğümler için bir rota oluşturulur ve sonra uygun araç rotalarına bölünür (Toth ve Vigo [4]).

3.2.2.1 Süpürme Algoritması (Sweep Algorithm)

Wren (1971), Wren ve Holliday (1972) ve Gillett ve Miller' ın (1974) sunmuş olduğu süpürme algoritması önce kümele sonra rotala yaklaşımlarının ilk örneği olarak bilinir. Algoritma düzlemsel (planar) ARP örneklerine uygulanmaktadır. Algoritma keyfi bir müşteriden başlar ve daha sonra depo ve başlangıç müşterisi ile ilgili polar açılarının artan sırasına göre ele alınarak mevcut araca geri kalan müşteriler atanır. En kısa sürede mevcut müşterinin mevcut araca atanması yapılamıyorsa bu müşteri ile yeni bir rota başlatılır. Tüm müşteriler araçlara atandığında her bir rota ayrı ayrı GSP çözülerek tanımlanır (Cordeau vd. [8]).

Bu metodun temel adımları aşağıdaki gibidir. Her bir i köşesinin kendi polar kutupları (θ_i, ρ_i) ile tanımlandığı varsayılın. Burada θ_i açı, ρ_i ışın uzunluğudur. Keyfi bir i^* köşesi için $\theta_{i^*} = 0$ olsun ve $(0, i^*)$ den geri kalan açılar hesaplınsın. Köşeler θ_i ' lerin artan derecelerine göre sıralanır.

Adım 1. Kullanılmamış bir k aracı seçilir.

Adım 2. En küçük açiya sahip rotalanmamış köşeden başlar ve araç kapasitesi aşılmamak üzere köşeler araca atanır. Eğer rotalanmamış köşe kaldı ise adım 1 e geri dönülür.

Adım 3. İlgili GSP' nin tam veya yaklaşık olarak çözülmesi ile rotalar ayrı ayrı optimize edilir(Toth ve Vigo [4]).

3.2.2.2 Fisher ve Jaikumar Algoritması (Fisher and Jaikumar Algorithm)

Fisher ve Jaikumar algoritması öncelikle genelleştirilmiş atama probleminin (GAP) çözülmesi ile mümkün müşteri kümelerinin oluşturulduğu ve sonra GSP algoritması ile her bir küme üzerinde bir araç rotasının belirlendiği iki aşamalı bir süreçtir (Cordeau vd. [8]).

1. Aşama (GAP çözümü)

Adım 1 (tohum seçimi). Her bir k kümesini başlatmak için V düğüm kümesinden j_k tohum düğümlerinin seçilmesi.

Adım 2 (düğümlerin tohumlara atanması). Her bir k kümesi için her bir i müşterisinin atamasının $d_{ik} = \min\{c_{0i} + c_{ij_k} + c_{j_k0}, c_{0j_k} + c_{j_ki} + c_{i0}\} - (c_{0j_k} + c_{j_k0})$ maliyetinin hesaplanması.

Adım 3 (genelleştirilmiş atama). d_{ij} maliyeti, q_i müşteri ağırlıkları ve Q araç kapasiteleri ile GAP' ın çözümü.

2. Aşama (GSP çözümü)

Adım 4. GAP çözümüne ilişkin her bir küme için ilgili GSP' nin çözümü (Toth ve Vigo [4]).

3.2.2.3 Taç Yapağı Algoritması (Petal Algorithm)

Süpürme algoritmasının doğal bir uzantısı olan petal sezgiseli mümkün rotaların büyük bir kümesini oluşturur ve bir dizi bölümlene problemini çözümleyerek son alt kümeyi seçer. Bölümlene probleminin matematiksel modeli:

$$\text{Min} \sum_{k \in S} d_k x_k \quad (3.1)$$

$$\sum_{k \in S} a_{ik} x_k = 1, \quad \forall i = 1, \dots, n, \quad (3.2)$$

$$x_k \in \{0,1\}, \quad \forall k \in S. \quad (3.3)$$

Burada S rotalar kümesi, k rotası eğer çözüme ait ise $x_k = 1$, a_{ik} sadece k rotasına ait i düğümü için 1 ve d_k k petalinin maliyetidir.

Formülasyon ilk olarak Balinski ve Quandt (1964) tarafından önerilmiştir fakat büyük $|S|$ değerleri için kullanışsız olmaktadır. Agarwal vd. (1989) küçük ölçekli ($10 \leq n \leq 25$) ARP optimalliğini çözmek için sütun üretme yöntemini kullanmışlardır (Toth ve Vigo [4]). Foster ve Ryan (1976) ve Ryan vd. [61] seçilecek rota kümelerini belirlemek için 1-petal olarak adlandırılan sezgisel kurallar önermişlerdir. Renaud vd. [62] 2-petal

olarak isimlendirdikleri petal sezgiselinin bir uzantısını tanımlamışlardır. 2-petal sezgiseli gömülü veya kesişen rotalardan oluşmaktadır. Petal sezgiselinin genel performansı süpürme algoritmasınıninkinden üstündür (Cordeau [44]).

3.2.2.4 Önce Rotala Sonra Grupla Algoritması (Route-First, Cluster-Second Algorithm)

Algoritma yan kısıtları göz ardı ederek ilk aşamada büyük bir GSP turu inşa eder, ikinci aşamada GSP turu mümkün araç rotalarına ayrıştırılır. Bu algoritma bağımsız araç sayılı problemlerde uygulanır. Algoritma ilk defa döngüsüz bir şebeke üzerinde $O(n^2)$ zamanda çözülen standart bir en kısa yol algoritma problemi olan ikinci aşamayı ele alan Beasley tarafından ortaya atılmıştır. En kısa yol algoritmasında i ve j düğümleri arasındaki seyahat maliyeti $d_{ij} = c_{0i} + c_{0j} + I_{ij}$ dir. Burada I_{ij} , GSP turu üzerindeki i den j ye seyahat maliyetidir. Önce rotala sonra grupla sezgisellerinin diğer yaklaşımlarla rekabet edebildiğini gösteren herhangi bir sayısal mukayese mevcut değildir (Neo [63]).

3.2.3 Tur Geliştirici Sezgiseller (Route Improvement Heuristics)

İyileştirme sezgiselleri araç rotaları içerisinde veya arasında bir dizi kenar veya düğüm değişimi gerçekleştirerek herhangi daha iyi uygun bir çözüme ulaşmaya çalışır (Toth ve Vigo [4]). Yerel arama algoritmaları genellikle diğer sezgiseller tarafından oluşturulmuş başlangıç çözümlerini geliştirmek için kullanılır. Belirli bir çözümden başlayarak bir yerel arama yöntemi daha iyi maliyetli komşu çözümler elde etmek için ark değişimi veya müşteri hareketleri gibi basit değişimler uygular. Eğer gelişen bir çözüm bulunursa başlangıç çözüm ile yer değiştirilir ve süreç tekrarlanır. Aksi durumda yerel bir minimum noktası tespit edilmiştir. Birçok çeşit komşuluk mevcuttur. Bunlar bir tek rota üzerinde işlem yapılıyorsa rota içi komşuluklar, aynı anda bir den fazla rota ele alınıyorsa rotalar arası komşuluklardır. En yaygın komşuluk türü GSP için Lin [64] tarafından önerilmiş λ – opt sezgiselidir (Cordeau [44]).

3.4 Meta Sezgiseller Yöntemler

İlk yıllarda, özelleştirilmiş sezgiseller genellikle ARP gibi karmaşık kombinatoriyal optimizasyon problemlerinin çözümü için geliştirilmişlerdir. Daha sonra Fred Glover [65] tarafından meta sezgisel olarak isimlendirilen daha genel çözüm yapısı tasarlanmıştır. Ortaya çıkan problemler için sıfırdan yeni bir sezgisel geliştirmekten ziyade problemin meta sezgisel çözüm yapısına adapte edilmesi daha az çalışma gerektirir. Ayrıca iyi bir meta sezgisel uygulaması makul hesaplama süresi içerisinde yakın optimal çözümleri sağlayabilir (Golden vd. [2]). Meta sezgisellerin temel dezavantajı daha uzun hesaplama zamanı, daha yüksek global minimum bulma olasılığı olarak tanımlanan durdurma kriterine sahip olmamalarıdır. Meta sezgisellerin birçoğu fiziksel veya biyolojik süreçlerin ilkelerine dayanmaktadır (Breedam [66]).

Yaklaşık son 30 yıldır araştırmacıların çoğu temel olarak yerel arama ve popülasyon arama prensiplerini kullanan meta sezgisel algoritmaların geliştirilmesi için çalışmalar yapmışlardır. Yerel arama yöntemlerinde yoğun bir keşif, mevcut çözümden kendi komşuluğundaki diğer bir mümkün çözüme olan her bir işlem adımındaki hareketler ile gerçekleştirilir. Tavlama benzetimi ve tabu arama bu prensibe örnek verilebilir. Popülasyon arama, iyi bir ebeveyn kümesi oluşturulmasını ve yavru üretimi için bu kümede ki ebeveynlerin birleştirilmesini içerir. Klasik bir örnek yavru üretimi için ebeveyn birleşimini içeren genetik aramadır. Ayrıca birçok yavru üretimi için birçok çiftin kullanıldığı adaptif hafıza prosedürüne karınca koloni algoritması örnek verilebilir (Cordeau [8]).

Meta sezgiseller arama sırasında yoğunlaşma ve çeşitlendirme ilkelerini uygulamaktadırlar. Çeşitlendirme sezgiselin yerel bir optimum çözüm tarafından yakalanmasını önlerken yoğunlaşma çözüm uzayının en umut verici bölgelerinin daha ayrıntılı aranmasını sağlar. Bu nedenle meta sezgiseller tarafından üretilen çözümlerin kalitesi genellikle klasik sezgisellerden elde edilenlere göre daha yüksektir. Fakat meta sezgiseller genellikle etkili arama için ince ayarlı parametreler gerektirir (Çiçekdeğ [67]).

3.4.1 Tavlama Benzetimi (Simulated Annealing)

Tavlama benzetimi (TB) ilkesi katıların fiziksel tavlama işleminden elde edilir. Kirckpatrick vd. [68] ve Cerny [69] kombinatoriyal optimizasyon problemleri için istatistiksel mekanik alanındaki Metropolis vd.'nin (1953) araştırmasına dayanan TB kullanımını önermişlerdir (Breedam [66]).

TB iki nedenden dolayı yerel arama içerisinde özel bir rol oynamaktadır. İlk olarak, birçok pratik problemde uygulandığında oldukça başarılı olduğu gözlemlenmektedir. İkinci olarak asimptotik yakınsamanın teorik analizini kolaylaştıran stokastik bir bileşene sahiptirler. TB stokastik bir algoritmadır. Zayıf yerel optimumdan kaçışları mümkün kılmak için kontrollü bir şekilde rastgele tepe tırmanma hareketlerine izin verir. TB yoğun madde fiziği alanındaki tavlama işlemli bir benzetmeye adını borçludur. Burada katı madde, sıvı faz halinde tüm parçacıkları gelişi güzel hareket edinceye kadar ısıtılır ve sonra sıvı yüksek bir kafes (lattice) yapısında düzenlenmiş parçacıklar ile donmuş hale gelene kadar sıcaklık dikkatli ve yavaş bir şekilde azaltılır ve sistemin enerjisi minimumdur. Bu temel durum yalnızca maksimum sıcaklığın yeterince yüksek olması ve soğutmanın yeterince yavaş olması ile gerçekleşebilir. Aksi takdirde bir yarı-kararlı durum elde edilir (Floudas ve Pardalos [26]).

Tavlama benzetiminin t iterasyonunda bir x çözümü x_t 'nin $N(x_t)$ komşuluğunda rastgele çizilir. Eğer $f(x) \leq f(x_t)$ ise $x_{t+1} = x$ olur, aksi durumda

$$x_{t+1} := \begin{cases} x, & p_t \text{ olasılığı ile} \\ x_t & 1 - p_t \text{ olasılığı ile} \end{cases} \quad (3.4)$$

dır. Burada p_t genellikle $f(x) - f(x_t)$ ve t 'nin azalan fonksiyonudur. Genel olarak p_t aşağıdaki şekilde tanımlanır:

$$p_t = \exp(-[f(x) - f(x_t)]/\theta_t). \quad (3.5)$$

Burada θ_t , t iterasyonundaki sıcaklığı ifade eder. Kural soğutma programı olarak adlandırılan θ_t 'nin tanımlanması için kullanılır. Genellikle θ_t t 'nin azalan bir basamak fonksiyonudur: başlangıç olarak θ_t bilinen bir $\theta_1 > 0$ değerine eşitlenir ve her T

iterasyonundan sonra $\alpha(0 < \alpha < 1)$ katsayısı ile çarpılır. Böylece kötü bir çözümün kabul olasılığı zaman ile azalmalıdır (Toth ve Vigo [4]).

Temel TB prosedürü aşağıdaki verilmiştir (Breedam [66]).

Adım 1. Başlangıç çözümü mevcut çözümdür ve kaydedilir.

Adım 2. Kabul oranı p_t dikkate alınarak başlangıç sıcaklığı θ_1 tanımlanır. Mevcut sıcaklık θ_t başlangıç sıcaklığı θ_1 e eşitlenir.

Adım 3. Mevcut çözümden rastgele seçilmiş bir hareket ile bir komşu çözüm üretilir.

Adım 4. Mevcut çözüm ve komşu çözüm arasındaki amaç fonksiyon değerindeki fark δ olarak hesaplanır. Eğer $\delta > 0$ ve $r \geq e^{-\delta/\theta_t}$ ise (r $[0,1]$ arasında rastgele bir sayı) adım 3 e geri dönülür.

Adım 5. Komşu çözüm mevcut çözüm olur. Şimdiye kadarki en iyi çözüm komşu ise kaydedilir. Termal denge kontrol edilir. Mevcut çözüm değeri her 10 adımda yani bir dönemde önceden kaydedilmiş dönem değerleri ile karşılaştırılır. Eğer sapma %10 dan fazla ise dengeye ulaşılmamış demektir ve adım 3 e geri dönülür.

Adım 6. Soğutma fonksiyonu $\theta_t = \alpha \cdot \theta_{t-1}$ tarafından mevcut sıcaklık θ_t azaltılır. Eğer kabul edilen hareket yüzdesi en iyi çözüm olmadan beş ardışık sıcaklık azalımı için kritik kabul oranı değersizdir ve sonra adım 7 e geçilir, değil ise adım 3 e gidilir.

Adım 7. Tutulan çözüm en iyi son çözümdür.

3.4.2 Tabu Arama (Tabu Search)

Tabu arama (TA) genel bir yinelemeli meta sezgisel olarak Glover tarafından [70] ve [71] makalelerinde tanıtılmıştır. Sayısal gözlemler TA' nın neredeyse bilinen tüm teknikler ile rekabet edebilecek ve esnekliği ile birçok klasik prosedürü yenebilecek iyi bir yaklaşım tekniği olduğunu göstermiştir. TA her bir iterasyonda mevcut çözüm S den komşuluğu $N(S)$ ' e veya hesaplama verimliliği için $N(S)$ ' nin bir alt kümesine hareketler ile çözüm uzayını araştırır ve bir $S' \in N(S)$ çözümüne ulaşılır. TA, S' nin $N(S)$ komşuluğundaki bir S çözümünden diğer bir S' çözümüne tekrarlı olarak yapılan hareketler tarafından yinelemeli bir teknik olarak görülebilir. Bu hareket amaç fonksiyonunun kötüleşmesine neden olsa bile TA bir çözümden en iyi kabul edilebilir

komşuya hareket eder. Döngüyü engellemek için algoritmanın son zamanlarında araştırılmış ve ilan edilmiş çözümler yasaktır veya bir dizi yineleme için tabu vardır. Belirli kriterler (aspirasyon kriterleri) yerine getirildiğinde bir çözümün tabu durumu geçersiz kılınır. Bazen yoğunlaşma ve çeşitlendirme stratejileri aramayı artırmak için kullanılır (Tarantilis vd. [72], Floudas ve Pardalos [26]).

Breedam [66] tarafından sunulan TA prosedürü aşağıdaki gibidir.

Adım 1. Başlangıç çözümü mevcut çözümdür.

Adım 2. Mevcut çözümün hareketlerinin tam değerlendirmesi yapılır. Rota sayısına göre rotalar seçilir ve rota sırasına göre durur.

Adım 3. Bulunan en iyi hamle amaç fonksiyon değerinde bir iyileşme yapıyorsa adım 5 e gidilir.

Adım 4. Eğer uzun süreli hafıza kullanılırsa en bozucu hareket seçildiğinde hareket ile ilgili ek bir ceza dikkate alınmalıdır. Eğer en iyi hareket tabu listesinde ise tabu listesinde olmayan ve amaç fonksiyonun en az bozulmasına neden olan hareket seçilir.

Adım 5. Hareket gerçekleştirilir. Yeni çözüm mevcut çözüm olur. Eğer yeni çözüm şimdiye kadarki çözümlerin en iyisi ise yeni çözüm kaydedilir. Eğer hala bazı kalan yerler varsa tabu listesine hareket eklenir. Aksi takdirde hareket atılır.

Adım 6. Listedeki ilk hareketi kaldırarak tabu listesini yeniden düzenlenir ve bir pozisyon öne kadar art arda itilir. Gerekirse tabu listesinin uzunluğu uydurulur.

Adım 7. Eğer yineleme sayısı 500 den daha az ise 2. adıma geri gidilir.

Adım 8. Kaydedilen çözüm nihai en iyi çözümdür.

3.4.3 Genetik Algoritma (Genetic Algorithm)

Genetik algoritmalar (GA) doğal seleksiyon ve doğal genetik mekanizmasına dayalı arama işlemleridir. İlk GA John H. Holland (1975) tarafından makine öğrenmesi ve fonksiyon optimizasyonu gibi kombinatoriyal problemler ve zor arama için çözüm gelişiminin hesaplanmasını sağlamak için geliştirilmiştir. Genetik algoritmalar özellikle araç rotalama problemi gibi problemler için ilgi çekici bir yaklaşım sunmaktadır. Çünkü

genetik algoritmalar genellikle büyük, doğrusal olmayan ve tam olarak anlaşılamamış uzayların hızlı global araması için oldukça etkilidirler (Floudas ve Pardalos [26]).

Genetik algoritma eşeyli üreme türlerinin gelişimi ile ilgili süreci taklit eden popülasyon tabanlı algoritmalar. Bir GA bazı yakınlaşma kriterleri karşılanana kadar yinelemeli bir süreç ile yeni yavru nesiller oluşturarak kromozom gibi kodlanmış bireylerin nüfusunu geliştirir. Bu sürecin sonunda rastgele üretilmiş kromozomların bir başlangıç popülasyonunun gelişmesi ve daha iyi yavrular ile yer değiştirmesi beklenmektedir. Üretilen en iyi kromozom ilgili çözümü sağlamak için açılır (Tarantilis vd. [72]).

$X^1 = \{x_1^1, \dots, x_N^1\}$ kromozomlarından rastgele üretilmiş bazı başlangıç popülasyonlarından başlayarak her bir $t = 1, \dots, T$ yinelemesinde adım 1, 2 ve 3 ($k \leq N/2$) olmak üzere k kez uygulanır ve sonra adım 4 e geçilir.

Adım 1 (yeniden çoğalma). X^1 kromozomlarından ebeveyn seçilir.

Adım 2 (yeniden birleşme). Bir çaprazlama operatörü kullanarak iki ebeveynden iki yavru üretilir.

Adım 3 (mutasyon). Her bir yavruya rastgele bir mutasyon uygulanır (küçük bir olasılıkla).

Adım 4 (nesil değiştirme). X^t deki en kötü $2k$ çözüm çıkarılarak ve yerlerine $2k$ yeni yavru konarak X^t den X^{t+1} oluşturulur.

Bu algoritma içerisinde T parametresi nesillerin sayısı ve k her bir üretim başına seçim sayısıdır. T nesilleri üzerinden üretilmiş en iyi çözümler bu algoritmanın nihai sonucudur (Toth ve Vigo [4]).

3.4.4 Karınca Koloni Optimizasyonu (Ant Colony Optimization)

Karınca koloni optimizasyonu (KKO) yemek arayışında olan bir karınca kolonisinin gerçek yaşam davranışlarından ilham almaktadır. Karıncalar yem araştırmalarında feromon adı verilen bir madde bırakarak kullandıkları yolları işaretlemektedirler. Yol üzerindeki feromon miktarı o yolun karıncalar için umut verici olduğunu veya olmadığını göstermektedir. Karıncaların bir optimizasyon problemi için çözüm kurucu süreçler olduğu meta sezgisel teknik Colorni vd. [73] tarafından sunulmuştur. Çözüm

uzayının nasıl keşfedildiğine göre, bazı değerler feromon eylemleri gibi benzer bir yol ile kaydedilir ve çözümlerin amaç değerleri besin kaynakları ile ilişkilendirilir. Bu algoritmanın önemli bir sorunu paralel çalışmasıdır. Aynı zamanda birçok çözüm kurulmakta ve karıncalar eski yinelemelerin bilgilerini kullanırken süreç esnasında bilgi değişimi yapmaktadır.

Temel KKO' nun her bir yinelemesinde her bir karınca adım adım bir problem çözümü oluşturur. Her bir adım da karınca bir olasılık fonksiyonunu takip ederek A_k yayılma durumlar kümesinin elemanları arasında gerçek kısmi çözüm seçimini tamamlamak için bir hareket gerçekleştirir. Her bir k karıncası için şu anki i durumundan başka bir j durumuna hareketinin $P_k(i, j)$ olasılığı aşağıdaki adımlar dikkate alınarak hesaplanır.

- Problem bilgisine göre hareketin η çekiciliği.
- Geçmişteki yemek taşımanın nasıl olduğunu gösteren hareket feromon seviyesi τ .
- Bir $Tabu_k$ yasaklı hareket listesi.

Orijinal karınca algoritmasında:

$$P_k(i, j) = \begin{cases} \frac{[\tau(i, j)]^\alpha [\eta(i, j)]^\beta}{\sum_{(i, z) \notin Tabu_k} [\tau(i, z)]^\alpha [\eta(i, z)]^\beta} & (i, j) \notin Tabu_k \\ 0 & (i, j) \in Tabu_k \end{cases} \quad (3.6)$$

dır. Burada α ve β η 'nin τ 'ya göreli etkisini belirlemek için kullanılan parametrelerdir. t yinelemesi tamamlandıktan sonra yani tüm karıncaları kendi çözümlerini tamamladıklarında feromon seviyeleri aşağıdaki formüle göre yenilenir.

$$\tau(i, j) = \varphi \cdot \tau(i, j) + \Delta\tau(i, j) \quad (3.7)$$

Burada φ feromon devamlılık katsayı seviyesini ifade eden bir değişkendir ve $\Delta\tau(i, j)$ tüm karıncaların çözümleri içerisinde seçmiş oldukları (i, j) hareketine tüm karıncaların katkısını ifade eder (Mazzeo ve Loiseau [74]).

BÖLÜM 4

UYGULAMA

Bu çalışmada, İstanbul Mega Center da konumlanmış gıda sektöründe faaliyet gösteren bir dağıtım işletmesinin, müşteri grubuna ürün dağıtımında kullandığı araç/araçlar için en uygun rotalar belirlenerek optimal dağıtım planı oluşturulmaya çalışılmıştır. Çalışmada kapasite kısıtlı araç rotalama problemi için yazılmış tamsayılı matematik programlama modeli kullanılmıştır.

4.1 Uygulamanın Önemi

Günümüzde dağıtım ve/veya toplama yapan birçok firma, sahip oldukları araç/araçların hangi sıra ile hangi müşterilere uğraması gerektiği ile ilgili karar verme aşamasında zorlanmakta ve hızlı bir şekilde etkin çözümü elde edememektedirler. Bu bilgiyi elde etmek için bilgisayar yardımı olmaksızın geçmiş deneyimlerden yararlanmakta veya deneme yanılma yoluna başvurmaktadırlar. Böylelikle elde ettikleri rotalar sonucu yüksek maliyetler ortaya çıkmaktadır. Günümüze kadar daha düşük maliyetli etkili rotaların oluşturulması için akademisyenler ve araştırmacılar tarafından çeşitli yöntemler ortaya atılmıştır. Geliştirilen yöntemlerin firmalar tarafından kullanılması sonucu daha hızlı ve daha az maliyetli rotalar elde edilebilmiştir.

4.2 Uygulamanın Özellikleri

Uygulamanın gerçekleştirildiği Yıldırım Yumurtacılık Gıda ve San. Tic. Ltd. şirketi anlaşmaya varmış olduğu Çağrı Marketler zincirine ait mağazaların yumurta tedarikini karşılamaktadır. Yıldırım Yumurtacılık bünyesinde bulundurmış olduğu paketleme makinesi ile yumurtaları dağıtım öncesi 10' luk ve 30' luk paketler haline getirmekte ve

yumurtaları müşterilere bu paketleri içeren kutularla teslim etmektedir. Her bir kutu 30' luk yumurta kartonlarından 10 adet veya 10' luk kartonlardan 30 adet almaktadır. Haftanın 6 günü dağıtım yapan Yıldırım Yumurtacılık bir gün öncesinden belli olan müşteri taleplerini iki veya üç araç ile karşılamaktadır. Bu araçlardan ikisi günlük dağıtım için çıkarılırken üçüncü araç büyük boyutlu siparişleri dağıtmak üzere kullanılmaktadır. Bu sebeple uygulama sadece iki araç üzerine yapılmaktadır. İki araç birbirinden bağımsız olarak kendi müşterilerine dağıtım yapmaktadır. Araçlar karayolları tonaj sınırından dolayı en fazla 30' luk 1000 karton taşıyabilmektedir. Bu sebeple araç kapasiteleri eşit olarak ele alınır. Dağıtım yapılacak mağazalar İstanbul ve İzmit olmak üzere iki il'e yayılmış durumdadır.

4.3 Uygulamada Kullanılacak Tamsayıli Lineer Matematiksel Model

Kapasite kısıtlı bir araç rotalama problemi $G = (V, A)$ şebekesi üzerinde tanımlı olsun. Burada $V = \{0, 1, \dots, N, N + 1\}$ düğümleri, $A = \{(i, j) : i, j \in V, i \neq j\}$ arkların kümesini ifade etmektedir. Ele aldığımız araç rotalama probleminde temel amaç depodan başlayıp tekrar depoda son bulan araç rotalarının var olan kısıtlar altında minimize edilmesidir. Problemimizde her bir müşteriye tek bir araç hizmet etmektedir. Model sonucu " M " araç sayısı kadar rota oluşturulmaktadır.

Parametreler:

- "0" : Depo
- "1, ... N" : Müşteriler
- "N + 1" : Sanal Depo (Depo ile aynı lokasyonda)
- M : Araç Sayısı
- N : Müşteri Sayısı
- C : Araç Kapasitesi
- D_{ij} : i ' müşterisinden j ' müşterisine uzaklık
- q_i : i müşterisinin talebi

İndisler:

$i, j, p : \{0, 1, \dots, N, N + 1\}$

$k : \{1, \dots, M\}$

Değişkenler:

$X_{ijk} : \begin{cases} 1, & k \text{ aracı } i' \text{ müşterisinden } j' \text{ müşterisine giderse} \\ 0, & \text{aksi takdirde} \end{cases}$

$Y_i : \text{Alt tur oluşmasını engelleyen değişken}$

Amaç Denklemi:

$$\min Z = \sum_{i=0}^{N+1} \sum_{\substack{j=0 \\ j \neq i}}^{N+1} \sum_{k=1}^M D_{ij} * X_{ijk} \quad (4.1)$$

Kısıtlar:

$$\sum_{k=1}^M \sum_{j=1}^{N+1} X_{ijk} = M, \quad i = 0, \quad (4.2)$$

$$\sum_{i=0}^{N+1} X_{ipk} - \sum_{j=0}^{N+1} X_{pjk} = 0, \quad \begin{matrix} k = 1, \dots, M, \\ p = 1, \dots, N, \end{matrix} \quad (4.3)$$

$$\sum_{k=1}^M \sum_{\substack{j=0 \\ j \neq i}}^{N+1} X_{ijk} = 1, \quad i = 1, \dots, N, \quad (4.4)$$

$$\sum_{j=1}^N X_{0jk} = 1, \quad k = 1, \dots, M, \quad (4.5)$$

$$\sum_{i=1}^N X_{i,N+1,k} = 1, \quad k = 1, \dots, M, \quad (4.6)$$

$$\sum_{i=0}^{N+1} \sum_{\substack{j=0 \\ j \neq i}}^{N+1} q_i * X_{ijk} \leq C, \quad k = 1, \dots, M, \quad (4.7)$$

$$Y_j \geq Y_i + 1 - N \left(1 - \sum_{k=1}^M X_{ijk} \right), \quad i \neq j, \quad \begin{matrix} i = 1, \dots, N, \\ j = 1, \dots, N, \end{matrix} \quad (4.8)$$

$$Y_j \geq 0, \quad j = 1, \dots, N. \quad (4.9)$$

Amaç fonksiyonunda araçlar tarafından toplam kat edilen mesafe minimize edilmeye çalışılmıştır. Kısıt (4.2) depodan çıkan araç sayısını belirtmektedir. Kısıt (4.3)' e göre bir düğüme giren ve çıkan yolların sayısı eşittir. Kısıt (4.4) düğümlerden çıkan yollardan sadece birinin mutlak kullanılması gerektiğini ifade eder. Kısıt (4.5) ve (4.6) sırasıyla her bir aracın depodan çıktığını ve depoya girdiğini gösterir. (4.7) kısıdı araç kapasitesinin aşamayacağını belirtir. Son olarak (4.8) ve (4.9) numaralı kısıtlar alt tur oluşumunu engelleyen kısıtlardır.

4.4 Verilerin Toplanması ve Düzenlenmesi

Çalışmada firmanın 8-13 Nisan 2013 tarihleri arasında hizmet verdiği mağazalar ele alınmıştır. Dağıtım noktaları ve araç rotaları ile ilgili bilgiler şirket çalışanlarından temin edilmiştir. Dağıtım noktaları adreslerine göre Google Earth, Google Inc.¹ ve İstanbul Şehir Rehberi² aracılığıyla bulunmuş ve depo dahil 59 farklı nokta Google Inc. üzerinde işaretlenmiş GPS koordinatları tespit edilmiştir. Alınan bilgilere göre harita üzerindeki yerleri hakkında emin olunamayan dağıtım noktaları bir kez daha firma yetkililerine danışılarak kesinleştirilmiştir.

Dağıtım noktaları ve fabrika lokasyonları arası mesafelerin bulunması için ilk etapta “The Geographic Distance Matrix Generator³” yazılımı kullanılmıştır. Yazılım noktalar arası mesafeyi kuş uçuşu mesafe yani iki lokasyon arası en kısa yol olarak

¹ <https://maps.google.com/>

² <http://sehirrehberi.ibb.gov.tr/map.aspx>

³ http://biodiversityinformatics.amnh.org/open_source/gdmg/documentation.php

hesaplamaktadır. Bu sebeple herhangi iki nokta arasındaki gidiş ve geliş yollarının uzunluğu eşit olarak hesaplanmakta ve uzaklık matrisi simetrik olmaktadır. Fakat bilindiği gibi İstanbul gibi büyük bir şehirde gidiş ve geliş yolları mesafe olarak bire bir değildir. Bu sebeple tezimizde kullanılmak üzere iki nokta arası eğrisel uzunluklar Google Inc. üzerinden tek tek hesaplanmış ve uzaklık matrisleri oluşturulmuştur. Uzaklık matrisleri Ek-A, Ek-B, Ek-C, Ek-D ve Ek-E de yer almaktadır.

4.5 Mevcut ve Elde Edilen Araç Rotaların Karşılaştırması

Birinci ve ikinci araçların gün bazında sahip oldukları müşteri kümeleri lokasyon olarak birbirlerine yakın olmadıklarından ayrı ayrı ele alınmıştır. Yalnız Çarşamba&Cumartesi günleri yapılan dağıtım kümelerinde yakın mevkiler yer aldığından müşteri kümeleri birleştirilerek model uygulanmıştır. Mağaza adları, bulundukları il/ilçeler ve müşteri talepleri aşağıda verilmektedir. Rota kurulumlarında depo "0" ve " $N + 1$ " düğümleri ile belirtilmiştir. Mevcut rotalar ve model sonucu elde edilen rotalar aşağıda verilmiştir.

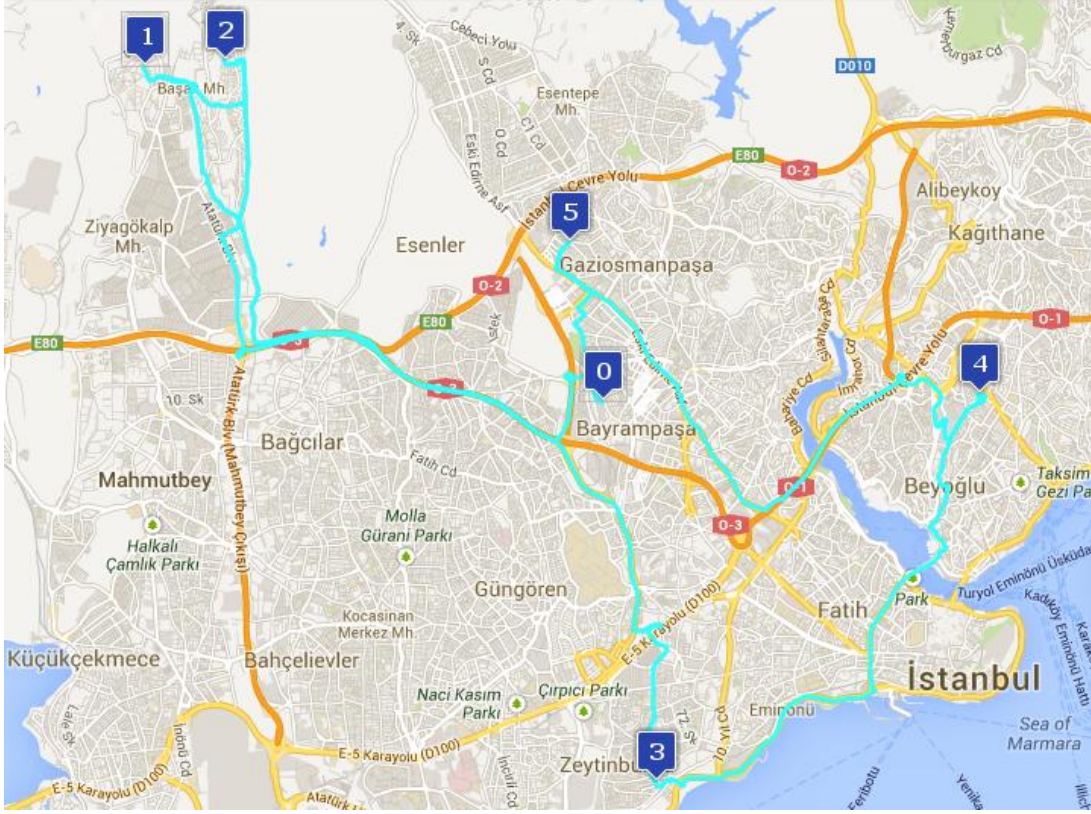
Pazartesi&Cuma (8-12 Şubat 2013) (1. Araç) (İstanbul Servisi)

Çizelge 4. 1 Pazartesi&Cuma günleri müşteri bilgileri

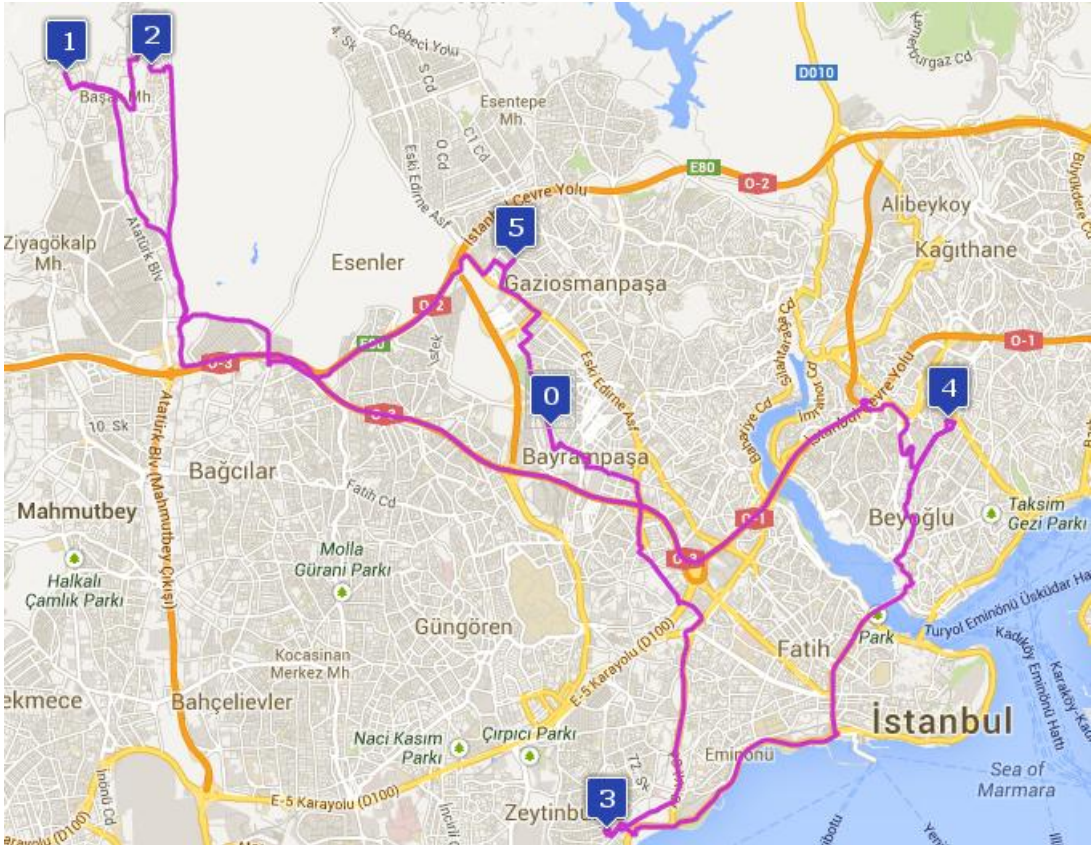
No	Mağaza Adı	İlçe	Talepler (30'luk karton)
1	Başakşehir	Başakşehir	350
2	Başakpazar	Başakpazar	150
3	Zeytinburnu	Zeytinburnu	150
4	Feriköy	Feriköy	200
5	Gaziosmanpaşa	Gaziosmanpaşa	150

Çizelge 4. 2 Pazartesi&Cuma günleri rota karşılaştırması

	İzlenecek Rota	Mesafe (m)	İyileştirme
Yıldırım Yumurtacılık Şirketi	0 → 1 → 2 → 3 → 4 → 5 → 6	71610	%4.349
Model ile Elde Edilen Rota	0 → 3 → 4 → 2 → 1 → 5 → 6	68500	



Şekil 4. 5 Pazartesi&Cuma günleri mevcut rota



Şekil 4. 6 Pazartesi&Cuma günleri elde edilen rota

Pazartesi&Cuma günleri içerisinde hizmet sunulan müşteri bilgileri Çizelge 4.1'de verilmektedir. Yıldırım Yumurta'nın mevcut rotası ile modelden elde edilen rota karşılaştırıldığında gün bazında 3110 m mesafe farkının ortaya çıktığı görülmektedir. İki gün olarak düşünüldüğünde toplam mesafe kısalımı 6220 m dir. Pazartesi&Cuma günlerine ait Çizelge 4.2'deki iki rota arasında %4.349 gibi bir iyileştirme mevcuttur.

Salı (9 Şubat 2013) (2. Araç) (İstanbul-İzmit Servisi)

Çizelge 4. 3 Salı günü müşteri bilgileri

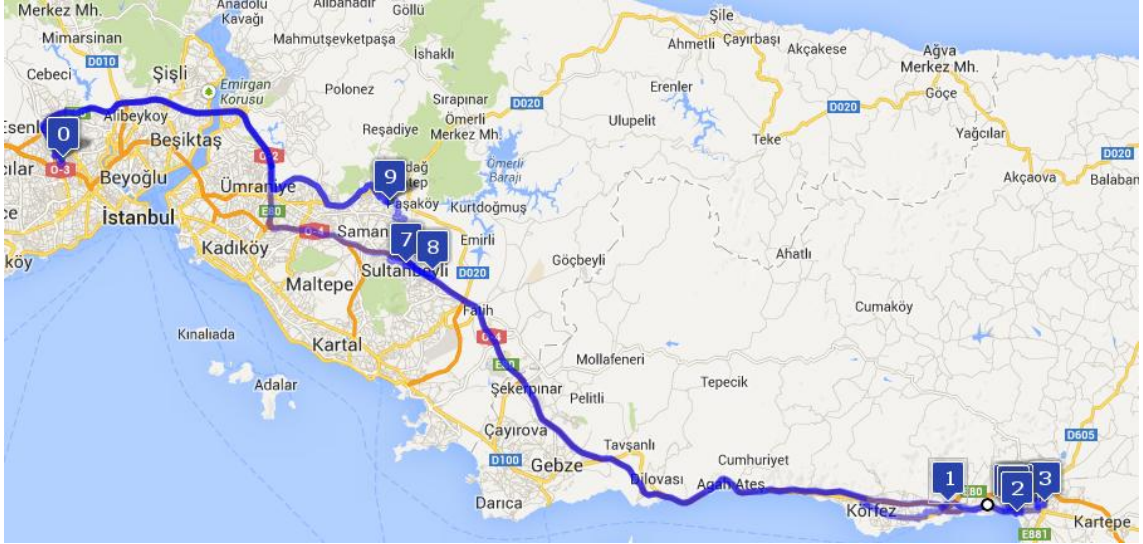
No	Mağaza Adı	İl	Talepler (30'luk karton)
1	Derince	İzmit	150
2	Ncty	İzmit	120
3	Yahya Kaptan	İzmit	130
4	Çukurbağ	İzmit	100
5	Çarşı	İzmit	100
6	Karabaş	İzmit	100
7	Sultanbeyli	İstanbul	100
8	Mecidiye Mah.	İstanbul	60
9	Yenidoğan	İstanbul	70

Çizelge 4. 4 Salı günü rota karşılaştırması

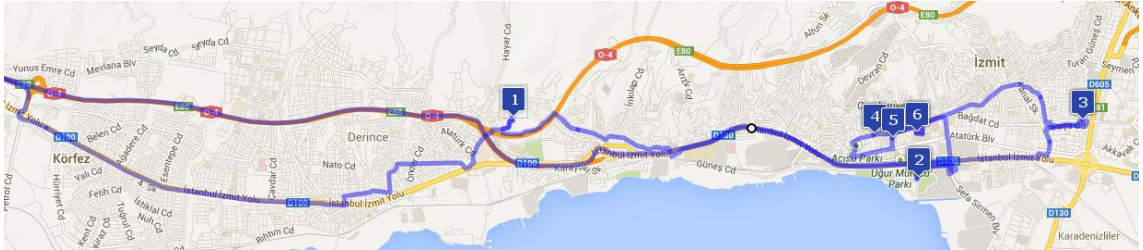
	İzlenecek Rota	Mesafe (m)	İyileştirme
Yıldırım Yumurtacılık Şirketi	0 → 1 → 2 → 3 → 4 → 5 → 6 → 7 → 8 → 9 → 10	267022	%4.207
Model ile Elde Edilen Rota	0 → 9 → 7 → 8 → 1 → 5 → 3 → 2 → 6 → 4 → 10	255789	

Yukarıda Çizelge 4.3'de verilmiş mağaza bilgilerine göre mevcut rota ile elde edilen rota karşılaştırması sonucu gidilen mesafenin 11233 m kısaldığı görülmektedir. Yani Salı gününe ait rota mesafesi yaklaşık %4.207 iyileştirilmiştir. Aşağıda tüm rota içerisindeki

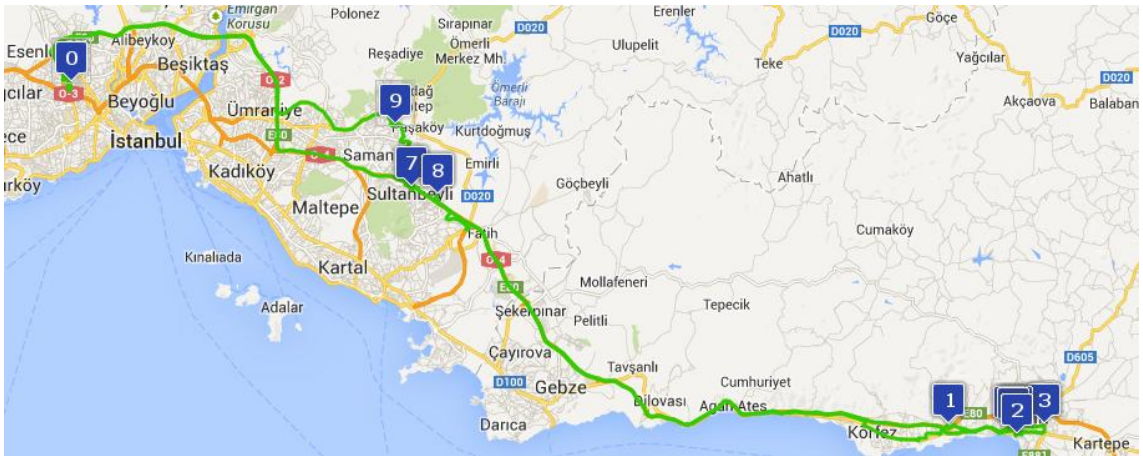
İzmit bölgesi ayrı olarak haritalanmıştır. Çizelge 4.4'deki her iki rotaya ait 7, 8 ve 9 nolu müşteriler arası kullanılan yollarda bir değişiklik olmadığından ayrı olarak gösterilmemiştir.



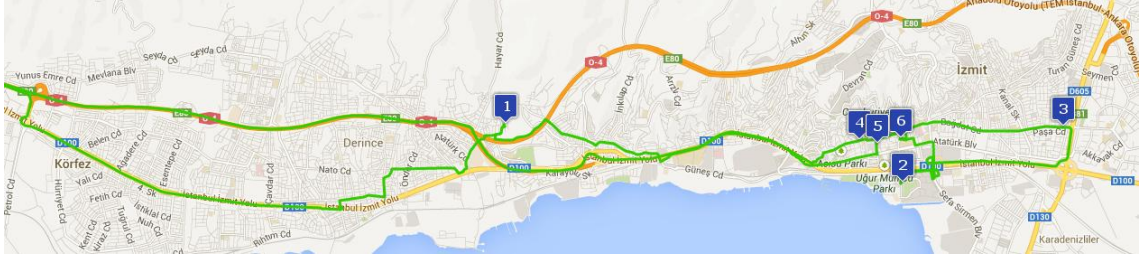
Şekil 4. 7 Salı günü mevcut rota



Şekil 4. 8 Salı günü İzmit bölgesi mevcut rota



Şekil 4. 9 Salı günü elde edilen rota



Şekil 4. 10 Salı günü İzmit bölgesi elde edilen rota

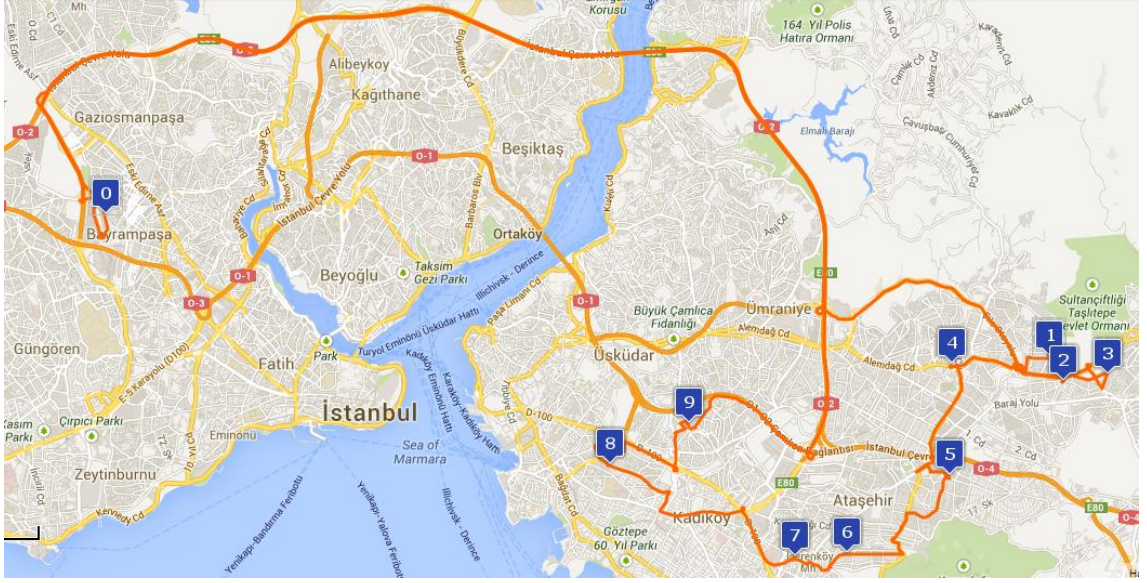
Perşembe (11 Şubat 2013) (2. Araç) (İstanbul Servisi)

Çizelge 4. 5 Perşembe günü müşteri bilgileri

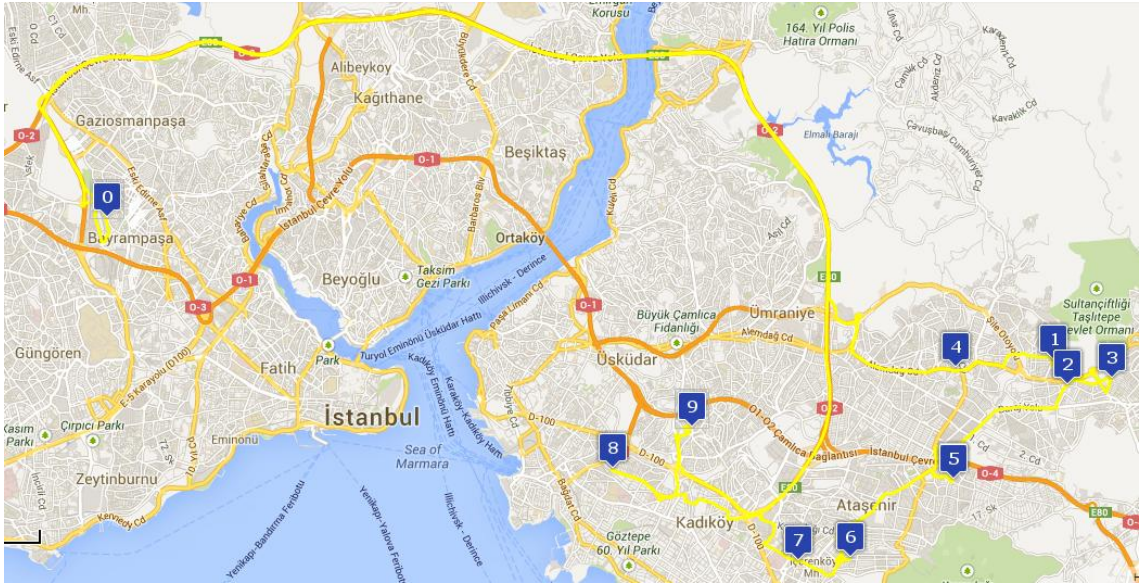
No	Mağaza Adı	İlçe	Talepler (30'luk karton)
1	Çekmeköy	Çekmeköy	150
2	Sancaktepe	Sancaktepe	120
3	Sarıgazi	Sancaktepe	130
4	Dudullu	Ümraniye	100
5	Yeni Çamlıca	Ataşehir	100
6	Fındıklı	Maltepe	100
7	İçerenköy	Kadıköy	100
8	Fikirtepe	Kadıköy	60
9	Fetih Mah.	Ataşehir	70

Çizelge 4. 6 Perşembe günü rota karşılaştırması

	İzlenecek Rota	Mesafe (m)	İyileştirme
Yıldırım Yumurtacılık Şirketi	0 → 1 → 2 → 3 → 4 → 5 → 6 → 7 → 8 → 9 → 10	115821	%5.741
Model ile Elde Edilen Rota	0 → 9 → 8 → 7 → 6 → 5 → 2 → 3 → 1 → 4 → 10	109172	



Şekil 4. 11 Perşembe günü mevcut rota



Şekil 4. 12 Perşembe günü elde edilen rota

Perşembe günü içerisinde hizmet sunulan müşteri bilgileri Çizelge 4.5'de verilmektedir. Mevcut rota ile modelden elde edilen rota karşılaştırıldığında 6649 m mesafe farkının ortaya çıktığı görülmektedir. Perşembe gününe ait Çizelge 4.5'deki iki rota arasında yaklaşık %5.741 gibi bir iyileştirme mevcuttur.

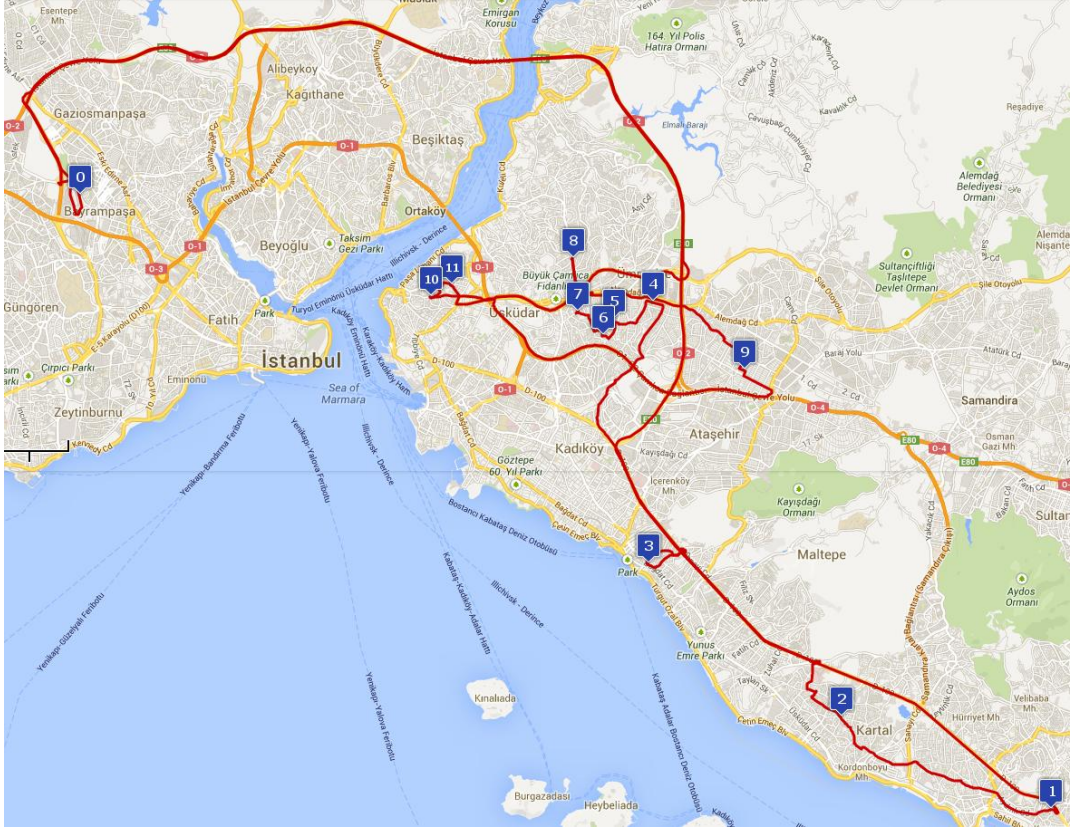
Çizelge 4. 7 Cuma günü müşteri bilgileri

No	Mağaza Adı	İlçe	Talepler (30'luk karton)
1	Kaynarca	Pendik	80
2	Kartal	Kartal	130
3	Küçükyalı	Maltepe	100
4	Atakent	Ümraniye	70
5	Namıkkemal	Ümraniye	100
6	Yunusemre	Ümraniye	100
7	Bulgurlu	Üsküdar	120
8	Çengelköy	Üsküdar	100
9	Şerifali	Ümraniye	100
10	Selami Ali	Üsküdar	50
11	Fıstıkağacı	Üsküdar	50

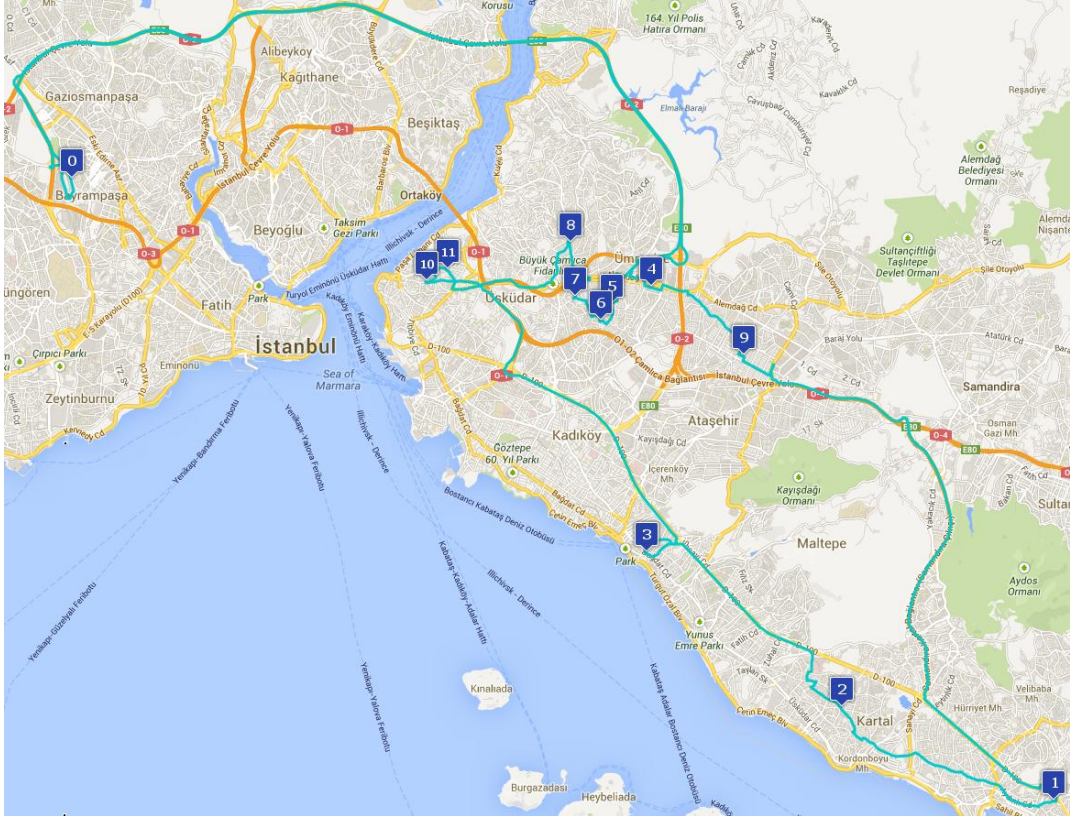
Çizelge 4. 8 Cuma günü rota karşılaştırması

	İzlenecek Rota	Mesafe (m)	İyileştirme
Yıldırım Yumurtacılık Şirketi	0 → 1 → 2 → 3 → 4 → 5 → 6 → 7 → 8 → 9 → 10 → 11 → 12	157016	%10.656
Model ile Elde Edilen Rota	0 → 5 → 6 → 7 → 8 → 10 → 11 → 3 → 2 → 1 → 9 → 4 → 12	140285	

Yukarıda Çizelge 4.7'de verilmiş bilgilere göre Çizelge 4.8'deki Cuma gününe ait rotalar karşılaştırıldığında gidilen mesafenin 16731 m kısaldığı görülmektedir. Yani Cuma gününe ait rota mesafesi yaklaşık %10.656 iyileştirilmiştir. Aşağıda her iki rota ile ilgili şekiller verilmiştir.



Şekil 4. 13 Cuma günü mevcut rota



Şekil 4. 14 Cuma günü elde edilen rota

Çarşamba&Cumartesi (10-13 Şubat 2013) (1. ve 2. Araç) (İstanbul Servisi)

Çizelge 4. 9 Çarşamba&Cumartesi günleri müşteri bilgileri

No	Mağaza Adı	İlçe	Talepler (30'luk karton)
1	Kazımkarabekir	Ümraniye	30
2	Atatürk	Ümraniye	50
3	FSM	Ümraniye	70
4	Çarşı	Ümraniye	100
5	Mehmetakif	Üsküdar	30
6	Selimiye	Üsküdar	40
7	İstiklal	Ümraniye	80
8	İtfaiye	Ümraniye	70
9	Tavukçuyolu	Ümraniye	100
10	Doğanevler	Ümraniye	30
11	İhlamurkuyu	Ümraniye	30
12	Dudullu	Ümraniye	70
13	Taşdelen	Çekmeköy	70
14	Ortaçeşme	Beykoz	120
15	Kavacık	Beykoz	30
16	Zümrütevler	Maltepe	50
17	Cevizli	Kartal	40
18	Gümüşpınar	Kartal	40
19	Kurfali	Kartal	60
20	Tuzla	Tuzla	50
21	Ahmetyesevi	Pendik	70
22	Esenler Mah.	Pendik	70

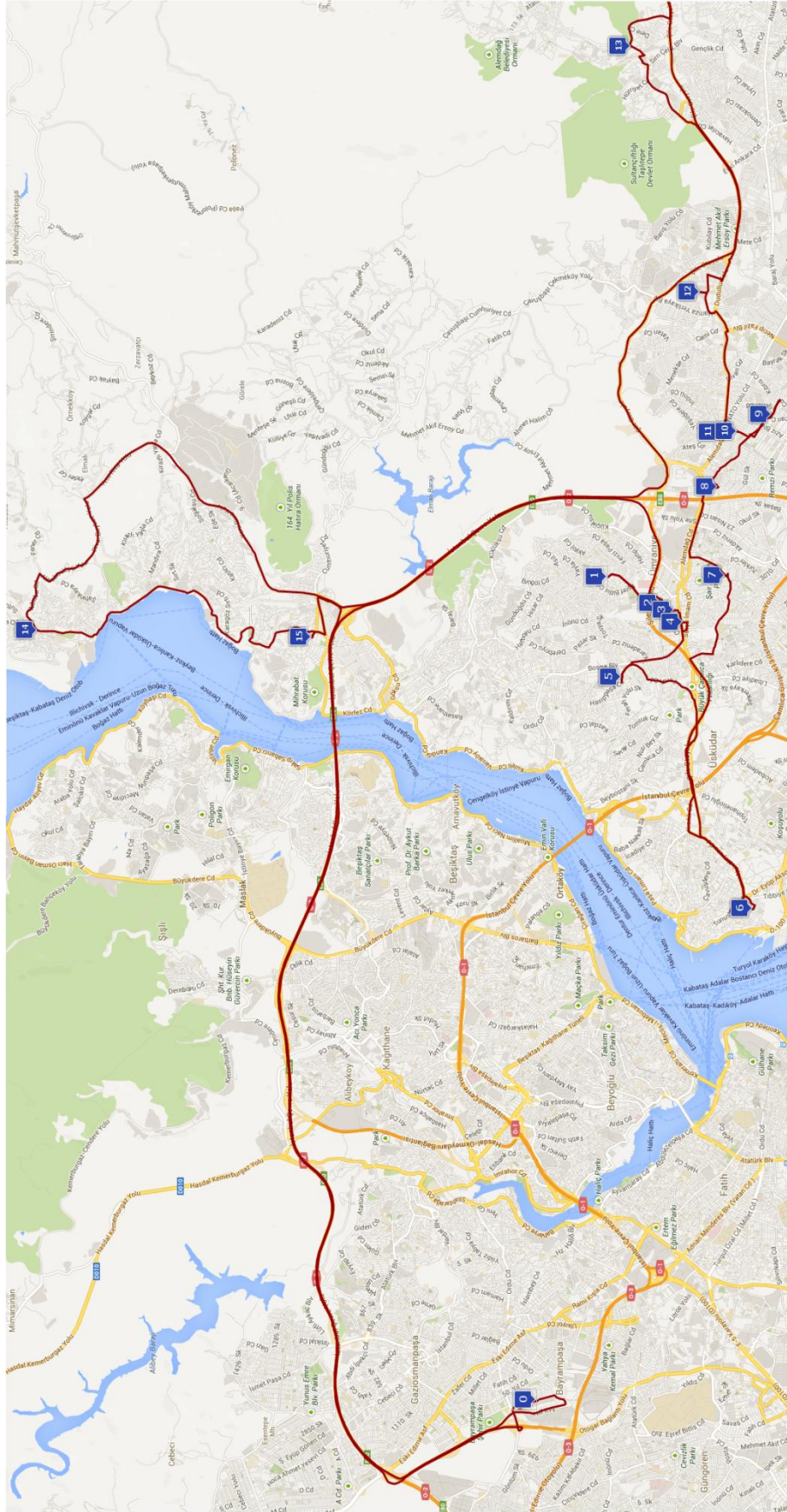
Çizelge 4. 9 Çarşamba&Cumartesi günleri müşteri bilgileri (devam)

23	İnönü Mah.	Kayışdağı	50
24	Kayışdağı	Kayışdağı	80

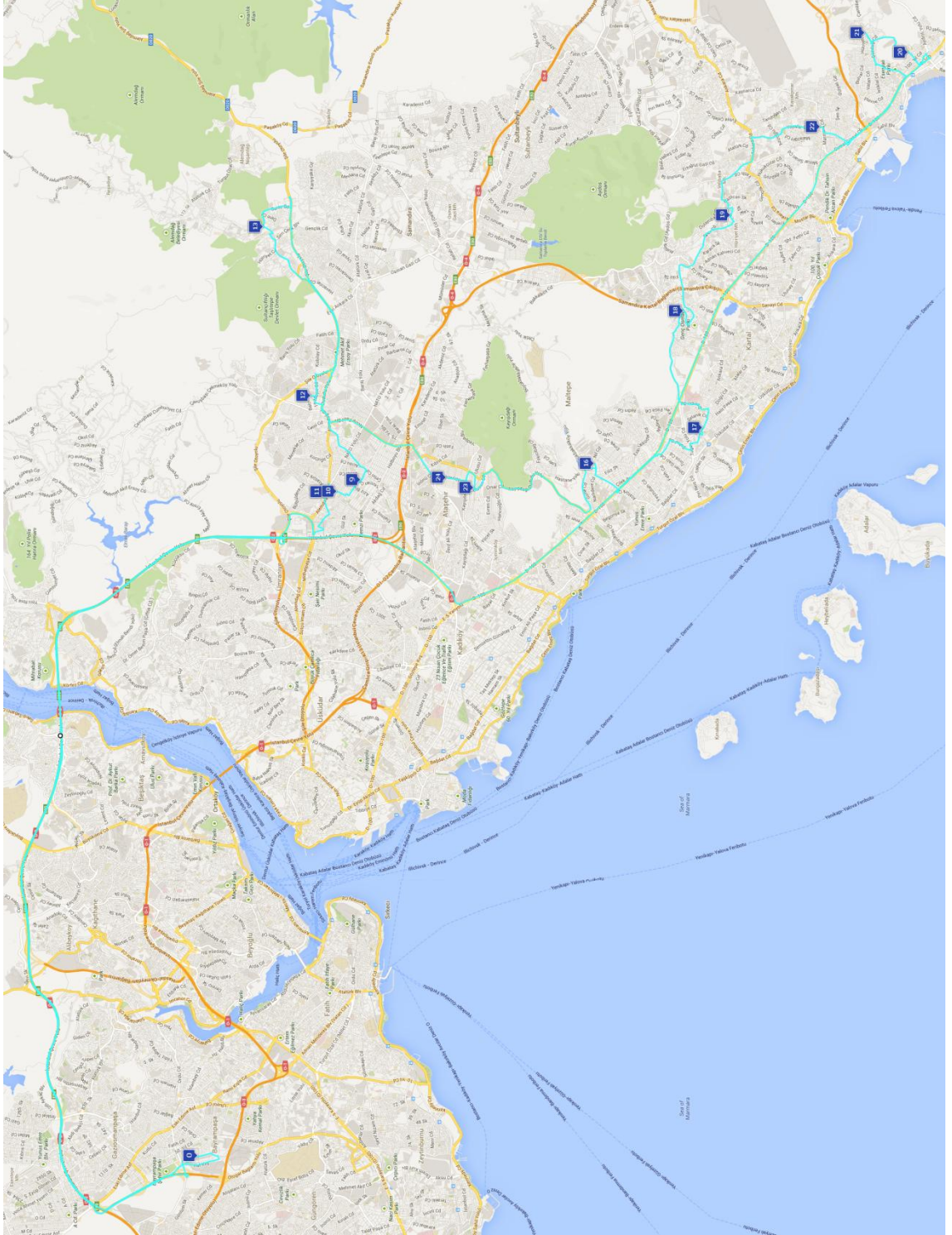
Çizelge 4. 10 Çarşamba&Cumartesi günleri rota karşılaştırması

	Araç	İzlenecek Rota	Mesafe (m)	İyileştirme
Yıldırım Yumurtacılık Şirketi	1. Araç	0 → 1 → 2 → 3 → 4 → 5 → 6 → 7 → 8 → 9 → 10 → 11 → 12 → 13 → 14 → 15 → 25	143030	%11.645
	2. Araç	0 → 16 → 17 → 18 → 19 → 20 → 21 → 22 → 23 → 24 → 25	141286	
Model ile Elde Edilen Rota	1. Araç	0 → 8 → 11 → 10 → 9 → 13 → 12 → 24 → 23 → 16 → 17 → 18 → 19 → 22 → 20 → 21 → 25	132898	
	2. Araç	0 → 15 → 3 → 4 → 5 → 6 → 7 → 2 → 1 → 14 → 25	118310	

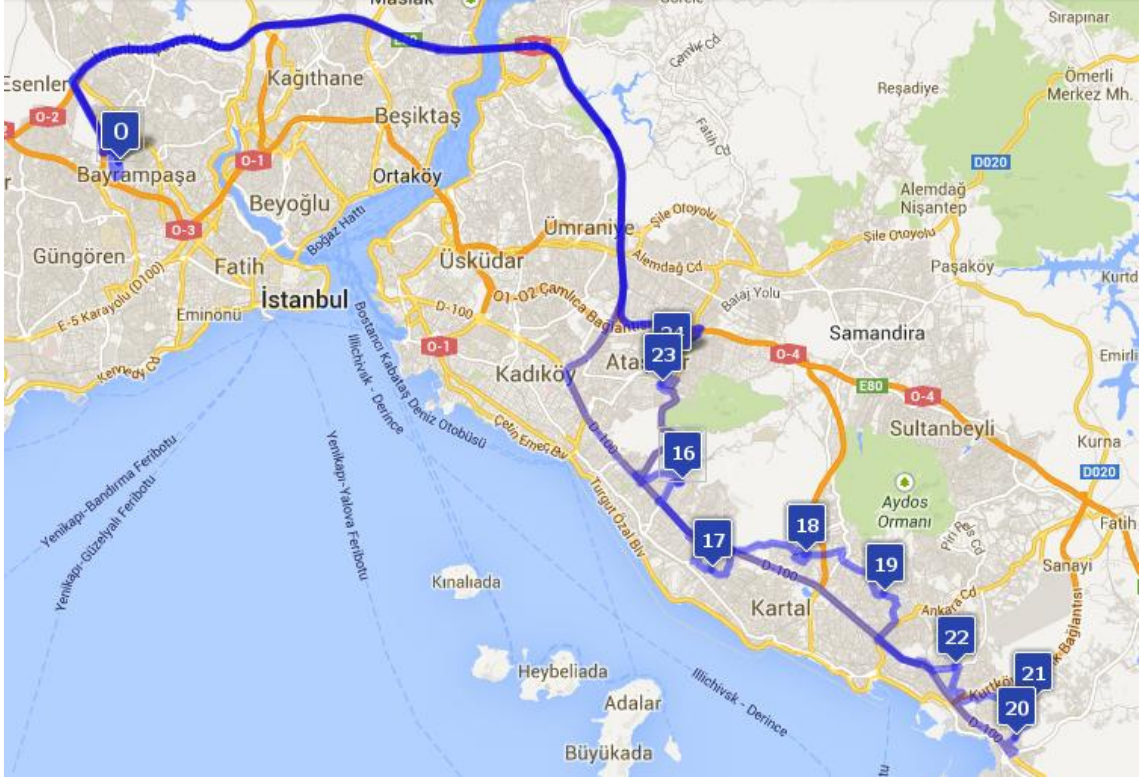
Çarşamba&Cumartesi günleri içerisinde hizmet sunulan müşteri bilgileri Çizelge 4.9'da verilmektedir. Mevcut rota ile modelden elde edilen rota karşılaştırıldığında 1. araçta 10132 m mesafe 2. araçta ise 22976 m mesafe farkının ortaya çıktığı görülmektedir. Çarşamba&Cumartesi günlerine ait Çizelge 4.10'daki iki rota arasında 1. araç için yaklaşık %7.084, 2. araç için ise yaklaşık %17.28 gibi bir iyileştirme oranı mevcuttur. Yani Çarşamba&Cumartesi günleri için toplamda %11.645 gibi bir iyileştirme oranı ortaya çıkmaktadır.



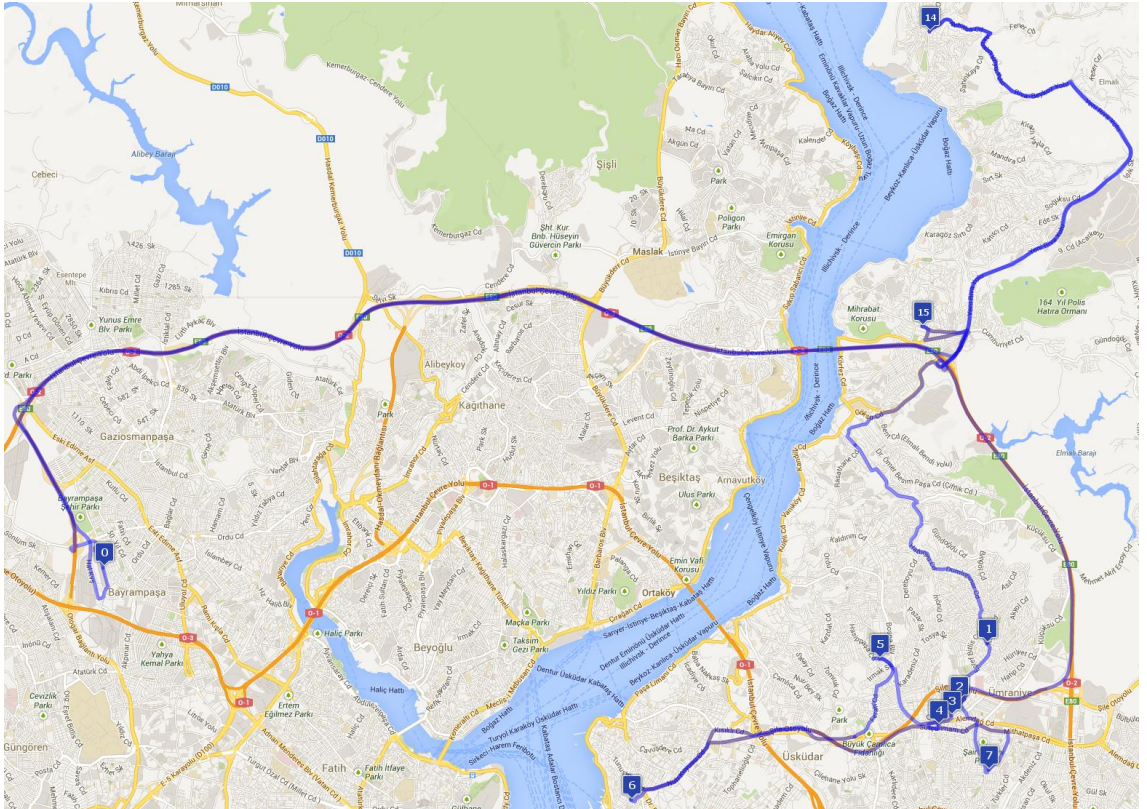
Şekil 4. 15 Çarşamba&Cumartesi günleri 1. aracın mevcut rotası



Şekil 4. 16 Çarşamba&Cumartesi günleri 1. aracın elde edilen rotası



Şekil 4. 17 Çarşamba&Cumartesi günleri 2. aracın mevcut rotası



Şekil 4. 18 Çarşamba&Cumartesi günleri 2. aracın elde edilen rotası

SONUÇ VE ÖNERİLER

Günümüz rekabet ortamında firmaların rakipleri ile yarışabilmeleri ve ayakta kalabilmeleri için dikkat etmeleri gereken konulardan birisi de lojistikte önemli karar alanlarından biri olan ürün dağıtımıdır. Bu çerçevede firmaların dağıtım ağlarını en optimal şekilde kurmaları gerekmektedir. Rota optimizasyonunu hedef alan problemlerin önde gelenlerinden birisi araç rotalama problemidir.

Tipik araç rotalama probleminin amacı, bir araç filosu için minimum maliyetli rota kümesini tasarlamaktır. Araç rotaları genellikle başladıkları depoda son bulmaktadır. Rotalar oluşturulurken problem çeşidine göre araç kapasitesi, mesafe uzunluğu, zaman, dağıtım toplama sırası vb. kısıtlar dikkate alınmaktadır.

Bu bağlamda tezimizde araç rotalama problemi ve türleri, şebeke yapıları ve literatürdeki mevcut modeller üzerinden ayrıntılı şekilde incelenmiştir. Sonrasında araç rotalama problem çözüm yöntemleri hakkında genel bilgiler verilmiştir.

Tezimizin uygulama bölümünde tamsayıli lineer matematiksel bir model kurulmuş İstanbul da gıda sektöründe faaliyet gösteren bir firmanın verileri üzerinde kullanılmıştır. Model bilgisayar ortamında GAMS 24.1.3 programlama dili kullanılarak yazılmıştır. Elde edilen sonuçlar ile mevcut durum karşılaştırılmıştır.

Karşılaştırma sonucu tek aracın kullanıldığı Pazartesi&Cuma rotasında 5 müşteri üzerinden %4.349, 9 müşteriye hizmet verilen Salı rotasında %4.207, Perşembe günü ürün dağıtımı yapılan 9 müşteriye ait rotada %5.741, 11 müşterinin bulunduğu Cuma gününe ait rotada ise %10.656 iyileştirme yapıldığı görülmüştür. Ayrıca iki aracın müşteri kümelerinin birleştirildiği ve toplamda 24 müşteriye sahip Çarşamba&Cumartesi rotasında %11.645 iyileştirme yapılmıştır. Bir haftalık araçlara ait

rotalar incelendiğinde mevcut toplam rota uzunluğunun 1,251,711 m olduğu, buna karşılık model sonucu elde edilen rota uzunluğunun 1,144,662 m olduğu görülmüştür. Buradan anlaşılabileceği gibi haftalık bazda rota uzunluğunda %8.55 iyileştirme yapılmıştır. Rota uzunluğundaki iyileştirme aylık ve yıllık olarak düşünüldüğünde rota uzunluğunda sırasıyla 463,879 ve 5,566,548 m. kazanç sağlandığı görülmektedir. Ülkemizdeki benzin fiyatları göz önüne alındığında bu iyileştirme oranlarının rota kısaltımında makul seviyede olduğu görülmektedir. Ayrıca gidilen mesafenin kısaltılması ile çevre kirliliği vb. etkenlerinde azaltılabildiği göz önüne alınmalıdır.

Elde edilen sonuçlar ve çalışma boyunca kazanılan tecrübelerle göre gelecek çalışmalar için meta sezgisel yöntemler kullanılarak sonuçlar karşılaştırılabilir, anlık olarak ortaya çıkan müşteri siparişlerinin rotaya dahil edilmesi üzerine çalışmalar yapılabilir veya zaman kısıdı vb. ek kısıtların eklenmesi ile problem farklı bir boyuta taşınarak farklı araç rotalama türleri üzerinden elde edilebilecek yeni rotalar incelenebilir.

KAYNAKLAR

- [1]. Dantzig, G.B. ve Ramser, J.H. (1959). "The truck dispatching problem", *Management science*, 6(1): 80-91.
- [2]. Golden, B., Raghavan, S. ve Wasil E. (2008). *The vehicle routing problem: latest advances and new challenges*, Springer, New York.
- [3]. Eilon, S., Watson-Gandy, C., ve Christofides, N., (1971). *Distribution Management: Mathematical modelling and practical analysis*, Griffin, London.
- [4]. Toth, P. ve Vigo, D. (2002). *The vehicle routing problem*. The United States of America, Philadelphia.
- [5]. Marinakis, Y., ve Migdalas, A. (2007). "Annotated bibliography in vehicle routing", *Operational Research*, 7(1): 27-46.
- [6]. Laporte, G., (1992). "The vehicle routing problem: An overview of exact and approximate algorithms", *European Journal of Operational Research*, 59: 345-358.
- [7]. Laporte G., ve Osman I.H., (1995). "Routing Problems: A bibliography", *Annals of Operations Research*, 61(1): 227-262.
- [8]. Cordeau, J. F., Gendreau, M., Laporte, G., Potvin, J. Y. ve Semet, F., (2002). "A guide to vehicle routing heuristics", *Journal of the Operational Research society*, 53(5): 512-522.
- [9]. Cordeau, J.F., Gendreau, M., Hertz, A., Laporte, G., ve Sormany, J.S., (2005). "New Heuristics for the Vehicle Routing Problem", Langevine, A. ve Riopel, D., (Eds.), *Logistics Systems: Design and Optimization*, Wiley and Sons, 279-298.
- [10]. Ekşioğlu, B., Vural, A.V. ve Reisman, A., (2009) .“The vehicle routing problem: A taxonomic review”, *Computers & Industrial Engineering* 57: 1472–1483.
- [11]. Potvin, J.Y., (2009). "State-of-the art review-evolutionary algorithms for vehicle routing", *INFORMS Journal on Computing*, 21(4): 518-548.
- [12]. Christofides N., Mignozzi, A. ve Toth, P., (1981). "Exact Algorithms for the Vehicle Routing Problem Based on Spanning Tree and Shortest Path Relaxation", *Mathematical Programming*, 19: 255-282.

- [13]. Laporte, G., Nobert, Y. ve Taillefer, S., (1987). "A branch-and-bound algorithm for the asymmetrical distance-constrained vehicle routing problem", *Mathematical Modelling*, 9(12): 857-868.
- [14]. Hadjiconstantinou, E., Christofides, N. ve Mingozzi, A., (1995). "A new exact algorithm for the vehicle routing problem based on q -paths and k -shortest paths relaxations", *Annals of Operations Research*, 61(1): 21-43.
- [15]. Eryavuz, M., ve Gencer, C., (2001). "Araç rotalama problemine ait bir uygulama", *Süleyman Demirel Üniversitesi BF Dergisi*, 6(1): 139-155.
- [16]. Toth, P. ve Vigo, D., (2002). "Models, relaxations and exact approaches for the capacitated vehicle routing problem", *Discrete Applied Mathematics*, 23: 487-512.
- [17]. Ralphs, T. K., Kopman, L., Pulleyblank, W. R., ve Trotter, L. E., (2003). "On the capacitated vehicle routing problem", *Mathematical Programming*, 94: 343-359.
- [18]. Lysgaard, J., Letchford, A. N., & Eglese, R. W., (2004). "A new branch-and-cut algorithm for the capacitated vehicle routing problem", *Mathematical Programming*, 100(2): 423-445.
- [19]. Başkaya, Z., ve Öztürk, B. A., (2005). "Dal Kesme Yöntemi Ve Bir Ekmek Fabrikasında Oluşturulan Araç Rotalama Problemine Uygulanması", *Uludağ Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, 24(1): 101-114.
- [20]. Fukasawa, R., Longo, H., Lysgaard, J., de Aragão, M. P., Reis, M., Uchoa, E. ve Werneck, R. F., (2006). "Robust branch-and-cut-and-price for the capacitated vehicle routing problem", *Mathematical programming*, 106(3): 491-511.
- [21]. Baldacci, R., Christofides, N., ve Mingozzi, A., (2008). "An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts", *Mathematical Programming*, 115(2): 351-385.
- [22]. Güvez, H., Dege, M. ve Eren, T., (2012). "Kırıkkale’de Araç Rotalama Problemi İle Tıbbi Atıkların Toplanması", *International Journal of Engineering*, 4(1): 41-45.
- [23]. Matai, R., Singh, S. P. ve Mittal, M.L., (2010). "Traveling Salesman Problem: an Overview of Applications, Formulations, and Solution Approaches", *Traveling Salesman Problem, Theory and Applications*, InTech.
- [24]. Ahuja, R. K., Magnanti, T. L., Orlin, J. B. (1993). *Network flows: theory, algorithms, and applications*, Prentice-Hall, Inc. Upper Saddle River, New Jersey.
- [25]. Laporte, G., (1992). "The traveling salesman problem: An overview of exact and approximate algorithms", *European Journal of Operational Research*, 59: 231-247.
- [26]. Floudas, C. A. ve Pardalos, P. M., (2008). *Encyclopedia of optimization*, Second Edition, Springer, New York.

- [27]. Balakrishnan, N., (1993). "Simple Heuristics for the Vehicle Routing Problem with Soft Time Windows", *Journal of the Operational Research Society*, 44(3): 279-287.
- [28]. Rego, C., (2001). "Node-ejection chains for the vehicle routing problem: Sequential and parallel algorithms", *Parallel Computing* 27: 201-222.
- [29]. Machado, P., Tavares, J., Pereira, F. B. ve Costa, E. (2002). "Vehicle Routing Problem: Doing It The Evolutionary Way", *GECCO*, 690-696.
- [30]. Özkan, P., (2006). Araç rotalama ve çizelgeleme, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul.
- [31]. Caccetta, L. ve Hill, S. P., (2001). "Branch and cut methods for network optimization", *Mathematical and computer modelling*, 33(4): 517-532.
- [32]. Şeker, Ş., (2007). Araç Rotalama Problemleri ve Zaman Pencereli Stokastik Araç Rotalama Problemine Genetik Algoritma Yaklaşımı, Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul.
- [33]. Yeun, L. C., Ismail, W. R., Omar, K. ve Zirour, M., (2008). "Vehicle routing problem: models and solutions", *Journal of Quality Measurement and Analysis*, 4(1): 205-218.
- [34]. Çalışkan, K., (2011). Karınca Kolonisi Optimizasyonu İle Araç Rotalama Probleminin Maliyetlerinin Kümeleme Tekniği İle İyileştirilmesi, Yüksek Lisans Tezi, TOBB Ekonomi ve Teknoloji Üniversitesi Fen Bilimleri Enstitüsü, Ankara.
- [35]. Bowerman, R., Hall, B. ve Calamai, P., (1995). "A multi-objective optimization approach to urban school bus routing: Formulation and solution method", *Transportation Research Part A: Policy and Practice* 29(2): 107-123.
- [36]. Tüfekçier, H., (2008). İki Amaçlı Açık Araç Rotalama Problemi İçin Bir Çözüm Yaklaşımı, Yüksek Lisans Tezi, Eskişehir Osmangazi Üniversitesi Fen Bilimleri Enstitüsü, Eskişehir.
- [37]. Düzakın, E. ve Demircioğlu, M., (2009). "Araç Rotalama Problemleri ve Çözüm Yöntemleri", *Çukurova Üniversitesi İİBF Dergisi* 13(1):68-87.
- [38]. Crainic, T.G. ve Laporte, G., (1997). "Planning models for freight transportation", *European Journal of Operational Research*, 97: 409-438.
- [39]. Yurtkuran, A., (2009). Araç Rotalama Problemlerinin Çözümü İçin Yeni Bir Meta-Sezgisel Yaklaşım Elektromanyetik Algoritma, Yüksek Lisans Tezi, Uludağ Üniversitesi Fen Bilimleri Enstitüsü, Bursa.
- [40]. Augerat, P., Belenguer, J.M., Benavent, E., Corberan, A. ve Naddef, D., (1998). "Separating capacity constraints in the CVRP using tabu search", *European Journal of Operational Research* 106(2): 546-557.
- [41]. Zachariadis, E. E. ve Kiranoudis, C.T., (2012). "An effective local search approach for the Vehicle Routing Problem with Backhauls", *Expert Systems with Applications*, 39(3): 3174-3184.

- [42]. Brandao, J. (2006). "A new tabu search algorithm for the vehicle routing problem with backhauls", *European Journal of Operational Research*, 173(2): 540-555.
- [43]. Tavakkoli-Moghaddam, R., Saremi, A.R. ve Ziaee, M.S. (2006). "A memetic algorithm for a vehicle routing problem with backhauls", *Applied Mathematics and Computation*, 181(2): 1049-1060.
- [44]. Cordeau, J.F., Laporte, G., Savelsbergh, M.W.P., Vigo, D., (2007). "Chapter 6: Vehicle Routing", *Handbook in OR & MS*, 14: 367-428.
- [45]. Dondo, R., ve Cerdá, J. (2007). "A cluster-based optimization approach for the multi- depot heterogeneous fleet vehicle routing problem with time windows", *European Journal of Operational Research*, 176(3): 1478-1507.
- [46]. Azi, N., Gendreau, M. Ve Jean-Yves Potvin, J.Y. (2007). "An exact algorithm for a single-vehicle routing problem with time windows and multiple routes", *European journal of operational research*, 178(3): 755-766.
- [47]. Tan, K.C., Lee, L.H., Zhu, Q.L., Ou, K. (2001). "Heuristic methods for vehicle routing problem with time windows", *Artificial intelligence in Engineering*, 15(3): 281-295.
- [48]. Nagy, G., ve Salhi, S. (2005). "Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries", *European Journal of Operational Research*, 162(1): 126-141.
- [49]. Montane, F.A.T., ve Galvao, R.D. (2006). "A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service", *Computers and Operations Research*, 33(3): 595-619.
- [50]. Zachariadis, E.E., Tarantilis, C.D. ve Kiranoudis, C.T., (2009). "A hybrid metaheuristic algorithm for the vehicle routing problem with simultaneous delivery and pick-up service", *Expert Systems with applications*, 36(2): 1070-1081.
- [51]. Baykoç, Ö.F. ve İşleyen, S.K. (2007). "Stokastik Talepli Araç Rotalama Problemi İçin Şans Kısıtı Yaklaşımı", *Teknoloji*, 10(1): 31-39.
- [52]. Networking and Emerging Optimization, Split Delivery VRP, <http://neo.lcc.uma.es/vrp/vrp-flavors/split-delivery-vrp/>, 29.08.2013.
- [53]. Kemer, B. (2010). Araç Rotalama Problemlerine Genetik Algoritma Yaklaşımı: Bir Gıda Dağıtım Firması Uygulaması, Yüksek Lisans Tezi, Karadeniz Teknik Üniversitesi Sosyal Bilimleri Enstitüsü, Trabzon.
- [54]. Alonso, F., Alvarez, M.J. ve Beasley, J.E. (2007). "A tabu search algorithm for the periodic vehicle routing problem with multiple vehicle trips and accessibility restrictions", *Journal of the Operational Research Society*, 59(7): 963-976.
- [55]. Keskindürk, T. (2009). Araç Rotalama Problemlerinin Global Karınca Koloni Optimizasyonu ile Çözümü, Doktora Tezi, İstanbul Üniversitesi Sosyal Bilimleri Enstitüsü, İstanbul.

- [56]. Li, F., Golden, B. ve Wasil, E. (2007). "The open vehicle routing problem: Algorithms, large-scale test problems, and computational results", *Computers & Operations Research*, 34(10): 2918-2930.
- [57]. Öztürk, A., (2011). *Yöneylem Araştırması*, 13. Baskı, Ekin basın yayın dağıtım, Bursa.
- [58]. Çetin, E., (2005). "Dinamik Programlama ile Sınır Tenörü Optimizasyonu", *Dogu ve Güneydogu Madenlerinin Degerlendirilmesi Sempozyumu*, 139-143.
- [59]. Ropke, S., (2005), *Heuristic and Exact Algorithms for Vehicle Routing Problems*, Ph.D. thesis, University of Copenhagen, Department of Computer Science, Denmark.
- [60]. Vigo, D. (1996). "A heuristic algorithm for the asymmetric capacitated vehicle routing problem", *European Journal of Operational Research*, 89: 108-126.
- [61]. Ryan, D.M., Hjorring, C. ve Glover, F. (1993). "Extensions of the petal method for vehicle routing", *Journal of Operational Research Society*, 44(3): 289–296.
- [62]. Renaud, J., Boctor, F.F. ve Laporte, G. (1996). "An improved petal heuristic for the vehicle routing problem", *Journal of the Operational Research Society*, 47: 329–336.
- [63]. *Networking and Emerging Optimization, Route-First Cluster-Second Method*, <http://neo.lcc.uma.es/vrp/solution-methods/heuristics/route-first-cluster-second-method/>, 29.08.2013.
- [64]. Lin, S. (1965). "Computer solutions of the travelling salesman problem". *Bell System Technical Journal*, 44: 2245–2269.
- [65]. Glover, F. (1986). "Future paths for integer programming and links to artificial intelligence", *Computers & Operations Research*, 13(5): 533–549.
- [66]. Breedam, A.V. (2001). "Comparing descent heuristics and metaheuristics for the vehicle routing problem", *Computers & Operations Research*, 28(4): 289-315.
- [67]. Çiçekdeş, B., (2011). "A Genetic Algorithm Approach For A Real Life Heterogeneous Capacitated Vehicle Routing Problem", Master thesis, Dokuz Eylül University, Graduate School Of Natural and Applied Sciences, İzmir.
- [68]. Kirckpatrick, S., Gelatt C.D. ve Vecchi M.P. (1983). "Optimization by simulated annealing", *Science* 220(4598): 671-80.
- [69]. Cerny V. (1985). "A thermodynamical approach to the travelling salesman problem: an efficient simulation algorithm", *Journal of Optimization Theory Application*, 45(1): 41-51.
- [70]. Glover, F. (1989). "Tabu Search I", *ORSA J Comput*, 1(3): 190–206.
- [71]. Glover, F. (1990). "Tabu Search II", *ORSA J Comput*, 2(1): 4–32.
- [72]. Tarantilis, C.D., Ioannou, G. ve Prastacos, G. (2005). "Advanced vehicle routing algorithms for complex operations management problems", *Journal of Food Engineering* 70(3): 455-471.

- [73]. Colorni, A., Dorigo, M. ve Maniezzo, V. (1991). "Distributed Optimization by Ant Colonies", First European Conference on Artificial Life, 134-142.
- [74]. Mazzeo, S., ve Loiseau, I. (2004). "An ant colony algorithm for the capacitated vehicle routing", Electronic Notes in Discrete Mathematics, 18: 181-186.

PAZARTESİ & CUMA GÜNLERİ UZAKLIK MATRİSİ

	0	1	2	3	4	5
	Depo	Başakşehir	Başakpazar	Zeytinburnu	Feriköy	Gaziosmanpaşa
0	Depo	99999	15400	10614	12121	4528
1	Başakşehir	15960	3669	21703	26252	13896
2	Başakpazar	15645	3404	21642	26191	13415
3	Zeytinburnu	11311	23781	99999	12163	14133
4	Feriköy	12574	25044	12838	99999	14316
5	Gaziosmanpaşa	3970	17609	13643	14776	99999

SALI GÜNÜ UZAKLIK MATRİSİ

	0	1	2	3	4
	Depo	Derince	Ncty	Yahya Kaptan	Çukurbağ
0	Depo	113245	118650	121320	117026
1	Derince	999999	9027	11597	10387
2	Ncty	10863	999999	3741	3058
3	Yahya Kaptan	11815	4220	999999	4170
4	Çukurbağ	7884	3776	4129	999999
5	Çarşı	8222	2487	3088	378
6	Karabaş	10295	2380	2800	1050
7	Sultanbeyli	66595	72950	75620	71326
8	Mecidiye Mah.	63520	69875	72545	68251
9	Yenidoğan	77114	83469	86139	81845

	5		6	7	8	9
		Çarşı	Karabaş	Sultanbeyli	Mecidiye Mah.	Yenidoğan
0	Depo	116990	117862	49827	53563	46179
1	Derince	7267	11223	67311	63494	76539
2	Ncty	2853	2120	74885	71068	84113
3	Yahya Kaptan	4153	3583	75837	72020	85065
4	Çukurbağ	1460	1894	71906	68089	81134
5	Çarşı	999999	853	72244	68427	81472
6	Karabaş	1180	999999	74317	70500	83545
7	Sultanbeyli	71290	72162	999999	3550	8853
8	Mecidiye Mah.	68232	69087	3327	999999	11150
9	Yenidoğan	81826	82681	8706	11542	999999

ÇARŞAMBA & CUMARTESİ GÜNLERİ UZAKLIK MATRİSİ

	0	1	2	3	4
	Depo	Kazımkarabekir	Atatürk	FSM	Çarşı
0	Depo	99999	34098	34450	37190
1	Kazımkarabekir	35010	1508	1810	2663
2	Atatürk	34229	1508	438	1241
3	FSM	34344	1860	99999	848
4	Çarşı	37742	2632	1072	99999
5	Mehmetakif	38880	3683	2720	2493
6	Selimiye	42218	9848	8288	7710
7	İstiklal	35328	3038	1893	1929
8	İttifaiye	33923	4227	3391	3890
9	Tavukçuyolu	36250	6708	5872	6371
10	Doğanevler	34880	5508	4672	5171
11	İhlamurkuyu	33129	6279	4343	4942
12	Dudullu	37690	9750	7889	8398

	5		6	7	8	9	10	11
		Mehmetakif	Selimiye	İstiklal	İtfaie	Tavukçuyolu	Doğanevler	İhlamurkuyu
0	Depo	38041	41423	34910	34494	36083	35050	35250
1	Kazımkarabekir	3446	9666	3684	4099	6574	5519	6550
2	Atatürk	2638	7951	2539	3389	5597	5569	5769
3	FSM	2425	7538	2142	3435	5505	4855	5055
4	Çarşı	2112	7275	2360	3703	5773	5123	5323
5	Mehmetakif	99999	7594	4831	5651	7721	7071	7271
6	Selimiye	7708	99999	9048	12952	12938	13508	13708
7	İstiklal	3979	9192	99999	2670	4740	4090	4290
8	İtfaie	5467	12986	3063	99999	2259	1789	2016
9	Tavukçuyolu	7948	13161	5544	2979	99999	1700	2122
10	Doğanevler	6748	13943	4344	1779	1033	99999	770
11	İhlamurkuyu	6519	13182	4115	2003	1623	590	99999
12	Dudullu	9975	16653	7571	5459	4465	3945	3435

	12	13	14	15	16	17	18
	Dudullu	Taşdelen	Ortaçeşme	Kavacak	Zümrütevler	Cevizli	Gümüşpınar
0	Depo	40040	36675	26460	45410	49141	51079
1	Kazımkarabekir	9393	20809	10014	16960	20691	22629
2	Atatürk	8612	22060	11845	16179	19910	21848
3	FSM	7998	22175	11960	17294	20025	21963
4	Çarşı	8266	25573	15358	23592	27323	22281
5	Mehmetakif	10214	21015	10270	20880	24611	23688
6	Selimiye	16551	30049	19834	15473	19204	21014
7	İstiklal	7233	22999	12784	13356	17087	19675
8	İtfaiye	5056	21854	11639	17273	21004	18252
9	Tavukçuyolu	3633	27481	13966	11810	16191	15279
10	Doğanevler	3623	22811	12596	12893	21961	16362
11	İhlamurkuyu	3663	22050	11835	13700	18281	17119
12	Dudullu	99999	25621	15406	13495	18076	16914

	19	20	21	22	23	24
	Kurfali	Tuzla	Ahmetyesevi	Esenler Mah.	İnönü Mah.	Kayışdağı
0	Depo	56064	62294	61700	58715	40000
1	Kazımkarabekir	27614	33844	33250	30265	11550
2	Atatürk	26833	33063	32469	29484	10769
3	FSM	26948	33178	32584	29599	10884
4	Çarşı	30346	36576	35982	32997	10464
5	Mehmetakif	31534	37764	37170	34185	15470
6	Selimiye	26127	32357	31763	28778	13513
7	İstiklal	24010	30240	29646	26661	7640
8	İtfaiye	27927	34157	33563	30578	6209
9	Tavukçuyolu	21156	28024	27430	24445	3900
10	Doğanevler	22239	29107	30712	25528	4983
11	İhlamurkuyu	28123	34353	33759	30774	5573
12	Dudullu	22791	29659	29065	26080	5585

	0	1	2	3	4	5	6
	Depo	Kazımkarabekir	Atatürk	FSM	Çarşı	Mehmetakif	Selimiye
13	Taşdelen 43500	15660	15238	15590	18330	17850	22563
14	Ortaçeşme 35715	21296	22104	23626	25196	22406	29429
15	Kavacık 27708	10091	12097	12449	15189	10861	19422
16	Zümrütevler 44324	16884	16462	16814	17346	18507	14603
17	Cevizli 49110	21670	21248	21600	22132	23293	19389
18	Gümüşpınar 50568	23128	22706	23058	22543	24851	20947
19	Kurfali 56349	28909	28487	28839	29371	30532	26628
20	Tuzla 60800	34459	32938	33290	32849	34983	31079
21	Ahmetyesevi 64300	34379	35056	35408	34841	36002	32098
22	Esenler Mah. 57388	29948	29526	30977	29437	31571	27667
23	İnönü Mah. 41055	13615	13193	13545	10416	17236	12736
24	Kayışdağı 39930	12490	12118	12420	10880	16111	13240

	7	8	9	10	11	12	13
	İstiklal	İtfaiye	Tavukçuyolu	Doğanevler	İhlamurkuyu	Dudullu	Taşdelen
13	Taşdelen 15346	13284	10795	10175	9665	7030	99999
14	Ortaçeşme 22916	22500	24089	23056	23256	28046	34523
15	Kavacık 12909	12493	14082	13049	13249	18039	24516
16	Zümrütevler 14107	15208	12281	13101	13993	13766	20577
17	Cevizli 18893	19994	16963	20505	18895	18448	22299
18	Gümüşpınar 20351	21452	16103	16923	17694	17538	16352
19	Kurfali 26132	27233	21958	22778	23099	23393	22307
20	Tuzla 31582	31684	26409	27229	28000	27844	26758
21	Ahmetyesevi 31165	33802	27428	28248	29019	28863	30989
22	Esenler Mah. 27171	28272	22997	23817	24588	24432	23346
23	İnönü Mah. 10838	7555	5456	6276	6997	6941	15048
24	Kayışdağı 7657	5984	4120	4940	5661	5605	13712

	14	15	16	17	18	19	20
	Ortaçeşme	Kavacık	Zümrütevler	Cevizli	Gümüşpınar	Kurfali	Tuzla
13	31431	21216	16341	23217	16163	22040	34044
14	99999	7724	33416	37147	39085	44070	50300
15	8088	99999	23409	27140	29078	34063	40293
16	30905	22040	99999	5566	8291	13304	19634
17	35691	26826	5879	99999	4861	10094	16324
18	37149	28284	7885	5129	99999	5368	14166
19	42741	34065	13928	11422	4729	99999	11033
20	47381	38516	18379	15873	12493	9909	99999
21	48400	39535	19398	16892	13512	10928	1879
22	43969	35104	14967	12461	9081	5700	6184
23	27636	18771	7319	11900	13610	18723	24953
24	26511	17646	8188	12769	14369	17756	25822

	21		22	23	24
	Ahmetyesevi		Esenler Mah.	inönü Mah.	Kayışdağı
13	Taşdelen	32119	25329	13067	11815
14	Ortaçeşme	49706	46721	29258	28006
15	Kavacık	39699	36714	18546	17999
16	Zümrütevler	19040	16055	7551	8485
17	Cevizli	15730	12745	12233	13167
18	Gümüşpınar	13572	10587	13691	14625
19	Kurfali	10439	5817	19472	17625
20	Tuzla	1956	5218	23923	24857
21	Ahmetyesevi	99999	4991	24942	25876
22	Esenler Mah.	4738	99999	20511	21445
23	inönü Mah.	24359	21374	99999	1660
24	Kayışdağı	25228	22243	1530	99999

PERŞEMBE GÜNÜ UZAKLIK MATRİSİ

	0	1	2	3	4
	Depo	Çekmeköy	Sancaktepe	Sarıgazi	Dudullu
0	Depo	99999	40250	41570	36996
1	Çekmeköy	39340	99999	5760	3141
2	Sancaktepe	41240	99999	1440	3742
3	Sarıgazi	41100	1529	99999	5391
4	Dudullu	36136	3503	4723	99999
5	Yeni Çamlıca	39260	5865	6723	4124
6	Fındıklı	41291	9417	11667	7676
7	İçerenköy	39722	9748	11998	8007
8	Fikirtepe	43220	16705	18885	13426
9	Fatih Mah.	40006	12871	15051	9839

	5		6	7	8	9
	Yeni Çamlıca		Fındıklı	İçerenköy	Fikirtepe	Fetih Mah.
0	Depo	39938	42996	41193	42940	41577
1	Çekmeköy	7203	10365	10625	16845	15177
2	Sancaktepe	6190	9990	10250	16470	15107
3	Sarıgazi	7007	11380	11640	17860	16497
4	Dudullu	4628	7790	8050	14270	9912
5	Yeni Çamlıca	99999	5090	5720	12940	9682
6	Fındıklı	5069	99999	1799	9019	7656
7	İçerenköy	5237	1813	99999	7212	6039
8	Fikirtepe	12077	10506	7201	99999	5077
9	Fetih Mah.	9184	9316	7513	4040	99999

CUMA GÜNÜ UZAKLIK MATRİSİ

	0	1	2	3	4
	Depo	AltKaynarca	Kartal	Küçükyalı	Atakent
0	Depo	99999	50068	43160	34735
1	AltKaynarca	58167	99999	17118	25530
2	Kartal	50140	99999	10190	18728
3	Küçükyalı	43323	9601	99999	11811
4	Atakent	34460	18363	12071	99999
5	Namikkemal	35500	18433	11525	1799
6	Yunusemre	36275	17861	10953	2455
7	Bulgurlu	36618	18068	11060	3103
8	Çengelköy	36973	20210	17982	4018
9	Şerifali	39041	16565	10917	4548
10	Selami Ali	40710	21708	14700	8375
11	Fıstıkkağacı	40200	21198	14190	7865

	5	6	7	8	9	10	11
	Namıkkemal	Yunusemre	Bulgurlu	Çengelköy	Şerifali	Selami Ali	Fıstıkağacı
0	Depo	35265	36026	35726	36250	40443	39594
1	AltKaynarca	24876	24315	25042	27148	22462	28012
2	Kartal	19316	18775	18114	20220	17778	21084
3	Küçükyalı	12499	11938	11297	17133	10955	14267
4	Atakent	1844	2806	3056	3520	5010	7404
5	Namıkkemal	99999	876	1490	2943	6294	6368
6	Yunusemre	876	99999	1342	3664	6883	6380
7	Bulgurlu	1532	1544	99999	2668	7668	4894
8	Çengelköy	2939	3499	2147	99999	8023	5262
9	Şerifali	6214	6159	7524	7698	99999	13385
10	Selami Ali	7296	7242	5842	6990	13983	977
11	Fıstıkağacı	6786	6732	5332	6480	13473	99999

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Feyyaz Celalettin KURUL
Doğum Tarihi ve Yeri : 16.10.1987 / SAMSUN
Yabancı Dili : İngilizce
E-posta : feyyazkurul@gmail.com

ÖĞRENİM DURUMU

Derece	Alan	Okul/Üniversite	Mezuniyet Yılı
Lisans	Matematik Mühendisliği	Yıldız Teknik Üniversitesi	2010
Lise	Fen	Özel Gökkuşığı Lisesi	2004