

**TÜRKİYE’ DE ÇOK MODLU TAŞIMACILIKTA
EN KISA YOLLARIN BELİRLENMESİ**

Melike YETGİN

**YÜKSEK LİSANS TEZİ
ENDÜSTRİ MÜHENDİSLİĞİ**

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**ŞUBAT 2011
ANKARA**

Melike YETGİN tarafından hazırlanan “TÜRKİYE’DE ÇOK MODLU TAŞIMACILIKTA EN KISA YOLLARIN BELİRLENMESİ” adlı bu tezin Yüksek Lisans tezi olarak uygun olduğunu onaylarım.

Prof. Dr. Cevriye GENCER

.....

Tez Danışmanı, Endüstri Mühendisliği Anabilim Dalı

Bu çalışma, jürimiz tarafından oy birliği ile Endüstri Mühendisliği Anabilim Dalında Yüksek Lisans tezi olarak kabul edilmiştir.

Yrd. Doç. Dr. Haluk AYGÜNEŞ

.....

Endüstri Müh. Anabilim Dalı, Çankaya Üni.

Prof. Dr. Cevriye GENCER

.....

Endüstri Müh. Anabilim Dalı, Gazi Üni.

Yrd. Doç. Dr. Feyzan ARIKAN ÖKTEMER

.....

Endüstri Müh. Anabilim Dalı, Gazi Üni.

18/02/2011

Bu tez ile G.Ü. Fen Bilimleri Enstitüsü Yönetim Kurulu Yüksek Lisans derecesini onamıştır.

Prof. Dr. Bilal TOKLU

.....

Fen Bilimleri Enstitüsü Müdürü

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü kaynağa eksiksiz atıf yapıldığını bildiririm.

Melike YETGİN

**TÜRKİYE’ DE ÇOK MODLU TAŞIMACILIKTA
EN KISA YOLLARIN BELİRLENMESİ
(Yüksek Lisans Tezi)**

Melike YETGİN

GAZİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

Şubat 2011

ÖZET

Çok modlu taşımacılık en az iki ya da daha fazla taşıma türünün (karayolu, demiryolu, denizyolu ya da havayolu) tek bir taşımacılık zincirinde birleştirilmesidir. Çok modlu taşımacılığın getirdiği faydalar ekonomik, çevresel ve operasyonel boyuttadır ve bu taşımacılığın önemi uluslararası ve şehirlerarası gibi uzun mesafe ulaştırma problemlerini içeren lojistik uygulamalarında önem kazanmaktadır.

Bu çalışmada, Türkiye’ de yurt içi taşımacılıkta mevcut coğrafi şartlar göz önüne alınarak karayolu ve demiryolunu içeren entegre bir sistem önerilmiştir. Önerilen sistemde, karayolları ve demiryollarının mevcut durumları değerlendirilerek, demiryolu ile ulaşımın olduğu yerlere demiryolu ile, demiryolunun olmadığı yerlere karayolu ile ulaşılarak tüm ülkede en kısa turun bulunması hedeflenmiştir. Bu bağlamda gezgin satıcı problemleri ile en kısa yolların belirlenmesinde karınca kolonisi algoritması kullanılmıştır.

Taşımacılık faaliyetlerinin maliyetleri arasında en büyük payı enerji maliyetleri oluşturmaktadır. Bu nedenle bulunan en kısa yollar tüketilen

enerji miktarı açısından incelenmiştir. Ayrıca, çevreye yayılan karbondioksit miktarı da değerlendirilmiş ve güvenlik, arazi kullanımı, gürültü ve alt yapı maliyetleri yorumlanmıştır.

Bilim Kodu : 906.1.148

Anahtar Kelime : karınca kolonisi algoritması, çok modlu taşımacılık, gezgin satıcı problemleri, sezgisel algoritmalar

Sayfa Adedi : 162

Tez Yöneticisi : Prof. Dr. Cevriye GENCER

**DETERMINING THE SHORTEST PATHS
FOR MULTI-MODAL TRANSPORTATION IN TURKEY**

(M.Sc. Thesis)

Melike YETGİN

**GAZI UNIVERSITY
INSTITUTE OF SCIENCE AND TECHNOLOGY
FEBRUARY 2011**

ABSTRACT

Multi-modal transport is combination of at least two or more type of transportation (road, rail, sea or air) as a single transport chain. Multi-modal transportation benefits brought by economic, environmental and operational size and importance of this international and inter-city transport logistics applications, including problems such as long distance transportation is gaining importance.

In this study, Turkey's geographical conditions, taking into consideration the existing domestic transport by road and rail with an integrated system is proposed. The proposed system, by evaluating the current status of roads and railways, rail transport by rail to places that the railway is not reached by road to places all over the country aimed to find the shortest tour. In this context, traveling salesman problems with ant colony algorithm is used to determine the shortest routes.

Energy costs constitute the largest share of the costs of transport activities. Therefore, the amount of the shortest paths is examined in terms of energy consumption. In addition, the amount of carbon dioxide emitted to the environment is evaluated and safety, land use, noise and infrastructure costs are interpreted.

Science Code : 906.1.148

**Key Words : ant colony algorithm, multi-modal transport, travelling
salesman problems, heuristic algorithms**

Page Number : 162

Adviser : Prof. Dr. Cevriye GENCER

TEŞEKKÜR

Çalışmalarım boyunca çok değerli yardım ve katkılarıyla beni yönlendiren Hocam Prof. Dr. Cevriye GENCER'e, çalıştığım kurumda maddi-manevi destekleriyle beni hiçbir zaman yalnız bırakmayan çok değerli Müdürüm Abuzer Karadağ ve iş arkadaşlarım ile sevgili eşim Ömer Emre YETGİN'e teşekkürü bir borç bilirim.

İÇİNDEKİLER

Sayfa

ÖZET	iv
ABSTRACT	vi
TEŞEKKÜR	viii
İÇİNDEKİLER	ix
ÇİZELGELERİN LİSTESİ	xiii
ŞEKİLLERİN LİSTESİ	xiv
HARİTALARIN LİSTESİ	xvi
SİMGELER VE KISALTMALAR	xvii
1. GİRİŞ	1
2. TAŞIMACILIK	4
2.1. Taşımacılık Türleri	4
2.1.1. Karayolu taşımacılığı.....	4
2.1.2. Demiryolu taşımacılığı	10
2.1.3. Denizyolu taşımacılığı.....	15
2.1.4. Havayolu taşımacılığı	21
2.1.5. Boru hattı taşımacılığı	25
2.1.6. İç – Su (nehir) yolu taşımacılığı.....	27
2.2. Taşımacılık Modları	27
2.2.1. Tek modlu taşıma (unimodal transport).....	27
2.2.2. Modlar arası taşıma (intermodal transport).....	28
2.2.3. Çok modlu taşımacılık (multimodal transport)	28
2.2.4. Kombine taşımacılık (combined transport)	28

Sayfa

2.3. Çok Modlu Taşımacılık	30
2.3.1. Çok modlu taşımacılığın gelişimi.....	31
2.4. Türkiye’ de Taşımacılık.....	32
3. GEZGİN SATICI PROBLEMİ	34
3.1. Gezgin Satıcı Probleminin Tarihsel Gelişimi.....	36
3.2. Tur Belirleme Problemlerinin Sınıflandırılması.....	39
3.2.1. Euler turlu problemler.....	39
3.2.2. Hamilton turlu problemler.....	40
3.3. Gezgin Satıcı Probleminin Temel Özellikleri	42
3.4. Gezgin Satıcı Problemi Çeşitler	44
3.5. Karmaşıklık Teorisi.....	47
3.6. GSP için Geliştirilen Çözüm Yöntemleri	50
3.6.1. Kesin çözüm yöntemleri	50
3.6.2. Sezgisel çözüm yöntemleri	64
3.6.3. Meta-sezgisel yöntemler	73
4. KARINCA KOLONİSİ OPTİMİZASYONU	85
4.1. Gerçek Karıncaların Davranışları	88
4.2. Yapay Karıncalar ve Karınca Koloni Algoritması	90
4.2.1. Yapay karıncalar	90
4.2.2. Karınca koloni algoritması.....	91
4.3. Geliştirilmiş Karınca Algoritmaları	100
4.3.1. Karınca sistemi.....	100
4.3.2. Elitist karınca sistemi (kselit).....	102

Sayfa

4.3.3. ANT-Q.....	103
4.3.4. Karınca koloni sistemi.....	104
4.3.5. Maksimum-minimum karınca sistemi.....	106
4.3.6. Rank temelli karınca sistem.....	107
4.3.7. Çoklu karınca sistemi.....	107
4.3.8. Melez karınca kolonisi optimizasyonu	108
4.3.9. Global karınca kolonisi optimizasyonu	108
4.3.10. Karıncalar	109
5. TÜRKİYE’DE ÇOK MODLU TAŞIMACILIKTA EN KISA YOLUN KARINCA KOLONİSİ ALGORİTMASI İLE ÇÖZÜMÜ	110
5.1. Neden Demiryolu Taşımacılığı	111
5.1.1. Artan ulaşım talebi	111
5.1.2. Enerji tüketimi	112
5.1.3. Trafik sıkışıklığı.....	113
5.1.4. Trafik kazaları	113
5.1.5. Arazi kullanımı ve yol yapım maliyetleri	113
5.1.6. Hava kirliliği	114
5.1.7. Gürültü	114
5.2. Demiryolu – Karayolu ile Taşıma Sistemlerinin Karşılaştırılması	115
5.3. Problemin Tanımı	119
5.4. Problemin Verileri.....	119
5.5. Problemin Çözümü.....	122
5.6. Problemin Sonucu	124

	Sayfa
5.7. Sonucun Değerlendirilmesi	127
6. SONUÇLAR VE ÖNERİLER	130
KAYNAKLAR.....	134
EKLER	143
EK-1 İller Arası Mesafe Çizelges	144
EK-2 Uygulamaya Ait Kaynak Kodları	145
ÖZGEÇMİŞ	162

ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 2.1. Taşıma türlerinin özelliklerine göre karşılaştırılması.....	29
Çizelge 4.1. Kronolojik olarak Karınca Koloni Optimizasyon Algoritmaları ...	86
Çizelge 4.2. Karınca Kolonisi Optimizasyon Uygulamaları	87
Çizelge 5.1. Trafikte kaza durumu	113
Çizelge 5.2. Trafikte kaza sonucu ölü ve yaralı sayısı	113
Çizelge 5.3. Örnek iller arası mesafe	120
Çizelge 5.4. Kullanılan parametrelere göre oluşan sonuçlar	123
Çizelge 5.5. Problemden kullanılan parametreler	124
Çizelge 5.6. İllere göre çok modlu en kısa Türkiye turu	126
Çizelge 5.7. Çalışmaların karşılaştırması	129

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 3.1. Örnek gezgin satıcı turu	35
Şekil 3.2. Hamilton ve euler grafi	40
Şekil 3.3. Çoklu gezgin satıcı (m-TSP) turu	45
Şekil 3.4. P, NP, NP-tam ve NP-zor küme çizimi	49
Şekil 3.5. GSP'nin dinamik programlama ile çözüm algoritması	52
Şekil 3.6. Minimum yayılan ağaç	65
Şekil 3.7. Geometrik yöntem	67
Şekil 3.8. Christofides algoritmasında M ağı	68
Şekil 3.9. (a) 2-opt değişimi (b) 3-opt değişimi	68
Şekil 3.10. Lin-Kerninghan yöntemi algoritma adımları	71
Şekil 3.11. Örnek hyperopt denemeleri	72
Şekil 3.12. Dal ve çevrim çıkarma	73
Şekil 4.1. Gerçek karıncaların yuvaları ile yiyecek kaynağı arasında yol almaları	88
Şekil 4.2. Gerçek karıncaların yolları üzerine engel konulması	88
Şekil 4.3. Gerçek karıncaların engel karşısında ilk etapta yol tercihleri	89
Şekil 4.4. Gerçek karıncaların zamanla kısa yolu bulmaları	89
Şekil 4.5. Gezgin satıcı problemi	91
Şekil 4.6. Yatay karıncanın ilk adımda yol tercihi (eşit olasılık)	93
Şekil 4.7. KKS algoritması	105
Şekil 5.1. Yolcu taşımacılığında birim nakliyat başına enerji tüketimi (kcal/kişi- km)	112

Şekil	Sayfa
Şekil 5.2. Yük taşımacılığında birim nakliyat başına enerji tüketimi (kcal/ton-km)	112
Şekil 5.3. Parametre seçimi	121
Şekil 5.4.Kullanılan arayüz.....	123
Şekil 5.5.Bulunan sonuca göre Türkiye turu	125

HARİTALARIN LİSTESİ

Harita	Sayfa
Harita 5.1. Türkiye' nin demiryolu haritası	120
Harita 5.2. Türkiye' nin karayolu haritası	121

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış bazı simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler	Açıklama
$a_{ij}(t)$	t. iterasyonda (i,j) kenarında karınca kararlılığı
B	Şehirleri gösteren bileşenler kümesindeki bağlantılar kümesi
c_{ij}	i müşterisinden j müşterisine maliyet
d_{ij}	i. ve j. nokta arası mesafe
E_i	Katının enerjisi
E_j	j durumuna geçen katının enerjisi
$f(x)$	Fonksiyon
$G(V,E)$	Ağ Kümesi
I^r	Rotalar kümesi
$Jr_i r_j$	r_i ve r_j arasındaki bağıntının maliyetini
k_B	Fiziksel tavlama sürecinde Boltzman Sabiti
L	Mesafe kısıtı
L^b	En iyi turun uzunluğu
l_{ij}	Gezgin satıcı probleminde i'den j'ye gitme maliyeti
$L^k(t)$	t iterasyonunda k karıncası tarafından üretilen turun uzunluğu
L_{nn}	En yakın komşu sezgiseli turu
m	Karınca sayısı
M_k	Karıncanın özel hafızası, Yasak listesi
n	İterasyondaki şehir sayısı
N_i	i noktasının komşuluğunda olan noktalar kümesi
N_i^k	k karıncasının henüz ziyaret etmediği i noktasının komşuluğunda olan noktalar kümesi
q_0	Sözde rastlantısal kural olasılığı
Q_1	Karınca-nicelik modelinde sabit feromen miktarı
Q_2	Karınca yoğunluk modelinde sabit feromen miktarı
Q_3	Karınca-çevrim modelinde sabit feromen miktarı
$p_{ij}^k(t)$	k karıncasının t. iterasyonda i noktasındayken j noktasını seçme olasılığı
R	Şehirleri gösteren bileşenler kümesi

(r, s)	Karıncaların yaptığı en iyi turdaki r ve s noktalarını birleştiren kenar
S	Uzay
s	Sözde-rastlantısal durum geçiş kuralı
s'	Alt uzay
T	Sıcaklık parametresi
$\tau_{ij}(t,)$	t zamanında (i,j) kenarı üzerindeki feromon miktarıdır.
$t_{\max}$	Maksimum iterasyon sayısı
$T^k(t)$	t iterasyonunda k karıncası tarafından üretilen tur
τ_0	Başlangıç feromon seviyesidir.
$[\tau_{\min}, \tau_{\max}]$	Feromon aralığı
δ	Dairesel permütasyon
$\Delta\tau_{ij}^k(t)$	k karıncasının t. iterasyonda (i,j) kenarına bıraktığı feromon miktarı
X_{ijk}	Karar değişkeni
x, y	Noktanın koordinatı
V	Düğüm Kümesi
Z	Amaç Fonksiyon
ρ	Buharlaştırma katsayısıdır.
η_{ij}	i noktasından j noktasında hareketin sezgisel değeri
α	Feromon izinin ağırlığını kontrol eden parametre
β	Sezgisel değeri kontrol eden parametre

Kısaltmalar**Açıklama**

GA	Genetik Algoritma
GSP	Gezgin Satıcı Problemleri
GKKO	Global Karınca Kolonisi Optimizasyonu
IAS	Geliştirilmiş Karınca Sistemi
KKS	Karınca Kolonisi Sistemi
KKO	Karınca Kolonisi Optimizasyonu
KS	Karınca Sistemi
MACS	Çoklu Karınca Kolonisi Sistemi
MKKO	Melez Karınca Kolonisi Optimizasyonunda
MMAS	Maks-Min Karınca Sistemi

TA

Tabu (Yasak) Arama

TB

Tavlama Benzetimi

1. GİRİŞ

Günümüz üretim, ticaret ve tüketim faaliyetlerinin verimli şekilde gerçekleştirilebilmesi için hammadde, yarı mamul ve bitmiş ürün taşımaları, etkin sistemler ve doğru zamanlamalar ile desteklenmelidir. Taşımacılık maliyetleri, ürün maliyetlerinin büyük bir kısmını oluşturması sebebiyle işletmelerin ve dolayısıyla ülkelerin rekabet edebilirliği üzerinde doğrudan etkiye sahiptir. Bununla birlikte, taşımacılık endüstrisinin hem ekonomik etkinlik hem de hizmet kalitesi açısından belirli performans kriterlerini sağlaması gerekmektedir.

Türkiye'deki taşımacılık sisteminin koordineli ve kombine yapıda faaliyet gösterebilmesi için demiryolu sektörünün gelişimi üzerinde de durulmalıdır. Diğer taşıma modlarının hızlı gelişimi karşısında gerekli önlemlerin alınmaması, demiryolu taşımacılığının geleceğini tehdit eder duruma gelmiştir. Avrupa ülkelerinin çoğu son 30 yıldır kendi demiryollarının mali ve pazar performansını gözden geçirmekte ve yeniden yapılandırmaktadır. Daha da genel olarak dünyaya bakıldığında demiryolu sektörünün sadece diğer taşıma modları karşısında rekabet gücünü artırmaya değil, aynı demiryolu alt yapısı üzerinde birden fazla işletmeciye faaliyet imkânı vererek sektör içi rekabet yaratılmaya çalışılmaktadır. Teknik değişim süreci içerisindeki faaliyetlerle de demiryollarının uluslararası entegrasyonu sinyalizasyon, telekomünikasyon, çeken ve çekilen araçlar ve ekipmanların birbirine uyumlu hale getirilmesi ile sağlanmaya çalışılmaktadır.

Türkiye demiryolu ağına bakıldığında, 1950'li yıllardan bu yana ülkedeki gelişmelere paralel olarak yeterli demiryolu inşa edilmemiş ve mevcut demiryollarının düşük olan fiziki ve geometrik standartları yeterli seviyede düzeltilmemiştir. Bunun altında taşımacılık politikaları, işletmecilik, mali yapı, alt yapı, çeken ve çekilen araç parkı, yönetim ve organizasyon yapısı ve insan kaynakları ilgili sorunlar yatmaktadır. Bu sorunlara kısa dönemde,

nitelikli personel ve vagon alımı gibi hızlı önlemlerle yaklaşılabılır. Uzun dönemde ise ülke içindeki taşımaların çağdaş seviyeye getirilmesi ve batıda Avrupa Birliği, doğuda ise Türkî Cumhuriyetler, Orta Doğu Ülkeleri ve Karadeniz çevresindeki Bağımsız Devletler Topluluğu ile uyumu ve bütünleşmeyi sağlamak üzere gerekli plan ve eylemlerin gerçekleştirilmesi gerekmektedir.

Karayolunun alternatifi olan demiryolu ve denizyolu taşımacılığı ile ilgili anlatılan eksikliklere rağmen taşıma modlarının birlikte kullanılması, varış ya da çıkış noktası Türkiye olan ya da transit geçen yüklerin taşınması açısından hayati öneme sahiptir. Mevcut durumda da lojistik alanında hizmet veren işletmelerin taşıma modlarının faydalarını her taşıma projesinde bir araya getirmeleri gerekmektedir. Çok modlu ağların en iyi şekilde koordine edilmesi ve maliyet, süre ve diğer faktörler açısından etkin taşıma projelerinin gerçekleştirilebilmesi, lojistik firmasının hizmet kalitesini, güvenilirliğini ve karlılığını artıracaktır. Ülke ekonomisi açısından bakıldığında, uygun liman yatırımları ve modlar arası geçişin sağlandığı aktarma noktalarının sayı ve kalitesinin artması durumunda hem mevcut yük hacmi daha etkin taşınacak, hem de ülkemiz üzerinden geçen yük hacminde artış olacaktır. Dolayısıyla çok modlu taşımacılık ağlarında taşıma optimizasyonu üzerine çalışmak özellikle Lojistik Hizmet Sağlayıcıları (LHS) işletmeler açısından ve dolaylı olarak ülke ekonomisi açısından önem taşımaktadır.

Tüm bu sayılan sebeplerden dolayı, taşımacılık faaliyetinde bulunan LHS işletmeler stratejik düzeyde kararlarını alırken, kuracakları modelde taşıma modlarının etkisini büyük oranda dikkate almalıdırlar. Mevcut literatüre bakıldığında, taşıma modları ve çok modlu taşımacılık üzerinde durulmuş birçok çalışmaya rastlanmaktadır. Problemler, çok modlu taşımacılık içerisinde gereken tesisler ile ilgili yer seçimleri, aktarma terminallerinin operasyonları, dağıtım veya tedarik aşamaları, uzun taşıma aşamaları gibi farklı alanları içermektedir. Diğer bir bakış açısı ise taşımacılık ağının

oluşturulması veya oluşturulmuş ağ üzerinde taşıma operatörlerinin uygun rotayı ve taşıma modunu seçmesidir. Çözüm yöntemi olarak da, deterministik, stokastik, sezgisel, simülasyon gibi yaklaşımlara karşılık sözel olarak yorumların yapıldığı ve değişik bakış açılarının yansıtıldığı çalışmalar bulunmaktadır.

Lojistik sisteminin en önemli parçası ve problemi Gezgin Satıcı Problemleridir (GSP) [1]. GSP için en iyi çözümü veren algoritmalar metasezgisel algoritmalarlardır. Metasezgisel algoritmaların popülasyon tabanlı türlerinden biri olan Karınca Kolonisi Algoritması GSP'lerin çözümünde bir çok çalışmada kullanılmış ve tatmin edici sonuçlar alınmıştır. Karınca Kolonisi Algoritması gerçek karıncaların yiyecek kaynağı ile yuvaları arasındaki en kısa yolu bulabilme özelliklerinin fark edilip incelenmesiyle ortaya çıkmıştır. Karıncaların kimyasal izlerden bilgi akışı sağlamalarının sonucu en kısa yolu bulabildikleri ortaya çıkmıştır. Karınca Kolonisi Algoritması birçok problemin çözülmesi çalışmalarında kullanılmıştır.

Bu çalışmada, Tedarik Zincirinin bir parçası olan lojistik sistemi ile doğrudan alakalı olan fiziksel dağıtım sürecinde sıkça rastlanılan gezgin satıcı problemleri üzerinde durulmuştur. 2. bölümde taşımacılık türlerinden, modlarından ve Türkiye'deki taşımacılıktan bahsedilmiştir. 3. bölümde GSP'nin çeşitli kısıt ve özellikleri ile türleri ve bu problemlere çözüm olarak sunulan değişik algoritmalar detaylı olarak incelenmiştir. 4. bölümde Karınca Kolonisi Algoritması detaylı bir şekilde incelenmiştir. Çalışmanın 5. bölümünde metasezgisel yöntemlerden olan Karınca Kolonisi Optimizasyonu kullanılarak Türkiye'de çok modlu taşımacılıkta en kısa yolun belirlenmesine çözüm aranmıştır. Çalışmanın son bölümünde ise yapılan uygulama sonucu ve sonuca bağlı önerilerden bahsedilmektedir.

2. TAŞIMACILIK

2.1. Taşımacılık Türleri

2.1.1. Karayolu taşımacılığı

Tarihte, Türkiye topraklarının coğrafi konumu, Avrupa ve Asya kıtaları arasında geçit oluşturması, ticari amaçla, günün şartlarına uygun yolların yapılmasını sağlamıştır. Selçuklu döneminde yapılan yollar ve kervansaraylar ile bir ulaşım ağı oluşturulmuştur. Osmanlı İmparatorluğu döneminde de yolların yapılmasına devam edilmiştir. Özellikle, yükseliş döneminde ihtiyaca cevap verebilecek bir yol ağı oluşturulmuş ve uzun süre korunmuştur. İmparatorluğun son dönemlerinde, o zamana kadar başarı ile hizmet veren yarı askeri özellikteki karayolu organizasyonu, yolları ihmal ederek bakımsız kalmalarına sebep olmuştur.

1923 yılında Türkiye Cumhuriyeti'nin kuruluşu itibarı ile 4000 km'lik kısmı iyi durumda olan 18 350 km'lik bir ağ mevcuttu. Cumhuriyetin ilk yıllarında yapımına öncelik verilen demiryollarının tek başına yeterli olmadığı görülerek, 1929 yılında Nafia Vekaleti (Bayındırlık Bakanlığı) içinde Şose ve Köprüler Reisliği kuruldu ve yol kanunu çıkarılarak, karayolu yapımı çalışmalarına hız verildi. 1948 yılında, Amerika Birleşik Devletleri'nden sağlanan maddi yardımların yol yapımına büyük katkısı olmuştur. Yolların yapım ve onarım çalışmalarına hız verilmiştir. 1950 yılında karayollarının toplam uzunluğu 45.000 km'ye ulaşmıştır. Cumhuriyet dönemi karayollarının gelişimi açısından, 1948 yılı ve öncesi, kazma kürek ve insan gücüne dayanan ve 1948 sonrası da makineli çalışma dönemi olarak ikiye ayrılabilir. Makineli dönemin başlaması ile, devlet ve il yollarında ucuz, süratli ve kademeli yapılacak bir sistemin uygulanmasına başlandı ve mevcut yolların envanteri çıkartılarak, Türkiye'nin ihtiyacını karşılayacak bir yol ağı belirlendi. Üzerinde durulan önemli bir nokta da, çalışmaların belli bir plan ve program

çerçevesinde yürütülmesiydi. Bu etkinliklerin yönlendirilmesi gereğinden hareketle, 1950 yılında Karayolları Genel Müdürlüğü (KGM) kuruldu. Yeni karayolu politikası ile birlikte, KGM'ye bağlı yurt çapında dağılmış bölge teşkilatları oluşturuldu, makineli çalışmanın yürütülebilmesi için gerekli makine parkı sağlandı, personel eğitime ağırlık verilerek planlama, etüt proje, yapım ve bakım çalışmalarına başlandı. O dönemde, az olan motorlu taşıt trafiği yüksek standartlı yollar gerektirmiyordu. 1950–1960 yılları arasında yapılan çalışmaların ana amacı, sağlık ve eğitim gibi sosyal hizmetlerin, ülkenin her yöresine götürülebilmesi için yolların yaz-kış hizmet verebilecek duruma getirilmesi olmuştur. Kademeli inaat sistemi ile, bir yandan trafik yoğunluğuna uygun olarak yollar iyileştirilmekte, diğer yandan yeni güzergahlar belirlenerek, modern ölçülere uygun yol şeritleri ile ülkenin ekonomik bölgeleri arasında bağlantı kurulmaktaydı. 1960'lı yıllara gelindiğinde, 60 000 km'lik bir yol ağı oluşturulmuştu. Yol ağının bu düzeye erişmesinden sonra artan trafik ile birlikte 1960–1970 yılları arası asfalt kaplamaya ağırlık verilmesi zorunluluk olmuştur. 1970'li yıllarda, motorlu taşıt sanayinin kurulması karayolu politikalarına yeni bir boyut getirmiştir. Trafik sıklığının söz konusu olduğu bazı ana akslarda ve büyük şehir çevrelerinde, çok şeritli ekspres yollar ya da otoyollar ekonomik olma niteliği kazanmıştır. KGM'nin sorumluluğu altında üç sınıf yol bulunmaktadır: Otoyollar, devlet yolları ve il yolları. Diğer kuruluşların sorumluluğundaki yollar ise, köy yolları, turistik yollar, orman yolları ve şehir içi yollardır.

Karayolu taşımacılığının çevreye verdiği zararlar, alt yapı bakım-onarım ve birim taşıma maliyetlerinin yüksek olması gibi ana nedenlerle AB, Türkiye' ye oranla daha düşük olan karayollarının payını düşürmeye çalışmaktadır. Karayolu taşımacılığının denizlere kaydırılmasını öngören "Karadan Denize-From Road to Sea" projesini teşvik etmektedir. Türkiye'de diğer sistemlere göre karayolu ağının geliştirilmesine daha çok önem verilmesi sonucu araç sayısı hızla artmış ve atıl kapasite oluşmuştur. Bu araçlar için yapılan ciddi yatırımlar ülke ekonomisine külfet getirmiş, ayrıca oluşan atıl kapasite

sebebiyle karayolu taşımacılık sektöründe mali açıdan güçlü, rekabet edebilir, kurumsallaşmış yapılar ortaya çıkmasına engel olmuştur.

Mali açıdan verimli olmayan bu yatırımlar, girişimcileri borç zorlaması nedeniyle ara vermeden uzun süreli çalışmaya ve aşırı yüklemeye itmektedir. Bu eğilim kara yollarının vaktinden önce yıpranmasına ve kazalara yol açmaktadır. KGM sorumluluğu altında bulunan karayolu uzunluğu 2008 tarihi itibari ile 64 033 km' dir [3]. Karayolları; otoyol, devlet yolu ve il yolu olmak üzere üç gruptan oluşmaktadır. 5339 sayılı kanunla otoyolların, devlet ve il yollarının planı, proje, bakım ve işletilmesi KGM'nin sorumluluğundadır. Turistik yollar, şehir içi yollar, köy yolları diğer kuruluşların sorumluluğu altındadır. Turistik yolların bakım ve onarımı Turizm bakanlığınca sağlanan finansmanla KGM tarafından yapılmaktadır.

2008 yılı sonu itibari ile Devlet Yolları uzunluğu 31 311 km, İl yolları uzunluğu 30 712 km'dir. 1980'lerde yapımına başlanan otoyol toplam uzunluğu ise 2004 yılı sonunda 2 010 km'ye ulaşmıştır [3]. Gerek araç sayısı fazlalığı ve gerekse aşırı yüklemeler, karayolu üzerindeki trafik güvenliğini olumsuz yönde etkilemektedir. Karayollarının geometrik ve fiziki kapasitesinin üzerinde yüklenmesi, karayolları üzerinde trafik güvenirliliğinin azalmasına, kazaların hızla artmasına sebep olmaktadır. Türkiye'de karayolları üzerinde ağır taşıt oranının yüksek olması trafik güvenliğini azaltan önemli hususlardan bir tanesidir.

Karayolu taşımacılığının dünyadaki gelişimi

1930 yıllarında ortaya çıkan küresel ekonomik bunalım ve İkinci Dünya Savaşı sonrası ortaya çıkan ekonomik olumsuzlukları aşan ve güçlenen ülkeler, ekonomik yönden güçsüz ve az gelişmiş ülkelerin ekonomileri üzerindeki etki ve denetimlerini artırmışlardır. "Öncelik tarım" telkinleri ile gelişmiş ülkeler güçsüz ekonomiye sahip ülkeleri bu yönde üretim yapmaya

yönlendirmişlerdir. Bunun sonucunda sanayiden uzaklaşan, tarıma dayalı bir ekonomik yapı oluşturan ülkeler, sanayileşme yolunda gelişme gösteren ülkelerin dış pazarı hâline gelmiştir.

Gelişmiş ülkelerin ekonomilerine çok yönlü faydalar sağlayacak olan bu durumun gerçekleşmesi ve gelişmesi için, her yerleşim alanına ulaşabilecek ve taşımayı gerçekleştirecek en önemli ulaşım sistemi kara yoluydu. Kara yolu ağının kurulması ve geliştirilmesi için “Öncelik kara yoluna” düşüncesini hâkim kılmıştır.

Karayolu ulaşım sisteminin özellikleri

Kara yolu taşıma sistemleri; teknoloji, şebekeler (ağlar), uluslararası ve yerel kurallar ve düzenlemeler, bilgi ve iletişim, lojistik ve hizmet anlayış ve uygulamalarından meydana gelmektedir. Bu sistemler taşıma operatörleri, iç ve dış müşteriler, ekonomik ve sosyal faktörler ile devletin ortaya koyduğu yasal çerçeve ile sürekli etkileşim hâlinindedir.

Kara yolu taşımacılık sektöründe arz ve talebin etkileşiminde destek hizmetler, taşımaya olan talep, güzergâh veya alternatif yollar, terminaller, gümrükler, sınırlar, ara geçiş noktaları, araçlar, işletmelerin sayısı, kurallar ve düzenlemeler etkin rol oynamaktadır. Buna göre taşıma sistemleri özellikle teknoloji, iletişim, ekonomi gibi dışsal faktörlerden yoğun olarak etkilenmektedir.

Kara yolu taşımacılığı, ücret karşılığında eşyanın bir yerden diğer bir yere taşınmasını kara yolu ile sağlayan ve taşımacı ile gönderici arasında bir sözleşme yapılmasını gerektiren bir taşıma şekli olarak tanımlanmaktadır. Bu tanıma göre kara yolu taşımacılığının dört temel boyutu vardır;

- Taşınacak eşyanın var olması,

- Eşya taşıma işinin üstlenilmesi,
- Ücret karşılığı olması,
- Taşıma türü olarak kara yolunun belirlenmesidir [4].

Karayolu sisteminin diğer sistemlere göre üstünlükleri

Uluslararası kara yolu eşya taşınması ile diğer taşıma türlerine göre bir karşılaştırma yapıldığında kara yolu taşımacılığının şu üstün yönleri bulunmaktadır:

- Kapıdan kapıya aktarmasız taşıma ve eşyanın yükleme yeri ile boşaltma yerleri dışında elleçlenmemesi, eşyanın yıpranmasını en aza indirmektedir. Diğer taşıma türlerinde, örneğin eşyanın limana veya demir yolu istasyonuna ulaşımına kadar bir yükleme boşaltma, ana taşıma aracına yüklenme, varış yeri limanı veya istasyonunda benzer elleçlemenin yapılması eşyaların daha fazla yıpranmasına sebep olmaktadır.
- Müşterinin istediği ve kara yolunun olduğu her noktaya taşıma seçeneği vermektedir.
- Kara yolu araçlarının taşıma kapasitelerinin deniz, demir yolu araçlarına göre daha küçük olması nedeniyle, taşımacılık sektöründe daha esnek hareket imkânı sağlanmaktadır.
- Türkiye gibi kara yoluyla uluslararası eşya naklinde kullanılabilecek taşıt sayısının talebe göre daha fazla olduğu ülkelerde, rekabet üst seviyelere ulaşmakta ve böylelikle müşteri lehine düşük navlun fiyatlarının oluşmasına neden olmaktadır.
- Kara yolu taşımacılığı diğer taşıma türlerine göre (deniz, hava, demir yolu, boru hattı vb.) daha az yatırım maliyeti gerektirmektedir.
- Kara yoluyla daha düzenli ve sık sefer imkânı bulunmaktadır.
- Göndericinin kendisinin organize ettiği, özellikle küçük çaplı sevkiyatlara oranla taşıma maliyetini azaltmaktadır.

- Ambalajlama ve sevkıyata hazırlamada zaman ve kaynaklardan tasarruf olanağı sağlanmaktadır.
- Karayollarının inşası kademeli olarak yapılabilmektedir.
- Yol alt yapısının oluşturulması ile kaplama tamamlanana kadar, gerektiğinde yol hizmete açılabilir. Her tür araziye uygulanabilirliği, ulaşım ağı oluşturulmasında sınırsızlık özelliği kazandırmaktadır.

Karayolu taşımacılığı, göndericiye ayrıca şu imkânları sağlayabilir:

- Kısa sürede teslim edilmesi gereken yükler için süratli sevkiyat yapılabilmesine yönelik proje bazlı çözüm yolları ortaya çıkarılabilir.
- Gelişmiş karayolu ağlarının getirdiği hızlı ve güvenli hizmet alma imkânları yaratılabilir.
- Taşıma ve lojistik şirketleriyle birebir ilişki içinde çalışma olanağının bulunması sayesinde sevk edilecek yüke özel muamele yapılması sağlanabilir.
- Karayolu taşımacılık hizmetleri ile bağlantılı diğer lojistik süreçlerle zenginleştirilerek hizmette çeşitlilik oluşturulabilir.
- Tahmin edilebilen maliyetlerin kolay hesaplanması ile tasarruf sağlanabilir [4].

Karayolu taşımacılığının olumsuz yönleri

Kara yolu ile taşımacılığının bir kısım üstünlükleri olmasına rağmen, bazı olumsuz yönleri de bulunmaktadır. Bunlar;

- Karayoluyla yapılan taşımacılıkta, ihracat ve ithalat ülkeleri arasında yükün transit geçtiği ülkelerde uyulması gereken gümrük mevzuatları bulunmaktadır. Buna karşılık deniz ve havayolunda çoğunlukla ithalatçı

ve ihracatçı olarak ülkeler arasında transit geçiş ülkeleri bulunmamaktadır.

- Karayoluyla yapılan taşımacılıkta, transit gümrükler ilave bekleme süreleri oluşturmaktadır.
- Numune veya benzeri nitelikte taşınacak yükün düşük miktarda olması halinde bu yükün bir araç içinde sevk yerlerine ayrı ayrı ulaştırılması gerekmektedir. Parsiyel malların toplanması, dağıtımı, zaman ve maliyet yönünden olumsuzluk yaratmaktadır. Havayolu parsiyel taşımalarda zaman ve maliyet yönünden karayoluna göre daha tercih edilebilir bir taşıma türüdür.
- Ağır ve yüksek hacimli yüklerin karayolu ulaştırma mevzuatına uyumlu olmaması, taşınmalarını çoğunlukla imkânsız hale getirmektedir. Bu tür yüklerin demiryolu veya denizyoluyla taşınması daha verimli olmaktadır.
- Karayolu ile ulaşım petrole bağımlı bir ulaşım sistemidir ve gaz emisyonları ve gürültü kirliliğine sebep vermesinden dolayı çevreye olumsuz etkileri vardır.
- Ulaşım sistemleri arasında, kaza oranı en yüksek olan dolayısıyla güvenliği en az olan sistemdir [4].

2.1.2. Demiryolu taşımacılığı

Türkiye’de ilk demiryolunun yapımına, 23 Eylül 1856 tarihinde bir İngiliz şirketine verilen imtiyazla başlanmış, İzmir-Aydın arasında 130 km uzunluğunda inşa edilmiş ve 1866 yılında tamamlanmıştır. Bir başka İngiliz şirketine verilen imtiyazla, İzmir-Kasaba, Turgutlu-Afyon hattı ile Manisa-Bandırma hattının 98 km’lik kısmı 1865 yılında tamamlanarak işletmeye açılmıştır. Hattın diğer bölümleri sonraki yıllarda tamamlanmıştır. 1869 yılında, Baron Hirsch’e imtiyaz ile verilen 2000 km’lik şark demiryollarının milli sınırlar içinde kalan 336 km’lik İstanbul-Edirne ve Kırklareli-Alpullu kesiminin, 1888 yılında bitirilerek işletmeye açılmasıyla da, İstanbul, Avrupa demiryollarına bağlanmıştır. Anadolu’da yapılması planlanan demiryollarının

devlet tarafından inşa edilmesi düşünülmüş ve 1871 yılında Haydarpaşa-İzmit hattının yapımına başlanmış ve emaneten üç bölümde yapılan 91 km'lik hat, 1873 yılında bitirilmiştir. Daha sonra ortaya çıkarılan mali imkânsızlık nedeniyle yapımına devam edilemeyen Anadolu demiryolları ile Bağdat ve Cenup demiryollarının yapımları Alman sermayesi ile gerçekleştirilmiştir.

Cumhuriyet döneminden önce çeşitli yabancı şirketler tarafından inşa edilerek işletilen demiryollarının 4000 km'lik kısmı Cumhuriyet'in ilanı ile çizilen milli sınırlar içerisinde kalmıştır. 25.05.1924 tarihinde çıkarılan 506 sayılı kanunla bu hatlar millileştirilmiş ve Anadolu-Bağdat Demiryolları Müdüriyeti Umumiyesi kurulmuştur. Demiryollarının yapım ve işletmesinin bir arada yürütülmesini ve daha geniş çalışma imkânları verilmesini sağlamak amacıyla çıkarılan 31.05.1927 tarih ve 1042 sayılı yasa ile Devlet Demiryolları ve Limanları İdare-i Umumiyesi adını almıştır. 1953 yılına kadar, kuruluş katma bütçeli olarak yönetilmiş, 29.07.1953 tarihinden itibaren 6186 sayılı kanunla Türkiye Cumhuriyeti Devlet Demiryolları İşletmesi adı altında kamu iktisadi devlet teşekkülü haline getirilmiştir.

Türkiye demiryolu tarihi, Cumhuriyet öncesi, Cumhuriyet dönemi (1923–1950) ve 1950 sonrası dönem olarak üç bölümde incelenebilir. Bu dönemlerin belirgin özelliği, birincisinde demiryolu hatlarının büyük bir kısmının yabancılara verilen imtiyazlarla yaptırılması, ikinci dönemin demiryolu ulaştırmasının altın çağı olması ve sonuncusunun ise demiryolu ulaştırmasının ihmal edildiği ve yok sayıldığı dönem olmasıdır. 1950 yılına kadar, karayolu, demiryolunu besleyecek ve tamamlayacak bir sistem olarak görülmüştü. Karayollarının demiryollarını destekleyecek şekilde geliştirilmesi gereken dönemde, Marchall yardımı ile hızlı biçimde karayolu yapımına başlandı. Bu çerçevede oluşturulan ulaştırma politikaları ile, sistemler içinde karayolu egemen bir hale gelmiş, deniz ve demiryolu ulaşım sistemlerinde bir gelişme sağlanamamıştır. 1960–1962 yılları arasında yapımı tamamlanan demiryolu uzunluğu 458 km'dir. 1960 sonrası, planlı kalkınma dönemlerinde,

demiryolları için öngörülen hedeflere ulaşamamıştır. Ulaşım sistemleri arasında eşgüdüm planlanmasına rağmen sonuç alınamamıştır ve karayollarına yapılan yatırımlar bütün plan dönemlerinde ağırlığını korumuştur. Sanayinin artan taşıma taleplerinin yerinde ve zamanında karşılanabilmesi için, demiryollarında yatırımlara, yeniden düzenleme ve güncelleme çalışmalarına ağırlık verilmesi öngörülmüşse de, bu planlar hayata geçirilememiştir. 1950- 1980 yılları arasında yılda ortalama 30 km yeni hat yapılabilmektedir. 1980'li yılların ortalarında ise, ülkemizde hızlı bir karayolu yapım seferberliği başlatılmış, otoyollar GAP ve Turizmden sonra ülkemizin 3. büyük projesi olarak kabul edilmiştir. Bu çerçevede 1990'li yılların ortalarına kadar otobanlar için yılda yaklaşık 2 milyar dolarlık yatırım yapılmıştır. Buna karşılık, özellikle önemli demiryolu altyapı yatırımları konusunda her hangi bir projenin hayata geçirilmediği görülmektedir. Mevcut demiryollarının büyük bölümü yüz yılın başında inşa edilen geometride kalmaya mahkûm olmuştur. İdame yatırımları için ayrılan kaynaklarda yetersiz kalmıştır. Ayrıca, ülkemizde yapılmış tek ulusal ulaştırma planı olan, ulaştırma sistemimizin iyileştirilmesi yönünde bir adım olarak görülen, karayolu ulaşım payının % 72'den % 36'ya düşürülmesini hedefleyen "1983–1993 Ulaştırma Ana Planı" da uygulanmamış ve 1986 yılından sonra uygulamadan kaldırılmıştır [5].

Toplam hat uzunluğu ise 1950 ile 2009 yılları arasında sadece %48 oranında artırılarak 7 671 km den 11 405 (Ana hat uzunluğu 9 080 km) km ye çıkabilmiştir[2]. 59 yılda sağlanan bu gelişme demiryollarına önem verilemediğini bir başka anlamda izlenen politikalar sonucu unutulduğunu açıkça ortaya koymaktadır.

Ankara-İstanbul arasında yapımı planlanan ve yapımına 2003 yılında başlanılan Yüksek Hızlı tren Hattının Esenkent-Hasanbey kesimi tamamlanmış (397 km) ve Ankara-Eskişehir arasında 13.03.2009 tarihinde YHT ile yolcu Taşımacılığına başlanılmıştır.

20. Yüzyıl başlarındaki teknoloji ile inşa edilen mevcut demir yolu ağının geometrik ve fiziki standartlarının düşük olması ve tek hat işletmeciliği modern işletmeciliği olanaksız kılan sorunlar yaratmakta, ülke boyutlarına ve nüfus yoğunluğuna göre yeterli hizmet üretememektedir. Faal personel mevcudunun ihtiyaç duyulan sayının altında kalması işletmede aksamalara neden olmaktadır. Ancak personel verimliliği açısından bakınca TCDD'nin durumu diğer ülkelere göre oldukça alt seviyededir.

Genel olarak demiryolu taşımacılığı

Demir yolu taşımacılığı, ağır ve hacimli yükler için çok yüksek maliyetlere katlanılmadan yapılabilecek bir taşımacılık türüdür. Demir yolları üzerindeki merkezlerin sayısına bağlı olarak verilen hizmetin sınırlı olduğu söylenebilir. Bu taşımacılık türünde kullanılan araçların hız kapasiteleri, verilen taşıma hizmetinin hızı ile paralellik göstermektedir.

Kömür, demir gibi yeraltı kaynakları ile tarım ve orman ürünlerinin alıcı merkezlerine aktarımı demir yolu taşımacılığıyla yapılabilmektedir. Çevre dostu olan bu taşımacılık türü, uzun mesafeli taşımalarda ciddi maliyet avantajı sağlamaktadır. Kitle taşımacılığına elverişli olması ile diğer taşıma türlerinden kaynaklanan yoğunlukları) azaltıcı fayda yaratmaktadır.

Mevcut altyapı yatırımları; demir yolu inşası, işletilmesi ve bakımı yüksek yatırım maliyetleri gerektirmektedir. Bu yatırımlar çoğunlukla devlet tarafından üstlenilmektedir. Dolayısıyla devlet tarafından işletilen demir yollarının bulunduğu ülkelerde bu türdeki taşımacılık faaliyetleri kamu organizasyonlarının etkinlik ve verimliliği ile doğru orantılıdır. Ulusal ve uluslararası hatlara yeni ilaveler, hukuki düzenlemeler ve tarifeli sefer sayıları gibi konular özel sektör ve kamu kurumları arasında sıkı işbirliğini gerektirmektedir.

Ulaşımında kara yollarının kötü hava koşullarında içine düştüğü durum göz önüne alındığında belirli yüklerin değil, tüm yüklerin taşınması için demir yolları daha elverişli hâle getirilmelidir.

Demiryolu yük taşımacılığı, son dönemde köklü bir değişime uğramış olup ağırlıklı olarak diğer taşıma türleriyle entegrasyonun sağlandığı bir sistem haline dönüştürülmüştür. Özellikle hızlı trenler, yeni vagon tipleri ve entegre teknolojilerle birlikte diğer taşımacılık türlerine alternatif olarak sunulmaktadır. Dünyada demiryoluna olan talebin artmasında, hatların üretim ve tüketim merkezleri ile birlikte kombine taşımacılık operasyonlarına elverişli hale gelmesi temel rol oynamaktadır. Uluslararası yasal düzenlemelerdeki basitleştirme ve uyum çalışmaları ile çevre güvenliğine yönelik ortak mutabakatlar, bu alana olan talebi artırmaktadır [6].

Demiryolu taşımacılığının diğer taşıma türlerine göre üstün yönleri

- Enerji tüketimi açısından diğer ulaşım sistemlerine göre daha avantajlıdır. Büyük miktarda yük ve yolcu taşınması ile birim başına yakıt tüketiminin az olması çevreye daha az zarar vermesini sağlamaktadır.
- Diğer taşıma türlerine göre daha güvenlidir. Küçük çekiş gücü ile emniyetli yüksek hızlara ulaşılabilir.
- Kara yolu trafik yükünü hafifletir.
- Genelde diğer taşıma alternatiflerinin aksine uzun dönem sabit fiyat garantisi vardır.
- Uluslararası geçişlerde kara yolunda geçiş sınırlamaları bulunurken transit ülkelerin tercih ettiği bir taşıma türü olmasından dolayı geçiş üstünlüğü verilmektedir.
- Transit süreleri kara yoluna göre biraz daha fazla olmasına rağmen sefer süreleri sabit durumdadır.
- Ağır tonajlı ve havaleli yükler için çok uygun bir taşıma türüdür.
- Büyük miktarda yük ve yolcu taşımacılığı yapılabilir.

- Vagonlar, taşınan yüke göre tasarlanabilir ve hava, ses ve ısı yalıtımları ile konforlu hizmet vermektedir.
- Hava koşullarından etkilenmez [4].

Demiryolu taşımacılığının dezavantajları

Demiryolu ulaşım sistemlerinin diğer sistemlere göre bazı dezavantajları da vardır:

- Transit süreleri bazı teslim bölgelerinde kara yolu ve deniz yoluna göre daha uzun zaman alabilmektedir.
- Özellikle Türkiye içindeki parkurda, teslimat daha uzun sürelerde yapılabilir. Örneğin; Almanya çıkışlı tren Kapıkule'ye 5 günde gelmesine rağmen, Derince' ye 3-4 günde varabilmektedir.
- Kapıdan kapıya teslimlerde Avrupa'da çoğu yerlerde fabrika içine kadar ray bağlantısı olmasına rağmen, Türkiye'de bu imkân yok denecek kadar azdır.
- Kapıdan kapıya teslimlerde çoğu yerde tren en yakın istasyona kadar gelebilmekte, fakat teslim adresi için ayrıca kamyon aktarmasına gereksinim duyulmaktadır.
- Çoğu durak yerlerinde büyük ve konforlu istasyonların yapılması gerekmektedir. Yol ve tesislerin yapımı, araçların temini ve/veya imali özel teşebbüsün kolaylıkla karşılayamayacağı büyük sermaye gerektirmektedir [6].

2.1.3. Denizyolu taşımacılığı

Türklerde deniz taşımacılığının gelişmesi, Anadolu'ya girilmesinden sonra başlar. İpek yolunu ellerinde bulunduran Türkler, az zamanda kendi denizlerine egemen olarak, deniz ticaret ve nakliyatına başlamışlardır.

Osmanlı deniz ticaretindeki bu baslangıç döneminden sonraki gelişmeleri, aşağıdaki aşamalarda incelemek yararlı olacaktır:

- Yabancı gemilerin taşıma yaptıkları devir,
- Yabancı gemilerin ve Türk gemilerin birlikte taşıma yaptıkları devir,
- Sadece Türk gemilerin taşıma yaptıkları devir.

Birinci Devir: 16.yüzyıl ile başlayan birinci devirde, Türk sularında yabancı gemilerin çalıştıklarını görüyoruz. Yukarıda ayrıntılarını açıkladığımız gibi, Kanuni Sultan Süleyman'ın Fransa Kralı I. François ile imzaladığı ticaret sözleşmesinde Avrupa ile olan Türk ticareti Fransız bayrağını taşıyan gemilere verilmişti. Bizim için kapitülasyon sözleşmesi olan, ancak Avrupa'da yayınlanan kitaplarda "modern anlamda ilk ticaret muahedesi" olarak geçen bu imtiyazdan yararlanılarak, Türk limanları arasında da Fransız gemileri işletiliyordu. Diğer ülkeler de Türkiye ile yaptıkları ticarette Fransız bayrağı taşıyan gemileri kullanmak zorunda idiler. 17.yy.da bu ülkelere ek olarak Avusturya, Rusya, İsveç, İspanya ve sonraları Prusya da, Babiâli'den Türkiye'ye ait olan ticaretlerini kendi gemileriyle yapma iznini almışlardır.

İkinci devri, 19.yüzyıldan başlatmak mümkündür. Bu devirde, yabancı gemiler yanında Türklerin de gemi işletmeye başladığını görüyoruz. Bu şekilde Türkiye kıyılarında ilk buharlı gemi 1828 senesinde görülmüş olup, Tanzimat'tan bir kaç sene sonra 1844'de Bahriye Nezareti tarafından İngiltere'den getirilen bir gemi ile (Seyr-i Bahri), İstanbul-Gemlik-İzmit ve Tekirdağ hattı açılmış bulunuyordu. Aynı sene içinde bir gemi daha getirtilerek (Eser-i Hayr) Bogaziçi'nde çalıştırılmaya başlanmıştır. "Hazine-i Hassa Müzesi" adı verilen bu işletme; işletmecilikte özerk bir yönetimin kurulmasını gerektirdiğinden, bunun için yine Bahriye Nezaretine bağlı olmak üzere 1843 Fevaid-i Osmaniye adı altında bir yönetim kurulmasını sağlamıştır. Fakat çürük bir kaç gemi ile ise başlayan ve önemli başarı gösteremeyen, özellikle yabancı rekabetine dayanamayan bu yönetim, bir

kaç sene sonra ayrıcalığı ile bir Fransız şirketine devretmeye mecbur kalmış ancak Fransız şirketi de bu işin üstesinden gelememiş ve işletme yine Türk yönetimine geri verilmiştir. Sultan Aziz devrinde, 1871'de İdare-i Aziziye ismini alan bu teşebbüsün adı 1878'de İdare-i Mahsusa olmuş, nihayet 1910 yılında Ticaret Nezareti'ne bağlı Osmanlı Seyr-i Sefain İdaresi'ne dönüşmüştür. Yine aynı dönemlerde, 1851 senesinde İstanbul ile Bogaziçi ve Kadıköy arasında gemi işletmek üzere Sirket-i Hayriye adında bir özel teşebbüs kurularak, başarılı hizmetler vermiştir.

Üçüncü devir, sadece Türk gemilerinin taşıma yaptıkları dönem olup bu aşamayı I.Dünya Savaşı (1914) ile başlatmak mümkündür. Gerçi savaşta kapitülasyonlar kaldırılmış ve denizciliğimiz de serbest alan bulmuştu. Fakat savaş senelerinde mevcut gemilerinde birçoğu düşman tarafından batırılmış olduğundan, ticaret filomuz daha da küçülmüştür. I.Dünya Savaşı'ndan önce, yaklaşık 110 000 tona ulaşan gemilerimizden, toplam tonajı 83 600 olan 63 Türk gemisi savaşta batırılmıştır. Mütareke senelerindeki istikrarsız durum, gemilerin daha da fazla eksilmesine neden olmuştur. Cumhuriyet dönemi de, özellikle kabotaj hakkının elde edilmesiyle, bu üçüncü devir içinde değerlendirilir. Ticaret filosunun büyütülmesi 1921–1962 döneminde daha hızlı olmuş, devlet deniz taşımacılığı büyük bir ivme ile gelişmiştir. 1945 yılında Ulaştırma Bakanlığı bünyesinde Liman ve Deniz İşleri Daire Bakanlığı'nın yanı sıra, İstanbul Şehir Hatları İşletmesi ve Devlet Demiryolları ve Limanlar İşletme Müdürlüğü kurulmuştur. 1950'li yıllara ulaşıldığında özel kesim deniz taşımacılığı kamu taşımacılığı kapasite yönünden geçmiştir. 1958 yılında Uluslararası Denizcilik Teşkilatı'na (IMO) üye olan Türkiye, Ulaştırma Bakanlığı'nın koordinatörlüğünde, denizciliğin hukuki, ekonomik ve teknik gelişimine uyum sağlamak için uluslararası kuruluşlar (IMO, OECD, UNCTAD) bünyesinde yapılan çalışmalara katılmıştır. Birinci Beş Yıllık Kalkınma Planı yürürlüğe girerken, 1962 yılında, deniz taşımacılığının kapasitesinin yüzde 40'ı kamu kesimine, yüzde 60'ı özel kesime aitti. Ancak ticaret filosunun gençleştirilme ihtiyacı vardı. Tersane kapasitelerinin yüzde

95'i kamuya, yüzde 5'i özel kesime aittir. Ülkenin ihtiyaç duyduğu gemilerin yapımına teknik yönden yeterlilik taşıyan tersaneler, eksiklerinin tamamlanması halinde, yaşlılıkları nedeniyle servisten çıkartılacak gemilerin yerlerine yenilerini yapmaya, ticaret filosunu gençleştirmeye elverişlilik göstermekteydi. Kullanılmış gemi fiyatlarının düşük olması yanında, yurt dışında gemi inşa süresinin kısalığı, ithalatın kullanılmış gemilere yönelmesine neden olmuştur. 1966 yılında Gölcük Deniz Kuvvetleri Tersanesi'nin artan kapasitesinin ticari gemi inşasına ayrılması sağlanmış, böylece büyük gemi yapımı olanakları hazırlanmıştır. Fakat mevcut kapasiteler yeterince kullanılamamıştır. Üçüncü Plan Dönemi'nde gemi dizel motorları fabrikasının kuruluş çalışmaları son aşamasına gelmiş, özel kesim Tuzla Tersanelerinin alt yapı çalışmalarına, Pendik Tersanesi'nin yapımına ve Alaybey Tersanesi'nin "Türkiye'nin en büyük bakım-onarım tersanesi konumuna ulaşması için" gerekli düzenlemelere başlanmıştır. Yine bu dönemde, Denizcilik Bankası'nın bazı tersaneleri ile Deniz Kuvvetleri tersanelerinden ticari gemi yapımına ayrılan süreler büyük ölçüde kullanılabilmiştir. Kazanılan deneyimler, elde edilen bilgi birikimi, iş gücü ve ucuzluğu, kurulu ve kurulmak üzere olan kapasiteler, gemi yapımında Türkiye'nin önemli bir potansiyele kavuştuğunu ortaya koymaktadır. Bunun yanında, Türkiye beklenmeyen bir transit deniz trafiğiyle karşılaşmış, bu da limanlarda darboğaz yaratmış, yeni altyapı düzenlemelerini zorunlu kılmıştır.

1981 yılında denizciliğe dinamizm kazandırabilmek için İstanbul Deniz Ticaret Odası kurulmuştur. Bunu izleyen yıllarda açılan subelerle DTO'nun faaliyet alanı tüm kıyılarımızı kapsayacak biçimde genişletilmiştir. 1983 yılına gelindiğinde, deniz ticaret filosu, dış ticaret ürünlerinin ancak yüzde 47,4'ünü taşıyabilmektedir. Dördüncü Beş Yıllık Kalkınma planından başlayarak, ticaret filosunun yaş ortalamasının küçültülmesi, yaşlı gemilerin servisten çıkartılması politikası da son bulmuştur. Limanların iyileştirilmesine, araç ve gereç eksiklerinin giderilmesine, modernizasyonuna yönelik birinci proje beşinci plan döneminin ilk yılında tamamlanmış bulunmaktadır. Bu nedenle

limanların yükleme ve boşaltma kapasiteleri arttırılmıştır. Denizcilik sistem ve hizmetlerini, ülkenin çıkarlarına ve ihtiyaçlarına uygun olarak tahsisi ve geliştirilmesi amacıyla 19.08.1993 tarih ve 491 sayılı kanun hükmünde kararname ile Başbakanlığa bağlı Denizcilik Müsteşarlığı kurulmuştur [7].

Uluslararası anlamda en yaygın kullanılan taşıma türüdür. Taşıma modları arasında en düşük maliyetli ve güvenli olanıdır. Çok büyük miktardaki ürünler (kuru yük, likit ve gaz, ekonomik değeri az ürünler, hammaddeler) ile konteynerlenebilen ürünler denizyolu ile taşınır. Havayoluna göre 22, karayoluna göre 7, demiryoluna göre 3,5 kat daha ucuz olmasından dolayı dünyada en çok tercih edilen ulaşım şeklidir [4]. Deniz taşımacılığı, dünya ticaretinin % 92'sini omuzlamaktadır. Ülkeler ve kıtalar arasında taşınan yük miktarı da yedi milyar tondur [8].

Tarihsel gelişim içerisinde, limandan limana deniz taşımacılığı, günümüzde kapıdan kapıya teslimin yani tedarik zincirinin ayrılmaz bir parçası haline gelmiştir. Dünya ticaretinin büyümesine paralel olarak denizyolu eşya taşımacılığı da hızla gelişmekte, her geçen gün yeni boyutlar kazanmaktadır. Denizyolu ile taşımacılığın temel eğilimleri, taşıma araç ve kapasitelerinin artması, terminal ve liman işletme anlayışının değişmesi, lojistik hizmetlerdeki çeşitlilik ve profesyonelliğin yaygınlaşması, iletişim teknolojilerinin kullanımı ve diğer taşımacılık türleriyle entegrasyon gelmektedir. Denizyolu yük taşımacılığı bileşenleri, taşıma araçları ve gemiler, terminal ve limanlar ve taşınacak yüklerdir. Bu bileşenlerden en önemlisi olan gemiler, bağımsız kendi başına yer değiştirebilen büyük boyutlu ve belirlenen amaçlara uygun olarak kullanılan denizyolu aracıdır. Ticaret gemilerinin, yükleme boşaltma donanımı, ambar yapısı ve motor aksamı taşıyacağı yükün özelliklerine göre tasarlanır. Ticaret gemiler, taşıma üniteleriyle yük taşıyanlar (konteyner ve ro-ro gemileri) ve dökme yük gemileri (genel kargo, tanker, dökme yük ve sıvı ve gaz halindeki yükler) olarak iki gruba ayrılır [4].

Liman ve konteyner terminalleri ile depolar çok çeşitli tip ve büyüklükte olabilmektedir. Bu çeşitlilik, kapasite ve işlevsellikten kaynaklanmaktadır. Gümrük süreçleri, transit geçiş noktaları, yüklerin bir arada toplanması, boş konteyner depo alanları gibi farklı operasyonların etkin, verimli ve düzen içerisinde gerçekleştirilebilmesi açısından liman ve terminaller büyük önem taşımaktadır. Limanlar gemilerin, yük ve yolcu indirip bindirme, yükleme boşaltma, bağlamaya elverişli, yeterli su derinliğine sahip, teknik ve sosyal altyapı tesisleri, yönetim, destek, bakım onarım ve depolama birimleri bulunan doğal ve suni olarak rüzgâr ve deniz etkilerinden korunmuş taşıma türleri arasında dönüşüm noktaları olan kıyı yapılarıdır.

Denizyolu ile taşınacak yükler, çeşitli tiplerde olup, kuru ve sıvı yükler olarak iki ana gruba ayrılabilir. Sıvı yükler, petrol, kimyevi ve bitkisel yağlar, gazlar vb. olarak sıralanabilir. Kuru yükler ise, maden cevheri, hububat gibi ambalajsız dökme yükler olabileceği gibi ambalajlı yükler de olabilir. Yük çeşitleri, gemi ve liman seçiminde belirleyici unsurdur. Kuru yükler, aynı tür veya karma eşyalardan oluşabilir. Yükler ambalajsız olarak doğrudan geminin ambarına yüklenebileceği gibi, çuval, karton kutu, sandık, palet veya konteynerlere konulara da yüklenebilir.

Taşıma üniteleri sandık, palet ve konteyner olarak üçe ayrılır. Sandıklar, standart boyutlarda olmadıklarından en az kullanılan taşıma ünitesidir. Paletler ise, taşımayı kolaylaştırma amacıyla yapılan, yüklerin ünite haline getirilerek tek bir birim olarak taşınmasını sağlayan alçak platformlardır. Paletler, indirme bindirme işlemlerinin forklift gibi araçlar sayesinde daha hızlı gerçekleştirilmesini sağlar. Uluslararası Standartlar Örgütü, konteyner paletleri için standart boyutlar kabul etmiştir. Bunlar 8x100, 80x120, 100x120, 120x160 ve 120x180 cm boyutlarındadır. Yükün satıcı ve alıcı arasında kesintisiz ve en seri şekilde sevk edilebilmesi için, üniteleri birbiriyle bütünleşmesini ve standart palet boyutlarını zorunlu kılmıştır. Avrupa Ambalaj Federasyonu, 80x120 ve 100x120 cm boyutlu paletlerin kullanımında ısrar

etmiş ve yaygınlaşmalarını sağlamıştır. Konteynerler, dayanıklı ve uzun ömürlü olmaları sebebiyle, taşıma sürecinin etkin ve verimli hale gelmesinde önemli bir işlev üstlenmektedir. Konteynerler, farklı yüklerin, tek bir yükleme ve taşıma ünitesi haline gelmesine yardım eder; taşıma ünitesi olarak indirme ve bindirme işlemlerini olanaklı kılar; bir araçtan diğerine teknik araçlarla aktarılabilir; uluslararası standart ölçüleriyle taşımanın seri olmasını sağlar. Konteynerler işlevlerine göre, parça yük, dökme yük, izole ve özel amaçlı olarak sınıflandırılabilir [4].

Türkiye hem jeopolitik hem de coğrafi konum itibarı ile deniz taşımacılığına çok uygun olmasına rağmen yanlış yatırımlar nedeni ile bu avantajlarını tam olarak kullanamamaktadır. Son yıllarda deniz ticaretimizi güçlendirmek adına bazı düzenlemeler yapıldıysa da hâla yetersizdir. 2000 yılı içinde Türkiye'nin toplam dış ticaretinin %90'ından büyük bölümü denizyolu ile taşınmıştır. Bu taşımacılıkta denizyollarının aldığı pay %30,5 ile 36 milyon tondur. Yabancı bayraklı gemilerin taşıdığı 82 milyon ton yük için ödenen navlun 2,5 milyar dolar civarındadır [13]. Ro-Ro taşımacılığı gibi, doğru yatırım ve vizyonlarla ülkemiz kısa sürede asıl olması gereken güce kavuşacaktır. Eğer denizcilik sektörü dinamikleri harekete geçirilirse, orta vadede 20 – 25 milyar dolarlık kaynak sağlama potansiyelini bünyesinde barındırmaktadır [9].

2.1.4. Havayolu taşımacılığı

Türkiye'de ilk havacılık çalışmaları, 1912 yılında, bugünkü Atatürk Hava Limanının hemen yakınındaki Sefaköy'de, tesis olarak iki hangar ve küçük bir meydanda başladı. 1925 yılında, daha sonra Türk Hava Kurumu adına alacak olan "Türk Tayyare Cemiyeti"nin kurulması ile Türk havacılığının kurumsal temelleri atılmıştır. 1933 yılında 5 uçaklık küçük bir filo ile "Türk Hava Postaları" adı ile ilk sivil hava taşımacılığı başlatılmıştır. 1933 yılında, Milli Savunma Bakanlığı'na bağlı olarak kurulan "Havayolları Devlet İşletme

İdaresi", Türkiye'de sivil hava yolları kurmak ve bu yolda taşıma yapmak üzere görevlendirilir.

Dünya Sivil Havacılığının hızlı bir gelişme göstermesi, teknolojinin büyük önem taşıması karşısında, ulusal çıkarların korunması ile uluslararası ilişkilerin düzenli bir şekilde yürütülmesi ve denetlenmesi için 1954 yılında Ulaştırma Bakanlığı bünyesinde kurulan "Sivil Havacılık Dairesi Başkanlığı", 1987 yılında "Sivil Havacılık Genel Müdürlüğü" olarak günün koşullarına göre yeniden teşkilatlandırıldı. Sivil Havacılık Genel Müdürlüğü, Ulaştırma Bakanlığının Ana Hizmet Birimi olarak 5431 Sayılı, Ulaştırma Bakanlığının Teşkilat ve Görevleri Hakkında Kanun çerçevesinde görev yapmaktadır. Sivil Havacılık Genel Müdürlüğünce, 2920 Sayılı Türk Sivil Havacılık Kanunu ve bu kapsamda yayımlanmış olan yönetmelik ve Havacılık Talimatı çerçevesinde hizmet verilmektedir.

Türkiye'nin halen üyesi olduğu Uluslararası Sivil Havacılık Teskilatı (ICAO), Avrupa Sivil Havacılık Konferansı (ECAC), Avrupa Havacılık Seyrüsefer Güvenliği Teşkilatı (EUROCONTROL) ve Avrupa Havacılık Otoriteleri Birliği (JAA) bünyesinde yapılan çalışmalar yakından izlenmektedir. 04.04.2001 tarihi itibarıyla JAA'ye tam üyelik hakkı kazanılmıştır. Bugün yaklaşık 50 ilde çeşitli büyüklükte, 21'i dış hat ve iç hat, 31'i sadece iç hat trafisine açık, 4'ü THK'na bağlı toplam 12 özel statülü 64 havaalanı sivil havacılık işletmelerinin kullanımına açıktır [10].

Özellikle 1980'lerin ikinci yarısından itibaren belirgin bir gelişme içine girmiştir. Bu dönemde THY'nin bir modernizasyon ve standardizasyon programı çerçevesinde filosunu geliştirmeye başladığı, hizmet standartlarını yükseltme çabasına girdiği ve yurtiçi hatlardan ziyade ekonomik açıdan avantajlı dış hatlara yönelmekte olduğu görülmektedir. Aynı dönemde, özel sektör havayollarının sayılarında, filo kapasitelerinde ve sektörden aldıkları payda önemli artışlar gözlenmektedir. İşletme sermayesi sıkıntısı, nispeten

yaşlı uçaklarla operasyon yapma dezavantajı, bakım-onarım ve diğer altyapı imkânlarının yetersizliği, faaliyetlerinin her kademesinde kalifiye personel temininde karşılaşılan güçlükler, sektörün yeteri kadar desteklenmemesi, bürokratik işlemler özel havayollarının genelde karşılaştıkları sorunlar olmuştur.

En temel tercih edilme sebebi hız faktörüdür. Bunun yanında birim taşıma maliyetleri, diğer taşıma türlerine nazaran oldukça yüksektir. 500 kilometreden daha uzak mesafeler için elverişlidir. Fakat havayolu nakliyesinin sağladığı hız avantajı aynı zamanda depolama maliyetlerini azaltıcı bir etken olarak düşünülebilir [11].

Bununla beraber, küreselleşme ile birlikte global şirketler için maliyet ile birlikte zaman ve hız kavramları da önem kazanmıştır. Bu sebeple havayolu taşımacılığının, entegre taşıma içeririnde rolü her geçen gün artmaktadır. Son on yılın istatistiklerine göre, dünyada taşınan toplam kargonun tonaj olarak sadece % 2'si havayolu taşımacılığı ile gerçekleştirilmiştir. Diğer yandan, havayolu ile taşımaların dolar olarak değeri toplam kargonun dolar olarak değeri içinde % 33'lük paya ulaşmıştır. Bu veri, havayolu taşımacılığındaki yüksek maliyete rağmen buna katlanıldığını göstermektedir. Türkiye'de ise havayolu taşımacılığı uzun yıllar neredeyse devlet tekeli altında yapıldıysa da 2003 yılında yapılan düzenlemeler ile bölgesel taşımacılığı önü açılmıştır. Bu düzenlemeden sonra, bölgesel yolcu taşımacılığında büyük artışlar kaydedilmiştir. Ayrıca yeni firmaların sektöre girişi hızlanmıştır. Sektöre giren şirket sayısı arttıkça da bu firmaların eğilimi yerel olarak belirli bölgelerde uzmanlaşmaya doğru gidecektir [12].

Havayolu ulaşımı, yolcu ve yüklerin en uygun zaman ve yöntemle ulaştırılmasını sağlayan bir ulaşım sistemidir. Havayolu ulaşımının yüksek hızı sayesinde, teslim süresi kısalmaktadır. Dünya genelinde havaalanı varlığı, yüksek emniyet ve güvenilirlik, kargoların elleçleme ve yüklenmesinde

gösterilen özen, planlı ve tarifeli hareketler ve diğer taşımacılık türlerine göre sigorta primlerinin daha düşük olması bu ulaşım sisteminin avantajlarıdır. Bununla birlikte, uçakların hacim ve ağırlık sınırları, diğer taşıma türlerine göre daha yüksek taşıma maliyetlerinin oluşmasına sebep olmaktadır.

Yük taşımacılığında, hava yolunun tercih edilme sebepleri:

- Çabuk bozulabilir kargonun en uzun mesafelere dahi kısa sürede taşınmasını sağlar.
- Güncelliklerini korudukları sürece bir ticari değere sahip oldukları kabul edilen ve satışları mümkün olan kargoların (gazete, dergi, vb) tam zamanında ulaştırılmasını sağlar.
- Değerli kargoların güvenli taşınmasına imkan verir.
- Kısa sürede ulaşmadıkları takdirde işletmeler açısından ağır ekonomik kayba yol açacak acil kargo için en hızlı seçenektir.
- Havayolu taşımacılığında sarsıntı veya çarpma gibi nedenlerle oluşabilecek hasarın en az seviyede bulunmasından dolayı, ölçüm ekipmanları, elektronik ve optik cihazlar gibi çok hassas kargoların taşınmasında güvenli koruma sağlar.
- Özellikle havaalanının büyük ekonomik merkezlerin yakınında bulunması durumunda taşıma öncesi ve sonrası akış maliyetleri azalmaktadır [4].

Havayollarında kargo tarafı incelendiğinde Türkiye’de havayolu kargo talebi hızla artmasına karşın havayolu kargo hizmetleri ile ilgili alt yapıda çok önemli eksiklikler bulunmaktadır. Yurt içi hava kargo hizmetlerinin gelişmesi gerekmektedir. Ancak İstanbul dışındaki şehirlerde hava kargo terminalleri sorunu bulunmaktadır. Hava kargo taşımacılığının deniz, kara, demiryolu taşımacılığı ile entegrasyonunda da sorun yaşanmaktadır. Yolcu trafiği ve kargo trafiğinin giriş ve çıkışlarının, bağlantı yollarının aynı hat üzerinde

olması, gümrükleme kısmında yaşanan bürokratik ve teknik sıkıntılar, kalifiyeli eleman bulunmama sıkıntısı önemli sorunlar arasındadır.

Havayolundaki sorunlara çözüm önerileri olarak aşağıdaki maddeleri kısaca sıralayabiliriz;

- Yeni tesislerin inşası yanında mevcut tesislerin modernleştirilmesi ve son teknolojik sistemlerle donatılması,
- Havalimanlarının emniyet sistemlerinin iyileştirilmesi,
- Türkiye’de ihtiyaç duyulan pilot sayısının karşılanabilmesi için önlemlerin alınması,
- Türk Hava Yollarının nitelikli personel ile hizmet kalitesinin artırılması,
- Hava kargo yollarında gerekli alt yapının oluşturulması,
- Bürokratik işlemlerinin kolaylaştırılmasıdır.

2.1.5. Boru hattı taşımacılığı

Boru hattı taşımacılığı uluslararası taşımacılıkta Türkiye’nin etkili olabileceği taşıma yöntemlerinin başında gelmektedir. Türkiye’den geçerek, ülkeye ekonomik ve siyasi güç kazandıracak uluslararası ham petrol ve doğal gaz boru hattı yatırımları önem arz etmektedir.

Gelecek 25 yıl içerisinde %60 oranında artması beklenen dünya enerji tüketiminin büyük bir bölümünün içinde bulunduğumuz bölgeden karşılanacağı öngörülmektedir. Dünya petrol ve doğal gaz rezervinin yaklaşık %75’i Orta Doğu, Avrupa, Rusya ve Orta Asya ülkelerinde bulunmaktadır. Orta Asya’daki rezervler dünya enerji talebini karşılamada önemli bir alternatif kaynak olarak ortaya çıkmaktadır. Türkiye, gerek coğrafi, gerekse jeopolitik konumu ile Orta Doğu ve Orta Asya’nın üretiminin dünya pazarlarına ulaşmasında hem bir köprü hem de bir terminal olma özelliğini taşımaktadır.

Bakü-Ceyhan projesinin tam olarak işlerliği ile 45 milyon ton petrol Akdeniz'e taşınacaktır. Hazar havzasındaki petrolün Türkiye üzerinden Akdeniz'deki terminale getirilerek buradan dünya pazarlarına ulaştırılmasını öngören Bakü –Ceyhan boru hattı projesi Türkiye için önemli bir projedir. Projenin gerçekleştirilmesi ile 45 milyon ton petrol Akdeniz'e taşınacaktır. Stratejik ve ekonomik açıdan Türkiye'ye çıkar sağlayacak bir oluşumdur. Bakü-Tiflis-Ceyhan Ham Petrol Boru Hattı (BTC) projesi 4 Haziran 2006 tarihinde Ceyhan İhraç Terminali'nden yüklenen ilk petrol tankeri ile işler hale gelmiş olup, Azerbaycan petrolünün uluslararası pazarlara taşınması başlamıştır.

Diğer bir önemli gelişme ise, Kazakistan petrolünün BTC Ham Petrol Boru Hattı üzerinden Ceyhan'a taşınmasına ilişkin hükümetler arası anlaşmanın Kazak ve Azeri Devlet Başkanları arasında 16 Haziran 2006 tarihinde imzalanmış olmasıdır. Söz konusu Anlaşma yılda 25 Milyon Ton Kazak petrolünün BTC Ham Petrol Boru Hattı üzerinden taşınmasını öngörmektedir. BTC Hattı'nın işletmeye alınması, Irak Türkiye Petrol Boru Hattı'nın tekrar maksimum taşıma kapasitesine ulaşması ve Samsun-Ceyhan Ham Petrol Boru Hattı Projesi'nin hayata geçirilmesi ile Ceyhan Terminali'nin, Akdeniz Bölgesi ve Avrupa genelinde çok önemli bir uluslararası petrol merkezi haline dönüştürülmesi ve dünya piyasalarında etkin olacak bir "Ceyhan Crude Oil Blend" yaratılması mümkün olacaktır. BTC projesine paralel olarak, Hazar Bölgesi ülkelerinde üretilecek doğal gazın Türkiye'ye, buradan da diğer Avrupa ülkelerine taşınmasını öngören Hazar-Türkiye-Avrupa Doğal Gaz 16 milyar m³ Rus doğal gazını Karadeniz altından Türkiye'ye taşıyacak Mavi Akım projesi, Türkmenistan'dan başlayarak Hazar Denizi altından Azerbaycan ve Gürcistan üzerinden geçerek Türkiye'ye ulaşacak Türkmen gazı projesi ve 10 milyar m³ kapasiteli İran-Türkiye doğal gaz boru hattı projesi de Türkiye'yi uluslararası taşımacılıkta önemli konuma taşıyacak projelerdir.

Demiryolunda olduđu gibi bu taşıma türünde de ilk yatırım maliyeti yüksektir. Genellikle ham petrol, doğalgaz gibi likit ya da gaz halinde olan ürünler taşınmaktadır. Güvenilir olması ve çok yüksek miktarda ürün taşıma imkânı vermesi büyük avantaj sağlar. Fakat esnekliđi son derece düşüktür.

Tüm dünyada olduđu gibi, ülkemizde de son yıllarda boru hatları ile taşımacılığa önem verilmeye başlanmıştır. Boru hatlarının genel karakteristiđi yanısıra, özellikle ülkemizin coğrafi konumu nedeni ile Türkiye üzerinden geçecek olan uluslararası ham petrol ve doğalgaz boru hatlarının hızla artması beklenmektedir [13].

2.1.6. İç – su (nehir) yolu taşımacılığı

Avrupa’da çok yaygın ve sık olarak kullanılmasına rağmen ülkemizde henüz emekleme aşamasındadır. Özel taşıma araçlarına ihtiyaç duyulmakta olup, araç kapasiteleri genelde suyun derinliđine göre deđişmektedir [4].

Ülkemiz akarsularının yüksek debi oranına sahip olması ve coğrafi sebeplerden dolayı taşıma yapacak yeterliliđe sahip olmamasından dolayı ülkemizde verimli şekilde faydalanmak çok fazla mümkün deđildir.

2.2. Taşımacılık Modları

2.2.1. Tek modlu taşıma (unimodal transport)

Eşyanın bir veya daha fazla taşıyıcı tarafından sadece bir taşıma türü (kara, deniz, hava, demiryolu) kullanılarak taşınmasıdır [14]. Geleneksel taşıma yöntemidir. Aynı zamanda en yaygın taşıma şeklidir. Ağırlıklı olarak kara, hava, deniz ve demiryolu ile yapılırken, iç – su (nehir) ve boru taşımacılığı şeklinde de yapılabilmektedir. Kendi aralarında avantaj ve dezavantajları mevcuttur.

2.2.2. Modlar arası taşıma (intermodal transport)

Kapıdan kapıya taşımacılık zincirinde en az iki taşıma türü (karayolu, demiryolu, denizyolu, havayolu) kullanılarak yükün nihai varış yerine ulaştırılmasıdır. Tek yükleme ile ve aynı taşıma birimi içinde ürünlerin ellenmeden, birden çok taşıma yöntemleri ile taşınması olarak tanımlanmaktadır. Bunda konteyner taşımacılığı veya treylerin hiç açılmadan karayolu, demiryolu veya denizyolu ile taşınması kastedilmektedir. Hedef, yüklemede ağzı kapatılan ünitenin teslim yerinde açılmasıdır. Boş konteynerin, boş treylerin içinde bir ürün olmadan taşınmasına, içlerinde ürün olmadığı için intermodal adı verilmemektedir.

2.2.3. Çok modlu taşımacılık (multimodal transport)

Farklı taşıma üniteleri veya aracıyla, birden fazla taşıma türü kullanılarak yapılan taşımacılıktır [14]. Örneğin, karayolu ile demiryolu hangarına gelen malzemelerin konteynere yüklenip taşınması multimodal taşımaya örnektir

1980 tarihli Birleşmiş Milletler, Çok Modlu Taşımacılık (Multimodal Transport) Konvansiyonu'na göre çoklu taşımacılık; taşımanın büyük bölümünün demiryolu, iç sular veya deniz yoluyla yapıldığı başlangıç veya nihai aşamalarında karayolunun olabildiğince kısa mesafelerde kullanıldığı çok türlü taşımadır.

2.2.4. Kombine taşımacılık (combined transport)

Eşyanın tek bir taşıma biriminin içinde yeniden yüklemeye gerek kalmadan en az iki taşıma yöntemi (kara+demiryolu veya kara+denizyolu gibi) kullanılarak taşınmasıdır [14].

Türkiye, diğer taşıma modları yanında, kombine taşıma konusunda oldukça ilerlemiş ve önemli işlerin altına imza atmıştır. Ro-Ro taşımacılığı buna en güzel örnektir. 1990'lara kadar Deniz Nakliyat bünyesinde sadece 2 gemi ile yapılan bu iş, 1992 yılındaki Yugoslavya iç savaşı ile birlikte tamamen özel sektörün öncülüğünde ilerlemeye başlamıştır. Şu günlerde 20'ye yaklaşan gemi sayısı ile kara nakliyesinde ilk tercih olmaktadır.

Ayrıca, TCDD ile Ulaştırma Bakanlığı'nın beraber yürüttüğü Ro-La projesinden de bahsedebiliriz. Projeye göre Halkalı'dan başlayıp Wels, Avusturya'ya kadar sürecek olan bir Ro-La hattı planlanmaktadır. Bu çerçevede Halkalı – Edirne arasındaki rayların rehabilitasyonu, Halkalı'da ki yükleme rampalarının modernizasyonu, ambarların tekrar inşası gibi projeler yer almaktadır.

Çizelge 2.1. Taşıma türlerinin özelliklerine göre karşılaştırılması [4].

Taşıma Türü	Maliyet	Ulaştırma Hızı	Hizmet Verilen Yerlerin Sayısı	Çeşitli Malları Kullanma Becerisi	Tarifeli Yüklemlerin Sıklığı	Tarifelerin Uygulanmasının Güvenliği
Karayolu	Yüksek	Hızlı	Çok Geniş	Yüksek	Yüksek	Yüksek
Denizyolu	Çok Düşük	Yavaş	Sınırlı	Çok Yüksek	Çok Düşük	Orta
Havayolu	Çok Yüksek	Çok Hızlı	Geniş	Sınırlı	Yüksek	Yüksek
Demiryolu	Düşük	Yavaş	Sınırlı	Yüksek	Düşük	Yüksek
Nehir yolu	Düşük	Yavaş	Sınırlı	Yüksek	Düşük	Orta
Boru Hattı	Düşük	Yavaş	Çok Sınırlı	Çok Sınırlı	Orta	Yüksek

2.3. Çok Modlu Taşımacılık

Uluslararası ticarete rekabetin artmasıyla birlikte taşımacılık ve lojistik faaliyetleri önem kazanmıştır. Taşımacılık operasyonlarının hızlı ve ekonomik olarak gerçekleştirilmesi zorunlu hale gelmiştir. Bu zorunluluk, taşıma türleri ve araçlarının teknolojik, hukuki ve işletim anlamında gelişmesine neden olmuştur. Gümrük geçişlerinin kolaylaştırılması, yasal düzenlemelerdeki uyum ve standartlaşma çabaları, taşıma türleri arasında koordinasyonun artmasına ve çok modlu taşımacılık anlayışının gelişmesine ön ayak olmuştur. Kıtalararası taşımacılıkta, iki veya daha fazla taşıma türü ve lojistik hizmetlere ihtiyaç vardır. Aynı kıta üzerinde yapılan taşımalarda da coğrafi etmenler kara, hava veya deniz taşıma türlerinin birçok kombinasyonunu gerekli kılabilir.

Artan uluslararası rekabet ortamında, taşımanın daha süratli ve ekonomik bir şekilde gerçekleştirilmesi bir zorunluluk haline gelmiştir. Günümüzde işletmeler dünya pazarlarına ulaşmak için taşımacılık hizmetlerinden faydalanmak zorundadırlar. İşletmelerin rekabet avantajlarını korumak için ürünlerini müşterilerin kapısına kadar taşıyacak ekonomik çözümlere ihtiyaç vardır.

Dünyada çevre duyarlılığının artması, devletleri bu konuda sıkı önlemler almaya itmektedir. Bu önlemlerden ulaştırma sistemleri de etkilenmektedir. Özellikle karayoluna getirilen kısıtlamalar, işletmeleri, çok modlu taşımacılık alternatiflerine yöneltmektedir. Karayolu araç trafiği yoğunluğu ve kazaların azaltılması amacıyla alınan tedbirler, taşımada yeni alternatif arayışlarını beraberinde getirmektedir.

Ayrıca yapılan düzenlemelerin ülkeden ülkeye farklılık göstermesi, transit geçişli taşımacılık alternatiflerinden kaçınılmasına yol açmaktadır. Taşınması yapılacak yükün, çıkış noktasından varış noktasına kadar birden fazla taşıma

türü ve aracının kullanılması şeklinde gelişen taşımacılık türü, modlar arası (intermodal) ve kombine (combined) taşımacılık olarak değerlendirilmektedir.

Çok modlu taşımacılık, büyük oranda yük taşımacılığı için kullanılmıştır. Bununla birlikte, yolcu taşımacılığı için de yaygın olarak kullanılabilir. Yolcuların havaalanlarına ulaşmak için, demiryolu veya başka bir ulaşım sistemi kullanması buna örnek olarak gösterilebilir. Ne var ki, kombine yolcu taşımacılığı, yani tek bilet ile iki veya daha çok ulaşım sisteminin kullanılmasının yaygın olmadığı düşünülse de, paket halinde satılan tatiller buna örnek oluşturabilir. Hızlı tren ve uçak taşıma araçlarının ardışık ve tek bilet ile kullanılmasını sağlayan kombine yolcu taşımacılığı sistemleri mevcuttur.

2.3.1. Çok modlu taşımacılığın gelişimi

1950'li yıllardan itibaren karayolu ulaşımının geliştirilmesine diğer sistemlerle göre, ağırlık verilmiştir. Bununla birlikte, karayolunun ağırlığı yalnızca yurt içi taşımacılık için söz konusuydu. 1980'li yıllara kadar uluslararası taşımacılıkta karayolunun ağırlığı fazla değildi. Bu yıllardan itibaren, karayolu, uluslar arası ticaretin de odak noktasına yerleşmeye başlamıştır. Büyüyen dış ticaret hacmi, aynı oranda gelişen bir karayolu taşıma filosunun da oluşmasını sağlamıştır.

Aynı yıllar, özellikle Avrupa Birliği ülkeleri ve diğer gelişmiş batı ülkelerinde, çevre duyarlılığının geliştiği, karayolu taşımacılığının sorgulandığı, demiryolu taşımacılığının kabuk değiştirdiği yıllar olma özelliğini taşımaktadır. Genel bir yaklaşıma göre 600 km'yi aşan taşıma mesafelerinde, demiryolu karayoluna tercih edilmeye başlamıştır. Demiryolu taşıma olanakları gelişirken, çok modlu taşımacılık anlayışı da oluşmaya başlamıştır. Demiryolu ağırlıklı olarak, deniz taşımacılığının bir uzantısı ve karayoluna ciddi bir alternatif olarak kullanılmaya başlanmıştır. İlk çok modlu taşımacılık denemesi 1985

yılında Avrupa'dan Derince'ye sıvı kimyasal madde taşımacılığıdır. Bu taşımalar, demiryolu taşımacılığının günümüzde çözümlenmemiş bir takım eksikliklerini ortaya çıkarmıştır. Bunlar; elleçleme ekipmanlarının eksikliği, tarife ve uygulama alanında yetersiz altyapı ve çok modlu taşıma mantığı ve gereğinin algılanmasında karşılaşılan zorluktur. Ne var ki, kolaycı yaklaşımlar, teknolojiye gelişim ve değişimlerden uzak olma, düşük eğitim seviyesi gibi faktörler, çok modlu taşımacılığın gelişmesini engellemiştir [15].

AB ülkeleri ve eski doğu bloğu ülkelerinin entegrasyonu çalışmalarında ulaştırmaya ilişkin yapılan düzenlemeler ve orta-uzun vadeli projeler, kara ve demiryolu çok modlu taşımacılığında demiryolunun ağırlığını arttırmayı amaçlamaktadır. UN/ECE (United Nations European Commission of Economy) tarafından oluşturulan, AGR (European Agreement on Main International Traffic Arteries), AGC (European Agreement on Main International Railway Lines) ve AGTC (European Agreement on Important International Combined Transport Lines and Related Installations) antlaşmaları ile belirlenen ulaştırma altyapısı ve hizmetlerin entegrasyonundan yük trafiğinde artış yanında yolcu trafiğinde artış da hedeflenmektedir [16].

2.4. Türkiye'de Taşımacılık

Türkiye'de yük taşımacılığının %77,5'i kara yolları, %4,5'i demiryolu, %2,5'i ise deniz yolları ve %15,5'i petrol boru hattı ile yapılmaktadır [2]. Bu dağılım ucuz ve verimli olan demir ve deniz yollarından yararlanılmadığını, bir başka tanımla bu alanda çok önemli alt yapı eksikliklerinin varlığını açıkça ortaya koymaktadır. Türkiye'nin üç tarafının denizlerle çevrili olduğu düşünüldüğünde, deniz yollarının aldığı payın düşündürücü boyutta olduğu görülmektedir. Türkiye'de iç su yollarında taşımacılıktan söz etmek ise olanaksızdır. Benzer olarak yurtiçinde hava yollarının aldığı pay da çok düşüktür. Son yıllarda sivil havacılık sektörünün

özel kesime açılmasıyla havayolu taşımacılığında olumlu yönde gelişmeler sağlanmıştır, ancak bu yeteri düzeyde değildir.

Yurtiçi taşımacılıkta denizyolları düşük paya sahip olmasına karşın dış ticarete bu taşıma türü ön plana çıkmaktadır. Denizyollarında tüm dünyada olduğu gibi konteynır taşımacılığı giderek daha yüksek pay almaktadır. Bu eğilim denizyolları taşımasında yeni altyapı ve teknoloji yatırımlarını gerekli kılmaktadır. Türkiye'nin dış ticareti içinde AB ve komşu ülkelerinin payının yüksek olması ve sektörel özellikler nedeniyle karayolu taşımacılığı da toplam dış ticaret taşımaları içinde oldukça yüksek bir pay almaktadır. Bu eğilim Türk girişimcilerin uluslararası karayolu ve kara-deniz kombine taşımacılık alanına yatırım yapmasında önemli rol oynamıştır. Türk taşıma firmaları, AB ve çevre ülke taşıyıcıları ile geçiş kotalarına karşın rekabet edilebilecek nitelik ve nicelikte taşıt filosuna sahip olmuşlardır.

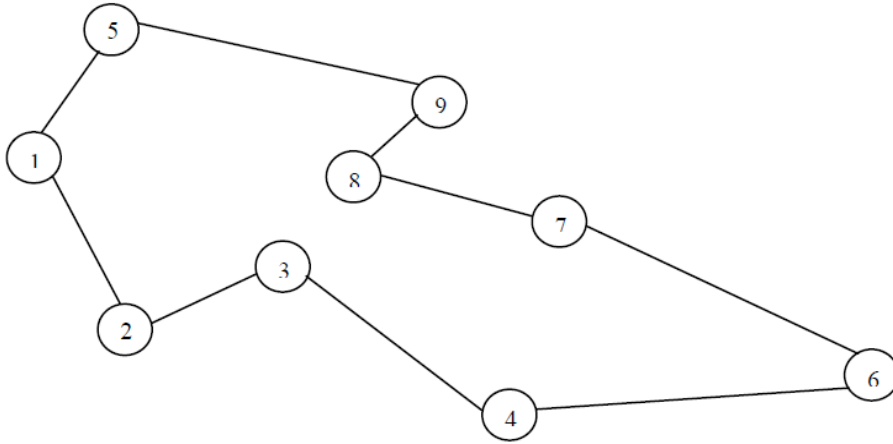
Sipariş sürelerinin kısılması, envanter maliyetleri ve özgün üretimin ön plana çıkması nedenleriyle daha küçük miktarlarda siparişlere yönelim, hızlı taşıma araçlarına talebi arttırmaktadır. Bu eğilim hava yolunun pahalı olmasına karşın giderek daha fazla kullanılmasına yol açmaktadır. Ancak hala miktar olarak havayolu taşımacılığı toplam dış ticaret taşımaları içinden çok düşük pay almaktadır. Taşımacılık türlerinin Türkiye'nin coğrafi konumu ve özellikleri nedeniyle kullanabilmesi, gerek taşımacılık gerekse lojistik sektörüne önemli üstünlükler kazandıracaktır.

Bu çalışmada Türkiye'de yurtiçi taşımacılık için, yukarıda bahsedilen nedenlerden dolayı karayolu ve demiryolu taşımacılığını içeren çok modlu taşımacılık üzerine çalışılmıştır.

3. GEZGİN SATICI PROBLEMİ

GSP, aralarındaki uzaklıklar bilinen belirli sayıda noktanın (şehir, parça veya düğüm gibi) her birisinden yalnızca bir kez geçerek başlangıç noktasına dönen en kısa veya en az maliyetli turun bulunmasını hedefleyen bir problemdir [17]. Burada bahsedilen maliyet değeri uzaklık, zaman ya da para gibi unsurlardan biri olabilir. Şebeke teorisi içinde ise GSP, “Verilen bir ağırlıklı şebekede (düğümler yani köşeler şehirleri; kenarlar ise şehirlerin arasındaki yolları göstermek; ağırlıklar da yolun maliyeti veya uzunluğu olmak üzere), en düşük maliyetli Hamilton Çevriminin bulunması” şeklinde tanımlanabilir [17].

Kombinasyonel problemler arasında en çok bilinenlerden biri olan GSP, yapısı nedeniyle doğal olarak ağ teorisinin de kapsamına girmektedir. Ağ teorisinde, sınırlı sayıda V düğümleri ve bu düğümler arasındaki A bağlarının bulunduğu bir $G(V,A)$ ağında, bu düğümleri birbirine bağlayan en kısa Hamilton turunun bulunması şeklinde anlatılabilir. İlk kez 19. yüzyılda yaşamış İrlandalı bir matematikçi olan Hamilton tarafından tanımlanan tur, bir ağdaki tüm düğümleri içeren bir çevrim (cycle) olarak tanımlanır. Burada çevrim, bir düğümden başlayarak belirli sayıdaki düğüme sadece bir defa uğrayarak, başlangıç düğümüne dönen bağlar tarafından oluşturulan bir yoldur. Hamilton turunun tanımında bağların uzunluğu önemli değildir. Belirli sayıdaki düğüm ve bu düğümler arasında sınırlı sayıda bağın bulunduğu bir ağda, bir Hamilton turunun olup olmadığı başka bir kombinasyonel problem çeşididir.



Şekil 3.1. Örnek gezgin satıcı turu

GSP gerçek yaşamda, devre kartlarında delikler açılması, robotların depolar ve üretim istasyonları arası yönlendirilmesi, büyük ölçekli devrelerin tasarlanması, tek makinede ürün değişiminde ayar süreleri en küçüklenmesi, lojistikte çeşitli dağıtım ve satın alma problemleri gibi problemlerde problemin tamamı ya da bir bölümü olarak karşımıza çıkabilmektedir. Sıralanan alanların dışında değişik kaynaklarda [18] birçok örnek yer almaktadır. Ancak, GSP'nin önemi, gerçek yaşamda çözümlediği problemlerin öneminden çok kombinasyonel eniyileme çalışmaları içerisindeki yerinden kaynaklanır. GSP, kombinasyonel eniyileme çalışmalarında geliştirilen yöntemler için en çok kullanılan başarımlı değerlendirme araçlarından biri olmakla birlikte, kendisi için geliştirilen birçok yöntem, diğer yöneylem araştırması problemleri için istenen çözümlere ulaşılabilmesine de olumlu etkiler yaratmaktadır.

GSP için geliştirilmiş ya da GSP üzerinde uygulanan birçok çözüm yöntemi bulunmaktadır. Bu yöntemlerin bazıları analitik süreçler yardımıyla, en kısa tura ulaşmayı garantilemeye çalışmaktadır. Tam sayma, dinamik programlama, dal ve sınır, dal ve kes gibi yöntemler bu grupta yer almaktadır. Bu tür yöntemler en kısa turun kesinliği karşılığında uzun süreler

alabilmektedir. Süre gereksinimini en aza indirebilmek için bazı yaklaşımlar geliştirilmiştir.

Çözüm yöntemlerinin bazıları özellikle GSP için geliştirilmiştir. Oldukça kısa sürede çözüm elde edilmesine imkân veren bu tür çalışmalardan elde edilen bilgilerin diğer kombinasyonel eniyileme problemleri için kullanılması oldukça zordur. Bunun için mevcut problemin, GSP formuna dönüştürülmesi gerekmektedir. Sezgisel en iyi tur oluşturma ve geliştirme yöntemleri bu gruba girmektedir. GSP problemleri için öne çıkan diğer bir grup çözüm yöntemi meta sezgisellerdir. Genetik Algoritmalar, Karınca Kolonileri, Tabu Arama, Tavlama Benzetimi, Sinir Ağları gibi meta sezgisel yöntemler GSP için kullanıldığı gibi diğer kombinasyonel problemler için de kullanılmakta ve başarılı sonuçlar elde edilmektedir. Bu yöntemler küresel en iyiyi garanti edememekle birlikte kısa sürede tatmin edici derecede iyi çözümler vermeleri nedeniyle sıklıkla kullanılmaktadırlar.

3.1. Gezgin Satıcı Probleminin Tarihsel Gelişimi

Lawler, Hassler Whitney'in 1934 yılında Princeton Üniversitesinde yaptığı seminer konuşmasında, GSP hakkında yapılmış olan çalışmaların sonuçları ve konunun kapsamı hakkında detaylı bir bilgi verdiğini bildirmektedir [19].

Dantzig(1954), özel olarak seçtiği 49 şehrin yol uzunluklarını harita üzerinden elde edip, bu şehirlerarasında eniyi Hamilton tur uzunluğunu hesaplamıştır. GSP' nin özlü bir açıklaması ile birlikte konunun ortaya çıkışına yönelik bazı tarihsel bilgilerde verilmiştir [19].

Christofides (1971), kendisinden önce GSP'nin eniyi çözümü için önerilen minumum yayılan ağaç ve atama problemine dayalı alt sınır hesaplamalarına ek olarak, yeni bir alt sınır algoritması vermiştir. 14 test problemi için bu yeni alt sınır algoritması uygulanmış ve simetrik durumda eniyi değerin % 4,7,

asimetrik durumda ise eniyi değerin sadece %3,8 altında alt sınır değerleri elde edilmiştir[19].

Lenstra ve Rinnooy Kan (1975), GSP'nin bilgisayar bağlantısı, taşıt güzergâhı atama, kümeleme uygulamalarını açıklayıp, problemleri GSP' ye göre formüle etmişlerdir[19].

Golden (1980), GSP için önerilen çok sayıda sezgisel yöntemi 3 grupta toplayıp, büyüklükleri 25 ile 100 düğüm arasında değişen 8 test problemi ile bu sezgisel yöntemlerin performanslarını karşılaştırmışlardır. Bu deneysel çalışma sonucunda, bir karma yöntem yardımıyla eniyi değerin % 2-3'ü dâhilinde bir çözüm elde edilebileceği kararına varılmıştır. Karma yöntem tekrarlı olarak uygulandığında ise eniyi değerin % 1-2'si dâhilinde bir GSP turu bulunabilmektedir[19].

Laporte (1987), şehirleri n kümeye en az bir kez, her düğümden en çok bir kez geçen ve n küme arasında bir Hamilton çevrimi arayan bu GSP genellemesi için tam algoritma vermişlerdir. Problem, bir tamsayılı doğrusal programa göre formüle edilip, daha sonra programın gevşetilmesi ile dal ve sınır algoritması ile çözülmüştür. Bu algoritma kullanılarak 100 düğüm ve 8 kümeye kadar olan problemler eniyi olarak çözülmüştür[19].

Arthur ve Friendewey (1988), GSP' nin çözümünde kullanılan yöntemlerin performanslarının araştırılmasında güçlüğü'n nedenini, eniyi tur uzunluğu bilinen büyük problemlerde ($n>100$) eksik olusuna bağlamıştır. Bu nedenle eniyi tur uzunluğu bilinen simetrik ve Öklit test problemleri için uzaklık matrisleri oluşturan bir algoritma vermişlerdir. Bu algoritma ile oluşturulan problemlerin rastgele oluşturulan problemler kadar zor olduğu belirtilmektedir[19].

Ong ve Huang (1989), birim alandaki n nokta arasında beklenen tur uzunluğunun, asimtotik olarak n ' nin karekökü ile orantılı olduğunu seçilen sezgisel yöntemler için bir deneysel çalışmayla göstermişlerdir. Her bir sezgisel yöntem yardımıyla bulunan tur uzunluğuna n ' nin karekökünün bir doğrusal fonksiyonu yardımıyla yaklaşılabilmektedir. Model parametreleri en küçük kareler yöntemiyle tahmin edilmiştir. GSP için iyi bir sezgisel yöntemin, n değeri arttığında da aynı özelliğe sahip olacağı bildirilmiştir[19].

Laporte (1992), GSP' nin çözümüne ilişkin o yıla kadarki bazı tam ve yaklaşık algoritmaları tanıtmıştır[19].

Belmore ve Nemhmelauser (1996), problemi anlattıktan sonra konuyla ilgili temel birkaç teorem vermişlerdir. Burada çözüm yöntemleri; tur kurma, tur ilerleme ve alt tur eleme olmak üzere 3 gruba ayrılmıştır. Alt tur eleme yönteminde, uzaklık matrisinin atama problemine göre çözüldüğü, eğer bulunan çözüm tek bir tur içermiyor ise dal ve sınır yönteminin kullanılacağından bahsedilmektedir[19].

Dorigo vd. (1997), çalışmalarında Karınca Kolonisi Sistemi (KKS) kullanarak bir algoritma yardımıyla GSP' yi çözmeye çalışmışlardır. Karınca adı verilen işbirlikçi etmenler GSP için iyi çözümler bulmaya çalışmaktadırlar. Feromon adı verilen bir maddeyle doğrudan olmayan bir iletişim yapan karıncalar bu maddeyi yolların üzerinde saklayarak en çok geçilen, en iyi yolun bulunmasında yardımcı olmaktadır. Çalışmada KKS deneyler yardımıyla anlaşılmaya çalışılmaktadır[19].

Obitko (1999), genetik algoritmaları tanıtmak için hazırladığı çalışmasında gezgin satıcı probleminin genetik algoritmalarla çözümünü de örneklemiştir. Bu çalışma genetik algoritma eğitiminde kullanılmak üzere bilgilendirici bir çalışmadır. Örnekler Java Appletlerinin kullanımıyla gerçekleştirilmiştir [19].

3.2. Tur Belirleme Problemlerinin Sınıflandırılması

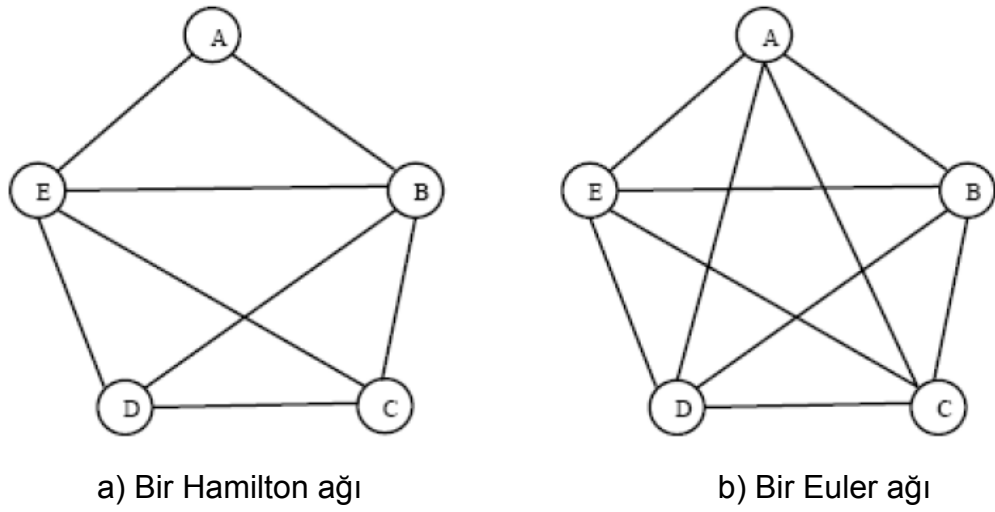
Tur Belirleme problemlerine dair çeşitli sınıflandırmalar vardır [20]. Ama genel olarak tur belirleme problemleri iki başlıkta incelenmektedir [19];

- Euler turlu problemler (Ayrıtlar için tur belirleme)
- Hamilton turlu problemler (Düğümmler için tur belirleme).

3.2.1. Euler turlu problemler

Euler turu, serim kuramının da kurucusu olarak kabul edilen İsviçreli matematikçi Leonhard Euler (1707-1783) tarafından Königsberg köprüleri üzerinde tasarlanan bir problemin çözümü için tanımlanmıştır [21]. Königsberg köprüleri problemi, eski doğu Prusya topraklarında kalan Königsberg kentinin (bugünkü adı Kaliningrad) halkı tarafından, kentin içinden geçen Pregel nehri üzerindeki 7 köprüden geçiş yapmaya dair Pazar eğlencesi olarak tanımlanmış bir oyundur. Oyundaki temel soru şudur: Acaba her köprüden yalnızca bir kere geçmek suretiyle, bir yerden başlayıp tekrar aynı yere dönülmesini sağlayacak yol var mıdır?[22]

Euler ağı (Euler's graph); ağ üzerinde kenarların tekrarlanmadığı kapalı bir dolaşımın yapılabileceği ağa Euler Ağı (Euler's Graph) denir. Euler çevrim, ağ üzerinde her kenarı kapsayan, kapalı bir gezi yapılabilecek bir yolun bulunmasıdır. Şekil 3.2' de bu ağa bir örnek verilmiştir [23].



Şekil 3.2. Hamilton ve euler ağı

Euler, bu problemi inceleyerek ilk kez bir problemin düğüm ve ayrıtlarından oluşan bir serim olarak tanımlanmasını sağlamış ve 1736 yılında yaptığı bir çalışma ile “Königsberg Köprüleri Problemi” terimini kullanarak, problemin tanımlandığı şekliyle bir çözümünün olamayacağını ispatlamıştır. Buna göre Euler turu, bir başlangıçtan yola çıkarak serimdeki bütün ayrıtlardan sadece bir kez geçip yine başlangıca dönen yoldur. Her ağda Euler turunun olması gerekmez. Euler, bunun koşullarını da incelemiş ve eğer serimdeki bütün düğümlerin dereceleri (düğüme bağlı ayrıt sayısı) çift sayı ise ağda Euler turunun olacağını, aksi halde olamayacağını göstermiştir.

3.2.2. Hamilton turlu problemler

Hamilton turu İrlandalı matematikçi William Rowan Hamilton (1805–1865) 1859 yılında tanımlanan bir matematiksel problem sonucu ortaya çıkmıştır [24]. Problem şöyle tanımlanmaktadır: Dünya yüzeyine dağılmış 20 tane şehir seçilerek bu şehirlere bir gezi düzenlenecektir. Amaç, her şehre yalnız bir kez uğramak ve geziye başlanan şehre geri dönmektir. Bu şartlar altında amaca uygun olarak hangi yollar izlenmelidir ve bu yollar nasıl belirlenmelidir?

Hamilton turu ile Euler turu arasındaki en önemli fark, problemin tanımlandığı ağlardır. Hamilton turu; uğranacak yerlerin düğüm, düğümleri birbirine bağlayan yolların da kenar olarak tanımlandığı ağlarda kullanılırken; Euler turu ise; kavşakların düğüm, uğranılması düşünülen yerlerin ise kenarlar üzerinde tanımlandığı ağlarda kullanılır [25]. Diğer bir ifadeyle, Euler turunda uğranacak yerler kenarlar üzerinde tanımlanır ve istenilen yere uğrayabilmek için ağdaki her kenardan mutlaka ve yalnızca bir kez geçilmesi istenir. Hamilton turunda ise, uğranılması düşünülen yerler düğümler üzerinde tanımlanır ve ağdaki her düğümden mutlaka ve yalnızca bir kez geçilmesi istenir. Bu sebeple, Euler turlu problemlere, kenarları gezecek gezgin için tur belirleme problemi; Hamilton turlu problemlere de, düğümleri gezecek gezgin için tur belirleme problemi demek doğru bir ifade olur [23].

Hamilton ağı; her düğümden yalnızca bir kez geçilmesi esasına dayanan ve kapalı bir dolaşım gerçekleştirilebilen ağa Hamilton Ağı denir. Hamilton çevrim, ağdaki her düğümü içine alan kapalı bir yolun olmasıdır. Gezgin satıcı problemi en az maliyetli Hamilton çevrim bulunmasına dayanan bir problem türüdür. Şekil 3.2’ de örnek bir Hamilton ağ gösterilmiştir [23].

Hamilton turunun söz konusu olduğu en temel problem, GSP dir. GSP, yöneylem araştırmasının en çok ilgi görmüş problemlerinden biridir ve bu ismi ilk olarak kimin kullandığı kesin olarak bilinmemektedir [26]. Aslında konuya olan ilgi Euler ile başlamış ve pek çok bilim adamı tarafından da incelenmiştir, ama problemle ilgili ilk ve en derli toplu çalışma 1954 yılında Dantzig, Fulkerson ve Johnson tarafından yayınlanan “Solution of a large-scale traveling salesman problem” isimli çalışmadır [19]. Bu çalışmada GSP açıkça ortaya konmakta ve en iyi çözümü bulmak için tamsayılı doğrusal karar modeli önerilmektedir.

GSP, hem pek çok uygulama alanı olan bir problem olması nedeniyle, hem de farklı pek çok problemin temelini oluşturması nedeniyle oldukça önemlidir.

Hamilton turlu diğer problemlerin tümü aslında GSP' den türemiş problemlerdir. Bunların arasında m-Gezgin Satıcı Problemi (m-GSP) ve Araç Turu Belirleme Problemi, üzerinde çok çalışılmış diğer Hamilton turlu problemlerdir. Hamilton turlu problemler, sadece bu çalışmada aktarıldığı kadar değildir. İncelenen sistemin bileşenlerindeki özel durumlardan yararlanılarak sürekli yeni problem türleri tanımlanmaktadır. Dikkati çeken nokta, yeni tanımlanan her problemin öncekilerden daha karmaşık olduğu ve çözüm sürecinin daha da zorlaştığıdır. Ama gittikçe daha özel durumların ve daha yeni yaklaşımların probleme katıldığı gözlenmektedir. Bir başka önemli noktada bilinen pek çok yöntemin bu problemlerin çözümünde kullanılmaya çalışıldığıdır. Örneğin en kısa yol, en küçük kapsayan ağaç, dal-sınır tekniği, dinamik programlama, küme kapsama ve ayrıştırma, tavlama benzetimi, modelleri, ayrıştırma ve atama problemleri, genetik algoritma, yapay sinir ağları, karınca kolonileri gibi yöntemler sıkça kullanılmaktadır.

Gelecekte hem daha karmaşık ve gerçek hayat problemlerindeki özel durumları içeren problem türlerinin geliştirileceğini, hem de yeni çözüm yöntemleri türetileceğini söylemek mümkündür. Geliştirilecek yeni yaklaşımlar var olan problemlerin daha büyük boyutlarda çözülmesine olanak tanıyacaktır.

3.3. Gezgin Satıcı Probleminin Temel Özellikleri

GSP, belirlenen sayıda şehri, her şehre yalnız bir kez uğramak şartı ile gezecek, başlangıç şehrine geri dönecek minimum yolu bulma olarak tanımlanır. GSP, eniyileme problemlerinden üzerinde en geniş çalışılmış ve en zor problemlerinden bir tanesidir. Eniyileme problemlerinde çok değişkenli fonksiyonların en küçüklenmesine ya da en büyüklenmesine yönelik etkin yöntemlerin araştırılması ile ilişkilidir. Söz konusu problemi çözmek amacıyla geliştirilen algoritmalar ve sezgisel yaklaşımlar, şu soruya cevap üretmeye çalışmaktadır; n adet şehir ve her bir şehir arasındaki mesafelerin verildiği bir

durumda, her bir şehrin sadece bir kez ziyaret edildiği ve tekrar başlangıç noktasına dönüldüğü en kısa tur nasıl oluşturulabilir?

n şehirli bir GSP'de şehirler i ile j şehirleri arasındaki mesafeyi d_{ij} ile gösterecek olursak, eğer her $d_{ji} = d_{ij}$ ise problem simetrik gezgin satıcı problemi (SGSP) olarak adlandırılır ve aşağıdaki gibi formüle edilebilir.

$$\text{Min}Z = \sum_{\forall ij, i \neq j} d_{ij} X_{ij} \quad (3.1)$$

$$\sum_{\substack{j=1 \\ j \neq i}}^n X_{ij} = 1, \quad i = 1, \dots, n \quad (3.2)$$

$$\sum_{\substack{i=1 \\ i \neq j}}^n X_{ij} = 1, \quad j = 1, \dots, n \quad (3.3)$$

$$\sum_{\substack{ni, nj \in S \\ i \neq j}} X_{ij} \leq |S| - 1 \quad \forall S \subset N; \quad 2 \leq |S| \leq n - 2 \quad (3.4)$$

$$X_{ij} = 0 - 1 \quad \forall i, j = 1, \dots, n; \quad i \neq j \quad (3.5)$$

Yukarıdaki kısıtlardan (Eş. 3.1) katedilen mesafeyi en küçükleyen; (Eş. 3.2) ve (Eş. 3.3) derece kısıtlarıdır. Bunlar her şehre sadece bir kere girilmesini ve her şehirden sadece bir defa çıkılmasını sağlar. (Eş. 3.4) ise alt tur eleme kısıtıdır. Bu kısıt S bağlı düğümler alt kümesinde sadece bir tane tur olmasını sağlar. (Eş. 3.5) ise X_{ij} 0-1 değişkenini ifade eder.

Burada eğer $X_{ij} = 1$ ise, gezgin satıcı i şehirden j şehrine gidiyor, $X_{ij} = 0$, ise şehirlerarasındaki mesafe gidilmiyor demektir [27]. GSP, şehir sayısının (n) artması ile birlikte, olası tur sayısının şehir sayısına bağlı olarak artış

göstermesi nedeniyle kolaylıkla çözülemez hale gelebilmektedir. GSP' nin çözümüne yönelik yapılan birçok çalışma olmuştur, fakat şimdiye kadar düğüm sayısı arttıkça işlem zamanının üstel olarak artmadığı bir çözüm algoritması gösterilmemiştir [28]. Problem çözüm uzayının üstel olarak artmasından dolayı NP-Complete problem sınıfına girmektedir. Bu ifade, makul büyüklükteki bir problem için ele alınması gereken çok sayıda varsayımın varlığından dolayı, eniyi bir çözüm aramanın (hesaplama maliyetlerinin çok yüksek olması nedeniyle) mümkün olmadığı problemler için kullanılmaktadır.

Bugüne kadar çözülen en büyük gezgin satıcı problemi 24,978 noktalıdır ve İsveç'te yerleşimi olan her nokta için çözülmüştür. Bu çözüm, Intel Xeon 2.8 ghz bir işlemcinin 92 yılına denk bir sürede yapılmıştır (öte yandan, 96 bilgisayarlı bir ağ üzerinde çözüldüğünden çözülmesi 3 yıl sürmüştür). Şu anda çözülmeye çalışılan en büyük problem Dünya üzerinde kayıtlı yerleşim olan her nokta için en kısa yolun ne olduğudur. Bu problem 1,904,711 şehir içermektedir [19].

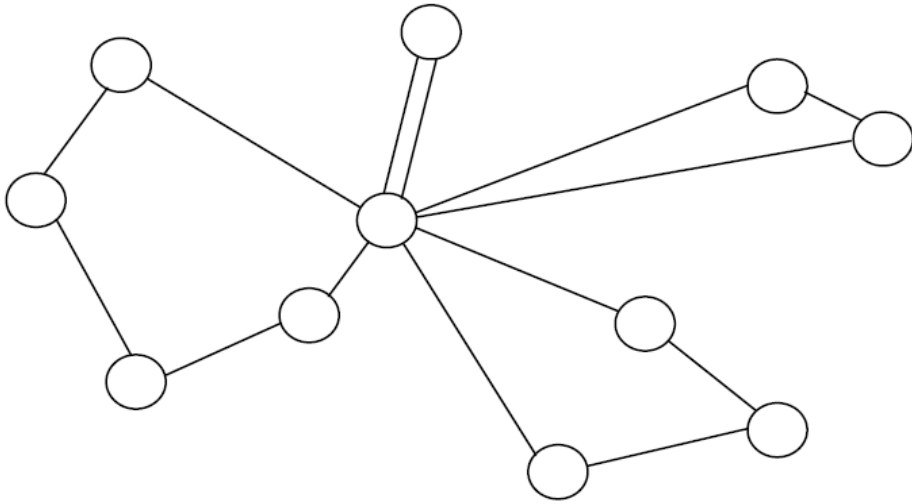
3.4. Gezgin Satıcı Problemi Çeşitleri

GSP'nin birçok çeşidi bulunmaktadır. En çok bilineni, nokta i 'den nokta j 'ye ulaşım maliyetinin, nokta j 'den nokta i 'ye ulaşım maliyetine eşit olduğu "Simetrik Gezgin Satıcı" problemidir. Gidiş dönüş maliyetlerinin farklı olması durumunda ise "Asimetrik Gezgin Satıcı" problemi olarak tanımlanır. Bu çeşit problemlerde satıcının hareket yönü önem kazanır. Blaser ve arkadaşları [29] ile Kwon ve arkadaşlarının [30] yaptığı çalışmalar bu problem türünün çözümü için değişik çözüm örnekleridir.

GSP'de, ziyaret edilmesi gereken şehirlerin yanında, satın alınması gereken bir malzeme listesi söz konusudur. Malzemeler farklı şehirlerde değişik fiyatlarla satılmaktadır. Amaç toplam yol ve satın alma maliyetini minimize

etmektir. Pearn ve arkadaşlarının yaptıkları çalışmada [31] mevcut turun geliştirilmesi için bazı öneriler sunulmaktadır.

Çoklu GSP'de (Şekil 3.3) bir ana depo ve bu depodan belli güzergahlara dağıtılması gereken mallar söz konusudur. Dağıtım belli sayıda, problemin yapısına göre sınırlı ya da sınırsız kapasiteye sahip, kamyonlar tarafından gerçekleştirilecektir. Her bir kamyonun gerçekleştireceği tur bir GSP'dir. Ancak kamyon turlarının birbirini etkilemesi nedeniyle farklı bir yapıya sahiptir. Bektaş [32] bu tip problemin çözümü için kullanılan yaklaşımları özetlemektedir.



Şekil 3.3. Çoklu gezgin satıcı (m-TSP) turu

Çinli Postacı probleminde [33], postacı mevcut ağdaki bütün yolları kullanarak ve bütün yollara bir kez uğrayarak en kısa uzunluktaki turu tamamlamak istenmektedir. Ağ teorisinde bu tur Euler turu olarak adlandırılır.

Zamana bağlı (Time dependent) GSP, Ödül Toplayan (Prize Collecting) GSP, Gezgin Vaiz (Traveling Preacher) Problemi, En Büyük (Maximum) GSP, On-line GSP, Öncelik kısıtlı GSP, Minimum gecikmeli GSP [18], Darboğaz GSP, Genelleştirilmiş GSP [34],v.b birçok gezgin satıcı problemi çeşidi bulunmaktadır.

Yukarıda sayılan problemlerin yanında ufak tefek değişikliklerle tanımlanabilecek birçok çeşit mevcuttur. Şehirlerin bir defa ya da daha çok sayıda ziyaretine izin verilmesi; gidiş dönüş uzaklıklarının farklı ya da aynı olması; aynı yolun birden çok kez kullanılması ya da kullanılmaması; bütün yolların kullanılması ya da kullanılmaması; satıcının uğradığı şehirlerde bazı görevleri yerine getirmek zorunda olup olmaması; satıcı için taşıma kapasitesinin olup olmadığı; birden çok satıcı olup olmadığı, başlangıç ve varış noktalarının aynı olup olamayacağı; bağlar ya da noktalarda olasılıklı yapının bulunup bulunmadığı; şehirlere ya da uzaklıklara ait bilgilerin zamanla değişip değişmediği; birden çok amacın bulunup bulunmadığı, gibi değişikliklerle onlarca değişik tipte gezgin satıcı problemi tanımlanabilir [34].

Bu tezde, en çok bilinen, üzerinde en çok çalışılan ancak henüz her büyüklükteki problem için makul sürede en iyi çözümü garanti eden yöntemin bulunamadığı, yukarıda tanımlanmış olan Simetrik GSP üzerinde çalışılacaktır. Çalışmanın bundan sonraki bölümünde GSP' den bahsedildiğinde Simetrik GSP ifade edilmek istenecektir.

GSP için çözümde farklılık yaratan bir başka nokta ise şehirlerarasındaki uzaklıkları gösteren matrisin yapısıdır. Uzaklık matrisi bir düzlem üzerinde yer alan noktalara ait koordinatlar arasındaki doğrusal uzaklıklarının hesaplanmasıyla oluşturulmuş ise problemin düzlemsel olduğu söylenir. Bu tür problemlerde önemli olan nokta, şehirlerarası yolların doğrusal olmasıdır. Bu tip problemler için grafik çözümler önerilebilmektedir. Bir plaka üzerinde matkapla delme problemi düzlemsel problemler için iyi bir örnek olarak gösterilebilir. Ancak gerçek yaşam problemlerinin birçoğu düzlemsel değildir. Düzlemsel yapıdaki iki boyutun yanına, şehirlerarası rakım farklarında olduğu gibi, üçüncü bir boyutun katılması da mümkündür. Bu iki tip problemde de doğrusallığın kontrolü için üçgen eşitsizliği kullanılır. Üçgen eşitsizliği basit olarak, bir üçgendeki iki kenarın uzunlukları toplamının, üçüncü kenar uzunluğundan büyük olması anlamına gelmektedir. Bu eşitsizliğin

sağlanmaması durumunda üçüncü kenarın doğrusal olmayan bir yapı gösterdiği anlaşılır. Gerçek yaşam problemlerine daha yakın olan bu tür problemlerin çözümü nispeten daha zordur. Bu nedenle geliştirilmiş olan yöntemlerin bazılarında, düzlemsellik ya da üçgen eşitsizliğini sağlama varsayımı yapılmaktadır.

3.5. Karmaşıklık Teorisi

Algoritmalar, problemlerin çözümü için geliştirilmiş adımlar bütünüdür. Karmaşıklık teorisi, problemlerin zorluk dereceleri açısından sınıflandırılması ve algoritmaların etkinliklerinin karşılaştırılması işlevini yerine getirmektedir. Karşılaştırma ve sınıflandırmalar problemlerin çözümü için gerekli hesaplama gücü açısından gerçekleştirilmektedir. Problemler genel olarak kolay, zor gibi sınıflamalara tabi tutulurlar. Algoritmalar ise problem büyüklüğü ile problemin çözümü için ihtiyaç duyulan kaynaklar arasındaki ilişkiye göre değerlendirilirler. Bu noktada bazı tanımların yapılması, konunun anlaşılması açısından fayda sağlayacaktır.

Tanım 3.1: A , X kümesinden girdiler kabul eden bir algoritma ve $f: N \rightarrow R^+$ olsun. Eğer A algoritması α bir sabit ve $\alpha > 0$ olmak üzere, her $x \in X$ için en fazla $\alpha \cdot f(\text{boyut}(x))$ işlem adımıyla çözüme ulaşıyorsa A 'nın $O(f)$ zamanda çalıştığı, zaman karmaşıklığının $O(f)$ olduğu söylenir [10].

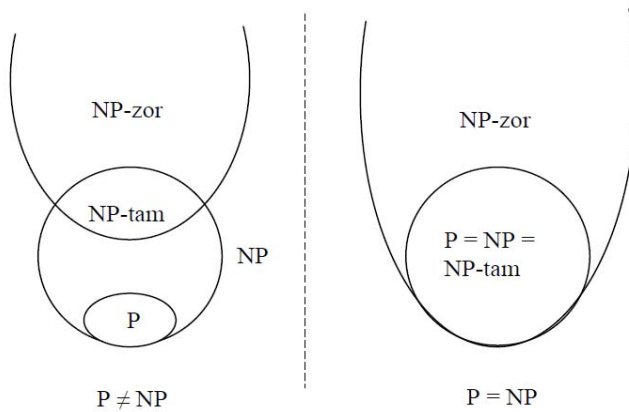
Tanım 3.2: Bir algoritma, n girdi büyüklüğü ve k bir tamsayı olmak üzere, çözümü $O(n^k)$ zamanda yapabiliyor ve hesaplamalar için gerekli veriler ortalama $O(n^k)$ bit yer tutuyor ise polinom zamanlı olarak nitelendirilir. Eğer $k=1$ olursa doğrusal zamanlı bir algoritma anlamına gelmektedir[35]. Bir algoritmanın çalışması için gerekli zaman, zaman karmaşıklığı (time complexity); gerekli bellek alanı ise yer karmaşıklığı (space complexity) hakkında bilgi verir. Her iki karmaşıklık tipi de ' $O(\)$ ' notasyonu ile

gösterilmektedir. Ancak bu notasyonunun kullanıldığı durumlarda, aksi belirtilmemişse, zaman karmaşıklığı ifade edilmektedir.

Karmaşıklık teorisinde, bazı tip problemlerin çözümü için en etkili algoritmaların; çalışma süresinin girilen verinin büyüklüğüne bir polinom cinsinden bağlı olduğu bilinmektedir (buna polinomsal zamanda çalışan algoritma adı verilir), bu tür problemler P kategorisindeki problemlerdir. Mesela verilen n basamaklı bir sayının asal olup olmadığını kontrol etmek için çalışma süresi n^6 mertebesindeki bir polinomla hesaplanabilen bir algoritma vardır. Dolayısıyla verilen bir sayının asal olup olmadığının araştırılması P kategorisinde bir problemdir. Buna karşılık bir diğer grup problem vardır ki bunlar için sorulan soruya girilen verinin büyüklüğüne polinom mertebesinde bağımlı bir sürede cevap verecek bir algoritma bilinmemektedir. Fakat bu tür bazı problemler için eğer bir şekilde cevabı tahmin edebiliyorsak, tahminimizin doğruluğunu sınamak için veri büyüklüğüne polinom mertebesinde bağımlı sürelerde çalışacak algoritmalar vardır. Tahminin doğruluğunun kontrolü için çalışma süresi, verinin büyüklüğüne polinom cinsinden bağımlı bir algoritma olan bu tür problemler de NP kategorisini oluştururlar. Örnek olarak verilen n basamaklı bir sayının asal çarpanlarının neler olduğu sorusunu düşünebiliriz. Bu sorunun cevabı için bilinen en iyi algoritmanın çalışma süresi n sayısına bir polinom cinsinden değil de üstel fonksiyonlar cinsinden (örneğin n) bağımlıdır (buna üstel zamanda çalışan algoritma denir), fakat bu problem için eğer bir şekilde cevabı tahmin edebiliyorsak tahminimizin doğruluğunu sınamak için n sayısına polinom mertebesinde bağımlı bir sürede çalışacak bir algoritma mevcuttur. Dolayısıyla verilen bir n basamaklı sayının asal çarpanlarının neler olduğu sorusu NP kategorisindedir. Bu iki kategoriden NP'nin P'yi içerdiğini görmek kolaydır. Eğer bir sorunun cevabını verinin büyüklüğüne polinom mertebesinde bağımlı sürede çalışacak bir algoritmayla bulabiliyorsak, bu soruya cevap olarak üretilmiş bir tahminin doğruluğunu da verinin büyüklüğüne polinom mertebesinde bağımlı sürede çalışacak bir

algoritmayla kontrol edebiliriz. Bunun için verilen sorunun cevabını verecek algoritmayı çalıştırıp, onun verdiği cevabı kendi tahminimizle karşılaştırmak yeterlidir. “ $P=NP$?” problemi bunun tersinin de doğru olup olmadığını sorar. Yani NP kategorisinde olup da P kategorisinde olmayan problemler var mıdır? Veya diğer bir dille asal çarpanların bulunması için polinom mertebesinde bir sürede çalışacak bir algoritma gerçekten yok mudur yoksa vardır da biz mi bulamıyoruz? Bu alanın uzmanlarının çoğunun görüşü bu tür algoritmaların gerçekten de var olmadıkları için bulunamadığı (yani P nin NP'ye eşit olmadığı) şeklindedir ancak bu soruya kesin bir cevap verilebilmesi şimdilik çok zor görünmektedir[18].

NP problemler zorluk seviyelerine göre derecelendirmeye tabi tutulmaktadırlar. Bazı problemler ise NP olmasalar da, zorluk açısından NP ile karşılaştırıldıklarından NP önadıyla anılan gruplarda yer almaktadırlar. NP-ailesi sınıflandırma sistemi Şekil 3.4’de görüldüğü gibi kafa karıştırıcı bir yapıya sahiptir. NP-zor (NP-hard) problemlerin hepsi, NP ön adını alsalar da NP değildirler. NP-zor olup, NP grubuna girenleri NP-tam (NP-complete) olarak isimlendirilir. Aslında NP ön adı NP ailesi ile zorluk ilişkisini gösterir. NP-tam problemler tam olarak NP içinde olan ve bu grupta çözülmesi en zor olan problemlerdir. NP-zor problemler ise çözülmesi en az NP kadar zor olan ancak NP içinde olmak zorunda olmayan problemlerdir[18].



Şekil 3.4. P, NP, NP-tam ve NP-zor küme çizimi [33]

Gezgin Satıcı Problemi NP-zor grubunda yer almaktadır. Problemin karar versiyonu (uzaklıklar ve bir x rakamı verilirse, x 'ten daha kısa bir tur var mıdır?) NP-tam grubunda, yani çözülmesi en zor gruptadır[18].

3.6. GSP için Geliştirilen Çözüm Yöntemleri

Yöneylem araştırmasındaki ve bilgisayar dünyasındaki her gelişme, bu problemlere yansımış ve yeni tekniklerin geliştirilmesine yol açmıştır. GSP için geliştirilen yöntemler 3 alt başlıkta incelenmiştir. Bunlar; kesin çözüm yöntemleri, geleneksel sezgisel yöntemler ve meta sezgisel yöntemlerdir.

3.6.1. Kesin çözüm yöntemleri

GSP' ye ait küresel en iyi çözümü elde etmek amacıyla kullanılabilecek kesin çözüm yöntemleridir. Yöneylem araştırması alanındaki GSP'ye benzer ya da farklı özellikteki birçok problemin çözümde de kullanılabilen yöntemlerden bir kısmının, GSP çözümünde daha etkin kullanılabilmesi amacıyla özel yaklaşımlar geliştirilmiştir. Bu özel yaklaşımlar diğer yöneylem araştırması problemleri için doğrudan katkı sağlamamalarına rağmen, problemlerin çözümü için farklı bakış açıları önermektedirler. Ayrıca belirli bir kısmı oldukça benzer yapıdaki araç çizelgeleme problemleri gibi ağ problemlerinin çözümüne doğrudan katkı sağlayabilmektedirler [18].

Dal ve sınır, kesme düzlemi gibi yöntemlerde en iyi çözüme ne kadar yaklaşıldığının belirlenmesi, çözümün en iyi olduğunun ispatlanması, çözüm sürecinin etkinliğinin artırılması gibi amaçlar için çözüme ait amaç değerinin belirli değerin altında olmayacağı ve üstüne çıkamayacağını gösteren alt sınır ve üst sınırın belirlenmesine ihtiyaç duyulmaktadır. Bu iki değer kesişmesi durumunda elde edilen çözümün en iyi olduğuna ait kanıt elde edilmiş olur. GSP gibi en küçükleme problemleri için üst sınır değeri iyi bir sezgisel yardımıyla elde edilirken, alt sınır değeri problemin gevşetmesine ait amaç

değeri ile belirlenir. Çözüm süreci bu iki değeri yaklaştırmaya yönelik olarak sürdürülür [18].

GSP’de eniyi çözüme ulaşılması çoğunlukla mümkün olmamakla beraber, 1000 civarı nokta için bilgisayar yardımı ile birkaç güne kadar eniyi sonuca ulaşmak mümkündür. Kesin çözüm yöntemi her zaman eniyi çözüm veren bir yöntemdir. Yaklaşık çözüm yöntemi ise bazı sezgisel yöntemler kullanarak hızlı bir şekilde eniyiye yakın çözümler üretmektedir.

Sayma yöntemi

GSP için küresel en iyiyi garanti eden akla gelen ilk yöntem bütün olası permütasyonların denenmesidir. Ancak n adet şehrin bulunduğu bir ağda, değerlendirilecek tur sayısının $n!$ ile orantılı olarak büyümesi nedeniyle, yöntemin zaman karmaşıklığı $O(n!)$ olarak belirlenir. Bu nedenle, az sayıda şehrin bulunduğu ağlarda bile, yıllarca süren bilgisayar hesaplamaları gerekmektedir [18].

Örneğin 20 şehrin bulunduğu bir ağda $(n-1)!/2 = 60.822.550.204.416.000$ adet farklı gezgin satıcı turu mevcuttur. Her bir turun uzunluğunun belirlenmesi ve mevcut en iyi çözümden daha iyi olup olmadığının belirlenmesi için sadece tek bir işlem yapıldığı ve 3 Ghz işlemci hızına sahip bir bilgisayarda çalışıldığı düşünülürse, denemelerin yaklaşık olarak 234 gün süreceği hesaplanır. Ancak, 21’inci şehirde bu sürenin 20 ile; 22. şehirde 20×21 ile çarpılması ve sayı arttıkça şehir sayısının bir eksiğiyle çarpılarak gerekli sürenin hesaplanması gerekmektedir. Ayrıca gerçek işlem süresinin, tek bir sıralamayı denemek için ‘ n ’ ile orantılı sayıda işlem yapılması gerektiğinden, bu sürenin en az ‘ n ’ katı olduğu düşünülürse, bu yöntemin makul sürede çözüm önermediği kolayca anlaşılabilir [19].

Dinamik programlama

Dinamik programlama yöntemi tam saymaya benzer şekilde, bir tür sayma mantığı ile çözüme ulaşmaktadır. Ancak sayma sırasında gerçekleştirilen tekrarlı hesaplamaların, işlemsel yükün bir kısmını bellek yüküyle değiştirmek suretiyle, önüne geçerek çok daha kısa sürelerde çözüm elde edilmesini sağlamaktadır. GSP'nin dinamik programlama yöntemi kullanılarak çözümü için Held ve Karp [36] tarafından önerilen yaklaşım bu alanda, bütün GSP'leri çözebilecek bir yaklaşım olarak bilenen en iyi zaman karmaşıklığına sahiptir. $O(n^2 2^n)$ karmaşıklık ile küresel en iyiyi garanti eden yaklaşım, $O(n!)$ karmaşıklık derecesine sahip sayma yönteminden çok daha iyi bir performans göstermesine karşın yine de polinomsal performans göstermekten oldukça uzaktır.

for: Bütün j ler için $C(\{1, j\}, j) = d_{1j}$.

for: $s = 3$ den n 'e kadar (alt kümelerin büyüklüğü burada belirlenmektedir)

for: S 'in n büyüklüğünde ve 1 'i içeren bütün alt kümeleri için,

for: Bütün $j \in S, j \neq 1$ için,

$$\left\{ C(S, j) = \min_{i \in S, i \neq j} [C(S - \{j\}, i) + d_{ij}] \right\}$$

en iyi = $\min_{j \neq 1} [C(\{1, 2, \dots, n\}, j) + d_{j1}]$.

Şekil 3.5. GSP'nin dinamik programlama ile çözüm algoritması

S 'in 1 şehrini ve farklı en az bir adet şehri içeren bir alt küme olduğu ve j 'nin S içinde 1 'den farklı bir şehir olduğu kabul edilsin. $C(S; j)$, 1 'den başlayan S içindeki bütün şehirleri dolaşan ve j 'de biten en kısa yolu ifade etsin. Bu durumda i ile j şehirlerini birbirine bağlayan yolun uzunluğu d_{ij} olmak üzere, en kısa tur Şekil 3.5' te görülen algoritma kullanılarak belirlenebilir.

Algoritmada her noktaya ulaşmak için geçerli en kısa yol ve uzaklık hafızada tutularak bu bilgiye ait yol her adımda bir şehir geliştirilir. İlk adımda bütün j noktalarına 1 noktasından en kısa ulaşım uzaklığı olarak $(1,j)$ uzunluğu atanır. Daha sonraki adımda ise öncelikle yoldaki şehir sayısı bir artırılarak üçe çıkarılır ve üçüncü j noktasına en kısa ulaşım uzaklığı ve yolu belirlenmeye çalışılır. Bunun için daha önce her j için hesaplanmış olan bütün diğer noktaların yol uzunluklarına üçüncü şehre uzunlukları eklenerek en kısa olanı belirlenir ve o şehir için en kısa yol ve uzunluğu olarak kaydedilir. Bu şekilde bütün şehirler yollara ekleninceye kadar devam edilir. En son aşamada mevcut en son şehirden başlangıç şehri olan 1'e uzunlukları da yol uzunluğuna eklenerek en kısa tur ve uzunluğu belirlenir [18].

Dal – Sınır Yöntemleri

Dal-sınır yöntemi arama ağacı ile ilişkili kısmi bir sıralama metodudur. Dal-sınır yönteminde ağaç kökleri bütün olası çözümleri içermektedir. Arama ağacının her bir düğüm noktası, amaç fonksiyonu değeri için daha düşük bir sınır noktası içeren çözüm kümesinin bir alt kümesidir. Alt sınır ilişkili olan gevşetilmiş doğrusal problemin çözülmesi ile elde edilir. Eniyi çözümü ararken, düğüm noktalarının artmasıyla daha dar bir çözüm kümesi elde edilir. Bu metot dallandırma yolu ile yeni arama yolları açmaktadır. Bu aramada belirli koşullar altında bazı düğüm noktaları elenir. Her bir düğüm noktası elendiğinde veya araştırıldığında işlem sona erer.

Dal-sınır yöntemi ilk defa Dantzig ve arkadaşları tarafından 1954 yılında 49 şehirlik problemi çözmek için kullanılmıştır. Dal-sınır yöntemi ilk defa Little ve arkadaşları tarafından 1963 yılında adlandırılmıştır. Dal-sınır yönteminin algoritması aşağıdaki gibidir [37].

Aşama 1: Üst limit sonsuz ($z^* = \infty$) olarak belirlenir. GSP için Q çözüm kümesi belirlenir.

Aşama 2: Eğer Q çözüm kümesi boşsa ve $z^* < \infty$ ise GSP turu enyidir. Eğer Q çözüm kümesi boş ve $z^* = \infty$ dir.

Farklı alt problem seçme kuralları, dallandırma kuralları ve gevşetme kuralları çeşitli yazarlar tarafından kullanılmıştır [38]. Pek çok yazar GSP'nin doğrusal programlama ile formülasyonunun gevşetilmesi için Atama problemi, 1-ağaç gevşetilmesi ve Lagrange gevşetilmesi gibi farklı yöntemler kullanmışlardır.

Atama Problemi

Atama problemi (AP), n tane işin m makineye atanması ve toplam maliyetin $\sum_i \sum_{j \neq i} c_{ij} x_{ij}$ minimum yapılmasını isteyen bir problem olarak tanımlanabilir.

Bu GSP için kullanılan ilk gevşetmelerden biridir. (Eş. 3.1) – (Eş. 3.4) denklemleri AP formülasyonunu sağlamaktadır. Buna göre GSP gevşetilmesi (Eş. 3.1) – (Eş. 3.5) Danztig, Fulkerson ve Johnson (1954) [39] tarafından sağlanmıştır. Asimetrik Gezgin Satıcı Problemi (AGSP) için pek çok algoritmanın temelinde AP vardır. Bu çalışmada simetrik problem ele alındığından AP gevşetmesine kısaca değinilecektir. AP kullanılarak geliştirilen algoritmaların etkin olanlarının bazıları Carpaneto ve Toth (1980) [40], Miller ve Penky (1991) [41] ve Carpaneto ve arkadaşları (1995) [42] tarafından geliştirilen algoritmalar. Aşağıda bu yazarların çalışmalarına kısaca değinilmiştir.

Carpaneto ve Toth Algoritması

Carpaneto ve Toth, AGSP için en küçük ilk dal-sınırına dayanan bir algoritma ortaya koymuşlardır. Dallandırma işlemi, amaç fonksiyon değeri en küçük olan düğüm noktasından yapılır. Her bir düğüm noktası için, örneğin h , (Eş. 3.1) – (Eş. 3.4) denklemleri dikkate alınarak ortaya çıkan problem Değiştirilmiş Atama Problemi (DAP_h) gibi çözülüp, çıkartılan kenarlar alt

kümesi E_h ve eklenen kenarlar alt kümesi I_h aşağıdaki gibi (Eş. 3.6) tanımlanmıştır:

$$E_h = \{(i, j) \in A : x_{ij} = 0\}, I_h = \{(i, j) \in A : x_{ij} = 1\} \quad (3.6)$$

Eğer DAP_h 'nin eniyi sonucu bir Hamilton turu oluşturmuyorsa ve DAP_h 'nin eniyi amaç fonksiyonu değeri, bilinen en iyi çözümden daha iyi ise, h düğüm noktasından m tane alt problem yeni bir dallandırma kuralına göre aşağıdaki gibi dallandırılır:

$G_q, q = 1, \dots, d$ DAP_h tarafından üretilen alt turlar olsun ve bir alt tur $G_q = (V_q, A_q)$, ile $V_q = \{r_{q1}, \dots, r_{qe_q}\}$ ve q . alt tur için $e_q = |V_q| = |A_q|$ durumda $A_q = \{(r_{q1}, r_{q2}), \dots, (r_{qe_q}, r_{q1})\}$ olsun. I_h kenar setinde yer almayan en az sayıda kenarlara sahip G_q alt turu dallandırmak için seçilir [40].

$$m = e_p - |A_p \cap I_h| = \min_{\{q=1, \dots, d\}} \{e_q - |A_q \cap I_h|\} \quad (3.7)$$

Önerilen bu metot ile 40 şehirden 240 şehire kadar rastgele 20 problem çözülmüştür [40].

Miller ve Penky Algoritması

Atama Probleminin duali Miller ve Penky tarafından aşağıdaki şekilde formüle edilmiştir.

Amaç Fonksiyonu

$$\max Z = \sum_{i=1}^n u_i + \sum_{j=1}^n v_j \quad (3.8)$$

Kısıtlar

$$c_{ij} - u_i - v_j \geq 0, \quad i, j \in V, i \neq j \quad (3.9)$$

Miller ve Penky, çok fazla sayıda çözümleri incelemeyi engelleyerek, etkin bir şekilde problemi çözmek için, orijinal problem ile aynı eniyi sonuca sahip olan ve daha hızlı çözülebilen daha basit bir model oluşturmuşlardır. AP' gevşetilmesi ile ilişkili $AGSP'$ için değiştirilmiş bir maliyet matrisi c'_{ij} düşünölsün:

$$c'_{ij} = \begin{cases} c_{ij} & \text{Eğer } c_{ij} \leq \lambda \\ \infty & \text{aksi halde} \end{cases} \quad (3.10)$$

$AGSP$ ve AP 'nin eniyi değeri sırasıyla $v(AGSP)$ ve $v(AP)$ olsun. Eğer

$$v(AGSP) - v(AP) \leq \lambda + 1 - u'_i - v'_{\max} \quad (3.11)$$

$$\lambda + 1 - u'_i - v'_{\max} \geq 0, \quad \forall i \in V \quad (3.12)$$

ve (u', v') dual AP' nin eniyi çözüümü ve $v'_{\max}$, v' için maksimum eleman ise, $AGSP'$ için eniyi sonuç olan z^* $AGSP$ için de eniyi sonuçtur. $\lambda + 1 - u'_i - v'_{\max}$ çıkartılan matris elemanının, hiçbir başka elemanın daha iyi bir sonuç elde edemeyeceğı en küçük azaltılmış maliyetidir. Dal-sınır metoduna, AP 'nin eniyi çözüm setinden $AGSP$ 'ye hızlı bir şekilde çözüm bulmak için bir özellik eklenmiştir. Bu özellik verilen grafikte $AGSP$ çözümünün azaltılmasını sağlamaktadır.

$\bar{G}$, V köşe noktalarına sahip ve u_i^* ve v_i^* eniyi dual AP 'nin değişkenleri olduğu, $\bar{A} = \{(i, j) | c_{ij} - u_i^* - v_j^* = 0\}$ kenarlarına sahip, $\bar{G} = (V, \bar{A})$ şeklinde

tanımlanan bir grafik olsun. $\bar{G}$ üzerindeki bir Hamilton turu AGSP için bir eniyi çözümdür. Eğer $\bar{G}$ bir Hamilton turu içermiyor ise, AP'nin alt sınırı en küçük sıfır olmayan bir indirgenmiş maliyet kadar artırılabilir. Bu özellik Dalsınır metotlarına eklenmiştir. Beş farklı özellikte rastgele üretilen problemler test edilmiş ve 500 000 şehirlik problem süper bilgisayar yardımı ile 12,623 saniyede çözülmüştür [41].

Carpaneto, Dell'amico ve Toth Algoritması

Carpaneto, Dell'amico ve Toth, Atama Problemini temel alan en küçük dalsınır metodu ve alt tur elemesi için yeni bir algoritma önermişlerdir. Bu algoritma Carpaneto ve Toth (1980) algoritmasının dört yönden geliştirilmiş halidir:

1. Kök düğüm noktasında (0. düğüm noktası), eniyi tura ait olamayacak kenarlar, azaltma yöntemi ile çıkartılır. Orijinal C matrisi, her bir matris elemanı u_i ve v_j ile azaltılan ($\bar{c}_{ij} = c_{ij} - u_i - v_j \geq 0$) $\bar{C}$ matrisine dönüştürülür.

u_i ve v_j AP ile ilişkili dual problemin eniyi çözümüdür. $\bar{c}_{ij}$ bir alt sınırı temsil etmektedir. z , AP'nin dual probleminin bir alt sınırı olsun. Eğer olurlu sonuç

z^* biliniyorsa, $\bar{c}_{ij} \geq z^* - \underline{z}$ olan, kenar eklenmesi z^* 'den daha küçük bir çözüm üretmeyen her kenar, $(i, j) \in \forall$, çıkartılır. Böylece orijinal G grafiği, daha seyrek olan $\bar{A} = \{(i, j) \in A; \bar{c}_{ij} < z^* - \underline{z}\}$ olduğu $\bar{G} = (V, \bar{A})$ grafiğine dönüşür.

2. Her bir düğüm noktasında GAP çözülür ve kullanılan algoritmanın etkinliği GAP'nin çözümüne büyük ölçüde etki eder. h düğüm noktasında, GAP_h 'yi çözmek için özel bir parametrik teknik kullanılır. Kenar alt kümesi olarak $\bar{A}$ kullanılır.

3. Karar ağacının düğüm noktası ile ilişkili bilgilerinin saklanması için etkin bir veri stoklama vardır. Bu da değişmeyen bilgileri güncellemeyi önlemektedir.

4. Birkaç eniyi sonuca sahip olan h düğüm noktası düşünölsün. Alt turların sayısını azaltan bağlantı yöntemi, aşağıda göröldüğü gibi tekrarlı bir şekilde uygulanır:

$G_a = (V_a, A_a)$ ve $G_b = (V_b, A_b)$ iki alt tur olsun. $(i_a, j_a) \in A_a$ ve $(i_b, j_b) \in A_b$, $\bar{c}_{i_a j_a} = \bar{c}_{i_b j_b} = 0$ olan iki kenar düşünölsün. G_a ve G_b aşağıdaki gibi tek bir alt tur olarak birleşebilir.

$$G_a = (V_a \cup V_b, A_a \cup A_b) = ((i_a, j_a) \cup (i_b, j_b) \cup (i_a, j_b) \cup (i_b, j_a)) \quad (3.13)$$

Eğer Hamilton turu bulunursa, bu sonuç h düğüm noktası için eniyi çözümdür. Geliştirilmiş algoritma kullanılarak 3 dakikadan kısa sürede, 2000 şehire kadar rastgele problemler çözülmüştür [42].

Ağaç Gevşetilmesi Yöntemi

$V = \{1, \dots, n\}$ köşe noktaları ve A kenar seti olan $G = (V, A)$ grafiğı düşünelim. c_{ij} , i noktasından j noktasına kenar uzunluğu olsun. Ağaç grafiğıe bir tur olmadan bağlıdır. Verilen G grafiğıne göre, minimum kapsar ağaç (minimum spanning-tree) problemi, G içerisindeki en küçük kenar uzunluğunun toplamına sahip kapsar ağacı bulma problemidir. 1-Ağaç iki ayrı kenarın bir noktada birleşmesidir. Minimum 1-ağaç problemi aşağıdaki gibi formöle edilmiştir [43]:

Amaç Fonksiyonu

$$MaxZ = \sum_{i < j} c_{ij} x_{ij} \quad (3.14)$$

Kısıtlar

$$\sum_j x_{ij} = 2 \quad (3.15)$$

$$\sum_{1 \leq i < j \leq n} x_{ij} = n \quad (3.16)$$

$$\sum_{i, j \in S, i < j} x_{ij} \leq |S| - 1, \quad S \subset \{2, 3, \dots, n\} \quad (3.17)$$

$$0 \leq x_{ij} \leq 1 \quad x_{ij} \text{ tamsayı} \quad (3.18)$$

x_{ij} ikili değişken olduğu için, (Eş. 3.15) kısıtı köşe noktası yani iki kenarın birleşmesini sağlamaktadır. (Eş. 3.16) kısıtı $\{(i, j) : x_{ij} = 1\}$ nokta setinin $V - \{1\}$ kenarları için ağaç oluşturmaktadır. (Eş. 3.17) kısıtı alt turların oluşmasını engelleyen alt tur eleme kısıtıdır. (Eş. 3.15) – (Eş. 3.18) kısıtlarını sağlayan x_{ij} çözümü eğer (Eş. 3.15) kısıtını sağlar ise GSP turu oluşturulmuş olur.

$$\sum_{j > i} x_{ij} + \sum_{j < i} x_{ij} = 2, \quad 2 \leq i \leq n - 1 \quad (3.19)$$

(Eş. 3.19) kısıtı her bir noktanın iki kenara sahip olmasını sağlamaktadır. Buna göre (Eş. 3.14) – (Eş. 3.19) denklemleri GSP için farklı bir formülasyon sağlamıştır [43].

(Eş. 3.14) – (Eş. 3.19) içeren GSP problemini çözmek için Lagrange gevşetilme tekniği kullanılmıştır. (Eş. 3.19)'daki kısıt (Eş. 3.14)'deki amaç fonksiyonuna eklenmiş ve Lagrange gevşetilmesi problemi aşağıdaki gibi tanımlanmıştır.

$$L(\pi) = \min \left\{ \sum_{i < j} c_{ij} x_{ij} + \sum_i \pi_i \left[\sum_{j > i} x_{ij} + \sum_{j < i} x_{ji} - 2 \right] : (4.16) - (4.19) \text{ sađlayan } x_{ij} \right\} \quad (3.20)$$

bu da,

$$L(\pi) = \min \left\{ \sum_{i < j} (c_{ij} + \pi_i + \pi_j) x_{ij} - 2 \sum_i \pi_i : (4.16) - (4.19) \text{ sađlayan } x_{ij} \right\} \quad (3.21)$$

Lagrange gevşetilmesi, (Eş. 3.22) denklemini sađlayan $\pi = \pi^*$ değeridir ve bu problem GSP için Lagrange duali olarak adlandırılmaktadır.

$$L(\pi^*) = \max_{\pi} \{L(\pi)\} \quad (3.22)$$

Lagrange gevşetilmesi problemi $L(\pi^*)$, GSP'nin eniyi tur uzunluđu için bir alt sınır sağlamaktadır. 20 noktadan 64 noktaya kadar çeşitli boyuttaki problemler bu metot ile çözülmüştür.

Held ve Karp'ın algoritması π 'nin hesaplanmasında deđişik metotlar kullanılarak geliştirilmiştir. Geliştirilen bu algoritma Held ve Karp'ın algoritmasından 25 kat daha hızlıdır. 10 noktadan 80 noktaya kadar problemler test edilmiştir [44].

Langrange çarpanlarını bulabilmek için ikili azalan bir algoritma kullanılarak, Held ve Karp'ın algoritmasından daha hızlı ve daha az bilgisayar süresi isteyen çözüm yöntemi geliştirilmiş ve 100 şehre kadar olan problemler çözülmüştür [45].

Dal-Kesme Yöntemi

Dal-Kesme yöntemi dal-sınır metodu ile çok yüzlü (polyhedral) metodun birleşimiyle oluşmaktadır. Bu metot büyük boyuttaki bazı zor problemleri başarı ile çözmüştür. Çok yüzlü teorisinin terminolojisi aşağıda verilmiştir:

p problemi, S olurlu sonuçların setinin olduğu $p : \min \{cx, x \in S\}$ olarak tanımlanabilir. p probleminin gevşetilmesi ise $p' : \min \{cx, \forall x \leq b, x \in \mathbb{R}^n\}$ şeklinde tanımlanır [46].

Dal-kesme yönteminin ilk fikri Dantzig ve arkadaşları (1954) tarafından ortaya konmuştur ve 49 şehirlik bir problem çözülmüştür. Gevşetilmiş problem (p') çözülmüş ve x çözümü elde edilmiştir. Eğer x S 'in dışında ise x S 'den, S üzerindeki tüm noktaları sağlayan ve x çözümünü içermeyen bir kesme düzlemi ile ayrılır. Bu eklemeye daha sıkı bir gevşetilebilir mümkün olmaktadır. Bu yöntem S içerisinde bir çözüm bulana kadar devam eder. Dal-kesme metodunun en önemli yanı, GSP için olurlu çözümlerin ortaya çıkmasına yarayan yüzlem (facet) eşitsizliklerinin bulunmasıdır. Yıllar geçtikçe literatürde GSP için bazı yüzlem eşitlikleri geliştirilmiştir. İlk yüzlem eşitsizliği Dantzig ve arkadaşları (1954) tarafından tamsayı doğrusal programlama formülasyonu için ortaya konan alt tur eleme kısıtıdır. Kesme düzlemlerinde kullanılmak için ikili-eşleme ve tarak adında iki tür eşitsizlik önerilmiştir [47]. Klik (clique) ağaç eşitsizlikleri adında farklı bir sınıf eşitsizlik ortaya konularak büyük boyuttaki GSP'nin çözümünde başarılı olunmuştur [46].

$G = (V, A)$, $(i, j) \in A$ kenarı ve (i, j) eşleşen noktaları V olan grafiği düşünelim. S , V 'deki noktaların bir alt kümesi ve $E(S) = \{(i, j) \in A \mid i \in S, j \in S\}$ şeklinde tanımlanmış olsun. Dört çeşit yüzlem eşitsizliği aşağıda kısaca açıklanmıştır.

a) *Alt tur eleme kısıtları:* Bu kısıtın amacı adından da anlaşılacağı gibi alt turların oluşmasını engellemektir.

$$\sum_{i,j \in S, i \neq j} x_{ij} \leq |S| - 1, \quad S \subset V, \quad 2 \leq |S| \leq n - 2 \quad (3.23)$$

(Eş. 3.23)'deki kısıtın başka bir formu aşağıdaki (Eş. 3.24) denklemdir.

$$\sum_{i \in S} \sum_{j \in \bar{S}} x_{ij} \geq 1, \quad S \subset V, \quad \bar{S} = V / S, \quad 2 \leq |S| \leq n - 2 \quad (3.24)$$

(3.20) kısıtı S'deki en az bir kenarın, S ile bağlantılı olduğunu göstermektedir [39].

b) *İkili eşleme kısıtları:* Bu kısıt her bir noktanın sadece iki kenar içermesini sağlamaktadır. Bu kısıtın formülasyonu aşağıdaki gibidir.

$$\sum_{(i,j) \in E(H)} x_{ij} + \sum_{(i,j) \in A'} x_{ij} \leq |H| + \frac{1}{2}(|A'| - 1) \quad (3.25)$$

Her $H \subset V$ ve her $A' \subset A$ aşağıdaki kısıtları sağlamaktadır:

- (i) $|\{i, j\} \cap H| = 1, \text{ her } (i, j) \in A'$
- (ii) $\{i_k, j_k\} \cap \{i_l, j_l\} = \emptyset, k \neq l \text{ ve } (i_k, j_k), (i_l, j_l) \in A'$
- (iii) $|A'| \geq 3 \text{ ve tek [47].}$

c) *Tarak eşitsizlikleri:* Bu kısıt aşağıdaki gibi tanımlanmıştır:

$$\sum_{k=1}^r \sum_{(i,j) \in E(H_k)} x_{ij} + \sum_{k=1}^s \sum_{(i,j) \in E(T_k)} x_{ij} \leq \sum_{k=1}^r |H_k| + \sum_{k=1}^s (|T_k| - t_k) - \frac{1}{2}(s+1) \quad (3.26)$$

Her $H, T_1, \dots, T_s \subseteq V$ aşağıdaki kısıtları sağlamaktadır:

- (i) $|T_k \cap H| \geq 1, k = 1, \dots, s$
- (ii) $|T_k / H| \geq 1, k = 1, \dots, s$
- (iii) $T_k \cap T_1 = \emptyset, 1 \leq k \leq s$
- (iv) $s \geq 3$ ve tek [47].

d) *Klik ağaç eşitsizlikleri*: Klik eşitsizliği aşağıdaki gibi tanımlanmıştır:

$$\sum_{k=1}^s \sum_{(i,j) \in E(H_k)} x_{ij} + \sum_{k=1}^s \sum_{(i,j) \in E(T_k)} x_{ij} \leq \sum_{k=1}^s |H_k| + \sum_{k=1}^s (|T_k| - t_k) - \frac{1}{2}(s+1) \quad (3.27)$$

Her $H_1, \dots, H_r \subseteq V$ ve $T_1, \dots, T_s \subseteq V$, tarak eşitsizliğinin sapları ve dişleridir. T_k dişi ile kesişen sapların sayısı, t_k olarak tanımlanmıştır. Klik ağacı aşağıdaki özellikleri sağlayan kliklere sahip K_n grafiği ile bağlıdır.

- (i) Klikler, saplar ve dişler adında iki farklı gruba ayrılır.
- (ii) Hiçbir iki dişi kesişemez.
- (iii) Hiçbir iki sap kesişemez.
- (iv) Her bir dişi, en az 2 ve en fazla $n-2$ nokta içerir.
- (v) Her bir sapı tek sayıda (≥ 3) dişler keser [46].

Güçlü bilgisayarların gelişmesiyle ve etkin Dal-kesme algoritmalarının ortaya çıkmasıyla büyük boyuttaki GSP problemlerinin eniyi sonuçları bulunmuştur. Crowder ve Padberg (1980) [48] tarafından, 48 şehir ile 318 şehir arasındaki problemler alt tur elemesi ve tarak kısıtları sayesinde çözülmüştür. Padberg ve Rinaldi (1987) [49] tarafından dal-kesme yöntemi kullanılarak, üst sınırı bulan sezgisel yöntem ve prosedür seti ile uymayan eşitsizliklerin saptanması yöntemiyle 532 şehirlik bir problem çözülmüştür. 1991 yılında büyük bir gelişme olmuş ve o zamana kadarki çözülebilen en büyük problem olan 2392

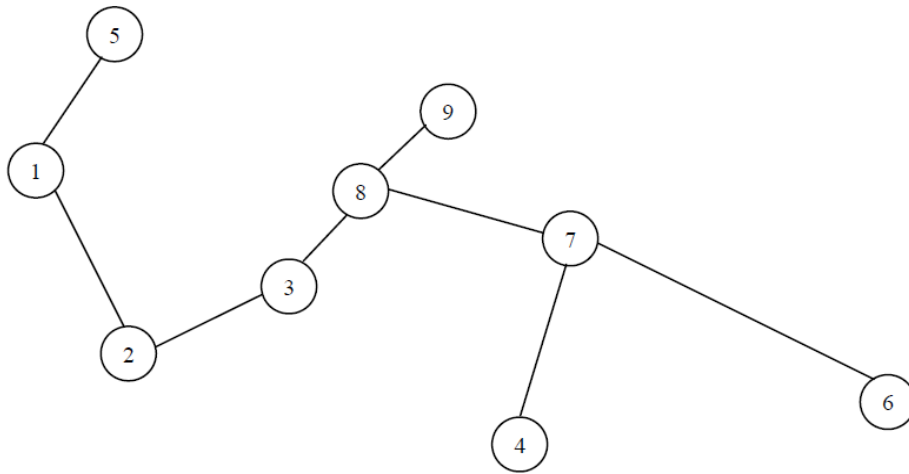
şehirlik problem, Padberg ve Rinaldi (1991) [50] tarafından çözülmüştür. Bu başarı pek çok faktöre bağlıdır [51]:

- Farklı yöntemlerin birleşimini, sezgisel ve kesin çözüm algoritmalarını ve alt tur eleme kısıtlarını, ikili eşleme kısıtlarını, tarak kısıtlarını ve klik-ağaç eşitsizliklerini kendi algoritmalarında kullanmışlardır.
- Hızlı bilgisayarlar, CYBER205 ve IBM3090/600, büyük boyuttaki problemleri çözmek ve çözüm zamanını düşürmek için kullanılmıştır. 2392 şehirlik problem, CYBER205 bilgisayarı ile 27 saat 20 dakikada, IBM3090/500 bilgisayarı ile 2 saat 40 dakikada çözülmüştür.
- XMP ve OSL adında iki farklı doğrusal programlama çözücüsü kullanılmıştır. Doğrusal programlama çözücüler, problemin çözüm süresine önemli derecede etki etmektedir.
- Lin ve Kernighan'ın (1973) [52] üst sınırı bulmak için geliştirdiği, etkin bir sezgisel yöntem kullanılmıştır.
- Alt tur elemesi ve ikili eşleme kısıtları için kesin çözüm yöntemleri kullanılmıştır. Tarak eşitsizlikleri ve klik-ağaç eşitsizliklerini çözmek için de sezgisel yöntemler kullanılmıştır. Applegate, Bixby, Chvatal ve Cook, GSP kütüphanesinden (TSPLIB) 1993 yılında 3.038 şehir, 4.461 şehir, 1995 yılında 7.397 şehir problemlerini çözmüşlerdir. 1998 yılında ise Applegate ve arkadaşları o ana kadarki en büyük problem olan 13.509 şehirlik problemi çözmüşlerdir. Geliştirdikleri algoritma 48 bilgisayardan oluşan bir ağda çalıştırılmıştır [51, 53].

3.6.2. Sezgisel çözüm yöntemleri

Literatürde birçok gezgin satıcı problemi için geliştirilmiş birçok sezgisel yer almaktadır. Bunun nedeni sezgisellerin, analitik yöntemlerin aksine polinomiyal olmayan bu tür problemler için istendiği gibi polinomiyal zamanda çözüm önermeleridir. GSP'ne benzeyen bir problem olan "Minimum Yayılan Ağaç (MYA) Problemi" sezgisel bir yöntem kullanılarak küresel en iyi çözüme

ulaştırılabilmektedir. Şekil 3.6' da görüldüğü gibi kısaca bir ağ sistemindeki bütün noktalara ulaşan en kısa yolun bulunması, olarak tanımlanan problem Kruskal ya da Prim tarafından önerilen basit sezgisellerden herhangi biri ile küresel en iyi çözüme ulaştırılmaktadır [54]. Bu problemin, çözüm uzayında gezgin satıcı probleminden çok daha fazla sayıda alternatif çözüm bulundurmasına karşın bu denli basit şekilde çözüme ulaştırılması GSP ve benzeri problemler için, polinomiyal zamanda çözüm üreten bir çözümün olabileceği umudunun oluşmasına neden olmaktadır. Bunun yanında sezgisel çözümler küresel en iyi çözümü garanti etmeseler de, polinomiyal zamanda kabul edilebilir derecede en iyi çözüm değerine yaklaşabildikleri durumda, zaman kısıtının bulunduğu uygulamalar için vazgeçilmez hale gelmektedirler.



Şekil 3.6. Minimum yayılan ağaç

En yakın komşu yöntemi

Bu yöntemde, var olan şehirler içerisinde rasgele bir şehir gezgin satıcı turu başlangıcı olarak seçilir ve bu şehre en yakın şehir turdaki sonraki adım olarak belirlenir. Ekleme işlemine bütün şehirler tura katılınca kadar devam edilir. Tüm şehirler tura eklenmeden önce turun kapanmasına izin verilmez. Oldukça basit ve akla yakın gelen yöntemde, hangi şehrin başlangıç olarak

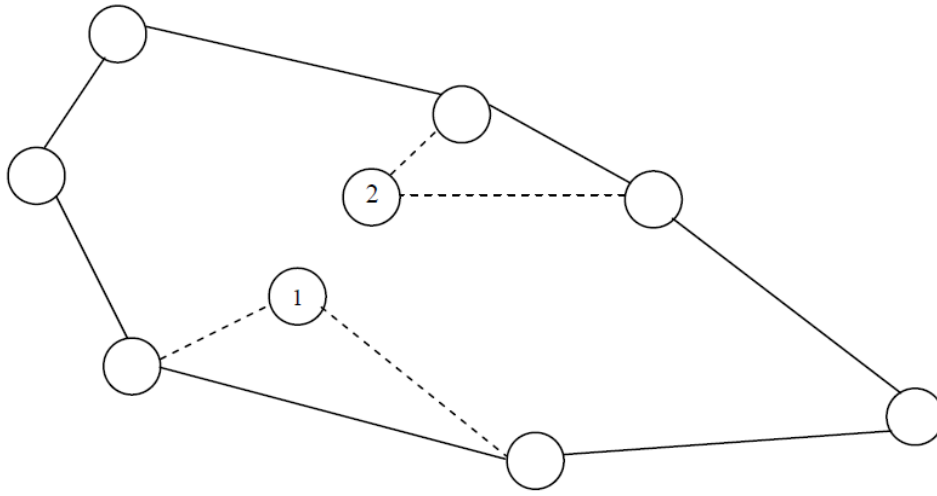
seçildiği ve mevcut yolun hangi ucundan devam edildiğini farklı basarımda sonuçlar elde edilebilmesine neden olabilmektedir. Ancak genellikle en yakındaki şehrin seçimi sırasında geride unutulmuş diğer şehirlerin tura katılması için uzun geri dönüşler gerektiğinden küresel en iyiden uzak çözümler elde edildiği görülmektedir [18].

En yakın ekleme yöntemi

İki şehir arasındaki bir tur ile çözüme başlanır. Daha sonra bu tura en yakındaki şehirlerin, sırayla mevcut turdaki şehirlerin arasına girerek tura eklenme durumları denenerek, turdaki mümkün olan en küçük büyüme bulunur ve bu şehir tura eklenir. Bu işlem bütün şehirler tura ekleninceye kadar devam ettirilir. Bu yöntemde de başlangıç şehrinin seçimi, ekleme sırası kuralı ve elde çözümü değiştirebilmektedir. En yakın ekleme ve en yakın komşu yöntemleri genellikle başlangıç çözüm elde edilmesi amacıyla kullanılır. Ancak DePuy ve arkadaşları [55] bu iki yöntemden yararlanarak GSP ve benzeri kombinasyonel problemlerin çözümü için kullanılabilecek bir meta sezgisel önermişlerdir. Meta sezgisel olarak nitelendirilmesinin nedeni diğer problemler için de kullanılabilmesi ve stokastik yapı içerdiğinden çözümün başarımının rasgelelik yardımıyla garanti edilmeye çalışılmış olmasıdır [18].

Geometrik yöntem

Şekil 3.7’ de görüldüğü gibi düzlemsel olarak yerleşmiş olan şehirlerin en dışında bulunanları dışbükey olarak kaplayacak biçimde mevcut bağlarla bağlanır. Daha sonra çizilen bölgenin içinde kalan şehirler içerisinde dış bölgedeki ardışık şehirlere yapılan bağlardan en geniş açığa sahip olanı tura eklenir. Bu ekleme bütün şehirler tura dahil olana kadar devam edilir [54].



Şekil 3.7. Geometrik yöntem

Greedy yöntemi

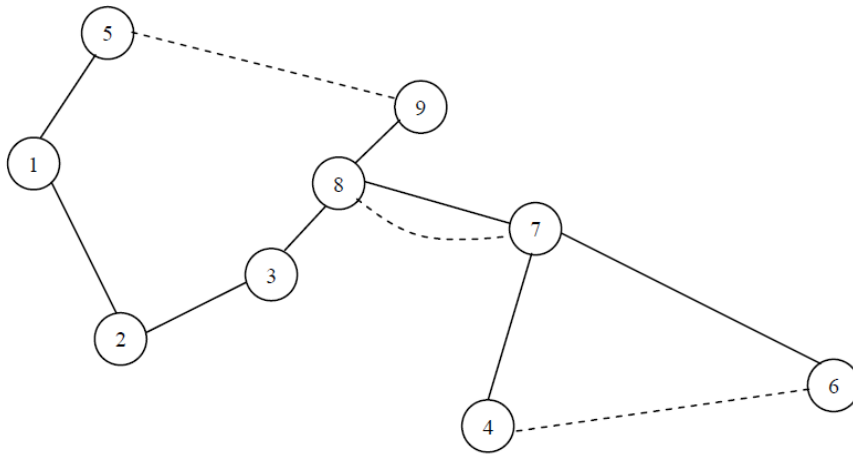
Olası bütün bağların küçükten büyüğe doğru sıralanması ile yöntemle başlanır. Daha sonra derecesi zaten iki olan noktalara tekrar bağlanmaksızın ve bağ sayısının N 'e eşit olmasından önce turun kapatılmasını önleyerek, en küçükten başlayarak, sırayla bağların eklenmesi yoluyla bir tur oluşturulur [34].

Bovurka yöntemi

Greedy yöntemine benzeyen bu yöntemde her şehre en yakın komşu uzunluğu belirlenir. Daha sonra belirlenen bağlar artan şekilde sıralanır. Bağlar sırayla alt tur oluşturmayacak ve bir şehre ikiden fazla bağ yapmayacak şekilde tura eklenir. Tura ekleme yapılırken bir şehre iki bağ yapılmış ise bu şehirle kurulabilecek olası diğer bağlar listeden çıkartılır. Yapılan ilk sıralamadaki bütün bağlar bittiğinde büyük olasılıkla bir tur tamamlanmamış olacaktır. Bu durumda bağ yapması halen mümkün olan şehirler için en yakın olası bağlar belirlenip sıralanarak önceki bölümde olduğu gibi tura ekleme yapılmaya devam edilir. Tam bir tur oluşturuluncaya kadar işleme devam edilir [34].

Christofides algoritması

Algoritma üçgen eşitsizliğini sağlayan GSP için Minimum Yayılan Ağaç (MYA) çözümü temel almaktadır. MYA, çözümünde derecesi tek olan noktalar arasında, her noktaya sadece tek bağın yapıldığı G (mükemmel eşleşme-perfect matching) ağı oluşturulur. G ağı ile başlangıçtaki MYA çözümü birleştirilerek M ağı (Şekil 3.8) oluşturulur. M ağı içerisinde Euler turu oluşturulur ve tur ziyaret edilen noktanın tekrar ziyaretini engelleyecek biçimde Şekil 3.2’de görüldüğü gibi Hamilton turuna dönüştürülerek çözüm sonlandırılır. Algoritmanın en kötü performans durumunda bile en kısa turdan 1,5 kat uzaklaşabileceği ispatlanmıştır[18,56].



Şekil 3.8. Christofides algoritmasında M ağı

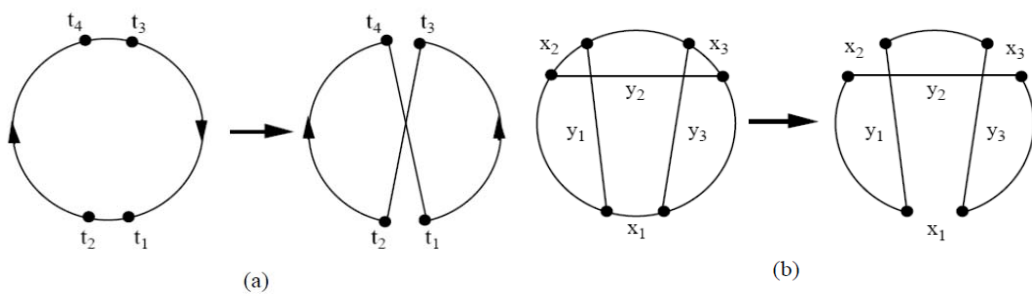
Grasp yöntemi

Tur oluşturmacı bir yaklaşım olan Grasp yöntemi, en yakın komşu yöntemine benzer şekilde çalışmaktadır. Ancak bu yöntemden farklı olarak olasılıklı bir yapı sergilemektedir. Tur oluştururken mevcut şehre en yakın ‘n’ adet şehir belirlenmektedir. Daha sonra bu şehirlerarasından rasgele seçim yapılarak sonraki şehir belirlenir ve bu işleme alt tur oluşturmada tur tamamlanıncaya kadar devam edilir. En yakın n adet komşu arasında seçim yapılırken tamamen rasgele davranılabileceği gibi, mevcut şehre en yakın olan bağa en

yüksek, en uzak olana ise an düşük olmak üzere yapılacak bağın uzunluğuyla orantılı olasılık atanarak seçim yapılabilir [18].

R-opt denemeleri

R-opt denemeleri mevcut bir turun geliştirilmesi için kullanılabilecek bir yöntemdir. Yöntemde “r” ile belirtilen sayıda bağ çıkarılıp, yerine tur yapısını bozmayacak ve daha kısa tur oluşturacak şekilde yeni “r” adet bağ eklenir. Denemelere gelişme oluşturacak hareket bulunamadığı noktaya kadar devam edilir. Kısaltma yapan değişimi elde etmek için birçok deneme yapılması gerekebilmektedir. R sayısı büyüdükçe yapılması olası deneme sayısı da üssel olarak artmaktadır, ancak en iyiyi bulma olasılığı da artmaktadır. En iyinin bulunması ile süre arasında bir ödünleşim söz konusu olduğundan genellikle 2-opt ve 3-opt kullanılmaktadır. Şekil 3.9 (a)’da görüldüğü gibi ikişer adet bağ, t_1-t_2 , t_4-t_3 çıkarılıp t_4-t_1 ve t_3-t_2 eklenerek 2-opt hareketi gerçekleştirilmektedir. Benzer yol ile Şekil 3.9 (b)’de görüldüğü gibi üçer adet bağ, x_1 , x_2 , x_3 çıkarılıp y_1 , y_2 , y_3 eklenerek 3-opt hareketi gerçekleştirilmektedir. Bu değişimlerde amaç daha kısa tur elde edilmesidir [18].



Şekil 3.9. (a) 2-opt değişimi (b) 3-opt değişimi

Lin-kerninghan yöntemi

Lin-Kerninghan yöntemi [57] simetrik gezgin satıcı problemi için küresel en iyi ya da ona yakın çözümler üretmek için kullanılan en iyi yöntemlerden bir

olarak bilinir [58]. Algoritma, r-opt yönteminin kullanımına dayanmaktadır. Yöntem, çalışması sırasında gereksinime göre “r” değerini değiştirmektedir. Her iterasyonda “r” değerinin artırılmasının çözümün daha da ksalmasına neden olup olmayacağı test edilir. Bu işlemlere bir durdurma koşulu sağlanıncaya kadar devam edilir.

Algoritma temel olarak bazı değişimlerle bir turdan daha kısa yeni bir tur elde edilmesini sağlamak üzere, geliştirme yapamadığı noktaya kadar devam eder. Bu denemeler rasgele oluşturulmuş birçok tur ile tekrarlanabilir. Yöntem başlangıç turuna bakmaksızın kendi basına oldukça başarılıdır. Yöntem aşağıda daha ayrıntılı olarak açıklanmaktadır.

T’nin başlangıç turu olduğunu varsayalım. Her iterasyonda $X=\{x_1, \dots, x_r\}$ ve $Y=\{y_1, \dots, y_r\}$ olmak üzere bağlardan oluşan iki küme oluşturulmaya çalışılır, öyle ki; X kümesindeki bağların mevcut turdan silinmesi ve yerine Y kümesindeki bağların eklenmesi durumunda daha kısa bir tur oluşacaktır. Bu bağ değişimi r-opt hareketi olarak adlandırılır. Etkin bir uygulamanın gerçekleştirilebilmesi için X ve Y kümesindeki elemanların aşağıdaki özellikleri yerine getirmeleri gerekmektedir.

(1) Sıralı değişim kriteri : Bağlar (x_i ve y_i) Sekil 3.9 (b)’de görüldüğü gibi $\{x_1, y_1, x_2, y_2, \dots, x_r, y_r\}$ şeklinde sıralı olmalıdır. Olumlu etki yaratan ancak zorunlu olmayan bir nokta ise zincirin sonunun, bası ile birleştirilmesidir. (2) Oluşturulacak yeni bağ yapısı olurlu bir tur ifade etmelidir. (3) İlk bağdan sonraki bağın seçimindeki kazanç ($g_i = c(x_i) - c(y_i)$) pozitif olmalıdır. Bu durumda toplam kazanç da $G_i = g_1 + g_2 + \dots + g_i$ pozitif olacaktır. Bu kural çok sınırlayıcı görünmekle birlikte yöntemin etkinliğini ve hızını arttıran bir kuraldır. (4) X ve Y kümeleri bağımsız kümeler olmalıdırlar.

Algoritma karışık görünen bir yapıya sahiptir ve bunların dışında birçok kural ile algoritma yalınlaştırılmıştır. Örneğin, çıkartılan bir bağlantı eklenmemeli ve

daha önce eklenmiş bir bağlantı çıkartılmamalıdır; $y_i = (t_{2i}, t_{2i+1})$ bağının eklenmesinde t_{2i} noktasının en yakın 5 noktası ile sınırlandırılmıştır. Bu sınırlamalar sonrada algoritma aşağıdaki gibi gösterilebilir [18].

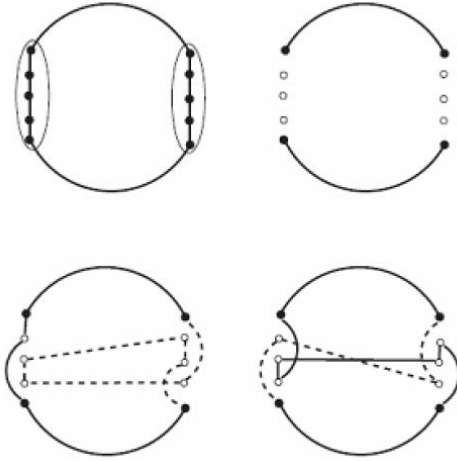
1. Rasgele bir T turu üret.
 2. $i = 1$ olsun, t_1 'i seç.
 3. $x_1 = (t_1, t_2) \in T$
 4. $y_1 = (t_2, t_3) \notin T$ yi $G_1 > 0$ olmak üzere seç.
- Eğer bu mümkün değilse Adım 12'ye git.
5. $i = i + 1$ olsun.
 6. $x_i = (t_{2i-1}, t_{2i}) \in T$ seç öyle ki
 - (a) Eğer t_{2i}, t_1 'e bağlanırsa değişim sonucu bir turdur ve
 - (b) $x_i \neq y_s$ her $s < i$ için.
 7. $y_i = (t_{2i}, t_{2i+1}) \notin T$ seç öyle ki
 - (a) $G_i > 0$
 - (b) $y_i \neq x_s$ her $s < i$ için ve
 - (c) $x_i = 1$ vardır.
- Eğer böyle bir y_i mevcut ise, adım 5'e git.
8. Eğer denenmemiş bir y_2 alternatifi varsa, $i=2$ yap ve adım 7'ye git
 9. Eğer denenmemiş bir x_2 alternatifi varsa, $i=2$ yap ve adım 6'ya git
 10. Eğer denenmemiş bir y_1 alternatifi varsa, $i=1$ yap ve adım 4'e git
 11. Eğer denenmemiş bir x_1 alternatifi varsa, $i=1$ yap ve adım 3'e git
 12. Eğer denenmemiş bir t_1 alternatifi varsa, adım 2'ye git
 13. Dur (ya da adım 1'e git)

Şekil 3.10. Lin-Kerninghan yöntemi algoritma adımları

Hyperopt denemeleri

Tur içerisinde ayırık, aynı sayıda şehrin bulunduğu h uzunluklu iki bölge tespit edilir. Başlangıç ve bitiş şehirleri sabit kalmak şartıyla bölgeler içi ve bölgeler

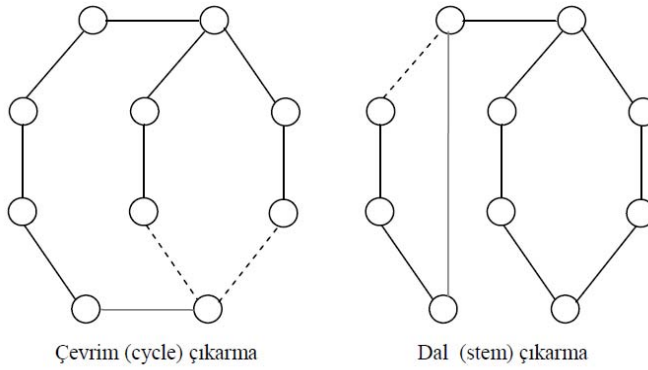
arası olası bağ değişiklikleri denenerek daha kısa turlar elde edilmeye çalışılır (Şekil 3.11) [59].



Şekil 3.11. Örnek hyperopt denemeleri [59]

S&C (stem and cycle) yöntemi

Gamboa ve arkadaşları [60] tarafından önerilen S&C (Stem and Cycle) yöntemi, Lin-Kerningham yöntemine alternatif olarak öne çıkmaktadır. Bir başlangıç turu, S&C yöntemi koordinasyonu altında Alt Tur Çıkarma Zinciri (Subpath Ejection Chain (SEC)) kullanılarak iterasyonel olarak geliştirilmeye çalışılmaktadır. Yöntem, Lin Kerningham yöntemindeki gibi statik alternatif yollar yerine, dinamik bir yol yapısı takip etmektedir. Yöntemde grafiğin alt bölümündeki bir şehir dal ve çevrimler ile ilişkilendirilerek tur geliştirilmeye çalışılmaktadır.



Şekil 3.12. Dal ve çevrim çıkarma [60]

3.6.3. Meta-sezgisel yöntemler

Literatürde ilk defa 1986 senesinde Glover tarafından kullanılan “metasezgisel” terimi Yunanca’da “daha ilerisi, üst derecede olanı, daha modern” anlamına gelen “meta” öneki ile İngilizce’de “sezgisel, buluşsal” anlamına gelen “heuristic” kelimesinin birleşiminden oluşmuştur. “Heuristic” kelimesi ise Arşimet’in ünlü “Eureka!” (Buldum!) sözünden türetilmiştir. Literatürde “Metasezgisel” kavramı, “Modern Sezgisel” veya “Yapay Zeka Yaklaşımı” olarak da geçmektedir [61].

Metasezgisel yöntemler çözüm uzayında etkili bir şekilde arama yapmak için, farklı yapılarıdaki alt kademe sezgisel algoritmaların zekice birleştirilmesi ile oluşturulmuş iteratif problem çözme prosedürleridir. Bu yöntemler her iterasyonda bir çözümden veya çözüm koleksiyonundan yola çıkarak yeni çözümler üretirler. Çoğu metasezgisel yaklaşım çözüm uzayında stokastik fakat bilinçli bir şekilde arama yapar [62].

Metasezgisel algoritmalarda amaç, arama uzayının en umut verici noktalarında arama yapıp yerel optimum noktalardan kurtularak, eniyi çözüme yakın sonuçlar elde etmektir. Bu yöntemler karmaşık komşuluk arama kuralları, bellek yapıları ve çözümlerin kombinasyonundan yeni çözümler elde ederek sonuç üretmeye çalışırlar. Bu metotlarla üretilen sonuçlar, klasik sezgisel algoritmalarla üretilen sonuçlardan daha iyidir fakat

hesaplama işlemi klasik algoritmalara kıyasla daha uzun sürmektedir. Metasezgisel yöntemler problemin özelliklerine bağımlıdır, diğer bir deyişle problem tipine göre farklı parametreler gerekebilmektedir. Bu da diğer problemlere uygulanabilirliğini zor kılar. Metasezgisel algoritmalar, klasik sezgisel algoritmaların doğadan esinlenerek geliştirilmiş hali olarak görülebilir [96].

Çoğu sezgisel algoritmalar probleme bağımlı algoritmalarlardır. Diğer bir deyişle, bir problem için en iyi performans gösterirken diğer bir problem için aynı şekilde başarılı çözüm üretmeyebilir. Bundan dolayı çok daha genel olarak uygulanabilen algoritmaların geliştirilmesine yönelik çalışmaların sayısı her geçen gün hızlı bir şekilde artmaktadır. Geçen 30-40 yılda bu özelliğe sahip çeşitli yaklaşımlar geliştirilmiştir. Bu yaklaşımlar sosyal, biyoloji, zooloji, fizik, bilgisayar ve karar verme gibi bilimler temel alınarak türetilmiştir. Bundan dolayı bu tür yaklaşımlara modern sezgisel yaklaşımlar ya da yapay zekâ yaklaşımları adı verilmektedir [97].

Blum ve Roli, metasezgisel yöntemler hakkında aşağıdaki genellemeleri yapmışlardır [62]:

- Metasezgisel yöntemler arama prosesi sırasında kılavuzluk yapan stratejilerdir.
- Genel amaç, arama uzayını etkili bir şekilde araştırıp eniyi veya eniyiye yakın sonuçlar elde etmektir.
- Metasezgisel yöntemler yerel arama tekniklerinden, karmaşık öğrenme prosedürlerine kadar yaygınlık gösterir.
- Metasezgisel algoritmalar yaklaşık yöntemlerdir ve genellikle deterministik değildir.
- Arama uzayında yerel eniyi noktalardan takılıp kalmayı engelleyecek mekanizmaları içinde barındırırlar.

- Metasezgisel teknikler probleme özel yöntemler değildirler. Genellikle tüm kombinatoriyal problemlere uygulanabilirler.
- Günümüzde gelişmiş metasezgisel algoritmalar arama sırasında kılavuzluk etmesi için hafıza tabanlı süreçler bulunmaktadır.

Metasezgisel algoritmalar zor kombinatoriyal optimizasyon problemlerinde iyi sonuçlar üretmek için ileri seviye sezgisel prosedürlerdir. Her metasezgisel yöntemin birden fazla parametreye sahip olması esneklik sağlar fakat bu durum problem tiplerine göre de değişiklik göstermektedir. Çoğu metasezgisel algoritma paralel işlem için uyumludur. Metasezgisel algoritmaların sınıflandırılması aşağıdaki şekilde yapılabilir [63]:

- Doğadan esinlenerek geliştirilenler – Doğadan esinlenmeden geliştirilenler
- Populasyon tabanlı algoritmalar – Tek nokta (yerel arama) algoritmalar
- Dinamik amaç fonksiyonlu – Statik amaç fonksiyonlu algoritmalar
- Tek komşu yapılı – Çok komşu yapılı algoritmalar
- Hafıza kullanan – Hafıza kullanmayan algoritmalar

Metasezgisel algoritmaların yerel komşuluk ve populasyon arama tabanlı iki türü vardır. Yerel komşuluk arama yöntemlerinde her iterasyonda bir çözüm ele alınıp uygun komşu çözümleri incelenerek arama uzayında belirli bir noktada yoğun bir tarama yapılır. Yerel arama yöntemine örnek olarak Tavlama Benzetimi ve Tabu Arama algoritması verilebilir. Populasyon tabanlı arama yöntemlerinde ise her iterasyonda bir çözüm havuzu içerisinde bazı çözümler seçilip uygun bir şekilde birleştirilerek yeni çözümler elde edilir. Genetik Algoritma, iki çözümün rasgele seçilen parçaları birleştirilerek yeni çözüm elde edilen bu tip yöntemlere bir örnektir. Adaptif Hafıza prosedürlerinde ise birçok çözümden birden fazla yeni çözüm elde edilir [64].

Tavlama benzetimi

Tavlama benzetimi ismi demirin tavlansından alınmıřtır. Demir ok yksek sıcaklıklarda ok yksek enerjilere sahip olabilen bir metaldir. Bununla birlikte iřlenen demirin sıcaklıęı nispeten dřk olmasına raęmen enerjisi yksek durumlarla karřılařılabilir [65]. Temel olarak, TB demirin bu zellięinden faydalanarak en iyiye gitmeyi hedefler. TB'de TA(Tabu Arama)'daki gibi, yeni zmler, komřuluk oluřturma stratejisi kullanılarak elde edilir. Fakat bu algoritmada yasak listesi yerine, stokastik bir yapı kullanılır.

Tavlama terimi fiziksel olarak, ısı banyosundaki bir katının yksek enerji durumlarından bařlayarak daha dřk enerji durumlarının elde edilme srecini temsil etmektedir. Bu sre ok genel olarak iki iřlemden oluřmaktadır [66]:

- Isı banyosunun bařlangı sıcaklıęının katının eriyebileceęi bir deęere kadar ykseltilmesi.
- Katılar dřk enerjili durumda, yani dřk sıcaklıkta daha kararlıdır. Yani katıların paracıkları dřk sıcaklıklarda daha dzenlidir. Bundan dolayı katının paracıkları kendini dzenleyene kadar ısı banyosunun sıcaklıęının giderek azaltılması.

Katıların tavlama srecinde geirdięi durumları benzetebilmek iin Metropolis ve ark. (1953) tarafından "Metropolis Algoritması" geliřtirilmiřtir. Algoritmada i durumunda bulunan katının enerjisi E_i iken, bir sonraki j durumuna geen katının enerjisi E_j 'dir. Eger j durumundaki enerji i durumundaki enerjiden kk veya eřitse, j durumu yeni mevcut zm olarak kabul edilir. Aksi takdirde j durumu Eř. 3.28 nolu formle gre elde edilen olasılık deęeri ile kabul edilir.

Bu olasılık değeri "Metropolis Kriteri" olarak anılır. Burada k_B fiziksel tavlama sürecinde "Boltzman Sabiti" olarak bilinen fiziksel bir sabiti ifade eder.

$$\exp\left[\frac{E_i - E_j}{k_B * T}\right] \quad (3.28)$$

TB algoritmasının amacı, tüm mümkün çözüm noktalarının bir alt kümesinde (S) tanımlanmış bir $f(x)$ fonksiyonunu eniyileyecek bir x çözümü bulmaktır. TB algoritması rassal olarak seçilen bir başlangıç çözümüyle aramaya başlar. Bundan sonra uygun bir mekanizma ile bu çözüme komşu bir çözüm seçilir ve $f(x)$ 'de meydana gelen değişim hesaplanır. Eğer değişim istenilen yönde ise komşu çözüm mevcut çözüm olarak alınır. Eğer istenen yönde bir değişim sağlanmamışsa, TB algoritması bu çözümü "Metropolis Kriteri" ile elde edilen olasılık değeri ile kabul eder. Amaç fonksiyonunda ters yönde bir değişim yaratan bir çözümün belli olasılık değeri ile kabulü, TB algoritmasının yerel en iyi noktalardan kurtulmasını sağlamaktadır. Yukarıdaki olasılık değerine göre T değeri yüksek olduğunda amaç fonksiyonunda meydana gelen artışların çoğu kabul edilecektir. T değeri azaldıkça kabul edilme oranı da azalacaktır. Bu nedenle TB algoritmasında yerel noktalara takılmamak için başlangıç sıcaklık değeri yüksek seçilerek yavaş yavaş azaltma yoluna gidilmektedir [66].

Osman'ın 1993 yılında yaptığı TB uygulaması hem daha anlaşılması zordur hem de daha başarılıdır. Bu algoritma daha iyi bir çözümden başlar, deneme fazında algoritmanın bazı parametreleri ayarlanmıştır. Zengin çözüm komşulukları keşfedilir ve soğutma çizelgesi daha karmaşıktır. Algoritmanın komşuluk yapısı seçilen iki rota p ve q arasında değişim üretim mekanizması kullanır [67].

Tabu (yasak) arama (TA)

Glover tarafından 1989 yılında önerilmiştir. Hesaplamalar tabu aramanın; esnekliği ile birçok kesin prosedürü yenen ve bilinen hemen hemen tüm tekniklerle rekabet edebilen kurulmuş iyi bir yaklaşık tekniği olduğunu göstermektedir. TA deterministik iteratif iyileştirme algoritmasını bir olasılıkla maliyet artış çözümleri de onaylamak için birleştirir. Bu; araştırmanın yerel minimum değerden uzağa yönlendirilerek araştırma uzayının diğer kısımlarının da keşfedilmesidir. Bir sonraki ziyaret edilen çözüm en iyi maliyetli mevcut çözümün geçerli komşusu seçilir, hatta eğer mevcut çözümün maliyetinden daha kötü olsa bile. Geçerli komşuların kümesi araştırmayı bir döngüden korumak için tasarlanan tabu listesi ile kısıtlanır. Algoritmanın yürütülmesi esnasında tabu listesi dinamik olarak güncellenir. Tabu; ileriki bir kaç iterasyonda kabul edilemeyecek çözümleri tanımlamaktadır. Ama tabu listesindeki bir çözüm eğer kalitesi bazı durumlarda yeterince iyiye kabul edilebilir [98].

TA akıllı problem çözme prensipleri üzerine kuruludur. Temel elementlerden biri esnek ve dinamik bellek yapısı yani bir tabu listesi kullanmasıdır. TA prosedürü basittir. Her iterasyonda en iyi hareket seçilir. Eğer bu hareket amaç fonksiyonunu kötüleştirirse sadece eğer ters hareketi yasak durumu yoksa yani tabu listesinde değilse gerçekleştirilir. Eğer tabu listesinde ise, yasak listesinde olmayan bir sonraki en iyi hareket seçilir ve uygulanır. Bu proses bir durma kriterine ulaşıncaya kadar tekrar eder [99]. Tabu Arama (TA) algoritmasının en temel elemanları ve bunların işleyiş şekli aşağıda sırasıyla verilmektedir [66].

Başlangıç Çözümünün Oluşturulması: En genel şekilde başlangıç çözümü rassal olarak elde edilir. Ancak, farklı problemler için geliştirilmiş olan bir sezgisel algoritmadan yararlanarak da başlangıç çözümünün elde edilmesi mümkündür.

Hareket Mekanizması: Mevcut bir çözümde yapılan bir değişiklikle yeni çözümün elde edilmesi hareket mekanizmasıyla gerçekleştirilir. Hareket mekanizmasındaki olası hareketler, mevcut çözümün komşularını oluşturur. Hareket mekanizması algoritmanın etkinliği açısından önemli olduğu için problemin yapısına bağlı olarak uygun bir şekilde seçilmelidir.

Aday Liste Stratejileri: TA algoritması yapılması mümkün olan, tabu olmayan ve amaç fonksiyonunun değeri açısından en iyi sonucu veren hareketlerin seçilmesi kuralına dayalı olarak çalışır. Aday liste stratejileri mümkün hareket listeleridir. Bu listelerden hareketler belirli stratejilere göre seçilir.

Hafıza: TA algoritmasının temel elemanlarından birisi de hafızadır. Arama boyunca ortaya çıkan durumlar, H hafızasına kaydedilir. Bu hafıza kısa dönemli hafıza olarak adlandırılır. Yapılmasına izin verilmeyen hareketler "tabu" olarak adlandırılır ve esnek hafıza içinde "tabu listesi" adı altında kaydedilirler. Bu hareketler belli bir süre sonra tabu listesinden çıkarılır ve yapılmasına izin verilir.

Tabu Yıkma Kriterleri: Tabu yıkma kriterleri, tabunun ortadan kalkabileceği durumları ifade etmektedir. En genel tabu yıkma kriteri, mevcut durumdan daha iyi bir sonuç verecek tabu hareketinin yapılmasına izin verilmesidir. Bu kriterin kullanılması TA algoritmasının etkinliğini arttırmaktadır. Ayrıca, eğer tüm mümkün hareketler tabu ise bu hareketlerden tabu süresinin bitmesine en yakın olan bir tabu hareketine izin verilir.

Durdurma Koşulu: TA algoritması, bir veya birden fazla durdurma koşulunu sağlayıncaya kadar aramasını sürdürmektedir. Bu koşullardan bazıları aşağıda verilmiştir.

- Seçilen bir komşu çözümün komşusunun olmaması,
- Belirli bir iterasyon sayısına ulaşılması,

- Belirli bir çözüm değerine ulaşılması,
- Algoritmanın bir yerde tıkanması ve daha iyi sonuç üretememesi

ARP ye ilk TA uygulamalarından biri Willard'a atfedilir (1989). Pureza ve Franca (1991) farklı bir rotaya bir nokta hareket ettirerek veya fizibilliği koruyarak iki rota arasındaki köşeleri değış tokuş ederek bir çözümün komşularını tanımlar. Willard'ın yaptığı gibi, her iterasyonda tabu olamayan en iyi hareket seçilir. Willard algoritmasından iyi olmakla birlikte iyi sonuçlar üretememiştir. İleri araştırmalar TA çalışması için daha teferruatlı araştırma mekanizmalarının gerekli olduğunu göstermiştir [67].

Genetik algoritma

Evrimsel Programlamanın bir parçası olan Genetik Algoritmalar (GA) Darwin'in evrim teorisinden esinlenerek ortaya çıkmıştır. Evrimsel programlama, 1960'lı yıllarda Rechenberg'in "Evrimsel Stratejileri" adlı çalışmalarıyla gündeme gelmiştir. Evrimsel programlama genellikle genetik algoritmaları, evrimleşme stratejisini ve genetik programlamayı içermektedir. Genetik algoritmalarla evrimleşme stratejisi arasında önemli fark genetik algoritmaların çaprazlama ve mutasyon operatörlerini, evrimleşme stratejisinin ise yalnız mutasyon işlemini kullanmasıdır [61].

GA genetik popülasyonu üzerine dayalı uyarlamalı bir sezgisel araştırma metodudur. Temel kavramları 1975 yılında Holland tarafından geliştirilmiştir, karmaşık problemlerin çözümünde GA kullanımının pratikliği De Jong (1975) ve Goldberg (1989) tarafından yapılan çalışmalarda ispat edilmiştir. GA, bazı yakınsama kriterleri ile karşılaşılıncaya kadar her iterative proses boyunca yavruların yeni jenerasyonunu oluşturarak kromozomlarla kodlanmış bireylerin popülasyonunu geliştirir. Bu kriterler mesela, jenerasyon sayısının maksimumluğuna veya benzer bireylerden oluşmuş homojen popülasyon yakınsamasına karşılık gelebilir. En iyi kromozom üretildiğinde kodu çözülür

ve uygun çözümü sağlar [38]. Bireylerin yeni jenerasyonunun oluşturulması aslında dört adım ya da evreyi kapsar: Gösterim, seçim, rekombinasyon ve mutasyondur.

Gösterim: Çözüm uzayının gösterimi çözümün önemli özelliklerinin şifre çözümünden oluşur. Bu populasyonun bir bireyini tanımlayan kromozomdur. Genellikle bir çözümün temel karakteristiklerini yakalayan ardışık genlerden yapılmıştır [68].

Seçim: Bu faz birleştirme amaçları için populasyondan iki çift bireyi rast gele gelmekten oluşur. Bir populasyon üyesinin seçim olasılığı genetik çeşitliliği koruyarak genetik kaliteyi vurgulamak için genellikle uygunluğu ile orantılıdır. Burda uygunluk çözüm uzayı keşfedilirken maksimum yapılacak kazanç, fayda ve iyiliğin ölçümüne işaret eder.

Rekombinasyon: Rekombinasyon yeni nesili şekillendiren yavruları üretmek için seçilen çiftlerin genlerini kullanır.

Mutasyon: İlave çözüm uzayı keşfi ve genetik çeşitliliği sağlamak ve sunmak için eş zamanlı olarak tek bireyin bazı genlerini rastgele olarak değiştirmeyi içerir [100]. Genetik algoritmanın çalışmasını aşağıdaki gibi özetlenebilir [101];

1. Adım: Olası çözümlerin kodlandığı bir çözüm grubu oluşturulur (çözüm grubu, biyolojideki benzerliği nedeniyle, toplum (population), çözümlerin kodları (string) da kromozom olarak adlandırılır).

2. Adım: Her kromozomun ne kadar iyi olduğu bulunur (uygunluk fonksiyonu).

3. Adım: Bu kromozomlar eşlenerek (mating), yeniden kopyalama (recombination) ve değiştirme (crossover) operatörleri uygulanır. Bu sayede yeni bir toplum oluşturulur.

4. Adım: Yeni kromozomlara yer açmak için eski kromozomlar ortadan kaldırılır.

5. Adım: Tüm kromozomların uygunlukları tekrar hesaplanır.

6. Adım: Eğer jenerasyon süresi dolmamışsa 3. adıma gidilir.

7. Adım: O ana kadar bulunmuş en iyi kromozom sonuçtur.

Kromozom sayısı, mutasyon oranı, çaprazlama özelliği, nesillerin değerlendirilmesi GA performansını etkileyen durumlardır [101].

Kromozom sayısı: Kromozom sayısını arttırmak çalışma zamanını arttırırken azaltmak da kromozom çeşitliliğini yok eder.

Mutasyon Oranı: Kromozomlar birbirine benzemeye başladığında hala çözüm noktalarının uzağında bulunuyorsa mutasyon işlemi GA'nın sıkıştığı yerden kurtulmak için tek yoludur. Ancak yüksek bir değer vermek GA'yı kararlı bir noktaya ulaşmaktan alıkoyacaktır.

Kaç Noktalı Çaprazlama Yapılacağı: Normal olarak çaprazlama tek noktada gerçekleştirilmekle beraber yapılan araştırmalar bazı problemlerde çok noktalı çaprazlamanın çok yararlı olduğunu göstermiştir.

Çaprazlamanın sonucu elde edilen bireylerin nasıl değerlendirileceği: Elde edilen iki bireyin birden kullanılıp kullanılamayacağı bazen önemli olmaktadır.

Nesillerin birbirinden ayrık olup olmadığı: Normal olarak her nesil tümüyle bir önceki nesle bağlı olarak yaratılır. Bazı durumlarda yeni nesli eski nesille birlikte yeni neslin o ana kadar elde edilen bireyleri ile yaratmak yararlı olabilir.

Parametre kodlanmasının nasıl yapıldığı: Kodlananın nasıl yapıldığı en önemli noktalardan biridir. Örnek vermek gerekirse kimi zaman bir parametrenin doğrusal ya da logaritmik kodlanması GA'nın performansında önemli bir farka yol açabilir.

Kodlama gösteriminin nasıl yapıldığı: Bu da GA'nın performansını etkileyen bir noktadır. İkilik düzen, kayan nokta aritmetiği ya da gray kodu ile gösterim en yaygın yöntemlerdir.

Başarı değerlendirmesinin nasıl yapıldığı: Akıllıca yazılmamış bir değerlendirme işlevi çalışma zamanını uzatabileceği gibi çözüme hiçbir zaman ulaşmamasına neden olabilir.

Yapay sinir ağları

İlk olarak McCulloch ve Pitts (1943) tarafından geliştirilen yapay sinir ağlarının tanımı Kohonen (1987) tarafından şu şekilde yapılmıştır: “*Yapay sinir ağları, paralel bağlı çok sayıdaki basit elemanın, gerçek dünyanın nesneleriyle biyolojik sinir sisteminin benzeri yolla etkileşim kuran hiyerarşik bir organizasyondur*”. Özellikle doksanlı yıllardan sonra popüleritesi artan YSA' lar günümüzde halen birçok alanda başarıyla kullanılmaktadır.

Bir yapay sinir ağı; ağırlıklandırılmış bağlantılarla zengin bir şekilde bağlanmış birimlerden oluşan bir sayısal modeldir, tıpkı insan beynindeki nöronlar gibi; bir birimden başka bir tanesine bir bağlantı boyunca sinyal gönderilir ve ilgili ağırlıkla değiştirilir [67]. Bir yapay sinir ağı ağırlıklandırılmış

bağlarla bağlanan temel noktalardan oluşan bir ağı içerir. Noktalar, basit bir hesaplamayı gerçekleştirmeye yetenekli, ağırlıklandırılmış verilerin toplamından oluşan, eşik veya yanlılık olarak nitelendirilen bir kısıtın eklenmesiyle takip edilen ve lineer olmayan cevap fonksiyonunun uygulaması olan hesaplama birimlerini gösterir. Bir birimin hesaplama sonucu çıktısını atar. Bu çıktı girgin bağlantılarla bağlanmış noktalar için girdidir. Ağın genel görevi noktaların ortaklaşa hesabının ortalaması ile gerekli girdi-çıkı ilişkisi gibi uygun ağ yapılandırmayı başarmaktır [98].

Görünüşte biyolojik karşılıkları ile ilgili olmalarına rağmen yapay sinir ağları insan anlayışıyla ilgili karakteristikler sergilerler. Özellikle, bu ağlar tecrübeden öğrenebilirler ve kendi ağırlıklarının aratan ayarlardan özel örneklerden genel kavramlar çıkarır. Bu modeller insan zekasıyla ilgili modeller ve yapay görüntü ve konuşma anlama gibi geleneksel hesaplamaların yetersiz kaldığı durumlar için tasarlanmıştır. Yapay sinir ağları ile TSP üzerine geliştirilen modeller etkili sonuçlar vermelerine rağmen diğer metasezgisellerle rekabet edebilir durumda değil [67].

Karınca kolonisi optimizasyon algoritması

Karınca kolonisi algoritması optimizasyonu (Ant Colony Optimization , ACO) bir yapay karınca kolonisinin işbirliği yaparak en iyi çözüme ulaştığı bir metasezgisel yöntemdir. Karıncalar salgıladıkları 'pheromone' maddesini kullanarak bir yiyecek kaynağına giden en kısa yolu kendi kendilerine bulabilme özelliğine sahiptirler. Bu özellikleri sayesinde karmaşık bir optimizasyon problemiyle benzerlik gösteren yol bulma problemlerini çözmektedirler. Karınca Kolonisi Algoritmaları bir sonraki bölümde detaylı olarak incelenmiştir.

4. KARINCA KOLONİSİ OPTİMİZASYONU

İnsanoğlu yaşamını sürdürmek için tabiata muhtaçtır. Beslenmeden barınmaya, ulaşımdan korunmaya tüm ihtiyaçlarını tabiata bağlı olarak halledebilmektedir. Onu diğer canlılardan ayıran akli sayesinde geliştirmiş olduğu aletler, makineler ve teknolojiler de, hammaddeleri itibariyle tamamen, model alma konusunda da çoğunluk itibariyle tabiattaki varlıklar ve olaylara dayanmaktadır. Günümüzün gelişen teknolojisi de bu alışkanlıktan vazgeçilmesine sebep olmamıştır. Anlaşılan o ki, insanoğlu varolduğu sürece çevresindeki diğer canlılar ve cansız varlıklara doğrudan ve dolaylı olarak bağımlı olacaktır. Değişen şartlarla birlikte ortaya çıkan farklı problemlerinin çözümünde, birikim ve tecrübelerinin yetersiz kaldığı yerde yine başını çevirip bakacağı yer çevresi, tabiat, olacaktır. Karınca koloni optimizasyonu da, böyle bir ihtiyacın çözümüne yönelik kafa yoran bilim adamlarının, tabiattaki gerçek karıncaların yol bulma kabiliyetlerini taklid ederek geliştirmiş oldukları bir algoritmadır [69].

Karınca Koloni Optimizasyonu (KKO), eniyileme problemlerinin çözümü için gerçek karıncalardan esinlenilerek geliştirilmiş bir yapay sistemdir. Konunun temellerinin atıldığı yayın 1991 yılında Dorigo, Maniezzo ve Colormi tarafından hazırlanan bir teknik rapordur. İlk KKO sistemi ise Karınca Sistemi (KS) adıyla Dorigo'nun doktora tezinde (1992) ele alınmıştır. KS, gezgin satıcı ve kuadratik atama problemlerine uygulanmıştır. Çizelge 4.1' de konuyla ilgili yapılan temel çalışmalar kronolojik olarak gösterilmiştir. Tablo hazırlanırken KKO ve geliştirilmiş modelleri dikkate alınmıştır[69].

Çizelge 4.1. Kronolojik olarak Karınca Koloni Optimizasyon Algoritmaları [69].

Algoritma	Yazarlar	Yıl
Ant System (AS)	Dorigo ve diğerleri	1991
	Dorigo	1992
	Dorigo ve diğerleri	1996
Elitist AS	Dorigo	1992
	Dorigo ve diğerleri	1996
Ant-Q	Gambardella ve Dorigo	1995a
Ant Colony System	Dorigo ve Gambardella	1996a
	Dorigo ve Gambardella	1997a
	Dorigo ve Gambardella	1997b
MAX-MIN AS	Stützle ve Hoos	1996
	Stützle	1999
	Stützle ve Hoos	2000
Rank-based AS	Bullnbeimer ve diğerleri	1999b
ANTS	Maniezzo	1999
BWAS	Cordon ve diğerleri	2000
Hyper-cube AS	Blum ve diğerleri	2001
	Blum ve Dorigo	2004
GKKO	Ksekintürk ve Söyler	2006
LACO	Bin ve diğerleri	2009

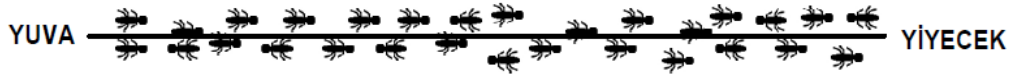
Karınca koloni optimizasyonu ilk olarak gezgin satıcı problemine uygulanmıştır. İlk olarak geliştirilen KS ve modifiye edilmiş versiyonları farklı bilim ve uygulama sahalarında birçok probleme uygulanmış ve başarılı sonuçlar elde edilmiştir. Çizelge 4.2’ de KKO ve geliştirilmiş versiyonlarının uygulandığı alanlar, kaynakları, yazarları ve yıllarıyla beraber yer almaktadır.

Çizelge 4.2. Karınca Kolonisi Optimizasyon Uygulamaları [69].

Problem Tipi	Problem Adı	Yazarlar	Yıl
Rotalama	Gezgin Satıcı	Dorigo ve diğerleri	1991,1996
		Dorigo ve	1997
		Stützle ve Hoos	1997, 2000
		Yang ve diğerleri	2008
	Araç Rotalama	Gambardella ve diğerleri	1999
		Reimann ve diğerleri	2004
		Bin ve diğerleri	2000
	Sipariş İzleme	Gambardella ve Dorigo	2000
Arama	Grafik Boyama	Costa ve Hertz	1997
		Maniezzo	1999
	Karesel atama	Stützle ve Hoos	2000
	Ders çizelgeleme	Socha ve diğerleri	2002,2003
	Kuadratik atama	Demirel ve Toksarı	2006
Çizelgeleme	Total weighted tardiness	Besten ve diğerleri	2000
		Merkle ve Middendorf	2003
	Proje çizelgeleme	Merkel ve diğerleri	2002
		Abdallah ve diğerleri	2009
	Open shop	Blum	2005
	Akış Tipi çizelgeleme	Yağmahan ve Yenisey	2008
	Entegre proses planlama ve çizelgeleme	Leung ve diğerleri	2009
Subset	Set covering	Lessing ve diğerleri	2004
	1-cardinality trees	Blum ve Blesa	2005
	Multiple knapsack	Leguizam'on ve Michalewicz	1999
		Min ve diğerleri	2008
	Maximum elique	Fenet ve Solnon	2003
Diğer	Kısıt sağlama	Solnon	2000, 2002
		Afshar	2007
	Sınıflandırma kuralları	Parpinelli ve diğerleri	2002
		Martens ve diğerleri	2006
	Bayesian ağlar	Campos ve diğerleri	2002a, 2002b
	Docking	Korb ve diğerleri	2006
	Sürekli dinamik programlama	Tfaily ve Siarry	2008
	Tamsayılı doğrusal olmayan programlama	Schlüter ve diğerleri	2009

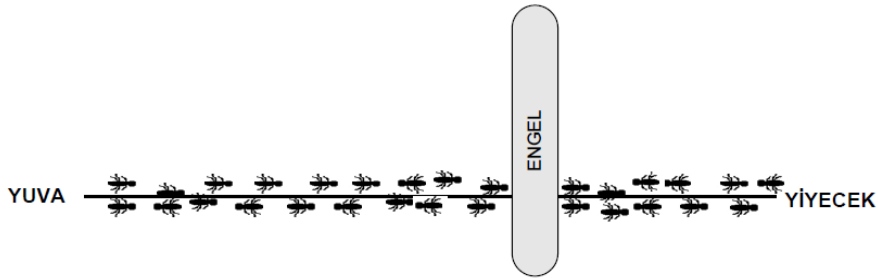
4.1. Gerçek Karıncaların Davranışları

Gerçek karıncalar, yuvaları ile yiyecek buldukları yer arasında en kısa yolu bulma kabiliyetine sahiptirler [70, 71]. Bunu yaparken görme duyularını kullanmamaktadırlar [72]. Ayrıca yollarının üzerine bir engel konulduğunda ya da coğrafi olaylar sebebiyle mevcut yolda bir değişiklik gerçekleştiğinde, yeni en kısa yolu bularak bu yeni duruma adaptasyon sağlamaktadırlar. Şekil 4.1’de karıncaların yuvalarıyla yiyecek kaynağı arasında izledikleri yol görülmektedir.



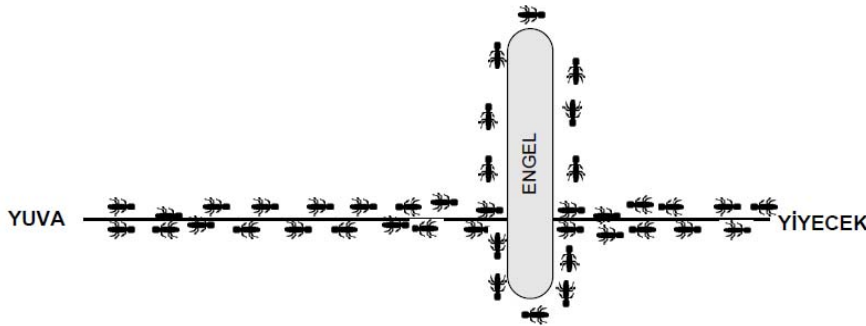
Şekil 4.1. Gerçek Karıncaların Yuvaları ile Yiyecek Kaynağı Arasında Yol Almaları [69].

Karıncaların belli bir yolu oluşturmaları ve her seferinde bu yolu takip etmeleri, yol üzerine bıraktıkları feromon sayesinde mümkün olmaktadır. Görme duyuları gelişmemiş olan karıncalar feromonun kokusunu takip etmektedirler. Yani feromon miktarının çok olduğu güzergâhı az olana tercih etmektedirler. Bu aynı zamanda karıncaların en kısa yolu bulmaları konusunda da bize ipucu vermektedir. Yollarının üzerine üstünden geçme imkanlarının olmadığı bir engel konulduğunda yol alternatifleri ikiye çıkmaktadır (Şekil 4.2).



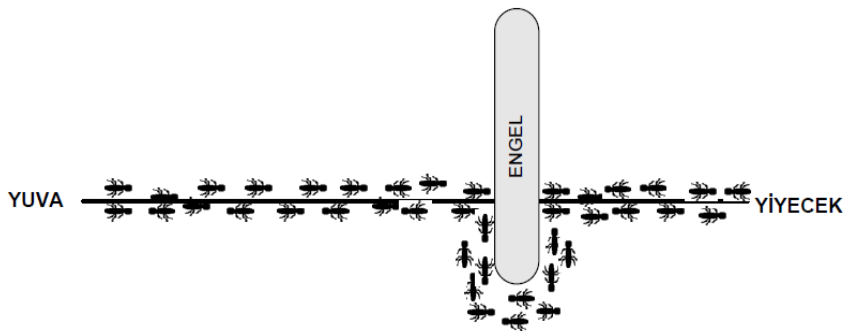
Şekil 4.2. Gerçek Karıncaların Yolları Üzerine Engel Konulması [69].

Karıncalar yolları üzerine konulan engelle karşılaştıklarında, takip edebilecekleri bir feromon olmadığından, sağ veya sol tarafa gitme konusunda eşit olasılıkla hareket edeceklerdir. Dolayısıyla yaklaşık olarak karıncaların yarısı sağ taraftan, yarısı sol taraftan yollarına devam edeceklerdir (Şekil 4.3).



Şekil 4.3. Gerçek Karıncaların Engel Karşısında İlk Etapta Yol Tercihleri [69].

Engelin kısa tarafını tercih eden karıncaların birim zamanda yol üzerine bırakacakları feromon miktarı daha çok olacaktır. Böylelikle zaman içerisinde kısa yolu tercih eden karınca sayısı artacaktır. Bu pozitif geri besleme prosesi sayesinde, belli bir süre sonra tüm karıncalar, engelin kısa tarafını kullanır olacaklardır (Şekil 4.4).



Şekil 4.4. Gerçek Karıncaların Zamanla Kısa Yolu Bulmaları [69].

Gerçek karıncaların yol alma hızları ve feromon bırakma miktarları yaklaşık olarak aynı olmasına rağmen, zamanla kısa yolda biriken feromon miktarının daha fazla olmasını şöyle açıklamak mümkündür: Uzun taraftan yol alma zamanı daha uzun sürmektedir. İki taraflı geliş gidişin olduğu

düşünüldüğünde, belli bir zaman diliminde toplam yoldan geçen karınca sayısı ve dolayısıyla bırakılan feromon miktarı kısa yolda daha fazla olacaktır.

4.2. Yapay Karıncalar ve Karınca Koloni Algoritması

4.2.1. Yapay karıncalar

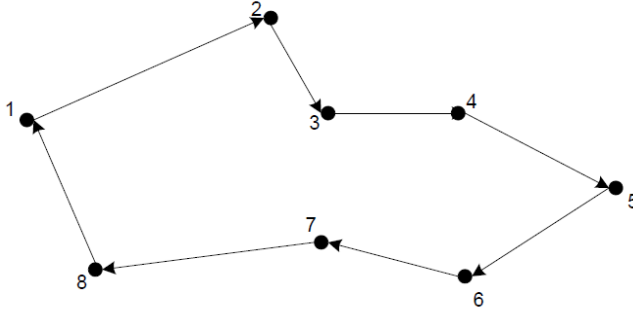
Her ne kadar algoritma önerilirken gerçek karıncalardan ilham alınsa da algoritmanın icrasında kullanılan yapay karıncalar gerçek karıncalardan bazı bakımlardan farklılıklar gösterir [73].

- Yapay karıncalar hafıza özelliğine sahiptir. Tabu listesi adı verilen hafıza sayesinde yapay karıncalar hangi şehirlere daha önce uğradıklarını bilir. GSP problemleri çözülürken uğranan bir şehir tabu listesine alınır ve o şehre bir daha uğranmaz.
- Gerçek karıncalar neredeyse kör oldukları halde yapay karıncalar görme yeteneğine sahiptir. Çünkü yapay karıncalar seçebilecekleri bir sonraki şehrin koordinatları ve uzaklığı hakkında matematiksel bilgiye sahiptir.
- Yapay karıncalar gerçek karıncalardan farklı olarak ayrık bir araştırma uzayında bulunur. Gerçek karınca yuvasına yiyecek ararken sürekli bir araştırma uzayı oları yeryüzü üzerinde sonsuz sayıda noktadan ve yoldan geçebilir. Oysa yapay karıncalar, hafıza ve zaman kısıtlaması gibi temel nedenlerle sınırlı sayıda ayrık veri kümesinden oluşan araştırma uzayında çözüm aramakta kullanılır.
- Yapay karıncalar ayrık zamanlı bir ortamda yaşarlar. Sistemde her yapay karınca ayrı ayrı zaman aralıklarında adım adım sırayla yol alır.

4.2.2. Karınca koloni algoritması

Algoritmanın temel olarak işleyişi anlatılırken GSP kullanılacaktır. Öncelikle GSP problemi hakkında genel olarak bilgi verilecektir.

GSP, doğrusal olmayan eniyileme problemlerindendir [74, 75]. Bir satış elemanının ya da bir aracın belli bir noktadan başlayıp, tüm müşterilere uğradıktan sonra yine aynı noktaya, toplam mesafeyi en az kılacak şekilde dönmesi şeklinde tanımlanabilir (Şekil 4.5).



Şekil 4.5. Gezgin Satıcı Problemi

$G = (N,A)$ bir ağ olmak üzere, A bir düğüm kümesini ve $A = \{(i,j) \mid i,j \in N, i \neq j\}$ bir ark kümesidir. Bu durumda probleme ait tüm mümkün çözümlerin sayısı ilk düğüm sabit olmamak koşuluyla ve asimetric GSP için $N!$ dir. C_{ij} , i düğümünden j düğümüne gidiş maliyeti (mesafe veya süre) ve X_{ij} karar değişkenidir:

$$X_{ij} = \begin{cases} 1, & \text{eğer } i \text{'den } j \text{'ye gidiliyorsa} \\ 0, & \text{aksi durumda} \end{cases} \quad (4.1)$$

Buna göre problemin tamsayılı programlama modeli aşağıdaki gibidir:

$$\min \sum_{(i,j) \in A} c_{ij} x_{ij} \quad (4.2)$$

Kısıtlar:

$$\sum_{j \in N} x_{ij} = 1 \quad \forall i \in N \quad (4.3)$$

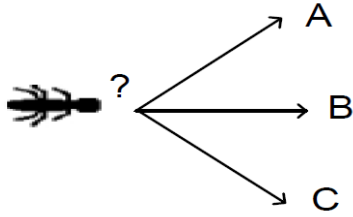
$$\sum_{i \in N} x_{ij} = 1 \quad \forall j \in N \quad (4.4)$$

$$x_{ij} \in \{0,1\} \quad (i, j) \in A \quad (4.5)$$

Amaç fonksiyonu (Eş. 4.2) toplam tur maliyetidir. Kısıt (Eş. 4.3) ve (Eş. 4.4) müşterinin sadece bir defa ziyaret edilmesi gerektiğini ifade etmektedir ve uygun olmayan çözümleri engellemektedir.

Düğüm sayısı arttıkça çözüm alternatifi üssel olarak artmakta ve problem, deterministik yöntemlerle çözümü zor hale gelmektedir. Örneğin 16 şehirden oluşan Homer's Ulysses problemi için 653,837,184,000 alternatif rota olduğu ve en iyi çözümün 92 saatlik bir taramadan sonra bulunabildiği Grötschel ve Padberg (1993) tarafından raporlanmıştır. Bu durumda devreye sezgiseller ve meta-sezgisel yöntemler girmektedir. Geliştirilmiş birçok meta-sezgisel GSP problemlerinin çözümünde kullanılmıştır. Bunlardan bazıları, genetik algoritma [77], yapay sinir ağları [76], benzetilmiş tavlama [78], karınca kolonisi algoritmasıdır [79].

KKO ile GSP probleminin çözümü için ilk olarak düğümlerden biri başlangıç noktası olarak belirlenmektedir. Yapay karıncaların ilk düğümden sonra gidecekleri düğüm eşit olasılıkla belirlenmektedir (Şekil 4.6).



Şekil 4.6. Yatay Karıncanın İlk Adımda Yol Tercihi (Eşit Olasılık)

N düğüm sayısı olmak üzere, yapay karıncaların başlangıç düğümünden gidecekleri düğümü belirlemeleri $1/N-1$ olasılıkla olacaktır. Her düğümden sonra bu olasılık değeri artacaktır. Algoritmanın ilk iterasyonunda karıncaların turları bu şekilde tamamlanacaktır.

Gerçek karıncalarda olduğu gibi, belirlenen sayıdaki karıncaların tümü için geçiş yapmış oldukları yolların feromon miktarları, karıncanın toplam tur uzunluğuyla ters orantılı olarak arttırılır. İlk turların tamamlanmasından ve feromon miktarlarının güncellenmesinden sonra ikinci iterasyona geçilir. Artık karıncalar yol tercihlerini eşit olasılık yerine feromon miktarlarına bağlı olarak gerçekleştireceklerdir. Yol tercih kuralları geçiş kuralı olarak adlandırılmaktadır.

Karıncalar algoritmalarının temel özellikleri

Karıncalar Algoritmalarının temel özellikleri: [73]

Zeki davranış biçimi: Zekâ tam olarak tanımlanamayan bir kavramdır. Fakat temelde kabul gören tanım şudur; *zekâ üstün özellikli bir arama yöntemidir.* İnsanların karşılarına çıkan problemlere karşı çok geniş bir araştırma uzayında çok az bir deneme ile makul bir zaman içinde yeterince iyi çözümler

üretiyor olması *zeki davranış biçimi* olarak kabul edilir. Buradan hareketle insanların zeki olarak kabul ettiği davranış biçimini gösteren algoritmalara *zeki algoritmalar* denir.

Yukarıdaki tanıma göre tek başına bir karıncanın zeki davranış biçimi göstermediği açıktır. Fakat koloni bir bütün olarak karşısına çıkan engellere adaptif yapıda ve doğru yönde kararlarla tepki vererek çok kombinasyonla bir güzergâh problemini çözebilmektedir. İşte bu yüzden gerçek karınca kolonisini model alan karınca algoritmasının yapay zeka vasfı taşıdığı söylenebilir.

Kombinasyonel Optimizasyon: Bazı problemler için çok geniş bir araştırma uzayı söz konusudur ve bu problemlerin kesin çözümünü bulmak mümkün değildir. Bu nedenle makul bir zamanda yeterince iyi bir çözüm bulabilmek amaçlanır. Problem kısıtlamaları çerçevesinde bulunabilen en iyi çözüme optimum çözüm adı verilir. Fakat optimum çözümün kesin olarak en iyi çözüm olduğu garanti edilemez. Doğadaki karıncalar yuva ile yiyecek kaynağı arasında güzergah oluşturabilecek sayısız yol kombinasyonu arasından eniyi çözüm olarak ifade edilebilecek bir çözüm bulabilirler. Bu yönüyle bakıldığında karıncalardan esinlenerek yapılan karınca algoritmasının kombinasyonel bir optimizasyon algoritması olduğu açıktır.

Sezgisel ve Olasılık Tabanlı: Bir engelle karşılaşan gerçek karıncalara bakıldığında hepsinin feromonun en yoğun olduğu yerden geçmediği, az bir kısmının da istisnai olarak feromonun az olduğu yolu seçtikleri gözlenmiştir. Karıncaların tercihi tamamen feromon yoğunluğu olduğu tarafa geçişi mümkün kılan bir kesinlikte değildir. Bunun yerine karıncaların, feromonun daha yoğun olduğu yerlere daha büyük bir olasılıkla geçtiğini kabul etmek isabetli bir tercih olacaktır. Bu yönüyle karıncaların davranışı olasılık tabanlıdır. Fakat tek bir karıncanın olasılık tabanlı tercihi dağınık yapıda grup davranışını, bir başka deyişle koloninin doğru yola yönelmesini

sağlamaktadır. Burada yönlendirici olan unsur feromon bulgusudur. Bu nedenle karıncaların grup olarak davranışı doğru çözüme yakınsayan nitelikte sezgiseldir.

Adaptif Yapı: Güzergâhı engelle bozulan bir karınca, o engeli hangi taraftan aşacağına dair bölgesel veya bütün bir güzergâhı nasıl tamamlayacağına dair küresel bir bilgiye sahip değildir. Engelin hangi taraftan yolu uzattığını da başlangıçta bilemez. Fakat engeli kısa yoldan aşan tarafta birim uzunluk başına düşen feromon yoğunluğunun fazlalığı karıncalar için yol gösterici olacaktır. Dolayısıyla sonraki karıncalar için kısa taraf daha çok tercih edilecektir. Buradan açıkça anlaşılmaktadır ki; gerçek karıncalar için çevrede meydana gelen değişimlere doğru yönde adaptasyon sağlamada feromon salgısıyla birlikte mesafe bilgisi dolaylı olarak etkili olmaktadır.

Pozitif Geri besleme: Karıncaların geçtikleri yollara bıraktıkları feromon maddesi pozitif geri besleme bilgisi sağlamaktadır. Bu bilgi sonraki karıncalar için o yolun seçilebilirliğini artırmaktadır.

Negatif Geri besleme: Doğada karıncaların geçtikleri yerlere bıraktıkları feromon maddesi zamanla dış etkenler nedeniyle azalmakta ve sürekli olarak buharlaşmak suretiyle uçmaktadır. Bu nedenle güzergâhı uzattığı için kullanılmayan bir yolun seçilebilirliği gittikçe azalmakta ve sonunda ortadan kalkmaktadır. Bu yönüyle feromonun buharlaşarak uçuşması sonraki karıncalar için kötü olar yolun seçilme ihtimalini azaltan ve ortadan kaldıran negatif bir geri besleme bilgisi olarak vazife görür. Bu durum insan beyninde kullanılmayan bilgilerin hafızadan atılması (unutmak) ve çok kullanılan bilgilere erişimin hızlanması (çabucak hatırlamak) olayına benzetilebilir.

Etkileşim: Yaşam çevresinde iz bırakmak suretiyle gerçekleştirilen iletişim türüne etkileşim denir. Koloni bireyleri arasındaki etkileşim rast gele olan birey davranışını doğru çözüme yakınsayan bir koloni davranışına

dönüştürür. Karıncalar arasındaki etkileşim maddesi salgıladıkları feromondur, etkileşim aracı ise koku alma duyularıdır.

Geçiş kuralı

Feromon miktarlarını dikkate alarak yapılacak yol tercihinde iki alternatif söz konusudur. İlk seçenek q_0 olasılıkla feromonun en yoğun olduğu yolun seçilmesidir. q_0 parametresi genel olarak %90 olarak belirlenir. $\tau(i,j)$, i ve j noktaları arasındaki feromon miktarı, $\eta(i,j)$, seçilebilirlik parametresi, i ve j noktaları arasındaki mesafenin tersi ($1/\delta(i,j)$), α ve β ayarlanabilir parametreler olmak üzere, i noktasında bulunan bir k karıncasının gideceği nokta u , aşağıdaki gibi seçilmektedir:

$$j = \arg \max_{u \in J_k(i)} \left\{ [\tau(i,u)]^\alpha [\eta(i,u)]^\beta \right\} \quad \text{eğer } q \leq q_0 \quad (4.6)$$

Denklem (Eş. 4.6)' da $J_k(i)$, henüz k karıncası tarafından ziyaret edilmemiş, i ' den gidilebilecek noktalar kümesidir.

İkinci alternatif ise gidilmesi mümkün olan yollardan birini, yollardaki feromon izleriyle orantılı olarak seçmektir. Bu şekilde yol seçimi olasılığı $1 - q_0$ ' dir.

$P_k(i, j)$ i noktasında iken j noktasının seçilme, j ' ye gitme olasılığıdır ve Denklem (Eş. 4.7)' deki gibi hesaplanmaktadır.

$$p_k(i, j) = \begin{cases} \frac{[\tau(i, j)]^\alpha [\eta(i, j)]^\beta}{\sum_{u \in J_k(i)} [\tau(i, u)]^\alpha [\eta(i, u)]^\beta}, & j \in J_k(i) \\ 0, & \text{diğer durumlarda} \end{cases} \quad (4.7)$$

Denklem (Eş. 4.7)'de α ve β iki parametre olup feromon izinin $\eta(i,j)$ ' nin birbirlerine göre göreceli üstünlüklerini belirler. Bu olasılıklara bağlı olarak

gidilecek noktalar dolayısıyla tur belirlenmektedir. Feromonun yoğun olduğu yolların seçilme olasılığı daha yüksektir.

Feromon güncellemesi

Feromon güncellemesinin amacı çözüm uzayının taranmasıdır. Güncelleme işlemi tüm karıncalar turlarını tamamladıktan sonra yapılır. İki temel bileşeni vardır[80]:

- 1) Tüm yollardaki feromonların, buharlaşma oranı $\rho \in (0,1]$ oranda buharlaştırılması,
- 2) Karıncaların geçtikleri yollardaki feromon miktarlarının, o yolu kullanan karıncanın yol uzunluğuyla ters orantılı olarak artırılması.

Buharlaşma oranı daha önceki çözümlerin öneminin azaltılmasını sağlamaktadır. Bununla amaçlanan yerel eniyiler takılmayı, aramanın yavaşlamasını veya durmasını engellemektir. Ayrıca kötü sonuç veren ark ikililerinin de seçilme olasılıkları azaltılmaktadır.

Yol uzunluğuyla ters orantılı olarak feromon artışı ise, iyi çözümlerin öneminin arttırılmasını temin eder [81].

Feromon yenilemesi klasik ve global olmak üzere iki şekilde gerçekleşmektedir.

Klasik Feromon Güncellemesi

$\tau_{ij}(t)$ iterasyon t' ye kadar biriken (i,j) arkındaki feromon düzeyi, $\Delta\tau_{ij}(t)$ karınca k' nın (i) arkına bırakacağı feromon miktarı ve m karınca sayısı olmak üzere $t+1$ iterasyonuna ait feromon düzeyi Denklem (Eş. 4.8)' deki formülle hesaplanır.

$$\tau_{ij}(t+1) = (1-p)\tau_{ij}(t) + \sum_{k=1}^m \Delta\tau_{ij}^k \quad (i, j) \in A \quad (4.8)$$

L^k k karıncasına ait toplam yol uzunluğu ve T^k k karıncası tarafından gerçekleştirilen tur olmak üzere feromon değişim düzeyi $\Delta\tau_{ij}$ k Denklem (Eş.4.9)' daki formülle hesaplanır.

$$\Delta\tau_{ij}^k = \begin{cases} 1/L^k, & \text{eğer } (i, j) \in T^k \\ 0, & \text{aksi durumda} \end{cases} \quad (4.9)$$

Feromon güncellemesi, turları dinamik olarak değiştirerek geçiş yapılan yolların tercih edilme olasılığını arttırmaktadır. Karıncalar, değişen feromon miktarlarına bağlı olarak her iterasyonda turlarını da değiştirmektedirler. Amaç, sürekli olarak daha kısa turları bulmaktır.

Global Feromon Güncellemesi

Global feromon güncellemesi, geçerli iterasyondaki en iyi sonuca sahip karıncanın izlediği yolun feromon düzeyinin, klasik güncellemeye ilaveten artırılmasıdır. Bununla amaçlanan, iterasyonlarda bulunan en iyi sonuçların belli bir oranda ileriki iterasyonlara aktarılmasıdır. Global feromon güncellemesi ve miktarı sırasıyla Denklem (Eş. 4.10) ve (Eş. 4.11)' deki gibi hesaplanmaktadır.

$$\tau_{ij}(t+1) = \tau_{ij}(t) + \sum_{k=1}^m \Delta\tau_{ij}^k \quad (i, j) \in A \quad (4.10)$$

$$\Delta\tau_{ij}^k = \begin{cases} 1/L^b, & \text{eğer } (i, j) \in T^b \\ 0, & \text{aksi durumda} \end{cases} \quad (4.11)$$

L_b mevcut en iyi turu gerçekleştiren karıncanın tur uzunluğu, T_b ise yine bu karıncanın gerçekleştirdiği tura ait (i,j) arklarının kümesidir.

Bir iterasyon boyunca gerçekleştirilen işlemler yukarıda anlatıldığı gibidir. Aşağıda karınca koloni algoritmasının adımları görülmektedir [82]:

Başlangıç

Tur sayısını belirle,

Karınca sayısını belirle,

α parametresini belirle,

β parametresini belirle,

ρ parametresini belirle,

Her karınca için rassal bir başlangıç şehri belirle,

Her tur için;

Her karınca için;

Bir sonraki düğüm noktasını (Eş. 4.6) ve (Eş. 4.7)' yi kullanarak belirle;

Tur uzunluğunu belirle

(Eş. 4.8) ve (Eş. 4.10)' u kullanarak feromon güncelleştirmesi yap

Dur.

Karınca koloni optimizasyonu ilk kez geliştirildiğinden bu yana üzerinde birçok değişiklikler yapılmıştır. Performansını arttırmaya ve daha farklı problemlerin çözümünde kullanmaya yönelik yapılan modifikasyonlar, farklı isimlerle anılan birçok karınca koloni optimizasyon algoritmasını ortaya çıkarmıştır.

4.3. Geliştirilmiş Karınca Algoritmaları

4.3.1. Karınca sistemi

Karınca sistemi ilk olarak Marco Dorigo tarafından 1992’de yazmış olduğu doktora tezinde ortaya atılmış ve gezgin satıcı problemiyle karesel atama problemine uygulanmıştır. KS’ nin farklı üç farklı versiyonu bulunmaktadır. Bunlar: *ant-density* [83], *ant-quantity* [84] ve *ant-cycle*’ dır [85]. İlk ikisi daha düşük performans gösterdiklerinden KS dendiğinde *ant-cycle* anlaşılmaktadır [86].

KS’de her bir yapay karınca aşağıdaki özelliklere sahiptir:

- Gideceği bir sonraki şehri, yolun uzunluğunun ve yol üzerindeki feromon miktarının bir fonksiyonu olan olasılığa göre belirlemektedir,
- Gidilen bir şehre bir daha uğramak suretiyle uygun olmayan çözümlerin oluşmasını engellemek için, gidilen şehirler tabu listesi adı verilen bir veritabanında tutulur,
- Turlar tamamlandıktan sonra geçilen yollara $(i,j) \in A$ iz olarak da adlandırabileceğimiz feromon atanmaktadır.

$\tau_{ij}(t)$ t anında kadar kenar (i,j) ’de biriken feromon miktarı olmak üzere, her bir karınca t anında gideceği bir sonraki şehri seçecektir. Yeni şehrin seçilmesi $t+1$ anına denk gelmektedir. Bu işlem bir iterasyon olarak adlandırıldığında $(t,t+1)$ zaman aralığında m adet karınca tarafından m adet iki şehir arası transfer gerçekleşmektedir. n şehir sayısı olmak üzere her n iterasyonda karıncalar bir turu tamamlanmış olmaktadır. Bu noktada feromon miktarları formül (Eş. 4.12)’ ye göre güncellenmektedir.

$$\tau_{ij}(t+n) = (1-p)\tau_{ij}(t) + \Delta\tau_{ij} \quad (i,j) \in A \quad (4.12)$$

$$\Delta \tau_{ij}^k = \sum_{k=1}^m \Delta \tau_{ij}^k \quad (i, j) \in A \quad (4.13)$$

$\Delta \tau_{ij}^k$ ($t, t+n$) zaman aralığında kenar (i, j) deki k karıncasının geçişinden dolayı arttırılan feromon miktarını temsil etmekte olup denklem (Eş. 4.14)' e göre hesaplanmaktadır.

$$\Delta \tau_{ij}^k \Delta \tau_{ij}^k(t+1) = \begin{cases} 1/L^k, & \text{eğer } (i, j) \in T^k \text{ ve } (t, t+n) \text{ zaman aralığında} \\ 0, & \text{aksi durumda} \end{cases} \quad (4.14)$$

$t=0$ zamanında feromon miktarı, tüm (i, j) kenarları çok küçük bir değer olan C sabiti miktarınca atanmaktadır. Bu değer, C_{nn} en yakın komşu sezgilene göre hesaplanan tur uzunluğu olmak üzere, $(i, j), \tau_{ij} = \tau_0 = m/C_{nn}$ şeklinde hesaplanabilir. Bu değer çok küçük olması durumunda ilk turlardaki nispeten iyi çözümlere takılma ihtimali vardır. Yüksek olması durumunda ise kötü sonuç veren (i, j) arklarındaki feromonların buharlaşması için iterasyonların geçmesi beklenecektir [86].

$tabu_k$, k karıncasına ait tabu listesi olup, o ana kadar uğramış olduğu şehirlerin olduğu bir veritabanıdır. Ziyaret edilen her şehir listeye eklenir. Gidilecek bir sonraki şehir seçilirken bu liste dikkate alınır. Amaç ziyaret edilen bir şehrin tekrar ziyaret edilmesini engellemek, böylelikle uygun olmayan çözümlerin devre dışı bırakılmasını sağlamaktır. $tabu_{k(s)}$ ise k karıncasının uğradığı s . şehri temsil etmektedir. k karıncasının i şehrinde j şehrine gitme olasılığı (geçiş olasılığı) denklem (Eş. 4.7) ile aynı şekilde hesaplanmaktadır.

$j_k, \{N - tabu_k\}$ 'ya eşit olup gidilebilecek şehirleri göstermektedir. α ve β feromonin seçilebilirliğe karşı göreceli üstünlüğünü belirleyen parametrelerdir.

Böylelikle geçiş olasılığı, t anında seçilebilirlik $\eta(i,j)$ (daha yakın şehirlerin seçilme olasılığı daha yüksektir) ile feromon miktarının (feromon miktarının yoğun olduğu şehirlerin seçilme olasılığı daha yüksektir) parametrelerle belirlenen önem derecelerine göre belirlenmektedir.

Algoritmanın başlangıcında ($t=0$), karıncalar tesadüfî olarak ilk şehirlerine yerleştirilirler (ilk şehrin her karınca için tesadüfî olarak seçilmesi ya da aynı şehirden başlatılmasının performans üzerindeki etkisi Dorigo ve diğerlerinin (1996) eserinde incelenmiştir. Başlangıç şehri çalışmamızda depo olduğundan bu alternatifler üzerinde durulmamıştır). Ayrıca tüm yollara başlangıç feromon değerleri $\tau_{ij}(0)$ atanır. Karıncalara ait tabu listelerine, ilgili karıncaların yerleştirildikleri şehirler eklenir. Daha sonra her bir karınca, (Eş. 4.7)'deki formülle hesaplanan olasılık $p_k(i,j)$ 'ye göre bulunduğu i şehirden seçilecek şehre gönderilir. n iterasyon sonra karıncalar turlarını tamamlamış ve tabu listeleri dolmuş olacaktır. Karıncalara ait tur uzunlukları (L^k) ve feromon miktarlarındaki değişim $\Delta\tau_{ij}^k$ hesaplanır. En kısa karınca turu güncellenir ve tabu listeleri boşaltılır. Bu proses maksimum iterasyona (NCMAX) ulaşılan kadar tekrarlanır.[87].

4.3.2. Elitist karınca sistemi (KSELİT)

Elitist karınca sistemi, KS'nin ilk defa geliştirilmiş versiyonudur [85, 86]. Elitist strateji olarak adlandırılan yöntem, genetik algoritmadaki uygulamalardan esinlenilerek geliştirilmiştir. En iyi tura ait kenarların feromon miktarları, her döngüde KS'de uygulanan standart feromon güncellemesine ilaveten fazladan feromon artırımına tabi tutulmaktadır. Bütün karıncalar turlarını tamamladığında o ana kadar yapıları en iyi turu σ adet karıncanın yapmış olduğu kabul edilir. Yollarda feromon güncellemesi yaparken en iyi tur güzergâhında bulunan yollara σ katsayısı ile orantılı miktarda fazladan $\Delta\tau_{ij}^*$ kadar feromon bırakılır. Burada amaç araştırmayı bulunan en iyi tur etrafında

sürdürmektir. KSelit algoritmasında; feromon güncelleme, toplam feromon değişimi, bir karıncanın bıraktığı feromon miktarı, seçilmiş karıncanın bıraktığı feromon miktarı sırasıyla denklem (Eş. 4.15)-(Eş. 4.18) ile hesaplanmaktadır.

$$\tau_{ij}(t+1) = (1-p)\tau_{ij}(t) + \Delta\tau_{ij} + \Delta\tau_{ij}^* \quad (4.15)$$

$$\Delta\tau_{ij}^k = \sum_{k=1}^m \Delta\tau_{ij}^k \quad (4.16)$$

$$\Delta\tau_{ij}^k = \begin{cases} Q/L^k, & k \text{ i'den } j'ye \text{ geçtiyse} \\ 0, & \text{geçmediyse} \end{cases} \quad (4.17)$$

$$\Delta\tau_{ij}^* = \begin{cases} \sigma \frac{Q}{L^*}, & k \text{ i'den } j'ye \text{ geçtiyse} \\ 0, & \text{geçmediyse} \end{cases} \quad (4.18)$$

4.3.3. ANT-Q

Gambardella ve Dorigo 1995'te, Q-öğrenmeden bazı fikirleri entegre eden KS'nin uzantısı olan Ant-Q algoritmasını önermişlerdir. Bu algoritmadaki en önemli farklılıklardan biri □ öğrenme basamağı parametresinin de göz önünde bulundurularak feromon güncelleme değerlerinin hesaplanmasıdır [89].

Ant-Q Karınca Kolonisi Sistemi'nden karıncaların adım adım çevrim içi feromon güncelleme gerçekleştirmede kullandığı τ_0 değerinde farklılık gösterir. Ant-Q'da, bir k karıncası çevrim içi adım adım aşağıdaki denklemle feromon güncellemesi yürütür [90]:

$$\tau_{ij}(t) = (1-p)\tau_{ij}(t) + \rho\gamma^* \max_{l \in N_j^k} \tau_{jl} \quad (4.19)$$

Daha sonra karınca kolonisi sisteminde olduğu sabit τ_0 değeri kullanmak Ant-Q'daki gibi komplike terim tahmininden dolayı çalışmalar hemen hemen aynı performans elde edilmesine rağmen Karınca Kolonisi Sistemi üzerinde yoğunlaştırılmıştır [90].

4.3.4. Karınca koloni sistemi

Karınca Koloni Sistemi (KKS), Dorigo ve Gambardella (1996) tarafından GSP problemlerinin çözümüne yönelik olarak geliştirilmiş bir karınca koloni meta-sezgiselidir. KKS'nin çalışması şu şekilde gerçekleşmektedir: m adet karınca n adet şehirden birine belli bir başlangıç kuralına göre yerleştirilmektedir. Tüm karıncalar durum geçiş kuralının (state transition rule) da uygulandığı bir süreçte turlarını tamamlamaktadır. Turlar gerçekleştirilirken, karıncaların geçiş yaptığı kenarlarda yerel feromon güncellemesi de yapılır. Turlar tamamlandıktan sonra kenarlara ait feromon miktarları global feromon güncelleme kullanılarak artırılır.

KKS'de durum geçiş kuralı şu şekilde kullanılmaktadır: r şehrindeki k karıncası gideceği şehir u 'yu denklem (Eş. 4.20)' ye göre belirlemektedir:

$$s = \begin{cases} \arg\max \left\{ [\tau(r,u)] [\tau(r,u)]^\beta \right\}, & \text{eğer } q \leq q_0 \\ S, & \text{değilse} \end{cases} \quad (4.20)$$

s gidilecek olan şehrin olasılık dağılımına göre belirlenmesi alternatifini temsil etmektedir. Olasılıkların hesaplanması (Eş. 4.21)' deki gibidir.

$$p_k(r,s) = \begin{cases} \frac{[\tau(r,u)][\tau(r,u)]^\beta}{\sum_{u \in J_k(r)} [\tau(r,u)][\tau(r,u)]^\beta}, & \text{eğer } s \in J_k(r) \\ 0, & \text{diğer durumlarda} \end{cases} \quad (4.21)$$

KKS'ye ait algoritma Şekil 4.7 ' de görölmektedir.

```

Başla
Loop {bu aşamadaki her döngü iterasyon olarak adlandırılmaktadır}
    Her bir karınca başlangıç düğümüne yerleştirilir
    Loop {bu aşamadaki her döngü adım olarak adlandırılır}
        Her bir karınca durum geçiş kuralını uygulayarak bir
        sonraki düğümü seçer
        Lokal feromon güncellemesi yapılır
    Until tüm karıncalar turlarını tamamlayıncaya kadar
    Global feromon güncellemesi yapılır
Until durdurma kriteri sağlanana kadar

```

Şekil 4.7. KKS algoritması

Global feromon güncellemesi tüm karıncalar turlarını tamamladıktan sonra denklem (Eş. 4.22) ve (Eş. 4.23)' e göre hesaplanır.

$$\tau(r,u) \leftarrow (1-\alpha)\tau(r,s) + \alpha \Delta\tau(r,s) \quad (4.22)$$

$$\Delta\tau(r,u) = \begin{cases} 1/L_{gb}, & \text{eğer } (r,s) \in \text{global en iyi tur} \\ 0, & \text{aksi durumda} \end{cases} \quad (4.23)$$

α buharlaşma oranı parametresi. Global feromon güncellemesinin amacı daha kısa turların feromon miktarlarını arttırıp cazip hale getirmektir.

KKS' de global feromon güncellemesine ilaveten her karınca k ' nın geçiş yaptığı her ark (i,j) için geçişi sonrası yapılan yerel feromon güncellemesi söz konusudur. ξ , $0 < \xi < 1$, genellikle 0.1 olarak belirlenen bir parametre ve

$\tau_0 = 1 / C_{mn}$ olmak üzere yerel feromon güncellemesi denklem (Eş. 4.24)' ya göre yapılmaktadır.

$$\tau_{ij} \leftarrow (1 - \xi)\tau_{ij} + \xi\tau_0 \quad (4.24)$$

τ_0 'ın hesaplanmasında kullanılan C_{mn} enyakın komşu algoritmasına göre elde edilen tur uzunluğudur.

4.3.5. Maksimum-minimum karınca sistemi

Maks-Min Karınca Sistemi (MMAS) Stützle ve Hoos tarafından 1997 yılında geliştirilmiştir [88]. MMAS'yi KS'den ayıran iki temel özellikten ilki, her iterasyonda sadece bir karıncanın feromon yenilemesine izin verilerek çözümün sağlanmasıdır. Bu karınca, bir önceki iterasyondaki en iyi çözümü veren karıncadır. İkinci özellik ise, araştırmadaki dalgalanmayı önlemek için, feromon izlerinin sınırlı bir aralıkta $[\tau_{\min}, \tau_{\max}]$ olmasıdır. Feromon izleri üst limitten başlatılır ve bu da başlangıçta yüksek oranda iyileşme sağlar. MMAS'de alt ve üst limitler uygun seçilmediği takdirde tüm karıncalar aynı yoldan gitmekte ve bu da iyi bir çözümün bulunmasını engellemektedir. Alt ve üst sınırlar aşağıdaki formüllerle hesaplanır [91].

$$\tau_{\max} = \frac{1}{\rho} * \frac{1}{L^b} \quad (4.25)$$

$$\tau_{\min} = \frac{\tau_{\max}}{2n} \quad (4.26)$$

Burada, L_b bulunan en iyi turun uzunlugunu, ρ feromon buharlaşma katsayısını, n ise karıncaların hareket ettikleri sistemdeki nokta sayısını göstermektedir.

4.3.6. Rank temelli karınca sistemi

Bullnheimer, Hartl ve Strauss (1997a) KS'nin başka bir değiştirilmiş hali olan Rank Temelli Karınca Sistemini (KSrank) önerdiler. Buna göre :

Tur uzunluklarına göre m karınca sıralanır ($L1(t) \leq L2(t) \leq \dots Lm(t)$) ve ilk $\sigma-1$ karıncanın ziyaret ettiği kenarlar ziyaret eden karınca sırasına orantısal feromon miktarı alır.

İzin başlangıcından beri en iyi turu üreten karıncanın kullandığı kenarlar ayrıca ek feromon alırlar (ve bu durum KS'de önerilen "elitist karıncalar" feromon güncellemesi ile aynıdır). Bir karıncanın feromon düzeyi karıncanın sıra değerine bağlıdır [91].

4.3.7. Çoklu karınca sistemi

Çoklu koloni yaklaşımı, iki veya daha fazla koloninin çözümünün Obirleştirilmesi esasına dayanır. Bu konudaki iki temel yaklaşımdan ilki olan heterojen yaklaşım tüm kolonilerin birbirinden farklı davranış sergilemesi esasına dayanmaktadır ve çok kriterli optimizasyon problemlerinin çözümünde kullanılmıştır. İkinci yaklaşımda tüm koloniler birbirine paralel olarak çalışırlar ve her t jenerasyon sonunda koloniler arasında feromonlar vasıtası ile karşılıklı bilgi alışverişi olur. Çoklu Karınca Sisteminde bilgi alışverişi için önerilen 4 yöntem vardır [91]:

1. Her bilgi alışverişinde global en iyi çözümler hesaplanır ve tüm kolonilere gönderilir. Böylece her koloni yeni bir yerel en iyi çözüme sahip olur.
2. Koloniler arasında bir köprü kurulur ve her bilgi alışverişi adımı, her koloni yerel en iyi çözümleri en yakınındaki bir ya da daha fazla koloniye gönderir.

3. Bir önceki seçenekteki bilgiler kullanılarak her bilgi alışverişi adımında, bir kolonideki en iyi m_b karınca, en yakınındaki en iyi m_b karıncayla karşılaştırılır ve $2m_b$ karıncanın en iyi m_b tanesi alınarak feromon güncellemesinde kullanılır.
4. Bu seçenek 2. ve 3. seçeneklerinin birleştirilmesinden oluşur.

4.3.8. Melez karınca kolonisi optimizasyonu

KKO ile SOP-3-değişimi yaklaşımının birleştirilmesinden elde edilen yeni algoritmaya verilen isimdir. KKO algoritmasına ek olarak yerel arama içerir. Melez Karınca Kolonisi Optimizasyonunda (MKKO) yerel arama şöyle uygulanır: Bir kere karıncalar kendi çözümlerini inşa ederler, daha sonra yerel arama rutininin uygulaması ile her çözüm kendi yerel optimumuna taşınır. Daha sonra yerel olarak eniyi çözümler feromon iz güncelleme kuralına göre kenarlardaki feromon izlerinin güncellenmesinde kullanılırlar [92].

4.3.9. Global karınca kolonisi optimizasyonu

Global Karınca Kolonisi Optimizasyonu (GKKO), tüm düğümlere uğrama kısıtını içermeyen problemlerin çözümüne yönelik olarak geliştirilmiş olan bir KKO algoritmasıdır. Geliştirilmiş birçok KKO sisteminden farklı olarak, karıncaların tam bir tur yapma zorunlulukları yoktur. Herhangi bir düğümden başlayarak ve herhangi bir ya da birden fazla düğüme uğrayarak bir çözüm alternatifi geliştirmek mümkündür. Geçiş kuralı bir fark dışında mevcut KKO algoritmaları ile aynıdır. Bu fark alternatifler arasından seçilenin iki düğüm arasındaki bir yol değil bütün bir tur olmasıdır. Feromon güncellemesi GKKO'da sadece global olarak yapılmaktadır. Ancak bu güncelleme sadece en iyi tura sahip karıncanın yolu için değil tüm karıncaların yolları için yapılmaktadır [91].

4.3.10. Karıncalar

Karıncalar (ANTS) 1999'da Maniezzo tarafından önerilmiş KS'nin uzantısı olan bir algoritmadır. ANTS iz dağılımını başlangıç durumuna getirme ve kullanma için çekicilik fonksiyonu gibi genel algoritmanın elementlerini belirtir. Bu durum, ağaç araştırma algoritmaları ile yapıda benzer hâsıl olan algoritmayı yapan genel KKO çerçevesinin varvasyonunu meydana getirir. Aslında, ANTS'ı ağaç araştırma algoritmasından ayıran temel özellik araştırma ağacının tamamlanmamış (yaklaşık) araştırması tarafından ve hareket edeceği durumun olasılıksal (non-deterministik) seçimi tarafından yerine getirilen komple gerileme mekanizmasının azlığıdır. Bu durum ayrıca ANTS adının arkasındaki gerekçedir. ANTS kelimesi Yaklaşık Nondeterministik Ağaç Arastırma'nın (Approximated Nondeterministic Tree Search) İngilizce karşılıklarının baş harflerinden oluşmuştur. ANTS algoritması iyi sonuçlar elde edebilmek için, bir ajan (karınca) geçmişte uygulanmış ve arzu edilir sonuçları üretmede (geçmiş çözümlere ait bilgileri kullanma) efektif olmuş faaliyetleri seçmelidir, ama bu faaliyetleri keşfetmek için önce seçilmemiş olan faaliyetleri de denemelidir (yeni çözümler araştırma). Kullanma ve araştırmanın her ikisi de görevde başarısız olmadan sadece takip edilir, buyüzden, ANTS algoritması araştırmayı kolaylaştıran durgunluğu önleme prosedürü ile hareketlerin arzu edilirliliğini belirleyen iz ve çekicilik tabanlı olasılık tanımlama mekanizmasını entegre eder [93].

5. TÜRKİYE' DE ÇOK MODLU TAŞIMACILIKTA EN KISA YOLUN BULUNMASI

Artan rekabet koşullarıyla birlikte işletmelerin müşteri memnuniyeti ve rekabet avantajı sağlamak için maliyetlerini düşürme gayretleri oldukça önem kazanmıştır. Taşımacılık faaliyetleri her iki unsur açısından da büyük önem arz etmektedir. Müşteri taleplerinin zamanında ve en az maliyetle karşılanması şirketlerin önemli hedeflerindendir. Bu noktada devreye giren rota belirleme problemi dağıtımın en ucuz, en kısa, en etkin şekilde yapılabilmesi için en etkin rotalama alternatifinin bulunmasını içermektedir. Rota problemlerinin türlerinden biri de GSP'dir. GSP' de rotanın kısa olmasının yanı sıra maliyeti minimize etmek için taşımacılık modlarının seçimi de önemlidir.

Taşımacılık, " kişilerin ve malların yer ve/veya zaman faydası sağlamak üzere ekonomik, hızlı ve güvenli yer değiştirmesi" olarak tanımlanabilir. Kuşkusuz ülke ekonomisinin dinamizmi, taşımacılığın aldığı biçim ile doğrudan ilişkili olup, taşımacılık alt sistemlerinin birbirine karşı üstünlüklerinin birbirlerini tamamlayacak şekilde kullanılması ile, ülke genelinde katma değer oluşumuna büyük katkılar sağlayacaktır. Tüm dünyada taşımacılık sistemlerinin koordineli ve kombine yapıda faaliyet göstermesi ekonomik gelişmenin yadsınamaz bir unsuru olarak kabul edilmektedir.

Taşımacılık sistemleri konfor, hız ve güvenilirlik yönünden devamlı gelişmeler kaydetmiş, bu sistemler içinde toplu taşımadaki üstünlükleri ve enerji kullanımındaki tasarrufu dolayısıyla genellikle gelişmiş ülkelerde demiryolu taşımacılığına ayrı bir önem ve yer verilmiştir. Ülkemizde ise rakip taşımacılık türlerinin hızla gelişmesi karşısında gerekli önlemlerin alınmaması, demiryolu taşımacılığının geleceğini tehdit eder hale gelmiştir [94].

Karınca kolonisi algoritmasının ilk uygulaması GSP üzerine olmuştur. Bunun nedeni en kısa yol mantığının karınca kolonisi davranışlarıyla birebir uyumlu olması ve kolayca adapte edilebilir olmasıdır. Ayrıca kolayca anlaşılabilir olması ve çok karmaşık teknikler ve hesaplar gerektirmemesi de algoritmanın en çok uygulandığı problem olmasını sağlamıştır [89]. Geniş bir uygulama alanı olması ve bahsedilen avantajlarından dolayı bu çalışmada da karınca kolonisi algoritmasının GSP' ye uygulamasına yer verilmiştir.

Bu çalışmada, Türkiye' de en ucuz taşımacılığı gerçekleştirmek için karayolu ve demiryolu türlerini içeren çok modlu taşımacılık çözümünde KKA kullanılarak rotaların belirlenmesi hedeflenmiştir.

Elde edilen çözümün kalitesini geliştirmek için modelde yer alan KKA formülasyonunda çözümün mesafe ve feromon bağımlılıklarını etkileyen parametreler değiştirilerek elde edilen sonuçlarda iyileştirmeler elde edilmeye çalışılmıştır.

5.1. Neden Demiryolu Taşımacılığı

5.1.1. Artan ulaşım talebi

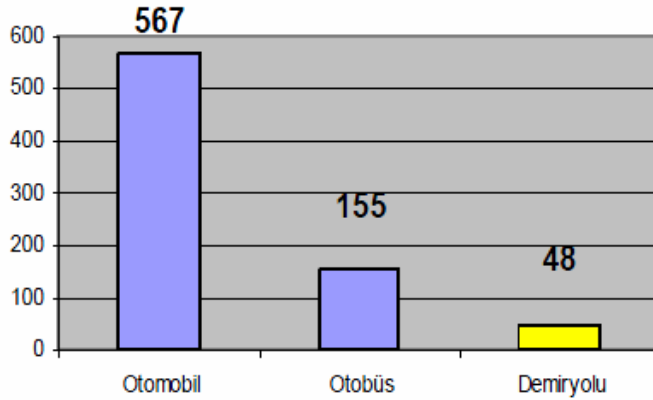
Türkiye'de son 25 yıldaki büyüme eğilimlerinin sürmesi durumunda 2020 yılında ;

- Yocu trafiğinin bugünkü düzeyinin yaklaşık 3,3 katına (540 Milyar Yolcu-Km)
- Yük tafiğinin ise 2,5 katına (300 Milyar Ton- Km) çıkacağı tahmin edilmektedir.

Yılda ortalama % 1,6 artan nüfusa paralel olarak sürekli yükselen ulaşım talebinin karşılanabilmesi için, demiryolu ağının mutlak genişletilmesi gerekmektedir.

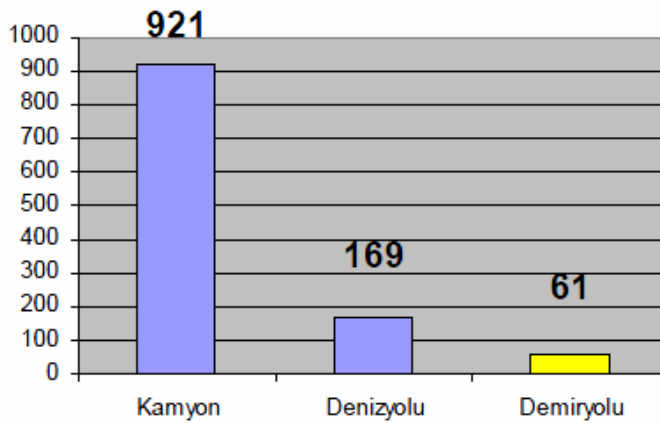
5.1.2. Enerji tüketimi

- Yolcu Taşımacılığında Demiryolu, otobüse göre 3 kat, otomobile göre de yaklaşık 12 kat daha az enerji tüketmektedir.



Şekil 5.1. Yolcu taşımacılığında birim nakliyat başına enerji tüketimi (kcal/kışi-km) [2]

- Yük Taşımacılığında ise Demiryolu, denizyoluna göre 2,7 kat, kamyonu göre de yaklaşık 15 kat daha az enerji tüketmektedir.



Şekil 5.2. Yük taşımacılığında birim nakliyat başına enerji tüketimi(kcal/ton-km) [2]

5.1.3. Trafik sıkışıklığı

Günümüzde her ay ortalama 70-80 bin yeni araç daha trafiğe çıkmaktadır. 1985 yılında 51,4 milyon olan nüfusumuz 2007 yılına gelindiğinde yaklaşık 70 milyona, motorlu taşıt sayısı ise 2,4 milyonda 12,5 milyona yükselmiştir. 22 yıllık dönemde nüfus %36 artarken motorlu taşıt sayısı %520 artış göstermiştir [3].

5.1.4. Trafik kazaları

Çizelge 5.1. Trafikte kaza durumu [2].

Ulaştırma Türü	Taşıt-km (Milyon)	Kaza Sayısı	100 Milyon Taşıt-km ye Düşen Kaza Sayısı	Oran
Demiryolları	13 804	478	3,46	1
Karayolları	40 504	54 746	135,16	39

Çizelge 5.2. Trafikte kaza sonucu ölü ve yaralı sayısı [2].

Ulaştırma Türü	Yolcu-km (Milyon)	Ölü ve Yaralı Sayısı	Milyar Yolcu-Km ye Düşen Ölü ve Yaralı Sayısı	Oran
Demiryolları	5277	455	86	1
Karayolları	182,152	127,195	698	8,1

Demiryolu taşımacılığında 10 Milyon Taşıt-Km ye düşen kaza oranı karayollarına göre yaklaşık 39 kat daha azdır. Aynı zamanda ölü ve yaralı sayısı bakımından da yaklaşık 8 kat daha güvenlidir [3].

5.1.5. Arazi kullanımı ve yol yapım maliyetleri

- AB’ de yapılan çalışmalar; aynı işi yapmak için gereken platform genişliğinin demiryolunda otoyola göre % 64 daha dar olduğunu göstermektedir.
- Saatte tek yönde 60.000 yolcu taşımak için,

- ✓ On iki şeritli bir otoyol gerekirken,
- ✓ Aynı miktardaki yolcuyla çift hatlı bir demiryolu ile taşımak mümkündür.

Yukarıda belirtilen talebi karşılayacak otoyolun km maliyeti yaklaşık 24 milyon \$ iken, çift hatlı, elektrikli ve sinyalli demiryolunun maliyeti sadece 4 milyon \$' dır. (Ayrıca, demiryolunun teknik ömrü 30 yıl karayolunun ise 10 yıldır.) Yapım maliyeti açısından demiryolunun daha avantajlı olduğu görülmektedir.

5.1.6. Hava kirliliği

Motorlu taşıtların çıkardığı egsoz gazı ortama kurşun ve diğer zehirli maddeler bırakır. Karayollarında yaygın olarak benzin kullanılırken, demiryollarında dizel yakıt veya elektrik kullanılmaktadır. Dizel yakıtın meydana getirdiği kirlilik benzine göre oldukça düşüktür. Demiryollarının hava kirliliğindeki payı dizelli çekim nedeniyle % 5 iken, karayollarının payı % 85 düzeyindedir. Elektrikli çekim gerçekleştirildiğinde ise hava kirliliğinden söz etmek mümkün değildir. Bir dizelli çekim yapan tren 42 km seyir sonucunda çevreye 1 kg karbondioksit yayarken, aynı miktarda karbondioksit otobüsle 12 km'de, otomobil ve uçakla 7 km'de yayılmaktadır [102].

5.1.7. Gürültü

İnsanlar demiryolunu çok daha gürültünü sanmaktadırlar. Ancak karayolundaki gürültü 80-90 dB arasındadır. Bu da otobanda, kamyon geçtiği zaman 105 dB e kadar çıkabilmektedir. Buna mukabil saatte 150 km hızla giden bir modern trenin gürültüsü 65 dB dir.

Trafiğin artması karayollarımızı devamlı çoğalan bir gürültü kaynağı haline getirmektedir. Trafik karayolunda gün boyu devam eder, demiryollarında ise

tren seyirleri arasında daha uzun bir aralık ve sessizlik vardır. Ulaştırma sistemlerinde konforlu bir seyahat için gürültü seviyesinin üst düzeyi 65 dB, rahatsızlık bölgesi 75-120 dB olarak kabul edilmektedir. Araştırmalarda karayollarındaki gürültü şiddetinin 72-92 dB arasında değiştiği tespit edilmiştir. Ağır taşıtlar için bu değer 103 dB'e kadar çıkmaktadır. Buna karşılık saatte 150 km hızla giden bir trenin gürültüsü 65-75 dB arasında değişmektedir [103].

5.2. Demiryolu – Karayolu ile Taşıma Sistemlerinin Karşılaştırılması

Güvenlik açısından

- Demiryollarının raya bağlı olması ve iklim koşullarından (kar, don, sis, yağmur vb.) karayoluna göre daha az etkilenmesi, güvenliği, konfor ve rahatlığı artırmaktadır.
- Avrupa'da yapılan bir araştırmaya göre; ulaştırma sistemlerinde ölüm riski 1 milyar yolcu-km başına demiryollarında 17 iken, karayollarında 140; yaralanma riski demiryollarında 41 iken, karayollarında 8 500–10 000'dir.
- Ülkemizde milyar yolcu-km başına düşen kaza sayısı, 1996 yılı itibariyle karayollarında 2 200 iken, demiryollarında 100'dür. Yani karayollarında demiryollarına göre 22 kat daha fazla kaza meydana gelmektedir. Meydana gelen kazalardaki ölü sayısı milyar yolcu-km. başına karayollarında 40 iken, demiryollarında Kuruluşun hatalarından kaynaklanan kazalarda 5'dir. Karayollarında yaralı sayısı milyar yolcu-km başına 675 iken, demiryollarında 38'dir.
- Ülkemizde karayolu kazalarında yılda ortalama 6 000 kişi ölmekte, 100 bin kişi yaralanmaktadır. Demiryollarında ise, 1996 yılında Kuruluşun kaynaklanan 539 adet kazada 13 kişi ölmüş, 160 kişi yaralanmıştır.

Cevre kirliliği açısından

- Araçların hareketinin sağlanması için kullandıkları yakıtlardan çıkan gazlar ve sektörlerdeki sanayi atıkları çevreyi kirletmektedir. Demiryollarının hava kirliliğindeki payı % 5 iken karayollarının payı % 85 düzeyindedir.
- Demiryollarının arazi ve suların kirlenmesinde de payı azdır. Buna karşın, karayolu araçlarından çıkan yağlar, benzin istasyonlarındaki sıvı karbüranlardan oluşan maddeler çevredeki arazi ve sulara zarar vermektedir.
- Avrupa Birliği çerçevesinde yapılan bir araştırmaya göre, ulaşım sistemlerinde meydana gelen trafik sıkışıklığı ve kazalarla neden olunan hava kirliliği ve gürültünün toplumsal maliyeti, Birlik hâsıllarının %4,1'i düzeyinde olup, bu maliyetin %90'ı karayolu ulaşımından kaynaklanmaktadır. Bu nedenle, Avrupa Birliği demiryolunda bireysel maliyetleri görece olarak azaltacak bir sistem olan sosyal maliyetlerin içselleştirilmesi üzerinde yoğun bir biçimde çalışmaktadır. Ülkemizde de bu konuda çalışmalara derhal başlanılmalıdır.

Arazi kullanımı açısından

- Bir ulaşım yolunun planlanması sırasında değerli arazi ve doğal kaynaklara zarar verilmemesine, özellikle dikkat edilmesi gereklidir. Bu ayrıntılar üzerinde yeterince durulmaması, hem doğal çevre dengesini “ekolojik dengeyi” bozmakta hem de ekonomik kayba neden olmaktadır. Doğal kaynakların yanlış kullanımı, arazinin bölünmesi, değerli arazinin yok edilmesi ve diğer çevre kirlilikleri ile eko-sistem arasında birebir etkileşim söz konusudur.
- Ülkelerde ulaşım için ayrılan alan; yerleşim bölgeleri, endüstri alanları ve ormanların yanında oldukça düşük bir pay oluşturmaktadır. Ulaşım ağı oldukça iyi olan ülkelerde bile ulaşım için ayrılan alan %5 dolayındadır ve

bunun yarıdan oldukça azı demiryoluna düşmekte olup, bu oranın içinde şev (eğimli) ve çimler de bulunmaktadır.

- Aynı kapasitede taşımacılık için demiryolları, karayollarına göre daha az arazi gerektirmektedir. Platform genişliği 13,7 m olan çift hatlı, elektrikli bir demiryolu hattı kapasite açısından, 37,5 m genişliğinde 6 şeritli bir otobana eşdeğerdir. Buna göre karayolları 2,7 kat daha fazla arazi kullanımı gerektirmektedir.

Gürültü açısından

- Karayolu motorlu araçlarında gürültü, motor ve susturuculara bağlı olarak değişmektedir.
- Yapılan araştırmalarda karayollarındaki gürültü şiddetinin 72–92 dB arasında değiştiği tespit edilmiştir. Ağır taşıtlar için bu değer 103 dB kadar çıkmaktadır. Havayollarında ise gürültü şiddeti 103–106 dB’ dir. Buna karşın, saatte 150 km hızla giden bir trenin gürültüsü 65–75 dB arasındadır. Japonya, Fransa ve Rusya’da kabul edilebilir gürültü standardı 40 ile 70 dB arasında değişmektedir.
- İnsan sağlığı açısından 8 saatlik bir çalışma için gürültü sınırının en fazla 90 dB olduğu göz önüne alındığında demiryollarının önemi daha da artmaktadır.

Enerji tüketimi açısından

- Türkiye’de ulaştırma sektörünün kullandığı enerji, toplam enerjinin yaklaşık %20’sidir.
- Demiryolunda birim işe düşen enerji tüketimi karayoluna göre 1/ 4–7 oranında daha azdır. Karayolunda yaygın şekilde benzin kullanılırken, demiryolunda dizel yakıt veya elektrik kullanılmaktadır.

- Dizel yakıtın meydana getirdiği kirlilik benzine göre oldukça azdır. Elektrikli çekimde ise, işletim sırasında emisyon oluşmazken yalnızca santral emisyonları dikkate alınmaktadır.
- Demiryolları, gerek yük, gerekse yolcu taşımacılığında diğer sistemlere göre daha az enerji harcamaktadır. Değişik ülkelerde yapılan araştırmalar, özellikle yük taşımacılığında demiryollarının çok ekonomik olduğunu ortaya koymuştur. Almanya'da yapılan bir araştırmaya göre; yolcu taşımacılığında demiryolu 1 birim kabul edilirse, otoyolda tüketilen enerji 3 birim olmaktadır. Buna eşdeğer taşıma yapan havayolunda, enerji tüketimi 5,2 birimdir. Yük taşımacılığında ise, demiryolunu 1 birim kabul edersek, karayolu 3 birim olmaktadır.

Altyapı Maliyetleri Açısından

- Demiryolu yapım maliyeti, karayolu otoban yapım maliyetlerinden daha azdır.
- Ülkemizde, otoyol maliyetlerinin yüksekliğinde; kredi şartları, araziler ve muayyen süreli krediler için süre uzamasından doğan \$ bazında birim fiyat yükselmesi etkili olup, otoyolun ortalama maliyetinin 8 milyon \$/km. olduğu görülmektedir.
- Altyapı maliyetleri açısından; Almanya'da kabul edilen esasa göre platform genişliği 13,7 m olan çift hatlı, elektrikli bir demiryolu hattı, kapasite açısından 37,5 m genişliğinde altı şeritli bir otobana eşdeğerdir. Buna göre Demiryolu altyapı başlangıç maliyetleri; düz arazide karayolunun %54,5'i, orta engebeli arazide %73,5'i dir. Faydalı ömür başına düşen yatırım tutarına göre ise, demiryolu maliyetleri karayolu maliyetinin düz arazide %27,3' ü, orta engebeli arazide %36,8'i, engebeli arazide de %60'ıdır.

5.3. Problemin Tanımı

Taşımacılık için iller arası mesafelere bakıldığında demiryolu uzunluklarının karayollarından fazla olduğu tespit edilmiştir. Fakat maliyetler açısından karşılaştırıldığında demiryolu maliyetinin karayolu maliyetine göre daha düşük olduğu görülmektedir. Benzer şekilde güvenlik, çevre kirliliği, arazi kullanımı, gürültü, tüketilen enerji ve alt yapı maliyetleri gibi faktörler açısından da değerlendirildiğinde demiryolu taşımacılığının neden tercih edilmesi gerektiği anlaşılmaktadır. Ancak bütün avantajlarına rağmen sadece demiryolu taşımacılığının kullanılamamasının sebebi ülkemiz ulaştırma politikaları ve arazi şartlarından dolayı her şehre demiryolu ile ulaşımın sağlanamamasıdır. Bu nedenle demiryolu ve karayolu taşımacılığını entegre edilmesi uygun görülmüştür. Buradaki amaç en ucuz taşımacılığı gerçekleştirmek için, en kısa Türkiye turunun bulunmasıdır.

Türkiye turu için demiryolu ulaşan illere demiryolundan, demiryolu ulaşmayan illere ise karayolundan faydalanılacaktır. Gerçekleştirilecek böyle bir Türkiye turunun, en ucuz şekilde nasıl yapılabileceği GSP problemi olarak ve karınca kolonisi algoritması ile belirlenmiştir.

5.4. Problemin Verileri

Öncelikle 81 ilimize ait mesafeler tablosu oluşturulmuştur. Mesafeler oluşturulurken demiryolunun ulaştığı illere demiryolu mesafesi, demiryolu ulaşmayan illere ise karayollarının mesafesi dikkate alınmıştır. Tüm illerimizin birbirlerine olan demiryolu ve karayolu uzaklığı kilometre cinsinden tablolaştırılmıştır. Simetrik GSP problemi olduğundan bütün d_{ij} ve d_{ji} 'ler birbirlerine eşittir. 81 ilimize ait mesafeler tablosunun bir kısmı örnek olarak Çizelge 5.3'de gösterilmiştir. Çizelgenin tamamı Ek-1 de verilmiştir.

Çizelge 5.3. Örnek iller arası mesafe matrisi

İLLER	ADANA	ADIYAMAN	AFYON	AĞRI	AMASYA
ADANA		329	642	961	767
ADIYAMAN	329		902	648	636
AFYON	642	902		1308	1136
AĞRI	961	648	1308		734
AMASYA	767	636	1136	734	

Türkiye’de demiryolları Adana, Adapazarı, Afyon, Amasya, Ankara, Aydın, Balıkesir, Batman, Bilecik, Burdur, Çankırı, Denizli, Diyarbakır, Edirne, Elazığ, Erzincan, Erzurum, Eskişehir, Gaziantep, Isparta, İstanbul, İzmir, İzmit, Kahramanmaraş, Karaman, Kars, Kayseri, Kırıkkale, Kırklareli, Konya, Kütahya, Malatya, Manisa, Mardin, Mersin, Muş, Niğde, Samsun, Sivas, Uşak, Van ve Zonguldak illerine kadar ulaşabilmektedir (Harita 5.1). Karayolları ise bütün illere ulaşabilmektedir (Harita 5.2).

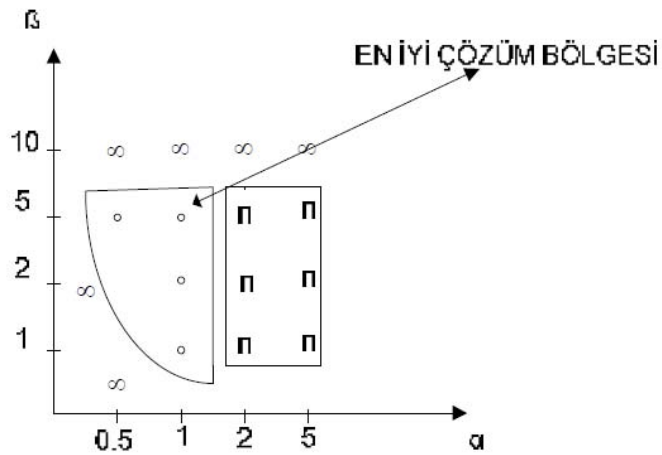


Harita 5.1. Türkiye demiryolları haritası



Harita 5.2. Türkiye karayolları haritası

Bölüm 4’de ifade edilen KKA geçiş kuralında kullanılan α ve β parametrelerinin literatüre göre aldığı çeşitli değerler karşısında çözümün nasıl etkilendiği Şekil 5.1’de gösterilmiştir [88]. Çalışmada α için 0,2; 0,4; 0,6 ve 0,8 değerleri; β için 2;3 ve 4 değerleri test edilmiştir.



Şekil 5.3. Parametre seçimi

5.5. Problemin Çözümü

Geliştirilen çözüm yaklaşımında amaç fonksiyonu her şehrin kullanılarak toplam katedilen mesafenin minimize edilmesidir. Programda çözüm aşamasını kolaylaştırmak için bir kullanıcı ara yüzü oluşturulmuştur. Kullanıcı arayüzü problemin çözüm aşamalarını ekrana yansıtarak adımların görülmesini sağlamıştır.

Mesafeler tablosu oluşturulduktan sonra şehirlerarası mesafeler Access veri tabanına girilmiştir. Problemin çözümü için tasarlanan karınca kolonisi algoritması Microsoft Visual Studio 2010 programında C # diliyle yazılmıştır. Problemin çözümünde yerel feromon güncellemesi kullanılmıştır. Geçiş kuralında hangi kuralı kullanacağımızı belirleyen q değeri küçük alınarak aramanın hızlandırılması sağlanmıştır. Karınca sayısı şehir sayısı ile eşit olarak girilmiştir. 81 adet karınca ile yazılan programa değişik parametreler girilmiş ve aşağıdaki sonuçlar ortaya çıkmıştır.

Kullanıcı ara yüzü (Şekil 5.4) ile, değişik parametreler girilerek çözümü kolaylaştırılmıştır. Arayüze yerleştirilen komutlar problem verisi olan α , β , buharlaşma katsayısı ve iterasyon sayısını test etme olanağı vermektedir. Ayrıca arayüzde her iterasyonda en kısa yolları ve tabu listesi görülebilmektedir. Yani her iterasyonla ilgi veriler takip edilebilmektedir.

Çizelge 5.4'de karınca sayısının şehir sayısına eşit olarak 81 adet alındığı ve buharlaşma katsayısının literatürde tanımlandığı gibi sabit 0,1 değerinde alındığı görülmektedir. α ve β değerleri ise değiştirilerek 12 adet test problemi türetilmiştir. Her bir test problemi 250 iterasyon denenmiş olup en iyi çözüm o test problemi sonucu olarak seçilmiştir. Toplam 3 000 iterasyon çalıştırılmış olan test problemleri yaklaşık 720 saat sürmüştür.

Şekil 5.4. Kullanılan Arayüz

Çizelge 5.4. Kullanılan parametrelere göre oluşan sonuçlar

Karınca sayısı	Buharlaşma katsayısı (0-1)	α (0-1)	β (0-5)	İterasyon sayısı	Katedilen mesafe
81	0,1	0,2	2	250	15 257
81	0,1	0,4	2	250	15 105
81	0,1	0,6	2	250	13 675
81	0,1	0,8	2	250	12 859
81	0,1	0,2	3	250	15 359
81	0,1	0,4	3	250	14 956
81	0,1	0,6	3	250	13 874
81	0,1	0,8	3	250	12 826
81	0,1	0,2	4	250	14 063
81	0,1	0,4	4	250	13 459
81	0,1	0,6	4	250	12 334
81	0,1	0,8	4	250	11 913

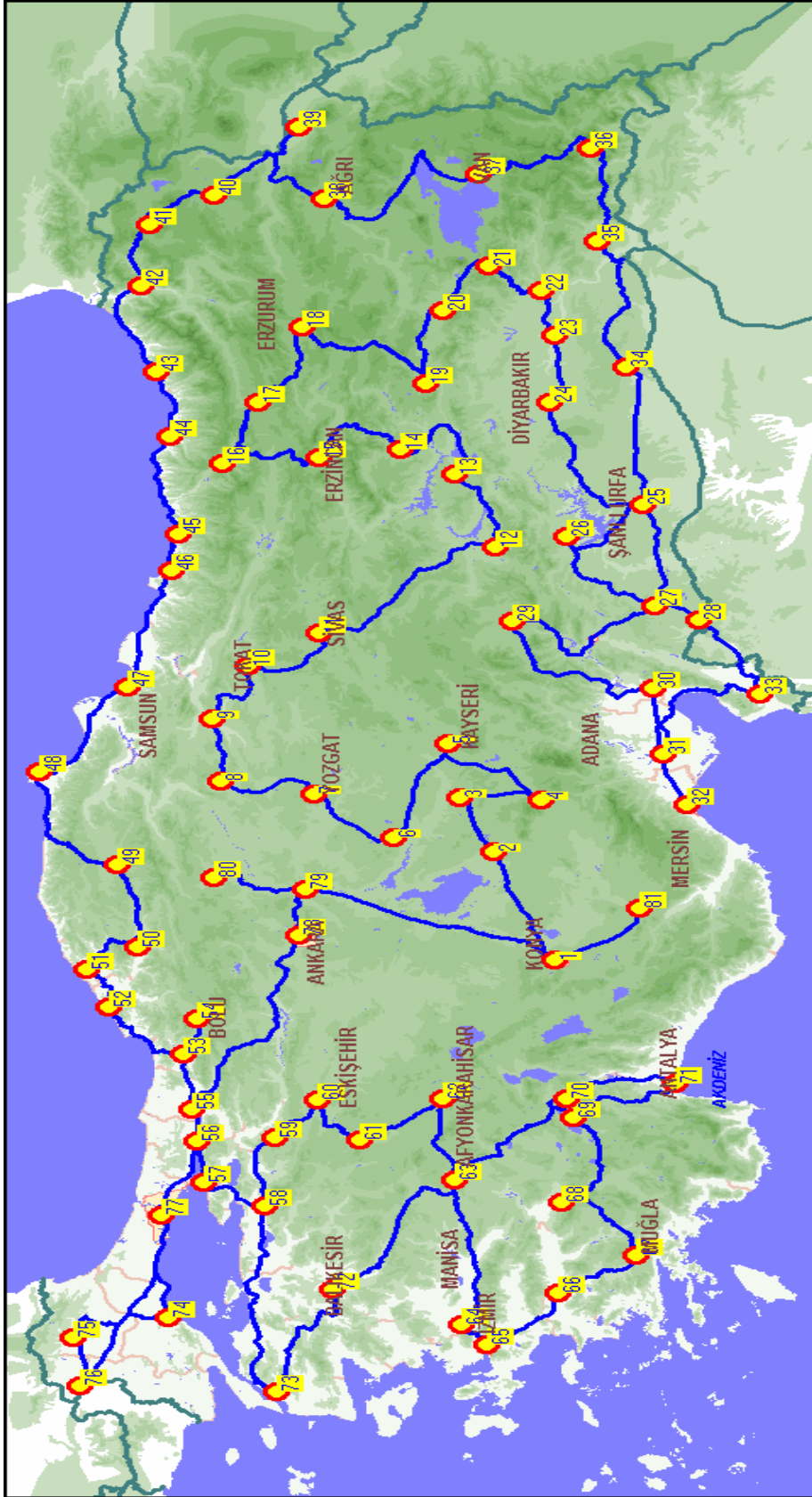
Çizelge 5.4 incelendiğinde, eniyi parametreler ve katedilen mesafenin Çizelge 5.5’ deki gibi olduğu tespit edilmiştir.

Çizelge 5.5. Problemden kullanılan parametreler ve elde edilen en iyi katedilen mesafe

Parametre	Değer
Karınca Sayısı	81
Buharlaştırma Katsayısı	0.1
α	0.80
β	4
İterasyon Sayısı	250
Katedilen Mesafe (km)	11 913

5.6. Problemin Sonucu

Program belirlenen iterasyon sayısı kadar çalıştırılmış ve en iyi sonuç 11 913 km değeri bulunmuştur. Bulunan bu en iyi sonuca ait çözüm değerleri Şekil 5.5’ de, detaylı olarak da Çizelge 5.6.’ da gösterilmiştir. Çizelge 5.6’ da koyu renkli yazılmış olana rotalar demiryolu ile katedilmiştir. Geliştirilen uygulama programının kaynak kodları Ek-2 de verilmiştir.



Şekil 5.5. Bulunan sonuca göre Türkiye turu

Çizelge 5.6. İllere göre çok modlu en kısa Türkiye turu

Çıkış Noktası	Varış Noktası	Çıkış Noktası	Varış Noktası	Çıkış Noktası	Varış Noktası
Karaman	Konya	Gaziantep	Kilis	Bolu	Sakarya
Konya	Aksaray	Kilis	Kahramanmaraş	Sakarya	Kocaeli
Aksaray	Nevşehir	Kahramanmaraş	Osmaniye	Kocaeli	Yalova
Nevşehir	Niğde	Osmaniye	Adana	Yalova	Bursa
Niğde	Kayseri	Adana	İçel	Bursa	Bilecik
Kayseri	Kırşehir	İçel	Hatay	Bilecik	Eskişehir
Kırşehir	Yozgat	Hatay	Mardin	Eskişehir	Kütahya
Yozgat	Çorum	Mardin	Şırnak	Kütahya	Afyon
Çorum	Amasya	Şırnak	Hakkari	Afyon	Uşak
Amasya	Tokat	Hakkari	Van	Uşak	Manisa
Tokat	Sivas	Van	Ağrı	Manisa	İzmir
Sivas	Malatya	Ağrı	Iğdır	İzmir	Aydın
Malatya	Elazığ	Iğdır	Kars	Aydın	Muğla
Elazığ	Tunceli	Kars	Ardahan	Muğla	Denizli
Tunceli	Erzincan	Ardahan	Artvin	Denizli	Burdur
Erzincan	Gümüşhane	Artvin	Rize	Burdur	Isparta
Gümüşhane	Bayburt	Rize	Trabzon	Isparta	Antalya
Bayburt	Erzurum	Trabzon	Giresun	Antalya	Balıkesir
Erzurum	Bingöl	Giresun	Ordu	Balıkesir	Çanakkale
Bingöl	Muş	Ordu	Samsun	Çanakkale	Tekirdağ
Muş	Bitlis	Samsun	Sinop	Tekirdağ	Kırklareli
Bitlis	Siirt	Sinop	Kastamonu	Kırklareli	Edirne
Siirt	Batman	Kastamonu	Karabük	Edirne	İstanbul
Batman	Diyarbakır	Karabük	Bartın	İstanbul	Ankara
Diyarbakır	Şanlıurfa	Bartın	Zonguldak	Ankara	Kırıkkale
Şanlıurfa	Adıyaman	Zonguldak	Düzce	Kırıkkale	Çankırı
Adıyaman	Gaziantep	Düzce	Bolu	Çankırı	Karaman

5.7. Sonucun Değerlendirilmesi

Çok modlu taşımacılıkta karayolları ve demiryolları bir arada kullanılarak elde edilen çözümde Türkiye’ de en kısa yol 11 913 km olarak bulunmuştur. 11 913 km’ nin 8 149 km’ si karayolu ile 3 764 km’ si demiryolu ile katedilmiştir.

Yük taşımacılığında birim nakliyat başına enerji tüketimi karayolları için 921 kcal/ton-km ve demiryolu için 61 kcal/ton-km olduğu durumda Türkiye turu için toplam enerji miktarı hesaplandığında;

- Demiryolunda harcanan enerji miktarı;
 $3764 * 61 = 229\ 604$ kcal/ ton
- Kararyolunda harcanan enerji miktarı;
 $8149 * 921 = 7\ 505\ 229$ kcal/ ton
- Toplam harcanan enerji miktarı;
 $229\ 604 + 7\ 505\ 229 = 7\ 734\ 833$ kcal/ton olarak bulunur.

Türkiye turunda hesaplanan 3 764 km’ lik demiryolunun 1 343 km’si elektrikli tren ile, 2 421 km’si dizel tren ile katedilmektedir. Elektrikli tren ile taşımacılıkta hava kirliliği söz konusu değildir. Dizelli tren ile taşımacılıkta ise çevreye her 42 km’de 1 kg karbondioksit; benzer şekilde karayolu taşımacılığında da her 12 km’de, 1 kg karbondioksit yayılmaktadır. Bu durumda Türkiye turu için hava kirliliği açısından çevreye yayılan toplam karbondioksit miktarı hesaplandığında:

- Elektrikli tren ile yayılan karbondioksit miktarı;
 $1\ 343 * 0 = 0$ kg
- Dizel tren ile yayılan karbondioksit miktarı;
 $2\ 421 / 42 = 57,64$ kg

- Karayolları yayılan karbondioksit miktarı;
 $8\,149 / 12 = 679,08 \text{ kg}$
- Toplam yayılan karbondioksit miktarı;
 $0 + 57,64 + 679,08 = 736,72 \text{ kg}$ olarak hesaplanmaktadır.

Türkiye’ de taşımacılıkta en kısa yolun bulunmasında benzer bir çalışma Söyler ve Keskintürk tarafından yapılmıştır. Söyler ve Keskintürk çalışmalarında karayolları taşımacılığında Türkiye’ de en kısa yolu KKA ile 9.954 km olarak hesaplamışlardır [95].

Söyler ve Keskintürk’ün çalışması ile tezde yapılan çalışma katedilen km açısından karşılaştırıldığında, 1 959 km farkla Söyler ve Keskintürk’ün Türkiye turunun daha kısa olduğu görülmektedir. Ancak bu çalışmada amaç sadece katedilen mesafe minimizasyonu değil; katedilen mesafeye bağlı olarak tüketilen enerji miktarı ve çevreye yaydığı karbondioksit miktarının minimizasyonudur. Bu nedenle Söyler ve Keskintürk’ün çalışmasında hesaplanan mesafe, tüketilen enerji miktarı ve çevreye yaydığı karbondioksit miktarı açısından düzenlenerek karşılaştırma yapılmıştır.

Söyler ve Keskintürk’ün çalışmasına enerji tüketimi açısından bakıldığında, toplam harcanan enerji miktarının:

$$9\,954 * 921 = 9\,167\,634 \text{ kcal/ton olduğu görülür.}$$

Söyler ve Keskintürk’ün çalışmasına hava kirliliği açısından bakıldığında, çevreye yayılan toplam karbondioksit miktarının:

$$9\,954 / 12 = 829,5 \text{ kg olduğu görülür.}$$

Çalışmaların karşılaştırması Çizelge 5.4’ de verilmiştir.

Çizelge 5.7. Çalışmaların karşılaştırması

	Tezde yapılan çalışma			Söyler ve Kesintürk'ün çalışması			Farklar
	Karayolu	Demir-yolu	Toplam	Karayolu	Demir-yolu	Toplam	
Katedilen mesafe (km)	8149	3764	11913	9954	-----	9954	1959
Harcanan enerji miktarı (kcal/ton)	7505229	229604	7734833	9167634	-----	9167634	-1432801
Çevreye yayılan karbondioksit (kg)	679,08	57,64	736,72	829,5	----	829,5	-92,78

Çizelge 5.7 incelendiğinde, katedilen mesafenin 1 959 km farkla Söyler ve Kesintürk'ün çalışmasında daha kısa olduğu; enerji tüketimi açısından karşılaştırıldığında 1 432 801 kcal/ton farkla tezde yapılan çalışmanın daha düşük maliyetli olduğu; çevreye yayılan karbondioksit açısından karşılaştırıldığında 92,78 kg farkla tezde yapılan çalışmanın çevreye daha az zarar verdiği görülmektedir.

6. SONUÇLAR VE ÖNERİLER

Artan uluslararası rekabet ortamında, taşımanın daha süratli ve ekonomik bir şekilde gerçekleştirilmesi bir zorunluluk haline gelmiştir. Günümüzde işletmeler dünya pazarlarına ulaşmak için taşımacılık hizmetlerinden faydalanmak zorundadırlar. İşletmelerin rekabet avantajlarını korumak için ürünlerini müşterilerin kapısına kadar taşıyacak ekonomik çözümlere ihtiyaç vardır.

Dünyada çevre duyarlılığının artması, devletleri bu konuda sıkı önlemler almaya itmektedir. Bu önlemlerden ulaştırma sistemleri de etkilenmektedir. Özellikle karayoluna getirilen kısıtlamalar, işletmeleri çok modlu taşımacılık alternatiflerine yöneltmektedir.

Farklı taşıma türlerinin entegrasyonu olan çok modlu taşımacılık, göndericiler ve alıcılar için özellikle kombine taşımacılık söz konusu olduğunda büyük avantaj sağlarken, çeşitli sıkıntıları da su yüzüne çıkarmaktadır. Türkiye’de bu alanda altyapı, yasal mevzuat ve uluslararası düzenlemeler olmak üzere atılması gereken adımlar bulunmaktadır. Ülkemizin bulunduğu coğrafyanın avantajlarından yararlanması için çok modlu taşımacılığı, bütün ulaştırma sistemleri düzenlenerek etkin bir şekilde kullanması gerekmektedir. Bu hedefe yönelik olarak, kara, hava, deniz ve demiryollarının bir arada uyum içerisinde çalışabileceği koşullar oluşturulmalı ve terminal ve limanlar elverişli hale getirilmelidir. Altyapı, uluslararası bağlantılar ve yasal çerçevedeki eksiklikler giderilmelidir.

Ulaştırma bilimcileri karayolunun ana ulaşım sistemi olmadığını, ulaşım ve taşımada havayolu, denizyolu ve demiryolunu destekleyici ve tamamlayıcı bir sistem olduğunu, toplu taşımacılığın ve milyonlarca tona ulaşan sıvı ve katı taşımacılığın karayolu ile yapılmasının hiçbir koşulda ülke menfaatleri ile bağdaşmadığını ve hiçbir bilimsel dayanağı olmadığını belirtmektedirler.

Gelenen noktada Türkiye'deki süreç bu şekilde karayolu ağırlıklı olarak devam edecek olursa bunca çevre kirliliği, ses kirliliği, sahil katliamı, trafik kazaları, binlerce can kayıplarının yanı sıra enerji açığımız ve otomotiv sanayimizle dışa bağımlılığımız daha da güçlenirken, en pahalı maliyetle dünyanın en pahalı taşımacılık sistemine kavuşacağız.

Demiryoluna baktığımızda özellikle son elli yılda ciddi hiçbir yatırım yapılmamıştır. Dünyanın en ucuz ve en çevreci taşımacılığı yani demiryolu taşımacılığı Türkiye'de verimli olarak yapılamamaktadır. Demiryollarının yıllardır ihmal edilmesi ulaştırma sektöründe Türkiye'nin rekabet gücünü olumsuz yönde etkileyen en önemli sebeplerden birisidir. Demiryolu ağ ve modernizasyonun yetersiz oluşu, yönetim sorunları, zamanında vagon temin edilememesi önemli sorunlar arasındadır.

Ülkemizde dünya demiryolu teknolojisindeki olumlu gelişmelere uyum sağlanamamış ve bütün Avrupa'da yaygın olarak oluşturulmaya çalışılan yüksek hız şebekesine entegre olacak somut adımlar atılamamıştır. Sonuçta, ABD'nin mali desteği ve politika önerileriyle birlikte karayolları artık egemen bir konuma gelmiş, demiryolları ise, fazla bir gelişme sağlanamamıştır.

Bu çalışmada, Türkiye'de yük taşımacılığında en kısa turun bulunmasında karayolu ve demiryolu entegrasyonu üzerine çalışılmıştır. Karayolları ve demiryollarının mevcut durumları değerlendirilerek, demiryolu ile ulaşımın olduğu yerlere demiryolu ile, demiryolunun olmadığı yerlere karayolu ile ulaşarak tüm ülkede en kısa tur bulunmuştur.

Ülkemizde, harcanan enerji miktarı üretilen enerji miktarından fazla olması nedeniyle dışa bağımlılık söz konusudur. Bu durum değerlendirildiğinde yük taşımacılığında amaç sadece katedilen mesafe enküçüklemesi değil, maliyetleride içermelidir. Ayrıca güvenlik, çevre kirliliği, arazi kullanımı,

gürültü, alt yapı maliyetleri ve ülke politikaları gibi etkenlerde amaç içinde değerlendirilebilir.

Çalışmada, Türkiye’ de karayolu ve demiryolu entegreli en kısa yol 11 913 km olarak bulunmuştur. 11 913 km yolun 3 764 km’si demiryolu ile, 8 149 km’si karayolu ile katedilmektedir. 11 913 km yol ile toplam 7 734 933 kcal / ton enerji harcanmaktadır. 11 913 km yolu katedilirken çevreye toplam 736,72 kg karbondioksit yayılmaktadır.

KKA kullanılarak sadece karayolu ulaşımı için en kısa yol Söyler ve Kesintürk tarafından 9.954 km olarak bulunmuştur. İlgili mesafeye göre 9 167 634 kcal/ton enerji tüketildiği ve çevreye 829,5 kg karbondioksit yayıldığı hesaplanmıştır.

Tüketilen enerji ve çevreye yayılan karbondioksit miktarı açısından değerlendirildiğinde çok modlu taşımacılığın daha avantajlı olduğu görülmektedir. Bunun nedeni demiryolu taşımacılığıdır.

Ayrıca güvenlik, gürültü ve arazi kullanımı açısından da karşılaştırıldığında demiryolu taşımacılığının daha avantajlı olduğu söylenebilir.

Yukardaki nedenlerden dolayı demiryolu taşımacılığının ülkemizde daha kullanılabilir hale getirmek için aşağıdaki çözümler önerilebilir:

- Demiryolları modlu taşımacılığının özendirilmesi,
- Çok modlu ve kombine taşımacılığa uygun yük istasyonlarının hazırlanması,
- Demiryolları ile ilgili yasal düzenlemeler yapılması,
- Vagon kapasiteleri ve sayıları ile lokomotif sayı ve hızlarının artırılması,
- Hızlı tren taşımacılığının yük ve yolcu taşımacılığındaki payının artırılması,

- Organize Sanayi Bölgelerinin demiryolu bağlantılarının oluşturulması, ilgili bölgelerde lojistik köyler oluşumunun hızlandırılması,
- Önemli sanayi ve ticaret merkezlerinden ihracat limanlarına demiryolu ile ulaşım sağlanması, ulaşımın var olduğu limanlarda ise demiryolu ile besleme oranlarının artırılması,
- Ulusal bir araştırma-geliştirme (AR-GE) merkezi kurulması,
- Lokomotif ve vagon üretiminin yurtdışında yaygınlaştırılması ve demiryollarında yan sanayi oluşturulması,
- Demiryolu ulaştırma sisteminin rekabetçi piyasa koşullarına uygun pazar işleyişi üzerine kurulması,
- Demiryolu işletmeciliğinin özel sektörde yaygınlaştırılması ve özendirilmesi konularında devletin yol gösterici, planlayıcı, düzenleyici ve denetleyici bir rol oynaması,
- Demiryolu ağının tüm yurdu kapsamı, Türkiye – Gürcistan (Kars – Tiflis) demiryolu projesi benzeri yatırımlar ile komşu ülkeler ile olan demiryolu bağlantılarının güçlendirilmesi,
- Demiryolu ulaştırması kapsamında düşünce kulüpleri oluşturulması,
- Personel verimliliğinin artırılması için gerekli eğitim programlarının düzenlenmesi,
- Personel sayısının en uygun düzeye getirilmesi.

KAYNAKLAR

1. Simchi-Levi, D., Bramel, J., Logic of Logistics: Theory, **Algorithms & Applications for Logistics Management**, Springer-Verlag, New York, 5-65 (1997).
2. “T.C. Devlet Demiryolları İstatistik Yıllığı”, **TCDD APK Dairesi Başkanlığı**, Ankara, 23-127 (2010).
3. “İstatistik Göstergeler”, **Türkiye İstatistik Kurumu**, Ankara, 367-384 (2010).
4. Çancı, M., Erdal, M., “Lojistik yönetimi: Freight forwarder el kitabı 1”, **UTİKAD – Uluslararası Taşımacılık ve Lojistik Hizmet Üretenleri Derneği Yayınları**, İstanbul, 1-302 (2003).
5. İnternet:T.C.D.D “Cumhuriyetimizin 80 Yıllık Tarihinde Demiryolu Politikaları” <http://www.tcdd.gov.tr/home/detail/?id=267> (2011).
6. Şerbetçi, E., “Ulaştırma Sektörleri Arasındaki Öncelik Sorunu ve Etkileri”, Yüksek Lisans Tezi, **Gazi Üniversitesi Fen Bilimleri Enstitüsü**, Ankara, 22- 44 (1998).
7. İnternet: Denizcilik Müsteşarlığı “Tarihçe ve Hukuki Statü” <http://www.denizcilik.gov.tr/dm/mustesarlik/tarihce.aspx> (2010)
8. Akten, N., “Taşımacılıkta Çağdaş Motif: Kapıdan Kapıya Taşıma”, **Logistical**, 3, 16-20 (2005).
9. Koparal, M. , “Türk Deniz Taşımacılığı”, **Logistical**, 2, 59-62 (2004).
10. Sivil Havacılık Genel Müdürlüğü “Havaalanlarımız” <http://web.shgm.gov.tr/kurumsal.php?page=havaalanlarimiz> (2010)
11. Baki, B., “Lojistik Yönetimi ve Lojistik Sektör Analizi”, **Volkan**, Ankara, 51-60 (2004).
12. Karagülle, A. Ö., “Ulaştırma Sektörünün Geleceği Yerel Havacılıkta”, **Logistical**, 2, 35 (2004).
13. İnternet: UND, “Uluslar arası taşımacılıkta karşılaşılan sorunlar”, <http://www.und.org.tr/index.php?go=ara.php>, (26 Nisan 2007).
14. Alkan, M., Erdal, M., “Lojistik ve Dış Ticaret Sözlüğü”, **UTİKAD (Uluslararası Taşımacılık ve Lojistik Hizmet Üretenler Derneği)**, İstanbul, 400, 270, 84 (2004).

15. Tırman, M., "Taşımacılık Sektöründe Kombine Taşımacılığın Dünü, Bugünü ve Yarınına Bir Bakış", **İkinci Ulusal Demiryolu Kongresi**, Ankara, 263-270 (1997).
16. Evren, G., Tekin, İ., "Türkiye'de Uluslararası kombine Taşımacılığın Avrupa ile Bütünleşme Bağlamında Değerlendirilmesi", **İkinci Ulusal Demiryolu Kongresi**, Ankara, 219-231 (1997).
17. Özkan, B., Cevre, U., Ugur, A., "Melez Bir Eniyileme Yöntemi ile Rota Planlama", **Akademik Bilişim**, Çanakkale, 120-131 (2008).
18. Terzi, Ü., "Gezgin Satıcı Problemi için Diferansiyel Gelişim Algoritması Tabanlı Bir Metasezgisel Önerisi", Doktora Tezi, **Kocaeli Üniversitesi Fen Bilimleri Enstitüsü**, 4-50 (2009).
19. Gönülol, S., "Gezgin Satıcı Problemi için Veri Madenciliği Tabanlı Sezgisel Bir Yaklaşım", Yüksek Lisans Tezi, **Kocaeli Üniversitesi Fen Bilimleri Enstitüsü**, 22-36 (2009).
20. Golden, B., Bodin, L., Doyle, T., Stewart, W., "Approximate traveling salesman algorithms", **Operations Research**, 694-712 (1980).
21. Minieka, E., "Optimization algorithms for network and graphs", **Marcel Dekker Inc.**, New York, 200-350 (1978).
22. Gondran, M., Minoux, M., Vajda, S., "Graphs and Algorithms", **John Wiley & Sons**, New York, 501-623 (1984).
23. BAYZAN, Şahin., "Araç Rotalarının En kısa Yol Algoritmaları Kullanılarak Belirlenmesi ve .Net Ortamında Simülasyonu", Yüksek Lisans Tezi, **Pamukkale Üniversitesi Fen Bilimleri Enstitüsü**, Denizli, 17-32 (2005).
24. Gondran, M., Minoux, M., "Graphs and Algorithms", **John Wiley and Sons Inc.**, New York, 350-605, (1984).
25. Sipahioglu, A., "Gezgin Satıcı ve Araç Turu Belirleme Problemleri İçin Yeni Alt Tur Engelleme Kısıtları", **Osmangazi Üniversitesi, Fen Bilimleri Enstitüsü**, Doktora Tezi, 1-65 (1996).
26. Lawler, E.L., Lenstra, J.K., Rinnooy, Kan, A.H.G., Shmoys, D.B., "The Traveling Salesman Problem", **John Wiley & Sons**, New York, 105-305 (1985).
27. Eiselt, H.A., Sandblom, C.L., "Integer Programming and Network Models", **Springer-Verlag**, Berlin, 315-341 (2000).

28. Bauk, Sanja, Nataša Kova, "Modeling Ship's Route By The Adaptation of Hopfield - Tank T.S.P Neural Algorithm", ***Journal of Maritime Research***, 1(3):51-66 (2004).
29. Blaser, M., Manthey, B., Sgall, J., "An Improved Approximation Algorithm for The Asymmetric TSP with Strengthened Triangle Inequality", ***Journal of Discrete Algorithms***, 4: 623-632 (2006).
30. Kwon, S.H., Kim, H.T., Kang, M.K., "Determination of the candidate arc set for the asymmetric traveling salesman problem", ***Computers & Operations Research***, 32(5): 1045-1057 (2005).
31. Pearn, W.L., Chien, R.C., "Improved solutions for the traveling purchaser problem", ***Computers & Operations Research***, 25(11): 879-885 (1998).
32. Bektaş, T., "The multiple traveling salesman problem: an overview of formulations and solution procedures" ***The International Journal of Management Science***, 34: 209-219 (2006)
33. "Chinese postman problem", Algorithms and Theory of Computation Handbook, ***CRC Press LLC***, 10-65 (1999).
34. Gutin, G., Punnen, A.P., "The Traveling Salesman Problem and Its Variations", ***Kluwer Academic Publishers***, 29-114 (2002).
35. Korte, B., Vygen, J., "Combinatorial Optimization Theory and Algorithms", Third Edition, ***Springer***, 1-6 (2006).
36. Held, M., Karp, R.M., "A dynamic programming approach to sequencing problems", ***Journal of SIAM***, 10: 196-210 (1962).
37. Laporte, G., "The Vehicle Routing Problem: An overview of exact and approximate algorithms", ***European Journal of Operational Research***, 59: 345-358 (1992)
38. Lawler, E., Lenstra, J., Rinnooy, A., Symoys, D., "The Traveling Salesman Problem", ***John Wiley & Sons Ltd***, 25-68 (1985).
39. Dantzig, G., Fulkerson, R., Johnson, S., "Solution of a Large-Scale Travelling Salesman Problem", ***Journal of Operations Research Society***, 2: 393-410 (1954).
40. Carpaneto, G., Toth, P., "Some New Branching and Bounding Criteria for the Asymmetric Traveling Salesman Problem", ***Management Science***, 26: 736-743 (1980).

41. Miller, D. L., Penky, J. F., "Exact Solution of Large Asymmetric Traveling Salesman Problems", **Science**, 251: 754-761 (1991).
42. Carpaneto, G., Dell'amico, M., Toth, P., "Exact Solution of Large Scale, Asymmetric Traveling Salesman Problems", **ACM Transaction on Mathematical Software**, 21: 394-409 (1995).
43. Held, M., Karp, R. M., "The Traveling Salesman Problem and Minimum Spanning Trees", **Operations Research**, 18: 1138-1162 (1970).
44. Helbig, H. K., Krarup, J., "Improvements of the Held-Karp Algorithm for the Symmetric Traveling Salesman Problem", **Mathematical Programming**, 7: 87-96 (1974).
45. Malik, K., Fisher, M. L., "A dual Ascent Algorithm for the 1-tree Relaxation of the Symmetric Traveling Salesman Problem", **Operations Research Letters**, 9: 1-7 (1990).
46. Grötschel, M., Pulleyblank, W.R., "Clique Tree Inequalities and the Symmetric Traveling Salesman Problem", **Mathematics of Operations Research**, 11: 537-569 (1986).
47. Grötschel, M., Padberg, M., "On the Symmetric Traveling Salesman Problem: Inequalities, Lifting Theorems and Facets", **Mathematical Programming**, 16: 265-302 (1979).
48. Crowder, H., Padberg, M., "Solving Large Scale Symmetric Traveling Salesman Problems to Optimality", **Management Science**, 26: 495- 509 (1980).
49. Padberg, M., Rinaldi, G., "Optimization of a 532-city Symmetric Traveling Salesman Problem by Branch and Cut", **Operations Research Letters**, 6: 1-7 (1987).
50. Padberg, M., Rinaldi, G., "A Branch-and-cut Algorithm for the Resolution of Large-Scale Symmetric Traveling Salesman Problems", **SIAM Review**, 33: 60-100 (1991).
51. Jünger, M., Reinelt, G., Rinaldi, G. (1995), "Handbook In Operations Research and Management Science", **Elsevier Science**, 7: 225-330 (1995).
52. Lin, S., Kernighan, B. W., "An Effective Heuristic Algorithm for the Traveling Salesman Problem", **Operation Research**, 21: 498-516 (1973).

53. Demircioğlu, M., "Araç Rotalama Probleminin Sezgisel Bir Yaklaşım ile Çözümlemesi Üzerine Bir Uygulama", Doktora Tezi, **Çukurova Üniversitesi Sosyal Bilimler Enstitüsü**, 22-50 (2009).
54. Foulds, L.R., "Combinatorial Optimization for Undergraduates", **Springer Verlag**, 174-177 (1984).
55. DePuy, G.W., Moraga, R. J., Whitehouse, G. E., "Meta-RaPS: a simple and effective approach for solving the traveling salesman problem", **Transportation Research Part E: Logistics and Transportation Review**, 41-2, 115-130 (2005).
56. Engebretsen, L., Karpinski, M., "TSP with bounded metrics", **Journal of Computer and System Sciences**, 72: 509–546 (2006).
57. Lin, S., Kernighan, B., "An effective heuristic algorithm for the traveling salesman problem", **Operations Research**, 21: 498-516 (1973).
58. Helsgaun, K., "An effective implementation of the Lin-Kernighan traveling salesman heuristic", **European Journal of Operational Research**, 126:106-130 (2000).
59. Cowling, P.I., Keuthen, R., "Embedded local search approaches for routing optimization", **Computers & Operations Research**, 32: 465–490 (2005).
60. Gamboa, D., Rego, C., Glover, F., "Implementation analysis of efficient heuristic algorithms for the traveling salesman problem", **Computers & Operations Research**, 33: 1154-1172 (2006).
61. Nabyev, V. V., "Yapay Zeka: Problemler, Yöntemler, Algoritma", **Seçkin Yayınevi**, Ankara, 50-90 (2003).
62. Blum, C., Roli, A., "Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison", **ACM Computing Surveys**, 35 (3): 268-308 (2003).
63. Tarantilis, C. D., Kiranoudis, C. T., Vassiliadis, V., S., "A Threshold Accepting Metaheuristic For The Heterogeneous Fixed Fleet Vehicle Routing Problem", **European Journal of Operation Research**, 152: 148-158 (2004).
64. Cordeau, J. F., Gendreau, M., Laporte, G., Potvin, J.Y., Semet, F., "A Guide To Vehicle Routing Heuristics", **Journal of the Operational Research Society**, 53: 512-522 (2002).

65. Kirkpatrick, S., Gelatt, Jr., C. D., Vecchi, M. P, "Optimization by simulated annealing", **Science**, 220: 671-680 (1983).
66. Güden, H., Vakvak, B., Özkan, B. E., Altıparmak, F., Dengiz, B., "Genel Amaçlı Arama Algoritmaları ile Benzetim Eniyilemesi: En İyi Kanban Sayısının Bulunması", **Endüstri Mühendisliği Dergisi**, 16(1): 2-15 (2005).
67. Gendreau, M., Laporte, G., Potvin, J.Y., "Metaheuristics for the Vehicle Routing Problem", **Les Cahiers du GERAD**, Canada, 52-98 (1999).
68. Bräysy, O., "Genetic Algorithms for the Vehicle Routing Problem with Time Windows", **Dept. of Mathematics and Statistics, University of Vaasa**, Finland, 55-78 (2001).
69. Keskinürk, T., "Araç Rotalama Problemlerinin Global Karınca Koloni Optimizasyonu ile Çözümü", Doktora Tezi, **İstanbul Üniversitesi Sosyal Bilimler Enstitüsü**, İstanbul, 1-120 (2009).
70. Beckers, R., Deneubourg, J.L., Goss, S., "Trails and U-turns in the selection of the shortest path by the ant *Lasius niger*", **Journal of Theoretical Biology**, 159: 397-415 (1992).
71. Goss, S., Aron, S., Deneubourg, J.L., Pasteels, J.M., "Self-organized shortcuts in the Argentine ant", **Naturwissenschaften**, 76: 579-581 (1989).
72. Hölldobler, B., Wilson, E.O., "The ants", **Berlin: Springer-Verlag**, 25-45 (1990).
73. Sarıkoç, F., "Paralel Karınca Kolonisi Optimizasyon Algoritması ve Test Problemlerindeki Performansının İncelenmesi ", Yüksek Lisans Tezi, **Erciyes Üniversitesi Fen Bilimleri Enstitüsü** , Kayseri, 1-99 (2004).
74. Taha, H.A., "Yöneylem Araştırması", **Literatür Yayıncılık**, İstanbul, 1-250 (2000).
75. Timor, M., "Yöneylem Araştırması ve İşletmecilik Uygulamaları", **İstanbul Üniversitesi İşletme Fakültesi Yayını**, İstanbul, 1-25 (2001).
76. Potvin, J.Y., "The Traveling Salesman Problem: A Neural Network Perspective", **INFORMS Journal on Computing**, 5: 328-348 (1993).
77. Potvin, J.Y. "Genetic Algorithms for the Traveling Salesman Problem", **Annals of Operations Research**, 63: 339-370 (1996).

78. Aarts, E.H.L., Korst, J.H.M., Laarhoven, "P.J.M. A Quantitative Analysis of the Simulated Annealing Algorithm: A Case Study for the Traveling Salesman Problem", **J. Stats. Phys**, 50: 189-206 (1988).
79. Dorigo, M., Gambardella, L.M., "Ant Colony System: A cooperative learning approach to the traveling salesman problem", **IEEE Transactions on Evolutionary Computation**, 53–66 (1997).
80. Stützle, T., Hoos, H.H., "MAX–MIN Ant System", **Future Generation Computer Systems**, 16(8): 889–914 (2000).
81. Dorigo, M., Gambardella, L.M., "Ant colonies for the traveling salesman problem", **BioSystems**, 43(2): 73–81 (1997).
82. Alaykiran, K., Engin, O., "Karıncı Kolonileri Metasezgiseli ve Gezgin Satıcı Problemleri Üzerinde Bir Uygulaması", **Gazi Üniv. Müh. Mim. Fak. Der.**, 20(1): 69-76 (2005).
83. Dorigo, M., Maniezzo, V., Coloni, A., "Positive Feedback as a Search Strategy", **Teknik Rapor**, İtalya, 26-45 (1991).
84. Coloni, A., Dorigo, M., Maniezzo, V., "Distributed Optimization by Ant Colonies", Proceedings of ECAL'91, **European Conference on Artificial Life**, Elsevier Publishing, Amsterdam, 36-87 (1991).
85. Dorigo, M., "Ottimizzazione, apprendimento automatico, ed algoritmi basati su metafora naturale (Optimization, Learning and Natural Algorithms)", Ph.D.Thesis, **Politecnico di Milano**, Italy, Italyanca 52-67(1992).
86. Dorigo, M., Stützle, T., "Ant Colony Optimization", **MIT Press**, USA, 65-72 (2004).
87. Dorigo, M., Gambardella, L.M., "Solving symmetric and asymmetric TSPs by ant colonies", in **Proceedings of the 1996 IEEE International Conference on Evolutionary Computation (ICEC'96)**, 23-59 (1996).
88. Dorigo, M., Maniezzo, V., Coloni, A., "Ant System: Optimization by a colony of cooperating agents", **IEEE Transactions on Systems, Man, and Cybernetics – Part B**, 26(1): 29–41 (1996).
89. Gambardella, L.M., Dorigo, M., "Ant-Q: A Reinforcement Learning Approach to the Traveling Salesman Problem", **Twelfth International Conference on Machine Learning**, A. Prieditis and S. Russell (Eds.), Morgan Kaufmann, 252-260 (1995).

90. Dorigo, M., Di Caro, G., Gambardella, L. M., "Ant Algorithms for Discrete Optimization", **Artificial Life**, 5(2): 137-172 (1999).
91. Keskindürk, T., Söyler, H., "Global Karınca Kolonisi Optimizasyonu", **Gazi Üniv. Müh. Mim. Fak. Der.**, 21(4): 689-698 (2006).
92. Gambardella, L.M., Dorigo, M., "Ant Colony System hybridized with a new local search for the sequential ordering problem", **INFORMS Journal on Computing**, 12(3): 237-255 (2000).
93. Maniezzo, V., Gambardella, L. M., Luigi, F., "European Commission Project BISON", (**IST-2001-38923**), 89-102 (2003).
94. EVREN, G., "Demiryollarımız", **Türkiye Mühendislik Haberleri**, Ankara 384:16-29 (1996).
95. Keskindürk, T., Söyler, H., "Karınca Kolonisi Algoritması ile Gezen Satıcı Probleminin Çözümü", **8. Türkiye Ekonometri ve İstatistik Kongresi.**, Malatya, 1-11 (2007).
96. Laporte, G., Gendreau, M., Potvin, J. Y., Semet, F., "Classical And Modern Heuristics For The Vehicle Routing Problem", **International Transactions in Operation Research**, 7: 285-300 (2000).
97. Karaboğa, D., "Yapay Zeka Optimizasyon Algoritmaları", **Atlas Yayın Dağıtım**, Çağaloğlu, İstanbul, 1-85 (2004).
98. Marinakis, Y., "Heuristic Solutions of Vehicle Routing Problems in Supply Chain Management", **DSS Laboratory, Department of Production Engineering and Management, Technical University of Crete**, Greece, 1-75 (2001).
99. Van Breedam, A., "Comparing Descent Heuristics And Metaheuristics For The Vehicle Routing Problem", **Computers & Operations Research**, 28: 289-315 (2001).
100. Bräysy, O., Gendreau, M., "Metaheuristics for the Vehicle Routing Problem with Time Windows", **SINTEF Report, SINTEF Applied Mathematics, Research Council of Norway**, Norway, 150-165 (2001).
101. Yılmaz, Ş., "Çok Depolu Araç Rotalama Probleminin Karınca Kolonisi Optimizasyonu ile Modellenmesi ile Modellenmesi ve Bir Çözüm Önerisi", Yüksek Lisans Tezi, **Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü**, İstanbul, 1-120 (2008)
102. Çakar, A. E., "Ankara Kentiçi Ulaşımının Dünü Bugünü Yarını", Gazi Üniversitesi, Ankara, (1997).

103. Gökdağ, M., Üçüncü, O., “Trafik ve Gürültü Tabiat ve İnsan”, Ankara, (1992).

EKLER

EK.1.İllerarası Mesafe

[illegible]

Ek-2. Uygulamaya ait kaynak kodları

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace KKA
{
    public partial class Form1 : Form
    {
        double alfa = 0.8;
        double beta = 4;
        double buharlasma = .1;
        int tursayisi = 10;

        DbClass db = new DbClass();

        public Form1()
        {
            InitializeComponent();

            List<string>[] tabu ;
            List<string> sehir = new List<string>();
            List<string> yedeksehir = new List<string>();

            //bir turdaki her bir karıncanın yol uzunluklarını
            saklar int[] tabuyollar;

            //turlar sonucunda en son en kısa güzergahı tutar
            List<string> enkisayol = new List<string>();

            int MinYolUzunluk = 0;

            private void Form1_Load(object sender, EventArgs e)
            {
                txtAlfa.Text = alfa.ToString();
                txtBeta.Text = beta.ToString();
                txtBuharlasma.Text = buharlasma.ToString();
                txtIterasyon.Text = tursayisi.ToString();

                yedeksehir = SehirDoldur();

                tabu = new List<string>[yedeksehir.Count];
            }
        }
    }
}

```

Ek-2.(Devam) Uygulamaya ait kaynak kodları

```

        tabuyollar = new int[yedeksehir.Count]; ;//tabu
listesindeki şehirler arası mesafe

    }

    void EnKisaYol(int kacincitur)
    {
        //her turda karınca tabularının yol uzunlukları
        bulunuyor ve en küçük olan yol ve tabu listesi
        //bulunuyor
        for (int k = 0; k < yedeksehir.Count; k++)
        {
            List<string> kguzeergah = tabu[k];
            int toplamyol = 0;

            string oncekisehir = kguzeergah[0];
            string sonrakisehir = "";
            for (int sıra = 1; sıra < kguzeergah.Count;
sıra++)
            {
                sonrakisehir = kguzeergah[sıra];
                string sql = "SELECT YOL FROM YOL WHERE
(CIKIS_SEHIR='" + oncekisehir + "' AND VARIS_SEHIR='" +
sonrakisehir + "') OR (CIKIS_SEHIR='" + sonrakisehir + "' AND
VARIS_SEHIR='" + oncekisehir + "')";

                object sonuc = db.ExecuteScalar(sql);

                if (sonuc != null)
                {
                    toplamyol = toplamyol +
Convert.ToInt32(sonuc);
                }
                oncekisehir = sonrakisehir;
            }

            tabuyollar[k] = toplamyol;
        }

        int minyol = tabuyollar[0];
        int minsıra = 0;
        for (int i = 1; i < tabuyollar.Length; i++)
        {
            if (tabuyollar[i] < minyol)
            {
                minyol = tabuyollar[i];
                minsıra = i;
            }
        }
    }

```

Ek-2.(Devam) Uygulamaya ait kaynak kodları

```

        if (MinYolUzunluk == 0)
        {
            enkisayol = tabu[minsira].ToList<string>();

            MinYolUzunluk = minyol;
            lblTur.Text = kacincitur.ToString();
        }
        else
        {
            if (minyol < MinYolUzunluk)
            {
                enkisayol =
tabu[minsira].ToList<string>();
                MinYolUzunluk = minyol;
                lblTur.Text = kacincitur.ToString();
            }
        }

        lblMinYol.Text = MinYolUzunluk.ToString();
    }

    private void button1_Click(object sender, EventArgs e)
    {
        grpDegisken.Enabled = false;
        button1.Enabled = false;
        try
        {
            alfa = Convert.ToDouble(txtAlfa.Text);
            beta = Convert.ToDouble(txtBeta.Text);
            buharlasma =
Convert.ToDouble(txtBuharlasma.Text);
            tursayisi =
Convert.ToInt32(txtIterasyon.Text);
        }
        catch
        {
            MessageBox.Show("Veri girişleri uygun değil.
Yeniden giriniz!");
            grpDegisken.Enabled = true;
            button1.Enabled = true;
            return;
        }

        //Program her çalıştırıldığında feromen
miktarları 1 olarak değiştiriliyor
        db.ExecuteNonQuery("UPDATE YOL SET FEROMEN=1");
        //dataGridView1.DataSource = db.Fill2("SELECT *
FROM YOL", "YOL");
        listBox1.Items.Clear();
        listBox2.Items.Clear();
    }

```

Ek-2.(Devam) Uygulamaya ait kaynak kodları

```

lblMinYol.Text = "";
lblTur.Text = "";
for (int tur = 0; tur < tursayisi; tur++)
{

    lblIterasyon.Text = tur.ToString() + ". tur";

    Refresh();//ekranı tazeliyor

    //her turda şehir listesi dolduruluyor
    sehir = yedeksehir.ToList<string>();

    //tabu listesi sıfırlanıyor
    for (int i = 0; i < tabu.Length; i++)
    {
        tabu[i] = new List<string>();
    }

    for (int k = 0; k < yedeksehir.Count; k++)
    {

        string ilk_sehir = sehir[0];
        tabu[k].Add(ilk_sehir);
        //tabu listesine eklenen ilk şehir,
        ulaşılacak şehir listesinden çıkartılıyor
        sehir.Remove(ilk_sehir);

        while (sehir.Count != 0)
        {
            //tabu listesindeki son şehre en
            yakın şehir bulunuyor.
            string sonraki_sehir =
            MinYolBul(tabu[k][tabu[k].Count - 1], k);
            //bulunan şehri tabu listesine ekliyor
            tabu[k].Add(sonraki_sehir);
            //min şehir şehir listesinden
            çıkarılıyor

            sehir.Remove(sonraki_sehir);

        }

        //ilk şehre geri dönülüyor
        tabu[k].Add(ilk_sehir);

        //ilk şehir yedek şehirden siliniyor
        yedeksehir.Remove(ilk_sehir);
        yedeksehir.Add(ilk_sehir);
    }
}

```

Ek-2.(Devam) Uygulamaya ait kaynak kodları

```

//şehirler yedek listeden tekrar
dolduruluyor        sehir = yedeksehir.ToList<string>();

                    }
                    //her turdaki en kısa yol bulunuyor
                    EnKisaYol(tur);

                    //en kısayolun güzergahı gösteriliyor
                    listBox2.Items.Clear();
                    for (int i = 0; i < enkisayol.Count; i++)
                    {
                        listBox2.Items.Add(enkisayol[i]);
                    }

                    //her turdaki en kısa yol listeye ekleniyor
                    listBox1.Items.Add(MinYolUzunluk);
                    FeromenGuncelle();
                    Refresh();//ekranı tazeliyor
                }

                grpDegisken.Enabled = true;
                button1.Enabled = true;
            }

            void FeromenGuncelle()
            {
                //buharlasma katsayısına göre bütün yollardaki
                feromenler güncelleniyor
                string sql1 = "UPDATE YOL SET FEROMEN=FEROMEN*(1-"
+ buharlasma.ToString().Replace(',', '.')+"")";
                db.ExecuteNonQuery(sql1);
                //her bir karıncanın güzergahındaki feromenler
                güncellenecek
                for (int k = 0; k < yedeksehir.Count; k++)
                {
                    List<string> kguzeergah = tabu[k];

                    string oncekisehir = kguzeergah[0];
                    string sonrakisehir = "";
                    for (int sira = 1; sira < kguzeergah.Count;
sira++)
                    {
                        sonrakisehir = kguzeergah[sira];

                        string sql = "UPDATE YOL SET
FEROMEN=FEROMEN+(1/"+tabuyollar[k]+ ") WHERE (CIKIS_SEHIR='" +
oncekisehir + "' AND VARIS_SEHIR='" + sonrakisehir + "') OR

```

Ek-2.(Devam) Uygulamaya ait kaynak kodları

```

(CIKIS_SEHIR='" + sonrakisehir + "' AND VARIS_SEHIR='" +
oncekisehir + "')";
        db.ExecuteNonQuery(sql);
        oncekisehir = sonrakisehir;
    }
}

string MinYolBul(string sehir_adi,int k)
{
    string sql = "SELECT * FROM YOL WHERE
(CIKIS_SEHIR='" + sehir_adi + "' OR VARIS_SEHIR='" + sehir_adi
+ "')";
    //tabu listesindeki son şehir hariç listedeki
değer şehirler aranılacak listeden çıkartılacak

    for (int sira = 0; sira < tabu[k].Count-1; sira++)
    {
        sql += " AND (NOT (CIKIS_SEHIR='" +
tabu[k][sira] + "' OR VARIS_SEHIR='" + tabu[k][sira] + "'))";
    }
    DataTable dt = db.Fill2(sql);

    double max = 0;
    string sonraki_sehir = "";
    for (int sira = 0; sira < dt.Rows.Count; sira++)
    {
        double
yol=Convert.ToDouble(dt.Rows[sira]["YOL"]);
        double feromen =
Convert.ToDouble(dt.Rows[sira]["FEROMEN"]);

        double deger = Math.Pow(feromen, alfa) *
Math.Pow(1 / yol, beta);
        if (deger > max)
        {
            max = deger;
            if (dt.Rows[sira]["CIKIS_SEHIR"].ToString()
== sehir_adi)
            {
                sonraki_sehir =
dt.Rows[sira]["VARIS_SEHIR"].ToString();
            }
            else
            {
                sonraki_sehir =
dt.Rows[sira]["CIKIS_SEHIR"].ToString();
            }
        }
    }
}

```

Ek-2.(Devam) Uygulamaya ait kaynak kodları

```

        return sonraki_sehir;
    }

    private List<string> SehirDoldur()
    {
        List<string> sehir = new List<string>();
        DataTable dt = db.Fill2("SELECT * FROM SEHIR");

        for (int sira = 0; sira < dt.Rows.Count ; sira++)
        {
            sehir.Add(dt.Rows[sira]["SEHIR_ADI"].ToString());
        }

        return sehir;
    }
}

using System;
using System.Collections.Generic;
using System.Text;
using System.Data;
using System.Data.OleDb;
using System.Windows.Forms;

namespace KKA
{
    public class DbClass
    {
        public string _connectionstring =
        "Provider=Microsoft.Jet.OleDb.4.0;Data
        Source="+Application.StartupPath+"/App_Data/vt.mdb;user
        id=admin;password=";

        OleDbConnection db = new OleDbConnection();

        public DbClass()
        {
            db.ConnectionString = _connectionstring;
        }

        public void ExecuteNonQuery(string sqlstr)
        {

```

Ek-2.(Devam) Uygulamaya ait kaynak kodları

```

        OleDbCommand cmd = new OleDbCommand(sqlstr, db);
        if (db.State==ConnectionState.Closed) db.Open();
        try
        {
            cmd.ExecuteNonQuery();
        }
        finally
        {
            db.Close();
        }
    }

    public int ExecuteNonQueryWithID(string sqlstr)
    {
        OleDbCommand cmd = new OleDbCommand(sqlstr, db);
        int deger = -1;
        if (db.State == ConnectionState.Closed) db.Open();
        try
        {
            deger = cmd.ExecuteNonQuery();
            OleDbCommand cmd1 = new OleDbCommand("SELECT
@@IDENTITY AS SAYI", db);
            deger = Convert.ToInt32(cmd1.ExecuteScalar());
        }
        finally
        {
            db.Close();
        }
        return deger;
    }

    public object ExecuteScalar(string sqlstr)
    {
        OleDbCommand cmd = new OleDbCommand(sqlstr, db);
        object deger = null;
        if (db.State == ConnectionState.Closed) db.Open();
        try
        {
            deger = cmd.ExecuteScalar();
        }
        finally
        {
            db.Close();
        }
        return deger;
    }

    public OleDbDataReader ExecuteReader(string sqlstr)
    {

```

Ek-2.(Devam) Uygulamaya ait kaynak kodları

```

        OleDbCommand cmd = new OleDbCommand(sqlstr, db);
        OleDbDataReader dr;
        if (db.State == ConnectionState.Closed) db.Open();
        dr =
cmd.ExecuteReader(CommandBehavior.CloseConnection);
        return dr;
    }

    public DataSet Fill1(string sqlstr, string tablename)
    {
        DataSet ds = new DataSet();
        OleDbDataAdapter dba = new
OleDbDataAdapter(sqlstr, db);
        dba.Fill(ds, tablename);
        return ds;
    }

    public DataTable Fill2(string sqlstr)
    {
        DataTable dt = new DataTable();
        OleDbDataAdapter dba = new
OleDbDataAdapter(sqlstr, db);
        dba.Fill(dt);
        return dt;
    }

    public bool Update(string sqlstr, DataSet ds)
    {
        OleDbDataAdapter dba = new
OleDbDataAdapter(sqlstr, db);
        OleDbCommandBuilder cmdbld = new
OleDbCommandBuilder(dba);
        if (ds.HasChanges() == true)
        {
            dba.Update(ds);
            return true;
        }
        else
        {
            return false;
        }
    }

    public bool Update(string sqlstr, DataTable dt)
    {
        OleDbDataAdapter dba = new
OleDbDataAdapter(sqlstr, db);

```

Ek-2.(Devam) Uygulamaya ait kaynak kodları

```

        OleDbCommandBuilder cmdbld = new
OleDbCommandBuilder(dba);

        if (dt.GetChanges() != null)
        {
            dba.Update(dt);
            return true;
        }
        else
        {
            return false;
        }
    }

}

namespace KKA
{
    partial class Form1
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components =
null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources
should be disposed; otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not
modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()

```

Ek-2.(Devam) Uygulamaya ait kaynak kodları

```

        {
            this.button1 = new System.Windows.Forms.Button();
            this.listBox1 = new
System.Windows.Forms.ListBox();
            this.label1 = new System.Windows.Forms.Label();
            this.label2 = new System.Windows.Forms.Label();
            this.label3 = new System.Windows.Forms.Label();
            this.label4 = new System.Windows.Forms.Label();
            this.grpDegisken = new
System.Windows.Forms.GroupBox();
            this.txtAlfa = new System.Windows.Forms.TextBox();
            this.txtBeta = new System.Windows.Forms.TextBox();
            this.txtBuharlasma = new
System.Windows.Forms.TextBox();
            this.txtIterasyon = new
System.Windows.Forms.TextBox();
            this.lblIterasyon = new
System.Windows.Forms.Label();
            this.label5 = new System.Windows.Forms.Label();
            this.listBox2 = new
System.Windows.Forms.ListBox();
            this.label6 = new System.Windows.Forms.Label();
            this.lblTur = new System.Windows.Forms.Label();
            this.label8 = new System.Windows.Forms.Label();
            this.label9 = new System.Windows.Forms.Label();
            this.label10 = new System.Windows.Forms.Label();
            this.lblMinYol = new System.Windows.Forms.Label();
            this.grpDegisken.SuspendLayout();
            this.SuspendLayout();
            //
            // button1
            //
            this.button1.Location = new
System.Drawing.Point(222, 187);
            this.button1.Name = "button1";
            this.button1.Size = new System.Drawing.Size(75,
23);

            this.button1.TabIndex = 0;
            this.button1.Text = "Başlat";
            this.button1.UseVisualStyleBackColor = true;
            this.button1.Click += new
System.EventHandler(this.button1_Click);
            //
            // listBox1
            //
            this.listBox1.FormattingEnabled = true;
            this.listBox1.Location = new
System.Drawing.Point(17, 250);
            this.listBox1.Name = "listBox1";

```

Ek-2.(Devam) Uygulamaya ait kaynak kodları

```

        this.listBox1.Size = new System.Drawing.Size(137,
173);
        this.listBox1.TabIndex = 3;
        //
        // label1
        //
        this.label1.AutoSize = true;
        this.label1.Location = new
System.Drawing.Point(15, 26);
        this.label1.Name = "label1";
        this.label1.Size = new System.Drawing.Size(25,
13);
        this.label1.TabIndex = 4;
        this.label1.Text = "Alfa";
        //
        // label2
        //
        this.label2.AutoSize = true;
        this.label2.Location = new
System.Drawing.Point(15, 59);
        this.label2.Name = "label2";
        this.label2.Size = new System.Drawing.Size(29,
13);
        this.label2.TabIndex = 5;
        this.label2.Text = "Beta";
        //
        // label3
        //
        this.label3.AutoSize = true;
        this.label3.Location = new
System.Drawing.Point(17, 92);
        this.label3.Name = "label3";
        this.label3.Size = new System.Drawing.Size(106,
13);
        this.label3.TabIndex = 6;
        this.label3.Text = "Buharlaştırma Katsayısı";
        //
        // label4
        //
        this.label4.AutoSize = true;
        this.label4.Location = new
System.Drawing.Point(18, 123);
        this.label4.Name = "label4";
        this.label4.Size = new System.Drawing.Size(80,
13);
        this.label4.TabIndex = 7;
        this.label4.Text = "İterasyon Sayısı";
        //
        // grpDegisken
        //

```

Ek-2.(Devam) Uygulamaya ait kaynak kodları

```

        this.grpDegisken.Controls.Add(this.label10);
        this.grpDegisken.Controls.Add(this.label9);
        this.grpDegisken.Controls.Add(this.label8);
        this.grpDegisken.Controls.Add(this.txtIterasyon);
        this.grpDegisken.Controls.Add(this.txtBuharlasma);
        this.grpDegisken.Controls.Add(this.txtBeta);
        this.grpDegisken.Controls.Add(this.txtAlfa);
        this.grpDegisken.Controls.Add(this.label11);
        this.grpDegisken.Controls.Add(this.label4);
        this.grpDegisken.Controls.Add(this.label2);
        this.grpDegisken.Controls.Add(this.label3);
        this.grpDegisken.Location = new
System.Drawing.Point(17, 19);
        this.grpDegisken.Name = "grpDegisken";
        this.grpDegisken.Size = new
System.Drawing.Size(280, 157);
        this.grpDegisken.TabIndex = 8;
        this.grpDegisken.TabStop = false;
        this.grpDegisken.Text = "Değişkenler";
        //
        // txtAlfa
        //
        this.txtAlfa.Location = new
System.Drawing.Point(134, 26);
        this.txtAlfa.Name = "txtAlfa";
        this.txtAlfa.Size = new System.Drawing.Size(100,
20);
        this.txtAlfa.TabIndex = 8;
        //
        // txtBeta
        //
        this.txtBeta.Location = new
System.Drawing.Point(134, 59);
        this.txtBeta.Name = "txtBeta";
        this.txtBeta.Size = new System.Drawing.Size(100,
20);
        this.txtBeta.TabIndex = 9;
        //
        // txtBuharlasma
        //
        this.txtBuharlasma.Location = new
System.Drawing.Point(134, 86);
        this.txtBuharlasma.Name = "txtBuharlasma";
        this.txtBuharlasma.Size = new
System.Drawing.Size(100, 20);
        this.txtBuharlasma.TabIndex = 10;
        //
        // txtIterasyon
        //

```

Ek-2.(Devam) Uygulamaya ait kaynak kodları

```

        this.txtIterasyon.Location = new
System.Drawing.Point(134, 115);
        this.txtIterasyon.Name = "txtIterasyon";
        this.txtIterasyon.Size = new
System.Drawing.Size(100, 20);
        this.txtIterasyon.TabIndex = 11;
        //
        // lblIterasyon
        //
        this.lblIterasyon.AutoSize = true;
        this.lblIterasyon.Location = new
System.Drawing.Point(14, 192);
        this.lblIterasyon.Name = "lblIterasyon";
        this.lblIterasyon.Size = new
System.Drawing.Size(35, 13);
        this.lblIterasyon.TabIndex = 9;
        this.lblIterasyon.Text = "label5";
        //
        // label5
        //
        this.label5.AutoSize = true;
        this.label5.Location = new
System.Drawing.Point(17, 231);
        this.label5.Name = "label5";
        this.label5.Size = new System.Drawing.Size(147,
13);
        this.label5.TabIndex = 10;
        this.label5.Text = "Her İterasyondaki En Kısa
Yol";
        //
        // listBox2
        //
        this.listBox2.FormattingEnabled = true;
        this.listBox2.Location = new
System.Drawing.Point(160, 250);
        this.listBox2.Name = "listBox2";
        this.listBox2.Size = new System.Drawing.Size(137,
173);
        this.listBox2.TabIndex = 11;
        //
        // label6
        //
        this.label6.AutoSize = true;
        this.label6.Location = new
System.Drawing.Point(171, 231);
        this.label6.Name = "label6";
        this.label6.Size = new System.Drawing.Size(112,
13);
        this.label6.TabIndex = 12;
        this.label6.Text = "En Kısa Yol Güzergahı";

```

Ek-2.(Devam) Uygulamaya ait kaynak kodları

```

//
// lblTur
//
this.lblTur.AutoSize = true;
this.lblTur.Location = new
System.Drawing.Point(204, 433);
this.lblTur.Name = "lblTur";
this.lblTur.Size = new System.Drawing.Size(23,
13);

this.lblTur.TabIndex = 13;
this.lblTur.Text = "Tur";
//
// label8
//
this.label8.AutoSize = true;
this.label8.Location = new
System.Drawing.Point(241, 26);
this.label8.Name = "label8";
this.label8.Size = new System.Drawing.Size(28,
13);

this.label8.TabIndex = 12;
this.label8.Text = "(0-1)";
//
// label9
//
this.label9.AutoSize = true;
this.label9.Location = new
System.Drawing.Point(241, 59);
this.label9.Name = "label9";
this.label9.Size = new System.Drawing.Size(28,
13);

this.label9.TabIndex = 13;
this.label9.Text = "(0-5)";
//
// label10
//
this.label10.AutoSize = true;
this.label10.Location = new
System.Drawing.Point(241, 92);
this.label10.Name = "label10";
this.label10.Size = new System.Drawing.Size(28,
13);

this.label10.TabIndex = 14;
this.label10.Text = "(0-1)";
//
// lblMinYol
//
this.lblMinYol.AutoSize = true;
this.lblMinYol.Location = new
System.Drawing.Point(199, 455);

```

Ek-2.(Devam) Uygulamaya ait kaynak kodları

```

        this.lblMinYol.Name = "lblMinYol";
        this.lblMinYol.Size = new System.Drawing.Size(39,
13);
        this.lblMinYol.TabIndex = 14;
        this.lblMinYol.Text = "MinYol";
        //
        // Form1
        //
        this.AutoScaleDimensions = new
System.Drawing.SizeF(6F, 13F);
        this.AutoScaleMode =
System.Windows.Forms.AutoScaleModeMode.Font;
        this.ClientSize = new System.Drawing.Size(322,
480);

        this.Controls.Add(this.lblMinYol);
        this.Controls.Add(this.lblTur);
        this.Controls.Add(this.label6);
        this.Controls.Add(this.listBox2);
        this.Controls.Add(this.label5);
        this.Controls.Add(this.lblIterasyon);
        this.Controls.Add(this.grpDegisken);
        this.Controls.Add(this.listBox1);
        this.Controls.Add(this.button1);
        this.Name = "Form1";
        this.Text = "Form1";
        this.Load += new
System.EventHandler(this.Form1_Load);
        this.grpDegisken.ResumeLayout(false);
        this.grpDegisken.PerformLayout();
        this.ResumeLayout(false);
        this.PerformLayout();

    }

    #endregion

    private System.Windows.Forms.Button button1;
    private System.Windows.Forms.ListBox listBox1;
    private System.Windows.Forms.Label label1;
    private System.Windows.Forms.Label label2;
    private System.Windows.Forms.Label label3;
    private System.Windows.Forms.Label label4;
    private System.Windows.Forms.GroupBox grpDegisken;
    private System.Windows.Forms.TextBox txtBuharlasma;
    private System.Windows.Forms.TextBox txtBeta;
    private System.Windows.Forms.TextBox txtAlfa;
    private System.Windows.Forms.TextBox txtIterasyon;
    private System.Windows.Forms.Label lblIterasyon;
    private System.Windows.Forms.Label label5;
    private System.Windows.Forms.ListBox listBox2;

```

Ek-2.(Devam) Uygulamaya ait kaynak kodları

```

        private System.Windows.Forms.Label label6;
        private System.Windows.Forms.Label lblTur;
        private System.Windows.Forms.Label label9;
        private System.Windows.Forms.Label label8;
        private System.Windows.Forms.Label label10;
        private System.Windows.Forms.Label lblMinYol;
    }
}

using System;
using System.Collections.Generic;
using System.Linq;
using System.Windows.Forms;

namespace KKA
{
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();

            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new Form1());
        }
    }
}

```

ÖZGEÇMİŞ**Kişisel Bilgiler**

Soyadı, adı : YETGİN Melike
Uyruğu : T.C.
Doğum tarihi ve yeri : 01.01.1983 KAYSERİ
Medeni hali : Evli
e-mail : melikeyetkin@gmail.com

Eğitim

Derece	Eğitim Birimi	Mezuniyet tarihi
Lisans	Erciyes Üniversitesi/endüstri Mühendisliği	2005

İş Deneyimi

Yıl	Yer	Görev
2010-	T.C.D.D. 2. Bölge Müdürlüğü	Mühendis

Yabancı Dil

İngilizce