



İSKENDERUN TEKNİK

ÜNİVERSİTESİ

MÜHENDİSLİK VE FEN BİLİMLERİ ENSTİTÜSÜ

**YÜKSEK
LİSANS
TEZİ**

**ÇOK TÜRLÜ TAŞIMACILIK
GÜZARGAHLARINDA LİNEER
OPTİMİZASYON VE PARÇACIKLI
SÜRÜ OPTİMİZASYONUNUN
KARŞILAŞTIRILMASI**

Mehmet Ali BOZKURT

İNŞAAT MÜHENDİSLİĞİ
ANABİLİM DALI

HAZİRAN 2019



**ÇOK TÜRLÜ TAŞIMACILIK GÜZERGÂHLARINDA LİNEER
OPTİMİZASYON VE PARÇACIKLI SÜRÜ OPTİMİZASYONUNUN
KARŞILAŞTIRILMASI**

Mehmet Ali BOZKURT

**YÜKSEK LİSANS TEZİ
İNŞAAT MÜHENDİSLİĞİ ANABİLİM DALI**

**İSKENDERUN TEKNİK ÜNİVERSİTESİ
MÜHENDİSLİK VE FEN BİLİMLERİ ENSTİTÜSÜ**

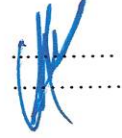
HAZİRAN 2019

Mehmet Ali BOZKURT tarafından hazırlanan “ÇOK TÜRLÜ TAŞIMACILIK GÜZERGÂHLARINDA LİNEER OPTİMİZASYON VE PARÇACIKLI SÜRÜ OPTİMİZASYONUNUN KARŞILAŞTIRILMASI” adlı tez çalışması aşağıdaki jüri tarafından OY BİRLİĞİ ile İskenderun Teknik Üniversitesi İnşaat Mühendisliği Anabilim Dalında YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Danışman: Dr. Öğr. Üyesi Ömer Faruk CANSIZ
İnşaat Mühendisliği Anabilim Dalı, İskenderun Teknik Üniversitesi
Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum.



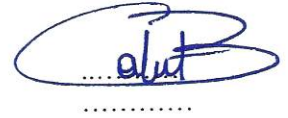
Başkan: Prof. Dr. Umur Korkut SEVİM
İnşaat Mühendisliği Anabilim Dalı, İskenderun Teknik Üniversitesi
Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum.



Üye: Prof. Dr. Fatih ÜNEŞ
İnşaat Mühendisliği Anabilim Dalı, İskenderun Teknik Üniversitesi
Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum.



Üye: Prof. Dr. Cahit BİLİM
İnşaat Mühendisliği Anabilim Dalı, Mersin Üniversitesi
Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum.



Tez Savunma Tarihi: 10/06/2019

Jüri tarafından kabul edilen bu tezin Yüksek Lisans Tezi olması için gerekli şartları yerine getirdiğini onaylıyorum.

Prof. Dr. Tolga DEPCİ
Mühendislik ve Fen Bilimleri Enstitüsü Müdürü



ETİK BEYAN

İskenderun Teknik Üniversitesi Mühendislik ve Fen Bilimleri Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmada;

- ☒ Yükseköğretim Kuruluna gönderilen kopya ile tarafından Mühendislik ve Fen Bilimleri Enstitüsü'ne verilen basılı ve/veya elektronik kopyaların birebir aynı olduğunu,
- ☒ Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- ☒ Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- ☒ Tez çalışmada yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- ☒ Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- ☒ Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

İmza

Mehmet Ali BOZKURT

10/06/2019



ÇOK TÜRLÜ TAŞIMACILIK GÜZERGAHLARINDA LİNEER OPTİMİZASYON VE
PARÇACIKLI SÜRÜ OPTİMİZASYONUNUN KARŞILAŞTIRILMASI
(Yüksek Lisans Tezi)

Mehmet Ali BOZKURT

İSKENDERUN TEKNİK ÜNİVERSİTESİ
MÜHENDİSLİK VE FEN BİLİMLERİ ENSTİTÜSÜ

Haziran 2019

ÖZET

Yolcu ve yük taşınmasında hemen her ülkede demiryolu, karayolu ve havayolu ulaştırmasının yanında ülkenin coğrafi konumuna göre deniz yolu ulaştırmasından yararlanmaktadır. Son yıllarda ki genel yaklaşım, taşımacılığı bir bütün olarak ele alıp, aynı taşımacılıkta her ulaşım türünden en etkin bir şekilde yararlanma yoluna gitmektir. Bu sayede taşıma maliyetlerinin ve toplam taşıma süresinin azaltılmasına kadar birçok konuda avantaj sağlanmaktadır. Bu çalışmada, 18 nokta ve bunlar arasından seçilen 3 ve 4 nokta için çok türlü taşımacılık ile mesafe, süre ve birim maliyet değerleri hesaplanmaktadır. Yapılan çalışmada Lineer Optimizasyon ve Parçacıklı Sürü Optimizasyonu (PSO) yöntemi kullanılmaktadır. Lineer Optimizasyon ile yapılan analizler sonucunda 3 nokta için elde edilen değerler; mesafe 1560 km, süre 36,97 saat ve birim maliyet 116,21 TL olarak hesaplanmaktadır. 4 nokta için elde edilen değerler; mesafe 1572 km, süre 37,75 saat ve birim maliyet 125,37 TL olarak hesaplanmaktadır. 18 nokta için elde edilen değerler ise; mesafe 3282 km, süre 71,65 saat ve birim maliyet 394,62 TL olarak hesaplanmaktadır. PSO için yapılan analizler sonucunda 3 nokta için elde edilen değerler; mesafe 1560 km, süre 36,97 saat ve birim maliyet 116,21 TL olarak hesaplanmaktadır. 4 nokta için elde edilen değerler; mesafe 1572 km, süre 37,75 saat ve birim maliyet 125,37 TL olarak hesaplanmaktadır. 18 nokta için elde edilen değerler ise; mesafe 3295 km, süre 71,65 saat ve birim maliyet 394,62 TL olarak hesaplanmaktadır. Lineer Optimizasyon ve PSO yöntemleri sonucu elde edilen değerler birbirleri ile kıyaslandığında; 18 nokta arasındaki mesafe değeri Lineer Optimizasyonda daha iyi çıktığı, diğer değerlerin ise aynı olduğu görülmektedir.

Anahtar Kelimeler : Çok Türlü Taşımacılık, Lineer Optimizasyon, PSO, Maliyet, Süre, Mesafe
Sayfa Adedi : 115
Danışman : Dr. Öğr. Üyesi Ömer Faruk CANSIZ

COMPARISON OF LINEAR OPTIMIZATION AND PARTICLE SWARM OPTIMIZATION FOR TRANSPORTATION ROUTES

(M. Sc. Thesis)

Mehmet Ali BOZKURT

ISKENDERUN TECHNICAL UNIVERSITY
ENGINEERING AND SCIENCE INSTITUTE

June 2019

ABSTRACT

In the transportation of passengers and freight, almost every country benefits from sea transportation according to the geographical location of the country besides railway and road transportation. The general approach in recent years is to take care of transportation as a whole and to make use of every mode of transportation in the same transportation in the most efficient way. This provides an advantage in many aspects, from transportation costs to total transportation time reduction. In this study, 18 cities and 3 and 4 cities selected among them are calculated with various types of transportation, distance, time and unit cost values. In this study, Linear Optimization and Particle Swarm Optimization (PSO) method is used. As a result of the analysis with Linear Optimization, the values obtained for 3 cities; The distance is calculated as 1560 km, the time is 36.97 hours and the unit cost is 116.21 TL. Values obtained for 4 cities; The distance is calculated as 1572 km, the duration is 37,75 hours and the unit cost is 125,37 TL. The values obtained for 18 cities are; the distance is calculated as 3282 km, the period of 71,65 hours and the unit cost is 394,62 TL. As a result of the analysis for the PSO, the values obtained for 3 cities; The distance is calculated as 1560 km, the time is 36.97 hours and the unit cost is 116.21 TL. Values obtained for 4 cities; The distance is calculated as 1572 km, the time is 37,75 hours and the unit cost is 125,37 TL. The values obtained for 18 cities are; the distance is calculated as 3295 km, the duration is 71,65 hours and the unit cost is 394,62 TL. Compared to Linear Optimization and PSO methods; it is seen that the distance between 18 cities is better for Linear Optimization and the other values are the same.

Key Words : Multimodal Transport, Linear Optimization, PSO, Cost, Time, Distance
Page Number : 115
Supervisor : Assist. Prof. Dr. Ömer Faruk CANSIZ

TEŞEKKÜR

Yüksek linsans tez konusunun belirlenmesinde, araştırılması ve yazımı sırasında sahip olduğu bilgi birikimi ve tecrübesi ile çalışmayı yönlendiren ve her türlü yardımı esirgemeyen saygıdeğer danışman hocam Dr.Öğr. Üyesi Ömer Faruk CANSIZ'a ve Ceng AYGÜN'e sonsuz saygı ve teşekkürlerimi sunarım.



İÇİNDEKİLER

	Sayfa
ÖZET	iv
ABSTRACT.....	v
TEŞEKKÜR.....	vi
İÇİNDEKİLER	vii
ÇİZELGELERİN LİSTESİ.....	xi
ŞEKİLLERİN LİSTESİ.....	xiii
SİMGELER VE KISALTMALAR.....	xvi
1. GİRİŞ.....	1
1.1. Lojistik	2
1.2. Taşımacılık Türleri.....	3
1.2.1. Tek türlü taşımacılık	3
1.2.2. Çok türlü taşımacılık	4
1.2.3. İntermodal taşımacılık.....	5
1.2.4. Kombine taşımacılık	5
1.3. Gezgin Satıcı Problemi	6
1.4. Lineer Optimizasyon	6
1.5. Parçacıklı Sürü Optimizasyonu	7
2. ÖNCEKİ ÇALIŞMALAR.....	9
3. MATERYAL VE YÖNTEM	25
3.1. Çalışmanın Amacı.....	25
3.2. Çalışma Güzergâhında Yaygın Kullanılan Lojistik Hat Bilgileri	26
3.2.1. Lojistik köyler ve ticari limanların dağılımı	26
3.2.2. Lojistik köyler ve ticari limanlar yaygın kullanılan hat bilgileri	28

	Sayfa
3.3. Lojistik Köyler ve Ticari Limanların Konum ve Koordinatları	29
3.4. Lojistik Köyler ve Ticari Limanların Birbirine Karayolu, Demiryolu ve Denizyolu Mesafeleri	30
3.5. Lojistik Köyler ve Ticari Limanların Birbirine Karayolu, Demiryolu ve Denizyolu Süreleri	33
3.6. Lojistik Köyler ve Ticari Limanların Birbirine Karayolu, Demiryolu ve Denizyolu Birim Maliyetleri	36
3.7. Lineer Optimizasyon	40
3.7.1. Üç noktalı lineer optimizasyon algoritmasının oluşturulması	40
4. ARAŞTIRMA BULGULARI VE TARTIŞMA	47
4.1. Lineer Optimizasyon İle Koordinatlara Göre Standart Çözümü.....	47
4.1.1. Güzergâhın Türkiye haritasında gösterimi	48
4.2. Lineer Optimizasyon İle Karayolu Mesafelerine Göre Çözümü	48
4.2.1. Güzergâhın Türkiye haritasında gösterimi	49
4.3. Lineer Optimizasyon İle Karayolu Sürelerine Göre Çözümü.....	49
4.3.1. Güzergâhın Türkiye haritasında gösterimi	50
4.4. Lineer Optimizasyon İle Karayolu Birim Maliyetlerine Göre Çözümü	51
4.4.1. Güzergâhın Türkiye haritasında gösterimi	51
4.5. Parçacıklı Sürü Optimizasyon İle Koordinatlara Göre Standart Çözümü	52
4.5.1. Güzergâhın Türkiye haritasında gösterimi	53
4.6. Parçacıklı Sürü Optimizasyon İle Karayolu Mesafelerine Göre Çözümü	53
4.6.1. Güzergâhın Türkiye haritasında gösterimi	54
4.7. Parçacıklı Sürü Optimizasyon İle Karayolu Sürelerine Göre Çözümü.....	54
4.7.1. Güzergâhın Türkiye haritasında gösterimi	55
4.8. Parçacıklı Sürü Optimizasyon İle Karayolu Birim Maliyetlerine Göre Çözümü	56

4.8.1. Güzergâhın Türkiye haritasında gösterimi.....	56
4.9. Lineer Optimizasyon İle Çok Türlü Mesafelere Göre Çözümü.....	57
4.9.1. Güzergâhın Türkiye haritasında gösterimi.....	58
4.10. Lineer Optimizasyon İle Çok Türlü Sürelere Göre Çözümü	59
4.10.1. Güzergâhın Türkiye haritasında gösterimi.....	60
4.11. Lineer Optimizasyon İle Çok Türlü Birim Maliyetlere Göre Çözümü.....	61
4.11.1. Güzergâhın Türkiye haritasında gösterimi.....	62
4.12. Parçacıklı Sürü Optimizasyonu İle Çok Türlü Mesafelere Göre Çözümü.....	63
4.12.1. Güzergâhın Türkiye haritasında gösterimi.....	64
4.13. Parçacıklı Sürü Optimizasyonu İle Çok Türlü Sürelere Göre Çözümü	65
4.13.1. Güzergâhın Türkiye haritasında gösterimi.....	66
4.14. Parçacıklı Sürü Optimizasyonu İle Çok Türlü Birim Maliyetlere Göre Çözümü.....	66
4.14.1. Güzergâhın Türkiye haritasında gösterimi.....	68
4.15. 18 Nokta için Karayolu Taşımacılığında Mesafe, Süre ve Birim Maliyet Değerlerinin Karşılaştırılması.....	68
4.16. 18 Nokta için Çok Türlü Taşımacılıkta Mesafe, Süre ve Birim Maliyet Değerlerinin Karşılaştırılması.....	69
5. SONUÇLAR VE ÖNERİLER	70
KAYNAKLAR	72
EKLER.....	76
EK-1. PSO Ana Algoritması: Çok Türlü Mesafe Örneği	76
EK-2. Model Fonksiyonu: Çok Türlü Mesafe Örneği	78
EK-3. Hesap Fonksiyonu: Çok Türlü Mesafe Örneği	92
EK-4. Mutasyon Fonksiyonu	93
EK-5. Yazdırma Fonksiyonu	94

Sayfa

EK-6. Lineer Optimizasyon Ana Algoritması: Çok Türlü Mesafe Örneği.....	94
EK-7. Lineer Optimizasyon Kodlaması: Multimodal Mesafe Örneğinde Kullanılan 18 Nokta Arasından Seçilen 4 Noktadan Oluşan Mesafe Örneği Kodlaması	111
ÖZGEÇMİŞ	114
DİZİN	115



ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 3.1. Yaygın kullanılan güzergâh bilgileri	28
Çizelge 3.2. Nokta koordinatları	30
Çizelge 3.3. Karayolu mesafeleri	31
Çizelge 3.4. Demiryolu mesafeleri	32
Çizelge 3.5. Denizyolu mesafeleri	33
Çizelge 3.6. Karayolu süreleri	34
Çizelge 3.7. Demiryolu süreleri	35
Çizelge 3.8. Denizyolu süreleri	36
Çizelge 3.9. Karayolu birim maliyetleri	38
Çizelge 3.10. Demiryolu birim maliyetleri	39
Çizelge 3.11. Denizyolu birim maliyetleri	40
Çizelge 3.12. Şehirlerarasındaki minimum mesafeleri alıp matris haline getiren kodlar	40
Çizelge 3.13. Şehirlerarasındaki minimum mesafeleri alıp matris haline getiren kodların sonuçları	41
Çizelge 3.14. Şehir çiftlerini ve mesafe vektörünü oluşturan kodlama	41
Çizelge 3.15. Şehir çiftlerini ve mesafe vektörünü oluşturan kodlamanın sonuçları	41
Çizelge 3.16. Eşitlik kısıtları kodlamaları	42
Çizelge 3.17. Eşitlik kısıtları kodlaması sonuçları	42
Çizelge 3.18. Eşitlik kısıtları kodlamaları	42
Çizelge 3.19. Eşitlik kısıtları kodlaması sonuçları	42
Çizelge 3.20. Olmayan yolların belirlenmesinde kullanılan kodlar	43
Çizelge 3.21. Olmayan yolların belirlenmesinde kullanılan kodların sonuçları	43
Çizelge 3.22. Alt ve üst sınırlamaların belirtildiği kodlar	44

Çizelge	Sayfa
Çizelge 3.23. Alt ve üst sınırlamaların belirtildiği kodların sonuçları	44
Çizelge 3.24. Üç nokta arasındaki güzergâhın belirlenmesinde kullanılan kodlar	45
Çizelge 3.25. Üç nokta arasındaki güzergâhın belirlenmesinde kullanılan kodların sonuçları.....	45
Çizelge 3.26. Altur turların belirlenmesinde kullanılan kodlar	45
Çizelge 3.27. Altur turların belirlenmesinde kullanılan kodların sonuçları	45
Çizelge 3.28. Alt turları elimine ederek optimum güzergâhı oluşturan kodlar.....	46
Çizelge 3.29. Alt turları elimine ederek optimum güzergâhı oluşturan kodların sonuçları	46

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 3.1. Çalışma güzergâhında yaygın kullanılan hat bilgileri şeması	27
Şekil 4.1. 3 ve 4 noktalı gsp örneğinin koordinatlara göre çözümünün harita üzerinde gösterimi	48
Şekil 4.2. 18 noktalı gsp örneğinin koordinatlara göre çözümünün harita üzerinde gösterimi	48
Şekil 4.3. 3 ve 4 noktalı gsp örneğinin gerçek mesafelere göre çözümünün harita üzerinde gösterimi	49
Şekil 4.4. 18 noktalı gsp örneğinin gerçek mesafelere göre çözümünün harita üzerinde gösterimi	49
Şekil 4.5. 3 ve 4 noktalı gsp örneğinin gerçek sürelerle göre çözümünün harita üzerinde gösterimi	50
Şekil 4.6. 18 noktalı gsp örneğinin gerçek sürelerle göre çözümünün harita üzerinde gösterimi	50
Şekil 4.7. 3 ve 4 noktalı gsp örneğinin gerçek birim maliyetlere göre çözümünün harita üzerinde gösterimi	51
Şekil 4.8. 18 noktalı gsp örneğinin gerçek birim maliyetlere göre çözümünün harita üzerinde gösterimi.	52
Şekil 4.9. 3 ve 4 noktalı gsp örneğinin koordinatlara göre çözümünün harita üzerinde gösterimi	53
Şekil 4.10. 18 noktalı gsp örneğinin koordinatlara göre çözümünün harita üzerinde gösterimi	53
Şekil 4.11. 3 ve 4 noktalı gsp örneğinin gerçek mesafelere göre çözümünün harita üzerinde gösterimi	54
Şekil 4.12. 18 noktalı gsp örneğinin gerçek mesafelere göre çözümünün harita üzerinde gösterimi	54
Şekil 4.13. 3 ve 4 şehirli gsp örneğinin gerçek sürelerle göre çözümünün harita üzerinde gösterimi	55
Şekil 4.14. 18 noktalı gsp örneğinin gerçek sürelerle göre çözümünün harita üzerinde gösterimi.....	55
Şekil 4.15. 3 ve 4 noktalı gsp örneğinin gerçek birim maliyetlere göre çözümünün harita üzerinde gösterimi.....	56

Şekil	Sayfa
Şekil 4.16. 18 noktalı gsp örneğinin gerçek birim maliyetlere göre çözümünün harita üzerinde gösterimi.....	57
Şekil 4.17. 3 ve 4 noktalı gsp örneğinin çok türlü taşımacılık mesafeye göre çözümünün harita üzerinde gösterimi.....	58
Şekil 4.18. 18 noktalı gsp örneğinin çok türlü taşımacılık mesafeye göre çözümünün harita üzerinde gösterimi.....	59
Şekil 4.19. 3 ve 4 noktalı gsp örneğinin çok türlü taşımacılık süreye göre çözümünün harita üzerinde gösterimi.....	60
Şekil 4.20. 18 noktalı gsp örneğinin çok türlü taşımacılık süreye göre çözümünün harita üzerinde gösterimi.....	61
Şekil 4.21. 3 ve 4 noktalı gsp örneğinin çok türlü taşımacılık birim maliyete göre çözümünün harita üzerinde gösterimi.....	62
Şekil 4.22. 18 noktalı gsp örneğinin çok türlü taşımacılık birim maliyete göre çözümünün harita üzerinde gösterimi.....	63
Şekil 4.23. 3 ve 4 noktalı gsp örneğinin çok türlü taşımacılık mesafeye göre çözümünün harita üzerinde gösterimi.....	64
Şekil 4.24. 18 noktalı gsp örneğinin çok türlü taşımacılık mesafeye göre çözümünün harita üzerinde gösterimi	64
Şekil 4.25. 3 ve 4 noktalı gsp örneğinin çok türlü taşımacılık süreye göre çözümünün harita üzerinde gösterimi	66
Şekil 4.26. 18 noktalı gsp örneğinin çok türlü taşımacılık süreye göre çözümünün harita üzerinde gösterimi.....	66
Şekil 4.27. 3 ve 4 noktalı gsp örneğinin çok türlü taşımacılık birim maliyete göre çözümünün harita üzerinde gösterimi.....	68
Şekil 4.28. 18 noktalı gsp örneğinin çok türlü taşımacılık birim maliyete göre çözümünün harita üzerinde gösterimi.....	68
Şekil 4.29. Karayolu taşımacılığında mesafe, süre ve birim maliyete göre karşılaştırma	69
Şekil 4.30. Çok türlü taşımacılıkta mesafe, süre ve birim maliyete göre karşılaştırma	69

SİMGELER VE KISALTMALAR

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

Kısaltmalar	Açıklamalar
ABCA	Yapay Arı Kolonisi Algoritması
AHP	Analitik Hiyerarşi Prosesi
AK	Avrupa Komisyonu
ARP	Araç Rotalama Problemi
CBS	Coğrafi Bilgi Sistemleri
DARP	Dinamik Araç Rotalama Problemi
ECE	Avrupa Ekonomi Komisyonu
GA	Genetik Algortima
GSP	Gezgin Satıcı Problemi
KGM	Karayolları Genel Müdürlüğü
PIC	Çevresel Arabilim Kontrol Cihazı
RPS	Rota Planlama Sistemi
RYS	Rota Yönlendirme Sistemi
TCDD	Türkiye Cumhuriyeti Devlet Demir Yolları
TCP	Geçiş Kontrol Protokolü
UBAK	Ulaştırma Bakanları Avrupa Konferansı
UNECE	Birleşmiş Milletler Avrupa Ekonomik Komisyonu
YBS	Yapay Bağışıklık Sistemi

1. GİRİŞ

Lojistik; pazarlama sektörü, taşımacılık ve nakliyat ile başladığı için ekonomi ile doğrudan etkileşim halindedir. Teknolojinin gelişmesi ile beraber globalleşen dünyamızda, lojistik sektörünün bu gelişime paralel olarak ilerlemesi ülke ekonomisinin gelişimi açısından büyük önem arz etmektedir. Lojistikte tüm masrafların ve nakliye sürelerinin minimize edilmesi için, hem lojistik ile doğrudan ilgili kurum ve kuruluşların ekonomisine, hem de ülke ekonomisine olumlu katkı sağlayacak ve bu durumu sürdürülebilir kılacak optimizasyonun sağlanması gerekmektedir.

Lojistik ülke içi taşımacılığı baz alındığında mesafelerin genel olarak daha kısa olmasından dolayı, tek türlü taşımacılık ile gerçekleştirilmektedir. Mesafenin kısa olması, diğer taşımacılık türlerinin elverişli olmaması ve yükleme boşaltmada yaşanacak zaman kaybı gibi sebeplerden dolayı genel olarak birimler için ekonomik boyutunu ikinci planda tutmaktadır ve yükün zamanında ulaşmasına özen göstermektedir. Uluslararası taşımacılıkta ise deniz bağlantısı olan ülkelere genellikle gemi taşımacılığı tercih edilmektedir. Gemi taşımacılığı ile gelen yükün ülkenin ulaşım altyapısına bağlı olarak ulaşım türü değişimi yapılarak yükün alıcıya ulaşması sağlanmaktadır. Bu şekilde ulaşım türleri değiştirilerek yapılan taşımacılık türüne çok türlü taşımacılık denilmektedir.

Ülkemiz lojistik altyapısı günümüz şartlarında çok türlü taşımacılık için çok uygun olmasa da gerek Türkiye Cumhuriyeti Devlet Demir Yolları (T.C.D.D.) tarafından çalışmaları devam eden lojistik köy projeleri, gerek ulaşım altyapılarına yapılan devlet yatırımları gerekse bu tarz optimizasyon çalışmaları ile teşvik sağlanmaya çalışılmaktadır.

Çalışmanın giriş bölümünde çalışmanın tanımı ve amacı açıklanmış, lojistikle ilgili kavramlar ve taşımacılık türleri detaylı olarak anlatılmaktadır. Materyal ve Yöntem bölümünde optmizasyon çalışması için kullanılacak olan ülkemiz lojistik köyleri ve ticari limanları arası tüm varyasyonlara ait; mesafeler, süreler ve birim maliyetler verilerek kaynakları ile beraber hesapları anlatılmıştır. Optimizasyon için GSP baz alınarak yazılan lineer optimizasyon ve PSO çözüm algoritması açıklanmış ve gerçek mesafelere göre çözüm yapabilmek ve taşımacılık türü değiştirebilmek için algoritma üzerinde yapılan modifiyeler anlatılmıştır. Araştırma Bulguları ve Tartışma bölümünde, öncelikle örnek optmizasyon çalışması için kullanılacak noktalar arasında GSP'nin nokta koordinatlarına

göre geleneksel doğrusal çözümü yapılmıştır. Daha sonra tek türlü karayolu için gerçek değerlere göre mesafe, süre ve maliyet optimizasyonları yapılmıştır. Son olarak ise yine gerçek değerlere göre çok türlü taşımacılık için mesafe, süre ve maliyet optimizasyonları yapılmıştır. Çalışmanın son bölümü olan Sonuç ve Öneriler bölümünde ise mesafe, süre ve maliyet parametrelerine göre çözümü yapılan optimizasyon sonuçları tek türlü karayolu taşımacılığı ve çok türlü taşımacılık için karşılaştırılmış ve üç parametre için de çok türlü taşımacılığın daha avantajlı olduğu görülmüştür.

1.1. Lojistik

Askerlik bilgisinin, savaş sırasında ya da askeri bir yürüyüşte ordunun yiyecek içecek sağlama, sağlık ve haberleşme hizmetleri, yol koşulları vb. yönlerinden en etkili durumda bulundurulması amacını güden çalışmaları bir araya getiren bölümü olarak en eski tanımı yapılabilen lojistik, tanımından da anlaşılacağı üzere askeri kökenli bir kavramdır. Ancak günümüzde özellikle 1900’lü yılların ortalarından sonra teknolojinin gelişimi ile beraber büyüyen dünya ekonomisi sınırları kaldırmış ve ticarete lojistiği ön plana çıkarmaya başlamıştır. Bu yıllardan itibaren lojistik ticari taşımacılıkla daha çok anılmaya başlanmıştır. Lojistiğin günümüzde daha çok kullanılan tanımı ise; üreticiden sonuncu tüketiciye varana kadar bir ürünün dağıtım, ambalajlama, gümrükleme, depolama ve nakliye gibi bütün süreçlerini kapsar.

Geleneksel taşımacılık ve ulaşım anlayışının, küreselleşmeyle yaşanan ticari eğilimler ve teknolojiye gelişmeler sonucunda köklü değişimlere uğraması lojistik kavramı içinde değerlendirilmektedir. Lojistik, yeni ekonomi anlayışıyla çok daha ileri bir boyuta taşınmış; satın alma, danışmanlık, eşyanın gümrük beyanının yapılması, sigortalanması, 3 depolama, sipariş izleme, envanter yönetimi, talep tahmini, yedek parça desteği, dağıtım, iade işlemleri, üretime malzeme verme, katma değerli işlemler (etiketleme, paketleme, fiyat-barkod, müşteri taleplerine göre ürün hazırlama, birleştirme-ayırma vs.), kıymetli evrakın hazırlanması ve ödemenin tahsilatı, araç optimizasyonu ile sevkiyat gibi çok çeşitli faaliyetler günümüzde bütünleşip lojistiğin sağladığı hizmetler olarak sunulmaktadır. Bu kapsamda lojistik tüm bu faaliyetlerin planlı ve entegre edilmiş bir şekilde gerçekleştirildiği hizmetler zinciridir (Akandere, 2013).

1.2. Taşımacılık Türleri

Taşımacılık türleri tanımı; karayolu taşımacılığı, denizyolu taşımacılığı, havayolu taşımacılığı, demiryolu taşımacılığı, boru hattı taşımacılığı ve iç su hattı taşımacılıklarını kapsamaktadır. Lojistik sektörü açısından ele alındığında yaygın kullanılan taşımacılık türleri; denizyolu taşımacılığı, karayolu taşımacılığı ve demiryolu taşımacılığıdır. Hammadde, yarı mamul veya bitmiş ürünlerin transferini kapsayan yük taşımacılığı, tedarik zincirinin önemli bir parçasıdır. Yük taşımacılığındaki amaç ürünlerin müşterilere, doğru zamanda, doğru kalitede ve doğru fiyata teslim edilmesidir (Urciuoli, 2016). Uluslararası lojistikte deniz bağlantısı olan ülkelere denizyolu taşımacılığı başı çekmektedir ve altyapı durumuna göre denizyolu taşımacılığını demiryolu ve karayolu taşımacılıkları takip eder. İç lojistikte ise mesafelerin daha kısa olmasının da etkisi ile önceliği karayolu taşımacılığı oluşturmakla birlikte yine altyapı durumuna göre demiryolu taşımacılığı ve denizyolu taşımacılığı takip etmektedir. Taşımacılık türlerinin kullanımına göre; tek türlü taşımacılık, çok türlü taşımacılık, intermodal taşımacılık ve kombine taşımacılık olmak üzere 4 genel taşımacılık modu bulunmaktadır.

1.2.1. Tek türlü taşımacılık

Tek modlu taşıma diğer adı ile tek türlü taşımacılık bir veya birden fazla taşımacı ile tek bir taşıma modülü ile yapılan taşıma sistemidir. Yapılan taşıma işleminde tek bir taşıma metodu kullanılması esastır. Tek modlu taşımacılıkta en yaygın olarak kullanılan taşıma türü kara yoludur. Karayolu taşımacılığının yaygın olarak kullanılmasının sebepleri; altyapı yatırımları diğer taşıma sistemlerine göre daha az olması, yükleme ve boşaltma işlemlerinde demiryolu veya hava yoluna göre daha kolay olması, terminal gereksiniminin mukayeseli olarak daha az olması ve coğrafi koşullar açısından değerlendirildiğinde ulaşım ağının geniş olmasından dolayı, kısa mesafelerde ekonomik çözümler sunması, tek modlu taşıma yönteminde karayolu en yaygın kullanılan taşımacılık olarak avantajları ile değerlendirilmektedir. Diğer taraftan hız faktörünün göz ardı edilebildiği ve ulaşım açısının da uygun olduğu tek modlu taşımalarda deniz yolu taşımacılığı da avantajları ile değerlendirilmektedir. Bu avantajlar; en düşük maliyetli taşıma modeli oluşu, transit geçişlerde gümrük işlemlerinin olmayışı olarak en önemlileri arasında sayılabilir. Diğer taraftan deniz yolu tek modlu taşımanın dezavantajları değerlendirildiğinde; hava şartlarından süre ve hasar miktarında artış ihtimali, hizmet verilebilecek yerleşkelerin

limanlar ile sınırlı olması, varış sürelerindeki esnekliğin düşük olması bu metod için değerlendirilmesi gereken faktörlerdir. Tek mod taşımacılıkta kullanılan diğer bir yöntem ise hava yolu taşımacılığıdır. Bu taşımacılık birim ağırlık açısından en yüksek maliyete sahip olması ile beraber süre açısından en esnek taşımacılık yöntemidir. Hava taşımacılık özellikle hassas ve pahada ağır olan ürünler, elektronik veya antika eşyalar gibi emtialar için yüksek güvenlik ve elleçleme sağladığı için avantajları ön plana çıkabilmektedir.

Taşımacılığın yalnızca bir ulaşım türü ile gerçekleştirildiği tek türlü taşımacılık açısından düşünüldüğünde: havayolu taşımacılığı en hızlı ve güvenli taşımacılık sağlayan ancak en maliyetli taşımacılık türü olduğu görülmektedir. Denizyolu taşımacılığı en ucuz ve kolay taşımacılık sağlayan ancak süre olarak en yavaş ve güvenliği düşük taşımacılık türüdür. Karayolu ve demiryolu taşımacılıkları nispeten daha hızlı ve güvenli taşımacılık sağlamakta ancak denizyolu taşımacılığına oranla daha maliyetli taşımacılık türleridir. Karayolu taşımacılığı ve demiryolu taşımacılıkları kendi aralarında karşılaştırıldıklarında ise demiryolu taşımacılığı daha ekonomik, karayolu taşımacılığı ise daha hızlı taşımacılık yapabilmektedir. İleri araştırmalar tek türlü karayolu taşımacılığının daha çok kısa mesafeli taşımacılık için kullanıldığını ortaya koymaktadır. Tek türlü demiryolu taşımacılığı ise daha uzun mesafeler için kullanılmaktadır.

1.2.2. Çok türlü taşımacılık

Değişik taşıma araçları veya türleri ile birden çok taşıma türü ile yapılan lojistik taşımacılığına çok türlü transport veya çoklu taşımacılık denir. Çok türlü taşımacılığın sıklıkla gündeme gelmesi modern konteynerlerin ortaya çıkması ile olmaktadır. Konteyner standart kap niteliğindedir ve yükün hem ambalajıdır hem de yükü taşıma görevini yerine getirir. Çoklu taşımacılığın temel kullanım malzemesi olan konteyner ayrıca taşımacılık yapan kurum ve kuruluşlar açısından bir yatırım aracıdır. Konteyner operatörleri çok türlü taşımacılığın ilk uygulayıcılarıdır. Taşıyıcı ile alıcı arasında yapılan tek yük senedi ile en az iki taşıma sistemi kullanılarak yapılan taşımacılık türüne çok türlü taşımacılık denir. Çok türlü taşımacılık uluslararası alanlarda yapılabileceği gibi ülkelerin ulusal sınırları içerisinde yerel olarakta yapılabilmektedir.

Çıkış noktası ile varış noktası arasında taşıma açısından bakıldığında günümüzde emtialar (Türk Dil Kurumu Büyük Türkçe Sözlüğüne göre) farklı taşıma araçları kullanılarak

ulaştırılmaktadır. Bu sebeple dağıtım esnasında emtianın dağıtım süresi, ekonomikliği ve güvenli olarak varış noktasına ulaşması önem kazanmaktadır. Birden fazla ulaştırma türünün entegrasyonu ile yapılan ve hızla yaygınlaşan taşımacılıkta; çok türlü transport, intermodal transport (türler arası transport), combined transport (kombine transport) ifadeleri kullanılmaktadır. Çok türlü taşımacılığın en genel özelliği konteyner içinde taşınması ve türler arası taşımacılık yapılırken en az iki taşıma türü kullanılmasıdır. Başka bir tanımla ise bir yükün en az iki ulaşım modu ile ve farklı taşımacılık araçları ile taşınmasını ve taşınması düşünülen yüklerin tek bir yük haline getirilmesi ile taşınabilen, ulaşım modları arası taşımacılık çok türlü taşımacılık olarak anlatılabilir.

1.2.3. İntermodal taşımacılık

Yük üzerinde herhangi bir uygulama yapmadan taşıma türünün değiştirildiği ve yükün herhangi iki nokta arasında birden fazla taşımacılık türü ile taşındığı taşıma yöntemine İntermodal taşımacılık denir. İntermodal taşımacılık ile taşınan ürünlerin yeniden paketlenmesine gerek kalmadan bir taşıma türünden diğer taşıma türüne geçirilmesinde özel taşıma ambalajları kullanılmaktadır. Yükün yükleme aşamasında kapatılarak herhangi bir işlem görmeden alıcıya ulaşması ve orada açılması hedeflenmektedir.

1.2.4. Kombine taşımacılık

Önceleri intermodal taşımacılık şeklinde anlaşılan kombine taşımacılık ise daha sonraları yayımlanan, ECE (Economic Commission for Europe) nin 19 numaralı tavsiye kararı ile kombine taşımacılık enerji harcanan taşımacılık türü ile enerji harcanmayan başka bir taşımacılık türünün birlikte kullanılması ile yapılan taşımacılık türü olarak tanımlanmıştır. Örneğin araç taşıyıcısına yüklenen araçların, Ro-Ro gemisine yüklenen kamyonların veya trene yükleme yapılan kamyonların durumu bu şekilde açıklanabilmektedir. Avrupa Birliği ise kombine taşımacılığı bir nakliye politikası olarak “yükün başlangıç ve bitiş noktalarındaki hareketleri mümkün olduğu kadar az bir şekilde kara yolu taşımacılığı ile yapılmak üzere ürünlerin diğer ulaşım türleri üzerinden taşındığı intermodal taşımacılık türü.” Şeklinde ifade edilmiştir. Söylenişi her ne olursa olsun şu an, çok türlü taşımacılık, intermodal taşımacılık ve kombine taşımacılık terimleri kullanılmaktadır. Öyle görünüyor ki bu tanımların kullanımı yerine daha anlaşılırı konulana ve herkes tarafından benimsenene kadar devam edecektir.

1.3. Gezgin Satıcı Problemi (GSP)

GSP, bir noktadan yola çıkıp eldeki bütün noktalara bir kere uğramak şartı ile başladığı noktaya geri dönmesi ve buna bağlı olarak en kısa turu araştıran problemdir. GSP, tanım olarak kolay ifade edilebilen ve anlaşılması kolay bir problem olmasına rağmen, özellikle uğranacak durak sayısının fazla olduğu durumlarda lineer çözümü oldukça zordur ve zor problemler sınıfındadır. GSP'nin n adet nokta için $n!$ kadar alternatif çözümü olması gerekirken iki nokta arasındaki gidiş ve dönüş mesafeleri eşit olduğu için toplam çözüm sayısı $\frac{1}{2}*(n-1)!$ dir (Lin, 1965).

GSP ilk defa 1800'lü yıllarda İrlandalı matematikçi Sir William Rowam Hamilton ve İngiliz matematikçi Thomas Penygton Kirkman tarafından gündeme getirilmiş ve üzerinde çalışılmaya başlanmıştır. Zaten GSP nin literatür tanımlarında “Çizge üzerinde, en kısa Hamilton turunun bulunması” ifadesi ile sık karşılaşılır. Burada bahsi geçen Hamilton Turu ise, çizge üzerindeki her noktadan sadece bir kez geçen ve başladığı noktada biten, matematikçi William Rowam Hamilton'ın adıyla anılan turdur.

1.4. Lineer Optimizasyon

Günümüzde, işletme, ekonomi ve muhasebe dallarını en yakından ilgilendiren konulardan bir olan Lineer Programlama, aynı zamanda yöneylem araştırmasında da en önemli konulardan biridir. Lineer Programlama, kaynakların optimal dağılımını elde etmeye, maliyetleri minimize, karı ise maksimize etmeye yarayan bir tekniktir.

Lineer Programlama, optimizasyon problemlerinin çözümünde kullanılan bir yöntemdir. 1947' de, George Dantzig, Lineer Programlama problemlerinin çözümünde kullanılan etkin bir yol olan Simpleks Algoritma' yı buldu ve bu buluşla birlikte Lineer Programlama, sıklıkla ve hemen hemen her sektörde kullanılmaya başlandı. Özellikle bankacılık, eğitim sektörlerinde ve askeriyede, optimizasyon problemlerinin çözümünde Lineer Programlama, çok defa kullanılmıştır ve kullanılmaya devam edilmektedir. Firmalarda karşılaşılabilecek darboğazların giderilmesinde, seçenekli üretim tekniklerinin kullanılmasının getirilerini belirlemede, kısıt kaynakların etkin kullanımında ve bunların gölge fiyatlarının belirlenmesinde kullanılacak en uygun politikaların belirlenmesinde kullanılan bir tekniktir. Fortune 500' e üye firmalar arasında yapılan bir araştırma sonucunda, bu

firmaların %85'inin Lineer Programlama yöntemini kullandığı öğrenilmiştir. Lineer Programlama o kadar önemlidir ki, Yöneylem Araştırması kitaplarının çok büyük bir kısmını tek başına kaplar.

Lineer Programlama, değişkenlere ve kısıtlara bağlı kalarak amaç fonksiyonunu en uygun (maksimum ya da minimum) kılmaya çalışır. Temel olarak, Lineer Programlama, kısıt kaynakların optimum şekilde dağılımını içeren deterministik bir matematiksel tekniktir.

1.5. Parçacıklı Sürü Optimizasyon

PSO; kuş ve balık sürülerinin doğadaki davranışlarından esinlenerek geliştirilmiş bir sezgisel yöntemdir. Kuş ve balık sürülerinin doğada yiyecek ararken birbirleri ile iletişim halinde olarak yiyeceği bulmaya çalışmasındaki davranış PSO'nun temelini oluşturmaktadır. PSO da popülasyon içinde parçacığın en iyisi, sürünün en iyisi ve global en iyi bulunmaktadır. Çözüm algoritması her iterasyonda bu üç olgu içerisinde seçim yaparak devam eder.

Örneğin Karınca Kolonisi Algoritmasının dayandırıldığı doğal benzetme yiyecek kaynağından yuvaya dönüşte en kısa yolu arayan gerçek karıncalardır (Dorigo ve Diğerleri, 1997). PSO'da Karınca Kolonisi Algoritmasındaki gibi yiyecek arayan sürülerin yiyecek ararken ki sergilediği davranışlardan esinlenerek geliştirilmiştir.

Sürü zekası belirli bir algoritma veya bir sistem değildir. Sürü zekası doğal veya yapay dağıtılmış, kendi kendine organize sistemlerin, kolektif bir davranış biçimidir. Sürü zekasını baz alarak işleyen algoritmalarla verilebilecek en bilinen örnekler karınca kolonisi algoritması ve parçacıklı sürü optimizasyonu algoritmasıdır. Diğer evrimsel algoritmalar ile karşılaştırıldığında bu algoritmalarda, birbirlerinin davranışlarından etkilenen sürü elemanlarının, bireysel hareket edenlere nazaran çözüm uzayına daha uygun bir şekilde yayıldığı görülmüştür. PSO'da sürü elemanları başlangıçta arama alanlarına yerleştirilir. Her biri mevcut konumundaki objektif fonksiyonu değerlendirir. Her parçacık bazı fonksiyonları birleştirerek arama alanlarını belirler. Her jenerasyonda parçacıklar taşındıktan sonra bireysel ve global değerlerden avantajlı olanlarla yer değiştirilir.

PSO’da tüm parçacıkların kendine ait hızları bulunmaktadır ve bu hızlar parçacıkları diğer parçacıklardan aldıkları bilgilerle optimum sonuç yönünde hızlandırır. Her jenerasyonda bu hızlar önceki en iyi sonuçlardan da yararlanılarak tekrar hesaplanıp güncellenir. Bu işlemler sayesinde popülasyona ait parçacıklar giderek daha iyi pozisyona gelirler.



2. ÖNCEKİ ÇALIŞMALAR

Roberti ve Wen (2016), bu çalışmalarında sera gazı emisyonlarını en aza indirmek için lojistik alanında elektrikli araçların kullanımının arttığı görülmüştür. Ortaya çıkan dağıtım planlama sorunları yeni hesaplama zorlukları ortaya koymaktadır. Zaman Penceleri ile Elektrikli Gezin Satıcı Problemi olarak adlandırılan bir problem ele alınmaktadır. Kısa hesaplama sürelerinde 20 müşteri örneğini ve Genel Değişken Komşuluk Arama ve Dinamik Programlamaya dayalı Üç Fazlı Sezgisel algoritmayı çözebilecek karışık bir tamsayı doğrusal formülasyonu önerilmektedir. Hesaplama sonuçları, sezgisel algoritmanın bir saniyenin onda biri içinde çoğu küçük boyutlu durumda en uygun çözümü bulabildiğini ve 200 müşteriye kadar olan durumlarda mal çözümleri sağladığını göstermektedir.

Dell'Amico ve ark. (2014), bu çalışmalarında bisiklet paylaşım sistemleri, şehir içinde farklı istasyonlarda bulunan halka açık bisikletlerin ortak kullanım için mevcut olduğu bir mobilite hizmeti sunmakta olup ayrıca bu sistemler daha sürdürülebilir bir mobilite elde edilmesine ve araç taşımacılığının yol açtığı trafik ve kirliliğin azaltılmasına katkıda bulunmaktadır. İlk bisiklet paylaşım sistemi 1965'te Amsterdam'da kurulduğundan beri, bu tür uygulamaların sayısı dikkat çekici şekilde artmaktadır ve böylece tüm dünyada yüzlerce sistem çalışmaktadır. Bir bisiklet paylaşım sisteminde, kullanıcılar bir istasyondan bisiklet alabilir, bir yolculuk yapmak için kullanabilir ve daha sonra sistemde herhangi bir istasyonda bisikleti bırakabilir. Bu davranış tipik olarak, bazı istasyonların dolu ve diğerlerinin boş olduğu bir duruma yol açar. Bu nedenle dengeli bir sistem, bisikletlerin istasyonlar arasında yeniden dağıtılmasını gerektirir. Bu çalışmada toplam maliyeti en aza indirmek amacıyla bisikletleri yeniden dağıtmak için yoğunlaştırılmış araç filosunun kullanıldığı Bisiklet paylaşım Dengeleme Sorunu (BRP) ele alınmaktadır. Bu, özel bir emtia teslim alma ve teslim etme kapasitesine sahip araç yönlendirme problemi olarak görülebilir. Bu problemin dört karma tamsayılı doğrusal programlama formülasyonunu sunulmaktadır. Önerilen formülasyonların üssel bir sınırlama içerdiğini, bu nedenle, özel olarak hazırlanmış dal ve kesim algoritmalarının bunları çözmek için geliştirildiğini belirtmek gerekir. BRP'nin matematiksel formülasyonları ilk önce İtalya'daki Reggio Emilia şehri için elde edilen veriler kullanılarak hesaplanmaktadır. Bu hesaplama çalışması daha sonra dünyanın diğer yerlerinden bisiklet paylaşım sistemlerini içerecek şekilde genişletilmektedir. Çalışmadan elde edilen bilgiler, web üzerinde herkese açık olarak

sunulan BRP için bir dizi referans örneği oluşturmak için kullanılmaktadır. Bu çalışmada sunulan dal ve kes algoritmalarının kapsamlı deneyi yapılmakta ve önerilen matematiksel formülasyonların ilginç bir hesaplama karşılaştırması sunulmaktadır. Son olarakta problemin hesaplama zorluğu ile ilgili çeşitli görüşler vurgulanmaktadır.

Brunner (2014), bu çalışmasında matematiksel programlama teknikleriyle bir Yer Gecikme Programı (GDP) verildiğinde, yeniden planlama problemini havayoluyla ele alınmakta ve amaç gecikme önlemlerini, mürettebat ve yolcu yanlış bağlantılarının maliyetini ve çeşitli kısıtlamalara tabi uçuş iptal maliyetlerini en aza indirmektir. Tüm hedefleri içeren yeni bir doğrusal tamsayı modeli sunulmaktadır. Gerçek hayattaki ve rastgele verileri kullanarak, standart yazılımla çözülen modeli değerlendirmek için kapsamlı hesaplamalar sunulmaktadır. Yüksek kaliteli çözümler hızlı bir şekilde, yani saniyeler içinde bulunmaktadır. Yanlış bağlantılar ve iptaller için maliyet değerlerinin ayarlanması yeni slot atamalarının üzerindeki önemli etkisini gösterilmektedir.

Tamannaei ve Rasti-Barzoki (2019), bu çalışmalarında yeni bir entegre tedarik zinciri çizelgeleme ve araç rotalama problemi geliştirilmekte ve çalışmada araçların sabit maliyetleri ve ağırlık seyahat maliyetleri bakımından toplam ağırlıklı gecikme ve nakliye maliyetlerini en aza indirmek hedeflenmektedir. Karışık bir tamsayılı doğrusal programlama ve Dal ve Sınır (B & B) algoritmasına dayalı kesin bir prosedür ve bu problemi çözmek için bir metasezgisel Genetik Algoritma (GA) gibi iki çözüm yaklaşımı önerilmektedir. Hesaplamalı deneyler hem küçük hem de büyük ölçekli analizler için yapılmaktadır. Küçük ölçekli analizlerin sonuçları, önerilen B & B algoritmasının, hem en iyi şekilde çözülmüş problem sayısı hem de çalışma süreleri bakımından, CPLEX yazılımına kıyasla daha verimli bir performans sağladığını göstermektedir. Sonuçlar, gerçek hayattaki büyük ölçekli problemlerin etkin bir şekilde çözülmesinde meta-sezgisel GA'nın kabiliyetini göstermektedir.

Elloumi ve ark. (2014), bu çalışmada PSO'nun son yıllarda çok popüler olduğu ve Parçacık sürüsü optimizasyonu (PSO) ve Arı kolonisi optimizasyonu (ACO) algoritmalarının, karmaşık optimizasyon problemlerini çözmedeki basitlikleri ve etkin oluşları nedeniyle araştırmacıların ilgisini çektiği belirtilmektedir. Hem ACO hem de PSO, Gezgin Satıcı Problemini (TSP) çözmek için başarıyla uygulanmaktadır. Konvansiyonel PSO algoritmasının performansı orta boyutlu ve arama alanlı küçük problemler için çok

tatmin edici olmakla beraber mekan daha karmaşık hale geldikçe, geleneksel yaklaşımlar aramada daha kötü çözümler sunma eğilimindedir. Bu makale performansı artırmak için ACO algoritması ile sentezlenmiş bir PSO'yu tanıtarak yeni bir yaklaşım sunmaktadır. Bu yeni yaklaşımın performansı, TSP kriter ve doğrulanma süresi ve en iyi uzunluk dikkate alınarak ampirik sonuçlar kullanılarak doğrulanmakta ve önerilen yöntemin yeterli olduğu görülmektedir.

Funke ve Kopfer (2016), bu çalışmalarında Çok boyutlu Karasal Konteyner Taşıma Probleminde (mICT), çeşitli toplama ve teslimat konumlarına sahip yollar boyunca bir seferde iki adet 20 fit veya bir adet 40 fit konteyner alabilen kamyonlar ele alınmaktadır. mICT probleminde toplam sürüş mesafesinin en aza indirilmesi ve kamyonların toplam çalışma süresinin en aza indirilmesi gibi iki alternatif amaç işlevi için karma tamsayılı bir doğrusal program kullanılarak sunulmaktadır. Sunulan model, büyüklük bakımından değişken olan durumlar üzerinde test edilmektedir. Hesaplamalı deneyler, sunulan model sayesinde küçük sorunlu durumların en iyi şekilde çözülebileceğini göstermektedir.

Yuan ve ark. (2019), bu çalışmalarında stokastik talep ve hizmet düzeyini göz önünde bulundurarak ve stratejik tasarım ve işletme kararlarını entegre bir şekilde birleştirerek, entegre ve uzun vadeli bir bisiklet paylaşım sisteminin (BSS) en uygun şekilde planlanmasını bir bakış açısıyla incelemektedirler. Mevcut araştırma çabasının metodolojik katkısı, toplam filo büyüklüğü tasarımı, depo yeri tasarımı, yeniden dengeleme ve bakım planları birleştirilmesi ve birlikte çözülebilmesi, bisiklet istasyonlarının sayısı, yeri ve kapasitesi gibi birçok alt problemin için birleşik bir karışık tamsayılı doğrusal programlama (MILP) modeli önermektir. Stokastik taleple başa çıkmak için senaryo amaçlı bir yaklaşım uygulanmaktadır. Ayrıca, bisikletlerin kapsama alanını karakterize etmek için subjektif mesafe kavramı önerilmektedir, bunlar bisikletlerin kullanılabilirlik oranı ile birlikte servis seviyesinin ölçütleridir. Pekin'de bir vaka incelendi ve anahtar parametreler için duyarlılık analizleri yapıldı. Önerilen yaklaşımın pratik değerini göstermek için, önerilen MILP modelinin çözdüğü planlama sonuçları ile BSS istasyon düzenine ilişkin gerçek hayatta bir karşılaştırma yapılmaktadır. Sonuçlardan, operatörün karşıladığı maliyetler ile kullanıcılar için hizmet seviyesi arasında bir denge sağlanabileceğini gözlemlenmektedir. İstasyonlar kapsama alanını artırarak hizmetin rahatlığında artıracak şekilde çalışma alanı boyunca eşit olarak dağıtılmaktadır.

Geliştirilen model BSS operatörleri tarafından planlama kararları için kullanılabileceği belirtilmektedir.

Chen ve ark. (2017), bu çalışmalarında Gezgin Satıcı Problemi'nin (TSP) en iyi bilinen birleşimsel optimizasyon sorunlarından biri olduğundan dolayı araştırmacıların ilgisini çektiğini belirtmektedirler. Birçok çalışma, iki boyutlu TSP'yi çözmek için çeşitli yöntemler önermektedir. Bu çalışmada, iki boyutlu TSP üç boyutlu TSP'ye, yani tüm noktaların (şehirlerin) ve yolların (çözümlerin) bir kürenin yüzeyinde bulunduğu küresel TSP'ye genişletilmektedir. Luciferin taşıyıcılarının ateşböceği stolu kenarlarından şehirler arasında dönüştürüldüğü ve olasılıksal formül ile luciferin güncelleme formülünün değiştirildiği Ateşböceği Sürüsü Optimizasyonuna (GSO) ve tamamlayıcı 2-opt algoritmasına dayanan bir hibrid algoritması önerilmektedir. Buna ek olarak, seçilen optimum rotaları optimize etmek için 2-opt algoritmasının tamamı gerçekleştirilir. Sayısal deney sonuçları, önerilen algoritmanın küresel TSP'yi çözmede temel GSO'dan daha iyi bir performansa sahip olduğunu göstermektedir. Bu arada, tam 2 optik algoritması yakınsama hızını hızlandırabilir.

Bertazzi ve ark. (2019), bu çalışmalarında pratik Çok Depolu Envanter Rotalama Problemini (MDIRP) çözmektedirler. Farklı depolardan müşterilere nasıl hizmet edileceğini belirleyerek ve stokların tükenmesini önlemek için envanter seviyelerini yöneterek yönlendirme maliyetlerini en aza indiren MDIRP, envanter ve yönlendirme kararları arasındaki dengeyi bütünleşik bir şekilde optimize eder. Bu problem karma bir tamsayı doğrusal programlama modeli olarak formüle edilmekte ve sorunu çözmek için üç fazlı bir matematik tasarlanmaktadır. Meta sezgiselin çözümleri, klasik IRP örneklerinde dallanma ve kesme algoritmasından, yeni örneklerden ve çok etkili olduğunu gösteren gerçek bir vaka çalışmasından elde edilen sonuçlarla karşılaştırılmaktadır.

Marinakis ve ark. (2019), bu çalışmalarında Zaman Pencere Araç Yönlendirme Probleminin (VRPTW) çözümü için Parçacık Sürü Optimizasyonu (PSO) algoritmasının yeni bir çeşidi sunulmaktadır. Önerilen Çok-Adaptif Parçacık Sürüsü Optimizasyonu (MAPSO) algoritmasında üç farklı adaptif strateji kullanılmaktadır. İlk uyarlamalı strateji, Açgözlü Randomize Uyarlamalı Arama Prosedürünün (GRASP) kullanımı ile ilgilidir. İlk çözümler üretildiğinde ve algoritmanın tekrarları sırasında yeni bir çözüm üretildiğinde uygulanır. İkinci uyarlama stratejisi, yeni bir uyarlama stratejisi olan Adaptif

Kombinatorial Komşuluk Topolojisi'nin kullanıldığı, parçacıkların bir çözümden diğerine hareketindeki uyarlanabilirlik ile ilgilidir. Son olarak, Parçacık Sürüsü Optimizasyonu algoritmasının tüm parametrelerinde bir uyarlanabilirlik vardır. Algoritma, parametrelerin rasgele değerleri ile başlar ve bazı koşullara dayanarak, tüm parametreler yinelemeler sırasında uyarlanır. Algoritma iki klasik referans noktası setinde test edilmektedir; bunlardan biri 100 düğümlü 56 örnek ve diğeri 200 ila 1000 arasında değişen düğüm sayısı ile 300 örnek içermektedir. Algoritma PSO'nun diğeri sürümleriyle ve literatürdeki en iyi performans gösteren algoritmalarla karşılaştırılmaktadır.

Chen ve Shi (2019), bu çalışmalarında modern lojistik için “son mil” teslimat zorluğunu yansıtan kentsel dağıtımdan kaynaklanmakta olan Zaman Pencereci Çok Bölmeli Bir Araç Rotalama Problemi (MCVRPTW) tanıtılmaktadır. MCVRPTW'un ayırt edici özelliği, ürünlerin bağımsız araç bölmelerinde taşınması gerektiğidir çünkü özellikleri farklılıkları nedeniyle birlikte karıştırılamazlar. Optimizasyon problemi için oluşturulan matematiksel formülasyonu temel alarak iki çözüm yaklaşımı önerilmektedir ve bunlardan biri benzetilmiş tavlama ile hibrit parçacık sürüsü optimizasyonu (HPSO), diğeri ise geleneksel partikül swarm optimizasyonudur (PSO). Önerilen çözüm yaklaşımlarının performansını araştırmak için Süleyman'ın araç yönlendirme problemine bağlı olarak zaman pencereleri (Solomon, 1987) 25 müşteri, 50 müşteri ve 100 müşteri ile deneysel örnekler geliştirildi. Sonuçlar, her iki yaklaşımın da MCVRPTW için oldukça verimli olduğunu göstermektedir. Dahası, HPSO algoritması, özellikle tüm durumlar için en iyi çözümü sağlamada genel olarak daha iyi bir performansa sahiptir ve HPSO algoritması, sorun boyutu arttıkça PSO algoritmasından daha verimli olur. Öte yandan, PSO algoritması en kötü çözüm ve standart sapma açısından hafif bir kenar göstermektedir.

Avraamidou ve Pistikopoulos (2019), bu çalışmalarında İki farklı karar seviyesindeki iki karar vericiyi içeren optimizasyon problemlerine iki seviyeli programlama problemleri olarak belirtilmektedir. Bu çalışmada, iki sınıflı iki seviyeli programlama probleminin kesin ve global çözümü için yeni algoritmalar sunulmaktadır. Yani (a) her iki optimizasyon seviyesinde hem tamsayı hem de sınırlanmış sürekli değişkenleri içeren iki seviyeli karma tamsayı doğrusal dış programlama problemleri (B-MILP) ve (b) iki seviyeli karma tamsayı dışbükey kuadratik programlama problemleri (B MIQP). Çoklu parametrik programlama teorisine dayanıldığında, üst seviye problemin optimizasyon değişkenleri alt seviye için sınırlanmış parametreler olarak kabul edilerek, ana fikir alt seviye problemi

multi-parametrik bir programlama problemi olarak tekrar etmektedir. Kesin çok parametrik karma tam sayı doğrusal ve kuadratik yöntemlerden elde edilen çözümler daha sonra, tek seviyeli, bağımsız, deterministik karışık tam sayı optimizasyon sorunlarının bir kümesi olarak çözülebilen üst düzey sorunla değiştirilir. Hem alt hem de üst seviyelerde sağ taraf belirsizliği dahil olmak üzere sorunların uzaması da tartışılmaktadır. Son olarak, hesaplamalı uygulama ve çalışmalar test problemleriyle sunulmaktadır.

Cao ve ark. (2018), bu çalışmalarında Dinamik Optimizasyon Problemlerine yönelik yeni bir Parçacık Sürüsü Optimizasyonu algoritması sunulmaktadır. Algoritma, parçacıkların keşif ve kullanma yeteneklerini arttırmak için Parçacık Sürüsü Optimizasyonunun hız güncellemesine bir komşu temelli öğrenme stratejisi içerir. Geleneksel sürü güncelleme şemasının aksine, sürüyü güncellemek için "en kötü değiştirme" stratejisi kullanılır, böylece sürüdeki en kötü parçacığın konumu daha iyi yeni oluşturulan bir pozisyon ile değiştirilir. Kısa süreli hafıza, en sık kullanılan ortamdan ara zindeliğe sahip çözümleri saklamak için kullanılır ve uzun süreli hafıza, önceki tüm ortamlarda bulunan en iyi tarihi çözümleri saklamaktır. Çevresel bir değişim tespit edildikten sonra, sürünün içindeki bazı parçacıklarının konumları kısa süreli belleğin üyeleri tarafından değiştirilir ve mevcut ortam altında uzun süreli bellekteki en iyi üye Gauss komşusu aktif sürüme eşlik eder ve sonra kalan parçacıkların konumları yeniden başlatılır. Önerilen algoritmanın performansı, Hareketli Tepeler Kıyaslaması problemleri ve Dinamik Dönme Zirvesi Kıyaslama Üretici üzerindeki en son altı dinamik algoritma ile karşılaştırılmaktadır. Deneysel sonuçlar, algoritmanın rakiplerle karşılaştırıldığında üstün performans elde ettiğini göstermektedir.

Murakami (2017), bu çalışmasında elektrikli araçların son zamanlarda çevresel ve finansal etkileri göz önüne alındığında büyük ilgi gördüğü belirtilmektedir. Bu çalışmada elektrikli ve dizel motorlu taşıtları hedef alan bir taşıt rotalama problemi olan elektrikli ve dizel motorlu taşıt rotasyonu üzerinde durulmaktadır. Orijinal grafik kavramı, yol eğimlerini, araç hızını ve ivmesini, trafik ışıklarında beklemeyi ve araç gövdesini ve bakım maliyetlerini göz önünde bulundurmak için yeni bir EDVRP modeli önerilmekte olup ve dahası, EDVRP'yi beş geçerli eşitsizliği tanıtarak ve sorunu karma bir tamsayı doğrusal programlama (MILP) formülasyonu olarak formüle edilmektedir. Simülasyonlarda, EV'lerin hangi koşullara sokulması gerektiğini ve EVS'yi tanıtmanın etkisini gözlenmektedir. Ayrıca simülasyonların sonuçları, yeni modelin gerçek dünyadaki faydasını göstermektedir.

Leggieri ve Haouari (2017), bu çalışmalarında Yeşil Araç Rotalaması (GVRP), çevre dostu araçlarla dağıtım faaliyetlerinin geliştirilmesi ve analizi ile ilgilenen aktif bir araştırma alanı olarak belirtilmekte ve GVRP için karma tamsayılı doğrusal bir formülasyon ve bir azaltma prosedürü kullanan pratik bir çözüm yaklaşımı önerilmektedir. Zaman süresi sınırları ve enerji tüketimi kısıtlamaları olan yeşil bir araç rotalama problemini ele alınmaktadır. Bu yeni formülasyon kompaktlık ve esneklik gibi iki önemli avantaj sunmaktadır. Formülasyon ve azaltma prosedürünün, genel amaçlı bir çözücü kullanarak orta büyüklükteki durumlar için en uygun çözümleri elde etmeyi sağladığına dair ampirik kanıtlar sunulmaktadır. Önerilen kesin yaklaşımın, en son teknolojiye sahip bir dal ve kesim algoritmasını sürekli olarak geride bıraktığını ve GVRP'lerin optimal şekilde çözülmesi için çekici ve pratik bir alternatif oluşturduğunu gösterilmektedir.

Zhang ve ark. (2014), bu çalışmalarında, Parçacık Zeka Optimizasyonu'nun (PIO) günümüzde önemli bir optimizasyon aracı haline geldiği ve yaygın olarak birçok uygulama alanında kullanıldığı ancak başarılı uygulamaların aksine, temelde teorik olarak oldukça zayıf olduğundan dolayıda hala çözülmesi gereken birçok sorununun bulunduğu belirtilmektedir. Birinci problem, algoritmanın performansını sonlu bir sürede nasıl ölçeceğinizi, yani algoritmanın pratik problemler için elde ettiği çözüm kalitesini nasıl değerlendirebileceğidir. Pratik problemlerde uygulama büyük ölçüde sınırlandırılmaktadır. Bu çalışmada akıllı optimizasyon için arama alanının analizine ve algoritmanın karakteristiğine dayanan deneysel bir analiz yöntemi önerilmektedir. “Değer performansı” yerine “sıralı performans” bu yöntemde değerlendirme kriteri olarak kullanılır. Uygun çözümler, çözüm numunelerini birkaç parçaya bölmek için mesafeye göre kümelenir. Daha sonra, kümelenme sonuçlarına göre çözüm alanı ve “yeterince iyi” küme ayrıştırılmaktadır. Son olarak, göreceli istatistik bilgisi kullanılarak değerlendirme sonucu elde edilmektedir. Önerilen yöntemi kıyaslamak için, Karınca Kolonisi Optimizasyonu (ACO), Parçacık Sürüsü Optimizasyonu (PSO) ve Yapay Balık Sürüsü Algoritması (AFS) gibi bazı akıllı algoritmalar seyahat eden satıcı sorununu çözmek için ele alınmaktadır. Hesaplamalı sonuçlar önerilen yöntemin uygulanabilirliğini göstermektedir.

Mavrovouniotis ve Yang (2017), bu çalışmalarında Karınca Kolonisi Optimizasyonu, Parçacık Sürüsü Optimizasyonu, Arı Esinli Algoritmaları, Bakteriyel Yiyecek Arama Optimizasyonu, Ateş Böceği Algoritmaları, Balık Sürüsü Optimizasyonu ve daha birçoklarını içeren Sürü Zekası (SI) algoritmaları, durağan ortamlarda zor optimizasyon

problemlerini ele almak için iyi yöntemler olduğu kanıtlandığı belirtilmektedir. Çoğu SI algoritması, durağan optimizasyon problemlerini ele almak için geliştirilmiş olup ve bu nedenle (neredeyse) optimum çözüme verimli bir şekilde yaklaşabilirler. Bununla birlikte, birçok gerçek dünya problemi zamanla değişen dinamik bir ortama sahiptir. Bu tür dinamik optimizasyon problemlerinde (DOP'ler), geleneksel bir SI algoritmasının, algoritma bir çözüm üzerinde birleştiğinde, optimum değişimi takip etmesi zor olmaktadır. Son yirmi yılda, adaptasyon yetenekleri nedeniyle SI algoritmaları kullanarak DOP'ları ele alma konusunda artan bir ilgi bulunmaktadır. Bu makale, ayrık, sürekli, kısıtlı, çok amaçlı ve sınıflandırma problemleri ve gerçek dünya uygulamaları gibi çeşitli problem sınıflarına odaklanan SI dinamik optimizasyonu (SIDO) hakkında geniş bir derleme sunmaktadır. Ek olarak, bu makale, SIDO'da kullanılan dinamik değişimleri, performans ölçümlerini ve kıyaslama jeneratörlerini ele almak için SI algoritmalarına entegre edilmiş geliştirme stratejilerine odaklanmaktadır.

Gupta ve ark. (2016), bu çalışmalarında metro demiryolu ağlarında enerji tasarruflu tarifeleri hesaplamak için yeni bir iki aşamalı doğrusal optimizasyon modeli önerilmektedir. Elde edilen zaman çizelgesi, tüm trenler tarafından tüketilen toplam enerjiyi en aza indirir ve demiryolu ağındaki kısıtlamalara bağlı olarak, frenleme trenleri tarafından üretilen rejeneratif enerjinin kullanımını en üst düzeye çıkarır. NP gibi zor olan diğer mevcut modellerin aksine, çalışma modeli hesaplamalı olarak en izlenebilir olan bir doğrusal programdır. Optimizasyon modeli, binlerce aktif trenle bir günlük (18 saat) tam hizmet süresine sahip Shanghai Metro Şebekesinin 8 numaralı hattının PES2 SFM2 servis farklı hizmetlerine uygulanmaktadır. Her durumda, model çok hızlı bir şekilde en uygun zaman çizelgesini bulmaktadır (en büyük çalışma süresi 13 saniyeden azdır). Etkin enerji tüketiminde belirgin düşüş olmaktadır (en kötü durum % 19,27'dir). Modele dayalı kod daha sonra dünya çapında iletişim tabanlı tren kontrol sistemlerinin en büyük kurulum tabanına sahip olan Thales Timetable Compiler ile bütünleştirilmektedir.

Marinakis (2015), bu çalışmasında ayrık optimizasyon problemlerine uygun Parçacık Sürü Optimizasyonu (PSO) algoritmasının yeni bir versiyonu sunulmakta ve yoğunlaştırılmış lokasyon yönlendirme probleminin çözümü ve stokastik taleplerle lokasyon yönlendirme probleminin yeni bir formülasyonu için uygulanmaktadır. Önerilen algoritma, bir daralma PSO algoritmasına dahil edilen üç farklı topolojiyi birleştirmekte ve bu nedenle, çok etkili bir yeni algoritma olan global ve yerel kombinasyonel genişleyen komşu topoloji PSO

geliştirilmektedir. Algoritma, Stokastik talepler ile probleme uygun olması için ilk olarak, ayrık talepler ile kapasitanslı lokasyon güzergâhı problemi için üç klasik kıyaslama örneği setinde test edilmektedir. Daha sonra, stokastik taleplerle lokasyon güzergâhı problemi için kıyaslama örnekleri olmadığından, bu numuneler sırayla uygun şekilde dönüştürülmektedir. Algoritma, bu dönüştürülmüş kıyaslama örnek kümeleri kullanılarak stokastik taleplerle ilgili problemde test edilmektedir. Ayrıca algoritma, PSO'nun bir dizi farklı uygulaması ile ve lokal yönlendirme problemi için ayrı ve stokastik taleplerle ilgili literatürdeki meta-sezgisel, evrimsel ve doğadan esinlenen algoritmalar ile karşılaştırılmaktadır.

Garzón-Garnica ve ark. (2015), bu çalışmalarında Büyük Ölçekli Kapasiteli Araç Rotalama Problemi (CVRP) için Otomatik Veri Toplama (ADA) prosedürünü sunulmaktadır. Çalışmada problem olarak Meksika otomotiv kümesinden bir üretim işletmesinde personel taşımacılığı için en uygun yolları bulmak ele alınmaktadır. Her CVRP'de olduğu gibi, veri miktarı katlanarak artmakta buda veri toplamayı zorlaştırmaktadır. 27 düğümlü küçük ölçekli bir model 729 çift, 244 düğümlü orta ölçekli bir model 59292 çift ve 728 düğümlü tam ölçekli bir model 529256 çift matris üreten üç rotalı bir örnek problem çözülmektedir. Orta ve büyük ölçekli modeller için veri matrisi elde edilmekte olup ve bu veriler gelecekteki araştırmalar için bir temel olarak kullanılabileceği belirtilmektedir. Doğrusal programlama yöntemleri kullanılarak bir örnek problem çözülmektedir. Küçük ölçekli model için elde edilen çözüm, üretici firma ampirik çözümüne göre %30' luk bir seyahat süresinde azalma olduğu gözlemlenmektedir.

Lei ve ark. (2014), bu çalışmalarında multimodal taşımacılık ve Sürü Zeka (SI) algoritması ile ilgili bazı temel kavramlar tanımlanmış ve multimodal ulaşım şeması kararı ve swarm zeka yöntemleri uygulama alanlarıyla ilgili literatürler incelenerek analiz edilmektedir. Daha sonra, bu çalışma ulaştırma maliyetlerine, ulaştırma süresine ve ulaştırma risklerine dayalı bir çok modlu bir ulaştırma planı karar optimizasyonu matematiksel modeli oluşturarak ilgili parametreleri ve modelin kısıtlarını detaylı bir şekilde açıklamakta ve çok amaçlı optimizasyon problemlerini tek bir hedef optimizasyon ulaşım planı karar problemine dönüştürmek için ağırlık katsayısını kullanmaktadır. İlk kez multimodal taşıma şeması kararının birleşik optimizasyon problemini çözmek için Parçacık Sürüsü Optimizasyonu (PSO) algoritması ve Karınca Kolonisi (ACO) Algoritmasının birleştirilmesiyle oluşturulan ve PSO ve ACO algoritmalarının avantajlarını etkili bir

şekilde taşıyan algoritma (PSACO) önerilmektedir. Çözümlerde PSACO algoritmasının iki algoritmanın avantajına sahip olduğunu ve kendi sorunlarını telafi ettiğini göstermektedir. Ayrıca PSACO algoritması zaman verimliliğinde ACO algoritmasından daha iyi olduğu görülmektedir. Doğruluğu multimodal taşıma şeması kararıyla ilgili sorunu çözmek için etkili bir sezgisel algoritma olduğu kanıtlanan PSO algoritmasından daha iyidir ve ekonomiktir. PSACO ulaştırma karar vericileri için makul ve güvenli ulaştırma planı referansı olabileceği belirtilmektedir.

Chiang ve ark. (2014), bu çalışmalarında arızalı ürünler için aşamalı ters lojistik kursu tasarlanmakta, böylece alt taraftaki ortaklarda üretilen hasarlı ürünlerin, yeniden işleme ve bakım için bir tedarik zincirinin aşamaları boyunca doğrudan üst taraftaki ortaklara iade edilebilmesi sağlanmaktadır. Bu ters tedarik zinciri tasarım problemini çözmek için, optimal çapraz aşamalı ters lojistik matematiksel model geliştirilmektedir. Ek olarak, bu matematiksel modeli destekleyecek çözümler bulmak için geliştirilen Genetik Algoritma (GA) ve Atalet Ağırlık Yöntemi (PSOA IWM), VMax Yöntemi (PSOA VMM) ve Daraltma Faktörü Yöntemi (PSOA CFM) gibi üç Parçacık Sürüş Optimizasyonu (PSO) algoritması kullanılmaktadır. Bu çalışmada önerilen modeli doğrulamak için kullanılan dört algoritmanın yürütme zamanlarını, yakınsama zamanlarını ve objektif fonksiyon değerlerini karşılaştırmak için farklı kapsamlara sahip gerçek bir vaka ve beş simülasyon durumu kullanılmaktadır. Sistem yürütme süresiyle ilgili olarak, GA diğer üç PSO'dan daha fazla zaman harcadığı ve objektif fonksiyon değeri ile ilgili olarak GA, PSOA IWM ve PSOA CFM, PSOA VMM'den daha düşük bir yakınsama değeri elde edebildiği gözlemlenmektedir. Son olarak PSOA IWM, PSOA VMM, PSOA CFM ve GA'dan daha hızlı bir yakınsama hızı gösterdiği belirtilmektedir.

Ma ve ark. (2017), bu çalışmalarında Otonom araç (AV) teknolojisinin geleneksel araç paylaşım sistemlerinin verimliliğini artırmak için büyük umut vaat ettiği belirtilmekte ve özerk araç paylaşımı ve rezervasyonu (AVSR) olarak adlandırılan AV'leri kullanan yeni bir araç paylaşım sistemini araştırılmaktadır. Böyle bir sistemde, yolcular vaktinden önce AV gezileri talep edebilir ve AVSR sistem operatörü bu isteklere göre AV toplama ve teslimat programları ile AV açma zincirlerini en iyi şekilde ayarlayacaktır. AV tur zincirleri için optimum çözümler ve inşa edilen AVSR ağları üzerinden gerekli filo boyutu için etkin bir şekilde çözmek için doğrusal bir programlama modeli önerilmektedir. Vaka çalışmalarının AVSR'nin araç kullanım oranını (VUR) önemli ölçüde artırabileceğini ve

sonuç olarak araç sahipliğini önemli ölçüde azaltabileceğini gösterdiği belirtilmektedir. AVSR sistemlerinde kaçınılmaz boş taşımalara rağmen, AVSR sistemlerinde harcanan gerçek araç millerinin (VMT), geleneksel taksilere göre önemli ölçüde fazla olmadığı tespit edilmektedir. Sonuçlar mevcut ulaşım sistemlerimizin mobilitesini ve sürdürülebilirliğini arttırmada AVSR sistemlerinden büyük potansiyel faydalar sağlayacağı anlamına gelmektedir.

Zhong ve ark. (2018), bu çalışmalarında en popüler Parçacık Zeka algoritmalarından biri olan Parçacık Sürüş Optimizasyonu (PSO) algoritması, çok sayıda sürekli ve kesikli optimizasyon problemine geniş çapta çalışılmakta ve uygulanmaktadır. Çalışmada Gezgin Satıcı Problemi (TSP) için Benzetilmiş Tavlama algoritmasının kabul kriterini kullanan ayrı bir kapsamlı öğrenme PSO algoritması önerilmektedir. TSP problemi için hem her bir parçacığın kişisel olarak en iyisini hem de eldeki problemin özelliklerini öğrenebilen yeni bir uçuş denklemi tasarlanmaktadır. Hızın etkinliğini arttırmak için her boyutta ancak ihtiyaç duyulduğunda hesaplanan tembel hız önerilmektedir. Çözüm hız bileşeni uygulandıktan sonra her bir ara çözüm değerlendiren istekli değerlendirmesi, çözüm alanını daha iyi araştırmak için önerilmektedir. Erken yakınsamadan kaçma kabiliyetini arttırmayı hedefleyen parçacık, yeni üretilmiş çözümleri kabul edip etmeme konusunda karar vermek için Metropolis kabul kriterini kullanır. Yeni uçuş denkleminin avantajını göstermek, yeterli çeşitliliği sağlamak için açgözlü olmayan kabul stratejisini kullanma zorunluluğunu doğrulamak ve tembel hızı ve istekli hızı karşılaştırmak için sistematik deneyler yapılmaktadır. Çok çeşitli kıyaslama TSP problemleri üzerinde yapılan karşılaştırma, önerilen algoritmanın diğer birçok modern algoritma ile daha iyi veya rekabetçi olduğunu göstermektedir.

Hungerländer ve Truden (2018), bu çalışmalarında Zaman Pencereli NP Zorlu Gezgin Satıcı Problemi (TSPTW), belirli bir amaç fonksiyonunu en aza indirecek şekilde, belirli bir müşteri setini kendi atanmış zaman pencereleri içinde ziyaret etmekle ilgilidir. Her müşterinin kendi zaman penceresini belirlediği geleneksel problemlerin aksine, modern web tabanlı sistemlerde tedarik şirketi müşterinin ondan birini seçebileceği bir zaman penceresi tanımlanır. Bu nedenle, tasarım gereği, tipik olarak birkaç müşteri aynı zaman penceresine atanır. Bu gelişme ve uygulayıcıların kolay ve hızlı bir şekilde uygulanabilecek formülasyonlar aramaları gerçeği ile motive edilmiş, asimetric TSPTW için zaman pencerelerinin yapısını hesaplamalı olarak kullanmaya izin veren ve üçgen

eşitsizliklerin dayanmadığı asimetrik seyahat süreleri için aynı zamanda uygulanabilir olan iki karma-tamsayılı doğrusal programı (MILP'ler) sunulmaktadır. Özellikle, MILP'lerimizdeki ikili değişkenlerin sayısını azaltmak için zaman pencereleri arasındaki ilişkileri analiz edilir ve kullanılır. Örtüşmeyen zaman pencerelerinin özel durumu için sınırlama kümesini daha da basitleştirebilir ve ayrıca gereken sürekli değişken sayısını azaltabileceği belirtilmektedir. Son olarakta MILP'lerin, bir çevrimiçi alışveriş uygulamasıyla ilgili referans örneklerinde verimliliğini kanıtlanmaktadır.

Norouzi ve ark. (2015), bu çalışmalarında rekabetçi bir ortamda ortaya çıkan periyodik araç rotalama probleminin (PVRP) verimliliğini ölçmek ve değerlendirmek için yeni bir matematiksel model sunulmaktadır. Rekabetçi ortamda, birçok distribütör rekabet içindedir ve diğer rakiplerden daha erken müşterilere ulaşması satış miktarını etkilemektedir. Ana hedefler seyahat maliyetini en aza indirmek ve diğer rakip distribütörlerden önce müşterilere hizmet vermek için satışı en üst düzeye çıkarmaktır. Sorunun etkinliğini ölçmek için geliştirilmiş bir parçacık sürüsü optimizasyonu (IPSO) algoritması önerilmekte ve doğruluk açısından orijinal bir PSO algoritması ile karşılaştırılmaktadır. Hesaplamalı sonuçlar, geliştirilmiş PSO algoritmasının doğruluk açısından iyi çalıştığını ancak orijinal PSO'nun hesaplama zamanında daha iyi performans gösterdiği belirtilmektedir.

Li ve Li (2016), bu çalışmalarında senaryo temelli karar verme problemleri (örneğin stokastik programlama senaryoları) için en büyük zorluklardan biri, senaryo sayısı arttıkça problemin karmaşıklığının hızla artmasındandır. Senaryo azaltma, çok sayıda senaryo kullanılmasının çözüm kalitesini korurken, hesaplama karmaşıklığını önemli ölçüde azaltmak için, çok sayıda karar verme tahmin senaryosunu temsil etmek için az sayıda senaryo seçmeyi amaçlamaktadır. Bu çalışmada, taşıma mesafesinin minimize edilmesine dayanan, hesaplamalı olarak verimli yeni bir senaryo indirgeme algoritması önerilmektedir. Önerilen algoritma, doğrusal programlama problemlerinin çözülmesine dayanmaktadır. Senaryo alt kümesi güncelleme aşaması ve olasılık tahsisi aşaması taşıma mesafesi bir araya gelinceye kadar tekrarlı bir şekilde gerçekleştirilmektedir. Mevcut senaryo azaltma yöntemleriyle karşılaştırılması, önerilen yöntemin büyük senaryo setinin azaltılmasında çok etkili olduğunu ortaya koymaktadır. Stokastik optimizasyon problemleri üzerine yapılan uygulama çalışmaları da önerilen yöntemin etkinliğini göstermektedir.

69Saud ve ark. (2018), bu çalışmada popüler ve klasik rota planlama problemlerinden biri olan gezgin satıcı probleminin (GSP) analizlerinde kullanılan farklı tipte ve optimizasyon sorunlarını analiz etme ve en uygun çözümü bulma becerisine sahip akıllı meta-sezgisel Yapay Zeka (YZ) tekniklerinin performansları sunulmaktadır. Bunlar yeni bir operatör ile Geliştirilmiş Parçacık Sürüsü Optimizasyonu (IPSO), Karışık Kurbağa Sıçrama Algoritmaları (SFLA), Karınca Kolonisi Optimizasyonu (ACO) ve çaprazlama ve mutasyon operatörleri kullanarak modifiye edilmiş sıçrayan kurbağa algoritmalarıdır (OX-IMSFLA) ve (CX-IMSFLA). Çalışmada TSPLIB kıyaslama problemleri üzerinde test edilmekte ve test sonuçları birbirleriyle karşılaştırılmaktadır. OX-IMSFLA ve CX-IMSFLA algoritmaları yaklaşık çözümü elde eden yeni metotlardır ve bu yöntemler simetrik küçük boyutlu GSP'lerde yakın optimum sonuçları elde edebilmekte ancak büyük boyutlu GSP örneklerin çözümünde analiz sürelerini artırmaktadırlar.

Lin ve ark. (2018), bu çalışmalarında her ne kadar Kuş Sürüsü Optimizasyonu (BSA) algoritması, sürekli optimizasyon problemlerini çözmede mükemmel performans gösterse de, seyahat eden satıcı problemi (TSP) gibi kombinasyon optimizasyon problemini çözmek için uygulanmasının kolay bir iş olmadığı belirtilmektedir. Bu nedenle, bu çalışmada TSP için bilgi entropi matrisine (DBSA) dayanan yeni bir ayrık BSA önerilmektedir. İlk olarak, DBSA algoritmasında bilgi entropi matrisi, yeni çözümler üretmek için bir rehber olarak oluşturulmaktadır. Bilgi entropi matrisinin her bir elemanı, i kentinden j kentine bilgi entropisini belirtilir. Bilgi entropisi ne kadar yüksek olursa, bir kentin ziyaret edilme olasılığı da o kadar yüksek olmaktadır. İkincisi, her TSP yolu bir dizi olarak temsil edilir ve dizinin her elemanı bir şehir dizinini temsil etmektedir. Daha sonra bu çalışmada önerilen eksi işlevinin gereksinimlerine göre, her bir TSP yolu, kenarların ilişkisini temsil eden bir Boolean matrisine dönüştürülmektedir. Üçüncüsü, eksi işlevi iki Boole matrisi arasındaki farkı değerlendirmek için tasarlanmaktadır. Eksi işlevi ve bilgi entropi matrisine dayanarak, kuşların konum güncelleme denklemleri, orijinal özellikleri değiştirmeden bilgi entropi matrisini güncellemek için yeniden tasarlanmaktadır. Ardından, güncellenen bilgi entropi matrisine göre yeni çözümler üretmek için üç TSP operatörü önerilmektedir. Son olarak, DBSA algoritmasının performansı çok sayıda kıyaslamalı TSP örneği üzerinde test edilmektedir. Deneysel sonuçlar, DBSA algoritmasının daha iyi veya rekabetçi bir şekilde en son teknolojiye sahip metaheuristik algoritmaları geride bıraktığını göstermektedir.

Chen (2015), bu çalışmasında çok sayıda zor sorunu çözmek için Dağıtım Algoritmalarının Tahmini (EDA) kullanılmakta ve bununla birlikte, grup içi optimizasyon problemleriyle kullanımları literatürde geniş bir şekilde tartışılmakta olduğunu belirtmektedir. İyi bilinen bir grup içi optimizasyon problemi, eşzamanlı atama ve sıralama prosedürlerini içeren ve farklı şekillerde gösterilen Çoklu Gezgin Satıcı Problemidir (mTSP). Bu çalışmada Minimum Yükleme Tahsisi (MLA) kuralı ile Kendi Kendine Yönlendirilen Genetik Algoritmaya dayanan EDAMLA adlı yeni bir algoritma sunulmaktadır. Bu yöntemde doğrudan kodlama yerine dönüştürülmüş tabanlı kodlama yaklaşımını kullanır. Önerilen yöntemin çözüm alanı sadece $n!$ 'dir. Önerilen algoritmayı optimum kodlama tekniği, iki parçalı kodlama genetik algoritması ile karşılaştırılmaktadır. Deneylerin ölçeği önceki çalışmalarda sunulanları aşmaktadır. Sonuçlar önerilen algoritmanın, toplam hareket mesafesini minimize etmek için iki parçalı kodlayıcı genetik algoritma terimlerinden daha üstün olduğunu göstermektedir. Özellikle önerilen algoritma, satıcı sayısı 3'ten 10'a yükseltildiğinde daha uzun bir seyahat mesafesine neden olmadığını göstermektedir. Ayrıca EDA araştırmacılarının, önerilen algoritmalarında doğrudan kodlama yerine MLA kuralını kullanmaları gerektiğini belirtilmektedir.

Wang ve ark. (2014), bu çalışmalarında poligonal yaklaşım, görüntü analizi, görüntü tanıma ve bilgisayar görüşü için etkili ancak zorlu bir dijital eğri temsil ettiği belirtilmektedir. Bu çalışma poligonal yaklaşım için yeni bir tamsayı parçacık sürüsü optimizasyonu (iPSO) önermektedir. Parçacık sürüsü optimizasyonunun (bPSO) geleneksel ikili versiyonuyla karşılaştırıldığında, yeni iPSO, aday çözümü temsil etmek için doğrudan bir tamsayı vektörünü kullanır ve çözüm işlemi için daha verimli ve uygun bir yol sağladığı görülmektedir. iPSO'daki hız ve konum güncelleme mekanizmaları yalnızca net bir fiziksel anlama sahip değil, aynı zamanda çözümlerin optimitesini de garanti etmektedir. Bu metot optimizasyon problemi olabilen poligonal yaklaşım için uygun olmaktadır. Önerilen yöntem, dünyada kullanılan dört ünlü gölün haritalarından çıkarılan yaygın olarak kullanılan sentezlenmiş şekiller ve göl hatları üzerinde test edilmektedir. Deneysel sonuçlar önerilen iPSO'nun bPSO tabanlı yöntemlerden daha iyi çözüm kalitesine ve hesaplama etkinliğine ve diğer modern yöntemlerden daha iyi çözüm kalitesine sahip olduğunu göstermektedir.

Staněk ve ark. (2019), Açısal Metrik Gezgin Satıcı Problemi (AngleTSP), tüm dönme açılarının toplamını taklit eden Öklid düzleminde verilen puanlar için bir Hamiltonian

döngüsünü talep eder. Amaç olarak bu açıların lineer birleşimi ile klasik TSP arasındaki mesafeler, açısal mesafe-metrik gezici satıcı problemine (AngleDistanceTSP) yol açar. Her iki problem de NP-zor olduğu için, ilk olarak, optimal çözümlerin tipik geometrik yapısıyla motive edilen geniş bir sezgisel yaklaşımlar ortaya koyulmaktadır. Özellikle, mercek şeklindeki kenar komşularından ve grafiğin dışbükey kabuk katmanlarına ayrıştırılmasından sonra bir tura birleştirilir. İkincisi, daha genel İkinci Dereceden Gezgin Satıcı Problemine (QTSP) dayanan bir ILP modeli ele alınmaktadır. Fraksiyonel çözümlerini yuvarlayarak yardımcı ILP modellerini içeren ve çeşitli stratejilerle bir tura birleştirilen alt-yollar, yollar ve izolasyon noktaları koleksiyonları elde edilmektedir. Son olarakta özellikle bir ILP yaklaşımı tarafından belirlenen belirli bir noktadaki dikdörtgen sektörler için çözümü yerel olarak yeniden açan bir matürist olan farklı imgeleme sezgiselliği önerilmektedir. Literatürden referans örnekleri kullanarak yapılan kapsamlı hesaplama deneylerinin sonuçları, algoritmaların Pareto-verimli sınırını (çalışma süresi, objektif değer) bir alanda göstermektedir. Bu yeni yöntemin daha önce yayınlanmış sezgisel arama algortimalarına göre açıkça daha baskın olduğu ortaya çıktığı belirtilmektedir.

Lin ve ark. (2014), bu çalışmalarında Parçacık Sürüsü Optimizasyon Algoritması (PSOA) bir avantaj optimizasyon aracı olduğunu ve bununla birlikte, özellikle orta ve büyük boy problemler için en yakın çözümde takılma eğiliminin bulunduğunu ve ince ayar parametreleriyle çözüm doğruluğunu iyileştirmenin zorluğuna sahip olduğunu belirtmektedirler. Yetersizliğe göre, bu çalışmada Yerel Ve Küresel Aramayı Parçacık Sürüsü Algoritmasını (LGSCPSOA) ve yakınsaklığını birleştirerek araştırılmakta ve yakınsaklık niteliğini elde edilmektedir. Aynı zamanda, bir dizi 8 sürekli fonksiyon seti ile test edilmekte ve optimizasyon sonuçlarını Orijinal Parçacık Sürüsü Algoritması (OPSOA) ile karşılaştırılmaktadır. Deneysel sonuçlar LGSCPSOA'nın özellikle orta ve büyük boy kıyaslama işlevlerinde arama performansını önemli ölçüde iyileştirdiğini göstermektedir.

Yang ve Zhou (2017), Dinamik ve stokastik ulaşım ağlarında güvenilir yollar bulmak için genel bir modelleme çerçevesi sağlamayı amaçlayan bu çalışmada, yaygın olarak kullanılan iki seyahat süresi güvenilirlik önleminin, yani zaman varış olasılığı ve yüzdelik yolculuk süresine göre yeniden yapılandırılması yoluyla iki aşamalı yönlendirme modeli sınıfına değinilmektedir. Zamana bağlı bir bağlantı seyahat süresi verilerinin mekansal ve zamansal olarak korelasyon göstermesini sağlamak için örnek bir temsili kabul edilir.

Kolayca çözülebilen eşdeğer doğrusal tamsayılı programlama modellerini oluşturmak için bir dizi yeni reformülasyon yöntemleri tanıtılmaktadır. Tahmin edilemeyen kuplaj kısıtlamalarını farklı numuneler arasında çiftleştirmek ve daha sonra gevşeyen modeli hesaplamalı olarak etkin zamana bağlı en düşük maliyetli bir yol alt problemine ayırmak için bir Lagrangian ayrıştırma yaklaşımında geliştirilmektedir. Çalışmada önerilen yaklaşımların çözüm kalitesini ve hesaplama performansını göstermek için sayısal deneyler uygulanmaktadır.



3. MATERYAL VE YÖNTEM

Çalışmada yöntem olarak öncelikle GSP'nin Öklid Bağıntısı ile kuş uçuşu doğrusal çözümü için lineer optimizasyon ve PSO ile bir çözüm algoritması yazılmıştır. Daha sonra yazılan algoritmaların gerçek değerlere göre çözüm yapabilmesi için girdiler, koordinatlardan ayırılarak çözümü girdi değerlerine göre, güzergâh çizimini koordinatlara göre yapabilen modifiyeli algoritmalar tasarlanmıştır.

Materyaller ise güzergâh üzerinde bulunan noktalara ait koordinatlar Google GPS den, karayolu mesafeleri Karayolları Genel Müdürlüğü (KGM)'den, demiryolu mesafeleri Türkiye Cumhuriyeti Devlet Demir Yolları (T.C.D.D.) den, denizyolu mesafeleri Coğrafi Bilgi Sistemleri (CBS)'den alınmıştır. Taşımacılık süreleri taşımacılık türlerinin ortalama hızları üzerinden yükleme boşaltma ve bekleme süreleri ihmal edilerek hesaplanmıştır. Taşımacılık ton için birim maliyetleri, demiryolu için T.C.D.D. yük hesaplama portalından alınmış, karayolu ve denizyolu taşımacılığı için sonraki bölümlerde detayları verilecek olan giderler üzerinden el ile hesaplanmıştır.

Öncelikle GSP'nin genel çözüm mantığını göstermek adına hesap noktaları üzerinde kuş uçuşu doğrusal çözüm yapılmıştır. Daha sonra hesap güzergâhında yaygın kullanılan taşımacılık türü olan tek türlü karayolu taşımacılığı için ve çok türlü taşımacılık için: mesafe, süre ve maliyet parametrelerine göre eldeki materyaller üzerinden çözüm yapılmıştır.

3.1. Çalışmanın Amacı

Yapılan çalışmadaki amaç: bir noktadan yola çıkıp bilinen belirli noktalara uğrayıp yükleme boşaltma yaptıktan sonra başlangıç noktasına dönen yükün Lojistik köy ve ticari limanlarda ulaşım modu değişimi yaparak, çok türlü taşımacılık ile yapılmasını, GSP baz alınıp; mesafe, maliyet, süre ve istenilen alternatif parametrelere göre güzergâh optimizasyonu yapmaktır.

Çalışmada yapılacak örnek optimizasyon uygulamasında ise İskenderun Limanı'ndan çıkan bir miktar konteynerin Ticari Limanlar ve batı hattındaki Lojistik köylerde yükleme boşaltma yaptıktan sonra tekrar İskenderun Limanı'na dönmesi gerektiği varsayılarak; çok

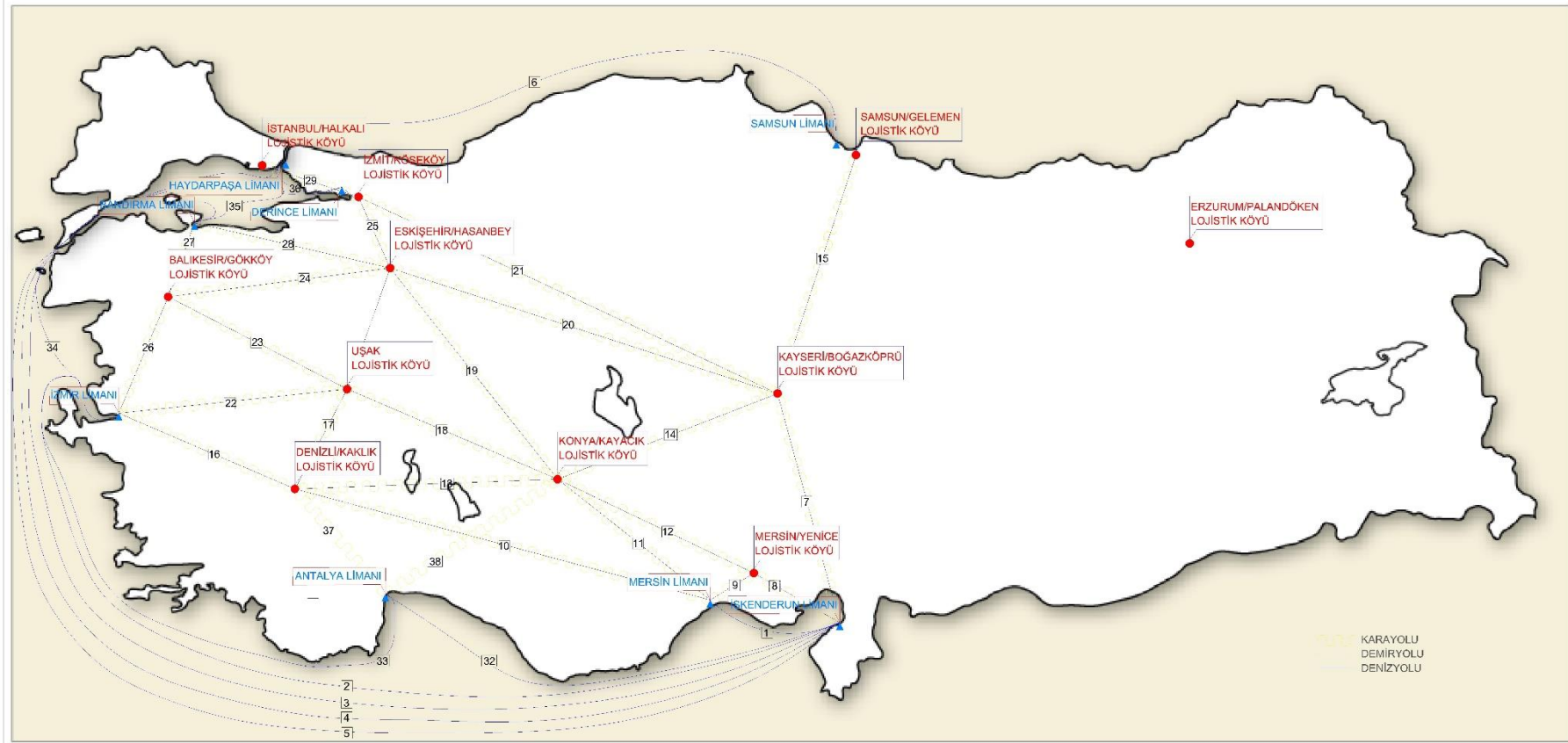
türlü taşımacılık ile mesafe, maliyet ve süre parametrelerine göre güzergâh optimizasyonu yapılması denlenmektedir. Çalışmada çoğunluk olarak uluslararası taşımacılıkta kullanılan çok türlü taşımacılığın iç taşımacılıkta kullanılması durumunda, iç taşımacılıkta yaygın olarak kullanılan taşımacılık türlerine olan avantajları ve dezavantajları araştırılmaktadır.

3.2. Çalışma Güzergâhında Yaygın Kullanılan Lojistik Hat bilgileri

Türkiye sınırları içerisinde taşımacılık sektöründe karayolu mesafelerinin kısa olmasından kaynaklanan daha hızlı ulaştırma ve daha az maliyet gibi etkenlerden dolayı karayolu taşımacılığı öncelikle tercih edilmektedir. Ayrıca karayolu ulaşım altyapısının gelişmiş olması ve diğer ulaşım türlerinin tüm bölgelerde olmaması, karayolu taşımacılığı için başka bir tercih sebebidir. Karayolu taşımacılığından sonra tercih edilen taşımacılık türü demiryolu taşımacılığıdır. Denizyolu taşımacılığı ise ağırlık olarak dış ticarete kullanılmaktadır.

3.2.1. Lojistik köyler ve ticari limanların dağılımı

Örnek çalışma güzergâhında bulunan lojistik köy, ticari limanlar ve bu lojistik tesisler arasında yaygın kullanılan lojistik güzergâh hat bilgileri Şekil 3.1' de gösterilmiştir.



Şekil 3.1. Çalışma güzergâhında yaygın kullanılan hat bilgileri şeması (Göçmen, 2018)

3.2.2. Lojistik köyler ve ticari limanlar yaygın kullanılan hat bilgileri

Örnek çalışma güzergâhında yaygın kullanılan lojistik güzergâh hat bilgileri Çizelge 3.1’ de listelenmiştir.

Çizelge 3.1. Yaygın kullanılan güzergâh bilgileri

Ağ Numarası	Düğüm Noktaları		Karayolu Mesafe (Km)	Demiryolu Mesafe (Km)	Denizyolu Mesafe (Mil)	Karayolu Birimfiyat (TL/Ton)	Demiryolu Birimfiyat (TL/Ton)	Denizyolu Birimfiyat (TL/Ton)
1	İskenderun Limanı	Mersin Limanı	-	-	76	-	-	15,21
2	İskenderun Limanı	İzmir Limanı	-	-	635	-	-	15,82
3	İskenderun Limanı	Bandırma Limanı	-	-	791	-	-	16,11
4	İskenderun Limanı	Haydarpaşa Limanı	-	-	835	-	-	16,3
5	İskenderun Limanı	Derince Limanı	-	-	860	-	-	16,47
6	Samsun Limanı	Haydarpaşa Limanı	-	-	449	-	-	16,3
7	İskenderun Limanı	Kayseri/Boğazköprü	423	445	-	48,27	42,12	-
8	İskenderun Limanı	Mersin/Yenice	157	161	-	33,64	16,31	-
9	Mersin Limanı	Mersin/Yenice	43,4	44	-	27,37	10,8	-
10	Mersin Limanı	Denizli/Kaklık	715	875	-	64,33	80,08	-
11	Mersin Limanı	Konya/Kayacık	358	388	-	44,69	36,41	-
12	Mersin/Yenice	Konya/Kayacık	324	347	-	43,10	32,61	-
13	Denizli/Kaklık	Konya/Kayacık	357	488	-	44,91	45,92	-
14	Konya/Kayacık	Kayseri/Boğazköprü	294	412	-	41,01	38,31	-
15	Kayseri/Boğazköprü	Samsun/Gelemen	446	588	-	49,64	54,52	-
16	Denizli/Kaklık	İzmir Limanı	252	289	-	39,74	27,17	-
17	Denizli/Kaklık	Uşak	120	353	-	31,60	32,61	-
18	Uşak	Konya/Kayacık	343	401	-	43,76	38,31	-
19	Konya/Kayacık	Eskişehir/Hasanbey	344	438	-	43,76	40,21	-
20	Eskişehir/Hasanbey	Kayseri/Boğazköprü	524	849	-	53,93	78,32	-
21	Kayseri/Boğazköprü	İzmit/Köseköy	626	1045	-	60,09	95,28	-
22	İzmir Limanı	Uşak	216	288	-	36,88	27,17	-
23	Balıkesir/Gökköy	Uşak	242	381	-	37,93	36,41	-
24	Balıkesir/Gökköy	Eskişehir/Hasanbey	345	350	-	43,70	32,61	-

Çizelge 3.1. (Devam) Yaygın kullanılan güzergâh bilgileri

Ağ Numarası	Düğüm Noktaları		Karayolu Mesafe (Km)	Demiryolu Mesafe (Km)	Denizyolu Mesafe (Mil)	Karayolu Birimfiyat (TL/Ton)	Demiryolu Birimfiyat (TL/Ton)	Denizyolu Birimfiyat (TL/Ton)
25	İzmit/Köseköy	Eskişehir/Hasanbey	218	218	-	36,94	19,93	-
26	İzmir Limanı	Balıkesir/Gökköy	182	232	-	35,01	21,74	-
27	Balıkesir/Gökköy	Bandırma Limanı	119	111	-	31,55	10,8	-
28	Bandırma Limanı	Eskişehir/Hasanbey	278	440	-	40,24	40,21	-
29	İstanbul/Halkalı	İzmit/Köseköy	135	186	-	32,21	18,12	-
30	Haydarpaşa Limanı	İzmit/Köseköy	100	0	-	30,50	10,8	-
31	Haydarpaşa Limanı	İstanbul/Halkalı	43,1	0	-	65,98	10,8	-
32	İskenderun Limanı	Antalya Limanı	-	-	282	-	-	15,38
33	Antalya Limanı	İzmir Limanı	-	-	402	-	-	15,82
34	İzmir Limanı	Bandırma Limanı	-	-	243	-	-	17,97
35	Bandırma Limanı	Haydarpaşa Limanı	-	-	64	-	-	18,44
36	Haydarpaşa Limanı	Derince Limanı	-	-	52	-	-	18,89
37	Antalya Limanı	Denizli/Kaklık	210	-	-	36,44	-	-
38	Antalya Limanı	Konya/Kayacık	319	-	-	42,55	-	-

3.3. Lojistik Köyler ve Ticari Limanların Konum ve Koordinatları

Çalışmada kullanılacak noktalara ait Google GPS'den alınan koordinat bilgileri Çizelge 3.2' de verilmiştir. Verilen koordinatlar GSP'nin doğrusal çözümünde ara mesafelerin Öklid Bağıntısı ile hesaplanmasında ve gerçek değerlere göre yapılan çözümde hat güzergâh çiziminin yapılmasında kullanılmıştır. Çözüm algoritması istenilen parametreye göre hesap değerlerini girdilerden alırken, hesaba denk gelen noktalara ait yapılacak güzergâh çizimi için nokta koordinatlarını Çizelge 3.2' deki nokta koordinat değerlerinden çekerek her parametre çözümüne ait ayrı çizim yapmıştır.

Çizelge 3.2. Nokta koordinatları

No	Düğüm Noktası	Düşey (Y) Koordinat	Yatay (X) Koordinat
1	İskenderun Limanı	36,600574	36,192790
2	Mersin Limanı	36,801896	34,639696
3	Antalya Limanı	36,839296	30,611871
4	İzmir Limanı	38,446685	27,156910
5	Bandırma Limanı	40,351112	27,962908
6	Haydarpaşa Limanı	41,005603	29,011459
7	Derince Limanı	40,754687	29,839334
8	Samsun Limanı	41,302517	36,333447
9	Mersin/Yenice	36,977463	35,063849
10	Kayseri/Boğazköprü	38,755488	35,322416
11	Konya/Kayacık	37,866168	32,476390
12	Denizli/Kaklık	37,829011	29,417109
13	Samsun/Gelemen	41,244760	36,432913
14	Uşak	38,664299	29,407196
15	Eskişehir/Hasanbey	39,793095	30,611766
16	Balıkesir/Gökköy	39,593498	27,821770
17	İzmit/Köseköy	40,748233	30,015052
18	İstanbul/Halkalı	41,018663	28,766118

3.4. Lojistik Köyler ve Ticari Limanların Birbirine Karayolu, Demiryolu ve Denizyolu mesafeleri

Multimodal taşımacılık için yapılan GSP nin PSO ile yazılan modifiyeli çözüm algoritmasının mesafe parametresine göre çözüm yaparken kullanmış olduğu;

KGM’ den alınan karayolu mesafeleri Çizelge 3.3’ de,

TCDD’ den alınan demiryolu mesafeleri Çizelge 3.4’ de,

CBS’ den alınan denizyolu mesafeleri Çizelge 3.5’ de verilmiştir.

CBS’den denizyolu mesafeleri deniz mili olarak alınmıştır.

Değer karşılaştırmalarında normalizasyonu sağlamak için denizyolu mesafeleri 1.85200’e bölünerek kilometreye çevrilmiştir.

Çizelge 3.3. Karayolu mesafeleri (Km)

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
		İskenderun Limanı	Mersin Limanı	Antalya Limanı	İzmir Limanı	Bandırma Limanı	Haydarpaşa Limanı	Derince Limanı	Samsun Limanı	Mersin/Yenice	Kayseri/Boğazköprü	Konya/Kayacık	Denizli/Kaklık	Samsun/Gelemen	Uşak Lojistik Merkezi	Eskişehir/Hasanbey	Balıkesir/Gökköy	İzmit/Köseköy	İstanbul/Halkalı
1	İskenderun Limanı	0	207	745	1018	1048	1047	961	847	157	423	473	824	867	804	791	1041	945	1076
2	Mersin Limanı	207	0	502	907	937	936	849	736	43	269	358	715	748	694	668	923	835	965
3	Antalya Limanı	745	502	0	461	608	693	626	956	603	614	319	208	933	302	452	514	618	721
4	İzmir Limanı	1018	907	461	0	276	462	439	1002	874	826	556	268	1013	216	420	182	454	490
5	Bandırma Limanı	1048	937	608	276	0	243	222	866	911	795	603	411	877	323	277	119	235	271
6	Haydarpaşa Limanı	1047	936	693	462	243	0	86	731	902	741	703	552	742	461	299	304	100	38
7	Derince Limanı	961	849	626	439	222	86	0	122	816	655	151	486	656	395	233	588	15	115
8	Samsun Limanı	847	736	956	1002	866	731	122	0	701	438	595	856	14	790	657	927	629	759
9	Mersin/Yenice	157	43	603	874	911	902	816	701	0	269	329	680	713	659	634	888	800	930
10	Kayseri/Boğazköprü	423	269	614	826	795	741	655	438	269	0	291	630	448	608	526	857	638	768
11	Konya/Kayacık	473	358	319	556	603	703	151	595	329	291	0	362	604	341	341	570	602	660
12	Denizli/Kaklık	824	715	208	268	411	552	486	856	680	630	362	0	863	120	310	311	464	574
13	Samsun/Gelemen	867	748	933	1013	877	742	656	14	713	448	604	863	0	801	668	938	640	770
14	Uşak	804	694	302	216	323	461	395	790	659	608	341	120	801	0	225	235	379	490
15	Eskişehir/Hasanbey	791	668	452	420	277	299	233	657	634	526	341	310	668	225	0	340	217	328
16	Balıkesir/Gökköy	1041	923	514	182	119	304	588	927	888	857	570	311	938	235	340	0	303	340
17	İzmit/Köseköy	945	835	618	454	235	100	15	629	800	638	602	464	640	379	217	303	0	131
18	İstanbul/Halkalı	1076	965	721	490	271	38	115	759	930	768	660	574	770	490	328	340	131	0

Çizelge 3.4. Demiryolu mesafeleri (Km)

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
		İskenderun Limanı	Mersin Limanı	Antalya Limanı	İzmir Limanı	Bandırma Limanı	Haydarpaşa Limanı	Derince Limanı	Samsun Limanı	Mersin/Yenice	Kayseri/Boğazköprü	Konya/Kayacık	Denizli/Kaklık	Samsun/Gelemen	Uşak Lojistik Köyü	Eskişehir/Hasanbey	Balıkesir/Gökköy	İzmit/Köseköy	İstanbul/Halkalı
1	İskenderun Limanı	0	204	-	1195	1228	1301	1157	1019	161	445	507	995	1019	908	945	1139	1141	1327
2	Mersin Limanı	204	0	-	1076	1109	1177	1037	913	44	326	388	875	913	789	826	1019	1021	1207
3	Antalya Limanı	-	-	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
4	İzmir Limanı	1195	1076	-	0	342	944	794	1687	1035	1100	689	289	1687	288	582	232	777	974
5	Bandırma Limanı	1228	1109	-	342	0	811	651	1720	1067	1132	721	628	1720	491	439	111	635	831
6	Haydarpaşa Limanı	1301	1177	-	944	811	0	150	1798	1146	1211	800	756	1798	669	384	712	167	30
7	Derince Limanı	1157	1037	-	794	651	150	0	1648	996	1061	650	606	1648	519	234	562	17	180
8	Samsun Limanı	1019	913	-	1687	1720	1798	1648	0	872	588	999	1487	30	1400	1437	1630	1632	1818
9	Mersin/Yenice	161	44	-	1035	1067	1146	996	872	0	285	347	834	872	748	784	978	980	1166
10	Kayseri/Boğazköprü	445	326	-	1100	1132	1211	1061	588	285	0	412	899	588	813	849	1043	1045	1231
11	Konya/Kayacık	507	388	-	689	721	800	650	999	347	412	0	488	999	401	438	632	634	820
12	Denizli/Kaklık	995	875	-	289	628	756	606	1487	834	899	488	0	1487	353	294	518	589	775
13	Samsun/Gelemen	1019	913	-	1687	1720	1798	1648	30	872	588	999	1487	0	1400	1437	1630	1632	1818
14	Uşak	908	789	-	288	491	669	519	1400	748	813	401	353	1400	0	307	381	503	689
15	Eskişehir/Hasanbey	945	826	-	582	439	384	234	1437	784	849	438	294	1437	307	0	350	218	404
16	Balıkesir/Gökköy	1139	1019	-	232	111	712	562	1630	978	1043	632	518	1630	381	350	0	546	732
17	İzmit/Köseköy	1141	1021	-	777	635	167	17	1632	980	1045	634	589	1632	503	218	546	0	186
18	İstanbul/Halkalı	1327	1207	-	974	831	30	180	1818	1166	1231	820	775	1818	689	404	732	186	0

Çizelge 3.5. Denizyolu mesafeleri (Km)

		1	2	3	4	5	6	7	8
		İskenderun Limanı	Mersin Limanı	Antalya Limanı	İzmir Limanı	Bandırma Limanı	Haydarpaşa Limanı	Derince Limanı	Samsun Limanı
1	İskenderun Limanı	0	122	454	1022	1273	1344	1384	2068
2	Mersin Limanı	122	0	357	924	1183	1246	1286	1970
3	Antalya Limanı	454	357	0	647	906	969	1025	1693
4	İzmir Limanı	1022	924	647	0	391	454	510	1046
5	Bandırma Limanı	1273	1183	906	391	0	103	153	747
6	Haydarpaşa Limanı	1344	1246	969	454	103	0	84	724
7	Derince Limanı	1384	1286	1025	510	153	84	0	779
8	Samsun Limanı	2068	1970	1693	1046	747	724	779	0

3.5. Lojistik Köyler ve Ticari Limanların Birbirine Karayolu, Demiryolu ve Denizyolu Süreleri

Multimodal taşımacılık için yapılan GSP'nin PSO ile yazılan modifiyeli çözüm algoritmasının süre parametresine göre çözüm yaparken kullanmış olduğu süreler hesaplanırken, yükleme boşaltma ve benzeri diğer süreler tüm taşımacılık türleri için hesaba katılmamış olup, mesafelere göre ortalama hızlardan, karayolu taşımacılığı, demiryolu taşımacılığı ve denizyolu taşımacılığı için süreler hesaplanmıştır. Ortalama hız bilgileri ilgili kurumların internet sitelerinden alınmıştır. Karayolu taşımacılığı ortalama hız hesabı yapılırken Karayolları Trafik Yönetmeliği'nde sürücüler için konulan 9 saatten fazla araç kullanamama sınırı ve dinlenme süreleri dikkate alınarak tır ve kamyonların azami hızları ve güzergâh mesafe bilgilerinden 24 saatlik ortalama hız bulunarak hesap yapılmıştır. Tüm bu hesaplamalardan sonra;

Karayolu süreleri Çizelge 3.6' da,

Demiryolu süreleri Çizelge 3.7' de,

Denizyolu süreleri Çizelge 3.8' de verilmiştir.

Çizelge 3.6. Karayolu süreleri (Saat)

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
		İskenderun Limanı	Mersin Limanı	Antalya Limanı	İzmir Limanı	Bandırma Limanı	Haydarpaşa Limanı	Derince Limanı	Samsun Limanı	Mersin/Yenice	Kayseri/Boğazköprü	Konya/Kayacık	Denizli/Kaklık	Samsun/Gelemen	Uşak Lojistik Köyü	Eskişehir/Hasanbey	Balıkesir/Gökköy	İzmit/Köseköy	İstanbul/Halkalı
1	İskenderun Limanı	0,00	4,60	16,56	22,62	23,29	23,27	21,36	18,82	3,49	9,40	10,51	18,31	19,27	17,87	17,58	23,13	21,00	23,91
2	Mersin Limanı	4,60	0,00	11,16	20,16	20,82	20,80	18,87	16,36	0,96	5,98	7,96	15,89	16,62	15,42	14,84	20,51	18,56	21,44
3	Antalya Limanı	16,56	11,16	0,00	10,24	13,51	15,40	13,91	21,24	13,40	13,64	7,09	4,62	20,73	6,71	10,04	11,42	13,73	16,02
4	İzmir Limanı	22,62	20,16	10,24	0,00	6,13	10,27	9,76	22,27	19,42	18,36	12,36	5,96	22,51	4,80	9,33	4,04	10,09	10,89
5	Bandırma Limanı	23,29	20,82	13,51	6,13	0,00	5,40	4,93	19,24	20,24	17,67	13,40	9,13	19,49	7,18	6,16	2,64	5,22	6,02
6	Haydarpaşa Limanı	23,27	20,80	15,40	10,27	5,40	0,00	1,87	16,24	20,04	16,47	15,62	12,27	16,49	10,24	6,64	6,76	2,22	0,84
7	Derince Limanı	21,36	18,87	13,91	9,76	4,93	1,91	0,00	2,71	18,13	14,56	3,36	10,80	14,58	8,78	5,18	13,07	0,33	2,56
8	Samsun Limanı	18,82	16,36	21,24	22,27	19,24	16,24	2,71	0,00	15,58	9,73	13,22	19,02	0,31	17,56	14,60	20,60	13,98	16,87
9	Mersin/Yenice	3,49	0,96	13,40	19,42	20,24	20,04	18,13	15,58	0,00	5,98	7,31	15,11	15,84	14,64	14,09	19,73	17,78	20,67
10	Kayseri/Boğazköprü	9,40	5,98	13,64	18,36	17,67	16,47	14,56	9,73	5,98	0,00	6,47	14,00	9,96	13,51	11,69	19,04	14,18	17,07
11	Konya/Kayacık	10,51	7,96	7,09	12,36	13,40	15,62	3,36	13,22	7,31	6,47	0,00	8,04	13,42	7,58	7,58	12,67	13,38	14,67
12	Denizli/Kaklık	18,31	15,89	4,62	5,96	9,13	12,27	10,80	19,02	15,11	14,00	8,04	0,00	19,18	2,67	6,89	6,91	10,31	12,76
13	Samsun/Gelemen	19,27	16,62	20,73	22,51	19,49	16,49	14,58	0,31	15,84	9,96	13,42	19,18	0,00	17,80	14,84	20,84	14,22	17,11
14	Uşak	17,87	15,42	6,71	4,80	7,18	10,24	8,78	17,56	14,64	13,51	7,58	2,67	17,80	0,00	5,00	5,22	8,42	10,89
15	Eskişehir/Hasanbey	17,58	14,84	10,04	9,33	6,16	6,64	5,18	14,60	14,09	11,69	7,58	6,89	14,84	5,00	0,00	7,56	4,82	7,29
16	Balıkesir/Gökköy	23,13	20,51	11,42	4,04	2,64	6,76	13,07	20,60	19,73	19,04	12,67	6,91	20,84	5,22	7,56	0,00	6,73	7,56
17	İzmit/Köseköy	21,00	18,56	13,73	10,09	5,22	2,22	0,33	13,98	17,78	14,18	13,38	10,31	14,22	8,42	4,82	6,73	0,00	2,91
18	İstanbul/Halkalı	23,91	21,44	16,02	10,89	6,02	0,84	2,56	16,87	20,67	17,07	14,67	12,76	17,11	10,89	7,29	7,56	2,91	0,00

Çizelge 3.7. Demiryolu süreleri (Saat)

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
		İskenderun Limanı	Mersin Limanı	Antalya Limanı	İzmir Limanı	Bandırma Limanı	Haydarpaşa Limanı	Derince Limanı	Samsun Limanı	Mersin/Yenice	Kayseri/Boğazköprü	Konya/Kayacık	Denizli/Kaklık	Samsun/Gelemen	Uşak	Eskişehir/Hasanbey	Balıkesir/Gökköy	İzmit/Köseköy	İstanbul/Halkalı
1	İskenderun Limanı	0,00	3,71	-	21,73	22,33	23,65	21,04	18,53	2,93	8,09	9,22	18,09	18,53	16,51	17,18	20,71	20,75	24,13
2	Mersin Limanı	3,71	0,00	-	19,56	20,16	21,40	18,85	16,60	0,80	5,93	7,05	15,91	16,60	14,35	15,02	18,53	18,56	21,95
3	Antalya Limanı	-	-	0,00	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
4	İzmir Limanı	21,73	19,56	-	0,00	6,22	17,16	14,44	30,67	18,82	20,00	12,53	5,25	30,67	5,24	10,58	4,22	14,13	17,71
5	Bandırma Limanı	22,33	20,16	-	6,22	0,00	14,75	11,84	31,27	19,40	20,58	13,11	11,42	31,27	8,93	7,98	2,02	11,55	15,11
6	Haydarpaşa Limanı	23,65	21,40	-	17,16	14,75	0,00	2,73	32,69	20,84	22,02	14,55	13,75	32,69	12,16	6,98	12,95	3,04	0,55
7	Derince Limanı	21,04	18,85	-	14,44	11,84	2,73	0,00	29,96	18,11	19,29	11,82	11,02	29,96	9,44	4,25	10,22	0,31	3,27
8	Samsun Limanı	18,53	16,60	-	30,67	31,27	32,69	29,96	0,00	15,85	10,69	18,16	27,04	0,55	25,45	26,13	29,64	29,67	33,05
9	Mersin/Yenice	2,93	0,80	-	18,82	19,40	20,84	18,11	15,85	0,00	5,18	6,31	15,16	15,85	13,60	14,25	17,78	17,82	21,20
10	Kayseri/Boğazköprü	8,09	5,93	-	20,00	20,58	22,02	19,29	10,69	5,18	0,00	7,49	16,35	10,69	14,78	15,44	18,96	19,00	22,38
11	Konya/Kayacık	9,22	7,05	-	12,53	13,11	14,55	11,82	18,16	6,31	7,49	0,00	8,87	18,16	7,29	7,96	11,49	11,53	14,91
12	Denizli/Kaklık	18,09	15,91	-	5,25	11,42	13,75	11,02	27,04	15,16	16,35	8,87	0,00	27,04	6,42	5,35	9,42	10,71	14,09
13	Samsun/Gelemen	18,53	16,60	-	30,67	31,27	32,69	29,96	0,55	15,85	10,69	18,16	27,04	0,00	25,45	26,13	29,64	29,67	33,05
14	Uşak	16,51	14,35	-	5,24	8,93	12,16	9,44	25,45	13,60	14,78	7,29	6,42	25,45	0,00	5,58	6,93	9,15	12,53
15	Eskişehir/Hasanbey	17,18	15,02	-	10,58	7,98	6,98	4,25	26,13	14,25	15,44	7,96	5,35	26,13	5,58	0,00	6,36	3,96	7,35
16	Balıkesir/Gökköy	20,71	18,53	-	4,22	2,02	12,95	10,22	29,64	17,78	18,96	11,49	9,42	29,64	6,93	6,36	0,00	9,93	13,31
17	İzmit/Köseköy	20,75	18,56	-	14,13	11,55	3,04	0,31	29,67	17,82	19,00	11,53	10,71	29,67	9,15	3,96	9,93	0,00	3,38
18	İstanbul/Halkalı	24,13	21,95	-	17,71	15,11	0,55	3,27	33,05	21,20	22,38	14,91	14,09	33,05	12,53	7,35	13,31	3,38	0,00

Çizelge 3.8. Denizyolu süreleri (Saat)

		1	2	3	4	5	6	7	8
		İskenderun Limanı	Mersin Limanı	Antalya Limanı	İzmir Limanı	Bandırma Limanı	Haydarpaşa Limanı	Derince Limanı	Samsun Limanı
1	İskenderun Limanı	0,00	3,71	13,75	30,97	38,58	40,72	41,94	62,67
2	Mersin Limanı	3,71	0,00	10,83	27,99	35,84	37,75	38,97	59,70
3	Antalya Limanı	13,75	10,83	0,00	19,60	27,46	29,36	31,06	51,30
4	İzmir Limanı	30,97	27,99	19,60	0,00	11,85	13,75	15,46	31,70
5	Bandırma Limanı	38,58	35,84	27,46	11,85	0,00	3,12	4,63	22,63
6	Haydarpaşa Limanı	40,72	37,75	29,36	13,75	3,12	0,00	2,54	21,95
7	Derince Limanı	41,94	38,97	31,07	15,46	4,63	2,54	0,00	23,60
8	Samsun Limanı	62,67	59,70	51,30	31,70	22,64	21,94	23,61	0,00

3.6. Lojistik Köyler ve Ticari Limanların Birbirine Karayolu, Demiryolu ve Denizyolu Birim Maliyetleri

Çok türlü taşımacılık için yapılan GSP'nin lineer optimizasyon ve PSO ile yazılan modifiyeli çözüm algoritmasının maliyet parametresine göre çözüm yaparken kullanmış olduğu 1 ton için birim maliyetlerler hesaplanırken;

Karayolu taşımacılığı birim maliyetler: birim yük için kilometre başına yakıt gideri, birim yük için tur başına şoför gideri ve birim yük için tur başına otoban, yıpranma ve diğer giderler toplanarak bulunmuştur.

Demiryolu taşımacılığı için birim maliyetler: her güzergâh ayrı ayrı girilerek (İnternet, 2018e)'den mesafelerle beraber birim yük için alınmıştır.

Denizyolu için birim maliyetler: birim yük için seyirde ortalama yakıt tüketimi, birim yük için limanda ortalama yakıt tüketimi, birim yük için terminal hizmeti, birim yük için yükleme boşaltma maliyeti, birim yük için direkt ve endirekt gemi giderleri ve birim yük için karayolu aktarma ücretleri toplanarak hesaplanmıştır.

Birim yük için maliyetler hesaplandığında maliyetlerin mesafelerle orantılı olmadığı görülmektedir. Bunun sebebi ise nakliye haricindeki diğer masraflardır. Özellikle denizyolu taşımacılığından bir yük gemisinin ortalama 8.000 ila 10.000 arasında konteyner

taşıdığı dikkate alındığında birim yük için mesafeye bağlı olan yakıt tüketimi neredeyse yok denilecek kadar azdır. Denizyolu taşımacılığında asıl maliyeti liman giderleri ve diğer giderler oluşturmaktadır ki bu giderler mesafe ile orantılı değildir. Keza aynı durum demiryolu taşımacılığı ve nispeten karayolu taşımacılığı için de geçerlidir.

Belirtilen kriterlere üzerinden taşımacılık türlerine göre birim maliyetler hesaplandıktan sonra, karayolu taşımacılığı birim maliyetleri Çizelge 3.9' da, demiryolu taşımacılığı birim maliyetleri Çizelge 3.10' da ve denizyolu taşımacılığı birim maliyetleri Çizelge 3.11' de verilmiştir.



Çizelge 3.9. Karayolu birim maliyetler (TL)

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
		İskenderun Limanı	Mersin Limanı	Antalya Limanı	İzmir Limanı	Bandırma Limanı	Haydarpaşa Limanı	Derince Limanı	Samsun Limanı	Mersin/Yenice	Kayseri/Boğazköprü	Konya/Kayacık	Denizli/Kaklık	Samsun/Gelemen	Uşak	Eskişehir/Hasanbey	Balıkesir/Gökköy	İzmit/Köseköy	İstanbul/Halkalı
1	İskenderun Limanı	0,00	36,39	65,98	80,99	82,64	82,59	77,86	71,59	33,64	48,27	51,02	70,32	72,69	69,22	68,51	82,26	76,98	84,18
2	Mersin Limanı	36,39	0,00	52,61	74,89	76,54	76,48	71,70	65,48	27,37	39,80	44,69	64,33	66,14	63,17	61,74	75,77	70,93	78,08
3	Antalya Limanı	65,98	52,61	0,00	50,36	58,44	63,12	59,43	77,58	58,17	58,77	42,55	36,44	76,32	41,61	49,86	53,27	58,99	64,66
4	İzmir Limanı	80,99	74,89	50,36	0,00	40,18	50,41	49,15	80,11	73,07	70,43	55,58	39,74	80,72	36,88	48,10	35,01	49,97	51,95
5	Bandırma Limanı	82,64	76,54	58,44	40,18	0,00	38,37	37,21	72,63	75,11	68,73	58,17	47,61	73,24	42,77	40,24	31,55	37,93	39,91
6	Haydarpaşa Limanı	82,59	76,48	63,12	50,41	38,37	0,00	29,62	65,21	74,61	65,76	63,67	55,36	65,81	50,36	41,45	41,72	30,50	27,09
7	Derince Limanı	77,86	71,70	59,43	49,15	37,21	29,73	0,00	31,71	69,88	61,03	33,31	51,73	61,08	46,73	37,82	57,34	25,83	31,33
8	Samsun Limanı	71,59	65,48	77,58	80,11	72,63	65,21	31,71	0,00	63,56	49,09	57,73	72,08	25,77	68,45	61,14	75,99	59,60	66,75
9	Mersin/Yenice	33,64	27,37	58,17	73,07	75,11	74,61	69,88	63,56	0,00	39,80	43,10	62,40	64,22	61,25	59,87	73,84	69,00	76,15
10	Kayseri/Boğazköprü	48,27	39,80	58,77	70,43	68,73	65,76	61,03	49,09	39,80	0,00	41,01	59,65	49,64	58,44	53,93	72,14	60,09	67,24
11	Konya/Kayacık	51,02	44,69	42,55	55,58	58,17	63,67	33,31	57,73	43,10	41,01	0,00	44,91	58,22	43,76	43,76	56,35	58,11	61,30
12	Denizli/Kaklık	70,32	64,33	36,44	39,74	47,61	55,36	51,73	72,08	62,40	59,65	44,91	0,00	72,47	31,60	42,05	42,11	50,52	56,57
13	Samsun/Gelemen	72,69	66,14	76,32	80,72	73,24	65,81	61,08	25,77	64,22	49,64	58,22	72,47	0,00	69,06	61,74	76,59	60,20	67,35
14	Uşak	69,22	63,17	41,61	36,88	42,77	50,36	46,73	68,45	61,25	58,44	43,76	31,60	69,06	0,00	37,38	37,93	45,85	51,95
15	Eskişehir/Hasanbey	68,51	61,74	49,86	48,10	40,24	41,45	37,82	61,14	59,87	53,93	43,76	42,05	61,74	37,38	0,00	43,70	36,94	43,04
16	Balıkesir/Gökköy	82,26	75,77	53,27	35,01	31,55	41,72	57,34	75,99	73,84	72,14	56,35	42,11	76,59	37,93	43,70	0,00	41,67	43,70
17	İzmit/Köseköy	76,98	70,93	58,99	49,97	37,93	30,50	25,83	59,60	69,00	60,09	58,11	50,52	60,20	45,85	36,94	41,67	0,00	32,21
18	İstanbul/Halkalı	84,18	78,08	64,66	51,95	39,91	27,09	31,33	66,75	76,15	67,24	61,30	56,57	67,35	51,95	43,04	43,70	32,21	0,00

Çizelge 3.10. Demiryolu birim maliyetler (TL)

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
		İskenderun Limanı	Mersin Limanı	Antalya Limanı	İzmir Limanı	Bandırma Limanı	Haydarpaşa Limanı	Derince Limanı	Samsun Limanı	Mersin/ Yenice	Kayseri/ Boğazköprü	Konya/ Kayacık	Denizli/ Kaklık	Samsun/ Gelemen	Uşak	Eskişehir/ Hasanbey	Balıkesir/ Gökköy	İzmit/ Köseköy	İstanbul/ Halkalı
1	İskenderun Limanı	0,00	19,93	-	109,27	113,94	112,47	109,27	95,28	16,31	42,12	47,64	90,62	95,28	83,59	87,11	104,61	104,61	123,27
2	Mersin Limanı	19,93	0,00	-	99,95	104,61	103,14	95,28	83,59	10,80	30,80	36,41	80,08	83,59	73,05	76,56	95,28	95,28	113,94
3	Antalya Limanı	-	-	0,00	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
4	İzmir Limanı	109,27	99,95	-	0,00	32,61	78,06	73,05	155,91	95,28	99,95	63,13	27,17	155,91	27,17	54,52	21,74	71,29	88,86
5	Bandırma Limanı	113,94	104,61	-	32,61	0,00	65,76	59,68	160,58	99,95	104,61	67,78	57,96	160,58	45,92	40,21	10,80	57,96	76,56
6	Haydarpaşa Limanı	112,47	103,14	-	78,06	65,76	0,00	10,80	162,05	101,42	110,75	70,48	67,04	162,05	58,44	32,54	63,60	21,60	27,11
7	Derince Limanı	109,27	95,28	-	73,05	59,68	10,80	0,00	151,25	90,62	99,95	59,68	56,24	151,25	47,64	21,74	52,80	10,80	16,31
8	Samsun Limanı	95,28	83,59	-	155,91	160,58	162,05	151,25	0,00	80,08	54,52	90,62	137,26	10,80	127,93	132,59	151,25	151,25	169,90
9	Mersin/Yenice	16,31	10,80	-	95,28	99,95	101,42	90,62	80,08	0,00	27,17	32,61	76,56	80,08	69,54	73,05	88,86	88,86	109,27
10	Kayseri/Boğazköprü	42,12	30,80	-	99,95	104,61	110,75	99,95	54,52	27,17	0,00	38,31	81,83	54,52	74,81	78,32	95,28	95,28	113,94
11	Konya/Kayacık	47,64	36,41	-	63,13	67,78	70,48	59,68	90,62	32,61	38,31	0,00	45,92	90,62	38,31	40,21	57,96	57,96	74,81
12	Denizli/Kaklık	90,62	80,08	-	27,17	57,96	67,04	56,24	137,26	76,56	81,83	45,92	0,00	137,26	32,61	36,41	47,64	54,52	71,29
13	Samsun/Gelemen	95,28	83,59	-	155,91	160,58	162,05	151,25	10,80	80,08	54,52	90,62	137,26	0,00	127,93	132,59	151,25	151,25	169,90
14	Uşak	83,59	73,05	-	27,17	45,92	58,44	47,64	127,93	69,54	74,81	38,31	32,61	127,93	0,00	28,98	36,41	47,64	63,13
15	Eskişehir/Hasanbey	87,11	76,56	-	54,52	40,21	32,54	21,74	132,59	73,05	78,32	40,21	36,41	132,59	28,98	0,00	32,61	19,93	38,31
16	Balıkesir/Gökköy	104,61	95,28	-	21,74	10,80	63,60	52,80	151,25	88,86	95,28	57,96	47,64	151,25	36,41	32,61	0,00	51,08	67,78
17	İzmit/Köseköy	104,61	95,28	-	71,29	57,96	21,60	10,80	151,25	88,86	95,28	57,96	54,52	151,25	47,64	19,93	51,08	0,00	18,12
18	İstanbul/Halkalı	123,27	113,94	-	88,86	76,56	27,11	16,31	169,90	109,27	113,94	74,81	71,29	169,90	63,13	38,31	67,78	18,12	0,00

Çizelge 3.11. Denizyolu birim maliyetler (TL)

		1	2	3	4	5	6	7	8
		İskenderun Limanı	Mersin Limanı	Antalya Limanı	İzmir Limanı	Bandırma Limanı	Haydarpaşa Limanı	Derince Limanı	Samsun Limanı
1	İskenderun Limanı	0,00	15,21	15,38	15,82	16,11	16,30	16,47	16,98
2	Mersin Limanı	15,21	0,00	16,25	16,70	16,99	17,17	17,35	17,85
3	Antalya Limanı	15,38	16,25	0,00	17,33	17,61	17,80	17,98	18,48
4	İzmir Limanı	15,82	16,70	17,33	0,00	17,97	18,15	18,34	18,77
5	Bandırma Limanı	16,11	16,99	17,61	17,97	0,00	18,44	18,62	19,08
6	Haydarpaşa Limanı	16,30	17,17	17,80	18,15	18,44	0,00	18,89	19,37
7	Derince Limanı	16,47	17,35	17,98	18,34	18,62	18,89	0,00	19,55
8	Samsun Limanı	16,98	17,85	18,48	18,77	19,08	19,37	19,55	0,00

3.7. Lineer Optimizasyon

3.7.1. Üç şehirli lineer optimizasyon algoritmasının oluşturulması

Yapılan çalışmada, kullanılan 18 adet şehir arasından seçilen 3 şehir için algoritmalar oluşturulmuştur. Bunun amacı kullanılan algoritmaların çalışma prensiblerine değinmektir. Çizelge 3.12’ deki kodlama şehirlerarasındaki minimum mesafeleri belirleyip bunları matris formatında kaydetmemizi sağlayan başlangıç algoritmasıdır. Çizelge 3.12’ deki kodlama sonucu elde edilen minimum mesafelerin matrise dönüştürülmüş hali Çizelge 3.13’ te verilmektedir.

Çizelge 3.12. Şehirlerarasındaki minimum mesafeleri alıp matris haline getiren kodlar

```

Iskenderun.Antalya.Kara=745;
Iskenderun.Antalya.Deniz=454;

Iskenderun.Usak.Kara=804;
Iskenderun.Usak.Demir=908;

a2=[Iskenderun.Antalya.Kara Iskenderun.Antalya.Deniz];
a3=[Iskenderun.Usak.Kara Iskenderun.Usak.Demir];

Antalya.Usak.Kara=302;

b3=[Antalya.Usak.Kara];

MesafeMatrisi=[0 min(a2) min(a3);...
               min(a2) 0 min(b3);...
               min(a3) min(b3) 0];

```


Çizelge 3.13. Şehirlerarasındaki minimum mesafeleri alıp matris haline getiren kodların sonuçları

Mesafe Matrisi			
	1. şehir	2. şehir	3. şehir
1. şehir	0	454	804
2. şehir	454	0	302
3. şehir	804	302	0

Oluşturulan algorithmada bir sonraki adım şehir çiftlerini oluşturmaktır. Bunun amacı şehirleri birbirleri ile eşleştirip, mesafe vektörünü oluşturularak aralarındaki mesafelerin belirlenmesidir. Şehir çiftlerini oluşturan algoritma Çizelge 3.14’ te verilmektedir. Çizelge 3.14’ te bulunan algoritma sonucunda elde edilen değerler Çizelge 3.15’ te verilmektedir.

Çizelge 3.14. Şehir çiftlerini ve mesafe vektörünü oluşturan kodlama

```

c=1;
for count = 1:SehirSayisi:(SehirSayisi*SehirSayisi);
    SehirCifti(count:SehirSayisi*c, 1) = c;
    SehirCifti(count:SehirSayisi*c, 2) = 1:SehirSayisi;
    MesafeVektörü(count:SehirSayisi*c, 1) = MesafeMatrisi(c,:);
    c=c+1;
end
UzaklikMesafeVektörü=length(MesafeVektörü);

```

Çizelge 3.15. Şehir çiftlerini ve mesafe vektörünü oluşturan kodlamanın sonuçları

Şehir Cifti		Mesafe vektörü	Uzaklık mesafe vektörü
1	1	0	9
1	2	454	
1	3	804	
2	1	454	
2	2	0	
2	3	302	
3	1	804	
3	2	302	
3	3	0	

Algorithmada bir sonraki aşama kısıtlamaları oluşturan algorithmaları yazmaktır. GSP’ nin mantığı bir noktadan başlayıp diğer noktalara uğrayarak tekrar başlangıç noktasına en kısa turu oluşturacak şekilde dönmektir. Bu en kısa turu oluştururken her noktaya bir defa uğraması gerekmektedir. Bunun için algorithmada gerekli kısıtlamalar girilmesi gerekir. Eşitlik kısıtlamları bir şehre yapılacak seyahat sayısını ve o şehirden yapılacak seyahat sayısının bir defa olmasını sağlayan algorithmadır. Eşitlik kısıtları için oluşturulan algoritma Çizelge 3.16 ve 18’ de verilmektedir. Algoritma sonucu elde edilen değerleri Çizelge 3.17 ve 19’ da verilmektedir.

Algoritmanın bir sonraki aşamasında alt ve üst sınırlar belirlenmektedir. Bu durum hangi yolun belirleneceğine karar vermek için uygulanmaktadır. Seçilecek yolun bir veya sıfır olarak değerlendirilip oluşacak optimal turun belirlenmesini sağlamaktadır. Alt ve üst sınırlamaların belirtildiği kodlamalar Çizelge 3.22’ de verilmektedir. Kodlama sonucu elde edilen değerler Çizelge 3.23’ te verilmektedir.

Çizelge 3.22. Alt üst sınırlamaların belirtildiği kodlar

```
intcon=1:UzaklikMesafeVektoru;
lb=zeros(UzaklikMesafeVektoru,1);
ub=ones(UzaklikMesafeVektoru,1);
```

Çizelge 3.23. Alt üst sınırlamaların belirtildiği kodların sonuçları

int con	lb	ub
1	0	1
2	0	1
3	0	1
4	0	1
5	0	1
6	0	1
7	0	1
8	0	1
9	0	1

Algoritmanın bir sonraki aşamasında linner optimizasyon kodlaması devreye girmektedir. Bu kodlama sayesinde oluşacak optimal tur belirlenmektedir. Oluşturulan kısıtlamalar sonucu optimal güzergâhı belirleyen kodlamalar Çizelge 3.24’ te verilmektedir. Kodlama sonucu elde edilen optimal turun başlangıç noktası ve bitiş noktası ve toplam km değeri Çizelge 3.25’ te verilmektedir.

Çizelge 3.24. Üç nokta arasındaki güzergâhın belirlenmesinde kullanılan kodlar

```
opts=optimoptions('intlinprog','CutGeneration','Advanced','NodeSelection','mininfeas','Display','off');
[KararDegiskenleri,OptimumMaliyet,CikisNoktasi,Cikti]=intlinprog(MesafeVektoru,intcon,[],[],Aeq,beq,lb,ub,opts);
```

Çizelge 3.25. Üç nokta arasındaki güzergâhın belirlenmesinde kullanılan kodların sonuçları

Karar değişkenleri	Optimum mesafe	Çıkış noktası	Çıktı
0	1560	1	relativegap: 0
1			absolutegap: 0
0			numfeaspoints: 1
0			numnodes: 0
0			constrviolation: 0
1			message: Optimal solution found. Intlinprog stopped at the root node because the objective value is within a gap tolerance of the optimal value, options. AbsoluteGapTolerance = 0 (the default value). The intcon variables are integer within tolerance, options. IntegerTolerance = 1e-05 (the default value).
1			
0			
0			

Algoritmanın bir sonraki aşamasında oluşan turun içinde alt turların olup olmadığını kontrol etmek için gereken kodlamalar yapılmaktadır. GSP örneğinin bir diğer kısıtı da alt turların oluşmasını engellemektir. Alt turlar birbirine yakın noktaların kendi aralarında tur oluşturmasına denir. Bunu engellemek için gereken kodlama Çizelge 3.26 da verilmektedir. Kodlama sonucu elde edilen alt tur sayısı ve başlangıç ve bitiş noktaları değerleri Çizelge 3.27’ de verilmektedir.

Çizelge 3.26. Altur turların belirlenmesinde kullanılan kodlar

```
Turlar=detectSubtours(KararDegiskenleri,SehirCifti);
AltTurSayisi=length(Turlar);
```

Çizelge 3.27. Altur turların belirlenmesinde kullanılan kodların sonuçları

Turlar	Alt tur sayısı
[2 3 1]	1

Algoritmanın son aşamasında ise hesaplanan alt turları elimine ederek en son optimal turun belirlenmesini sağlayan algortma yazılmaktadır. Yazılan algoritma Çizelge 3.28’de verilmektedir. Kodlama sonucu elde edilen veriler Çizelge 3.29’ da sunulmaktadır.

Çizelge 3.28. Alt turları elimine ederek optimum güzergâhı oluşturan kodlar

```

A=spalloc(0,UzaklikMesafeVektoru,0);
b=[];
while AltTurSayisi>1
    b=[b;zeros(AltTurSayisi,1)];
    A=[A;spalloc(AltTurSayisi,UzaklikMesafeVektoru,SehirSayisi)];
    for count=1:AltTurSayisi;
        EsitsizlikKisitNumarasi=size(A,1)+1;
        AltTurNo=Turlar{count};
        AltTurCifti=nchoosek(1:length(AltTurNo),2);
        for JJ=1:size(AltTurCifti,1);
            AltTurDegiskeni=(sum(SehirCifti==AltTurNo(AltTurCifti(JJ,1)),2))&...
                (sum(SehirCifti==AltTurNo(AltTurCifti(JJ,2)),2));
            A(EsitsizlikKisitNumarasi,AltTurDegiskeni)=1;
        end
        b(EsitsizlikKisitNumarasi)=length(AltTurNo)-1;
    end
    fprintf('\n Alt turları elimine ederek sorunu tekrar çözme\n');
    [KararDegiskenleri,OptimumMaliyet,CikisNoktasi,Cikti]=intlinprog(MesafeVektoru,intcon,A,b,Aeq,beq,lb,ub,opts);
    fprintf('Alt turları tekrar kontrol et\n');
    Turlar=detectSubtours(KararDegiskenleri,SehirCifti);
    AltTurSayisi=length(Turlar);
    fprintf('Alt tur sayıları: %d\n',AltTurSayisi);
end

```

Çizelge 3.29. Alt turları elimine ederek optimum güzergâhı oluşturan kodların sonuçları

A	b	while dongu	Karar değiş.	Optimum mesafe	Çıkış noktası	Çıktı
All zero sparse: 0-by-9	[]		0	1560	1	relativegap: 0
			1			absolutegap: 0
			0			numfeaspoints: 1
			0			numnodes: 0
			0			constrviolation: 0
			1			message: Optimal solution found.
			1			Intlinprog stopped at the root node
			0			because the objective value is within
			0			a gap tolerance of the optimal value,
						options. AbsoluteGapTolerance = 0
						(the default value). The intcon
						variables are integer within
						tolerance, options. IntegerTolerance
						= 1e-05 (the default value).

4. ARAŞTIRMA BULGULARI VE TARTIŞMA

Yapılan çalışmada daha önceki bölümlerde gösterilen veriler üzerinden 3, 4 ve 18 şehir için MATLAB programı kullanılarak kuş uçuşu mesafeler ile koordinatlara göre standart çözüm yapılmaktadır. Daha sonra ise yine aynı şehirler üzerinden lineer optimizasyon ve modifiye edilmiş PSO algoritması ile yine daha önceki bölümlerde gösterilen gerçek değerler üzerinden sadece karayolu taşımacılığı ve çok türlü taşımacılık için mesafe, süre ve maliyet çözümleri yapılarak değerlendirilmektedir.

4.1. Lineer Optimizasyon İle Koordinatlara Göre Standart Çözümü

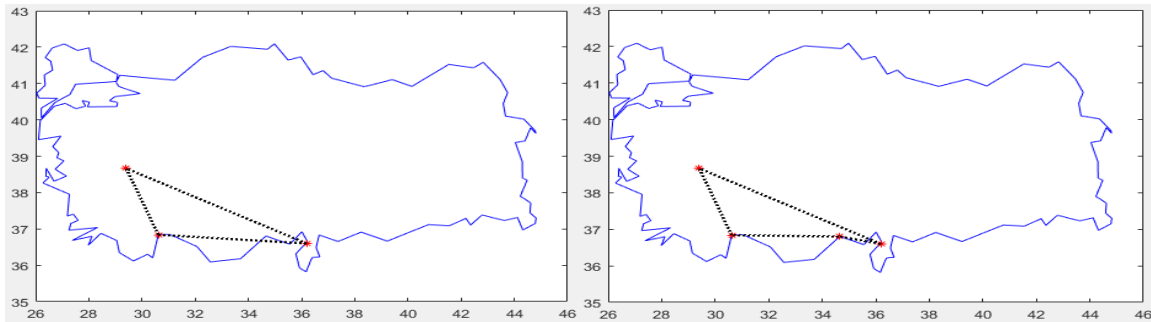
18 adet şehre ait Çizelge 3.2’ de verilen koordinatlar üzerinden Öklid bağıntısı ile bulunan kuş uçuşu mesafelere göre 3, 4 ve 18 şehir için liner optimizasyon algoritması ile yapılan çözüme göre en kısa tur:

3 şehir için; İskenderun Limanı – Antalya Limanı –Uşak Lojistik Köyü – İskenderun Limanı şeklinde belirlenmektedir. Kuş uçuşu toplam mesafe 1486,52 km’ dir.

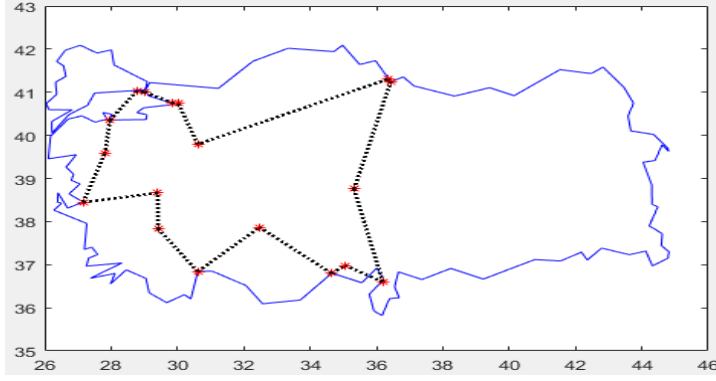
4 şehir için; İskenderun Limanı – Mersin Limanı – Uşak Lojistik Köyü – Antalya Limanı – İskenderun Limanı şeklinde belirlenmektedir. Kuş uçuşu toplam mesafe 1487,33 km’ dir.

18 şehir için; İskenderun Limanı – Mersin limanı – Mersin/Yenice – Kayseri/Boğazköprü – Samsun/Gelemen – Samsun Limanı – Derince Limanı – Konya/Kayacık – Eskişehir/Hasanbey – İzmit/Köseköy – İstanbul/Halkalı – Haydarpaşa Limanı – Bandırma Limanı – Balıkesir/Gökköy – İzmir Limanı – Uşak Lojistik Köyü – Denizli/Kaklık – Antalya Limanı – İskenderun Limanı şeklinde belirlenmektedir. Kuş uçuşu toplam mesafe 2747,20 km’ dir.

4.1.1. Güzergâhın Türkiye haritasında gösterimi



Şekil 4.1. 3 ve 4 noktalı gsp örneğinin koordinatlara göre çözümünün harita üzerinde gösterimi



Şekil 4.2. 18 noktalı gsp örneğinin koordinatlara göre çözümünün harita üzerinde gösterimi

4.2. Lineer Optimizasyon İle Karayolu Mesafelerine Göre Çözümü

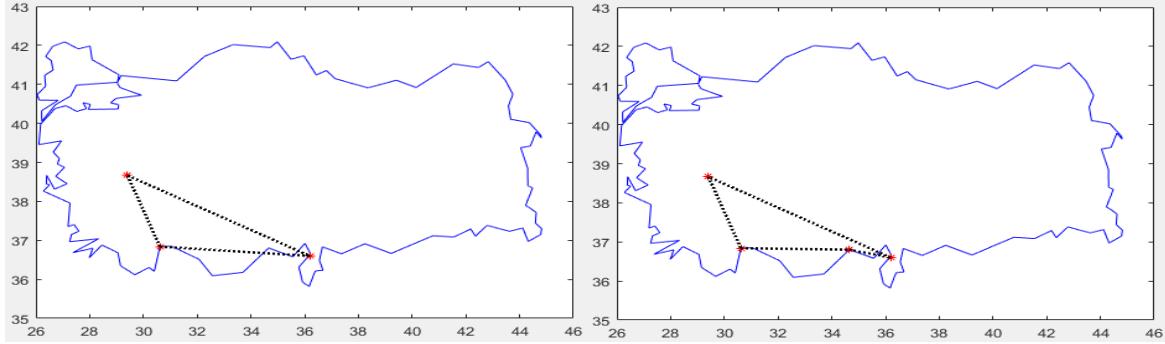
18 adet şehre ait Çizelge 3.2’ de verilen gerçek karayolu mesafeleri üzerinden, lineer optimizasyon algoritması kullanılarak yapılan analizler sonucunda elde edilen optimum tur:

3 şehir için; İskenderun Limanı – Antalya Limanı – Uşak Lojistik Köyü – İskenderun Limanı şeklinde belirlenmektedir. Hesaplanan toplam mesafe 1851 km’ dir.

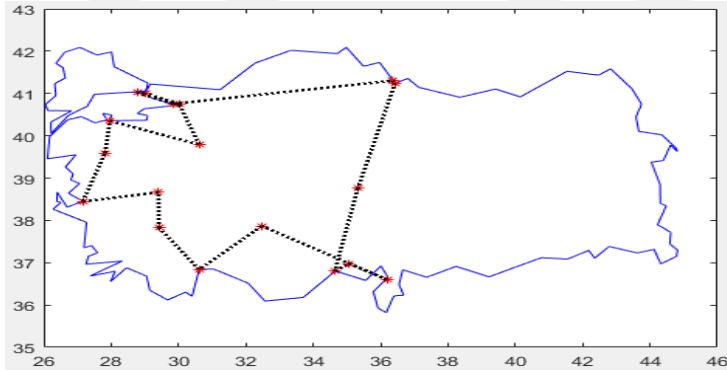
4 şehir için; İskenderun Limanı – Uşak Lojistik Köyü – Antalya Limanı – Mersin Limanı – İskenderun Limanı şeklinde belirlenmektedir. Hesaplanan toplam mesafe 1815 km’ dir.

18 şehir için; İskenderun Limanı – Mersin limanı – Mersin/Yenice – Kayseri/Boğazköprü – Samsun/Gelemen – Samsun Limanı –Konya/Kayacık – Derince Limanı – Eskişehir/Hasanbey – İzmit/Köseköy – Haydarpaşa Limanı – İstanbul/Halkalı –Bandırma Limanı – Balıkesir/Gökköy – İzmir Limanı –Denizli/Kaklık – Antalya Limanı – Uşak Lojistik Köyü – İskenderun Limanı şeklinde belirlenmektedir. Hesaplanan toplam mesafe 3437 km’ dir.

4.2.1. Güzergâhın Türkiye haritasında gösterimi



Şekil 4.3. 3 ve 4 noktalı gps örneğinin gerçek karayolu mesafelere göre çözümünün harita üzerinde gösterimi



Şekil 4.4. 18 noktalı gps örneğinin gerçek karayolu mesafelere göre çözümünün harita üzerinde gösterimi

4.3. Lineer Optimizasyon İle Karayolu Sürelerine Göre Çözümü

18 adet şehre ait Çizelge 3.4' de verilen gerçek karayolu süreleri üzerinden, lineer optimizasyon algoritması kullanılarak yapılan analizler sonucunda elde edilen optimum tur:

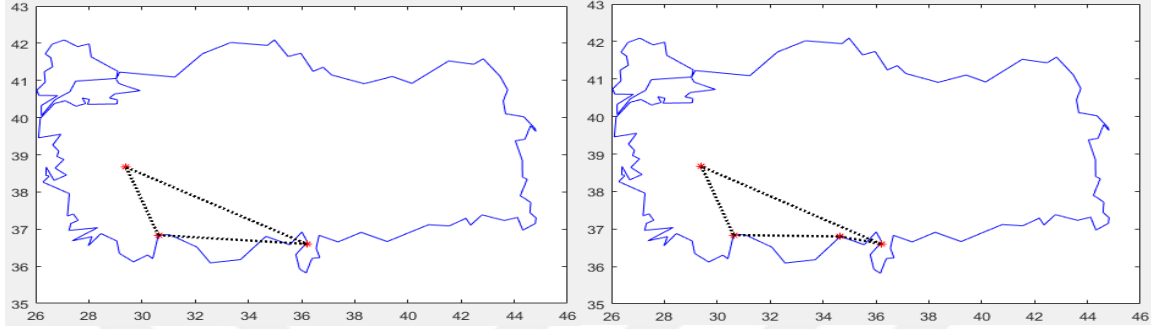
3 şehir için; İskenderun Limanı – Antalya Limanı – Uşak Lojistik Köyü – İskenderun Limanı şeklinde belirlenmektedir. Hesaplanan toplam süre 41,14 saat' tir.

4 şehir için; İskenderun Limanı – Uşak Lojistik Köyü – Antalya Limanı – Mersin Limanı – İskenderun Limanı şeklinde belirlenmektedir. Hesaplanan toplam süre 40,34 saat' tir.

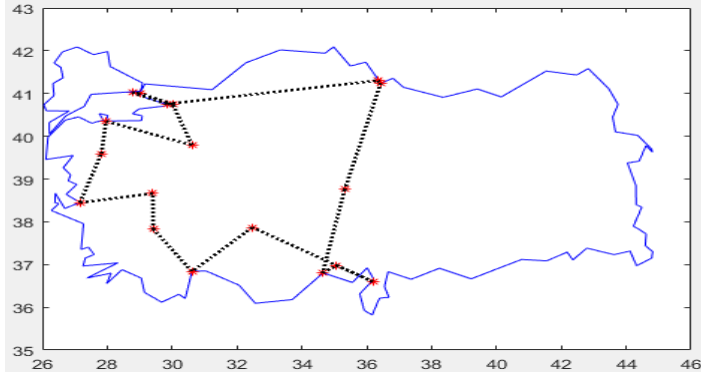
18 şehir için; İskenderun Limanı – Mersin limanı – Mersin/Yenice – Kayseri/Boğazköprü – Samsun/Gelemen – Samsun Limanı – Derince Limanı – Konya/Kayacık – Eskişehir/Hasanbey – İzmit/Köseköy – Haydarpaşa Limanı – İstanbul/Halkalı – Bandırma Limanı – İzmir Limanı – Balıkesir/Gökköy – Denizli/Kaklık – Uşak Lojistik Köyü –

Antalya Limanı – İskenderun Limanı şeklinde belirlenmektedir. Hesaplanan toplam süre 76.38 saat' tir.

4.3.1. Güzergâhın Türkiye haritasında gösterimi



Şekil 4.5. 3 ve 4 noktalı gps örneğinin gerçek karayolu sürelerine göre çözümünün harita üzerinde gösterimi



Şekil 4.6. 18 noktalı gps örneğinin gerçek karayolu sürelerine göre çözümünün harita üzerinde gösterimi

4.4. Lineer Optimizasyon İle Karayolu Birim Maliyetlerine Göre Çözümü

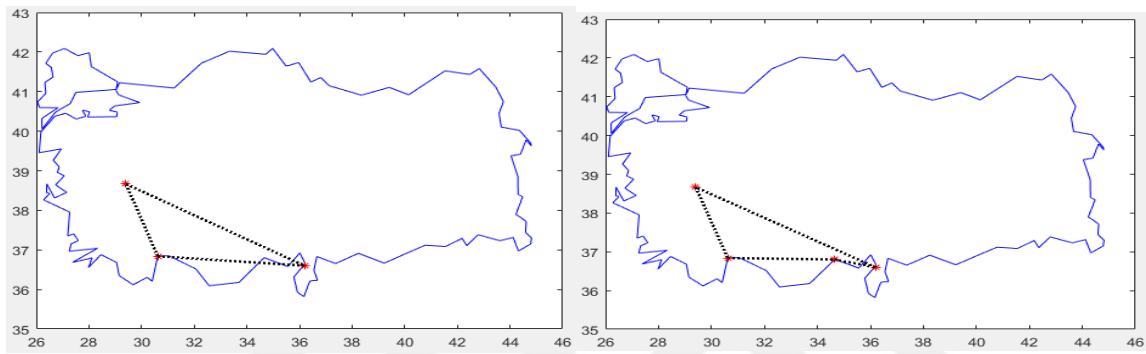
18 adet şehre ait Çizelge 3.5' de verilen gerçek karayolu birim maliyetleri üzerinden, lineer optimizasyon algoritması kullanılarak yapılan analizler sonucunda elde edilen optimum tur:

3 şehir için; İskenderun Limanı – Antalya Limanı – Uşak Lojistik Köyü – İskenderun Limanı şeklinde belirlenmektedir. Hesaplanan toplam maliyet 166,81 TL' dir.

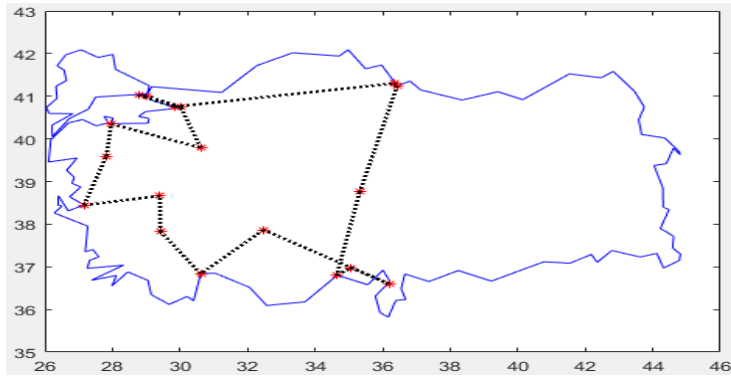
4 şehir için; İskenderun Limanı – Uşak Lojistik Köyü – Antalya Limanı – Mersin Limanı – İskenderun Limanı şeklinde belirlenmektedir. Hesaplanan toplam maliyet 189,83 TL' dir.

18 şehir için; İskenderun Limanı – Mersin limanı – Mersin/Yenice – Kayseri/Boğazköprü – Samsun/Gelemen – Samsun Limanı – Konya/Kayacık — Derince Limanı – Denizli/Kaklık – Eskişehir/Hasanbey – İzmit/Köseköy – İstanbul/Halkalı – Haydarpaşa Limanı – Bandırma Limanı – İzmir Limanı – Balıkesir/Gökköy –Uşak Lojistik Köyü Antalya Limanı – İskenderun Limanı şeklinde belirlenmektedir. Hesaplanan toplam maliyet 639,08 TL’ dir.

4.4.1. Güzergâhın Türkiye haritasında gösterimi



Şekil 4.7. 3 ve 4 noktalı gps örneğinin gerçek karayolu birim maliyetlerine göre çözümünün harita üzerinde gösterimi



Şekil 4.8. 18 noktalı gps örneğinin gerçek karayolu birim maliyetlerine göre çözümünün harita üzerinde gösterimi

4.5. Parçacıklı Sürü Optimizasyon İle Koordinatlara Göre Standart Çözümü

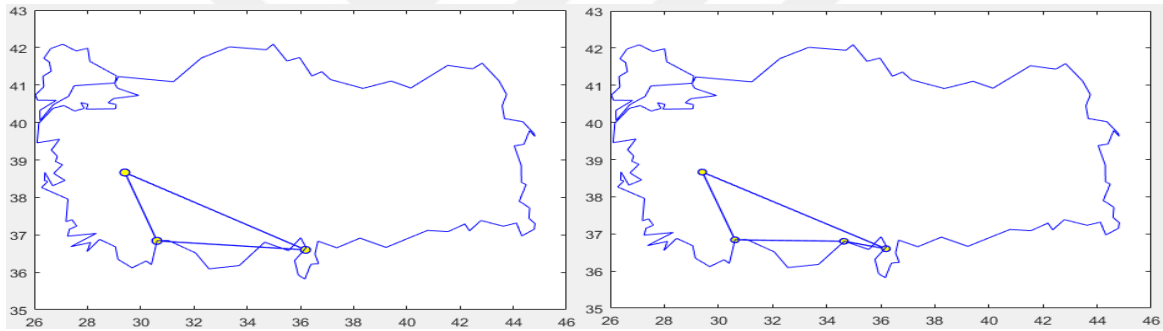
18 adet şehre ait Çizelge 3.2’ de verilen koordinatlar üzerinden Öklid bağıntısı ile bulunan kuş uçuşu mesafelere göre 3, 4 ve 18 şehir için modifiye edilmiş PSO algoritması ile yapılan çözüme göre en kısa tur:

3 şehir için; İskenderun Limanı – Antalya Limanı –Uşak Lojistik Köyü – İskenderun Limanı şeklinde belirlenmektedir. Kuş uçuşu toplam mesafe 1486,52 km’ dir.

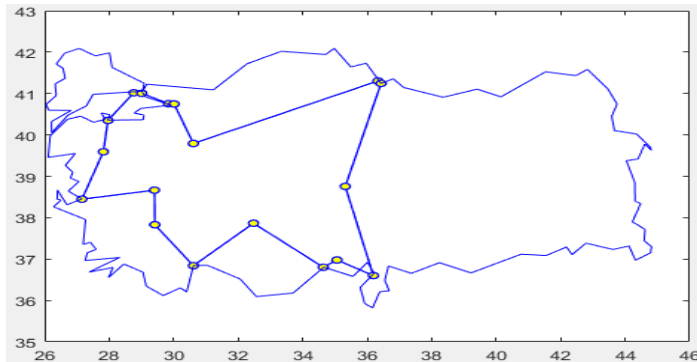
4 şehir için; İskenderun Limanı – Mersin Limanı – Uşak Lojistik Köyü – Antalya Limanı – İskenderun Limanı şeklinde belirlenmektedir. Kuş uçuşu toplam mesafe 1487,33 km’ dir.

18 şehir için; İskenderun Limanı – Mersin limanı – Mersin/Yenice – Kayseri/Boğazköprü – Samsun/Gelemen – Samsun Limanı – Derince Limanı – Konya/Kayacık – Eskişehir/Hasanbey – İzmit/Köseköy – İstanbul/Halkalı – Haydarpaşa Limanı – Bandırma Limanı – Balıkesir/Gökköy – İzmir Limanı – Uşak Lojistik Köyü – Denizli/Kaklık – Antalya Limanı – İskenderun Limanı şeklinde belirlenmektedir. Kuş uçuşu toplam mesafe 2747,20 km’ dir.

4.5.1. Güzergâhın Türkiye haritasında gösterimi



Şekil 4.9. 3 ve 4 noktalı gsp örneğinin koordinatlara göre çözümünün harita üzerinde gösterimi



Şekil 4.10. 18 noktalı gsp örneğinin koordinatlara göre çözümünün harita üzerinde gösterimi

4.6. Parçacıklı Sürü Optimizasyon İle Karayolu Mesafelerine Göre Çözümü

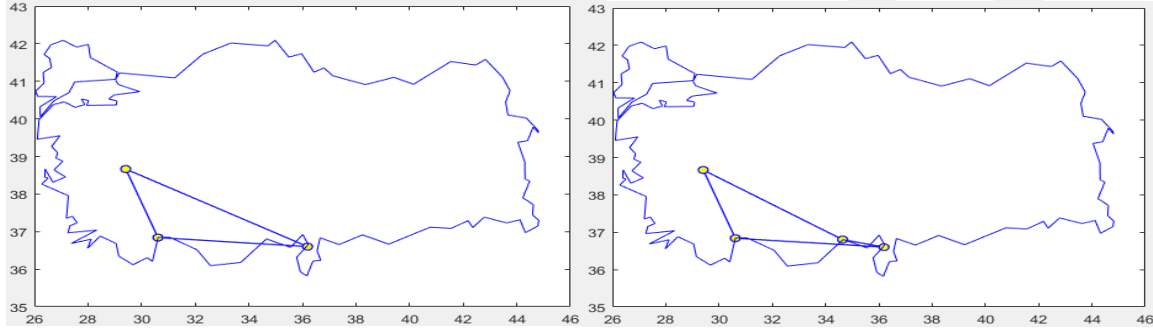
18 adet şehre ait Çizelge 3.2’ de verilen gerçek karayolu mesafeleri üzerinden, modifiye edilmiş PSO algoritması kullanılarak yapılan analizler sonucunda elde edilen optimum tur:

3 şehir için; İskenderun Limanı – Antalya Limanı – Uşak Lojistik Köyü – İskenderun Limanı şeklinde belirlenmektedir. Hesaplanan toplam mesafe 1851 km’ dir.

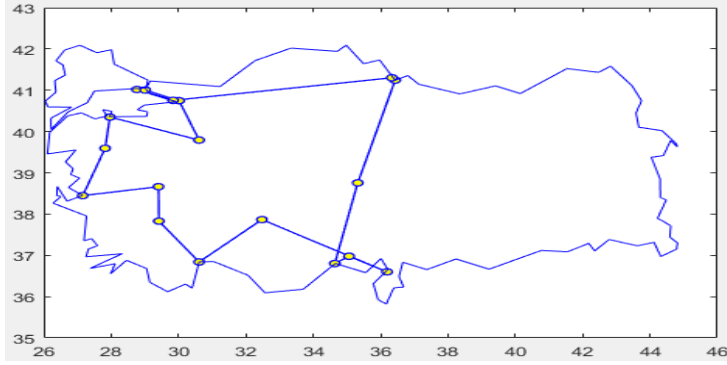
4 şehir için; İskenderun Limanı – Uşak Lojistik Köyü – Antalya Limanı – Mersin Limanı – İskenderun Limanı şeklinde belirlenmektedir. Hesaplanan toplam mesafe 1815 km’ dir.

18 şehir için; İskenderun Limanı – Mersin limanı – Mersin/Yenice – Kayseri/Boğazköprü – Samsun/Gelemen – Samsun Limanı – Derince Limanı – Konya/Kayacık – Eskişehir/Hasanbey – İzmit/Köseköy – İstanbul/Halkalı – Haydarpaşa Limanı – Bandırma Limanı – Balıkesir/Gökköy – İzmir Limanı – Antalya Limanı – Denizli/Kaklık – Uşak Lojistik Köyü – İskenderun Limanı şeklinde belirlenmektedir. Hesaplanan toplam mesafe 3437 km’ dir.

4.6.1. Güzergâhın Türkiye haritasında gösterimi



Şekil 4.11. 3 ve 4 noktalı gsp örneğinin gerçek karayolu mesafelerine göre çözümünün harita üzerinde gösterimi



Şekil 4.12. 18 noktalı gsp örneğinin gerçek karayolu mesafelerine göre çözümünün harita üzerinde gösterimi

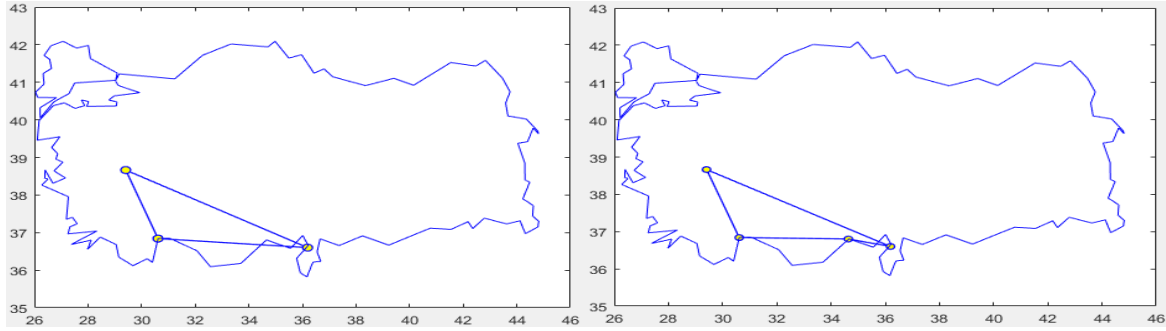
4.7. Parçacıklı Sürü Optimizasyon İle Karayolu Sürelerine Göre Çözümü

18 adet şehre ait Çizelge 3.4’ de verilen gerçek karayolu süreleri üzerinden, modifiye edilmiş PSO algoritması kullanılarak yapılan analizler sonucunda elde edilen optimum tur: 3 şehir için; İskenderun Limanı – Antalya Limanı – Uşak Lojistik Köyü – İskenderun Limanı şeklinde belirlenmektedir. Hesaplanan toplam süre 41,14 saat’ tir.

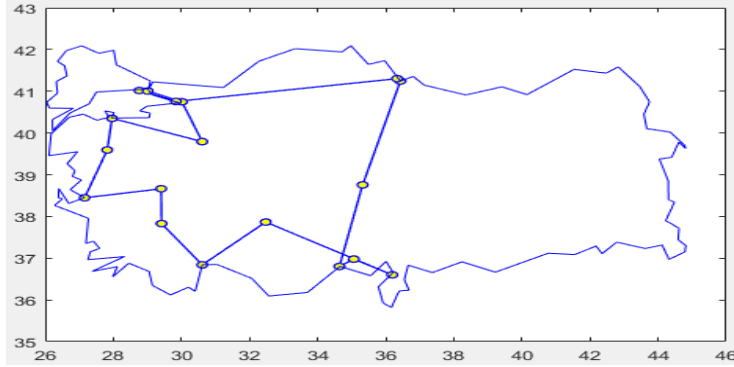
4 şehir için; İskenderun Limanı – Uşak Lojistik Köyü – Antalya Limanı – Mersin Limanı – İskenderun Limanı şeklinde belirlenmektedir. Hesaplanan toplam süre 40,34 saat’ tir.

18 şehir için; İskenderun Limanı – Mersin limanı – Mersin/Yenice – Kayseri/Boğazköprü – Samsun/Gelemen – Samsun Limanı – Derince Limanı – Konya/Kayacık – Eskişehir/Hasanbey – İzmit/Köseköy – İstanbul/Halkalı – Haydarpaşa Limanı – Bandırma Limanı – Balıkesir/Gökköy – İzmir Limanı – Uşak Lojistik Köyü – Denizli/Kaklık – Antalya Limanı – İskenderun Limanı şeklinde belirlenmektedir. Hesaplanan toplam süre 76,38 saat’ tir.

4.7.1. Güzergâhın Türkiye haritasında gösterimi



Şekil 4.13. 3 ve 4 noktalı gsp örneğinin gerçek karayolu sürelerine göre çözümünün harita üzerinde gösterimi



Şekil 4.14. 18 noktalı gsp örneğinin gerçek karayolu sürelerine göre çözümünün harita üzerinde gösterimi

4.8. Parçacıklı Sürü Optimizasyon İle Karayolu Birim Maliyetlerine Göre Çözümü

18 adet şehre ait Çizelge 3.5’ de verilen gerçek karayolu birim maliyetleri üzerinden, modifiye edilmiş PSO algoritması kullanılarak yapılan analizler sonucunda elde edilen optimum tur:

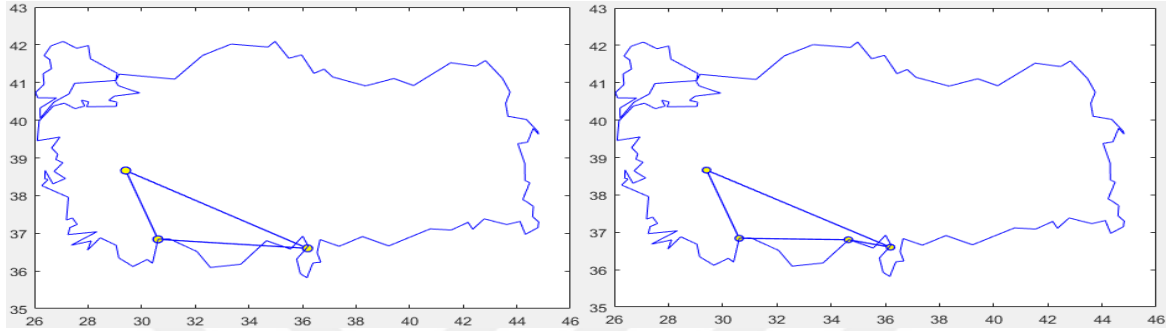
3 şehir için; İskenderun Limanı – Antalya Limanı – Uşak Lojistik Köyü – İskenderun Limanı şeklinde belirlenmektedir. Hesaplanan toplam maliyet 166,81 TL’ dir.

4 şehir için; İskenderun Limanı – Uşak Lojistik Köyü – Antalya Limanı – Mersin Limanı – İskenderun Limanı şeklinde belirlenmektedir. Hesaplanan toplam maliyet 189,83 TL’ dir.

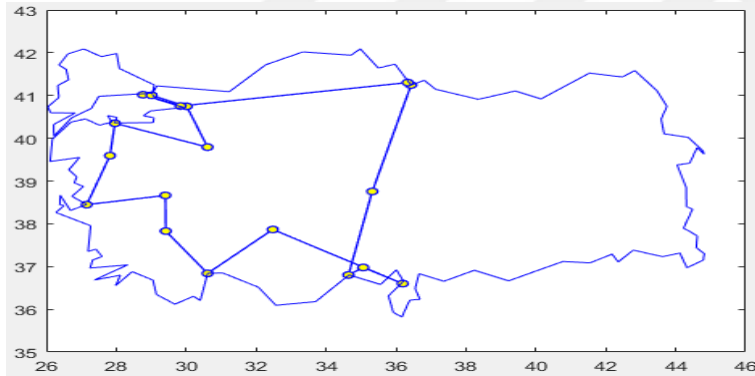
18 şehir için; İskenderun Limanı – Mersin limanı – Mersin/Yenice – Kayseri/Boğazköprü – Samsun/Gelemen – Samsun Limanı – Derince Limanı – Konya/Kayacık – Eskişehir/Hasanbey – İzmit/Köseköy – İstanbul/Halkalı – Haydarpaşa Limanı – Bandırma

Limanı – Balıkesir/Gökköy – İzmir Limanı – Uşak Lojistik Köyü – Denizli/Kaklık – Antalya Limanı – İskenderun Limanı şeklinde belirlenmektedir. Hesaplanan toplam maliyet 639,08 TL’ dir.

4.8.1. Güzergâhın Türkiye haritasında gösterimi



Şekil 4.15. 3 ve 4 noktalı gsp örneğinin gerçek karayolu birim maliyetlerine göre çözümünün harita üzerinde gösterimi



Şekil 4.16. 18 noktalı gsp örneğinin gerçek karayolu birim maliyetlerine göre çözümünün harita üzerinde gösterimi

4.9. Lineer Optimizasyon İle Çok Türlü Mesafelere Göre Çözümü

Lineer optimizasyonda ilk önce 3 şehirli sonra 4 şehirli ve en son olarak ta 18 şehirli örnekler çözülmektedir. Yapılan çalışmada, 3, 4 ve 18 şehre ait mesafeler kullanılarak yapılan analizler sonucu elde edilen optimum mesafe sırası ile 1560 km, 1572 km ve 3282 km olarak hesaplanmaktadır. Şehirler arasında kullanılan ulaşım modları;

3 şehir için;

İskenderun Limanı – Antalya Limanı arası ise denizyolu

Antalya Limanı –Uşak Lojistik Köyü arası karayolu,

Uşak Lojistik Köyü – İskenderun Limanı arası ise karayolu taşımacılığı şeklinde belirlenmektedir.

4 şehir için;

İskenderun Limanı – Mersin Limanı arası denizyolu,

Mersin Limanı – Uşak Lojistik Köyü arası karayolu,

Uşak Lojistik Köyü – Antalya Limanı arası karayolu,

Antalya Limanı – İskenderun Limanı arası ise denizyolu taşımacılığı şeklinde belirlenmektedir.

18 şehir için;

İskenderun Limanı – Mersin Limanı arası denizyolu,

Mersin limanı – Mersin/Yenice arası karayolu,

Mersin/Yenice – Kayseri/Boğazköprü arası karayolu,

Kayseri Boğazköprü – Samsun/Gelemen arası karayolu,

Samsun/Gelemen – Samsun Limanı arası karayolu,

Samsun Limanı – Derince Limanı arası karayolu,

Derince Limanı – Konya/Kayacık arası karayolu,

Konya/Kayacık – Eskişehir/Hasanbey arası karayolu,

Eskişehir/Hasanbey – İzmit/Köseköy arası karayolu,

İzmit/Köseköy – İstanbul/Halkalı arası karayolu,

İstanbul/Halkalı – Haydarpaşa Limanı arası demiryolu,

Haydarpaşa Limanı – Bandırma Limanı arası denizyolu,

Bandırma Limanı – Balıkesir/Gökköy arası demiryolu,

Balıkesir/Gökköy – İzmir Limanı arası karayolu,

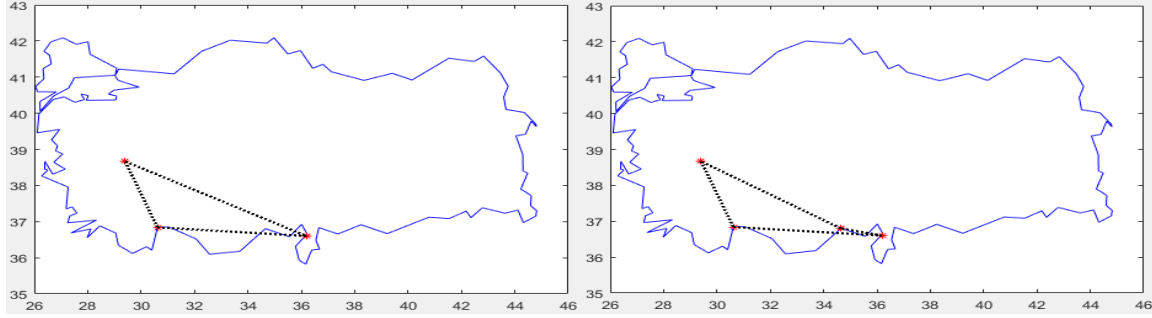
İzmir Limanı – Uşak Lojistik Köyü arası karayolu,

Uşak Lojistik Köyü – Denizli/Kaklık arası karayolu,

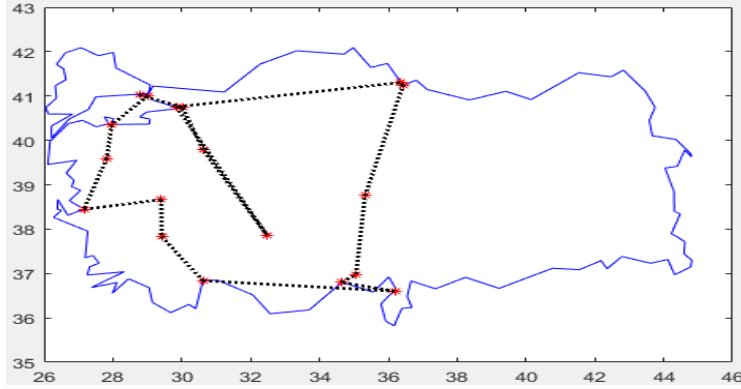
Denizli/Kaklık – Antalya Limanı arası karayolu,

Antalya Limanı – İskenderun Limanı arası denizyolu taşımacılığı şeklinde belirlenmektedir.

4.9.1. Güzergâhın Türkiye Haritasında Gösterimi



Şekil 4.17. 3 ve 4 noktalı gsp örneğinin çok türlü taşımacılık mesafeye göre çözümünün harita üzerinde gösterimi



Şekil 4.18. 18 noktalı gsp örneğinin çok türlü taşımacılık mesafeye göre çözümünün harita üzerinde gösterimi

4.10. Lineer Optimizasyon İle Çok Türlü Sürelere Göre Çözümü

Yapılan çalışmada, 3, 4 ve 18 şehre ait süre değerleri kullanılarak yapılan analizler sonucu elde edilen optimum süre değerleri sırası ile 36,97 saat, 37,75 saat ve 71,65 saat olarak hesaplanmaktadır. Şehirler arasında kullanılan ulaşım modları;

3 şehir için;

İskenderun Limanı – Antalya Limanı arası denizyolu

Antalya Limanı –Uşak Lojistik Köyü arası karayolu,

Uşak Lojistik Köyü – İskenderun Limanı arası ise demiryolu taşımacılığı şeklinde belirlenmektedir.

4 şehir için;

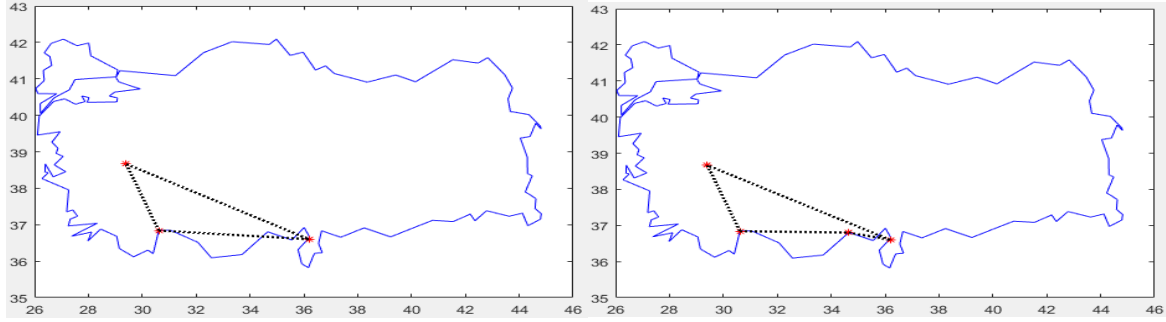
İskenderun Limanı – Mersin Limanı arası denizyolu,

Mersin Limanı – Antalya Limanı arası denizyolu,
Antalya Limanı – Uşak Lojistik Köyü arası karayolu,
Uşak Lojistik Köyü – İskenderun Limanı arası ise demiryolu taşımacılığı şeklinde belirlenmektedir.

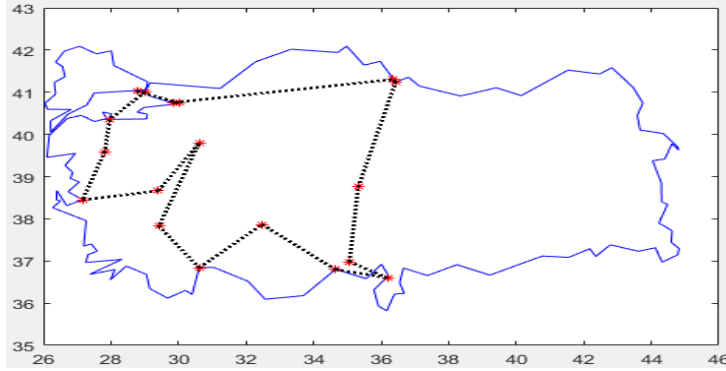
18 şehir için;

İskenderun Limanı – Mersin/Yenice arası demiryolu,
Mersin/Yenice – Kayseri/Boğazköprü arası karayolu,
Kayseri Boğazköprü – Samsun/Gelemen arası karayolu,
Samsun/Gelemen – Samsun Limanı arası karayolu,
Samsun Limanı – Derince Limanı arası karayolu,
Derince Limanı – İzmit/Köseköy arası karayolu,
İzmit/Köseköy – İstanbul/Halkalı arası karayolu,
İstanbul/Halkalı – Haydarpaşa Limanı arası demiryolu,
Haydarpaşa Limanı – Bandırma Limanı arası denizyolu,
Bandırma Limanı – Balıkesir/Gökköy arası demiryolu,
Balıkesir/Gökköy – İzmir Limanı arası karayolu,
İzmir Limanı – Uşak Lojistik Köyü arası karayolu,
Uşak Lojistik Köyü – Eskişehir/Hasanbey arası karayolu,
Eskişehir/Hasanbey – Denizli/Kaklık arası demiryolu,
Denizli/Kaklık – Antalya Limanı arası karayolu,
Antalya Limanı – Konya/Kayacık arası karayolu,
Konya/Kayacık – Mersin limanı arası demiryolu,
Mersin limanı – İskenderun Limanı arası ise denizyolu taşımacılığı şeklinde belirlenmektedir.

4.10.1. Güzergâhın Türkiye haritasında gösterimi



Şekil 4.19. 3 ve 4 noktalı gsp örneğinin çok türlü taşımacılık süreye göre çözümünün harita üzerinde gösterimi



Şekil 4.20. 3 ve 4 noktalı gsp örneğinin çok türlü taşımacılık süreye göre çözümünün harita üzerinde gösterimi

4.11. Lineer Optimizasyon İle Çok Türü Birim Maliyetlere Göre Çözümü

Yapılan çalışmada, 3, 4 ve 18 şehre ait birim maliyet değerleri kullanılarak yapılan analizler sonucu elde edilen optimum birim maliyet değerleri sırası ile 116,21 TL, 125,37 TL ve 394,62 TL olarak hesaplanmaktadır. Şehirler arasında kullanılan ulaşım modları;

3 şehir için;

İskenderun Limanı – Antalya Limanı arası denizyolu,

Antalya Limanı –Uşak Lojistik Köyü arası karayolu,

Uşak Lojistik Köyü – İskenderun Limanı arası ise demiryolu taşımacılığı şeklinde belirlenmektedir.

4 şehir için;

İskenderun Limanı – Mersin Limanı arası denizyolu,

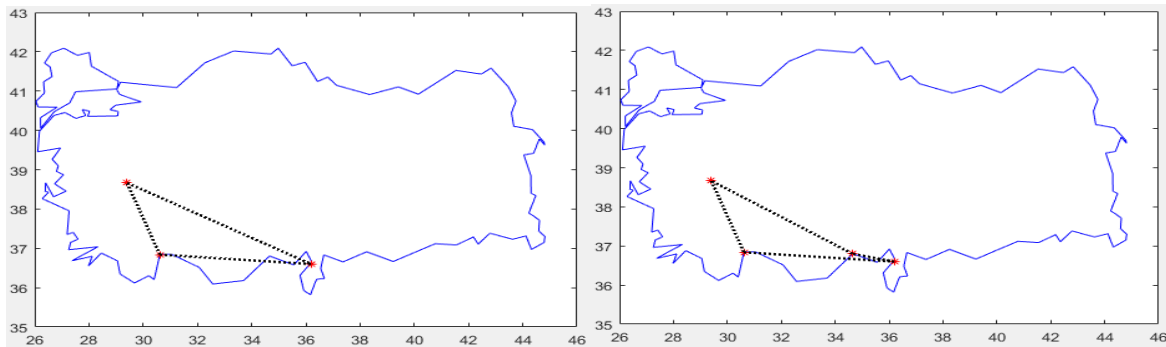
Mersin Limanı – Uşak Lojistik Köyü arası karayolu,

Uşak Lojistik Köyü – Antalya Limanı arası karayolu,
Antalya Limanı – İskenderun Limanı arası ise denizyolu taşımacılığı şeklinde belirlenmektedir.

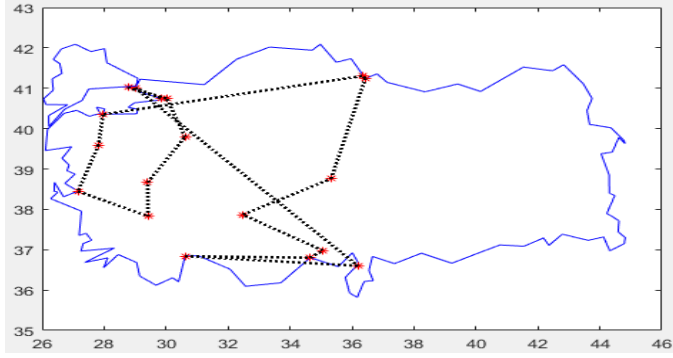
18 şehir için;

İskenderun Limanı – Antalya Limanı arası denizyolu,
Antalya Limanı – Mersin limanı arası denizyolu,
Mersin limanı – Mersin/Yenice arası demiryolu,
Mersin/Yenice – Konya/Kayacık arası demiryolu,
Konya/Kayacık – Kayseri/Boğazköprü arası demiryolu,
Kayseri/Boğazköprü – Samsun/Gelemen arası karayolu,
Samsun/Gelemen – Samsun Limanı arası demiryolu,
Samsun Limanı – Bandırma Limanı arası denizyolu,
Bandırma Limanı – Balıkesir/Gökköy arası demiryolu,
Balıkesir/Gökköy – İzmir Limanı arası demiryolu,
İzmir Limanı – Denizli/Kaklık arası demiryolu,
Denizli/Kaklık – Uşak Lojistik Köyü arası karayolu,
Uşak Lojistik Köyü – Eskişehir/Hasanbey arası demiryolu,
Eskişehir/Hasanbey– İzmit/Köseköy arası demiryolu,
İzmit/Köseköy – İstanbul/Halkalı arası demiryolu,
İstanbul/Halkalı – Derince Limanı arası demiryolu,
Derince Limanı – Haydarpaşa Limanı arası demiryolu,
Haydarpaşa Limanı – İskenderun Limanı arası ise denizyolu taşımacılığı şeklinde belirlenmektedir.

4.11.1. Güzergâhın Türkiye haritasında gösterimi



Şekil 4.21. 3 ve 4 noktalı gsp örneğinin çok türlü taşımacılık birim maliyete göre çözümünün harita üzerinde gösterimi



Şekil 4.22. 18 noktalı gsp örneğinin çok türlü taşımacılık birim maliyete göre çözümünün harita üzerinde gösterimi

4.12. Parçacıklı Sürü Optimizasyonu İle Çok Türlü Mesafelere Göre Çözümü

Yapılan çalışmada, 3, 4 ve 18 şehre ait mesafeler kullanılarak yapılan analizler sonucu elde edilen optimum mesafe sırası ile 1560 km, 1572 km ve 3295 km olarak hesaplanmaktadır.

Şehirlerarasında kullanılan ulaşım modları;

3 şehir için;

İskenderun Limanı – Uşak Lojistik Köyü arası karayolu,

Uşak Lojistik Köyü – Antalya Limanı arası karayolu,

Antalya Limanı – İskenderun Limanı arası ise denizyolu taşımacılığı şeklinde belirlenmektedir.

4 şehir için;

İskenderun Limanı – Antalya Limanı arası denizyolu,

Antalya Limanı – Uşak Lojistik Köyü arası karayolu,

Uşak Lojistik Köyü – Mersin Limanı arası karayolu,

Mersin Limanı – İskenderun Limanı arası ise denizyolu taşımacılığı şeklinde belirlenmektedir.

18 şehir için;

İskenderun Limanı – Mersin limanı arası denizyolu,

Mersin limanı – Kayseri/Boğazköprü arası karayolu,

Kayseri/Boğazköprü – Samsun/Gelemen arası karayolu,

Samsun/Gelemen – Samsun Limanı arası karayolu,

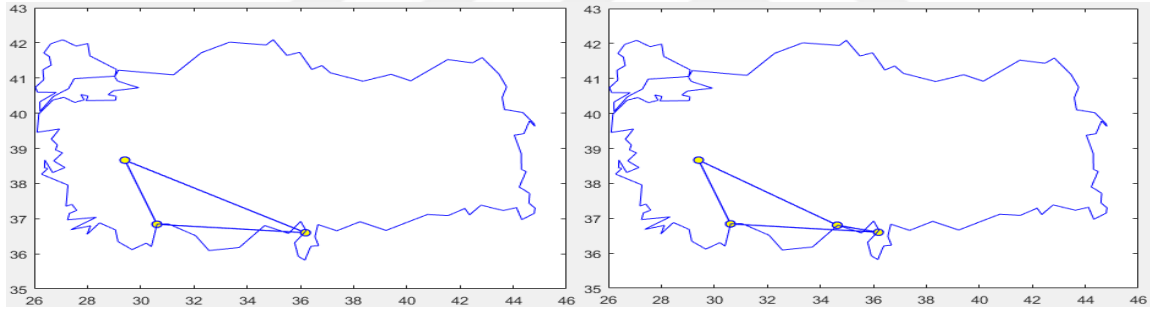
Samsun Limanı – Derince Limanı arası karayolu,

Derince Limanı – İzmit/Köseköy arası karayolu,

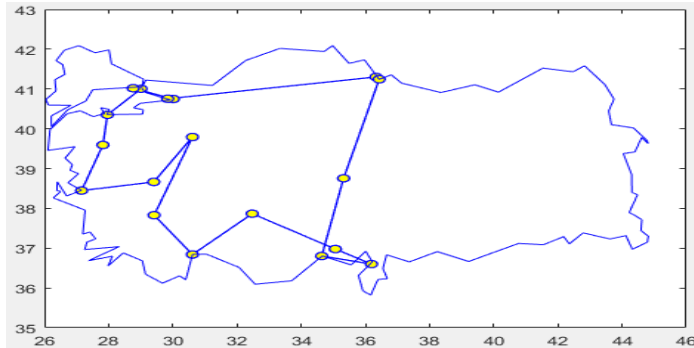
İzmit/Köseköy – İstanbul/Halkalı arası karayolu,

İstanbul/Halkalı – Haydarpaşa Limanı arası demiryolu,
 Haydarpaşa Limanı – Bandırma Limanı arası denizyolu,
 Bandırma Limanı – Balıkesir/Gökköy arası demiryolu,
 Balıkesir/Gökköy – İzmir Limanı arası karayolu,
 İzmir Limanı – Uşak Lojistik Köyü arası karayolu,
 Uşak Lojistik Köyü – Eskişehir/Hasanbey arası karayolu,
 Eskişehir/Hasanbey – Denizli/Kaklık arası demiryolu,
 Denizli/Kaklık – Antalya Limanı arası karayolu,
 Antalya Limanı – Konya/Kayacık arası karayolu,
 Konya/Kayacık – Mersin/Yenice arası karayolu,
 Mersin/Yenice – İskenderun Limanı arası ise karayolu taşımacılığı şeklinde belirlenmektedir.

4.12.1. Güzergâhın Türkiye haritasında gösterimi



Şekil 4.23. 3 ve 4 noktalı gsp örneğinin çok türlü taşımacılık mesafeye göre çözümünün harita üzerinde gösterimi



Şekil 4.24. 18 noktalı gsp örneğinin çok türlü taşımacılık mesafeye göre çözümünün harita üzerinde gösterimi

4.13. Parçacıklı Sürü Optimizasyonu İle Çok Türlü Sürelere Göre Çözümü

Yapılan çalışmada, 3, 4 ve 18 şehre ait süre değerleri kullanılarak yapılan analizler sonucu elde edilen optimum süre değerleri sırası ile 36,97 saat, 37,75 saat ve 71,65 saat olarak hesaplanmaktadır. Şehirlerarasında kullanılan ulaşım modları;

3 şehir için;

İskenderun Limanı – Uşak Lojistik Köyü arası demiryolu,
Uşak Lojistik Köyü – Antalya Limanı arası ise karayolu,
Antalya Limanı – İskenderun Limanı arası ise denizyolu taşımacılığı şeklinde belirlenmektedir.

4 şehir için;

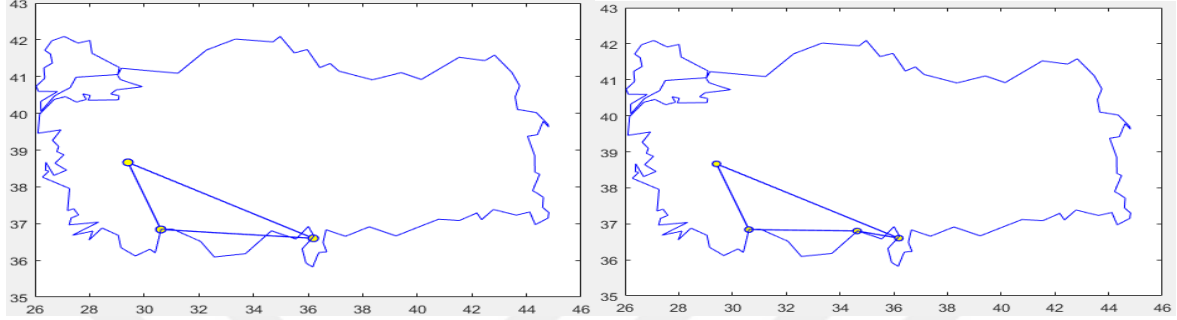
İskenderun Limanı – Uşak Lojistik Köyü arası demiryolui,
Uşak Lojistik Köyü – Antalya Limanı arası karayolu,
Antalya Limanı – Mersin Limanı arası denizyolu,
Mersin Limanı – İskenderun Limanı arası ise denizyolu taşımacılığı şeklinde belirlenmektedir.

18 şehir için;

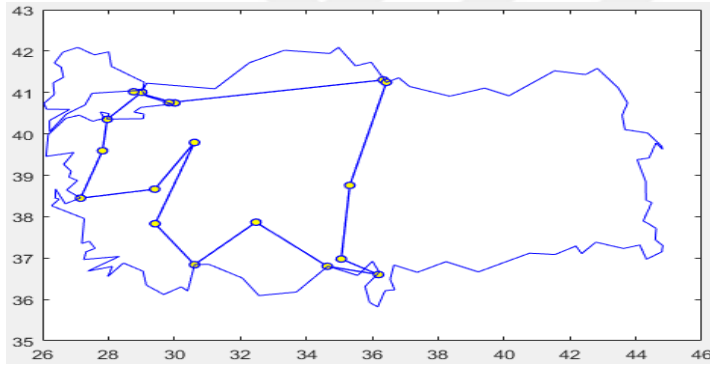
İskenderun Limanı – Mersin limanı arası denizyolu,
Mersin limanı – Konya/Kayacık arası demiryolu,
Konya/Kayacık – Antalya Limanı arası karayolu,
Antalya Limanı – Denizli/Kaklık arası karayolu,
Denizli/Kaklık – Eskişehir/Hasanbey arası demiryolu,
Eskişehir/Hasanbey – Uşak Lojistik Köyü arası karayolu,
Uşak Lojistik Köyü – İzmir Limanı arası karayolu,
İzmir Limanı – Balıkesir/Gökköy arası karayolu,
Balıkesir/Gökköy – Bandırma Limanı arası demiryolu,
Bandırma Limanı – Haydarpaşa Limanı arası denizyolu,
Haydarpaşa Limanı – İstanbul/Halkalı arası demiryolu,
İstanbul/Halkalı – İzmit/Köseköy arası karayolu,
İzmit/Köseköy – Derince Limanı arası demiryolu,
Derince Limanı – Samsun Limanı arası karayolu,
Samsun Limanı – Samsun/Gelemen arası karayolu,
Samsun/Gelemen – Kayseri Boğazköprü arası karayolu,
Kayseri Boğazköprü – Mersin/Yenice arası demiryolu,

Mersin/Yenice – İskenderun Limanı arası ise demiryolu taşımacılığı şeklinde belirlenmektedir.

4.13.1. Güzergâhın Türkiye Haritasında Gösterimi



Şekil 4.25. 3 ve 4 noktalı gsp örneğinin çok türlü taşımacılık süreye göre çözümünün harita üzerinde gösterimi



Şekil 4.26. 18 noktalı gsp örneğinin çok türlü taşımacılık süreye göre çözümünün harita üzerinde gösterimi

4.14. Parçacıklı Sürü Optimizasyonu İle Çok Türlü Birim Maliyetlere Göre Çözümü

Yapılan çalışmada, 3, 4 ve 18 şehre ait birim maliyet değerleri kullanılarak yapılan analizler sonucu elde edilen optimum birim maliyet değerleri sırası ile 116,21 TL, 125,37 TL ve 394,62 TL olarak hesaplanmaktadır. Şehirler arasında kullanılan ulaşım modları; 3 şehir için;

İskenderun Limanı – Antalya Limanı arası denizyolu,

Antalya Limanı –Uşak Lojistik Köyü arası karayolu,

Uşak Lojistik Köyü – İskenderun Limanı arası ise karayolu taşımacılığı şeklinde belirlenmektedir.

4 şehir için;

İskenderun Limanı – Mersin Limanı arası ise denizyolu,

Mersin Limanı – Uşak Lojistik Köyü arası karayolu,

Uşak Lojistik Köyü – Antalya Limanı arası karayolu,

Antalya Limanı – İskenderun Limanı arası ise denizyolu taşımacılığı şeklinde belirlenmektedir.

18 şehir için;

İskenderun Limanı – Haydarpaşa Limanı arası denizyolu,

Haydarpaşa Limanı – Derince Limanı arası demiryolu,

Derince Limanı – İstanbul/Halkalı arası demiryolu,

İstanbul/Halkalı – İzmit/Köseköy arası demiryolu,

İzmit/Köseköy – Eskişehir/Hasanbey arası demiryolu,

Eskişehir/Hasanbey – Uşak Lojistik Köyü arası demiryolu,

Uşak Lojistik Köyü – İzmir Limanı arası demiryolu,

İzmir Limanı – Balıkesir/Gökköy arası demiryolu,

Balıkesir/Gökköy – Bandırma Limanı arası denizyolu,

Bandırma Limanı – Samsun Limanı arası denizyolu,

Samsun Limanı – Samsun/Gelemen arası karayolu,

Samsun/Gelemen – Kayseri/Boğazköprü arası karayolu,

Kayseri/Boğazköprü – Konya/Kayacık arası demiryolu,

Kayseri/Boğazköprü – Konya/Kayacık arası demiryolu,

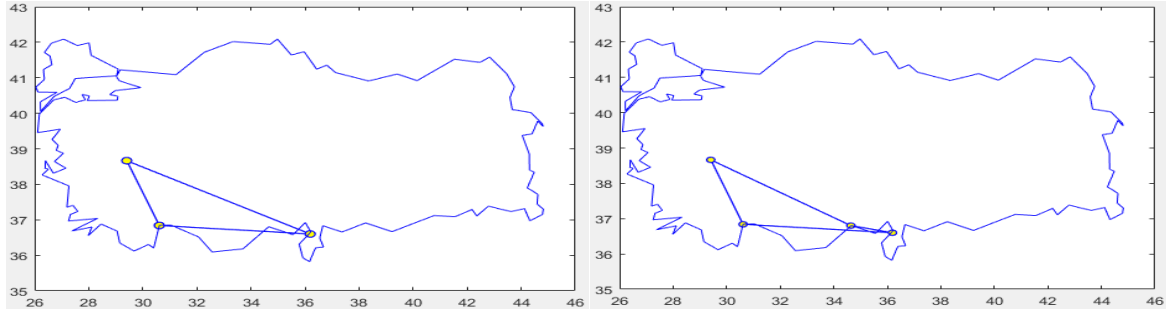
Konya/Kayacık – Mersin/Yenice arası demiryolu,

Mersin/Yenice – Mersin limanı arası demiryolu,

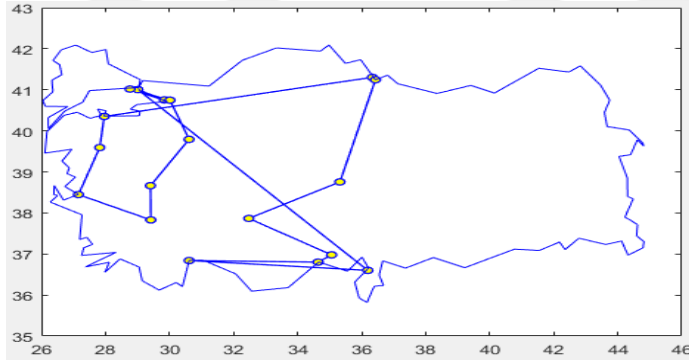
Mersin limanı – Antalya Limanı arası denizyolu,

Antalya Limanı – İskenderun Limanı arası ise denizyolu taşımacılığı şeklinde belirlenmektedir.

4.14.1. Güzergâhın Türkiye haritasında gösterimi



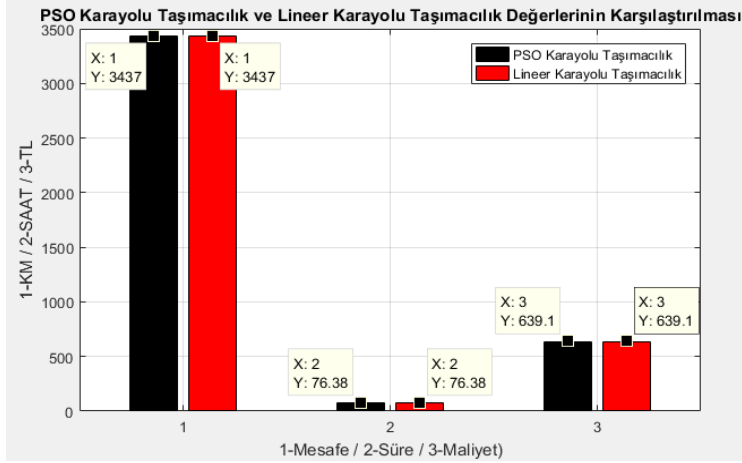
Şekil 4.27. 3 ve 4 noktalı gps örneğinin çok türlü taşımacılık birim maliyete göre çözümünün harita üzerinde gösterimi



Şekil 4.28. 18 noktalı gps örneğinin çok türlü taşımacılık birim maliyete göre çözümünün harita üzerinde gösterimi

4.15. 18 nokta için Karayolu Taşımacılığın Mesafe, Süre ve Birim Maliyetlerine Göre Karşılaştırılması

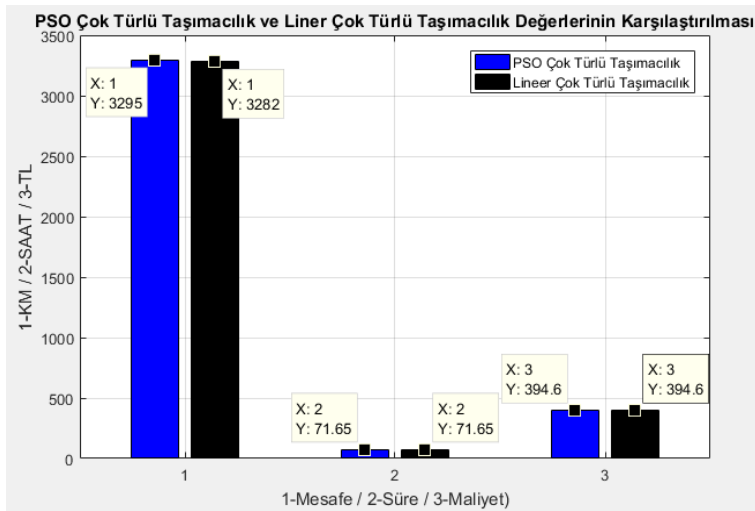
Bu çalışmada, 18 şehir için karayolu taşımacılığı ile mesafe, süre ve birim maliyet değerleri hesaplanmıştır. Lineer Optimizasyon ile yapılan analizler sonucunda 18 şehir için elde edilen değerler; mesafe 3437 km, süre 76,38 saat ve birim maliyet 639,1 TL olarak hesaplanmaktadır. PSO için yapılan analizler sonucunda 18 şehir için elde edilen değerler; mesafe 3437 km, süre 76,38 saat ve birim maliyet 639,1 TL olarak hesaplanmıştır. Lineer Optimizasyon ve PSO yöntemleri sonucu elde edilen değerler birbirleri ile kıyaslandığında; 18 nokta için Lineer Optimizasyon ve PSO değerlerin aynı olduğu görülmektedir. Karşılaştırmanın grafiği Şekil 4.29' da verilmektedir.



Şekil 4.29. Karayolu taşımacılığında mesafe, süre ve birim maliyete göre karşılaştırma

4.16. 18 nokta için Çok Türlü Taşımacılık İçin Mesafe, Süre ve Birim Maliyetlerinin Karşılaştırılması

Bu çalışmada, 18 şehir için çok türlü taşımacılık ile mesafe, süre ve birim maliyet değerleri hesaplanmıştır. Lineer Optimizasyon ile yapılan analizler sonucunda 18 şehir için elde edilen değerler; mesafe 3282 km, süre 71,65 saat ve birim maliyet 394,62 TL olarak hesaplanmaktadır. PSO için yapılan analizler sonucunda 18 şehir için elde edilen değerler; mesafe 3295 km, süre 71,65 saat ve birim maliyet 394,62 TL olarak hesaplanmıştır. Lineer Optimizasyon ve PSO yöntemleri sonucu elde edilen değerler birbirleri ile kıyaslandığında; 18 şehir arasındaki mesafe değeri Lineer Optimizasyonda daha iyi çıktığı, diğer değerlerin ise aynı olduğu görülmektedir. Karşılaştırmanın grafiği Şekil 4.30’ da verilmektedir.



Şekil 4.30. Çok türlü taşımacılıkta mesafe, süre ve birim maliyete göre karşılaştırma

5. SONUÇLAR VE ÖNERİLER

Bu çalışmada, 18 nokta ve bunlar arasından seçilen 3 ve 4 nokta için çok türlü taşımacılık ile mesafe, süre ve birim maliyet değerleri hesaplanmıştır. Yapılan çalışmada Lineer Optimizasyon ve Parçacıklı Sürü Optimizasyonu (PSO) yöntemi kullanılmıştır.

Lineer Optimizasyon ile yapılan analizler sonucunda 3 nokta için elde edilen değerler; mesafe 1560 km, süre 36,97 saat ve birim maliyet 116,21 TL olarak hesaplanmıştır. 4 nokta için elde edilen değerler; mesafe 1572 km, süre 37,75 saat ve birim maliyet 125,37 TL olarak hesaplanmıştır. 18 nokta için elde edilen değerler ise; mesafe 3282 km, süre 71,65 saat ve birim maliyet 394,62 TL olarak hesaplanmaktadır. PSO için yapılan analizler sonucunda 3 nokta için elde edilen değerler; mesafe 1560 km, süre 36,97 saat ve birim maliyet 116,21 TL olarak hesaplanmıştır. 4 nokta için elde edilen değerler; mesafe 1572 km, süre 37,75 saat ve birim maliyet 125,37 TL olarak hesaplanmıştır. 18 nokta için elde edilen değerler ise; mesafe 3295 km, süre 71,65 saat ve birim maliyet 394,62 TL olarak hesaplanmıştır. Lineer Optimizasyon ve PSO yöntemleri sonucu elde edilen değerler birbirleri ile kıyaslandığında; 18 nokta arasındaki mesafe değeri Lineer Optimizasyonda daha iyi çıktığı, diğer değerlerin ise aynı olduğu görülmektedir. Lineer optimizasyon PSO ya göre süre bakımından daha çabuk ve direk olarak minimum sonuçları vermektedir. PSO iterasyon üzerine olduğu için kaçınıcı iterasyonda minimum değeri bulacağını kestirmek güçtür ve sürekli algoritmayı yeniden çalıştırarak optimal değerli bulmaya çalışılır. Lineer optimizasyon bu konuda daha avantajlı durumdadır. Lineer optimizasyon problemin amaç fonksiyonunu minimumda tutmak ve bunu sağlamak için kısıtlar girildiğinden direk olarak en optimal sonuçları vermektedir.

Genel olarak yapılan çözümlerde multimodal taşımacılığın, hem mesafeye, hem süreye hem de maliyete göre unimodal karayolu taşımacılığından daha avantajlı olduğu görülmüştür. Mesafe olarak teoride kuş uçuşu ilerlemeden dolayı en avantajlısı denizyolu olmasına rağmen uygulamada her yerde ulaşılabilir olmaması denizyolu taşımacılığını en son plana itmektedir.

Ülkemizde Lojistik sektörü ağırlıklı olarak karayolu taşımacılığı ile hizmet vermektedir. Çok türlü taşımacılığın yaygın olarak kullanılmaya başlanması ve karayolu yükünün hafiflemesiyle çevre kirliliği de azalacaktır. Multimodal taşımacılık yaygın taşımacılık

türlerindeki trafik yoğunluğunun azalmasında olumlu katkı sağlayacaktır. Geliştirilen programın kullanılmaya başlamasıyla çok türlü taşımacılığa teşvik artacak, çok türlü taşımacılık kullanımının artmasına paralel olarak alt yapısı daha da kuvvetlenecek ve birim maliyetler daha da azalacaktır.



KAYNAKLAR

- Akandere, G., (2013). *Lojistik köylerin etkin yönetimi: konya iline yönelik bir model önerisi*. Selçuk Üniversitesi Sosyal Bilimler Enstitüsü Yüksek Lisans Tezi, 5, Konya.
- Avraamidou, S., & Pistikopoulos, E. N. (2019). A Multi-Parametric optimization approach for bilevel mixed-integer linear and quadratic programming problems. *Computers & Chemical Engineering*.
- Bertazzi, L., Coelho, L. C., De Maio, A., & Laganà, D. (2019). A matheuristic algorithm for the multi-depot inventory routing problem. *Transportation Research Part E: Logistics and Transportation Review*, 122, 524-544.
- Brunner, J. O. (2014). Rescheduling of flights during ground delay programs with consideration of passenger and crew connections. *Transportation Research Part E: Logistics and Transportation Review*, 72, 236-252.
- Cao, L., Xu, L., & Goodman, E. D. (2018). A neighbor-based learning particle swarm optimizer with short-term and long-term memory for dynamic optimization problems. *Information Sciences*, 453, 463-485.
- Chen, J., & Shi, J. (2019). A Multi-compartment Vehicle Routing Problem with Time Windows for Urban Distribution—A Comparison Study on Particle Swarm Optimization Algorithms. *Computers & Industrial Engineering*.
- Chen, S. H. (2015). Minimization of the total traveling distance and maximum distance by using a transformed-based encoding EDA to solve the multiple traveling salesmen problem. *Mathematical Problems in Engineering*, 2015.
- Chen, X., Zhou, Y., Tang, Z., & Luo, Q. (2017). A hybrid algorithm combining glowworm swarm optimization and complete 2-opt algorithm for spherical travelling salesman problems. *Applied Soft Computing*, 58, 104-114.
- Chiang, T. A., Che, Z. H., & Cui, Z. (2014). Designing a multistage supply chain in cross-stage reverse logistics environments: application of particle swarm optimization algorithms. *The Scientific World Journal*, 2014.
- Dell'Amico, M., Hadjicostantinou, E., Iori, M., & Novellani, S. (2014). The bike sharing rebalancing problem: Mathematical formulations and benchmark instances. *Omega*, 45, 7-19.
- Dorigo, M., Member, S., IEEE, Gambardella, M., (1997). Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions On Evolutionary Computation*, 1(1), 53-66.

- Elloumi, W., El Abed, H., Abraham, A., & Alimi, A. M. (2014). A comparative study of the improvement of performance using a PSO modified by ACO applied to TSP. *Applied Soft Computing*, 25, 234-241.
- Funke, J., & Kopfer, H. (2016). A model for a multi-size inland container transportation problem. *Transportation Research Part E: Logistics and Transportation Review*, 89, 70-85.
- Garzón-Garnica, E. A., Cruz-Benítez, D. P., Badillo-Valenzuela, O. D., Sánchez-Partida, D., & Martínez-Flores, J. L. (2015). Automated data acquisition for a large scale capacitated vehicle routing problem. *IFAC-PapersOnLine*, 48(3), 1393-1398.
- Gupta, S. D., Tobin, J. K., & Pavel, L. (2016). A two-step linear programming model for energy-efficient timetables in metro railway networks. *Transportation Research Part B: Methodological*, 93, 57-74.
- Hungerländer, P., & Truden, C. (2018). Efficient and Easy-to-Implement Mixed-Integer Linear Programs for the Traveling Salesperson Problem with Time Windows. *Transportation research procedia*, 30, 157-166.
- Leggieri, V., & Haouari, M. (2017). A practical solution approach for the green vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review*, 104, 97-112.
- Lei, K., Zhu, X., Hou, J., & Huang, W. (2014). Decision of multimodal transportation scheme based on swarm intelligence. *Mathematical Problems in Engineering*, 2014.
- Li, Z., & Li, Z. (2016). Linear programming-based scenario reduction using transportation distance. *Computers & Chemical Engineering*, 88, 50-58.
- Lin, S., (1965). *Computer solutions of the traveling salesman problem*. Manuscript received, August 18, 1965.
- Lin, M., Zhong, Y., Lin, J., & Lin, X. (2018). Discrete Bird Swarm Algorithm Based on Information Entropy Matrix for Traveling Salesman Problem. *Mathematical Problems in Engineering*, 2018.
- Lin, W., Lian, Z., Gu, X., & Jiao, B. (2014). A local and global search combined particle swarm optimization algorithm and its convergence analysis. *Mathematical Problems in Engineering*, 2014.
- Ma, J., Li, X., Zhou, F., & Hao, W. (2017). Designing optimal autonomous vehicle sharing and reservation systems: A linear programming approach. *Transportation Research Part C: Emerging Technologies*, 84, 124-141.

- Marinakis, Y. (2015). An improved particle swarm optimization algorithm for the capacitated location routing problem and for the location routing problem with stochastic demands. *Applied Soft Computing*, 37, 680-701.
- Marinakis, Y., Marinaki, M., & Migdalas, A. (2019). A Multi-Adaptive Particle Swarm Optimization for the Vehicle Routing Problem with Time Windows. *Information Sciences*, 481, 311-329.
- Mavrovouniotis, M., Li, C., & Yang, S. (2017). A survey of swarm intelligence for dynamic optimization: Algorithms and applications. *Swarm and Evolutionary Computation*, 33, 1-17.
- Murakami, K. (2017). A new model and approach to electric and diesel-powered vehicle routing. *Transportation Research Part E: Logistics and Transportation Review*, 107, 23-37.
- Norouzi, N., Sadegh-Amalnick, M., & Alinaghiyan, M. (2015). Evaluating of the particle swarm optimization in a periodic vehicle routing problem. *Measurement*, 62, 162-169.
- Roberti, R., & Wen, M. (2016). The electric traveling salesman problem with time windows. *Transportation Research Part E: Logistics and Transportation Review*, 89, 32-52.
- Saud, S., Kodaz, H., & Babaoğlu, İ. (2018). Solving Travelling Salesman Problem by Using Optimization Algorithms. *KnE Social Sciences*, 3(1), 17-32.
- Staněk, R., Greistorfer, P., Ladner, K., & Pferschy, U. (2019). Geometric and LP-based heuristics for angular travelling salesman problems in the plane. *Computers & Operations Research*.
- Tamannaie, M., & Rasti-Barzoki, M. (2019). Mathematical programming and solution approaches for minimizing tardiness and transportation costs in the supply chain scheduling problem. *Computers & Industrial Engineering*, 127, 643-656.
- Wang, B., Brown, D., Zhang, X., Li, H., Gao, Y., & Cao, J. (2014). Polygonal approximation using integer particle swarm optimization. *Information Sciences*, 278, 311-326.
- Yang, L., & Zhou, X. (2017). Optimizing on-time arrival probability and percentile travel time for elementary path finding in time-dependent transportation networks: Linear mixed integer programming reformulations. *Transportation Research Part B: Methodological*, 96, 68-91.
- Yuan, M., Zhang, Q., Wang, B., Liang, Y., & Zhang, H. (2019). A mixed integer linear programming model for optimal planning of bicycle sharing systems: A case study in Beijing. *Sustainable Cities and Society*, 101515.

Zhang, Z., Wang, G. G., Zou, K., & Zhang, J. (2014). A solution quality assessment method for swarm intelligence optimization algorithms. *The Scientific World Journal*, 2014.

Zhong, Y., Lin, J., Wang, L., & Zhang, H. (2018). Discrete comprehensive learning particle swarm optimization algorithm with Metropolis acceptance criterion for traveling salesman problem. *Swarm and Evolutionary Computation*, 42, 77-88.



EKLER

EK-1. PSO Ana Algoritması: Multimodal Mesafe Örneği

```

clc;
clear;
clear all;

global NFE;
NFE=0;

model=CreateModel();

CostFunction=@(x) MyCost(x,model);

nVar=model.N;
VarSize=[1 nVar];
VarMin=0;
VarMax=1;
MaxIt=500;
nPop=100;
w=1;
wdamp=0.99;
c1=0.5;
c2=0.5;

VelMax=0.1*(VarMax-VarMin);
VelMin=-VelMax;

empty_particle.Position=[];
empty_particle.Cost=[];
empty_particle.Sol=[];
empty_particle.Velocity=[];
empty_particle.Best.Position=[];
empty_particle.Best.Cost=[];
empty_particle.Best.Sol=[];

particle=repmat(empty_particle,nPop,1);

GlobalBest.Cost=inf;

for i=1:nPop
particle(i).Position=unifrnd(VarMin,VarMax,VarSize);
particle(i).Velocity=zeros(VarSize);
[particle(i).Cost particle(i).Sol]=CostFunction(particle(i).Position);
particle(i).Best.Position=particle(i).Position;
particle(i).Best.Cost=particle(i).Cost;
particle(i).Best.Sol=particle(i).Sol;
if particle(i).Best.Cost<GlobalBest.Cost

```

```

    GlobalBest=particle(i).Best;

end

end

BestCost=zeros(MaxIt,1);
nfe=zeros(MaxIt,1);

for it=1:MaxIt

for i=1:nPop

particle(i).Velocity = w*particle(i).Velocity ...
+c1*rand(VarSize).*(particle(i).Best.Position-particle(i).Position) ...
+c2*rand(VarSize).*(GlobalBest.Position-particle(i).Position);

particle(i).Velocity = max(particle(i).Velocity,VelMin);
particle(i).Velocity = min(particle(i).Velocity,VelMax);
particle(i).Position = particle(i).Position + particle(i).Velocity;
IsOutside=(particle(i).Position<VarMin | particle(i).Position>VarMax);
particle(i).Velocity(IsOutside)=-particle(i).Velocity(IsOutside);
particle(i).Position = max(particle(i).Position,VarMin);
particle(i).Position = min(particle(i).Position,VarMax);
[particle(i).Cost particle(i).Sol] = CostFunction(particle(i).Position);

NewSol.Position=Mutate(particle(i).Position);
[NewSol.Cost NewSol.Sol]=CostFunction(NewSol.Position);
if NewSol.Cost<=particle(i).Cost
particle(i).Position=NewSol.Position;
particle(i).Cost=NewSol.Cost;
particle(i).Sol=NewSol.Sol;

end

if particle(i).Cost<particle(i).Best.Cost

particle(i).Best.Position=particle(i).Position;
particle(i).Best.Cost=particle(i).Cost;
particle(i).Best.Sol=particle(i).Sol;

if particle(i).Best.Cost<GlobalBest.Cost

GlobalBest=particle(i).Best;

end
end
end

NewSol.Position=Mutate(GlobalBest.Position);

```

```

[NewSol.Cost NewSol.Sol]=CostFunction(NewSol.Position);
if NewSol.Cost<=GlobalBest.Cost
GlobalBest=NewSol;

end

BestCost(it)=GlobalBest.Cost;

nfe(it)=NFE;

disp(['Iteration ' num2str(it) ': NFE = ' num2str(nfe(it)) ', Best Cost = '
num2str(BestCost(it))]);

w=w*wdamp;

figure(1);
PlotSolution(GlobalBest.Sol.Tour,model);

end

figure;
plot(nfe,BestCost,'LineWidth',2);
xlabel('Değerlendirme Sayısı');
ylabel('En İyi Mesafe');

```

EK-2. Model Fonksiyonu: Multimodal Mesafe Örneği

```

function model=CreateModel()

x=[36.600574 36.801896 36.839296 38.446685 40.351112 41.005603 40.754687
41.302517 36.977463 38.755488 37.866168 37.829011 41.244760 38.664299 39.793095
39.593498 40.748233 41.018663];

y=[36.192790 34.639696 30.611871 27.156910 27.962908 29.011459 29.839334
36.333447 35.063849 35.322416 32.476390 29.417109 36.432913 29.407196 30.611766
27.821770 30.015052 28.766118];

N=numel(x);

Iskenderun.Mersin.Kara=207
Iskenderun.Mersin.Demir=204
Iskenderun.Mersin.Deniz=122

Iskenderun.Antalya.Kara=745
Iskenderun.Antalya.Deniz=454

Iskenderun.Izmir.Kara=1018
Iskenderun.Izmir.Demir=1195

```

Iskenderun.Izmir.Deniz=1022

Iskenderun.Bandirma.Kara=1048
Iskenderun.Bandirma.Demir=1228
Iskenderun.Bandirma.Deniz=1273

Iskenderun.Haydarpasa.Kara=1047
Iskenderun.Haydarpasa.Demir=1301
Iskenderun.Haydarpasa.Deniz=1344

Iskenderun.Derince.Kara=961
Iskenderun.Derince.Demir=1157
Iskenderun.Derince.Deniz=1384

Iskenderun.Samsun.Kara=847
Iskenderun.Samsun.Demir=1019
Iskenderun.Samsun.Deniz=2068

Iskenderun.Yenice.Kara=157
Iskenderun.Yenice.Demir=161

Iskenderun.Bogazkopru.Kara=423
Iskenderun.Bogazkopru.Demir=445

Iskenderun.Kayacik.Kara=473
Iskenderun.Kayacik.Demir=507

Iskenderun.Kaklik.Kara=824
Iskenderun.Kaklik.Demir=995

Iskenderun.Gelemen.Kara=867
Iskenderun.Gelemen.Demir=1019

Iskenderun.Usak.Kara=804
Iskenderun.Usak.Demir=908

Iskenderun.Hasanbey.Kara=791
Iskenderun.Hasanbey.Demir=945

Iskenderun.Gokkoy.Kara=1041
Iskenderun.Gokkoy.Demir=1139

Iskenderun.Kosekoy.Kara=945
Iskenderun.Kosekoy.Demir=1141
Iskenderun.Halkali.Kara=1076
Iskenderun.Halkali.Demir=1327

a2=[Iskenderun.Mersin.Kara Iskenderun.Mersin.Demir Iskenderun.Mersin.Deniz]

a3=[Iskenderun.Antalya.Kara Iskenderun.Antalya.Deniz]

a4=[Iskenderun.Izmir.Kara Iskenderun.Izmir.Demir Iskenderun.Izmir.Deniz]

a5=[Iskenderun.Bandirma.Kara Iskenderun.Bandirma.Demir Iskenderun.Bandirma.Deniz]
 a6=[Iskenderun.Haydarpasa.Kara Iskenderun.Haydarpasa.Demir
 Iskenderun.Haydarpasa.Deniz]
 a7=[Iskenderun.Derince.Kara Iskenderun.Derince.Demir Iskenderun.Derince.Deniz]
 a8=[Iskenderun.Samsun.Kara Iskenderun.Samsun.Demir Iskenderun.Samsun.Deniz]
 a9=[Iskenderun.Yenice.Kara Iskenderun.Yenice.Demir]
 a10=[Iskenderun.Bogazkopru.Kara Iskenderun.Bogazkopru.Demir]
 a11=[Iskenderun.Kayacik.Kara Iskenderun.Kayacik.Demir]
 a12=[Iskenderun.Kaklik.Kara Iskenderun.Kaklik.Demir]
 a13=[Iskenderun.Gelemen.Kara Iskenderun.Gelemen.Demir]
 a14=[Iskenderun.Usak.Kara Iskenderun.Usak.Demir]
 a15=[Iskenderun.Hasanbey.Kara Iskenderun.Hasanbey.Demir]
 a16=[Iskenderun.Gokkoy.Kara Iskenderun.Gokkoy.Demir]
 a17=[Iskenderun.Kosekoy.Kara Iskenderun.Kosekoy.Demir]
 a18=[Iskenderun.Halkali.Kara Iskenderun.Halkali.Demir]

Mersin.Antalya.Kara=502
 Mersin.Antalya.Deniz=357

Mersin.Izmir.Kara=907
 Mersin.Izmir.Demir=1076
 Mersin.Izmir.Deniz=924

Mersin.Bandirma.Kara=937
 Mersin.Bandirma.Demir=1109
 Mersin.Bandirma.Deniz=1183

Mersin.Haydarpasa.Kara=936
 Mersin.Haydarpasa.Demir=1177
 Mersin.Haydarpasa.Deniz=1246

Mersin.Derince.Kara=849
 Mersin.Derince.Demir=1037
 Mersin.Derince.Deniz=1286

Mersin.Samsun.Kara=736
 Mersin.Samsun.Demir=913
 Mersin.Samsun.Deniz=1970

Mersin.Yenice.Kara=43
 Mersin.Yenice.Demir=44

Mersin.Bogazkopru.Kara=269
 Mersin.Bogazkopru.Demir=326

Mersin.Kayacik.Kara=358
 Mersin.Kayacik.Demir=388

Mersin.Kaklik.Kara=715

Mersin.Kaklik.Demir=875

Mersin.Gelemen.Kara=748
Mersin.Gelemen.Demir=913

Mersin.Usak.Kara=694
Mersin.Usak.Demir=789

Mersin.Hasanbey.Kara=668
Mersin.Hasanbey.Demir=826

Mersin.Gokkoy.Kara=923
Mersin.Gokkoy.Demir=1019

Mersin.Kosekoy.Kara=835
Mersin.Kosekoy.Demir=1021

Mersin.Halkali.Kara=965
Mersin.Halkali.Demir=1207

b3=[Mersin.Antalya.Kara Mersin.Antalya.Deniz]
b4=[Mersin.Izmir.Kara Mersin.Izmir.Demir Mersin.Izmir.Deniz]
b5=[Mersin.Bandirma.Kara Mersin.Bandirma.Demir Mersin.Bandirma.Deniz]
b6=[Mersin.Haydarpasa.Kara Mersin.Haydarpasa.Demir Mersin.Haydarpasa.Deniz]
b7=[Mersin.Derince.Kara Mersin.Derince.Demir Mersin.Derince.Deniz]
b8=[Mersin.Samsun.Kara Mersin.Samsun.Demir Mersin.Samsun.Deniz]
b9=[Mersin.Yenice.Kara Mersin.Yenice.Demir]
b10=[Mersin.Bogazkopru.Kara Mersin.Bogazkopru.Demir]
b11=[Mersin.Kayacik.Kara Mersin.Kayacik.Demir]
b12=[Mersin.Kaklik.Kara Mersin.Kaklik.Demir]
b13=[Mersin.Gelemen.Kara Mersin.Gelemen.Demir]
b14=[Mersin.Usak.Kara Mersin.Usak.Demir]
b15=[Mersin.Hasanbey.Kara Mersin.Hasanbey.Demir]
b16=[Mersin.Gokkoy.Kara Mersin.Gokkoy.Demir]
b17=[Mersin.Kosekoy.Kara Mersin.Kosekoy.Demir]
b18=[Mersin.Halkali.Kara Mersin.Halkali.Demir]

Antalya.Izmir.Kara=461
Antalya.Izmir.Deniz=647

Antalya.Bandirma.Kara=608
Antalya.Bandirma.Deniz=906
Antalya.Haydarpasa.Kara=693
Antalya.Haydarpasa.Deniz=969

Antalya.Derince.Kara=626
Antalya.Derince.Deniz=1025

Antalya.Samsun.Kara=956
Antalya.Samsun.Deniz=1693

Antalya.Yenice.Kara=603

Antalya.Bogazkopru.Kara=614

Antalya.Kayacik.Kara=319

Antalya.Kaklik.Kara=208

Antalya.Gelemen.Kara=933

Antalya.Usak.Kara=302

Antalya.Hasanbey.Kara=452

Antalya.Gokkoy.Kara=514

Antalya.Kosekoy.Kara=618

Antalya.Halkali.Kara=721

c4=[Antalya.Izmir.Kara Antalya.Izmir.Deniz]

c5=[Antalya.Bandirma.Kara Antalya.Bandirma.Deniz]

c6=[Antalya.Haydarpasa.Kara Antalya.Haydarpasa.Deniz]

c7=[Antalya.Derince.Kara Antalya.Derince.Deniz]

c8=[Antalya.Samsun.Kara Antalya.Samsun.Deniz]

c9=[Antalya.Yenice.Kara]

c10=[Antalya.Bogazkopru.Kara]

c11=[Antalya.Kayacik.Kara]

c12=[Antalya.Kaklik.Kara]

c13=[Antalya.Gelemen.Kara]

c14=[Antalya.Usak.Kara]

c15=[Antalya.Hasanbey.Kara]

c16=[Antalya.Gokkoy.Kara]

c17=[Antalya.Kosekoy.Kara]

c18=[Antalya.Halkali.Kara]

Izmir.Bandirma.Kara=276

Izmir.Bandirma.Demir=342

Izmir.Bandirma.Deniz=391

Izmir.Haydarpasa.Kara=462

Izmir.Haydarpasa.Demir=944

Izmir.Haydarpasa.Deniz=454

Izmir.Derince.Kara=439

Izmir.Derince.Demir=794

Izmir.Derince.Deniz=510

Izmir.Samsun.Kara=1002

Izmir.Samsun.Demir=1687
Izmir.Samsun.Deniz=1046

Izmir.Yenice.Kara=874
Izmir.Yenice.Demir=1035

Izmir.Bogazkopru.Kara=826
Izmir.Bogazkopru.Demir=1100

Izmir.Kayacik.Kara=556
Izmir.Kayacik.Demir=689

Izmir.Kaklik.Kara=268
Izmir.Kaklik.Demir=289

Izmir.Gelemen.Kara=1013
Izmir.Gelemen.Demir=1687

Izmir.Usak.Kara=216
Izmir.Usak.Demir=288

Izmir.Hasanbey.Kara=420
Izmir.Hasanbey.Demir=582

Izmir.Gokkoy.Kara=182
Izmir.Gokkoy.Demir=232

Izmir.Kosekoy.Kara=454
Izmir.Kosekoy.Demir=777

Izmir.Halkali.Kara=490
Izmir.Halkali.Demir=974

d5=[Izmir.Bandirma.Kara Izmir.Bandirma.Demir Izmir.Bandirma.Deniz]
d6=[Izmir.Haydarpasa.Kara Izmir.Haydarpasa.Demir Izmir.Haydarpasa.Deniz]
d7=[Izmir.Derince.Kara Izmir.Derince.Demir Izmir.Derince.Deniz]
d8=[Izmir.Samsun.Kara Izmir.Samsun.Demir Izmir.Samsun.Deniz]
d9=[Izmir.Yenice.Kara Izmir.Yenice.Demir]
d10=[Izmir.Bogazkopru.Kara Izmir.Bogazkopru.Demir]
d11=[Izmir.Kayacik.Kara Izmir.Kayacik.Demir]
d12=[Izmir.Kaklik.Kara Izmir.Kaklik.Demir]
d13=[Izmir.Gelemen.Kara Izmir.Gelemen.Demir]
d14=[Izmir.Usak.Kara Izmir.Usak.Demir]
d15=[Izmir.Hasanbey.Kara Izmir.Hasanbey.Demir]
d16=[Izmir.Gokkoy.Kara Izmir.Gokkoy.Demir]
d17=[Izmir.Kosekoy.Kara Izmir.Kosekoy.Demir]
d18=[Izmir.Halkali.Kara Izmir.Halkali.Demir]

Bandirma.Haydarpasa.Kara=243
Bandirma.Haydarpasa.Demir=811

Bandirma.Haydarpasa.Deniz=103

Bandirma.Derince.Kara=222
 Bandirma.Derince.Demir=651
 Bandirma.Derince.Deniz=153

Bandirma.Samsun.Kara=866
 Bandirma.Samsun.Demir=1720
 Bandirma.Samsun.Deniz=747

Bandirma.Yenice.Kara=911
 Bandirma.Yenice.Demir=1067

Bandirma.Bogazkopru.Kara=795
 Bandirma.Bogazkopru.Demir=1132

Bandirma.Kayacik.Kara=603
 Bandirma.Kayacik.Demir=721

Bandirma.Kaklik.Kara=411
 Bandirma.Kaklik.Demir=628

Bandirma.Gelemen.Kara=877
 Bandirma.Gelemen.Demir=1720

Bandirma.Usak.Kara=323
 Bandirma.Usak.Demir=491

Bandirma.Hasanbey.Kara=277
 Bandirma.Hasanbey.Demir=439

Bandirma.Gokkoy.Kara=119
 Bandirma.Gokkoy.Demir=111

Bandirma.Kosekoy.Kara=235
 Bandirma.Kosekoy.Demir=635

Bandirma.Halkali.Kara=271
 Bandirma.Halkali.Demir=831

e6=[Bandirma.Haydarpasa.Kara Bandirma.Haydarpasa.Demir
 Bandirma.Haydarpasa.Deniz]

e7=[Bandirma.Derince.Kara Bandirma.Derince.Demir Bandirma.Derince.Deniz]

e8=[Bandirma.Samsun.Kara Bandirma.Samsun.Demir Bandirma.Samsun.Deniz]

e9=[Bandirma.Yenice.Kara Bandirma.Yenice.Demir]

e10=[Bandirma.Bogazkopru.Kara Bandirma.Bogazkopru.Demir]

e11=[Bandirma.Kayacik.Kara Bandirma.Kayacik.Demir]

e12=[Bandirma.Kaklik.Kara Bandirma.Kaklik.Demir]

e13=[Bandirma.Gelemen.Kara Bandirma.Gelemen.Demir]

e14=[Bandirma.Usak.Kara Bandirma.Usak.Demir]

e15=[Bandirma.Hasanbey.Kara Bandirma.Hasanbey.Demir]
 e16=[Bandirma.Gokkoy.Kara Bandirma.Gokkoy.Demir]
 e17=[Bandirma.Kosekoy.Kara Bandirma.Kosekoy.Demir]
 e18=[Bandirma.Halkali.Kara Bandirma.Halkali.Demir]

Haydarpasa.Derince.Kara=86
 Haydarpasa.Derince.Demir=150
 Haydarpasa.Derince.Deniz=84

Haydarpasa.Samsun.Kara=731
 Haydarpasa.Samsun.Demir=1798
 Haydarpasa.Samsun.Deniz=724

Haydarpasa.Yenice.Kara=902
 Haydarpasa.Yenice.Demir=1146

Haydarpasa.Bogazkopru.Kara=741
 Haydarpasa.Bogazkopru.Demir=1211

Haydarpasa.Kayacik.Kara=703
 Haydarpasa.Kayacik.Demir=800

Haydarpasa.Kaklik.Kara=552
 Haydarpasa.Kaklik.Demir=756

Haydarpasa.Gelemen.Kara=742
 Haydarpasa.Gelemen.Demir=1798

Haydarpasa.Usak.Kara=461
 Haydarpasa.Usak.Demir=669

Haydarpasa.Hasanbey.Kara=299
 Haydarpasa.Hasanbey.Demir=384

Haydarpasa.Gokkoy.Kara=304
 Haydarpasa.Gokkoy.Demir=712

Haydarpasa.Kosekoy.Kara=100
 Haydarpasa.Kosekoy.Demir=167

Haydarpasa.Halkali.Kara=38
 Haydarpasa.Halkali.Demir=30

f7=[Haydarpasa.Derince.Kara Haydarpasa.Derince.Demir Haydarpasa.Derince.Deniz]
 f8=[Haydarpasa.Samsun.Kara Haydarpasa.Samsun.Demir Haydarpasa.Samsun.Deniz]
 f9=[Haydarpasa.Yenice.Kara Haydarpasa.Yenice.Demir]
 f10=[Haydarpasa.Bogazkopru.Kara Haydarpasa.Bogazkopru.Demir]
 f11=[Haydarpasa.Kayacik.Kara Haydarpasa.Kayacik.Demir]
 f12=[Haydarpasa.Kaklik.Kara Haydarpasa.Kaklik.Demir]

f13=[Haydarpasa.Gelemen.Kara Haydarpasa.Gelemen.Demir]
 f14=[Haydarpasa.Usak.Kara Haydarpasa.Usak.Demir]
 f15=[Haydarpasa.Hasanbey.Kara Haydarpasa.Hasanbey.Demir]
 f16=[Haydarpasa.Gokkoy.Kara Haydarpasa.Gokkoy.Demir]
 f17=[Haydarpasa.Kosekoy.Kara Haydarpasa.Kosekoy.Demir]
 f18=[Haydarpasa.Halkali.Kara Haydarpasa.Halkali.Demir]

Derince.Samsun.Kara=122
 Derince.Samsun.Demir=1648
 Derince.Samsun.Deniz=779

Derince.Yenice.Kara=816
 Derince.Yenice.Demir=996

Derince.Bogazkopru.Kara=655
 Derince.Bogazkopru.Demir=1061

Derince.Kayacik.Kara=151
 Derince.Kayacik.Demir=650

Derince.Kaklik.Kara=486
 Derince.Kaklik.Demir=606

Derince.Gelemen.Kara=656
 Derince.Gelemen.Demir=1648

Derince.Usak.Kara=395
 Derince.Usak.Demir=519

Derince.Hasanbey.Kara=233
 Derince.Hasanbey.Demir=234

Derince.Gokkoy.Kara=588
 Derince.Gokkoy.Demir=562

Derince.Kosekoy.Kara=15
 Derince.Kosekoy.Demir=17
 Derince.Halkali.Kara=115
 Derince.Halkali.Demir=180

g8=[Derince.Samsun.Kara Derince.Samsun.Demir Derince.Samsun.Deniz]
 g9=[Derince.Yenice.Kara Derince.Yenice.Demir]
 g10=[Derince.Bogazkopru.Kara Derince.Bogazkopru.Demir]
 g11=[Derince.Kayacik.Kara Derince.Kayacik.Demir]
 g12=[Derince.Kaklik.Kara Derince.Kaklik.Demir]
 g13=[Derince.Gelemen.Kara Derince.Gelemen.Demir]
 g14=[Derince.Usak.Kara Derince.Usak.Demir]
 g15=[Derince.Hasanbey.Kara Derince.Hasanbey.Demir]
 g16=[Derince.Gokkoy.Kara Derince.Gokkoy.Demir]
 g17=[Derince.Kosekoy.Kara Derince.Kosekoy.Demir]

g18=[Derince.Halkali.Kara Derince.Halkali.Demir]

Samsun.Yenice.Kara=701
Samsun.Yenice.Demir=872

Samsun.Bogazkopru.Kara=438
Samsun.Bogazkopru.Demir=588

Samsun.Kayacik.Kara=595
Samsun.Kayacik.Demir=999

Samsun.Kaklik.Kara=856
Samsun.Kaklik.Demir=1487

Samsun.Gelemen.Kara=14
Samsun.Gelemen.Demir=30

Samsun.Usak.Kara=790
Samsun.Usak.Demir=1400

Samsun.Hasanbey.Kara=657
Samsun.Hasanbey.Demir=1437

Samsun.Gokkoy.Kara=927
Samsun.Gokkoy.Demir=1630

Samsun.Kosekoy.Kara=629
Samsun.Kosekoy.Demir=1632

Samsun.Halkali.Kara=759
Samsun.Halkali.Demir=1818

h9=[Samsun.Yenice.Kara Samsun.Yenice.Demir]
h10=[Samsun.Bogazkopru.Kara Samsun.Bogazkopru.Demir]
h11=[Samsun.Kayacik.Kara Samsun.Kayacik.Demir]
h12=[Samsun.Kaklik.Kara Samsun.Kaklik.Demir]
h13=[Samsun.Gelemen.Kara Samsun.Gelemen.Demir]
h14=[Samsun.Usak.Kara Samsun.Usak.Demir]
h15=[Samsun.Hasanbey.Kara Samsun.Hasanbey.Demir]
h16=[Samsun.Gokkoy.Kara Samsun.Gokkoy.Demir]
h17=[Samsun.Kosekoy.Kara Samsun.Kosekoy.Demir]
h18=[Samsun.Halkali.Kara Samsun.Halkali.Demir]

Yenice.Bogazkopru.Kara=269
Yenice.Bogazkopru.Demir=285

Yenice.Kayacik.Kara=329
Yenice.Kayacik.Demir=347

Yenice.Kaklik.Kara=680

Yenice.Kaklik.Demir=834

Yenice.Gelemen.Kara=713
Yenice.Gelemen.Demir=872

Yenice.Usak.Kara=659
Yenice.Usak.Demir=748

Yenice.Hasanbey.Kara=634
Yenice.Hasanbey.Demir=784

Yenice.Gokkoy.Kara=888
Yenice.Gokkoy.Demir=978

Yenice.Kosekoy.Kara=800
Yenice.Kosekoy.Demir=980

Yenice.Halkali.Kara=930
Yenice.Halkali.Demir=1166

i10=[Yenice.Bogazkopru.Kara Yenice.Bogazkopru.Demir]
i11=[Yenice.Kayacik.Kara Yenice.Kayacik.Demir]
i12=[Yenice.Kaklik.Kara Yenice.Kaklik.Demir]
i13=[Yenice.Gelemen.Kara Yenice.Gelemen.Demir]
i14=[Yenice.Usak.Kara Yenice.Usak.Demir]
i15=[Yenice.Hasanbey.Kara Yenice.Hasanbey.Demir]
i16=[Yenice.Gokkoy.Kara Yenice.Gokkoy.Demir]
i17=[Yenice.Kosekoy.Kara Yenice.Kosekoy.Demir]
i18=[Yenice.Halkali.Kara Yenice.Halkali.Demir]

Bogazkopru.Kayacik.Kara=291
Bogazkopru.Kayacik.Demir=412

Bogazkopru.Kaklik.Kara=630
Bogazkopru.Kaklik.Demir=899
Bogazkopru.Gelemen.Kara=448
Bogazkopru.Gelemen.Demir=588

Bogazkopru.Usak.Kara=608
Bogazkopru.Usak.Demir=813

Bogazkopru.Hasanbey.Kara=526
Bogazkopru.Hasanbey.Demir=849

Bogazkopru.Gokkoy.Kara=857
Bogazkopru.Gokkoy.Demir=1043

Bogazkopru.Kosekoy.Kara=638
Bogazkopru.Kosekoy.Demir=1045

Bogazkopru.Halkali.Kara=768
Bogazkopru.Halkali.Demir=1231

j11=[Bogazkopru.Kayacik.Kara Bogazkopru.Kayacik.Demir]
j12=[Bogazkopru.Kaklik.Kara Bogazkopru.Kaklik.Demir]
j13=[Bogazkopru.Gelemen.Kara Bogazkopru.Gelemen.Demir]
j14=[Bogazkopru.Usak.Kara Bogazkopru.Usak.Demir]
j15=[Bogazkopru.Hasanbey.Kara Bogazkopru.Hasanbey.Demir]
j16=[Bogazkopru.Gokkoy.Kara Bogazkopru.Gokkoy.Demir]
j17=[Bogazkopru.Kosekoy.Kara Bogazkopru.Kosekoy.Demir]
j18=[Bogazkopru.Halkali.Kara Bogazkopru.Halkali.Demir]

Kayacik.Kaklik.Kara=362
Kayacik.Kaklik.Demir=488

Kayacik.Gelemen.Kara=604
Kayacik.Gelemen.Demir=999

Kayacik.Usak.Kara=341
Kayacik.Usak.Demir=401

Kayacik.Hasanbey.Kara=341
Kayacik.Hasanbey.Demir=438
Kayacik.Gokkoy.Kara=570
Kayacik.Gokkoy.Demir=632

Kayacik.Kosekoy.Kara=602
Kayacik.Kosekoy.Demir=634

Kayacik.Halkali.Kara=660
Kayacik.Halkali.Demir=820

k12=[Kayacik.Kaklik.Kara Kayacik.Kaklik.Demir]
k13=[Kayacik.Gelemen.Kara Kayacik.Gelemen.Demir]
k14=[Kayacik.Usak.Kara Kayacik.Usak.Demir]
k15=[Kayacik.Hasanbey.Kara Kayacik.Hasanbey.Demir]
k16=[Kayacik.Gokkoy.Kara Kayacik.Gokkoy.Demir]
k17=[Kayacik.Kosekoy.Kara Kayacik.Kosekoy.Demir]
k18=[Kayacik.Halkali.Kara Kayacik.Halkali.Demir]

Kaklik.Gelemen.Kara=863
Kaklik.Gelemen.Demir=1487

Kaklik.Usak.Kara=120
Kaklik.Usak.Demir=353

Kaklik.Hasanbey.Kara=310
Kaklik.Hasanbey.Demir=294

Kaklik.Gokkoy.Kara=311

Kaklik.Gokkoy.Demir=518

Kaklik.Kosekoy.Kara=464
Kaklik.Kosekoy.Demir=589

Kaklik.Halkali.Kara=574
Kaklik.Halkali.Demir=775

l13=[Kaklik.Gelemen.Kara Kaklik.Gelemen.Demir]
l14=[Kaklik.Usak.Kara Kaklik.Usak.Demir]
l15=[Kaklik.Hasanbey.Kara Kaklik.Hasanbey.Demir]
l16=[Kaklik.Gokkoy.Kara Kaklik.Gokkoy.Demir]
l17=[Kaklik.Kosekoy.Kara Kaklik.Kosekoy.Demir]
l18=[Kaklik.Halkali.Kara Kaklik.Halkali.Demir]

Gelemen.Usak.Kara=801
Gelemen.Usak.Demir=1400

Gelemen.Hasanbey.Kara=668
Gelemen.Hasanbey.Demir=1437

Gelemen.Gokkoy.Kara=938
Gelemen.Gokkoy.Demir=1630
Gelemen.Kosekoy.Kara=640
Gelemen.Kosekoy.Demir=1632

Gelemen.Halkali.Kara=770
Gelemen.Halkali.Demir=1818

m14=[Gelemen.Usak.Kara Gelemen.Usak.Demir]
m15=[Gelemen.Hasanbey.Kara Gelemen.Hasanbey.Demir]
m16=[Gelemen.Gokkoy.Kara Gelemen.Gokkoy.Demir]
m17=[Gelemen.Kosekoy.Kara Gelemen.Kosekoy.Demir]
m18=[Gelemen.Halkali.Kara Gelemen.Halkali.Demir]

Usak.Hasanbey.Kara=225
Usak.Hasanbey.Demir=307

Usak.Gokkoy.Kara=235
Usak.Gokkoy.Demir=381

Usak.Kosekoy.Kara=379
Usak.Kosekoy.Demir=503

Usak.Halkali.Kara=490
Usak.Halkali.Demir=689

n15=[Usak.Hasanbey.Kara Usak.Hasanbey.Demir]
n16=[Usak.Gokkoy.Kara Usak.Gokkoy.Demir]
n17=[Usak.Kosekoy.Kara Usak.Kosekoy.Demir]

n18=[Usak.Halkali.Kara Usak.Halkali.Demir]

Hasanbey.Gokkoy.Kara=340
Hasanbey.Gokkoy.Demir=350

Hasanbey.Kosekoy.Kara=217
Hasanbey.Kosekoy.Demir=218

Hasanbey.Halkali.Kara=328
Hasanbey.Halkali.Demir=404

o16=[Hasanbey.Gokkoy.Kara Hasanbey.Gokkoy.Demir]
o17=[Hasanbey.Kosekoy.Kara Hasanbey.Kosekoy.Demir]
o18=[Hasanbey.Halkali.Kara Hasanbey.Halkali.Demir]

Gokkoy.Kosekoy.Kara=303
Gokkoy.Kosekoy.Demir=546

Gokkoy.Halkali.Kara=340
Gokkoy.Halkali.Demir=732

p17=[Gokkoy.Kosekoy.Kara Gokkoy.Kosekoy.Demir]
p18=[Gokkoy.Halkali.Kara Gokkoy.Halkali.Demir]
Kosekoy.Halkali.Kara=131
Kosekoy.Halkali.Demir=186

q18=[Kosekoy.Halkali.Kara Kosekoy.Halkali.Demir]

D=[0 min(a2) min(a3) min(a4) min(a5) min(a6) min(a7) min(a8) min(a9) min(a10)
min(a11) min(a12) min(a13) min(a14) min(a15) min(a16) min(a17) min(a18);...
min(a2) 0 min(b3) min(b4) min(b5) min(b6) min(b7) min(b8) min(b9) min(b10) min(b11)
min(b12) min(b13) min(b14) min(b15) min(b16) min(b17) min(b18);...
min(a3) min(b3) 0 min(c4) min(c5) min(c6) min(c7) min(c8) min(c9) min(c10) min(c11)
min(c12) min(c13) min(c14) min(c15) min(c16) min(c17) min(c18);...
min(b4) min(b4) min(c4) 0 min(d5) min(d6) min(d7) min(d8) min(d9) min(d10) min(d11)
min(d12) min(d13) min(d14) min(d15) min(d16) min(d17) min(d18);...
min(b5) min(b5) min(c5) min(d5) 0 min(e6) min(e7) min(e8) min(e9) min(e10) min(e11)
min(e12) min(e13) min(e14) min(e15) min(e16) min(e17) min(e18);...
min(a6) min(b6) min(c6) min(d6) min(e6) 0 min(f7) min(f8) min(f9) min(f10) min(f11)
min(f12) min(f13) min(f14) min(f15) min(f16) min(f17) min(f18);...
min(a7) min(b7) min(c7) min(d7) min(e7) min(f7) 0 min(g8) min(g9) min(g10) min(g11)
min(g12) min(g13) min(g14) min(g15) min(g16) min(g17) min(g18);...
min(a8) min(b8) min(c8) min(d8) min(e8) min(f8) min(g8) 0 min(h9) min(h10) min(h11)
min(h12) min(h13) min(h14) min(h15) min(h16) min(h17) min(h18);...
min(a9) min(b9) min(c9) min(d9) min(e9) min(f9) min(g9) min(h9) 0 min(i10) min(i11)
min(i12) min(i13) min(i14) min(i15) min(i16) min(i17) min(i18);...
min(a10) min(b10) min(c10) min(d10) min(e10) min(f10) min(g10) min(h10) min(i10) 0
min(j11) min(j12) min(j13) min(j14) min(j15) min(j16) min(j17) min(j18);...
min(a11) min(b11) min(c11) min(d11) min(e11) min(f11) min(g11) min(h11) min(i11)
min(j11) 0 min(k12) min(k13) min(k14) min(k15) min(k16) min(k17) min(k18);...

```

min(a12) min(b12) min(c12) min(d12) min(e12) min(f12) min(g12) min(h12) min(i12)
min(j12) min(k12) 0 min(l13) min(l14) min(l15) min(l16) min(l17) min(l18);...
min(a13) min(b13) min(c13) min(d13) min(e13) min(f13) min(g13) min(h13) min(i13)
min(j13) min(k13) min(l13) 0 min(m14) min(m15) min(m16) min(m17) min(m18);...
min(a14) min(b14) min(c14) min(d14) min(e14) min(f14) min(g14) min(h14) min(i14)
min(j14) min(k14) min(l14) min(m14) 0 min(n15) min(n16) min(n17) min(n18);...
min(a15) min(b15) min(c15) min(d15) min(e15) min(f15) min(g15) min(h15) min(i15)
min(j15) min(k15) min(l15) min(m15) min(n15) 0 min(o16) min(o17) min(o18);...
min(a16) min(b16) min(c16) min(d16) min(e16) min(f16) min(g16) min(h16) min(i16)
min(j16) min(k16) min(l16) min(m16) min(n16) min(o16) 0 min(p17) min(p18);...
min(a17) min(b17) min(c17) min(d17) min(e17) min(f17) min(g17) min(h17) min(i17)
min(j17) min(k17) min(l17) min(m17) min(n17) min(o17) min(p17) 0 min(q18);...
min(a18) min(b18) min(c18) min(d18) min(e18) min(f18) min(g18) min(h18) min(i18)
min(j18) min(k18) min(l18) min(m18) min(n18) min(o18) min(p18) min(q18) 0];

```

```

model.N=N;
model.x=x;
model.y=y;
model.D=D;
end

```

EK-3. Hesap Fonksiyonu: Multimodal Mesafe Örneği

```

function [z sol]=MyCost(x,model)

```

```

global NFE;
if isempty(NFE)
NFE=0;

```

```

end

```

```

NFE=NFE+1;

```

```

N=model.N;
D=model.D;

```

```

[~, Tour]=sort(x);

```

```

L=0;

```

```

for k=1:N

```

```

i=Tour(k);

```

```

if k<N
j=Tour(k+1);
else
j=Tour(1);

```

```

end

L=L+D(i,j);

end

z=L;

sol.Tour=Tour;
sol.L=L;

end

```

EK-4. Mutasyon: Multimodal Mesafe Örneği

```

function xnew=Mutate(x)

[~, Tour]=sort(x);

M=randi([1 3]);

NewTour=DoReversion(Tour);

xnew=zeros(size(x));

xnew(NewTour)=x(Tour);
end

function NewTour=DoReversion(Tour)

n=numel(Tour);

i=randsample(n,2);
i1=min(i);
i2=max(i);

NewTour=Tour;
NewTour(i1:i2)=Tour(i2:-1:i1);

End

```

EK-5. Yazdırma Fonksiyonu: Multimodal Mesafe Örneği

```

function PlotSolution(tour,model)

```

```

x=model.x;
y=model.y;

tour=[tour tour(1)];

plot(x(tour),y(tour),'r-s',...
'LineWidth',1,...
'MarkerSize',6,...
'MarkerFaceColor','y');

end

```

EK-6. Lineer Optimiasyon Ana Kodlaması: Multimodal Mesafe örneği

```

clear all;
close all;
clc;

Iskenderun.Mersin.Kara=207;
Iskenderun.Mersin.Demir=204;
Iskenderun.Mersin.Deniz=122;

Iskenderun.Antalya.Kara=745;
Iskenderun.Antalya.Deniz=454;

Iskenderun.Izmir.Kara=1018;
Iskenderun.Izmir.Demir=1195;
Iskenderun.Izmir.Deniz=1022;

Iskenderun.Bandirma.Kara=1048;
Iskenderun.Bandirma.Demir=1228;
Iskenderun.Bandirma.Deniz=1273;

Iskenderun.Haydarpasa.Kara=1047;
Iskenderun.Haydarpasa.Demir=1301;
Iskenderun.Haydarpasa.Deniz=1344;

Iskenderun.Derince.Kara=961;
Iskenderun.Derince.Demir=1157;
Iskenderun.Derince.Deniz=1384;

Iskenderun.Samsun.Kara=847;
Iskenderun.Samsun.Demir=1019;
Iskenderun.Samsun.Deniz=2068;

Iskenderun.Yenice.Kara=157;
Iskenderun.Yenice.Demir=161;

Iskenderun.Bogazkopru.Kara=423;

```

Iskenderun.Bogazkopru.Demir=445;

Iskenderun.Kayacik.Kara=473;
Iskenderun.Kayacik.Demir=507;

Iskenderun.Kaklik.Kara=824;
Iskenderun.Kaklik.Demir=995;

Iskenderun.Gelemen.Kara=867;
Iskenderun.Gelemen.Demir=1019;

Iskenderun.Usak.Kara=804;
Iskenderun.Usak.Demir=908;

Iskenderun.Hasanbey.Kara=791;
Iskenderun.Hasanbey.Demir=945;

Iskenderun.Gokkoy.Kara=1041;
Iskenderun.Gokkoy.Demir=1139;

Iskenderun.Kosekoy.Kara=945;
Iskenderun.Kosekoy.Demir=1141;

Iskenderun.Halkali.Kara=1076;
Iskenderun.Halkali.Demir=1327;

a2=[Iskenderun.Mersin.Kara Iskenderun.Mersin.Demir Iskenderun.Mersin.Deniz];
a3=[Iskenderun.Antalya.Kara Iskenderun.Antalya.Deniz];
a4=[Iskenderun.Izmir.Kara Iskenderun.Izmir.Demir Iskenderun.Izmir.Deniz];
a5=[Iskenderun.Bandirma.Kara Iskenderun.Bandirma.Demir Iskenderun.Bandirma.Deniz];
a6=[Iskenderun.Haydarpasa.Kara Iskenderun.Haydarpasa.Demir
Iskenderun.Haydarpasa.Deniz];
a7=[Iskenderun.Derince.Kara Iskenderun.Derince.Demir Iskenderun.Derince.Deniz];
a8=[Iskenderun.Samsun.Kara Iskenderun.Samsun.Demir Iskenderun.Samsun.Deniz];
a9=[Iskenderun.Yenice.Kara Iskenderun.Yenice.Demir];
a10=[Iskenderun.Bogazkopru.Kara Iskenderun.Bogazkopru.Demir];
a11=[Iskenderun.Kayacik.Kara Iskenderun.Kayacik.Demir];
a12=[Iskenderun.Kaklik.Kara Iskenderun.Kaklik.Demir];
a13=[Iskenderun.Gelemen.Kara Iskenderun.Gelemen.Demir];
a14=[Iskenderun.Usak.Kara Iskenderun.Usak.Demir];
a15=[Iskenderun.Hasanbey.Kara Iskenderun.Hasanbey.Demir];
a16=[Iskenderun.Gokkoy.Kara Iskenderun.Gokkoy.Demir];
a17=[Iskenderun.Kosekoy.Kara Iskenderun.Kosekoy.Demir];
a18=[Iskenderun.Halkali.Kara Iskenderun.Halkali.Demir];

Mersin.Antalya.Kara=502;
Mersin.Antalya.Deniz=357;

Mersin.Izmir.Kara=907;
Mersin.Izmir.Demir=1076;

Mersin.Izmir.Deniz=924;

Mersin.Bandirma.Kara=937;
Mersin.Bandirma.Demir=1109;
Mersin.Bandirma.Deniz=1183;

Mersin.Haydarpasa.Kara=936;
Mersin.Haydarpasa.Demir=1177;
Mersin.Haydarpasa.Deniz=1246;

Mersin.Derince.Kara=849;
Mersin.Derince.Demir=1037;
Mersin.Derince.Deniz=1286;

Mersin.Samsun.Kara=736;
Mersin.Samsun.Demir=913;
Mersin.Samsun.Deniz=1970;

Mersin.Yenice.Kara=43;
Mersin.Yenice.Demir=44;

Mersin.Bogazkopru.Kara=269;
Mersin.Bogazkopru.Demir=326;

Mersin.Kayacik.Kara=358;
Mersin.Kayacik.Demir=388;

Mersin.Kaklik.Kara=715;
Mersin.Kaklik.Demir=875;

Mersin.Gelemen.Kara=748;
Mersin.Gelemen.Demir=913;

Mersin.Usak.Kara=694;
Mersin.Usak.Demir=789;

Mersin.Hasanbey.Kara=668;
Mersin.Hasanbey.Demir=826;

Mersin.Gokkoy.Kara=923;
Mersin.Gokkoy.Demir=1019;

Mersin.Kosekoy.Kara=835;
Mersin.Kosekoy.Demir=1021;

Mersin.Halkali.Kara=965;
Mersin.Halkali.Demir=1207;

b3=[Mersin.Antalya.Kara Mersin.Antalya.Deniz];

b4=[Mersin.Izmir.Kara Mersin.Izmir.Demir Mersin.Izmir.Deniz];

b5=[Mersin.Bandirma.Kara Mersin.Bandirma.Demir Mersin.Bandirma.Deniz];
 b6=[Mersin.Haydarpasa.Kara Mersin.Haydarpasa.Demir Mersin.Haydarpasa.Deniz];
 b7=[Mersin.Derince.Kara Mersin.Derince.Demir Mersin.Derince.Deniz];
 b8=[Mersin.Samsun.Kara Mersin.Samsun.Demir Mersin.Samsun.Deniz];
 b9=[Mersin.Yenice.Kara Mersin.Yenice.Demir];
 b10=[Mersin.Bogazkopru.Kara Mersin.Bogazkopru.Demir];
 b11=[Mersin.Kayacik.Kara Mersin.Kayacik.Demir];
 b12=[Mersin.Kaklik.Kara Mersin.Kaklik.Demir];
 b13=[Mersin.Gelemen.Kara Mersin.Gelemen.Demir];
 b14=[Mersin.Usak.Kara Mersin.Usak.Demir];
 b15=[Mersin.Hasanbey.Kara Mersin.Hasanbey.Demir];
 b16=[Mersin.Gokkoy.Kara Mersin.Gokkoy.Demir];
 b17=[Mersin.Kosekoy.Kara Mersin.Kosekoy.Demir];
 b18=[Mersin.Halkali.Kara Mersin.Halkali.Demir];

Antalya.Izmir.Kara=461;
 Antalya.Izmir.Deniz=647;

Antalya.Bandirma.Kara=608;
 Antalya.Bandirma.Deniz=906;

Antalya.Haydarpasa.Kara=693;
 Antalya.Haydarpasa.Deniz=969;

Antalya.Derince.Kara=626;
 Antalya.Derince.Deniz=1025;

Antalya.Samsun.Kara=956;
 Antalya.Samsun.Deniz=1693;

Antalya.Yenice.Kara=603;

Antalya.Bogazkopru.Kara=614;

Antalya.Kayacik.Kara=319;

Antalya.Kaklik.Kara=208;

Antalya.Gelemen.Kara=933;

Antalya.Usak.Kara=302;

Antalya.Hasanbey.Kara=452;

Antalya.Gokkoy.Kara=514;

Antalya.Kosekoy.Kara=618;

Antalya.Halkali.Kara=721;

c4=[Antalya.Izmir.Kara Antalya.Izmir.Deniz];
 c5=[Antalya.Bandirma.Kara Antalya.Bandirma.Deniz];
 c6=[Antalya.Haydarpasa.Kara Antalya.Haydarpasa.Deniz];
 c7=[Antalya.Derince.Kara Antalya.Derince.Deniz];
 c8=[Antalya.Samsun.Kara Antalya.Samsun.Deniz];
 c9=[Antalya.Yenice.Kara];
 c10=[Antalya.Bogazkopru.Kara];
 c11=[Antalya.Kayacik.Kara];
 c12=[Antalya.Kaklik.Kara];
 c13=[Antalya.Gelemen.Kara];
 c14=[Antalya.Usak.Kara];
 c15=[Antalya.Hasanbey.Kara];
 c16=[Antalya.Gokkoy.Kara];
 c17=[Antalya.Kosekoy.Kara];
 c18=[Antalya.Halkali.Kara];

Izmir.Bandirma.Kara=276;
 Izmir.Bandirma.Demir=342;
 Izmir.Bandirma.Deniz=391;

Izmir.Haydarpasa.Kara=462;
 Izmir.Haydarpasa.Demir=944;
 Izmir.Haydarpasa.Deniz=454;

Izmir.Derince.Kara=439;
 Izmir.Derince.Demir=794;
 Izmir.Derince.Deniz=510;

Izmir.Samsun.Kara=1002;
 Izmir.Samsun.Demir=1687;
 Izmir.Samsun.Deniz=1046;

Izmir.Yenice.Kara=874;
 Izmir.Yenice.Demir=1035;

Izmir.Bogazkopru.Kara=826;
 Izmir.Bogazkopru.Demir=1100;

Izmir.Kayacik.Kara=556;
 Izmir.Kayacik.Demir=689;

Izmir.Kaklik.Kara=268;
 Izmir.Kaklik.Demir=289;

Izmir.Gelemen.Kara=1013;
 Izmir.Gelemen.Demir=1687;

Izmir.Usak.Kara=216;
 Izmir.Usak.Demir=288;

Izmir.Hasanbey.Kara=420;
Izmir.Hasanbey.Demir=582;

Izmir.Gokkoy.Kara=182;
Izmir.Gokkoy.Demir=232;

Izmir.Kosekoy.Kara=454;
Izmir.Kosekoy.Demir=777;

Izmir.Halkali.Kara=490;
Izmir.Halkali.Demir=974;

d5=[Izmir.Bandirma.Kara Izmir.Bandirma.Demir Izmir.Bandirma.Deniz];
d6=[Izmir.Haydarpasa.Kara Izmir.Haydarpasa.Demir Izmir.Haydarpasa.Deniz];
d7=[Izmir.Derince.Kara Izmir.Derince.Demir Izmir.Derince.Deniz];
d8=[Izmir.Samsun.Kara Izmir.Samsun.Demir Izmir.Samsun.Deniz];
d9=[Izmir.Yenice.Kara Izmir.Yenice.Demir];
d10=[Izmir.Bogazkopru.Kara Izmir.Bogazkopru.Demir];
d11=[Izmir.Kayacik.Kara Izmir.Kayacik.Demir];
d12=[Izmir.Kaklik.Kara Izmir.Kaklik.Demir];
d13=[Izmir.Gelemen.Kara Izmir.Gelemen.Demir];
d14=[Izmir.Usak.Kara Izmir.Usak.Demir];
d15=[Izmir.Hasanbey.Kara Izmir.Hasanbey.Demir];
d16=[Izmir.Gokkoy.Kara Izmir.Gokkoy.Demir];
d17=[Izmir.Kosekoy.Kara Izmir.Kosekoy.Demir];
d18=[Izmir.Halkali.Kara Izmir.Halkali.Demir];

Bandirma.Haydarpasa.Kara=243;
Bandirma.Haydarpasa.Demir=811;
Bandirma.Haydarpasa.Deniz=103;

Bandirma.Derince.Kara=222;
Bandirma.Derince.Demir=651;
Bandirma.Derince.Deniz=153;

Bandirma.Samsun.Kara=866;
Bandirma.Samsun.Demir=1720;
Bandirma.Samsun.Deniz=747;

Bandirma.Yenice.Kara=911;
Bandirma.Yenice.Demir=1067;

Bandirma.Bogazkopru.Kara=795;
Bandirma.Bogazkopru.Demir=1132;

Bandirma.Kayacik.Kara=603;
Bandirma.Kayacik.Demir=721;

Bandirma.Kaklik.Kara=411;

Bandirma.Kaklik.Demir=628;

Bandirma.Gelemen.Kara=877;
Bandirma.Gelemen.Demir=1720;

Bandirma.Usak.Kara=323;
Bandirma.Usak.Demir=491;

Bandirma.Hasanbey.Kara=277;
Bandirma.Hasanbey.Demir=439;

Bandirma.Gokkoy.Kara=119;
Bandirma.Gokkoy.Demir=111;

Bandirma.Kosekoy.Kara=235;
Bandirma.Kosekoy.Demir=635;

Bandirma.Halkali.Kara=271;
Bandirma.Halkali.Demir=831;

e6=[Bandirma.Haydarpasa.Kara Bandirma.Haydarpasa.Demir
Bandirma.Haydarpasa.Deniz];

e7=[Bandirma.Derince.Kara Bandirma.Derince.Demir Bandirma.Derince.Deniz];
e8=[Bandirma.Samsun.Kara Bandirma.Samsun.Demir Bandirma.Samsun.Deniz];
e9=[Bandirma.Yenice.Kara Bandirma.Yenice.Demir];
e10=[Bandirma.Bogazkopru.Kara Bandirma.Bogazkopru.Demir];
e11=[Bandirma.Kayacik.Kara Bandirma.Kayacik.Demir];
e12=[Bandirma.Kaklik.Kara Bandirma.Kaklik.Demir];
e13=[Bandirma.Gelemen.Kara Bandirma.Gelemen.Demir];
e14=[Bandirma.Usak.Kara Bandirma.Usak.Demir];
e15=[Bandirma.Hasanbey.Kara Bandirma.Hasanbey.Demir];
e16=[Bandirma.Gokkoy.Kara Bandirma.Gokkoy.Demir];
e17=[Bandirma.Kosekoy.Kara Bandirma.Kosekoy.Demir];
e18=[Bandirma.Halkali.Kara Bandirma.Halkali.Demir];

Haydarpasa.Derince.Kara=86;
Haydarpasa.Derince.Demir=150;
Haydarpasa.Derince.Deniz=84;

Haydarpasa.Samsun.Kara=731;
Haydarpasa.Samsun.Demir=1798;
Haydarpasa.Samsun.Deniz=724;

Haydarpasa.Yenice.Kara=902;
Haydarpasa.Yenice.Demir=1146;

Haydarpasa.Bogazkopru.Kara=741;
Haydarpasa.Bogazkopru.Demir=1211;

Haydarpasa.Kayacik.Kara=703;
Haydarpasa.Kayacik.Demir=800;

Haydarpasa.Kaklik.Kara=552;
Haydarpasa.Kaklik.Demir=756;

Haydarpasa.Gelemen.Kara=742;
Haydarpasa.Gelemen.Demir=1798;

Haydarpasa.Usak.Kara=461;
Haydarpasa.Usak.Demir=669;

Haydarpasa.Hasanbey.Kara=299;
Haydarpasa.Hasanbey.Demir=384;

Haydarpasa.Gokkoy.Kara=304;
Haydarpasa.Gokkoy.Demir=712;

Haydarpasa.Kosekoy.Kara=100;
Haydarpasa.Kosekoy.Demir=167;
Haydarpasa.Halkali.Kara=38;
Haydarpasa.Halkali.Demir=30;

f7=[Haydarpasa.Derince.Kara Haydarpasa.Derince.Demir Haydarpasa.Derince.Deniz];
f8=[Haydarpasa.Samsun.Kara Haydarpasa.Samsun.Demir Haydarpasa.Samsun.Deniz];
f9=[Haydarpasa.Yenice.Kara Haydarpasa.Yenice.Demir];
f10=[Haydarpasa.Bogazkopru.Kara Haydarpasa.Bogazkopru.Demir];
f11=[Haydarpasa.Kayacik.Kara Haydarpasa.Kayacik.Demir];
f12=[Haydarpasa.Kaklik.Kara Haydarpasa.Kaklik.Demir];
f13=[Haydarpasa.Gelemen.Kara Haydarpasa.Gelemen.Demir];
f14=[Haydarpasa.Usak.Kara Haydarpasa.Usak.Demir];
f15=[Haydarpasa.Hasanbey.Kara Haydarpasa.Hasanbey.Demir];
f16=[Haydarpasa.Gokkoy.Kara Haydarpasa.Gokkoy.Demir];
f17=[Haydarpasa.Kosekoy.Kara Haydarpasa.Kosekoy.Demir];
f18=[Haydarpasa.Halkali.Kara Haydarpasa.Halkali.Demir];

Derince.Samsun.Kara=122;
Derince.Samsun.Demir=1648;
Derince.Samsun.Deniz=779;

Derince.Yenice.Kara=816;
Derince.Yenice.Demir=996;

Derince.Bogazkopru.Kara=655;
Derince.Bogazkopru.Demir=1061;

Derince.Kayacik.Kara=151;
Derince.Kayacik.Demir=650;

Derince.Kaklik.Kara=486;

Derince.Kaklik.Demir=606;

Derince.Gelemen.Kara=656;
Derince.Gelemen.Demir=1648;

Derince.Usak.Kara=395;
Derince.Usak.Demir=519;

Derince.Hasanbey.Kara=233;
Derince.Hasanbey.Demir=234;

Derince.Gokkoy.Kara=588;
Derince.Gokkoy.Demir=562;

Derince.Kosekoy.Kara=15;
Derince.Kosekoy.Demir=17;

Derince.Halkali.Kara=115;
Derince.Halkali.Demir=180;

g8=[Derince.Samsun.Kara Derince.Samsun.Demir Derince.Samsun.Deniz];
g9=[Derince.Yenice.Kara Derince.Yenice.Demir];
g10=[Derince.Bogazkopru.Kara Derince.Bogazkopru.Demir];
g11=[Derince.Kayacik.Kara Derince.Kayacik.Demir];
g12=[Derince.Kaklik.Kara Derince.Kaklik.Demir];
g13=[Derince.Gelemen.Kara Derince.Gelemen.Demir];
g14=[Derince.Usak.Kara Derince.Usak.Demir];
g15=[Derince.Hasanbey.Kara Derince.Hasanbey.Demir];
g16=[Derince.Gokkoy.Kara Derince.Gokkoy.Demir];
g17=[Derince.Kosekoy.Kara Derince.Kosekoy.Demir];
g18=[Derince.Halkali.Kara Derince.Halkali.Demir];

Samsun.Yenice.Kara=701;
Samsun.Yenice.Demir=872;

Samsun.Bogazkopru.Kara=438;
Samsun.Bogazkopru.Demir=588;

Samsun.Kayacik.Kara=595;
Samsun.Kayacik.Demir=999;

Samsun.Kaklik.Kara=856;
Samsun.Kaklik.Demir=1487;

Samsun.Gelemen.Kara=14;
Samsun.Gelemen.Demir=30;

Samsun.Usak.Kara=790;
Samsun.Usak.Demir=1400;

Samsun.Hasanbey.Kara=657;
Samsun.Hasanbey.Demir=1437;

Samsun.Gokkoy.Kara=927;
Samsun.Gokkoy.Demir=1630;

Samsun.Kosekoy.Kara=629;
Samsun.Kosekoy.Demir=1632;
Samsun.Halkali.Kara=759;
Samsun.Halkali.Demir=1818;

h9=[Samsun.Yenice.Kara Samsun.Yenice.Demir];
h10=[Samsun.Bogazkopru.Kara Samsun.Bogazkopru.Demir];
h11=[Samsun.Kayacik.Kara Samsun.Kayacik.Demir];
h12=[Samsun.Kaklik.Kara Samsun.Kaklik.Demir];
h13=[Samsun.Gelemen.Kara Samsun.Gelemen.Demir];
h14=[Samsun.Usak.Kara Samsun.Usak.Demir];
h15=[Samsun.Hasanbey.Kara Samsun.Hasanbey.Demir];
h16=[Samsun.Gokkoy.Kara Samsun.Gokkoy.Demir];
h17=[Samsun.Kosekoy.Kara Samsun.Kosekoy.Demir];
h18=[Samsun.Halkali.Kara Samsun.Halkali.Demir];

Yenice.Bogazkopru.Kara=269;
Yenice.Bogazkopru.Demir=285;

Yenice.Kayacik.Kara=329;
Yenice.Kayacik.Demir=347;

Yenice.Kaklik.Kara=680;
Yenice.Kaklik.Demir=834;

Yenice.Gelemen.Kara=713;
Yenice.Gelemen.Demir=872;

Yenice.Usak.Kara=659;
Yenice.Usak.Demir=748;

Yenice.Hasanbey.Kara=634;
Yenice.Hasanbey.Demir=784;

Yenice.Gokkoy.Kara=888;
Yenice.Gokkoy.Demir=978;

Yenice.Kosekoy.Kara=800;
Yenice.Kosekoy.Demir=980;

Yenice.Halkali.Kara=930;
Yenice.Halkali.Demir=1166;

i10=[Yenice.Bogazkopru.Kara Yenice.Bogazkopru.Demir];

i11=[Yenice.Kayacik.Kara Yenice.Kayacik.Demir];
 i12=[Yenice.Kaklik.Kara Yenice.Kaklik.Demir];
 i13=[Yenice.Gelemen.Kara Yenice.Gelemen.Demir];
 i14=[Yenice.Usak.Kara Yenice.Usak.Demir];
 i15=[Yenice.Hasanbey.Kara Yenice.Hasanbey.Demir];
 i16=[Yenice.Gokkoy.Kara Yenice.Gokkoy.Demir];
 i17=[Yenice.Kosekoy.Kara Yenice.Kosekoy.Demir];
 i18=[Yenice.Halkali.Kara Yenice.Halkali.Demir];

Bogazkopru.Kayacik.Kara=291;
 Bogazkopru.Kayacik.Demir=412;

Bogazkopru.Kaklik.Kara=630;
 Bogazkopru.Kaklik.Demir=899;

Bogazkopru.Gelemen.Kara=448;
 Bogazkopru.Gelemen.Demir=588;
 Bogazkopru.Usak.Kara=608;
 Bogazkopru.Usak.Demir=813;

Bogazkopru.Hasanbey.Kara=526;
 Bogazkopru.Hasanbey.Demir=849;

Bogazkopru.Gokkoy.Kara=857;
 Bogazkopru.Gokkoy.Demir=1043;

Bogazkopru.Kosekoy.Kara=638;
 Bogazkopru.Kosekoy.Demir=1045;

Bogazkopru.Halkali.Kara=768;
 Bogazkopru.Halkali.Demir=1231;

j11=[Bogazkopru.Kayacik.Kara Bogazkopru.Kayacik.Demir];
 j12=[Bogazkopru.Kaklik.Kara Bogazkopru.Kaklik.Demir];
 j13=[Bogazkopru.Gelemen.Kara Bogazkopru.Gelemen.Demir];
 j14=[Bogazkopru.Usak.Kara Bogazkopru.Usak.Demir];
 j15=[Bogazkopru.Hasanbey.Kara Bogazkopru.Hasanbey.Demir];
 j16=[Bogazkopru.Gokkoy.Kara Bogazkopru.Gokkoy.Demir];
 j17=[Bogazkopru.Kosekoy.Kara Bogazkopru.Kosekoy.Demir];
 j18=[Bogazkopru.Halkali.Kara Bogazkopru.Halkali.Demir];

Kayacik.Kaklik.Kara=362;
 Kayacik.Kaklik.Demir=488;

Kayacik.Gelemen.Kara=604;
 Kayacik.Gelemen.Demir=999;

Kayacik.Usak.Kara=341;
 Kayacik.Usak.Demir=401;

Kayacik.Hasanbey.Kara=341;
Kayacik.Hasanbey.Demir=438;

Kayacik.Gokkoy.Kara=570;
Kayacik.Gokkoy.Demir=632;

Kayacik.Kosekoy.Kara=602;
Kayacik.Kosekoy.Demir=634;

Kayacik.Halkali.Kara=660;
Kayacik.Halkali.Demir=820;

k12=[Kayacik.Kaklik.Kara Kayacik.Kaklik.Demir];
k13=[Kayacik.Gelemen.Kara Kayacik.Gelemen.Demir];
k14=[Kayacik.Usak.Kara Kayacik.Usak.Demir];
k15=[Kayacik.Hasanbey.Kara Kayacik.Hasanbey.Demir];
k16=[Kayacik.Gokkoy.Kara Kayacik.Gokkoy.Demir];
k17=[Kayacik.Kosekoy.Kara Kayacik.Kosekoy.Demir];
k18=[Kayacik.Halkali.Kara Kayacik.Halkali.Demir];

Kaklik.Gelemen.Kara=863;
Kaklik.Gelemen.Demir=1487;

Kaklik.Usak.Kara=120;
Kaklik.Usak.Demir=353;

Kaklik.Hasanbey.Kara=310;
Kaklik.Hasanbey.Demir=294;

Kaklik.Gokkoy.Kara=311;
Kaklik.Gokkoy.Demir=518;

Kaklik.Kosekoy.Kara=464;
Kaklik.Kosekoy.Demir=589;

Kaklik.Halkali.Kara=574;
Kaklik.Halkali.Demir=775;

l13=[Kaklik.Gelemen.Kara Kaklik.Gelemen.Demir];
l14=[Kaklik.Usak.Kara Kaklik.Usak.Demir];
l15=[Kaklik.Hasanbey.Kara Kaklik.Hasanbey.Demir];
l16=[Kaklik.Gokkoy.Kara Kaklik.Gokkoy.Demir];
l17=[Kaklik.Kosekoy.Kara Kaklik.Kosekoy.Demir];
l18=[Kaklik.Halkali.Kara Kaklik.Halkali.Demir];

Gelemen.Usak.Kara=801;
Gelemen.Usak.Demir=1400;

Gelemen.Hasanbey.Kara=668;

Gelemen.Hasanbey.Demir=1437;

Gelemen.Gokkoy.Kara=938;
Gelemen.Gokkoy.Demir=1630;

Gelemen.Kosekoy.Kara=640;
Gelemen.Kosekoy.Demir=1632;

Gelemen.Halkali.Kara=770;
Gelemen.Halkali.Demir=1818;

m14=[Gelemen.Usak.Kara Gelemen.Usak.Demir];
m15=[Gelemen.Hasanbey.Kara Gelemen.Hasanbey.Demir];
m16=[Gelemen.Gokkoy.Kara Gelemen.Gokkoy.Demir];
m17=[Gelemen.Kosekoy.Kara Gelemen.Kosekoy.Demir];
m18=[Gelemen.Halkali.Kara Gelemen.Halkali.Demir];

Usak.Hasanbey.Kara=225;
Usak.Hasanbey.Demir=307;

Usak.Gokkoy.Kara=235;
Usak.Gokkoy.Demir=381;

Usak.Kosekoy.Kara=379;
Usak.Kosekoy.Demir=503;

Usak.Halkali.Kara=490;
Usak.Halkali.Demir=689;

n15=[Usak.Hasanbey.Kara Usak.Hasanbey.Demir];
n16=[Usak.Gokkoy.Kara Usak.Gokkoy.Demir];
n17=[Usak.Kosekoy.Kara Usak.Kosekoy.Demir];
n18=[Usak.Halkali.Kara Usak.Halkali.Demir];

Hasanbey.Gokkoy.Kara=340;
Hasanbey.Gokkoy.Demir=350;

Hasanbey.Kosekoy.Kara=217;
Hasanbey.Kosekoy.Demir=218;

Hasanbey.Halkali.Kara=328;
Hasanbey.Halkali.Demir=404;

o16=[Hasanbey.Gokkoy.Kara Hasanbey.Gokkoy.Demir];
o17=[Hasanbey.Kosekoy.Kara Hasanbey.Kosekoy.Demir];
o18=[Hasanbey.Halkali.Kara Hasanbey.Halkali.Demir];

Gokkoy.Kosekoy.Kara=303;
Gokkoy.Kosekoy.Demir=546;

Gokkoy.Halkali.Kara=340;
Gokkoy.Halkali.Demir=732;

p17=[Gokkoy.Kosekoy.Kara Gokkoy.Kosekoy.Demir];
p18=[Gokkoy.Halkali.Kara Gokkoy.Halkali.Demir];

Kosekoy.Halkali.Kara=131;
Kosekoy.Halkali.Demir=186;

q18=[Kosekoy.Halkali.Kara Kosekoy.Halkali.Demir];

D=[0 min(a2) min(a3) min(a4) min(a5) min(a6) min(a7) min(a8) min(a9) min(a10)
min(a11) min(a12) min(a13) min(a14) min(a15) min(a16) min(a17) min(a18);...
min(a2) 0 min(b3) min(b4) min(b5) min(b6) min(b7) min(b8) min(b9) min(b10)
min(b11) min(b12) min(b13) min(b14) min(b15) min(b16) min(b17) min(b18);...
min(a3) min(b3) 0 min(c4) min(c5) min(c6) min(c7) min(c8) min(c9) min(c10)
min(c11) min(c12) min(c13) min(c14) min(c15) min(c16) min(c17) min(c18);...
min(b4) min(b4) min(c4) 0 min(d5) min(d6) min(d7) min(d8) min(d9) min(d10)
min(d11) min(d12) min(d13) min(d14) min(d15) min(d16) min(d17) min(d18);...
min(b5) min(b5) min(c5) min(d5) 0 min(e6) min(e7) min(e8) min(e9) min(e10)
min(e11) min(e12) min(e13) min(e14) min(e15) min(e16) min(e17) min(e18);...
min(a6) min(b6) min(c6) min(d6) min(e6) 0 min(f7) min(f8) min(f9) min(f10)
min(f11) min(f12) min(f13) min(f14) min(f15) min(f16) min(f17) min(f18);...
min(a7) min(b7) min(c7) min(d7) min(e7) min(f7) 0 min(g8) min(g9) min(g10)
min(g11) min(g12) min(g13) min(g14) min(g15) min(g16) min(g17) min(g18);...
min(a8) min(b8) min(c8) min(d8) min(e8) min(f8) min(g8) 0 min(h9) min(h10)
min(h11) min(h12) min(h13) min(h14) min(h15) min(h16) min(h17) min(h18);...
min(a9) min(b9) min(c9) min(d9) min(e9) min(f9) min(g9) min(h9) 0 min(i10)
min(i11) min(i12) min(i13) min(i14) min(i15) min(i16) min(i17) min(i18);...
min(a10) min(b10) min(c10) min(d10) min(e10) min(f10) min(g10) min(h10) min(i10) 0
min(j11) min(j12) min(j13) min(j14) min(j15) min(j16) min(j17) min(j18);...
min(a11) min(b11) min(c11) min(d11) min(e11) min(f11) min(g11) min(h11) min(i11)
min(j11) 0 min(k12) min(k13) min(k14) min(k15) min(k16) min(k17) min(k18);...
min(a12) min(b12) min(c12) min(d12) min(e12) min(f12) min(g12) min(h12)
min(i12) min(j12) min(k12) 0 min(l13) min(l14) min(l15) min(l16) min(l17) min(l18);...
min(a13) min(b13) min(c13) min(d13) min(e13) min(f13) min(g13) min(h13) min(i13)
min(j13) min(k13) min(l13) 0 min(m14) min(m15) min(m16) min(m17) min(m18);...
min(a14) min(b14) min(c14) min(d14) min(e14) min(f14) min(g14) min(h14) min(i14)
min(j14) min(k14) min(l14) min(m14) 0 min(n15) min(n16) min(n17) min(n18);...
min(a15) min(b15) min(c15) min(d15) min(e15) min(f15) min(g15) min(h15) min(i15)
min(j15) min(k15) min(l15) min(m15) min(n15) 0 min(o16) min(o17) min(o18);...
min(a16) min(b16) min(c16) min(d16) min(e16) min(f16) min(g16) min(h16) min(i16)
min(j16) min(k16) min(l16) min(m16) min(n16) min(o16) 0 min(p17) min(p18);...
min(a17) min(b17) min(c17) min(d17) min(e17) min(f17) min(g17) min(h17) min(i17)
min(j17) min(k17) min(l17) min(m17) min(n17) min(o17) min(p17) 0 min(q18);...
min(a18) min(b18) min(c18) min(d18) min(e18) min(f18) min(g18) min(h18)
min(i18) min(j18) min(k18) min(l18) min(m18) min(n18) min(o18) min(p18) min(q18) 0];

```

Model.D=D;
load('mesafe_tr_koor.mat')

MesafeMatrisi=D;

plot(trLon,trLat,'color','b','MarkerSize',20)
hold on
plot(x,y,'*r')
plot(AvpX,AvpY,'color','b','MarkerSize',20)

%%şehir çifti oluşturma ve uzaklık kare matrisini uzaklık mesafe vektörüne dönüştürme
fprintf('Şehir çiftleri oluşturma\n');
SehirSayisi = size(MesafeMatrisi,1); %%Şehir sayısı
c=1;
for count = 1:SehirSayisi:(SehirSayisi*SehirSayisi);
    SehirCifti(count:SehirSayisi*c, 1) = c;
    SehirCifti(count:SehirSayisi*c, 2) = 1:SehirSayisi;
    MesafeVektoru(count:SehirSayisi*c, 1) = MesafeMatrisi(c,:);
    c=c+1;
end
UzaklikMesafeVektoru=length(MesafeVektoru);

%%Eşitlik kısıtlamaları
fprintf('Eşitlik kısıtlamaları yaratılıyor \n');
Aeq=spones(1:length(SehirCifti));
beq=SehirSayisi;

%%Bir şehre yapılacak seyahat sayısı=1 şehirden yapılan seyahat sayısı=1
Aeq=[Aeq;spalloc(2*SehirSayisi,length(SehirCifti),2*SehirSayisi*(SehirSayisi+SehirSayisi-1))];

%%Eşitlik kısıtlamaları için belleği önceden tahsis etmek için seyrek bir matris tahsis etmemiz gerekir.
c=1;
for count = 1:2:((2*SehirSayisi)-1);
    SutunToplami=sparse(SehirCifti(:,2)==c);
    Aeq(count+1,:)=SutunToplami';%%Sınırlama matrisine dahil et
    SatirToplami=SehirCifti(:,1)==c;
    Aeq(count+2,:)=SatirToplami';
    c=c+1;
end
beq=[beq;ones(2*SehirSayisi,1)];

%%Mevcut olmayan yollar
nonExists=sparse(MesafeVektoru==0);
Aeq(2*c,:)=nonExists';
beq=[beq;0];

%%Karar değişkenlerini ikili değişken olarak atama

```

```

intcon=1:UzaklikMesafeVekturu;
lb=zeros(UzaklikMesafeVekturu,1);
ub=ones(UzaklikMesafeVekturu,1);

%%İntlinprog kullanarak optimize etme
fprintf('Sorunu Çözme \n');
opts=optimoptions('intlinprog','CutGeneration','Advanced','NodeSelection','mininfeas','Display','off');
[KararDegiskenleri,OptimumMaliyet,CikisNoktasi,Cikti]=intlinprog(MesafeVekturu,intcon,[],[],Aeq,beq,lb,ub,opts);
hold on
segments = find(KararDegiskenleri);
lh = zeros(SehirSayisi,1);
lh = updateSalesmanPlot(lh,KararDegiskenleri,SehirCifti,x,y);
%% alt turları algılama
Turlar=detectSubtours(KararDegiskenleri,SehirCifti);
AltTurSayisi=length(Turlar);
fprintf('alt tur sayıları: %d\n',AltTurSayisi);

%% Alt tur sınırlamaları
A=spalloc(0,UzaklikMesafeVekturu,0); %% seyrek eşitsizlik kısıtı matrisi oluşturma
b=[];
while AltTurSayisi>1 %%Sadece bir alt tur olana kadar tekrarla
    b=[b;zeros(AltTurSayisi,1)]; %% RHS Eşitsizlik sınırlarını girme
    A=[A;spalloc(AltTurSayisi,UzaklikMesafeVekturu,SehirSayisi)]; %%LHS eşitsizlik
    kısıtlamalarına girme
    for count=1:AltTurSayisi
        EsitsizlikKisitNumarasi=size(A,1)+1;
        AltTurNo=Turlar{count}; %% Alt turları birer birer çıkarma
        %%alt tur kısıtlamaları ekleme (eşitsizlik kısıtlamaları)
        AltTurCifti=nchoosek(1:length(AltTurNo),2);
        for JJ=1:size(AltTurCifti,1) %% Geçerli alturla ilgili değişkenleri bulma
            AltTurDegiskeni=(sum(SehirCifti==AltTurNo(AltTurCifti(JJ,1)),2))&...
                (sum(SehirCifti==AltTurNo(AltTurCifti(JJ,2)),2));
            A(EsitsizlikKisitNumarasi,AltTurDegiskeni)=1;
        end
        b(EsitsizlikKisitNumarasi)=length(AltTurNo)-1; % Alt turu birer birer çıkarma ve alt
        nokta kısıtlamaları ekleyerek eşitsizlik kısıtlamaları oluşturma, A-B-A:2->1
    end
    fprintf('\n Alt turları elimine ederek sorunu tekrar çözme\n');

[KararDegiskenleri,OptimumMaliyet,CikisNoktasi,Cikti]=intlinprog(MesafeVekturu,intcon,A,b,Aeq,beq,lb,ub,opts);
%% Alt turları tekrar kontrol et
fprintf('Alt turları tekrar kontrol et\n');
lh = updateSalesmanPlot(lh,KararDegiskenleri,SehirCifti,x,y);
Turlar=detectSubtours(KararDegiskenleri,SehirCifti);
AltTurSayisi=length(Turlar);
fprintf('Alt tur sayıları: %d\n',AltTurSayisi);

```

end

```
%%Çözüm Kalite değeri ne kadar küçükse, çözüm de o kadar iyi olur
fprintf('Çözüm kalitesi: %f (ne kadar azsa o kadar iyi)\n',Cikti.absolutegap);
fprintf('Optimize Edilmiş tur rotaları:');
celldisp(Turlar);
fprintf('not: Sayılar giriş dosyasındaki şehirlerin sırasına karşılık gelir\n');
fprintf('Optimum yolun toplam mesafesi: %d\n',OptimumMaliyet);
```

Ek-7. Lineer Optimizasyon Kodlaması: Multimodal Mesafe Örneğinde Kullanılan 18 Nokta Arasından Seçilen 4 Noktadan Oluşan Mesafe Örneği Kodlaması

```
clear all;
close all;
clc;

Iskenderun.Mersin.Kara=207;
Iskenderun.Mersin.Demir=204;
Iskenderun.Mersin.Deniz=122;

Iskenderun.Antalya.Kara=745;
Iskenderun.Antalya.Deniz=454;

Iskenderun.Usak.Kara=804;
Iskenderun.Usak.Demir=908;

a2=[Iskenderun.Mersin.Kara Iskenderun.Mersin.Demir Iskenderun.Mersin.Deniz];
a3=[Iskenderun.Antalya.Kara Iskenderun.Antalya.Deniz];
a4=[Iskenderun.Usak.Kara Iskenderun.Usak.Demir];

Mersin.Antalya.Kara=502;
Mersin.Antalya.Deniz=357;

Mersin.Usak.Kara=694;
Mersin.Usak.Demir=789;

b3=[Mersin.Antalya.Kara Mersin.Antalya.Deniz];
b4=[Mersin.Usak.Kara Mersin.Usak.Demir];

Antalya.Usak.Kara=302;

c4=[Antalya.Usak.Kara];

D=[0 min(a2) min(a3) min(a4);...
   min(a2) 0 min(b3) min(b4);...
   min(a3) min(b3) 0 min(c4);...
   min(a4) min(b4) min(c4) 0];
```

```

Model.D=D;
%Yukarıdaki kodlama sonucunda 4 şehir arasındaki demiryolu, karayolu ve denizyolu
mesafelerinden en kısa olanı bulup D matrisinin içine atmak için gereken kodlama

load('mesafe_tr_koor_4sehir.mat') % önceden hazırlanmış Türkiye'nin ve kullanılan 4
şehrin koordinatlarını içeren matlab uzantılı dosyayı çağırmak için kullanılan komut

MesafeMatrisi=D; % belirlenen minimum değerleri mesafe matrisi haline getirmek için
gereken kodlama

% Türkiye'nin sınır koordinatlarını kullanarak Türkiye haritasını çizmek ve 4 şehrin
koordinatlarını kullanarak Türkiye haritasına aplike etmek için gereken kodlama
plot(trLon,trLat,'color','b','MarkerSize',20)
hold on
plot(x,y,'*r')
plot(AvpX,AvpY,'color','b','MarkerSize',20)

%şehir çifti oluşturma ve uzaklık kare matrisini uzaklık mesafe vektörüne dönüştürme
fprintf('Şehir çiftleri oluşturma\n');
SehirSayisi = size(MesafeMatrisi,1); %%Şehir sayısı
c=1;
for count = 1:SehirSayisi:(SehirSayisi*SehirSayisi);
    SehirCifti(count:SehirSayisi*c, 1) = c;
    SehirCifti(count:SehirSayisi*c, 2) = 1:SehirSayisi;
    MesafeVektoru(count:SehirSayisi*c, 1) = MesafeMatrisi(c,:);
    c=c+1;
end
UzaklikMesafeVektoru=length(MesafeVektoru);

%%Eşitlik kısıtlamaları
fprintf('Eşitlik kısıtlamaları yaratılıyor \n');
Aeq=spones(1:length(SehirCifti));
beq=SehirSayisi;
%%Bir şehre yapılacak seyahat sayısı=1 şehirden yapılan seyahat sayısı=1 olması için
gereken kodlama
Aeq=[Aeq;spalloc(2*SehirSayisi,length(SehirCifti),2*SehirSayisi*(SehirSayisi+SehirSayi
si-1))];
%%Eşitlik kısıtlamaları için belleği önceden tahsis etmek için seyrek bir matris tahsis
etmemiz gerekir.
c=1;
for count = 1:2:((2*SehirSayisi)-1);
    SutunToplami=sparse(SehirCifti(:,2)==c);
    Aeq(count+1,:)=SutunToplami'; %%Sınırlama matrisine dahil et
    SatirToplami=SehirCifti(:,1)==c;
    Aeq(count+2,:)=SatirToplami';
    c=c+1;
end
beq=[beq;ones(2*SehirSayisi,1)];
%%Mevcut olmayan yollar
nonExists=sparse(MesafeVektoru==0);

```

```

Aeq(2*c,:)=nonExists';
beq=[beq;0];
%%Karar deęişkenlerini ikili deęişken olarak atama
intcon=1:UzaklikMesafeVektoru;
lb=zeros(UzaklikMesafeVektoru,1);
ub=ones(UzaklikMesafeVektoru,1);
%%İntlinprog komutunu kullanarak optimize etme
fprintf('Sorunu Çözme \n');
opts=optimoptions('intlinprog','CutGeneration','Advanced','NodeSelection','mininfeas','Display','off');
[KararDegiskenleri,OptimumMaliyet,CikisNoktasi,Cikti]=intlinprog(MesafeVektoru,intcon,[],[],Aeq,beq,lb,ub,opts);
hold on
segments = find(KararDegiskenleri);
lh = zeros(SehirSayisi,1);
lh = updateSalesmanPlot(lh,KararDegiskenleri,SehirCifti,x,y);
%% alt turları algılama
Turlar=detectSubtours(KararDegiskenleri,SehirCifti);
AltTurSayisi=length(Turlar);
fprintf('alt tur sayıları: %d\n',AltTurSayisi);
%% Alt tur sınırlamaları
A=spalloc(0,UzaklikMesafeVektoru,0); %% seyrek eşitsizlik kısıtı matrisi oluşturma
b=[];
while AltTurSayisi>1 %%Sadece bir alt tur olana kadar tekrarla
    b=[b;zeros(AltTurSayisi,1)]; %% RHS Eşitsizlik sınırlarını girme
    A=[A;spalloc(AltTurSayisi,UzaklikMesafeVektoru,SehirSayisi)]; %%LHS eşitsizlik
    kısıtlamalarına girme
    for count=1:AltTurSayisi
        EsitsizlikKisitNumarasi=size(A,1)+1;
        AltTurNo=Turlar{count}; %% Alt turları birer birer çıkarma
        %%alt tur kısıtlamaları ekleme (eşitsizlik kısıtlamaları)
        AltTurCifti=nchoosek(1:length(AltTurNo),2);
        for JJ=1:size(AltTurCifti,1) %% Geçerli alturla ilgili deęişkenleri bulma
            AltTurDegiskeni=(sum(SehirCifti==AltTurNo(AltTurCifti(JJ,1)),2))&...
                (sum(SehirCifti==AltTurNo(AltTurCifti(JJ,2)),2));
            A(EsitsizlikKisitNumarasi,AltTurDegiskeni)=1;
        end
        b(EsitsizlikKisitNumarasi)=length(AltTurNo)-1; % Alt turu birer birer çıkarma ve alt
        nokta kısıtlamaları ekleyerek eşitsizlik kısıtlamaları oluşturma, A-B-A:2->1
    end
    fprintf('\n Alt turları elimine ederek sorunu tekrar çözme\n');
    [KararDegiskenleri,OptimumMaliyet,CikisNoktasi,Cikti]=intlinprog(MesafeVektoru,intcon,A,b,Aeq,beq,lb,ub,opts);
    %% Alt turları tekrar kontrol et
    fprintf('Alt turları tekrar kontrol et\n');
    lh = updateSalesmanPlot(lh,KararDegiskenleri,SehirCifti,x,y);
    Turlar=detectSubtours(KararDegiskenleri,SehirCifti);
    AltTurSayisi=length(Turlar);
    fprintf('Alt tur sayıları: %d\n',AltTurSayisi);
end

```

```
%%Çözüm Kalite değeri ne kadar küçükse, çözüm de o kadar iyi olur
fprintf('Çözüm kalitesi: %f (ne kadar azsa o kadar iyi)\n',Cikti.absolutegap);
fprintf('Optimize Edilmiş tur rotaları:');
celldisp(Turlar);
fprintf('not: Sayılar giriş dosyasındaki şehirlerin sırasına karşılık gelir\n');
fprintf('Optimum yolun toplam mesafesi: %d\n',OptimumMaliyet);
```



ÖZGEÇMİŞ**Kişisel Bilgiler**

Soyadı, adı : BOZKURT, Mehmet Ali
Uyruğu : T.C.
Doğum tarihi ve yeri : 09.03.1987, İskenderun
Medeni hali : Evli
Telefon : 0 537 762 8347
Faks : -
e-mail : mehmetalibozkurt86@hotmail.com

**Eğitim**

Derece	Eğitim Birimi	Mezuniyet Tarihi
Yüksek lisans	İskenderun Teknik Üniversitesi / İnşaat Mühendisliği	Devam ediyor
Lisans	Doğu Akdeniz Üniversitesi / İnşaat Mühendisliği	2011
Lise	İskenderun Lisesi	2003

İş Deneyimi

Yıl	Yer	Görev
2012-Halen	MESA MÜHENDİSLİK	Sahibi

Yabancı Dil

İngilizce

DİZİN

A

Abstract · v
 Analiz · iv, 11, 12, 16, 18, 21, 22,
 23, 49, 50, 51, 54, 55, 56, 57,
 59, 61, 63, 65, 66, 68, 69, 70

B

Balıkesir · 29, 30, 31, 32, 33, 35,
 36, 39, 40, 48, 49, 50, 52, 53,
 54, 55, 57, 58, 60, 62, 64, 65,
 67
 Bandırma · 29, 30, 31, 32, 33, 34,
 35, 36, 37, 39, 40, 41, 48, 49,
 50, 52, 53, 54, 55, 56, 58, 60,
 62, 64, 65, 67
 Birim Maliyet · iv, viii, ix, xi,
 xiii, xiv, 2, 26, 37, 38, 39, 40,
 41, 51, 52, 56, 57, 61, 62, 63,
 66, 68, 69, 70, 71,

Ç

Çok türlü · iv, vii, ix, x, xiv, 2, 3,
 4, 5, 6, 26, 27, 37, 48, 57, 59,
 61, 62, 63, 64, 65, 66, 68, 69,
 70, 71,

D

Demiryolu · iv, viii, xi, 4, 5, 17,
 26, 27, 29, 30, 31, 33, 34, 36,
 37, 38, 40, 58, 59, 60, 61, 62,
 64, 65, 66, 67,
 Denizli · 29, 30, 31, 32, 33, 35,
 36, 39, 40, 48, 49, 50, 52, 53,
 54, 55, 57, 58, 60, 62, 64, 65
 Denizyolu · viii, xi, 4, 5, 26, 27,
 29, 30, 31, 34, 37, 38, 41, 57,
 58, 59, 60, 61, 62, 63, 65, 66,
 67, 70,

E

Eskişehir · 29, 30, 31, 32, 33, 35,
 36, 39, 40, 48, 49, 50, 52, 53,
 54, 55, 57, 58, 60, 62, 64, 65,
 67

G

Gelemen · 29, 30, 31, 32, 33, 35,
 36, 39, 40, 48, 49, 50, 52, 53,
 54, 55, 57, 58, 60, 62, 64, 65,
 67

Gerçek · xiii, xiv, 2, 3, 5, 8, 11,
 12, 13, 15, 17, 19, 20, 21, 26,
 30, 48, 49, 50, 51, 52, 54, 55,
 56, 57
 Gezin Satıcı · xv, 7, 10, 11, 13,
 20, 22, 23, 24,
 Güzergah · iv, vii, viii, ix, xi, xii,
 xiii, 18, 26, 27, 28, 29, 30, 34,
 37, 45, 46, 47, 48, 50, 51, 52,
 53, 54, 56, 57, 59, 61, 62, 64,
 66, 68

H

Hesap · iv, ix, 2, 9, 10, 11, 12, 15,
 16, 17, 20, 21, 23, 24, 25, 26,
 30, 34, 37, 38, 46, 49, 50, 51,
 52, 54, 55, 56, 57, 59, 61, 63,
 65, 66, 68, 69, 70,

İ

İskenderun · 29, 30, 31, 32, 33,
 34, 35, 36, 37, 39, 40, 41, 48,
 49, 50, 52, 53, 54, 55, 56, 58,
 60, 62, 64, 65, 67
 İzmir · 29, 30, 31, 32, 33, 34, 35,
 36, 37, 39, 40, 41, 48, 49, 50,
 52, 53, 54, 55, 56, 58, 60, 62,
 64, 65, 67

K

Kaklık · 29, 30, 31, 32, 33, 35,
 36, 39, 40, 48, 49, 50, 52, 53,
 54, 55, 57, 58, 60, 62, 64, 65
 Karayolu · iv, viii, ix, xi, xiv, 3,
 4, 5, 26, 27, 29, 30, 31, 32, 34,
 35, 37, 38, 39, 48, 49, 50, 51,
 52, 54, 55, 56, 57, 58, 59, 60,
 61, 62, 63, 64, 65, 66, 67, 68,
 69, 70
 Kayseri · 29, 30, 31, 32, 33, 35,
 36, 39, 40, 48, 49, 50, 52, 53,
 54, 55, 57, 58, 60, 62, 64, 65

L

Lineer · iv, vii, viii, ix, x, 2, 7, 8,
 24, 26, 37, 41, 48, 49, 50, 51,
 59, 61, 68, 69, 70
 Liman · vii, viii, 2, 5, 26, 27, 29,
 30, 31, 32, 33, 34, 35, 36, 37,
 38, 39, 40, 41, 48, 49, 50, 51,
 52, 53, 54, 55, 56, 57, 58, 59,
 60, 61, 62, 63, 64, 65, 66, 67
 Lojistik · vii, viii, 2, 3, 4, 5, 10,
 14, 19, 26, 27, 29, 30, 31, 32,

3, 34, 35, 37, 48, 49, 50, 51,
 52, 53, 54, 55, 56, 57, 58, 59,
 60, 61, 62, 63, 64, 65, 66, 67,
 70

M

Mersin · 29, 30, 31, 32, 33, 34,
 35, 36, 37, 39, 40, 41, 48, 49,
 50, 52, 53, 54, 55, 56, 58, 60,
 62, 64, 65, 67
 Mesafe · iv, viii, ix, x, xi, xiii,
 xiv, 2, 3, 4, 5, 7, 12, 16, 21,
 23, 24, 26, 27, 29, 30, 31, 32,
 33, 34, 37, 38, 41, 42, 44, 45,
 46, 47, 48, 49, 50, 52, 53, 54,
 55, 57, 59, 63, 64, 68, 69, 70

O-Ö

Optimizasyon · iv, vii, viii, ix, x,
 2, 3, 7, 8, 10, 13, 14, 15, 16,
 24, 26, 27, 37, 41, 45, 48, 49,
 50, 51, 52, 54, 55, 56, 57, 59,
 61, 63, 65, 68, 69, 70
 Özet · iv

S

Samsun · 29, 30, 31, 32, 33, 35,
 36, 39, 40, 48, 49, 50, 52, 53,
 54, 55, 57, 58, 60, 62, 64, 65
 Süre · iv, viii, ix, xi, xiii, xiv, 2, 3,
 4, 5, 6, 10, 11, 12, 14, 15, 16,
 17, 18, 19, 20, 21, 22, 24, 26,
 27, 34, 35, 36, 37, 48, 50, 51,
 55, 56, 59, 61, 65, 66, 68, 69,
 70

T

Tek türlü · vii, 2, 3, 4, 5, 26,

U

Ulaşım · iv, 2, 3, 4, 5, 6, 18, 20,
 24, 26, 27, 57, 59, 61, 63, 65,
 66

Y

Yol · 3, 7, 10, 15, 23, 24, 25



TEKNOVERSİTE



teknoversite AYRICALIĞINDASINIZ

İSTE

