

Haydar GÜNDOĞDU



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

İNŞAAT MÜHENDİSLİĞİ ANA BİLİM DALI

**YÜKSEK  
LİSANS  
TEZİ**

**YÜK TAŞIMACILIĞINDA  
OPTİMİZASYON TEKNİKLERİNİN  
KARŞILAŞTIRILMASI**

**Haydar GÜNDOĞDU**

**İNŞAAT MÜHENDİSLİĞİ  
ANABİLİM DALI**

**HAZİRAN 2020**

HAZİRAN 2020



**YÜK TAŞIMACILIĞINDA OPTİMİZASYON TEKNİKLERİNİN  
KARŞILAŞTIRILMASI**

**Haydar GÜNDOĞDU**

**YÜKSEK LİSANS TEZİ  
İNŞAAT MÜHENDİSLİĞİ ANABİLİM DALI**

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

**HAZİRAN 2020**

Haydar GÜNDOĞDU tarafından hazırlanan “YÜK TAŞIMACILIĞINDA OPTİMİZASYON TEKNİKLERİNİN 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:** Doç.Dr. Talha EKMEKYAPAR  
İnşaat Mühendisliği Anabilim Dalı, Gaziantep Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum.

.....  
.....

Tez Savunma Tarihi: 23/06/2020

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

Haydar GÜNDOĞDU

23/06/2020

YÜK TAŞIMACILIĞINDA OPTİMİZASYON TEKNİKLERİNİN KARŞILAŞTIRILMASI  
(Yüksek Lisans Tezi)

Haydar GÜNDOĞDU

İSKENDERUN TEKNİK ÜNİVERSİTESİ  
MÜHENDİSLİK VE FEN BİLİMLERİ ENSTİTÜSÜ

Haziran 2020

ÖZET

Ülkemizde yolcu ve yük taşımacılığının yaklaşık % 90'nı diğer taşımacılık sistemleri ile karşılaştırıldığında daha düşük altyapı maliyetlerine sahip karayolları taşımacılığı ile gerçekleştirilmektedir. Ülkemizde son yıllarda mevcut karayolu ağlarındaki yatırımlar ile birlikte hava, demir, deniz ve boru taşımacılığı altyapılarında da ciddi yatırımlar gerçekleştirilerek taşımacılık sektörü güçlendirilmektedir. Taşımacılık sistemlerinin altyapılarının iyileştirilmesinin yanında taşıma mesafelerinin kısaltılması ve farklı taşımacılık sistemlerinde gerekli bağlantıları sağlayarak çok türlü taşımacılığın güçlendirilmesi ile taşımacılık maliyetlerinin azaltılması ülke ekonomilerine ciddi katkıları olmaktadır. Bu tez çalışmasında, taşımacılık maliyetlerinin düşürülmesi çalışmalarına katkı sağlamak amacı ile ülkemizdeki 81 şehirde bulunan Organize Sanayi Bölgeleri (OSB) arasında karayolu taşımacılığının optimum mesafe değeri hesaplanmaktadır. Yapılan çalışmada Gezgin Satıcı Problemi (GSP) baz alınarak Genetik Algoritma (GA), Lineer Optimizasyon (LO) ve Parçacıklı Sürü Optimizasyon (PSO) yöntemleri kullanılmaktadır. LO ile yapılan analizler sonucunda elde edilen optimum mesafe değeri 9923 km olarak hesaplanmaktadır. GA ile yapılan analizler sonucunda elde edilen optimum mesafe değeri 10689 km olarak hesaplanmaktadır. PSO ile yapılan analizler sonucunda elde edilen optimum mesafe değeri 11313 km olarak hesaplanmaktadır. LO, PSO ve GA yöntemleri sonucu elde edilen değerler birbirleri ile kıyaslandığında, Lineer Optimizasyonun genel anlamda daha iyi sonuç verdiği görülmektedir.

Anahtar Kelimeler : Genetik Algoritma, GSP, Lineer Optimizasyon, PSO

Sayfa Sayısı : 96

Danışman : Dr. Öğr. Üyesi Ömer Faruk CANSIZ

## COMPARISON OF OPTIMIZATION TECHNIQUES IN FREIGHT TRANSPORTATION

(M. Sc. Thesis)

Haydar GÜNDOĞDU

ISKENDERUN TECHNICAL UNIVERSITY  
ENGINEERING AND SCIENCE INSTITUTE

June 2020

## ABSTRACT

In our country, approximately 90% of passenger and cargo transportation is carried out by road transport with lower infrastructure costs compared to other transportation systems. The transportation sector is strengthened by making serious investments in the air, iron, sea and pipe transportation infrastructures along with the investments in the existing road networks in our country in recent years. In addition to improving the infrastructure of transportation systems, shortening the transportation distances and strengthening many kinds of transportation by providing the necessary connections in different transportation systems, and reducing transportation costs have serious contributions to the national economies. In this thesis, the optimum distance value of road transportation is calculated between the Organized Industrial Zones (OIZ) located in 81 cities in our country in order to contribute to the reduction of transportation costs. In the study, Genetic Algorithm (GA), Linear Optimization (LO) and Particle Swarm Optimization (PSO) methods are used based on the Traveling Salesman Problem (TSP). The optimum distance obtained as a result of the analyzes made with LO is calculated as 9923 km. The optimum distance obtained as a result of the analyzes with GA is calculated as 10689 km. The optimum distance obtained as a result of the analyzes with PSO is calculated as 11313 km. When the values obtained as a result of LO, PSO and GA methods are compared, it is seen that Linear Optimization generally gives better results

Key Words : Genetic Algorithm, TSP, Linear Optimization, PSO

Page Number : 96

Supervisor : Assist. Prof. Dr. Ömer Faruk CANSIZ

## TEŐEKKÜR

Yüksek Lisans tez konusunun belirlenmesinde, araştırılması ve yazımı sırasında sahip olduđu bilgi birikimi ve tecrübesi ile çalışmayı yönlendiren ve her türlü yardımı esirgemeyen saygıdeđer danışman hocam Dr.Öğr.Üyesi Ömer Faruk CANSIZ' a sonsuz saygı ve teşekkürlerimi sunarım.



## İÇİNDEKİLER

	<b>Sayfa</b>
ÖZET .....	iv
ABSTRACT .....	v
TEŞEKKÜR .....	vi
İÇİNDEKİLER .....	vii
ÇİZELGELERİN LİSTESİ .....	x
ŞEKİLLERİN LİSTESİ .....	xii
SİMGELER VE KISALTMALAR.....	xiii
<b>1. GİRİŞ</b> .....	<b>1</b>
1.1. Lojistik .....	3
1.2. Türkiye’de Karayolu Taşımacılığı.....	3
1.3. Gezgin Satıcı Problemi .....	4
1.3.1. Kesin algoritmalar .....	5
1.3.2. Sezgisel algoritmalar .....	5
1.3.3. Turu oluşturan algoritmalar .....	5
1.3.4. Turu geliştiren algoritmalar .....	5
1.3.5. Melez algoritmalar .....	5
1.4. Lineer Optimizasyon.....	6
1.5. Parçacıklı Sürü Optimizasyonu .....	7
1.6. Genetik Algoritma .....	7
<b>2. ÖNCEKİ ÇALIŞMALAR</b> .....	<b>9</b>
<b>3. MATERYAL VE YÖNTEM</b> .....	<b>23</b>
3.1. Çalışmanın Amacı.....	23
3.2. Dört OSB’ye Ait Lineer Optimizasyon Algoritmasının Oluşturulması .....	32



	<b>Sayfa</b>
3.3. Dört OSB'ye Ait Genetik Algoritmasının Oluşturulması .....	42
3.4. Dört OSB'ye Ait PSO Algoritmasının Oluşturulması .....	47
<b>4. ARAŞTIRMA BULGULARI VE TARTIŞMA .....</b>	<b>54</b>
4.1. Lineer Optimizasyon İle Koordinatlara Göre Standart Çözümü .....	54
4.1.1. Güzergâhın Türkiye haritasında gösterimi .....	54
4.2. Lineer Optimizasyon İle Karayolu Mesafelerine Göre Çözümü .....	55
4.2.1. Güzergâhın Türkiye haritasında gösterimi .....	55
4.3. Parçacıklı Sürü Optimizasyon İle Koordinatlara Göre Standart Çözümü .....	55
4.3.1. Güzergâhın Türkiye haritasında gösterimi .....	56
4.4. Parçacıklı Sürü Optimizasyon İle Karayolu Mesafelerine Göre Çözümü .....	56
4.4.1. Güzergâhın Türkiye haritasında gösterimi .....	56
4.5. Genetik Algoritma İle Koordinatlara Göre Standart Çözümü .....	57
4.5.1. Güzergâhın Türkiye haritasında gösterimi .....	57
4.6. Genetik Algoritma İle Karayolu Mesafelerine Göre Çözümü .....	57
4.6.1. Güzergâhın Türkiye haritasında gösterimi .....	58
<b>5. SONUÇLAR VE ÖNERİLER .....</b>	<b>59</b>
<b>KAYNAKLAR .....</b>	<b>60</b>
<b>EKLER .....</b>	<b>65</b>
EK-1. Lineer Optimizasyon Algoritması .....	65
EK-2. PSO Ana Algoritması .....	68
EK-3. PSO Model Algoritması .....	71
EK-4. PSO Mutasyon Algoritması .....	84
EK-5. PSO Hesaplama Algoritması .....	86
EK-6. PSO Yazdırma Algoritması .....	87

	<b>Sayfa</b>
EK-7. GA Ana Algoritması .....	88
EK-8. GA Model Algoritması .....	90
EK-9. GA Mutasyon Algoritması .....	91
EK-10. GA Hesaplama Algoritması .....	92
EK-11. GA Yazdırma Algoritması .....	93
ÖZGEÇMİŞ.....	94
DİZİN .....	95

## ÇİZELGELERİN LİSTESİ

<b>Çizelge</b>	<b>Sayfa</b>
Çizelge 3.1. OSB'ye ait koordinat bilgileri (Google Earth).....	24
Çizelge 3.2. OSB'ye ait km değerleri (Google Earth) .....	26
Çizelge 3.3. Şehirlerarasındaki mesafeleri alıp matris haline getiren kodlama .....	32
Çizelge 3.4. Şehirlerarasındaki mesafeleri matris haline getiren kodlamanın sonucu ...	32
Çizelge 3.5. OSB'lerin Türkiye haritası üzerinde gösteren kodlama .....	32
Çizelge 3.6. Şehir çiftlerini ve mesafe vektörünü oluşturan kodlama .....	33
Çizelge 3.7. Şehir çiftlerini ve mesafe vektörünü oluşturan kodlamanın sonucu.....	33
Çizelge 3.8. Eşitlik kısıtları kodlamaları .....	34
Çizelge 3.9. Eşitlik kısıtları kodlaması sonuçları .....	34
Çizelge 3.10. Eşitlik kısıtları kodlamaları .....	34
Çizelge 3.11. Eşitlik kısıtları kodlaması sonuçları .....	35
Çizelge 3.12. Olmayan yolların belirlenmesinde kullanılan kodlama .....	37
Çizelge 3.13. Olmayan yolların belirlenmesinde kullanılan kodlama sonucu.....	37
Çizelge 3.14. Alt üst sınırlamaların belirtildiği kodlar.....	37
Çizelge 3.15. Alt üst sınırlamaların belirtildiği kodların sonuçları .....	38
Çizelge 3.16. Dört OSB için güzergâhın belirlenmesinde kullanılan kodlama .....	38
Çizelge 3.17. Dört OSB için güzergâhın belirlenmesinde kullanılan kodlama sonucu .	39
Çizelge 3.18. Alttur turların belirlenmesinde kullanılan kodlama .....	39
Çizelge 3.19. Alttur turların belirlenmesinde kullanılan kodlama sonucu.....	39
Çizelge 3.20. Alt turları eleyerek optimum güzergâhı oluşturan kodlama .....	39
Çizelge 3.21. Alt turları eleyerek optimum güzergâhı oluşturan kodlama sonucu.....	41
Çizelge 3.22. Create_permutations.m dosyasındaki kodlama .....	43
Çizelge 3.23. Crossover_permutations.m dosyasındaki kodlama .....	44

<b>Çizelge</b>	<b>Sayfa</b>
Çizelge 3.24. Crossover_permutations.m dosyasındaki kodlama .....	44
Çizelge 3.25. Traveling_salesman_fitness.m dosyasındaki kodlama.....	45
Çizelge 3.26. Traveling_salesman_plot.m dosyasındaki kodlama .....	45
Çizelge 3.27. Genetik algoritma genel kodlaması birinci kısım.....	46
Çizelge 3.28. Genetik algoritma genel kodlaması ikinci kısım .....	46
Çizelge 3.29. Genetik algoritma genel kodlaması üçüncü kısım.....	46
Çizelge 3.30. Genetik algoritma genel kodlaması dördüncü kısım.....	47
Çizelge 3.31. CreateModel.m dosyasındaki kodlama .....	47
Çizelge 3.32. CreateModel.m dosyasındaki kodlama çıktısı .....	48
Çizelge 3.33. Mutate.m dosyasındaki kodlama .....	48
Çizelge 3.34. MyCost.m dosyasındaki kodlama.....	49
Çizelge 3.35. PlotSolution.m dosyasındaki kodlama .....	50
Çizelge 3.36. PSO genel algoritmasının birinci kısmı .....	50
Çizelge 3.37. PSO genel algoritmasının ikinci kısmı .....	51
Çizelge 3.38. PSO genel algoritmasının üçüncü kısmı .....	51
Çizelge 3.39. PSO genel algoritmasının dördüncü kısmı.....	51
Çizelge 3.40. PSO genel algoritmasının beşinci kısmı.....	52
Çizelge 3.41. PSO genel algoritmasının altıncı kısmı.....	52

**ŞEKİLLERİN LİSTESİ**

<b>Şekil</b>	<b>Sayfa</b>
Şekil 3.1. OSB'lerin Türkiye haritası üzerindeki görünümü.....	33
Şekil 3.2. Oluşan alt turların harita üzerinde gösterimi.....	40
Şekil 3.3. Güzergâhın harita üzerinde gösterimi.....	42
Şekil 4.1. LO GSP örneğinin koordinatlara göre çözümünün harita üzerinde gösterimi .....	54
Şekil 4.2. LO GSP örneğinin gerçek karayolu mesafelere göre çözümünün harita üzerinde gösterimi .....	55
Şekil 4.3. PSO GSP örneğinin koordinatlara göre çözümünün harita üzerinde gösterimi .....	55
Şekil 4.4. PSO GSP örneğinin gerçek karayolu mesafelere göre çözümünün harita üzerinde gösterimi .....	56
Şekil 4.5. GA GSP örneğinin koordinatlara göre çözümünün harita üzerinde gösterimi .....	57
Şekil 4.6. GA GSP örneğinin gerçek karayolu mesafelere göre çözümünün harita üzerinde gösterimi .....	57

## SİMGELER VE KISALTMALAR

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

<b>Kısaltmalar</b>	<b>Açıklamalar</b>
<b>TSP</b>	Traveling Selesman Problem
<b>CBS</b>	Coğrafi Bilgi Sistemleri
<b>GA</b>	Genetik Algoritma
<b>GSP</b>	Gezgin Satıcı Problemi
<b>PSO</b>	Parçacıklı Sürü Optimizasyonu
<b>OSB</b>	Organize Sanayi Bölgeleri
<b>GOA</b>	Guguk Kuşu Optimizasyon Algoritması
<b>ACO</b>	Karınca Kolonisi Optimizasyonu
<b>SFLA</b>	Karışık Kurbağa Sıçrama Algoritmaları
<b>A3</b>	Yapay Atom Algoritması
<b>LO</b>	Lineer Optimizasyon
<b>KGM</b>	Karayolları Genel Müdürlüğü

## 1. GİRİŞ

Ekonomi ile teknolojiadaki ileri seviyedeki iyileşmeler lojistikteki rekabeti küresel bir hal almaya itmektedir. Lojistik şirketleri, küresel seviyedeki yerlerini koruyabilmeleri için kaynakları artan taleplerine zamanında en hızlı şekilde ulaştırmak zorundadır. Kaynak maliyetleri üreticilerin problemleri olmakla beraber lojistikteki asıl rekabeti etkileyici faktör mesafe maliyet ilişkisidir. Mesafe maliyet ilişkisini etkileyen faktörlerin başında olan taşımacılık ağlarının biçimleri ve durumları, gelişmiş ülkelerde ağırlık olarak demiryolu, denizyolu, boru ve hava yolu ile taşımacılığın yapılması ile çözülmüş gözükme ile birlikte ekonomik ve teknolojik yetersizliklerden dolayı gelişmekte olan ülkelerde ise lojistiğin üzerinde sağlandığı ana taşımacılık sistemini karayolu taşımacılığı oluşturmaktadır. Ülkemizde yolcu ve yük taşımacılığının yaklaşık %90'nı diğer taşımacılık sistemleri ile karşılaştırıldığında daha düşük altyapı maliyetlerine sahip karayolları taşımacılığı ile gerçekleştirilmektedir.

Ülkemizde son yıllarda mevcut karayolu ağlarındaki yatırımlar ile birlikte hava, demir, deniz ve boru taşımacılığı altyapılarında da ciddi yatırımlar gerçekleştirilerek taşımacılık sektörü güçlendirilmektedir. Taşımacılık sistemlerinin altyapılarının iyileştirilmesi yanında taşıma mesafelerinin kısaltılması ve farklı taşımacılık sistemlerinde gerekli bağlantıları sağlayarak çok türlü taşımacılığın güçlendirilmesi ile taşımacılık maliyetlerinin azaltılması ülke ekonomilerine ciddi katkıları olmaktadır. Lojistik sektöründe yük taşımacılığı için birçok mesafe ve maliyet ilişkisi analizleri yapılmaktadır. Bunlardan bir tanesi Gezgin Satıcı Problemi (GSP) olarak geçmektedir. William Rowan Hamilton'un 1856'da incelediği, her köşede sadece bir defa bulunan ve Platonik katların birinin kenarlarının uzunluğunca hiçbir noktada açık olmayan yollar, grafiksel teoride bugün Hamilton turları olarak yerini almaktadır. GSP genel olarak anlaması ve az sayıda yerleşim yeri içinde çözümü de kolay olan fakat yerleşim yeri sayısı artıkça çözümlenmesi oldukça zor bir klasik birleşimsel optimizasyon problemlerinden biridir. Literatürde oldukça yer edinmiş ve çözüm analizleri kesin, sezgisel ve meta sezgisel teknikler ile yapılabilmektedir. GSP'nin çözüm analizleri en kısa Hamilton Turu'nu yani birden fazla yerleşim yerinde, her birinde sadece bir kez bulunacak biçimde ve ilk başlanan yerleşim yerine tekrardan dönmek ile oluşturulan güzergahların içinde en kısa güzergahın bir çizelgede bulmayı hedeflemektedir. GSP'deki en optimal güzergahları bulmakta yararlanılan bazı teknikler literatürde kullanılmakta ve yenilerinin araştırmaları da devam etmektedir. GSP çözüm

analizlerinin zorluğu çözüm sayısının yerleşim yeri sayısı ile faktöriyel olarak bağıntılı olmasından kaynaklanmakta olup, asimetrik olmayan bir problemde sonuç sayısı ( $n$  yerleşim yeri sayısı)  $-1)! / 2$  ile eşit olmaktadır. Buradan az sayıdaki yerleşim yeri için GSP çözümleri hem zaman açısından hem de kesin sonuçların bulunması yönünden mümkün olmakta, ancak yerleşim yeri sayısının ellili veya yüzülü yerleşim yeri olarak ifade edildiğinde ise bir GSP probleminin en kısa turunun bulunması her Hamilton turunu 10-9 saniyede tamamlama kapasitesindeki bir bilgisayar ile milyonlarca yıl sürebileceği gerçeği ortaya çıkmaktadır. Bu yüzden yerleşim yeri sayısı artıkça çözüm analizlerinde kullanılan fazla zaman gerektiren kesin algoritmaların yerini daha kısa sürelerde yaklaşık sonuçlar verebilen yapıcı ve geliştirici sezgisel metotlar veya meta-sezgisel metotlar almakta olup literatürde bu amaçla bazı sezgisel algoritmaların tasarlanması, bunların geliştirilmesi, birleştirilmesi veya kıyaslanması ile ilgili oldukça fazla çalışma bulunmaktadır. Ancak henüz herhangi bir algoritma için GSP çözümlerinin içinde en optimal güzergahı belirleyen algoritmadır denilememektedir. Bu çalışmada da literatürdeki diğer çalışmalar gibi GSP çözümlerinin en optimal olanını bulmaya yardımcı olabilmek için bazı algoritmaların kıyaslamalarının yapılması hedeflenmektedir. Bununla major anlamda seyahat sürelerinin ve güzergah boylarını düşürerek direkt olarak taşıma harcamalarının azaltılmasına katkı sağlayabileceği düşünülmektedir.

Tez çalışmasının giriş bölümünde çalışmanın tanımı ve amacı açıklanmış, ülkemizde karayolu taşımacılığı ilgili bilgiler verilmektedir. Materyal ve Yöntem bölümünde optimizasyon çalışması için kullanılacak olan ülkemizdeki organize sanayi bölgelerine ait; mesafe ve koordinat değerleri verilerek kaynakları ile beraber hesapları anlatılmıştır. Optimizasyon için GSP baz alınarak yazılan GA, lineer optimizasyon ve PSO çözüm algoritması açıklanmış ve gerçek mesafelere göre çözüm yapabilmek için algoritma üzerinde yapılan modifiyeler anlatılmıştır. Araştırma Bulguları ve Tartışma bölümünde, öncelikle örnek optimizasyon çalışması için kullanılacak noktalar arasında koordinatlara göre doğrusal çözümü yapılmıştır. Daha sonra gerçek değerlere göre optimum mesafe değerleri hesaplanmıştır. Çalışmanın son bölümü olan Sonuç ve Öneriler bölümünde ise hesaplanan optimum mesafe değerlerine göre hangi optimizasyon yönteminin daha iyi olduğu kanaatine varılmıştır.



## 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. Türkiye'de Karayolu Taşımacılığı

Cumhuriyetin ilanından 1948 yılına kadar demiryolları politikası ulaşımda önemli bir durumdayken bu durum 1948 yılında değişmiş ve karayollarında çok önemli bir dönüm noktasına gelinmiştir. Karayollarıyla ilgili çalışmaların daha verimli ve modern yöntemler ile yönlendirilmesi gerektiğinden 1950 yılında Karayolları Genel Müdürlüğü (KGM) kurulmuştur. Ülkemizde Tarihin en eski taşımacılık türünden biri olan karayolu taşımacılığı yıllar boyu istikrarını koruyan bir taşımacılık türü olmuştur. Karayolu taşımacılığı ülkemizde her zaman tercih edilen bir türdür. Bunun en önemli

sebeplerinden bir kaçı diđer taşımacılık türlerine oranla daha ucuz olması, aktarmasız taşıma olanađı sağlaması, diđer taşıma türlerine nazaran daha hızlı olması gibi etkenlerden kaynaklanmaktadır. Günümüzde Türkiye’ de yolcu taşımacılıđının %97’ si, yük taşımacılıđının ise %89’u karayolu taşımacılıđı ile gerekleşmektedir. Taşımacılık sistemlerinin altyapılarının iyileştirilmesi yanında taşıma mesafelerinin kısaltılması ve farklı taşımacılık sistemlerinde gerekli bađlantıları sađlayarak ok türlü taşımacılıđın güçlendirilmesi ile taşımacılık maliyetlerinin azaltılması lke ekonomilerine ciddi katkıları olmaktadır.

### 1.3. Gezgin Satıcı Problemi (GSP)

GSP, bir noktadan yola ıkıp eldeki bütün noktalara bir kere uğramak şartı ile bařladıđı noktaya geri dönmesi ve buna bađlı olarak en kısa turu arařtıran problemdir. GSP, tanım olarak kolay ifade edilebilen ve anlaşılması kolay bir problem olmasına rađmen, özellikle uğranacak durak sayısının fazla olduđu durumlarda lineer özümü oldukça zordur ve zor problemler sınıfındadır. GSP ilk defa 1800’lü yıllarda İrlandalı matematiki Sir William Rowan Hamilton ve İngiliz matematiki Thomas Penyngton Kirkman tarafından gündeme getirilmiř ve üzerinde alıřılmaya bařlanmıřtır. Zaten GSP’nin literatür tanımlarında “izge üzerinde, en kısa Hamilton turunun bulunması” ifadesi ile sık karřılařılır. Burada bahsi geen Hamilton Turu ise, izge üzerindeki her noktadan sadece bir kez geen ve bařladıđı noktada biten, matematiki William Rowan Hamilton’un adıyla anılan turdur.

GSP’de tüm Hamilton güzergâhlarının belirlenmesi, her güzergâhın uzunluklarının hesaplanması ve tüm güzergâhlar arasındaki en kısa mesafeli olanının bulunması sırası ile probleme esas Hamilton turlarının iindeki en düşük mesafeli turun belirlenmesi amacı güdülmektedir. Problemin sahip olduđu nokta sayıları az miktarda olduđunda GSP özümleri kesin algoritmalar kullanılarak kısa sürelerde tam sonuç verebilmekte olup; ancak nokta sayılarının fazlalıđında ise kesin sonuçları elde etmek zaman aısından imkânsızı ifade edebilmektedir. Problemin özümü  $(n - 1)! / 2$  (Simetrik özüm Bölenu) Hamilton turunun ierisinde bulunmaktadır. Bu da basit bir dilde, 10 noktalı bir GSP’de 181 440 tur, 20 noktalı bir GSP’de  $6,08226 \cdot 10^{16}$  tur ve bu alıřmada kullanılan 81 noktalı GSP’de ise  $4,07958 \cdot 10^{62}$  tur anlamına gelmektedir.

### 1.3.1. Kesin algoritmalar

GSP'de çözümlerinin içinde tam olarak en doğru çözümü sağlayan metotlar kesin yöntemler olarak isimlendirilirler. GSP'yi çözenin basit bir yolu, tüm uygulanabilir turları bulmak, objektif fonksiyon değerlerini değerlendirmek ve en iyisini seçmektir. GSP'nin orta büyüklükteki bir sorun için bile çok sayıda olası çözümü nedeniyle, fena halde verimsiz olduğu ve uygulanamayacağı açıktır (Ahmed, Z. H., 2010). Dinamik programlama, Dal ve Sınır (Branch and Bound) ve Dal ve Kesme (Branch-Cut) yöntemleri kesin algoritma uygulamalarına örnek gösterilebilirler (Pulat, M., and Kocakoç, İ. D.).

### 1.3.2. Sezgisel algoritmalar

Sezgiseller GSP'nde verimli bir şekilde çok iyi çözümler üretmeyi amaç edinmektedirler. Fakat her zaman en uygun çözümü bulamazlar ya da en azından bulunan çözümlerin optimizasyonunu garanti etmezler. Ancak sezgisel algoritmalarının kısa çözüm süreleri, uygulamalarının kolay, esnek ve basit oluşları sebeplerinden ötürü GSP'nde kullanımlarını yaygın hale getirmektedir. Sezgiseller tur oluşturan algoritmalar, turu geliştiren algoritmalar ve birleşik algoritmalar olarak üç sınıfa ayrılabilirler (Jos\_e Fernando Oliveira, Maria, 2009).

### 1.3.3. Tur oluşturan algoritmalar

GSP'de start verilen bir yerleşim yerinden başlayarak probleme ait tüm yerleşim yerlerinin güzergah kapsamına alınana kadar her defasında alt tura yeni bir yerleşim yeri katarak güzergahı tamamlamak ve güzergah tamamlandığında çözüm aramayı durduran sezgiseller olarak belirtilebilirler (Keskintürk, T., Kiremitci, B., and Kiremitci, S, 2016). En yakın komşu, Açgözlü ve Ekleme Sezgiselleri literatürde en çok yer edinen tur oluşturan algoritmalarıdır (Johnson, D. S., and McGeoch, L. A., 1997).

### 1.3.4. Turu geliştiren algoritmalar

Bu yöntemler uygulanabilir bir çözüme ulaşmaya çalışmaz, turlarının bir tanesinden başlayabilir (belki de yapıcı bir yöntemle elde edilir) ve daha sonra verilen bir koşul gerçekleştiğinde biten yinelemeli bir süreçle onu iyileştirir. 2-opt, Genetik, Benzetim

Tavlama ve Karınca Kolonisi algoritmaları bu sezgisellere örnek olarak gösterilebilir (Johnson, D. S., and McGeoch, L. A., 1997).

### 1.3.5. Melez algoritmalar

GSP'nde tur oluşturan ve tur geliştiren metotların birlikte kullanıldıkları yinelemeli ve Lin-Kernighan gibi birleşik metotlarda literatürde bulunmaktadır. Melez yöntemler diğer yöntemlere göre daha başarılı sonuçlar elde etmektedir (Johnson, D. S., 1990).

## 1.4. Lineer Optimizasyon (LO)

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, 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. 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.

### 1.5. Parçacıklı Sürü Optimizasyon (PSO)

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 itteresyonda 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 Diğeri, 1997). PSO'da Karınca Kolonisi Algoritmasındaki gibi yiyecek arayan sürülerin yiyecek ararken ki sergilediği davranışlardan esinlenerek geliştirilmiştir.

Sürü zekâsı belirli bir algoritma veya bir sistem değildir. Sürü zekâsı doğal veya yapay dağıtılmış, kendi kendine organize sistemlerin, kolektif bir davranış biçimidir. Sürü zekâsını baz alarak işleyen algoritmalara 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.

### 1.6. Genetik Algoritma

Genetik algoritma, problemlerin çözümü için evrimsel süreci bilgisayar ortamında taklit ederler. Diğer eniyileme yöntemlerinde olduğu gibi çözüm için tek bir yapının geliştirilmesi yerine, böyle yapılardan meydana gelen bir küme oluştururlar. Problem için olası pek çok çözümü temsil eden bu küme genetik algoritma terminolojisinde nüfus adını

alır. Nüfuslar, kromozom veya birey adı verilen sayı dizilerinden oluşur. Birey içindeki her bir elemana gen adı verilir. Nüfustaki bireyler evrimsel süreç içinde genetik algoritma işlemcileri tarafından belirlenirler. Genetik algoritmalar problemlere tek bir çözüm üretmek yerine farklı çözümlerden oluşan bir çözüm kümesi üretir. Böylelikle, arama uzayında aynı anda birçok nokta değerlendirilmekte ve sonuçta bütünsel çözüme ulaşma olasılığı yükselmektedir. Çözüm kümesindeki çözümler birbirinden tamamen bağımsızdır. Her biri çok boyutlu uzay üzerinde bir vektördür.

## 2. ÖNCEKİ ÇALIŞMALAR

Çolak (2010), Adana şehrinde üretim yapan bir gıda sektör firması esas alınarak gezgin satıcı problemine, bir modern optimizasyon yöntemi olan genetik algoritmalar ile sonuç bulmaya çalışılmaktadır. Şirketin kendisinin sürekli kullanmakta olduğu güzergâhlar ile algoritma sonuç güzergâhları kıyaslanarak algoritmanın değerlendirilmesi yapılmaktadır. Şirketin iki bölgesinde farklı güzergâh belirlenmesi başlangıcında genetik algoritmaların kullanılması ile oluşturulan her yeni bir güzergâhın sürekli kullanılmakta olan eski güzergâhlara oranla yaklaşık %20 daha mesafeleri azalttığı tespit edilmekte ve bu mesafe azalışında şirketin nakliye fiyatlarını düşürmeye yararlarının bulunacağı belirtilmektedir.

Rego ve ark. (2011), zorlu ve çeşitli simetrik ve asimetric TSP kıyaslama problemleri seti üzerinde yapılan hesaplama deneylerinin deneysel bir analizi ile birlikte başarılı uygulamalarından sorumlu öncü yöntemleri ve özel bileşenleri araştırılmaktadır. Bu çalışmada incelenen TSP için en etkili ve verimli yerel arama fırlatma zinciri algoritmaları, simetrik GSP'ler, sorunun asimetric versiyonu için bu yöntemlerin üç genellemesine ek olarak Lin-Kernighan (LK) yaklaşımının altı varyasyonunu ve bunun için kök ve döngü (S&C) enjeksiyon zinciri yönteminin iki varyasyonunu içermektedir. S&C yaklaşımlarının, en iyi çözümleri elde etmek için daha uzun çalışma süreleri kullanmasına rağmen, çözüm kalitesi açısından temel LK uygulamalarından açıkça üstün olduğu görülmektedir.

Cansız ve Ark. (2018) bu çalışmada, taşımacılık faaliyetlerinde taşınan yükün nakliye aşamaları ele alınmıştır. Oluşturulan güzergâhlar yük taşımacılığına uygun olarak seçilmiştir. Mesafe, maliyet ve zamana göre optimum değerler elde edilmeye çalışılmıştır. Yapılan analizler sonucunda elde edilen değerler doğrultusunda çok türlü taşımacılığın tek türlü taşımacılığa oranla daha avantajlı olduğu gözlemlenmiştir. Kıtalar arası taşımacılık dışında şehirlerarası taşımacılıkta da daha uygun olduğu gözlemlenmiştir.

Singh ve Chopra (2012), kaba kuvvet yöntemi, en yakın komşu, dal ve sınır, Dijkstra en kısa yol algoritması, Bellman Ford, Floyd Warshall algoritması ve Gezgin Satıcı Problemini çözmek için kullanılan karınca kolonisi optimizasyonu ve genetik algoritma gibi bazı sezgisel teknikler incelenmektedir. GSP'yi çözmek için kullanılan bu teknikleri inceledikten sonra geleneksel algoritmaların negatif kenar ağırları için uygun olmadıkları ve

yüksek işlemsel karmaşıklığı sergiledikleri sonuçlarına varılmaktadır. Ayrıca bunun yanında araştırmacıların en kısa yol problemini çözmek için yer ve zaman karmaşıklığı açısından daha iyi bir algoritma tasarlamada; Karınca Kolonisi Algoritması ile daha verimli, daha kısa sürede sorunu çözebilen ve performansı uygun problem parametre ayarlarına bağlı dikkate değer bir başarı elde ettikleri belirtilmektedir.

Chauhan ve ark. (2012), gezgin satıcı problemi (GSP) için yapıcı bir yöntem kullanarak lineer zamanda asimptotik olarak tatmin edici sonuçlar veren bir sezgisel yaklaşım tasarlama hedeflenmektedir. Daha önceki çalışmalar, tam grafiği GSP' ye bir girdi olarak kabul eder ancak önerilen bu yaklaşımla oluşturulan GSP çözücüsü ise tamamlanmamış grafiğin yanı sıra tam çizge ile de çalışabilmektedir. Önerilen yaklaşım, dinamik yaklaşım ile kıyaslandığında en büyük avantajı hızlı uygulanabilir bir çözüm sunabilmesidir ancak bulunduğu çözümler her durumda uygun çözümler olmayabilir.

Gülsün ve ark. (2018), kombine taşımacılığın katkıları, yapılabirliği ve negatif yönlerinin ele alındığı bu çalışmada gelişmekte olan ülkemizin temel problemlerinden başında gelen taşımacılık lojistik ile birlikte değerlendirilmektedir. Çalışmada Ülkemiz için değerlendirilen yöntem Gezgin Satıcı Problemi (GSP)'nin bir türüdür. Problem çözümlerini kesin ve yaklaşık sonuç olarak veren farklı metotlara sahip olan GSP'nin ulaşım sistemi tasarımında faydaları olduğu belirtilmektedir. Kullanıcının algoritma seçme yöntemleri genellikle çözüm süreleri ve hesaplamaların yoğunluğu sebepleri ile değişebilmektedir. Çalışma problemi Genetik Optimizasyon yöntemi kullanılarak PyGenetix programı ile analiz edilmiş ve Ülkemizde ulaşım sistemi problemlerine katkı sağlayabilecek sonuçlara ulaşılmıştır.

Cansız ve Göçmen (2018) bu çalışmada, gezgin satıcı problemine (GSP) dayalı parçacık sürü optimizasyonu (PSO) ile çok türlü bir ulaşım optimizasyonu deneyi gerçekleştirilmiştir. Çok türlü taşımacılık için ulaşım modu değişimine uygun lojistik köyler ve ticari limanlar seçilmiştir. Optimizasyon ile ülkemizde yaygın olarak kullanılan tek türlü karayolu taşımacılığı ve çok türlü taşımacılık için uzaklık parametresi çözümlenmiştir. Değerlendirme sonucunda mesafe parametresi için çok türlü taşımacılığın daha avantajlı olduğu gözlemlenmiştir.



Keskintürk ve ark. (2015), çalışmada analiz metotları ile birlikte araç rotalama problemleri (ARP) incelenmektedir. Akademik çalışmalarda geniş yer bulan farklı birçok yöntemler halinde gruplara ayrılan araç rotalama problemleri bu çalışmada farklılıklarına göre analiz edilerek kayda alınmıştır. Problem analiz metotları kesin algoritmalar, sezgisel algoritmalar ve meta sezgisel algoritmalar olarak üç farklı dalda değerlendirilmektedir. Ulusal çalışmalar içerisinde bu alanda metotlar ile alakalı gruplandırma çalışmalarına katkı sağlaması hedefi çalışma sürdürülmektedir. Çalışmanın sürecinde iki farklı sezgisel algoritma ile analizi yapılan kapasite kısıtlı araç rotalama problemi detaylı olarak ele alınmakta olup ve problemlerin hesap analizleri yapılmaktadır.

Pulat ve Karakoç (), araştırmacıların yoğun ilgisini çeken ve optimizasyon ile alakalı çalışmalarda literatürde fazlasıyla yer alan çözümü zor (NP-hard) bir problemi olan Gezgin Satıcı Problemi, tariflenmesi basit olarak belirtilmekle beraber en uygun sonucu bulmakta oldukça zorlanmaktadır. Problemlerdeki düşük şehir sayısına bağlı olarak kesin sonuçlara elde etmekte kesin algoritmalar yeterli olmakla birlikte şehir sayılarının artması ile kesin sonuçlarda fazlası ile uzaklaşmaktadır. Bu nedenlerden ötürü problem çözümleri için bize yaklaşık sonuçları veren farklı sezgisel algoritmalar geliştirilmekte olup ve GSP'nin çözümleri için bu algoritmalarından faydaniılmaktadır. Çalışmalarda yaygın olarak kullanılan ve genellikle GSP gibi zor optimizasyon problemlerinde ideal optimum değerlerini veren sezgisel algoritmalar genetik algoritmalar olmaktadır. Bu çalışmada sezgisel yöntemlerden olan genetik algoritmaların çalışma sistemini önemli derecede etkileyen faktörlerin farklı üç boyuttaki problem ve çaprazlama operatörleri kullanılarak yapılan analizlerin karşılaştırmaları yapılmaktadır.

Özsağlam ve Çunkaş (2008), tabiatta var olan hayvan veya bitki davranışları izlenerek problemler için optimum değerleri elde etmek için optimizasyon problemleri çözüm algoritmaları elde edilmektedir. Parçacık Sürü Optimizasyonu (PSO)'nda balık ve kuş gruplarının hareketleri formülize edilerek elde edilen bir sürü esaslı sezgisel optimizasyon algoritmadır. Çalışmada Parçacık Sürü Optimizasyonu, Genetik Algoritma (GA) ve Diferansiyel Evrim Algoritmasının (DEA) analiz sonuçları analiz verileri ele alınarak kıyaslanmaktadır. Ele alınan problemlerin çözüm analiz sonuçlarına göre PSO diğer optimizasyon algoritmalarından daha iyi performans ile yakınsama hızı değerleri elde ettiği görülmektedir.

Köfteci ve Gerçek (2010), bu çalışmada, bir stokastik logit yöntemi yararlananların yük taşımacılığında ulaşım tiplerini bulma modellerinin oluşturulması hedefi ile değerlendirilmektedir. Burada yük taşımacılığı lojistik ağı ile birlikte gözlenmekte olup ve lojistik sistemi maliyet kategorisinde analizi yapılmaktadır. Oluşturulan modelin kalibrasyonunda çok değişkenli istatistiksel analiz metotlarından olan lojistik regresyon analizinden faydaniılmaktadır. Çalışma kapsamında Antalya'daki demir ve karayolu üzerinden kamyonlar ile yapılan çok türlü taşımacılığının verilerinden faydaniılmıştır. En uygun optimum sonucu veren model lojistik maliyete bağlı logit tür seçim modeli belirlenmektedir. Elde bu model kullanılarak gerçekleştirilen duyarlılık analizleri verilerinde yük taşımacılığı sınıf seçimi kararlarında esas belirleyici karar verileri taşıma maliyeti ve süresi ile aktarma gecikmesi olarak gözlenmektedir. Ayrıca çalışma içerisinde belirlenen güzergâhta çok türlü taşımacılığın kullanımının yaygınlaştırılmasına hangi metotlar ile katkı sağlanacağı değerlendirilmektedir.

Hojda ve ark. (2018), bu çalışmada ulaştırma işlerini seçme ve ulaşım yolları tasarlama konularının küçük yük taşıma filosu sahipleri için ortak bir sorun olduğunu düşünölmektedir. Çalışma problemi sürücü, filo sahibi ve müşteriler gibi çeşitli paydaşlarla çok kriterli bir optimizasyon problemi olarak formüle edilmektedir. Problemin formülasyonu sürüş ve çalışma süresi düzenleme kısıtlamalarını içermekte olup ayrıca sürücü konforunu ve güvenliğini, iş yürütmenin gecikmesini, teslim alma ve teslimat üzerindeki zaman kısıtlamalarını ön planda tutmaktadır. Analize aracın seçimi karar verme sürecinin bir parçası olarak dâhil edilmektedir. Çalışma büyük ölçüde kısıtlanmış olup ve makul bir sürede tam olarak çözüme ulaşamamıştır. Burada Polonya ulaşım ağının bir alt bölümünün verileri kullanılarak gerçekleştirilen bir dizi hesaplama denemeleri ile sonucunda yinelemeli bir çözüm algoritması önerilmektedir.

Dolgoplov ve ark. (2019), çalışma demiryolu ağındaki güzergâhların rasyonelleştirilmesine kapsamakta olup ve nöro-bulanık modelleme ve rotaların oluşturulması için tasarlanmış bir genetik algoritma kullanımına dayanan bir karar destek sisteminin modelinin geliştirilmesi önermektedir. Geliştirilmiş modele dayanarak, yolcu ve yük trenleri için en uygun güzergâhların oluşturulması için otomatik bir kontrol sistemi oluşturmak mümkün olduğu belirtilmektedir. Burada demiryolu ağı kapasite kontrolünün optimizasyon matematiksel modeli de Ford-Fulkerson yöntemi ile geliştirilmektedir. Çalışmada oluşturulan model, düzenli depolama sahalarının kapasitesinin sınırlarını, tren

akışlarının büyüklüğünü (hız dâhil) ve her bölüm için treni takip etme maliyetini dikkate almaktadır. Modelin uygulanması, yolcuların ve yüklerin toplu olarak taşınması koşullarında demiryolu ağındaki tren trafiğinin daha verimli bir şekilde dağıtılmasını mümkün kılacağı düşünülmektedir.

Kiraly ve Abonyi (2011), çalışmada Araç Yönlendirme Problemi (VRP), üniform kapasiteye sahip bir araç filosu, ortak bir depo ve birkaç müşteri talebi gibi tüm taleplere hizmet eden toplam asgari güzergah maliyetine sahip güzergah setini bulabilecek karmaşık bir kombinatoriyal optimizasyon problemi olarak belirtilmektedir. Ve yine Çoklu Gezgin Satıcı Problemi (MTSP) çözümlerde birden fazla gezgin satıcının kullanılmasına izin verilen Gezgin Satıcı Probleminin (TSP) genelleştirilmesi olarak tanımlanmaktadır. MTSP tabanlı algoritmalar bazı ek kısıtlamalar dahil edilerek çeşitli VRP'lerde de kullanılabilirler. Bu çalışmada, bu problemleri çözmek için genetik algoritmaların nasıl uygulanabileceğini gözden geçirmek ve yeni yorumlanabilir bir temsile dayalı algoritma önermek hedeflenmektedir. Ayrıntılı sezgisel algoritma yöntemi ile araç kapasitesinin değiştirilmesi ve maliyet elemanlarının ve veri yapısının değişmesinin etkileri gibi dağıtım sisteminin en uygun şekilde çalışması için diğer görevlerin belirlenmesi için farklı yuvarlak tur tiplerinin belirlenmesi dikkate alınarak örneklerle gösterilmeye çalışılmaktadır.

Cansız ve Ünsalan (2019) bu çalışmada, güzergâh karşılaştırmasında, tek türlü yük taşımacılığının yanı sıra çok türlü taşımacılığın avantajları ön plana çıkarılırken, ulaşım coğrafyasının çok türlü taşımacılığa olan etkisi incelenmiştir. Hatay ile Van arasında, ulaşım coğrafyasından dolayı tek türlü yük taşımacılığı güzergâhı oluşturulmuştur. Bunun yanı sıra iki il arasında karayolu, demiryolu ve feribot taşımacılığı birleştirilerek birden fazla çok türlü taşımacılık güzergâhı oluşturulmuştur. Güzergâhlar, yakıt tüketimi, taşıma maliyeti, taşıma süresi ve CO2 salınımı parametrelerine bağlı olarak incelenmiştir. Yapılan analizler sonucunda, çok türlü taşımacılığın üç rotasının diğer rotalar içerisinde en uygun güzergâh olduğu bulunmuştur. Hatay ile Van arasında tek türlü demiryolu taşımacılığı yapılamazken, feribot taşımacılığı ile avantajlı çok türlü taşımacılık güzergâhları oluşturulacağı gözlemlenmiştir.

Cansız ve Topal (2019) bu çalışmada, çok türlü taşımacılık ağının oluşturulması üzerinde durulmuştur. Çalışmada coğrafi bilgi sistemi (CBS)'ni ele alan ArcGIS programı

kullanılarak çok türlü taşımacılık haritası oluşturulmuş ve birleştirilmiştir. Türkiye'nin kuzeyinden güneyine ve doğusundan batısına doğru en kısa güzergâh analizleri yapılmıştır. Elde edilen bulgular değerlendirilmiştir.

Lei ve ark. (2014), bu çalışmada önce çok modlu ulaştırma ve sürü zekâsı ile ilgili bazı temel kavramlar tanımlanmakta ve çok modlu ulaştırma plan kararları ve sürü zekâ yöntemleri uygulama alanları ile ilgili literatürdeki çalışmalar incelenmektedir. Çalışmada daha sonra ulaşım maliyetleri, nakliye süresi ve nakliye risklerine dayanan çok modlu bir ulaşım şeması matematiksel modeli oluşturulmakta ve ilgili parametreleri ve bunlardaki kısıtlamaları ayrıntılı olarak açıklanmış ve ağırlık katsayı optimizasyon problemlerini tek bir objektif optimizasyon ulaştırma planı karar problemine dönüştürmek için ağırlık katsayısı kullanılmaktadır. İlk kez çok modlu ulaştırma planı kararının kombinatoryal optimizasyon problemini çözmek için Parçacık Sürü Optimizasyonu Algoritması (PSO) ve Karınca Kolonisi Algoritması (ACO) birleştirilerek oluşturulan Parçacık Sürü Karınca Kolonisi Algoritması (PSACO) önerilmektedir. Oluşturulan bu algoritma PSO ve ACO'nun avantajlarını etkili bir şekilde yansıttığı görülmektedir. Analiz sonuçları PSACO algoritmasının iki algoritmanın avantaj ve sorunlarına sahip olduğunu göstermektedir. PSACO algoritması, zaman verimliliğinde ACO'dan daha iyi ve doğrulukta da PSO'dan daha iyi olduğu gözlemlenmektedir. Ayrıca bu algoritmanın çok modlu ulaştırma planı kararıyla ilgili problemi çözmek için etkili bir sezgisel algoritma olduğu kanıtlanmış ve ulaştırma karar vericileri için ekonomik, makul ve güvenli ulaştırma planı referansı sağlayabileceği öngörülmektedir.

Yıldırım ve Karcı (2018), çalışmada, hedef şehirleri tekrar etmeden ziyaret etmek ve en kısa rotayı kullanarak başlangıç kentine geri dönmeyi hedefleyen Gezgin Satıcı Problemi'nin (GSP) sıklıkla üzerinde çalışılan kombinatoryal optimizasyon alanında NP açısından zor bir problem olduğu belirtilmektedir. Yüksek karmaşıklığı nedenleri ile GSP analizlerinde genellikle meta-sezgisel yöntemler kullanılmaktadır. Çalışma için Türkiye'de 81 ilini kapsayan GSP'nin çözümünde yeni bir meta-sezgisel yaklaşım olan Yapay Atom Algoritması kullanılmaktadır. Çalışma sonunda, kullanılan bu algoritmanın performansı aynı problemi çözmek için literatürde daha önce kullanılan Genetik Algoritma, Parçacık Sürüsü Optimizasyonu ve Yapay Arı Koloni Algoritması analiz sonuçları ile kıyaslanmaktadır.

Cansız ve ark. (2019) Bu çalışmada Hatay'ın İskenderun ilçesinde üretimi yapılan filtre malzemesinin Türkiye'nin her yerine taşınması dikkate alınarak analizler yapılmıştır. Güzergâh seçiminde farklı ulaşım türlerinin karşılaştırılması yanı sıra çok türlü taşımacılık güzergâhları ile yük aktarma duraklarında doğabilecek sorunların aşılıp aşılmadığı incelenmiştir. Aynı zamanda çok türlü taşımacılıkta coğrafyadan kaynaklı zamansal kayıpların aşılıp aşılmadığı dikkate alınmıştır.

Lv ve ark. (2019), transit konsolidasyon problemi, yükün konteynırlara yüklenmesi ve boşaltılmasının eşzamanlı olarak yapıldığı kapasitif araç rotalama probleminin bir uzantısına benzetilmektedir. Bu problemde birden fazla köken ve varış noktasına sahip küçük kargo partileri, yeniden sıralandıkları ve paketlenenleri montaj noktasına taşınmalı ve daha sonra farklı varış yerlerine veya farklı müşterilere göre yerel mal kaynağı ile birlikte yeniden yüklenmelidir. Çok Modlu taşımacılığın artan önemi tüm sürecin verimliliğini dikkate almayı gerekli kılmaktadır. Transit konsolidasyon problemi araç rotalama problemi gibi çözümü zor NP problemlerinden olup ve bu çalışmada transit konsolidasyon sürecini ölçmek için bir model önerilmekte ve bu problemi çözmek için genetik algoritma kullanılmaktadır. Modelin ilgili parametrelerinin hassasiyeti bir vaka çalışması ile analiz edilmektedir.

Wang ve Meng (2017), çalışmada ağ planlamacısının, genel bir rota seçim modeli intermodal operatör modeli altında taşıyıcıların ve hub operatörlerinin toplam işletme maliyetini en aza indirmek için bir bağlantı oluşturup oluşturmayacağını veya genişletmeyeceğini belirlemesi gereken yük taşımacılığı için ayrı bir intermodal ağ tasarım problem çözülmesi hedeflenmektedir. Burada problem tıkanıklık efektleri, parçalı doğrusal maliyet fonksiyonları ve sabit nokta kısıtlaması içeren karışık tamsayı doğrusal ve dışbükey olmayan bir program olarak formüle edilmektedir. Problemin zorluğunu azaltmak, denklilikleri göstermek ve teorik sonuçlar sağlamak için bir dizi basit ve eşdeğer model geliştirilmektedir. Analizlerde çözümler için biri çözüm bulgusu ve diğeri küresel olarak en uygun çözümü üretmek için iki çözüm yöntemi sunulmaktadır. Bu iki çözüm algoritmasını test etmek ve performanslarının karşılaştırılmalarına ışık tutmak için iki farklı sayısal deneyde algoritmalar değerlendirilmektedir.

Cansız ve Ünsalan (2020) bu çalışmada ulaştırma sektöründeki CO<sub>2</sub> salınımı, ulaşım rotalarına göre karşılaştırılması yapılmıştır. Türkiye'nin İskenderun ilçesinden diğer tüm

illere 5, 10 ve 14 ton yükün taşınmasını esas almaktadır. Burada farklı yük tonajlarının minimum CO<sub>2</sub> salınımı açısından taşımacılık türü seçimini ne derece etkilediği araştırılmıştır. Yapılan analizler sonucunda elde edilen değerler ışığında çok türlü taşımacılık güzergâhının kullanılması tek türlü taşımacılık olan karayolu taşımacılığına oranla daha az CO<sub>2</sub> salınımı yaptığı gözlemlenmiştir. Yapılan analizlere göre çok türlü taşımacılık rotalarının 5 ton yük için verimli olduğu il yüzdesi %60, 10 ton yük için %63, 14 ton yük için ise %66 çıkmıştır.

Hao ve Yue (2016), konteyner taşımacılığının ve lojistik ekonomisinin hızlı patlamasıyla birlikte, çok modlu taşımacılık ekonomik olarak giderek daha önemli hale gelmiştir. Çalışmada konteyner multimodal taşımacılığı, genel kargo taşımacılığını optimize etmeyi amaçlayan bir kombine taşımacılık modeli olarak tanımlanmaktadır. Konteyner multimodal taşıma sisteminde çeşitli taşıma modlarının optimum organizasyonu, kargo taşımacılığının zamanını, maliyetini ve kalitesini doğrudan ilgilendirmektedir. Optimum organizasyon problemini tanımlamak için, dinamik programlamaya dayanan bir optimizasyon modeli bu makalede sunulmakta olup ve gerçeklik kısıtlamalarından memnun kalınmıştır. Daha sonra taşıma modlarının optimal kombinasyon stratejisini elde etmek için dinamik bir programlama algoritması önerilmektedir. Son olarak önerilen modelin fizibilitesini ve etkinliğini göstermek için ampirik bir çalışma kullanılmaktadır.

Cansız ve Gündoğdu (2020) Bu çalışmada, taşımacılıkta maliyetlerin düşürülmesi amacı ile ülkemizde bulunan OSB'ler arasında karayolu taşımacılığı ile optimum mesafe değeri hesaplanmıştır. Yapılan çalışmada Gezgin Satıcı Problemi baz alınarak lineer optimizasyon ve genetik algoritma yöntemleri kullanılmıştır. Analizler sonucunda elde edilen değerler birbirleri ile kıyaslandığında, lineer optimizasyonun genel anlamda daha iyi sonuç verdiği gözlemlenmiştir.

Pu ve ark. (2019), demiryolu güzergahlarının optimize edilmesi oldukça karmaşık ve zaman alıcı bir mühendislik problemi olarak belirtilmektedir. Burada büyük sürekli arama alanı, karmaşık kısıtlamalar, örtük nesnel işlev ve bu sorunun sınırsız potansiyel alternatifleri birçok zorluğu beraberinde getirmektedir. Özellikle dağlık bölgelerde, son derece karmaşık arazi ve kısıtlamalar için optimal bir güzergah bulmak, mevcut yöntemlerin çoğu ile tatmin edici bir şekilde çözülemeyen zorlu bir sorundur. Bu çalışmada, dağlık bölgelerde demiryolu güzergâh optimizasyonu için kademeli ve hibrit

parçacık sürü-genetik algoritması geliştirilmeye çalışılmaktadır. Demiryolu güzergâh tasarımına uygun sürekli bir arama yöntemidir. Üç boyutlu demiryolu güzergah optimizasyon problemini çözmek için yatay ve dikey hizalamaları iki tür parçacık olarak tanımlayan kademeli bir yatay dikey integral yaklaşımı önerilmektedir. Başlangıçtaki çeşitliliği ve momentumu arttırmak için kelebek şeklindeki alanlar, yatay parçacıkları başlatmak için iki yönlü bir mesafe dönüşümü ile oluşturulan bir yol üzerinde önceden ayarlanmaktadır. Çözümleri için, rulet tekerleği seçimi, dört çapraz geçiş ve iki mutasyon dahil spesifik genetik operatörler, parametreye bağlı performansı ele almak ve erken yakınsamayı önlemek için kademeli partikül sürüsü yöntemine entegre edilmektedir. Ek olarak, sorunlu alanı derinlemesine araştırmak için bir kübik polinom ağırlık güncelleme stratejisi kullanılmaktadır. Bu sentez yöntemi, çok dağlık bir bölgedeki gerçek dünyadaki bir duruma uygulanmaktadır. Ayrıntılı veri analizleri ile deneyimli tasarımcılar tarafından tasarlanan alternatiflere ve genetik algoritma veya aşamalı olmayan parçacık sürüsü algoritması ile üretilen alternatiflere kıyasla daha umut verici çözümler sunabileceği görülmektedir.

Kuzu ve ark. (2014), gezgin satıcı problemine çözüm bulma amaçlarında yaygın olarak kullanımda olan 8 farklı meta-sezgisel metodun açıklamaları yapıldıktan sonra, bu metotlar ile literatür içerisinde yer almış olan ve birbirine benzer olmayan boyutlardaki problemler üzerinde çalışılmaktadır. Çalışma sonuçlarına göre, çözümleri içerisinde sonuç arayan modern meta-sezgisel optimizasyon algoritmaları özellikle karınca kolonisi algoritması, bir çözüm ile sonuç arayan algoritmalarından daha iyi sonuç verdiği belirtilmektedir.

Cansız ve ark. (2019) bu çalışmada gezgin satıcı problemi baz alınarak Türkiye'deki 81 ilin birbirleri arasındaki karayolu mesafeleri kullanılarak optimum mesafe hesaplanmaya çalışılmıştır. Çalışmada yedi farklı meta sezgisel algoritma kullanılmıştır. Analizler için SPSS programının gezgin satıcı problemi uzantısı kullanılmıştır. Analizlerde kullanılan sezgisel algoritma yöntemleri; en yakın komşu yöntemi, en yakın ekleme yöntemi, en ucuz ekleme yöntemi, tekrarlayan en yakın komşu yöntemi, keyfi ekleme yöntemi, 2-opt değişim yöntemi ve en uzak ekleme yöntemidir. Analizler sonucunda elde edilen değerlere göre en iyi optimum mesafe değerini veren algoritma en uzak ekleme yöntemi olmuştur.

Dikmen ve ark. (2014), bir güzergâh belirleme problemi olan gezgin satıcı probleminin çözümünde literatürde kullanılmakta olan optimizasyon yöntemlerinden karınca kolonisi algoritması ve genetik algoritmaların sonuçları kıyaslanmaktadır. Türkiye'nin seksen bir ili esas alınarak yapılacak analizler ile en iyi güzergâhın belirlenmesi amaç edinilmektedir. Hem Karınca Kolonisi algoritması hem de genetik algoritmasının güzergâh uzunluğu bakımından değerlendirme düzeyi ve bu güzergâhı hesaplama zamanı araştırılmaktadır. Çalışma içerisindeki analiz sonuçları karınca kolonisi algoritmasının hem güzergâh uzunluğu hem de hesaplama zamanı bakımından genetik algoritmalara göre daha başarılı olduklarının görülmektedir.

Cansız ve Ünsalan (2020) bu çalışmada, konteyner yük taşımacılığı için güzergâh karşılaştırılması yapılmıştır. Tek türlü ve çok türlü taşımacılık modu incelemesi ile birlikte farklı güzergâh seçim kriterleri ele alınmıştır. Bu kriterler yakıt tüketimi, maliyet, taşıma süresi ve karbondioksit salınımıdır. Rotaları değerlendirmek için yükün Hatay'dan Tekirdağ'a taşındığı kabul edilmiştir. Bu iki nokta arasında karayolu, demiryolu ve denizyolu olmak üzere üç tek türlü ve bu taşımacılık türlerinin kombinasyonlarından oluşan dört farklı çok türlü taşımacılık rotası oluşturulmuştur. Geliştirilen rotaların genelleştirilmiş maliyetleri incelendiğinde %94,35-%72,43 performans aralığı ile optimum güzergâh demiryolu çıkmaktadır. En verimsiz güzergâh ise %59,27-%33,01 performans aralığı ile karayolu taşımacılık rotası olmuştur. Multimodal taşımacılık rotalarında tür değişim noktalarında yaşanan mali ve zamansal kayıplardan dolayı verimliliğin düştüğü gözlemlenmiştir.

Nuriyeva ve Kızılateş (2016), gezgin Satıcı Problem çözüm yöntemlerine yeni bir sezgisel algoritma sunulmaktadır. Çalışma başlangıcında bir merkez yerleşim yeri ve 4 sınır yerleşim yeri N sayıda şehir içerisinde seçilip, sonra ise merkez yerleşim yeri ile ikişer-ikişer uç yerleşim yerlerinin orta noktaları belirlenerek merkez yerleşim yerinden başlanarak bu 9 yerleşim yerinden geçen bir rota tasarlanmaktadır. En kısa yol ve ekleme sezgiseli metotları ile tasarlanan bu güzergâh tüm yerleşim yerlerine uğrayacak biçimde genişletilmektedir. Bu çalışmada tasarlanan MKHA1 ve MKHA2 algoritmalarının kütüphane problemlerinin çözüm analizleri yapılmakta ve sonuçları en yakın komşu algoritması sonuçları ile kıyaslandığında bu tasarlanan algoritmaların en kısa yol algoritmalarına nazaran daha verimli olduğu belirtilmektedir.



Demirtaş ve Zengin (2016), gezgin Satıcı Problemlerinde çözümlerinde yapay zeka optimizasyon yöntemlerinden içerisinde bulunan Guduk Kuşu Optimizasyon Algoritması (GOA)'nın kullanılabilirliğinin üzerinde durulmaktadır. Ramin Rajabioun'un tasarladığı ve 2011 yılındaki çalışması ile tasarımı literatüre eklenmiş yeni bir sezgisel algoritma yöntemidir. Çalışma içerisinde NET yazılım program kullanılmakta olup ve tespit edilen güzergâhların simülasyonları yapılmaktadır. GSP'nin güzergâh uzunluğu bakımından başarımlar değerlendirilmesi ve çözüm zamanları incelenmektedir. Çalışma bulgularından yola çıkarak GOA'nın hem güzergâh uzunluğu hem de çözüm zamanı bakımından genetik algoritmalara nazaran daha iyi sonuçlar verdiği görülmektedir.

Yıldırım ve ark. (2016), yer altında yaşayan ve gözleri görmeyen farelerin yer altında kullanmış oldukları geçitlerdeki setleri gözleri görmeden geçebilme yeteneklerinde faz alınarak, gezgin satıcı problemini çözümü için modern sezgisel optimizasyon yöntemlerine benzer yeni bir algoritma oluşturulma çabasında bulunmaktadır Çalışma ile tasarlanan metot Kör Fare Algoritması olarak adlandırılmaktadır. Bu yeni modern sezgisel optimizasyon metoduyla farklı boyutlardaki simetrik test veri setleri için deneyler yapılmış ve sonuçları bilinen en iyi sonuçlar ile kıyaslanmaktadır. Az şehirli problemlerde bu meta-sezgisel algoritma literatürdeki kullanılan diğer algoritmaların sonuçları ile denkleşebilecek seviyede sonuçlar vermesinin yanında, çalışmanın ilk deney sonuç değerlerinin kayda değer olabildiği belirtilmektedir.

Cansız ve Ünsalan (2018) bu çalışmada başlangıç noktası Iğdır bitiş noktası Çanakkale olan tek türlü ve çok türlü güzergâhlar belirlenmiştir. Çok türlü taşımacılıkta güzergâhlar, lojistik köy alanları ve limanlar değişim noktası olarak kullanılıp belirlenmiştir. Güzergâhlar belirlenirken yakıt tüketimi, zaman, maliyet ve emisyon değerleri hesaplanmıştır. Hesaplar yapılırken yükün konteyner ile taşındığı, tonajının 26 ton ve kıymet değerinin 100.000 TL olduğu kabul edilmiştir. Yakıt tüketimi konteyner başı yakıt tüketimi olarak ele alınmıştır. Maliyet değeri için, yakıt tüketimi, personel gideri, liman masrafları, yükleme boşaltma masrafları, yük tonajı ve yükün kıymet değeri ele alınmıştır. Zaman hesabı için, seyahat süresi, sürücü dinlenme süresi ve yükleme boşaltma süreleri dikkate alınmıştır. Emisyon hesabı için ise yük tonajı, mesafe, ulaşım aracına göre belirlenen emisyon faktörü dikkate alınmıştır.

Ertuğrul ve Özçil (2016), herhangi bir siyasi partinin 21 yerleşim yerinde yapacağı mitinglerin programlarının planlaması amacı ile Gezgin Satıcı Problemi kullanılmakta ve analizler Genetik Algoritma kullanılarak yapılmaktadır. Çalışmada değerlendirilen partinin parti programında miting program güzergâhı ile genetik algoritma ile analizli Gezgin Satıcı Problemi model analizi sonuçları karşılaştırılmakta ve değerler arasında yaklaşık yarı yarıya fark olduğu görülmektedir. Bu farklar doğrultusunda çalışma sonunda siyasi parti miting yerleşim yerlerine ulaşım güzergâh belirlenmesinin finansal ve zaman kategorilerinde değerlendirilmeleri yapılmaktadır.

Saud ve ark. (2018), 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.

Cansız ve Ünsalan (2018) bu çalışmada, çok türlü taşımacılık için İskenderun baz alınarak incelenmiştir. Nakliye için İskenderun'da üretimi yapılan filtre malzemesi seçilmiştir. Ele alınan taşımacılık güzergâhlarında mesafeler göz önüne alınarak ton başı birim fiyat hesaplaması yapılmıştır. Elde edilen değerlere göre aynı çıkış ve varış noktası için taşımacılık güzergâhları arasında kar-zarar değerleri incelenmiştir. Yapılan güzergâhlar; İskenderun'dan 71 adet ile tek türlü karayolu taşımacılığı, 39 adet ile tek türlü demiryolu taşımacılığı ve 9 adet ile tek türlü denizyolu taşımacılığı yapılmıştır. Tek türlü taşımacılık ile oluşturulan güzergâhlar çok türlü ile kıyaslandığında karayolları ve denizyollarında yük oranda çok türlü taşımacılık daha elverişli çıkmıştır. Ancak demiryolu ile oluşturulan güzergâhlarda çok türlü taşımacılık elverişli çıkmadığı gözlemlenmiştir.

Doğan (2014), evsel ilaç atıklarının toplanması projesinde atık ilaçların taşınması faaliyetine yönelik genetik algoritmanın kullanılması araştırılmıştır. Araştırmanın kapsamında hesaplama işlemlerinin yapılabilmesi için bir yazılım geliştirilmiştir. Yazılımda test edilmek üzere üç farklı taşıma modeli oluşturulmuştur. Bu üç model, gerçek veriler ve çeşitli varsayımlar ışığında karşılaştırılmış, test sonuçları sunulmuştur. Genetik algoritmaların etkin kullanımına yönelik olarak yapılan bu araştırmanın ileriye dönük bir rehber niteliğinde olması amaçlanmaktadır.

Şahin (2019), GSP'nin çözümü için Akışkan Genetik Algoritma, En Yakın Komşu ve 2-Opt sezgiselleri üzerine kurulu melez bir yöntem sunulmaktadır. Önerilen yöntemin performansı literatürde bulunan En Yakın Komşu, Genetik Algoritma, Tabu Arama, Karınca Kolonisi Optimizasyonu ve Ağaç Fizyolojisi Optimizasyon algoritmaları kullanılarak elde edilen çözüm değerleri ile kıyaslanmıştır. Önerilen yöntemin sonuçları çözüm süresi ve kalitesi bakımından üstünlük göstermektedir.

Öznur (2003), Genetik algoritmanın nasıl çalıştığı ve yöneylem araştırması problemleri arasında yer alan gezgin satıcı probleminin genetik algoritma ile çözümü üzerinde durulmuş ve bunun için geliştirilen bir java program ile çözümü tanıtılmıştır. Ayrıca genetik algoritma çözümü ve klasik yöntemlerle çözümleri karşılaştırılmaktadır.

Cansız ve Bozkurt (2019) bu çalışmada Türkiye'de bulunan 18 şehirdeki lojistik köyler için Lineer optimizasyon ve PSO yöntemleri kullanılarak bu lojistik köyler arasındaki optimum mesafe, süre ve birim maliyet değerleri hesaplanmıştır. Yapılan bu çalışmada 18 lojistik köy içinden 3 ve 4 adet seçilerek ayrı güzergâh analizi yapılmıştır. Yapılan analizler sonucunda elde edilen değerler ışığında lineer optimizasyon ve PSO yöntemleri birbirleri ile kıyaslandığında; 18 şehir arasındaki mesafe değeri Lineer Optimizasyonda daha iyi çıktığı, 3 ve 4 şehir için elde edilen değerlerin ise aynı olduğu gözlemlenmiştir.

Özkan (2003), tek model deterministik montaj hatları problemlerinin genel tanımları yapıldıktan sonra geleneksel metotların etkin bir şekilde çözmediği problemlere uygulanan GA yaklaşımı araştırılmıştır. Genetik algoritmanın ana prensipleri ve montaj hatları dengeleme problemlerine uygulanması süreci açıklanmıştır. Sonraki aşamada konuyla ilgili olarak 1994-2002 yılları arasındaki literatür çalışmaları incelenmiş olup, tez çalışması iyi bilinen bir klima markası olan Arçelik-LG Gebze İşletmesi'ndeki dış ünite montaj hattı

üzerinde yapılan bir uygulama çalışmasıyla pekiştirilmiştir. Elde edilen sonuçlar bize GA'nın etkin bir metot olduğunu özellikle büyük ölçekli montaj hattı dengeleme problemleri çözümünde gelecek vaat ettiğini göstermiştir.

Dikmen (2014), rota planlama problemlerinden olan gezgin satıcı probleminin (GSP) çözümünü gerçekleştirmek için yapay zekâ tekniklerinden olan karınca kolonisi ve genetik algoritmaların performansları karşılaştırılmıştır. Türkiye haritası üzerinde gerçekleştirilen çalışmada en iyi rotanın planlanması hedeflenmiştir. Her iki algoritmanın rota mesafesi yönünden başarımları ve bu rotayı hesaplama süresi incelenmiştir. Uygulamanın gerçekleştirilmesi ve deneysel sonuçların gözlemlenmesi için C# tabanlı bir arayüz tasarlanmıştır. Uygulama sonucunda karınca kolonisi algoritmasının hem rota mesafesi hem de başarımları yönünden genetik algoritmalara göre daha üstün olduğu gözlemlenmiştir.

### 3. MATERYAL VE YÖNTEM

Bu tez çalışmasında yöntem olarak öncelikle GSP'nin Öklid Bağıntısı ile koordinatlara bağlı olarak doğrusal çözümü için lineer optimizasyon, GA ve PSO ile bir çözüm algoritması yazılmıştır. Daha sonra yazılan algoritmaların gerçek değerlere göre çözüm yapabilmesi için koordinatlar yerine gerçek km değerleri girdi olarak eklenmiştir. Analizler bu girdi değerlerine göre işlem yapılmış, güzergâh çizimini ise koordinatlara göre yapabilen modifiyeli algoritmalar tasarlanmıştır. Materyaller ise güzergâh üzerinde bulunan noktalara ait koordinatlar Google GPS den, karayolu mesafeleri ise Google Earth'ten alınmıştır.

Öncelikle GSP'nin genel çözüm mantığını göstermek amacıyla hesap noktaları üzerinde kuş uçuşu doğrusal çözümlenmiştir. Daha sonra hesap güzergâhında yaygın kullanılan bir taşımacılık türü olan karayolu taşımacılığı için optimum mesafe değeri elde edilen veriler üzerinden çözüm yapılmıştır.

#### 3.1. Çalışmanın Amacı

Yapılan çalışmadaki amaç: bir noktadan yola çıkıp bilinen belirli noktalara uğrayıp yük taşımacılığı yapıldıktan sonra başlangıç noktasına dönen yükün organize sanayi bölgelerine karayolu taşımacılığı ile yapılmasını sağlamaktır. Bunun için GSP mantığı baz alınarak optimum mesafeyi parametrelere göre hesaplamaktır.

OSB' ye ait koordinatlar ve gerçek km değerleri Çizelge 3.1 ve Çizelge 3.2' de verilmektedir.

Çizelge 3.1. OSB'ye ait koordinatlar (Google Earth)

SIRA	ORGANİZE SANAYİ BÖLGELERİ	KOORDİNATLAR	
		X	Y
1	Adana Kozan Organize Sanayi Bölgesi	37,4614	35,7656
2	Adıyaman Organize Sanayi Bölgesi	37,7722	38,2134
3	Afyonkarahisar Organize Sanayi Bölgesi	38,7960	30,5594
4	Ağrı Organize Sanayi Bölgesi	39,7202	43,0326
5	Aksaray Organize Sanayi Bölgesi	38,2796	34,0399
6	Amasya Organize Sanayi Bölgesi	40,5837	35,6437
7	Ankara Sanayi Odası I. Organize Sanayi Bölgesi	39,9853	32,5576
8	Antakya Organize Sanayi Bölgesi	36,4382	36,2661
9	Antalya Organize Sanayi Bölgesi	37,0814	30,6054
10	Ardahan Organize Sanayi Bölgesi	41,1112	42,7049
11	Artvin Arhavi Organize Sanayi Bölgesi	41,3314	41,3003
12	Aydın Organize Sanayi Bölgesi	37,8428	27,8905
13	Balıkesir Organize Sanayi Bölgesi	39,5940	27,8436
14	Bartın Merkez I. Organize Sanayi Bölgesi	41,6401	32,3285
15	Batman Organize Sanayi Bölgesi	37,9029	41,1336
16	Bayburt Organize Sanayi Bölgesi	40,2892	40,1647
17	Bilecik I. Organize Sanayi Bölgesi	40,1438	29,9748
18	Bingöl Organize Sanayi Bölgesi	38,8103	40,5278
19	Bitlis Organize Sanayi Bölgesi	38,4342	42,1387
20	Bolu Karma Ve Tekstil İhtisas Organize Sanayi Bölgesi	40,7381	31,7107
21	Burdur II. Organize Sanayi Bölgesi	37,8025	30,3876
22	Bursa Organize Sanayi Bölgesi	40,2448	28,9642
23	Çanakkale Organize Sanayi Bölgesi	40,1570	26,4643
24	Çankırı Korgun Organize Sanayi Bölgesi	40,7292	33,5163
25	Çorum Organize Sanayi Bölgesi	40,4785	34,8570
26	Denizli Organize Sanayi Bölgesi	37,8039	29,2412
27	Diyarbakır Organize Sanayi Bölgesi	38,0791	40,0686
28	Düzce Organize Sanayi Bölgesi	40,8417	31,1770
29	Edirne Organize Sanayi Bölgesi	41,8253	26,8325
30	Elazığ Organize Sanayi Bölgesi	38,6205	39,3002
31	Erzincan Organize Sanayi Bölgesi	39,7533	39,3981
32	Erzurum Merkez I. Organize Sanayi Bölgesi	39,9349	41,1861
33	Eskişehir Sanayi Odası Organize Sanayi Bölgesi	39,7439	30,6221
34	Gaziantep Organize Sanayi Bölgesi	37,1492	37,2955
35	Giresun Organize Sanayi Bölgesi	40,8930	38,4199
36	Gümüşhane Organize Sanayi Bölgesi	40,4329	39,5183
37	Hakkâri Organize Sanayi Bölgesi	37,5743	43,7375
38	İğdır Organize Sanayi Bölgesi	39,9342	44,0496
39	İsparta Süleyman Demirel Organize Sanayi Bölgesi	37,8881	30,4100
40	İstanbul Anadolu Yakası Organize Sanayi Bölgesi	40,8758	29,3418
41	İzmir Atatürk Organize Sanayi Bölgesi	38,4919	27,0379
42	Kahramanmaraş Organize Sanayi Bölgesi	37,6011	36,7725
43	Karabük Organize Sanayi Bölgesi	41,1376	32,6430
44	Karaman Organize Sanayi Bölgesi	37,2218	33,3080
45	Kars Organize Sanayi Bölgesi	40,6036	43,0973
46	Kastamonu Organize Sanayi Bölgesi	41,5220	33,7761
47	Kayseri Organize Sanayi Bölgesi	38,7293	35,3858
48	Kilis Organize Sanayi Bölgesi	39,8140	33,4884
49	Kırıkkale Silah Sanayi İhtisas Organize Sanayi Bölgesi	41,6921	27,3364
50	Kırklareli Organize Sanayi Bölgesi	39,0785	34,1919
51	Kırşehir Organize Sanayi Bölgesi	36,7205	37,2176

Çizelge 3.1. (Devam) OSB'ye ait koordinatlar (Google Earth)

SIRA	ORGANİZE SANAYİ BÖLGELERİ	KOORDİNATLAR	
		X	Y
52	Kocaeli Gebze Kömürçüler İhtisas Organize Sanayi Bölgesi	40,7865	29,5281
53	Konya Organize Sanayi Bölgesi	37,9739	32,6106
54	Kütahya Organize Sanayi Bölgesi	39,3918	30,1178
55	Malatya Merkez I. Organize Sanayi Bölgesi	38,3408	38,2014
56	Manisa Organize Sanayi Bölgesi	38,6158	27,3654
57	Mardin Organize Sanayi Bölgesi	37,2831	40,6687
58	Mersin Tarsus Organize Sanayi Bölgesi	36,8997	34,7967
59	Muğla Milas Organize Sanayi Bölgesi	37,3031	27,7831
60	Muş Organize Sanayi Bölgesi	38,7471	41,5004
61	Nevşehir Acıgöl Organize Sanayi Bölgesi	38,5545	34,5172
62	Niğde Organize Sanayi Bölgesi	37,9477	34,6164
63	Ordu Organize Sanayi Bölgesi	40,9724	37,9314
64	Osmaniye Organize Sanayi Bölgesi	37,0087	36,1106
65	Rize Organize Sanayi Bölgesi	40,9175	40,4170
66	Sakarya I. Organize Sanayi Bölgesi	40,7218	30,4217
67	Samsun Merkez Organize Sanayi Bölgesi	41,2379	36,4336
68	Siirt Organize Sanayi Bölgesi	37,9248	41,9447
69	Sinop Organize Sanayi Bölgesi	42,0259	35,1516
70	Sivas Merkez I. Organize Sanayi Bölgesi	39,7717	37,0429
71	Şanlıurfa Organize Sanayi Bölgesi	37,1086	38,6500
72	Şırnak Cizre Organize Sanayi Bölgesi	37,4613	42,2977
73	Tekirdağ Organize Sanayi Bölgesi	41,1972	27,5348
74	Tokat Merkez Organize Sanayi Bölgesi	40,3295	36,5172
75	Trabzon Arsin Organize Sanayi Bölgesi	40,9588	39,9617
76	Tunceli Organize Sanayi Bölgesi	39,0755	39,5347
77	Uşak Organize Sanayi Bölgesi	38,6772	29,2201
78	Van Organize Sanayi Bölgesi	38,5765	43,3167
79	Yalova Gemi İhtisas Organize Sanayi Bölgesi	40,6562	29,3512
80	Yozgat Kaleseramik Özel Organize Sanayi Bölgesi	39,6231	34,4856
81	Zonguldak Ereğli Organize Sanayi Bölgesi	41,2824	31,4367















### 3.2. Dört OSB'ye Ait Lineer Optimizasyon Algoritmasının Oluşturulması

Tez çalışmasında kullanılan 81 adet OSB arasından seçilen 4 OSB için algoritmalar oluşturulmuştur. Seçilen dört OSB; Adana, Adıyaman, Afyonkarahisar ve Ağrı'dır. Bu uygulamanın amacı oluşturulan algoritmaların çalışma prensiplerine değinmektir. Çizelge 3.3' teki kodlama şehirlerarasındaki mesafeleri matris formatında kaydetmemizi sağlayan başlangıç algoritmasıdır. Çizelge 3.3' teki kodlama sonucu elde edilen mesafe matrisi Çizelge 3.4' te verilmektedir.

Çizelge 3.3. Şehirlerarasındaki mesafeleri alıp matris haline getiren kodlama

```
D=[0 325 646 950;...
 325 0 897 650;...
 646 897 0 1318;...
 950 650 1318 0];

load('mesafe_tr_koor_4sehir.mat')
MesafeMatrisi=D
```

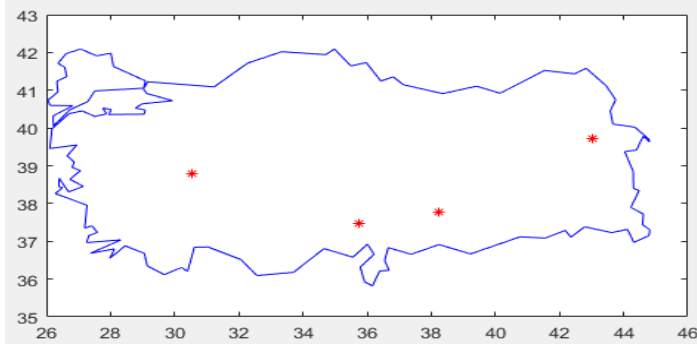
Çizelge 3.4. Şehirlerarasındaki mesafeleri matris haline getiren kodlamanın sonucu

<b>Mesafe Matrisi</b>				
	1. şehir	2. şehir	3. şehir	4. Şehir
1. şehir	0	325	646	950
2. şehir	325	0	897	650
3. şehir	646	897	0	1318
4. Şehir	950	650	1318	0

Aşağıdaki belirtilen kodlama Türkiye haritasını ve kullanılan 4 OSB'yi çizdirmek için gereken kodlamadır. Kodlama Çizelge 3.5'te verilmektedir. Kodlama sonucu elde edilen çıktı Şekil 3.1'de verilmektedir.

Çizelge 3.5. OSB'lerin Türkiye haritası üzerinde gösteren kodlama

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



Şekil 3.1. OSB'lerin Türkiye haritası üzerindeki görünümü

Oluşturulan algoritmada bir sonraki adım şehir çiftlerini oluşturmaktır. Bunun amacı şehirleri birbirleri ile eşleştirip, mesafe vektörünü oluşturularak aralarındaki mesafelerin belirlenmesidir. Şehir çiftlerini oluşturan algoritma Çizelge 3.6' da verilmektedir. Çizelge 3.6' da bulunan algoritma sonucunda elde edilen değerler Çizelge 3.7' de verilmektedir.

Çizelge 3.6. 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;
    MesafeVektoru(count:SehirSayisi*c, 1) = MesafeMatrisi(c,:);
    c=c+1;
end
UzaklikMesafeVektoru=length(MesafeVektoru);

```

Çizelge 3.7. Şehir çiftlerini ve mesafe vektörünü oluşturan kodlamanın sonucu

Şehir Çifti		Mesafe vektörü	Uzaklık mesafe vektörü
1	1	0	9
1	2	325	
1	3	646	
1	4	950	
2	1	325	
2	2	0	
2	3	897	
2	5	650	
3	1	646	
3	2	897	
3	3	0	
3	4	1318	
4	1	950	
4	2	650	
4	3	1318	
4	4	0	

Algoritmada bir sonraki aşama kısıtlamaları oluşturan algoritmaları 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. Eşitlik kısıtlamaları bir şehre yapılacak seyahat sayısını ve o şehirden yapılacak seyahat sayısının bir defa olmasını sağlayan algoritmadır. Eşitlik kısıtları için oluşturulan algoritma Çizelge 3.8 ve Çizelge 3.10' da verilmektedir. Algoritma sonucu elde edilen değerleri Çizelge 3.9 ve Çizelge.11' de verilmektedir.

Çizelge 3.8. Eşitlik kısıtları kodlamaları

```
fprintf('Eşitlik kısıtlamaları yaratılıyor \n');
Aeq=spones(1:length(SehirCifti));
beq=SehirSayisi;
Aeq=[Aeq;spalloc(2*SehirSayisi,length(SehirCifti),2*SehirSayisi*(SehirSayisi+SehirSayisi-1))];
```

Çizelge 3.9. Eşitlik kısıtları kodlaması sonuçları

Aeq 1. hal														b	Aeq 2. hal																				
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Çizelge 3.10. Eşitlik kısıtları kodlaması

```
c=1;
for count = 1:2:((2*SehirSayisi)-1);
    SutunToplami=sparse(SehirCifti(:,2)==c);
    Aeq(count+1,:)=SutunToplami';
    SatirToplami=SehirCifti(:,1)==c;
    Aeq(count+2,:)=SatirToplami';
    c=c+1;
end
beq=[beq;ones(2*SehirSayisi,1)];
```





Çizelge 3.11. (Devam) Eşitlik kısıtları kodlaması sonuçları

C=3 için Sütun Top.	Aeq=count+1																Satır top.	Aeq=count+2																									
	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1		1	1	1	1					
0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0						
1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						
0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0						
0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0						
0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0						
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0							
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						
0																		1																									
1																		1																									
0																		1																									
0																		0																									
1																		0																									
0																		0																									
C=4 için Sütun Top.	Aeq=count+1																Satır top.	Aeq=count+2																beğ 2.									
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	4	
0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	
0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1		
1	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	1	0	1		
0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1			
0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	1	0			
0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	1				
1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	1	1			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	
0																		0																									
0																		0																									
1																		0																									
0																		1																									
0																		1																									
0																		1																									
1																		1																									

Algoritmanın bir sonraki aşamasında kısıtlar sonucunda elde edilen olmayan yolların mesafe değerlerini sıfır almak için kodlama yapılmaktadır. Bu kodlama Çizelge 3.12' de verilmektedir. Kodlama sonucu elde edilen değerler Çizelge 3.13' te verilmektedir.

Çizelge 3.12. Olmayan yolların belirlenmesinde kullanılan kodlama

```

OlmayanYollar=sparse(MesafeVektoru==0);
Aeq(2*c,:)=OlmayanYollar';
beq=[beq;0];

```

Çizelge 3.13. Olmayan yolların belirlenmesinde kullanılan kodlama sonucu

Olmayan yollar	Aeq son hal																beq son
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	4
0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1
0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	1
0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	1
1	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	1
0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	1
0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0
1																	
0																	
0																	
0																	
0																	
1																	

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 3.14' te verilmektedir. Kodlama sonucu elde edilen değerler Çizelge 3.15' te verilmektedir.

Çizelge 3.14. Alt üst sınırlamaların belirtildiği kodlar

```

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

```

Çizelge 3.15. 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
10	0	1
11	0	1
12	0	1
13	0	1
14	0	1
15	0	1
16	0	1

Algoritmanın bir sonraki aşamasında lineer optimizasyon kodlaması devreye girmektedir. Bu kodlama sayesinde oluşacak optimal tur belirlenmektedir. Oluşturulan kısıtlamalar sonucu optimal güzergâhı belirleyen kodlamalar Çizelge 3.16’ da verilmektedir. Kodlama sonucu elde edilen optimal turun başlangıç noktası ve bitiş noktası ve toplam km değeri Çizelge 3.17’ de verilmektedir.

Çizelge 3.16. Dört OSB için güzergâhın belirlenmesinde kullanılan kodlama

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

Çizelge 3.17. Dört OSB için güzergâhın belirlenmesinde kullanılan kodlama sonucu

Karar değişkenleri	Optimum mesafe	Çıkış noktası	Çıktı
0	2592	1	relativegap: 0
1			absolutegap: 0
0			numfeaspoints: 1
0			numnodes: 0
1			constrviolation: 0
0			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).
0			
0			
0			
0			
0			
1			
0			
0			
1			
0			
0			

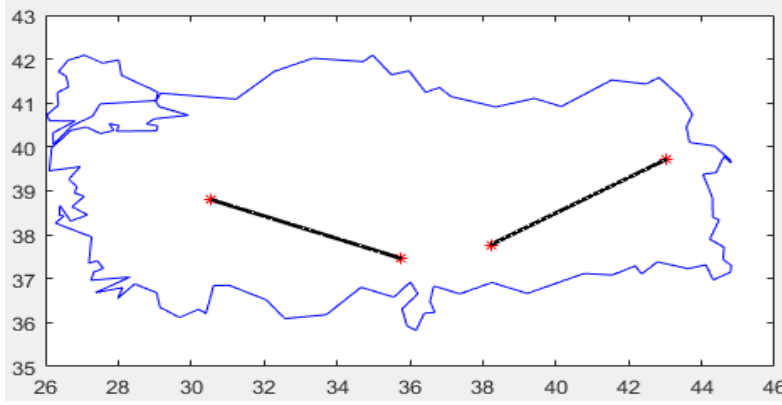
Algoritmanın bir sonraki aşamasında oluşan turun içinde alt turların olup olmadığını kontrol etmek için gereken kodlamalar yapılmaktadır. GSP örneğinin bir diğer kısıtı da alt turların oluşmasını engellemektir. Alt turlar birbirine yakın noktaların kendi aralarında tur oluşturmasına denir. Bunu engellemek için gereken kodlama Çizelge 3.18’de verilmektedir. Kodlama sonucu elde edilen alt tur sayısı ve başlangıç ve bitiş noktaları değerleri Çizelge 3.19’ da verilmektedir. Türkiye haritası üzerinde gösterimi Şekil 3.2’de verilmektedir.

Çizelge 3.18. Alttur turların belirlenmesinde kullanılan kodlama

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

Çizelge 3.19. Alttur turların belirlenmesinde kullanılan kodlama sonucu

Turlar	Alt tur sayısı
[3 1]	2
[4 2]	



Şekil 3.2. Oluşan alt turların harita üzerinde gösterimi

Algoritmanın son aşamasında ise hesaplanan alt turları elimine ederek en son optimal turun belirlenmesini sağlayan algoritma yazılmaktadır. Yazılan algoritma Çizelge 3.20’de verilmektedir. Kodlama sonucu elde edilen veriler Çizelge 3.21’de sunulmaktadır.

Çizelge 3.20. Alt turları eleyerek optimum güzergâhı oluşturan kodlama

```

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

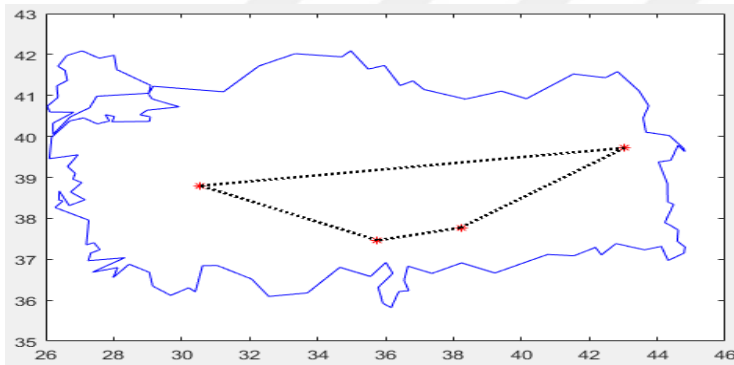
```



Çizelge 3.21. (Devam) Alt turları eleyerek optimum güzergâhı oluşturan kodların sonuçları

Karar değış.	Optimum mesafe	Çıkış noktası	Çıktı
0	2939	1	relativegap: 0
1			absolutegap: 0
0			numfeaspoints: 1
0			numnodes: 0
0			constrviolation: 0
0			message: Optimal solution found. Intlinprog stopped at the root node
0			because the objective value is within a gap tolerance of the optimal
1			value, options. AbsoluteGapTolerance = 0 (the default value). The
1			intcon variables are integer within tolerance, options. IntegerTolerance =
0			1e-05 (the default value).
0			
0			
0			
0			
1			
0			
0			

Lineer optimizasyon sonucu elde edilen güzergahın Türkiye haritası üzerindeki gösterimi Şekil 3.3'te gösterilmektedir.



Şekil 3.3. Güzergâhın harita üzerinde gösterimi

### 3.3. Dört OSB'ye Ait Genetik Algoritmasının Oluşturulması

Lineer optimizasyonda olduğu gibi seçilen 4 OSB için algoritmalar oluşturulmuştur. Ancak burada gerçek mesafeler yerine şehirlerin koordinatları kullanılarak Öklid bağlantısı ile birbirleri arasındaki mesafe değerleri hesaplanıp optimum mesafe bulunmaya çalışılmıştır. Bunun amacı koordinatlar ile de nasıl işlem yapılacağını göstermektir. Analizler MATLAB programı kullanılarak yapılmıştır. MATLAB .m uzantılı dosyalar ile çalışmaktadır. Genetik Algoritma kodlamayı kendi arasında farklı .m uzantılı dosyalar oluşturularak işlem yapar.



Genel kodlama ile bu dosyaları çağırarak optimum mesafeyi hesaplamaya çalışılmaktadır. Bu .m uzantılı dosyalar;

- Create\_permutations
- Crossover\_Permutation
- Mutate\_Permutation
- Traveling\_salesman\_plot
- Traveling\_salesman\_fitness

Create\_permutations.m uzantılı dosyada bulunan kodlama ile nüfus popülasyonu oluşturulur. Bu bölüm, gerekli üç işlevin nasıl oluşturulacağını ve kaydedileceğini gösterir. Gezgin satıcı sorunu için popülasyondaki bir birey sıralı bir kümedir ve böylece popülasyon bir hücre dizisi kullanılarak kolayca temsil edilebilir. Gezgin satıcı sorunu için özel oluşturma işlevi, her bir ögenin permütasyon vektörü olarak sıralı bir şehirler kümesini temsil ettiği bir hücre dizisi oluşturur, örneğin | P | Yani, satıcı | P {i} | 'de belirtilen sırada seyahat edecektir. Oluşturma işlevi | PopulationSize | boyutunda bir hücre dizisini döndürmektedir. Kodlama Çizelge 3.22'de verilmiştir. İşlevin bağımsız değişkenleri;

- NVARs: Değişken sayısı
- FITNESSFCN: Uygunluk fonksiyonu
- OPTIONS: GA tarafından kullanılan seçenekler yapısı

Çizelge 3.22. Create\_permutations.m dosyasındaki kodlama

```
function pop = create_permutations (NVARs, FitnessFcn, options)
totalPopulationSize = sum (options.PopulationSize);
n = NVARS;
pop = cell (totalPopulationSize,1);
for i = 1:totalPopulationSize
    pop{i} = randperm(n);
end
```

Crossover\_permutation.m uzantılı dosyada bulunan kodlama ile gezgin satıcı için özel çaprazlama işlevi gerçekleştirir. Özel crossover işlevi, bir hücre dizisini, popülasyonu alır ve crossover'dan kaynaklanan çocuklar olan bir hücre dizisini döndürür. Kodlama Çizelge 3.23'te verilmiştir. İşlevin bağımsız değişkenleri;

- PARENTS: Seçim işlevi tarafından seçilen ebeveynler.
- STATE: GA çözücü tarafından kullanılan durum yapısı.
- THISSCORE: Mevcut nüfusun puanlarının vektörü.

- THISPOPULATION: Mevcut popülasyondaki bireylerin matrisi.

Çizelge 3.23. Crossover\_permutations.m dosyasındaki kodlama

```
function xoverKids = crossover_permutation(parents,options,NVARS,FitnessFcn,thisScore, thisPopulation)
nKids = length(parents)/2;
xoverKids = cell(nKids,1);
index = 1;
for i=1:nKids
    parent = thisPopulation{parents(index)};
    index = index + 2;
    p1 = ceil((length(parent) - 1) * rand);
    p2 = p1 + ceil((length(parent) - p1 - 1) * rand);
    child = parent;
    child(p1:p2) = fliplr(child(p1:p2));
    xoverKids{i} = child;
end
```

Mutate\_permutation.m uzantılı dosyada bulunan kodlama ile gezgin satıcı için özel mutasyon işlevini gerçekleştirir. Mutasyona uğramış çocukları yeniden oluşturmak için ebeveynleri değiştirir. Özel mutasyon işlevi sıralı şehirler kümesi olan bir kişiyi alır ve mutasyona uğramış bir sıralı seti döndürür. Kodlama Çizelge 3.24'te verilmiştir. İşlevin bağımsız değişkenleri;

- MUTATIONRATE: Mutasyon oranı.

Çizelge 3.24. Crossover\_permutations.m dosyasındaki kodlama

```
function mutationChildren = mutate_permutation(parents ,options,NVARS, FitnessFcn, state, thisScore,
thisPopulation,mutationRate)
mutationChildren = cell(length(parents),1);
for i=1:length(parents)
    parent = thisPopulation{parents(i)};
    p = ceil(length(parent) * rand(1,2));
    child = parent;
    child(p(1)) = parent(p(2));
    child(p(2)) = parent(p(1));
    mutationChildren{i} = child;
end
```

Traveling\_salesman\_fitness.m uzantılı dosyada bulunan kodlama gezgin satıcı için özel form koruma işlevini gerçekleştirir. Bireylerin uygunlunu kontrol etmeye yarayan kodlamadır. Bireyin uygunluğu, düzenli bir şehirler kümesi için kat edilen toplam mesafeyi temsil eder. Uygunluk, X'deki sıralı şehirler kümesi için seyahat edilen toplam mesafedir. DISTANCE, şehirlere ait olan mesafe matrisini temsil eder. Ayrıca gezgin satıcı sorunu için bir tur fonksiyonuna ihtiyaç vardır. Kondisyon fonksiyonu ayrıca toplam mesafeyi

hesaplamak için mesafe matrisine ihtiyaç duymaktadır. Kodlama Çizelge 3.25'te verilmiştir.

Çizelge 3.25. Traveling\_salesman\_fitness.m dosyasındaki kodlama

```
function scores = traveling_salesman_fitness(x,distances)
scores = zeros(size(x,1),1);
for j = 1:size(x,1)
    p = x{j};
    f = distances(p(end),p(1));
    for i = 2:length(p)
        f = f + distances(p(i-1),p(i));
    end
    scores(j) = f;
end
```

Traveling\_salesman\_plot.m uzantılı dosyada bulunan kodlama gezgin satıcı için özel bir çizim fonksiyonu işlevini gerçekleştirir. Şehirlerin konumlarını ve aralarındaki bağlantıyı harita üzerinde göstermeye yarar. Kodlama ait bilgiler Çizelge 3.26'da verilmiştir.

Çizelge 3.26. Traveling\_salesman\_plot.m dosyasındaki kodlama

```
function state = traveling_salesman_plot(options,state,flag,locations)
persistent x y xx yy
if strcmpi(flag,'init')
    load('usborder.mat','x','y','xx','yy');
end
plot(x,y,'Color','red');
axis([-0.1 1.5 -0.2 1.2]);
hold on;
[unused,i] = min(state.Score);
genotype = state.Population{i};
plot(locations(:,1),locations(:,2),'bo');
plot(locations(genotype,1),locations(genotype,2));
hold off
```

Genetik algoritmanın genel kodlaması bölümler halinde anlatılmıştır. İlk kısımda Çizelge 3.27'de verilen kodlamada kaç şehrin olacağını, bu şehirlerin koordinatlarını ve bu koordinatlara bağlı olarak aralarındaki mesafe değerlerini belirlemeye yarar. Oluşturulan mesafe değerlerini matris haline getirir.

Çizelge 3.27. Genetik algoritma genel kodlaması birinci kısım

```

cities = 4;
locations = [37.4613620000000 37.7721820000000 38.7960380000000 39.7202420000000;...
 35.7655520000000 38.2133670000000 30.5594300000000 43.0326270000000]';
xp=[37.4613620000000 37.7721820000000 38.7960380000000 39.7202420000000]';
yp=[35.7655520000000 38.2133670000000 30.5594300000000 43.0326270000000]';
distances = zeros(cities);
for count1=1:cities,
    for count2=1:count1,
        x1 = locations(count1,1);
        y1 = locations(count1,2);
        x2 = locations(count2,1);
        y2 = locations(count2,2);
        distances(count1,count2)=sqrt((x1-x2)^2+(y1-y2)^2);
        distances(count2,count1)=distances(count1,count2);
    end;
end;

```

Algoritmanın ikinci kısmında yukarıda anlatılan genetik algoritmanın farklı .m uzantılı dosyalarını çağırma yarayan komut satırıdır. Bu komut satırı Çizelge 3.28'de verilmektedir.

Çizelge 3.28. Genetik algoritma genel kodlaması ikinci kısım

```

type create_permutations.m
type crossover_permutation.m
type mutate_permutation.m
type traveling_salesman_fitness.m
FitnessFcn = @(x) traveling_salesman_fitness(x,distances);
type traveling_salesman_plot.m

```

Algoritmanın üçüncü kısmında optimum mesafeyi elde etmek için gereken ayarları oluşturan komut satırları bulunmaktadır. İlk olarak, özel bir veri türünü ve nüfus aralığını belirtmek için bir seçenekler kabı oluşturulur. Oluşturulan özel veri turu, geçiş, mutasyon ve çizim işlevlerini seçmenin yanı sıra bazı durma (kısıtlama) koşulları belirlenmektedir. Bu komut satırı Çizelge 3.29'da verilmektedir.

Çizelge 3.29. Genetik algoritma genel kodlaması üçüncü kısım

```

my_plot = @(options,state,flag) traveling_salesman_plot(options,state,flag,locations);
options = optimoptions(@ga, 'PopulationType', 'custom', 'InitialPopulationRange',[1;cities]);
options =
optimoptions(options,'CreationFcn',@create_permutations,'CrossoverFcn',@crossover_permutation,'MutationFcn',@mutate_permutation, ...
'PlotFcn', my_plot,
'MaxGenerations',500,'PopulationSize',60,'MaxStallGenerations',200,'UseVectorized',true);

```

Algoritmanın dördüncü ve son kısmında bulunan komut satırı genetik algoritmayı çağırır ve optimum mesafe değerini elde etmeye yaramaktadır. Komut satırı Çizelge 3.30'da verilmiştir.

Çizelge 3.30. Genetik algoritma genel kodlaması dördüncü kısım

```
numberOfVariables = cities;
[x,fval,reason,output] =ga(FitnessFcn,numberOfVariables,[],[],[],[],[],[],[],options)
displayEndOfDemoMessage(mfilename)
```

### 3.4. Dört OSB'ye Ait PSO Algoritmasının Oluşturulması

Yapılan çalışmada, lineerde ve genetik algoritma için seçilen 4 OSB için algoritmalar oluşturulmuştur. Kullanılan PSO algoritması, genetik algoritma gibi kodlamayı kendi arasında farklı .m uzantılı dosyalar oluşturularak işlem yapmaktadır. Genel kodlama ile bu dosyaları çağırarak optimum mesafeyi hesaplamaya çalışılmaktadır. Bu .m uzantılı dosyalar (kendi adlandırmamız);

- CreateModel.m
- Mutate.m
- MyCost.m
- PlotSolution.m

CreateModel.m uzantılı dosyada bulunan kodlama ile OSB şehirlerine ait koordinat verilerini bünyede tutar ve girilen mesafe değerlerini matrise dönüştürür. Bu dosyaya ait kodlama Çizelge 3.31'de verilmektedir. Kodlama sonucu elde edilen çıktı Çizelge 3.32'de verilmektedir.

Çizelge 3.31. CreateModel.m dosyasındaki kodlama

```
function model=CreateModel()
    y=[37.461362 37.772182 38.796038 39.720242];
    x=[35.765552 38.213367 30.55943 43.032627];
    N=numel(x);
    D=[0 325 646 950;...
        325 0 897 650;...
        646 897 0 1318;...
        950 650 1318 0];
    model.N=N;
    model.x=x;
    model.y=y;
    model.D=D;
```

Çizelge 3.32. CreateModel.m dosyasındaki kodlama çıktısı

<b>Mesafe Matrisi</b>				
	1. şehir	2. şehir	3. şehir	4. Şehir
1. şehir	0	325	646	950
2. şehir	325	0	897	650
3. şehir	646	897	0	1318
4. Şehir	950	650	1318	0

Mutate.m uzantılı dosyada bulunan kodlama ile oluşabilecek turların belirlenmesine yardımcı olmaktadır. Oluşacak turları bir döngü kullanarak bulmaya çalışır. Bu dosyaya ait kodlama Çizelge 3.33'te verilmektedir.

Çizelge 3.33. Mutate.m dosyasındaki kodlama

```
function xnew=Mutate(x)
    [~, Tour]=sort(x);
    M=randi([1 3]);
    switch M
        case 1
            NewTour=DoSwap(Tour);
        case 2
            NewTour=DoReversion(Tour);
        case 3
            NewTour=DoInsertion(Tour);
    end
    xnew=zeros(size(x));
    xnew(NewTour)=x(Tour);
end
function NewTour=DoSwap(Tour)
    n=numel(Tour);
    i=randsample(n,2);
    i1=i(1);
    i2=i(2);
    NewTour=Tour;
    NewTour([i1 i2])=Tour([i2 i1]);
end
function xnew=Mutate(x)
    [~, Tour]=sort(x);
    M=randi([1 3]);
    switch M
        case 1
            NewTour=DoSwap(Tour);
        case 2
            NewTour=DoReversion(Tour);
        case 3
            NewTour=DoInsertion(Tour);
    end
    xnew=zeros(size(x));
    xnew(NewTour)=x(Tour);
end
```

Çizelge 3.24. (Devam) Mutate.m dosyasındaki kodlama

```

function NewTour=DoSwap(Tour)
    n=numel(Tour);
    i=randsample(n,2);
    i1=i(1);
    i2=i(2);
    NewTour=Tour;
    NewTour([i1 i2])=Tour([i2 i1]);
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
function NewTour=DoInsertion(Tour)
    n=numel(Tour);
    i=randsample(n,2);
    i1=i(1);
    i2=i(2);
    if i1<i2
        NewTour=[Tour(1:i1) Tour(i2) Tour(i1+1:i2-1) Tour(i2+1:end)];
    else
        NewTour=[Tour(1:i2-1) Tour(i2+1:i1) Tour(i2) Tour(i1+1:end)];
    end
end
end

```

MyCost.m uzantılı dosyada bulunan kodlama ile oluşturulan turlar arasında en uygun turu bulmaya yarar. Bulunan bu tura bağlı olarak optimum mesafeyi hesaplar. Bu dosyaya ait kodlama Çizelge 3.34'te verilmektedir.

Çizelge 3.34. MyCost.m dosyasındaki kodlama

```

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
    end

```

Çizelge 3.34. (Devam) MyCost.m dosyasındaki kodlama

```

L=L+D(i,j);
end
z=L;
sol.Tour=Tour;
sol.L=L;
end

```

PlotSolution.m uzantılı dosyada bulunan kodlama ile oluşturulan turun Türkiye haritası üzerinde gösterilmesini sağlar. Bu dosyaya ait kodlama Çizelge 3.35'te verilmektedir.

Çizelge 3.35. PlotSolution.m dosyasındaki kodlama

```

function PlotSolution(tour,model)
x=model.x;
y=model.y;
tour=[tour tour(1)];
plot(x(tour),y(tour),'b-o',...
'LineWidth',1,...
'MarkerSize',5,...
'MarkerFaceColor','y');
end

```

PSO algoritmasının genel kodlaması genetik algoritma gibi bölümler halinde anlatılmıştır. İlk kısımda Çizelge 3.36'da verilen komut satırı ile oluşturulan CreateModel ve MyCost dosyalarını içeri aktarmamıza yararmaktadır. Bu komut satırı oluşabilecek karar değişkenleri (optimum mesafe) sayısı ve sınırları belirlenmektedir.

Çizelge 3.36. PSO genel algoritmasının birinci kısmı

```

global NFE;
NFE=0;
model=CreateModel(); % mesafe matrisi fonksiyonu
CostFunction=@(x) MyCost(x,model); % Optimum Mesafe fonksiyonu
nVar=model.N; % karar değişkeni sayısı
VarSize=[1 nVar]; % karar değişkenlerinin matrisinin boyutu
VarMin=0; % Değişkenlerin alt sınırı
VarMax=1; %Değişkenlerin üst sınırı

```

Algoritmanın ikinci kısmında algoritmanın kaç defa iterasyon yapacağını ve popülasyon büyüklüğünü belirlemeye yarayan komut satırlarını içermektedir. Oluşabilecek varyasyonların sınırları belirlenmektedir. Komut satırı Çizelge 3.37' de verilmiştir.



Çizelge 3.37. PSO genel algoritmasının ikinci kısmı

```

MaxIt=200; % Maksimum iterasyon sayısı
nPop=100; % Nüfus büyüklüğü (sürü büyüklüğü)
w=1; % atalet ağırlığı
wdamp=0.99; % atalet ağırlığının sönümleme oranı
c1=0.2; % kişisel öğrenme katsayısı
c2=0.4; % Global öğrenme katsayısı
% hız sınırları
VelMax=0.1*(VarMax-VarMin);
VelMin=-VelMax;

```

Algoritmanın üçüncü kısmında algoritmanın ayarlamaları yapılmaktadır. Burada oluşacak optimum turu ve mesafeyi belirlemeye yaran komut satırı bulunmaktadır. Komut satırı Çizelge 3.38’ de verilmiştir.

Çizelge 3.38. PSO genel algoritmasının üçüncü kısmı

```

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;

```

Algoritmanın dördüncü kısmında oluşan popülasyonu döngü ile döndürür ve başlangıç noktasını belirler. Belirlenen noktaya göre optimum mesafeyi hesaplar. Komut satırı Çizelge 3.39’da verilmiştir.

Çizelge 3.39. PSO genel algoritmasının dördüncü kısmı

```

for i=1:nPop
    % konumu başlatma komutu
    particle(i).Position=unifrnd(VarMin,VarMax,VarSize);
    % Initialize Velocity
    particle(i).Velocity=zeros(VarSize);
    % Değerlendirme komutu
    [particle(i).Cost particle(i).Sol]=CostFunction(particle(i).Position);
    % Kişisel En iyiyi bulma komutu
    particle(i).Best.Position=particle(i).Position;
    particle(i).Best.Cost=particle(i).Cost;
    particle(i).Best.Sol=particle(i).Sol;
    % Global En iyiyi bulma komutu
    if particle(i).Best.Cost<GlobalBest.Cost
        GlobalBest=particle(i).Best;
    end
end
end

```

Çizelge 3.40'ta verilen ve algoritmanın beşinci kısmını oluşturan komut satırı ile bir önceki komut satırınca oluşturulan optimum güzergahın kontrolü sağlanır. Elde edilen mesafe değerinin optimum olup olmadığını kontrol eder. Optimum mesafeyi bulana kadar sürekli konumu güncelleyerek bulmaya devam eder. Optimum mesafeyi bulana kadar elde edilen bütün değerleri grafiğe dökmemizi de sağlamaktadır.

Çizelge 3.40. PSO genel algoritmasının beşinci kısmı

```

BestCost=zeros(MaxIt,1);
nfe=zeros(MaxIt,1);
or it=1:MaxIt
    for i=1:nPop
        % Hızı Güncelleme komutu
        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);
        % Hız sınırlarını uygulama komutu
        particle(i).Velocity = max(particle(i).Velocity,VelMin);
        particle(i).Velocity = min(particle(i).Velocity,VelMax);
        % Konumu güncelleme komutu
        particle(i).Position = particle(i).Position + particle(i).Velocity;
        % Hız ayna Efektı komutu
        IsOutside=(particle(i).Position<VarMin | particle(i).Position>VarMax);
        particle(i).Velocity(IsOutside)=-particle(i).Velocity(IsOutside);
        % Konum sınırlarını uygulama komutu
        particle(i).Position = max(particle(i).Position,VarMin);
        particle(i).Position = min(particle(i).Position,VarMax);
        % Değerlendirme komutu
        [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
        % Kişisel en iyiyi güncelleme komutu
        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;
        end
        % Global en iyiyi güncelleme komutu
        if particle(i).Best.Cost<GlobalBest.Cost
            GlobalBest=particle(i).Best;
        end
    end
end
end

```

Çizelge 3.40. (Devam) PSO genel algoritmasının beşinci kısmı

```

% Global en iyiyi güncelleme komutu
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

```

Algoritmanın altıncı ve son kısmında Türkiye haritasını çizmeye yarayan komut satırı bulunmaktadır. Elde edilen optimum mesafe güzergahını ve değerlendirme sonuçlarını grafiğe döken komut satırıdır. Bu komut satırı Çizelge 3.41’de verilmiştir.

Çizelge 3.41. PSO genel algoritmasının altıncı kısmı

```

load('mesafe_tr_koor.mat')
hold on
plot(trLon,trLat,'color','k','MarkerSize',20)
hold on
plot(AvpX,AvpY,'color','k','MarkerSize',20)
figure;
plot(nfe,BestCost,'LineWidth',2);
xlabel('Değerlendirme');
ylabel('En İyi Mesafe');

```

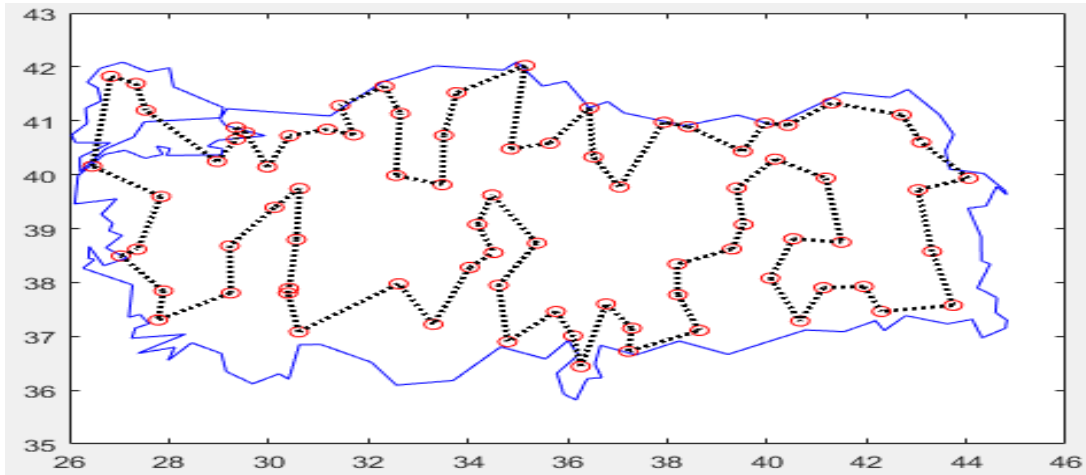
## 4. ARAŞTIRMA BULGULARI VE TARTIŞMA

Yapılan çalışmada daha önceki bölümlerde gösterilen veriler üzerinden Türkiye'deki 81 ildeki OSB'ler için MATLAB programı kullanılarak kuş uçuşu mesafeler ile koordinatlara göre standart çözüm yapılmaktadır. Daha sonra ise yine aynı OSB'ler üzerinden lineer optimizasyon, PSO algoritması ve GA ile yine daha önceki bölümlerde gösterilen gerçek değerler üzerinden optimum mesafe çözümleri yapılarak değerlendirilmektedir. GA, MATLAB programının help menüsünde bulunan kodlamadan yola çıkılarak üstünde değişiklikler yapılarak elde edilmiştir. Türkiye'ye ait veriler giriler analizler yapılmıştır. PSO algoritması, GA gibi modifiye edilerek modifiyeli PSO oluşturulmuştur. GA ile aynı mantıkta çalışmaktadır. Lineer optimizasyon algoritması da MATLAB programının help menüsünde bulunan kodlamadan yola çıkılarak elde edilmiştir. MATLAB programı kullanılarak bütün analizler yapılmıştır.

### 4.1. Lineer Optimizasyon İle Koordinatlara Göre Standart Çözümü

81 adet organize sanayi bölgesi verilerine ait Çizelge 3.2' de verilen koordinatlar üzerinden Öklid bağıntısı ile bulunan kuş uçuşu mesafelere göre lineer optimizasyon algoritması ile yapılan çözümde elde edilen optimum mesafe değeri 73.39 km olarak bulunmaktadır.

#### 4.1.1. Güzergâhın Türkiye haritasında gösterimi

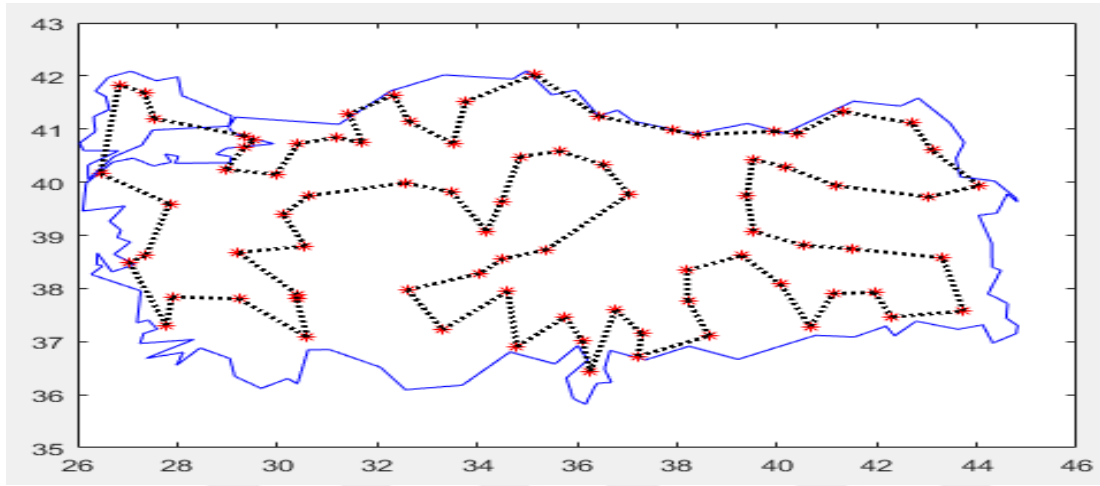


Şekil 4.1. LO GSP örneğinin koordinatlara göre çözümünün harita üzerinde gösterimi

## 4.2. Lineer Optimizasyon İle Karayolu Mesafelerine Göre Çözümü

81 adet organize sanayi bölgesi verilerine ait Çizelge 3.2' de verilen gerçek karayolu mesafeleri üzerinden lineer optimizasyon algoritması ile yapılan çözümde elde edilen optimum mesafe değeri 9 923 km olarak bulunmaktadır

### 4.2.1. Güzergâhın Türkiye haritasında gösterimi

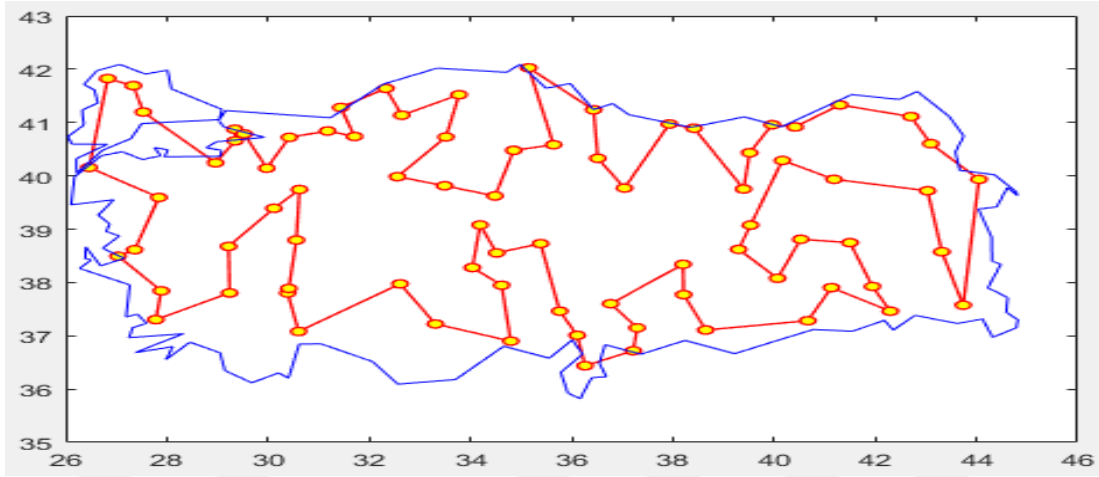


Şekil 4.2. LO GSP örneğinin gerçek karayolu mesafelere göre çözümünün harita üzerinde gösterimi

## 4.3. Parçacıklı Sürü Optimizasyon İle Koordinatlara Göre Standart Çözümü

81 adet organize sanayi bölgesi verilerine ait Çizelge 3.2' de verilen koordinatlar üzerinden Öklid bağıntısı ile bulunan kuş uçuşu mesafelere göre PSO algoritması ile yapılan çözümde elde edilen optimum mesafe değeri 77,23 km olarak bulunmaktadır.

#### 4.3.1. Güzergâhın Türkiye haritasında gösterimi

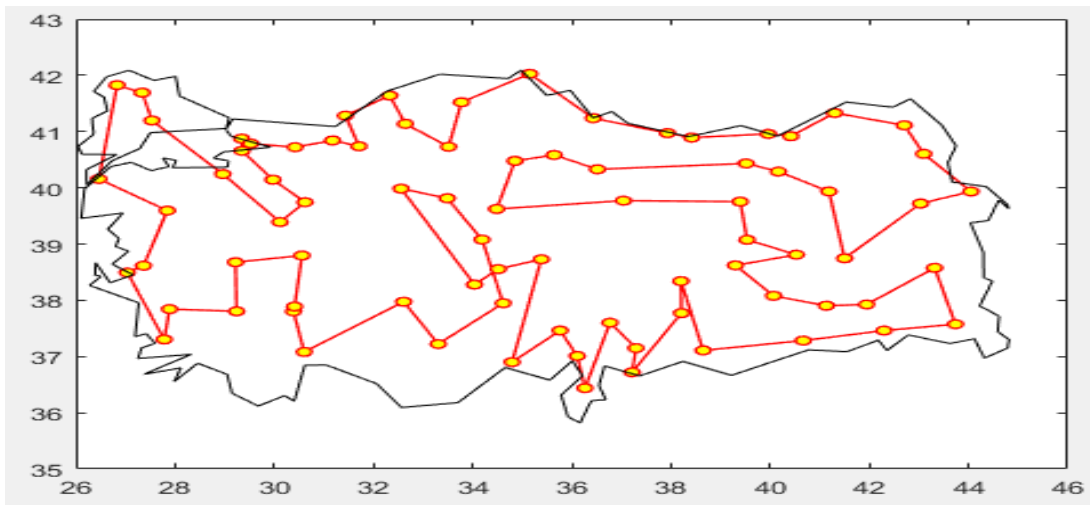


Şekil 4.3. PSO GSP örneğinin koordinatlara göre çözümünün harita üzerinde gösterimi

#### 4.4. Parçacıklı Sürü Optimizasyon İle Karayolu Mesafelerine Göre Çözümü

81 adet organize sanayi bölgesi verilerine ait Çizelge 3.2' de verilen gerçek karayolu mesafeleri üzerinden PSO algoritması ile yapılan çözümde elde edilen optimum mesafe değeri 11 313 km olarak bulunmaktadır

#### 4.4.1. Güzergâhın Türkiye haritasında gösterimi

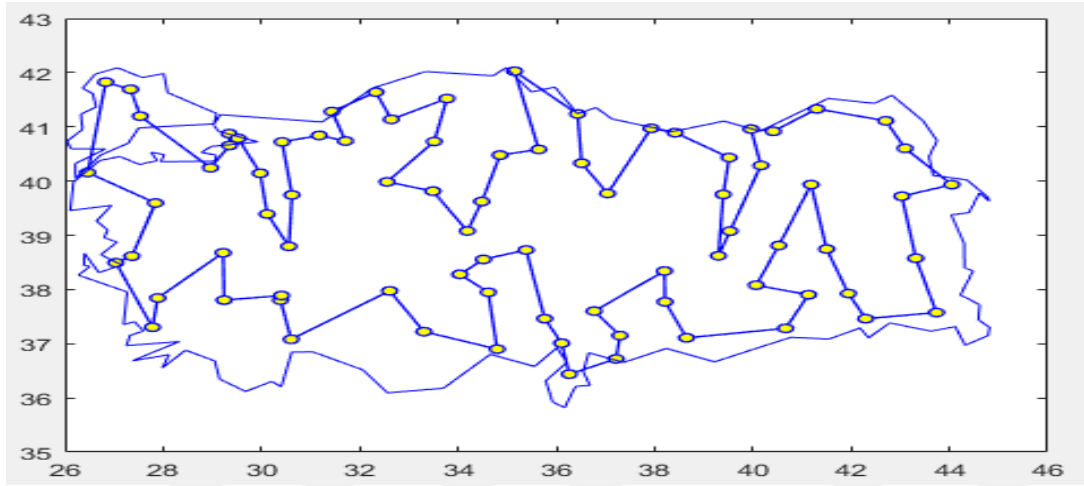


Şekil 4.4. PSO GSP örneğinin gerçek karayolu mesafelerine göre çözümünün harita üzerinde gösterimi

#### 4.5. Genetik Algoritma İle Koordinatlara Göre Standart Çözümü

81 adet organize sanayi bölgesi verilerine ait Çizelge 3.2' de verilen koordinatlar üzerinden Öklid bağıntısı ile bulunan kuş uçuşu mesafelere göre Genetik algoritma ile yapılan çözümde elde edilen optimum mesafe değeri 76,05 km olarak bulunmaktadır.

##### 4.5.1. Güzergâhın Türkiye haritasında gösterimi

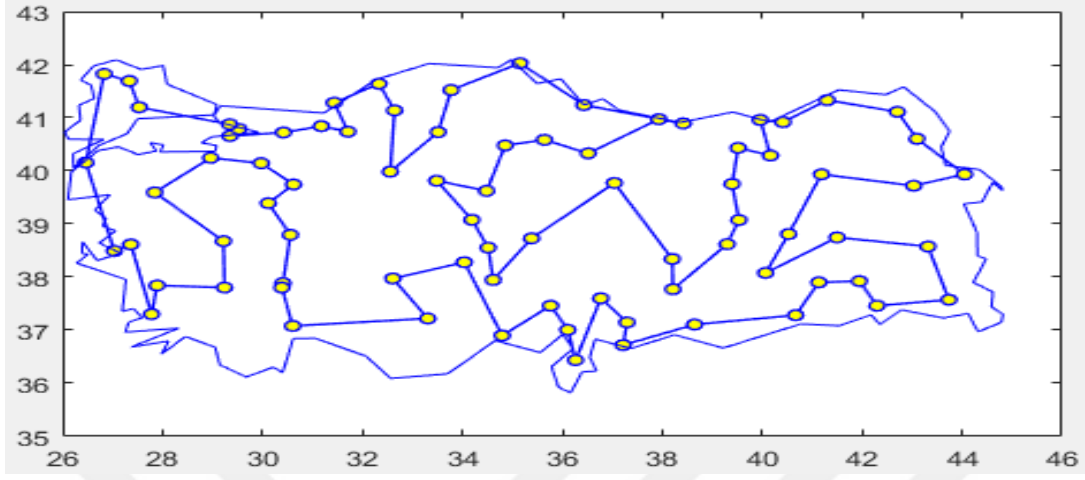


Şekil 4.5. GA GSP örneğinin koordinatlara göre çözümünün harita üzerinde gösterimi

#### 4.6. Genetik Algoritma İle Karayolu Mesafelerine Göre Çözümü

81 adet organize sanayi bölgesi verilerine ait Çizelge 3.2' de verilen gerçek karayolu mesafeleri üzerinden Genetik algoritma ile yapılan çözümde elde edilen optimum mesafe değeri 10 689 km olarak bulunmaktadır

#### 4.6.1. Güzergâhın Türkiye haritasında gösterimi



Şekil 4.6. GA GSP örneğinin gerçek karayolu mesafelerine göre çözümünün harita üzerinde gösterimi



## 5. SONUÇLAR VE ÖNERİLER

Bu çalışmada, 81 ildeki organize sanayi bölgeleri için yük taşımacılığında karayolu taşımacılığı kullanılarak optimum mesafe değeri hesaplanmıştır. Yapılan çalışmada GA, Lineer Optimizasyon ve PSO yöntemi kullanılmıştır.

Lineer Optimizasyon ile yapılan analizler sonucunda organize sanayi bölgeleri için elde edilen optimum mesafe değeri 9 923 km olarak bulunmuştur. Koordinatlara göre çözüm yapıldığında elde edilen değer 73,39 km olarak bulunmuştur. PSO için yapılan analizler sonucunda organize sanayi bölgeleri için elde edilen optimum mesafe değeri 11 313 km olarak bulunmuştur. Koordinatlara göre çözüm yapıldığında elde edilen değer 77,23 km olarak bulunmuştur. GA için yapılan analizler sonucunda organize sanayi bölgeleri için elde edilen optimum mesafe değeri 10 689 km olarak bulunmuştur. Koordinatlara göre çözüm yapıldığında elde edilen değer 76,05 km olarak bulunmuştur. Lineer Optimizasyon, PSO ve GA yöntemleri sonucu elde edilen değerler birbirleri ile kıyaslandığında Lineer optimizasyon GA ve PSO' ya göre optimum mesafe değerini vermektedir. PSO ve GA iterasyon üzerine olduğu için kaçınıcı itteresyonda 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.

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. Çok türlü taşımacılık yaygın taşımacılık türlerindeki trafik yoğunluğunun azalmasında olumlu katkı sağlayacaktır.

## KAYNAKLAR

- Ahmed, Z. H. Genetic algorithm for the traveling salesman problem using sequential constructive crossover operator. *International Journal of Biometrics and Bioinformatics (IJBB)*, 3(6), 96, 2010.
- Cansız, O. F., Ünsalan, K., Çalışıcı, M., and Göçmen, S. (2018). Çok türlü taşımacılık güzergâhlarının gezgin satıcı problemleri baz alınarak tasarlanması: güzergâhların optimizasyonu ve akıllı ulaşım sistemlerine entegrasyonu. 1. *Uluslararası Akıllı Ulaşım Sistemleri Kongresi*, 58-59.
- Cansız, Ö. F. and BOZKURT, M. A. Çok Türlü Taşımacılık Güzergâhlarında Lineer Optimizasyon ve Parçacıklı Sürü Optimizasyonun Karşılaştırılması. 6. *Uluslararası Multidisipliner Çalışmaları Kongresi*, 241-253.
- Cansız, Ö. F. and GÜNDOĞDU, H. Gezgin Satıcı Probleminde Optimizasyon Tekniklerinin Karşılaştırılması: Türkiye Organize Sanayi Bölgeleri Örneği. 5th *International Conference on Multidisciplinary Sciences (icomus)*, 321-335.
- Cansız, Ö. F. TOPAL, H. A. Coğrafi Bilgi Sistemi Tabanlı Optimizasyon Analizleri İçin Çok Türlü Taşımacılık Güzergâhlarının Oluşturulması. 6. *Uluslararası Multidisipliner Çalışmaları Kongresi*, 255-271.
- Cansız, Ö. F., and Ünsalan, K. (2020). Multimodal ve unimodal taşımacılık için rota karşılaştırılması: Hatay-Tekirdağ vaka analizi. *Balıkesir Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 22(1), 300-317.
- Cansız, Ö. F., and ÜNSALAN, K. Demiryolu Yük Taşımacılığının Çok Türlü Rotalara Etkisinin Vaka Analizi ile İncelenmesi: Hatay-Van. *Bitlis Eren Üniversitesi Fen Bilimleri Dergisi*, 8(3), 921-931.
- Cansız, Ö. F., and ÜNSALAN, K. Yük Taşımacılığında Tek Türlü ve Çok Türlü Taşımacılık Rotalarının Karbon Ayak İzinin Karşılaştırılması. *Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi*, 11(2), 809-816.
- Cansız, Ö. F., ÇALIŞICI, M., ÜNSALAN, K. and ERGİNER, İ. Taşımacılık Sürelerinin Tek Türlü ve Çok Türlü Taşımacılık Rotaları Esas Alınarak Karşılaştırılması. *Uluslararası 1. İnşaat Mühendisliğinde İnovasyon, Sürdürülebilirlik, Teknoloji ve Eğitim Konferansı*, 1118-1123.
- Cansız, Ö. F., ERGİNER, İ. and GÜNDOĞDU, H. Gezgin Satıcı Probleminin Analizlerinde Kullanılan Bazı Algoritmaların Kıyaslanması. III. *Uluslararası Avrasya Multidisipliner Çalışmalar Kongresi*, 634-645.
- Cansız, O. F., and Göçmen, S. (2018). Distance analysis of multimodal transportation based on traveling salesman problem with particle swarm optimization method. *International Journal of Advanced Engineering Research and Science*, 5(6).

- Cansiz, O. F., and Unsalan, K. (2018). Determination of Multimodal Transportation Routes by The Hamilton Method and Comparison of Performance Index. *V. International Multidisciplinary Congress of Eurasia*, 132-138.
- Cansiz, O. F., and Unsalan, K. (2018). Multimodal Transport and Cost Analysis: A Case of Iskenderun . *V. International Multidisciplinary Congress of Eurasia*, 173-181.
- Chauhan, C., Gupta, R., and Pathak, K. TSP Solver using Constructive Method of Heuristic Approach. *International Journal of Computer Applications*, 53(1), 33-38, 2012.
- Çolak, S. (2010). Genetik algoritmalar yardımı ile gezgin satıcı probleminin çözümü üzerine bir uygulama. *Çukurova Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, 19(3), 423-438.
- Demirtaş, F., and Zengin, K. *Guguk Kuşu Algoritmasının Gezgin Satıcı Problemine Uygulanması ve Simülasyonu*, 2016.
- Dikmen, H., Dikmen, H., Elbir, A., Eksi, Z., and Çelik, F. (2014). Gezgin satıcı probleminin karınca kolonisi ve genetik algoritmalarla eniyilemesi ve karşılaştırılması. *Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 18(1), 8-13.
- Doğan, Ö. İ., and Kırdı, K. (2014). Evsel İlaç Atıklarının Toplanması Tersine Lojistik Ağı Üzerine Bir Uygulama. *Dokuz Eylül Üniversitesi Denizcilik Fakültesi Dergisi*, 6(1), 1-22.
- Dolgoplov, P., Konstantinov, D., Rybalchenko, L., and Muhitovs, R. (2019). Optimization of train routes based on neuro-fuzzy modeling and genetic algorithms. *Procedia Computer Science*, 149, 11-18.
- Ertuğrul, İ., and Özçil, A. Siyasi Parti Mitinglerinin Gezgin Satıcı Problemi Yaklaşımı ile Analizi. *Siyaset, Ekonomi ve Yönetim Araştırmaları Dergisi*, 4(4), 2016.
- Ghadle, K. P. An application of assignment problem in traveling salesman problem (TSP). *International Journal of Engineering Research and Applications*, 4(1), 169-172, 2014.
- Girdhar Gopal, R. K., Jawa, I., and Kumar, N. Enhanced Order Crossover for Permutation Problems. *International Journal of Innovative Research in Science, Engineering and Technology (An ISO 3297: 2007 Certified Organization) Vol, 4*, 2015.
- Gülsün, B., and Erkayman, B. (2018). LOJİSTİKTE TAŞIMA ŞEKİLLERİNİN BELİRLENMESİ: BİR KOMBİNE TAŞIMACILIK ÖRNEĞİ. *Mehmet Akif Ersoy Üniversitesi Uygulamalı Bilimler Dergisi (MAKUUBD)*, 2(2), 37-51.
- Hao, C., and Yue, Y. (2016). Optimization on Combination of Transport Routes and Modes on Dynamic Programming for a Container Multimodal Transport System. *Procedia Engineering*, 137(1), 382-390.

Hindistan Maharastra 226 Şehir <https://sites.google.com/site/awazyp/tsp>

Hojda, M., Žak, J., and Filcek, G. (2018). Multiple Criteria Optimization of the Joint Vehicle and Transportation Jobs Selection and Vehicle Routing Problems for a Small Road Freight Transportation Fleet. *Transportation research procedia*, 30, 178-187.

İller Arası Mesafe Cetveli, Karayolları Genel Müdürlüğü, <http://www.kgm.gov.tr/Sayfalar/KGM/SiteTr/Root/Uzakliklar.aspx>

Johnson, D. S. Local optimization and the traveling salesman problem. In *International Colloquium on Automata, Languages, and Programming* (pp. 446-461). Springer, Berlin, Heidelberg, 1990.

Johnson, D. S., and McGeoch, L. A. The traveling salesman problem: A case study in local optimization. *Local search in combinatorial optimization*, 1(1), 215-310, 1997.

Jos\_e Fernando Oliveira, Maria Ant\_onia Carravilla. FEUP. <https://paginas.fe.up.pt/~mac/ensino/docs/OR/CombinatorialOptimizationHeuristicsLocalSearch.pdf>, 2009.

Kara, İ., Derya, T., Demir, E., and Bektaş, T. Genelleştirilmiş Gezgin Satıcı Probleminin Tamsayı Doğrusal Karar Modeli. *Yöneylem Araştırması/Endüstri Mühendisliği* 25. Ulusal Kongresi, 4-6, 2005.

Kenya Şehirler Arası Mesafe Cetveli [https://logcluster.org/sites/default/files/documents/Road\\_Distance\\_Matrix](https://logcluster.org/sites/default/files/documents/Road_Distance_Matrix)

Keskintürk, T., Kiremitci, B., and Kiremitci, S. 2-Opt Algoritması Ve Başlangıç Çözümünün Algoritma Sonuçları Üzerindeki Etkisi. *Journal of Industrial Engineering (Turkish Chamber of Mechanical Engineers)*, 27(3), 2016.

Keskintürk, T., Topuk, N., and Özyeşil, O. (2015). Araç Rotalama Problemleri ve Çözüm Yöntemleri. *İşletme Bilimi Dergisi*, 3(2), 77-107.

Király, A., and Abonyi, J. (2011). Optimization of multiple traveling salesmen problem by a novel representation based genetic algorithm. In *Intelligent computational optimization in engineering* (pp. 241-269). Springer, Berlin, Heidelberg.

Köfteci, S., and Gerçek, H. (2010). Yük taşımacılığında taşıma türü seçimi için lojistik maliyetlere dayalı ikili lojit model. *İMO Teknik Dergi*, 5087(5112), 333.

Kuzu, S., Önay, O., Şen, U., Tunçer, M., Yıldırım, B., and Keskintürk, T. Gezgin satıcı problemlerinin metasezgiseller ile çözümü. *İstanbul Üniversitesi İşletme Fakültesi Dergisi*, 43(1), 1-27, 2014.

Lei, K., Zhu, X., Hou, J., and Huang, W. (2014). Decision of multimodal transportation scheme based on swarm intelligence. *Mathematical Problems in Engineering*, 2014.

- Lv, B., Yang, B., Zhu, X., and Li, J. (2019). Operational optimization of transit consolidation in multimodal transport. *Computers and Industrial Engineering*, 129, 454-464.
- Matai, R., Singh, S., and Mittal, M. L. Traveling salesman problem: an overview of applications, formulations, and solution approaches. In *Traveling salesman problem, theory and applications*. IntechOpen, 2010.
- Nilsson, C. Heuristics for the traveling salesman problem. Linkoping University, 1-6, 2003.
- Nuriyeva, F., and Kızılateş, G. Gezgin Satıcı Problemi İçin Merkezden Kenarlara Hipersezgisel Yöntem. *Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 20(2), 2016.
- Özkan, R. (2003). Tek Modelli Deterministik Montaj Hattı Dengeleme Problemlerine Genetik Algoritma İle Çözüm Yaklaşımı (Doctoral dissertation, Fen Bilimleri Enstitüsü).
- Öznur, İ. Ş. Ç. İ., and Korukoğlu, S. (2003). Genetik algoritma yaklaşımı ve yöneylem araştırmasında bir uygulama. *Yönetim ve Ekonomi: Celal Bayar Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, 10(2), 191-20
- ÖZSAĞLAM, M. Y., and ÇUNKAŞ, M. (2008). Optimizasyon problemlerinin çözümü için parçaçık sürü optimizasyonu algoritması. *Politeknik Dergisi*, 11(4), 299-305.
- Prof. Ana Inés Gómez de Castro Universidad Complutense de Madrid, The Travelling Salesman Problem Heuristics [https://www.houspain.com//extern/ Traveller\\_beta/Manual\\_2\\_eng.pdf](https://www.houspain.com//extern/Traveller_beta/Manual_2_eng.pdf)
- Pu, H., Song, T., Schonfeld, P., Li, W., Zhang, H., Hu, J., ... and Wang, J. (2019). Mountain railway alignment optimization using stepwise and hybrid particle swarm optimization incorporating genetic operators. *Applied Soft Computing*, 78, 41-57.
- PULAT, M., and KOCAKOÇ, İ. D. (2017). GEZGİN SATICI PROBLEMİNİN ÇÖZÜMÜNDE KULLANILAN GENETİK ALGORİTMANIN PARAMETRELERİNİN İNCELENMESİ. *Uluslararası İktisadi ve İdari İncelemeler Dergisi*, 21-36.
- Rego, C., Gamboa, D., Glover, F., and Osterman, C. Traveling salesman problem heuristics: Leading methods, implementations and latest advances. *European Journal of Operational Research*, 211(3), 427-441, 2011.
- Saud, S., Kodaz, H., and Babaoğlu, İ. Solving Travelling Salesman Problem by Using Optimization Algorithms. *KnE Social Sciences*, 3(1), 17-32, 2018.
- Singh, G., and Chopra, V. (2012). An analysis of various techniques to solve travelling salesman problem: A review. *International Journal of Advanced Research in Computer Science*, 3(5), 2012.

- Şahin, Y., and Karagül, K. (2019). Gezin satıcı probleminin melez akışkan genetik algoritma (MAGA) kullanarak çözümü.
- Wang, X., and Meng, Q. (2017). Discrete intermodal freight transportation network design with route choice behavior of intermodal operators. *Transportation Research Part B: Methodological*, 95, 76-104.
- Yıldırım, T., Kalyacı, C. B., and Mutlu, Ö. Gezin satıcı problemi için yeni bir meta-sezgisel: kör fare algoritması, 2016.
- Yıldırım, A. E., and Karci, A. (2018). Application of traveling salesman problem for 81 provinces in Turkey using artificial atom algorithm. In *Proceedings of 7th International Conference on Advanced Technologies (ICAT'18)* (pp. 722-726).



## EKLER

### EK-1. Linear Optimizasyon Algoritması

```

clear all;
close all;
clc;
load('mesafe_tr_koor');
SureMatrisi=gsp;
plot(trLon,trLat,'color','b','MarkerSize',20);
hold on
plot(y,x,'*r')
plot(AvpX,AvpY,'color','b','MarkerSize',20);
fprintf('Şehir çiftleri oluşturma\n');
SehirSayisi = size(SureMatrisi,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) = SureMatrisi(c,:);
    c=c+1;
end
UzaklikMesafeVektoru=length(MesafeVektoru);
fprintf('Eşitlik kısıtlamaları yaratılıyor \n');
Aeq=spones(1:length(SehirCifti));
beq=SehirSayisi;
Aeq=[Aeq;spalloc(2*SehirSayisi,length(SehirCifti),2*SehirSayisi*(SehirSayisi+SehirSayisi-1))];
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)];
nonExists=sparse(MesafeVekturu==0);
Aeq(2*c,:)=nonExists';
beq=[beq;0];
intcon=1:UzaklikMesafeVekturu;
lb=zeros(UzaklikMesafeVekturu,1);
ub=ones(UzaklikMesafeVekturu,1);
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,y,x);
Turlar=detectSubtours(KararDegiskenleri,SehirCifti);
AltTurSayisi=length(Turlar);
fprintf('alt tur sayıları: %d\n',AltTurSayisi);
A=spalloc(0,UzaklikMesafeVekturu,0);
b=[];
while AltTurSayisi>1
    b=[b;zeros(AltTurSayisi,1)];
    A=[A;spalloc(AltTurSayisi,UzaklikMesafeVekturu,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
    end
end

```



```

end
b(EsitsizlikKisitNumarasi)=length(AltTurNo)-1;
end
fprintf('\n Alt turları elimine ederek sorunu tekrar çözme\n');

[KararDegiskenleri,OptimumMaliyet,CikisNoktasi,Cikti]=intlinprog(MesafeVektoru,intco
n,A,b,Aeq,beq,lb,ub,opts);
fprintf('Alt turları tekrar kontrol et\n');
lh = updateSalesmanPlot(lh,KararDegiskenleri,SehirCifti,y,x);
Turlar=detectSubtours(KararDegiskenleri,SehirCifti);
AltTurSayisi=length(Turlar);
fprintf('Alt tur sayıları: %d\n',AltTurSayisi);
end
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-2. PSO Ana Algoritması

```
clear all;
close all;
clc;
global NFE;
NFE=0;
model=CreateModel();
CostFunction=@(x) MyCost(x,model);
nVar=model.N;
VarSize=[1 nVar];
VarMin=0;
VarMax=1;
MaxIt=10000;
nPop=100;
w=1;
wdamp=0.99;
c1=0.2;
c2=0.4;
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
load('mesafe_tr_koor.mat')
hold on
plot(trLon,trLat,'color','k','MarkerSize',20)
hold on
plot(AvpX,AvpY,'color','k','MarkerSize',20)
figure;
plot(nfe,BestCost,'LineWidth',2);
xlabel('Değerlendirme');
ylabel('En İyi Mesafe');

```

## EK-3. PSO Model Algoritması

```
function model=CreateModel()
```

```
    y=[37.461362 37.772182 38.796038 39.720242 38.27962 40.583659 39.985281
36.438213 37.081421 41.111246 41.331422 37.842823 39.594038 41.640073 37.902854
40.289232 40.143815 38.810304 40.738056 37.802546 40.244841 40.157048 40.729186
40.478478 37.803913 38.07913 40.841706 41.825306 38.620488 39.753335 39.934872
39.74385 37.149188 40.892954 40.432911 37.574256 39.934206 37.888089 40.875825
38.491909 37.601103 41.137601 37.221808 40.603598 41.522034 38.72927 39.813963
41.692118 39.078548 36.720531 40.786472 37.973946 39.391802 38.340777 38.615848
37.283114 36.899682 37.303062 38.747088 38.554517 37.947681 40.972422 37.008662
40.917517 40.721792 41.237923 37.924799 42.025929 39.771742 37.108623 37.461253
41.197187 40.329479 40.958776 39.075473 38.677248 38.576521 40.656191 39.623133
41.282363];
```

```
    x=[35.765552 38.213367 30.55943 43.032627 34.03994 35.64367 32.557627 36.266054
30.605369 42.704906 41.300299 27.890524 27.843575 32.328515 41.13364 40.164728
29.974772 40.527762 31.710704 30.387612 28.964237 26.464344 33.516319 34.857005
29.241221 40.068557 31.177041 26.832519 39.300231 39.398121 41.186106 30.622113
37.29552 38.419926 39.518272 43.737493 44.049613 30.410037 29.341796 27.037909
36.772474 32.642985 33.308022 43.097329 33.776064 35.385758 33.488427 27.336426
34.191881 37.217575 29.528102 32.610573 30.117757 38.201391 27.365395 40.668681
34.796679 27.78307 41.50037 34.517217 34.616439 37.931364 36.110558 40.416974
30.421702 36.433645 41.944743 35.151646 37.042907 38.64999 42.297676 27.534818
36.517194 39.961717 39.534661 29.22011 43.31672 29.351163 34.485649 31.43668];
```

```
    N=numel(x);
```

```
    D=[0 325 646 950 335 537 578 176 700 1018 896 916 1043 849 614 705 816 643 737
687 902 1152 664 509 793 523 793 1249 450 599 786 722 200 645 674 896 1050 689 963
986 146 772 351 997 752 237 534 1209 421 269 944 428 742 333 957 531 138 997 717
338 251 622 91 810 864 666 690 795 369 342 697 1116 427 760 527 772 885 975 417 862
325 0 897 650 586 557 716 289 950 758 787 1166 1227 982 316 542 1000 345 871 938
1086 1336 759 635 1044 225 926 1382 215 421 534 906 155 592 539 655 752 940 1097
1236 180 838 602 734 809 387 615 1342 511 207 1078 678 992 108 1207 290 389 1248
419 476 502 614 263 678 998 686 390 814 350 114 455 1250 447 621 280 1023 587 1109
587 996
```

646 897 0 1318 360 572 255 725 265 1362 1157 327 337 517 1189 1025 212 1102 405  
153 298 548 420 493 205 1098 376 667 958 944 1130 129 775 875 994 1471 1428 141  
382 343 775 440 351 1342 508 504 343 627 425 843 368 225 99 840 314 1106 561 428  
1198 421 455 825 650 1071 303 686 1264 679 714 917 1271 535 634 1021 1023 130 1460  
350 454 440

950 650 1318 0 967 762 1082 918 1434 259 428 1637 1579 1139 369 310 1361 365 1140  
1376 1439 1689 985 844 1515 461 1196 1652 484 376 189 1279 758 536 374  
426 150 1382 1366 1653 821 1097 1118 170 986 824 979 1612 938 790 1347  
1100 1353 601 1624 516 1018 1693 244 903 950 579 892 400 1268 721 331  
886 608 640 445 1519 676 417 458 1440 221 1379 864 1266

335 586 360 967 0 370 245 413 471 1009 886 629 693 516 829 696 484 744 404 412  
569 818 331 306 506 702 460 916 600 590 777 389 463 628 665 1159 1074 403 630  
699 408 424 190 988 419 146 201 864 88 532 595 137 455 482 669 795 250 722 840 63  
98 579 339 800 532 501 908 585 360 606 960 783 386 751 665 485 1066 643 187 530  
537 557 572 762 370 0 341 653 784 794 609 901 844 403 792 479 625 629 405 726 703  
953 244 82 778 664 460 916 548 388 575 543 571 326 448 1048 873 714 633 917 495  
362 568 786 251 286 242 876 279 644 612 475 617 453 887 786 554 1001 669 319 397  
277 590 523 532 148 846 266 230 662 960 784 111 473 526 703 895 643 181 530

578 716 255 1082 245 341 0 655 473 1124 927 570 543 278 1018 787 284 924 166  
405 402 714 181 266 448 890 222 678 788 717 903 212 672 645 767 1408 1201 384  
377 586 596 201 408 1115 255 340 116 638 224 745 358 276 286 671 557 1008 500  
661 997 295 354 596 589 841 295 456 1097 426 487 819 1173 541 407 742 853 372  
1255 388 227 292

176 289 725 918 413 653 655 0 779 1022 1006 995 1059 928 580 819 895 609 816  
766 981 1231 743 636 872 489 872 1328 451 652 799 801 153 759 788 862 1016  
768 1044 1065 162 851 430 998 831 416 613 1288 500 127 1025 510 821 334  
1036 497 218 1076 683 422 331 736 90 920 945 780 656 909 484 307 662 1195 542  
870 517 851 851 1033 570 941

700 950 265 1434 471 784 473 779 0 1480 1353 325 499 741 1245 1167 454  
1215 630 115 540 690 644 709 215 1154 618 910 1071 1061 1248 391 830 1091  
1136 1526 1546 125 626 464 767 664 353 1459 732 617 549 818 560 399 618 339  
341 953 429 1162 517 372 1311 519 518 1042 642 1267 546 902 1320 903 831  
973 1327 777 850 1218 1136 287 1516 592 658 682

1018 758 1362 259 1009 794 1124 1022 1480 0 127 1636 1566 1099  
579 353 1348 416 1184 1419 1426 1676 972 845 1513 559 1183 1695 535  
420 239 1323 863 466 417 639 222 1425 1412 1697 872 1141 1161 89 948  
868 1022 1655 981 895 1393 1144 1397 652 1667 640 1123 1718 391 965  
994 509 997 290 1312 651 568 817 651 745 682 1564 707 314 505 1483 434 1422  
907 1309  
896 787 1157 428 886 609 927 1006 1353 127 0 1488 1391 924 638 224 1173  
447 952 1298 1251 1501 797 670 1365 590 1008 1464 566 361 270 1130 893  
291 261 818 401 1301 1180 1504 853 910 1040 268 773 746 829 1424 860 826  
1161 1023 1204 683 1474 671 1023 1570 465 843 872 334 987 115 1081 476  
642 642 530 776 756 1331 532 139 499 1290 613 1191 786 1078  
916 1166 327 1637 629 901 570 995 325 1636 1488 0 315 845 1457 1324 458  
1372 734 261 444 444 749 821 125 1330 683 829 1228 1218 1405 415 1045  
1203 1293 1741 1703 249 546 140 1036 768 611 1616 836 774 672 607 717  
1114 538 495 345 1110 148 1376 831 126 1468 676 714 1154 920 1399 614  
1014 1535 1007 988 1187 1542 568 962 1350 1293 207 1694 512 783 747  
1043 1227 337 1579 693 844 543 1059 499 1566 1391 315 0 609 1522 1298  
256 1436 440 382 150 198 619 785 295 1394 389 356 1292 1206 1393 326 1107  
1108 1267 1803 1691 371 252 187 1100 566 683 1605 687 844 635 313 728  
1176 244 557 252 1174 142 1439 893 365 1532 752 788 1059 983 1304 313  
919 1597 856 1006 1250 1604 274 926 1255 1345 212 1758 218 746 453  
849 982 517 1139 516 403 278 928 741 1099 924 845 609 0 1282 861 377 1011  
169 669 447 720 230 356 722 1098 210 684 856 770 956 486 936 648 806 1429 1254  
658 400 806 861 100 672 1168 159 600 318 643 432 1010 381 540 514 862 750 1194  
763 926 1050 519 607 599 854 845 274 460 1279 338 639 1084 1369 551 519 795  
908 646 1276 387 428 138  
614 316 1189 369 829 792 1018 580 1245 579 638 1457 1522 1282 0 508  
1364 183 1170 1235 1450 1614 1002 877 1341 136 1226 1682 264 450  
385 1269 426 735 572 366 469 1237 1398 1533 503 1137 899 546 1052 687  
914 1642 810 458 1379 973 1289 354 1504 158 686 1545 190 794 799 777 560  
599 1299 897 77 1026 593 308 190 1549 691 615 318 1320 305 1408 829 1295  
705 542 1025 310 696 479 787 819 1167 353 224 1324 1298 861 508 0 1079  
316 859 1109 1157 1407 703 562 1203 462 914 1370 400 127 124 998 654 227  
65 703 422 1115 1087 1372 664 816 851 335 705 557 697 1330 671 798 1068

834 1072 454 1342 543 834 1407 337 654 683 270 757 124 987 412 515 578 341  
646 628 1238 394 108 276 1158 499 1097 597 984  
816 1000 212 1361 484 625 284 895 454 1348 1173 458 256 377 1364 1079 0  
1206 221 341 105 365 400 556 355 1164 170 440 1062 988 1174 97 942 889 1048  
1638 1472 329 157 436 870 347 533 1386 469 614 406 400 498 1010 149 407 128  
945 395 1273 728 541 1302 523 582 840 817 1085 98 701 1432 638 777 1084  
1438 308 697 1036 1126 250 1549 117 517 234  
643 345 1102 365 744 629 924 609 1215 416 447 1372 1436 1011 183 316 1206  
0 1079 1157 1273 1523 867 726 1251 147 1078 1590 146 258 194 1093  
451 537 364 501 479 1163 1307 1444 483 979 884 393 869 595 823 1550 719 483  
1287 880 1167 262 1414 229 711 1456 122 702 716 572 566 408 1207 705 240  
834 486 333 348 1458 558 424 139 1230 348 1317 724 1204  
737 871 405 1140 404 405 166 816 630 1184 952 734 440 169 1170 859 221 1079  
0 555 300 550 179 404 607 1041 56 513 939 767 954 312 822 669 827 1523 1252  
544 230 636 747 126 558 1165 248 485 255 473 369 896 210 426 346 821 580 1159  
650 799 1047 445 505 620 740 865 130 480 1242 417 625 970 1324 381 517 815  
905 465 1329 240 366 129  
687 938 153 1376 412 726 405 766 115 1419 1298 261 382 669 1235 1109 341  
1157 555 0 431 584 576 649 141 1112 509 801 1010 1000 1187 282 816 1030  
1075 1512 1485 15 499 392 816 596 382 1399 664 556 494 760 499 884 491 282  
233 893 362 1147 602 318 1250 459 488 981 691 1226 437 842 1305 835 771 958  
1312 668 790 1177 1075 178 1477 469 598 573  
902 1086 298 1439 569 703 402 981 540 1426 1251 444 150 447 1450 1157  
105 1273 300 431 0 258 478 644 417 1253 248 361 1151 1065 1252 185 1030  
967 1125 1726 1550 392 111 329 959 425 622 1464 546 703 494 321 586 1099  
103 496 191 1033 288 1362 816 507 1391 611 671 918 906 1163 172 778 1520  
715 865 1173 1527 228 785 1114 1203 255 1617 77 605 312  
1152 1336 548 1689 818 953 714 1231 690 1676 1501 444 198 720 1614  
1407 365 1523 550 584 258 0 729 895 491 1504 499 246 1402 1317 1503 436  
1281 1218 1377 1977 1801 566 362 304 1210 676 873 1715 797 954 746 265  
838 1350 354 747 436 1284 319 1613 1067 506 1642 863 922 1169 1157  
1414 423 1029 1710 967 1116 1424 1778 247 1036 1365 1455 407 1868  
328 856 563



664 759 420 985 331 244 181 743 644 972 797 749 619 230 1002 703 400 867 179 576  
 478 729 0 181 622 866 234 690 749 611 797 387 662 512 671 1270 1095 558 407  
 761 587 136 495 1009 90 354 130 650 244 765 388 363 461 654 731 1028 575 827  
 891 331 419 463 665 708 307 323 1072 261 431 839 1194 558 322 659 749 547 1117  
 417 215 290  
 509 635 493 844 306 82 266 636 709 845 670 821 785 356 877 562 556 726 404 649 644  
 895 181 0 699 742 461 917 625 470 657 463 554 385 530 1130 955 635 633 837 479  
 311 496 868 199 261 162 877 220 628 614 396 537 531 808 864 504 903 750 294 348 336  
 613 581 534 196 927 280 308 702 1038 784 192 532 608 623 977 644 122 481  
 793 1044 205 1515 506 778 448 872 215 1513 1365 125 295 722 1341 1203  
 355 1251 607 141 417 491 622 699 0 1206 523 654 1103 1094 1280 312 920  
 1079 1169 1616 1578 125 526 252 912 633 487 1492 712 650 536 611 592 989  
 518 371 242 986 222 1252 707 209 1343 552 594 1029 796 1275 446 890 1410  
 883 864 1063 1417 677 838 1225 1169 154 1570 478 658 579  
 523 225 1098 461 702 664 890 489 1154 559 590 1330 1394 1098 136 462 1164  
 147 1041 1112 1253 1504 866 742 1206 0 1092 1548 130 350 337 1050  
 330 621 468 498 561 1121 1265 1401 407 1004 803 536 919 553 781 1508 677  
 362 1245 839 1124 220 1372 124 590 1449 221 660 663 643 464 551 1165 777  
 205 906 460 212 289 1416 557 543 190 1224 396 1275 697 1162  
 793 926 376 1196 460 460 222 872 618 1183 1008 683 389 210 1226 914 170  
 1078 56 509 248 499 234 461 523 1092 0 462 994 823 1009 261 878 724 883 1482  
 1307 494 179 585 802 182 614 1221 304 541 310 422 425 951 159 482 292 877 529  
 1214 706 748 1103 501 560 675 795 920 78 536 1303 473 681 1062 1416 330  
 572 871 961 414 1384 189 421 78  
 1249 1382 667 1652 916 916 678 1328 910 1695 1464 829 356 684 1682  
 1370 440 1590 513 801 361 246 690 917 654 1548 462 0 1457 1285 1472 554  
 1382 1187 1345 2078 1770 787 296 539 1265 645 991 1684 766 1009 773  
 58 893 1451 317 865 586 1339 674 1714 1168 893 1566 963 1023 1138  
 1258 1383 392 998 1872 935 1144 1525 1879 151 1035 1384 1423 550  
 1847 334 884 524  
 450 215 958 484 600 548 788 451 1071 535 566 1228 1292 856 264 400 1062 146  
 939 1010 1151 1402 749 625 1103 130 994 1457 0 237 314 948 318 499 355  
 621 599 1019 1162 1299 338 874 739 513 790 451 678 1406 575 386 1143 736

1022 118 1270 253 552 1329 242 558 571 521 421 528 1063 655 338 784 331 304  
427 1313 428 430 126 1085 468 1173 568 1060  
599 421 944 376 590 388 717 652 1061 420 361 1218 1206 770 450 127 988 258  
767 1000 1065 1317 611 470 1094 350 823 1285 237 0 189 906 519 283 120  
662 487 1009 995 1401 558 724 745 401 613 451 605 1239 565 580 976 726 980  
319 1345 460 731 1320 283 548 577 326 692 259 896 451 460 579 234 511 634 1146  
303 242 141 1066 564 1005 490 892  
786 534 1130 189 777 575 903 799 1248 239 270 1405 1393 956 385 124 1174  
194 954 1187 1252 1503 797 657 1280 337 1009 1472 314 189 0 1093 642  
349 187 593 301 1195 1182 1588 651 911 931 214 800 638 801 1425 751 674  
1162 913 1175 431 1532 419 902 1551 214 735 764 392 776 214 1082 534 391  
700 421 524 504 1333 489 230 275 1262 377 1192 677 1079  
722 906 129 1279 389 543 212 801 391 1323 1130 415 326 486 1269 998 97 1093  
312 282 185 436 387 463 312 1050 261 554 948 906 1093 0 847 843 963 1543  
1396 266 267 517 776 408 439 1310 476 520 311 510 404 916 259 313 103 850 461  
1179 633 679 1208 428 488 794 723 1039 186 654 1337 647 682 990 1344 418  
602 989 1050 224 1473 233 422 320  
200 155 775 758 463 571 672 153 830 863 893 1045 1107 936 426 654 942 451 822  
816 1030 1281 662 554 920 330 878 1382 318 519 642 847 0 678 707 703 857 817  
1093 1113 81 785 479 839 779 334 562 1336 458 62 1073 555 869 205 1084  
339 266 1125 524 471 379 655 140 854 993 699 497 827 402 150 504 1244 460 797  
387 900 693 1103 495 990  
645 592 875 536 628 326 645 759 1091 466 291 1203 1108 648 735 227 889 537  
669 1030 967 1218 512 385 1079 621 724 1187 499 283 349 843 678 0 164 960  
678 1017 896 1302 607 625 794 592 488 501 544 1139 601 756 877 777 919 494  
1246 737 780 1465 594 598 627 50 741 202 796 191 771 357 284 703 885 1047 250  
153 418 1005 755 906 503 793  
674 539 994 374 665 448 767 788 1136 417 261 1293 1267 806 572 65 1048 364  
827 1075 1125 1377 671 530 1169 468 883 1345 355 120 187 963 707 164 0  
802 521 1090 1069 1354 646 798 826 434 650 540 679 1313 653 795 1050 807  
1054 460 1419 581 812 1439 436 629 658 211 781 177 970 353 614 519 323 632  
727 1220 377 127 262 1140 598 1079 579 966  
896 655 1471 426 1159 1048 1408 862 1526 639 818 1741 1803 1429 366  
703 1638 501 1523 1512 1726 1977 1270 1130 1616 498 1482 2078

621 662 593 1543 703 960 802 0 422 1514 1651 1810 782 1380 1175 553  
 1269 1071 1298 2033 1194 735 1632 1252 1566 738 1781 383 963 1822  
 380 1167 1076 971 837 793 1552 1107 292 1235 950 561 210 1940 1047 809  
 613 1596 207 1661 1187 1549  
 1050 752 1428 150 1074 873 1201 1016 1546 222 401 1703 1691 1254  
 469 422 1472 479 1252 1485 1550 1801 1095 955 1578 561 1307 1770  
 599 487 301 1396 857 678 521 422 0 1492 1479 1885 937 1208 1228 136  
 1097 935 1089 1722 1048 890 1460 1210 1464 715 1829 614 1117 1803  
 389 1032 1061 689 991 511 1357 831 430 997 718 740 544 1630 786 527 572  
 1550 217 1489 974 1376  
 689 940 141 1382 403 714 384 768 125 1425 1301 249 371 658 1237 1115 329  
 1163 544 15 392 566 558 635 125 1121 494 787 1019 1009 1195 266 817 1017  
 1090 1514 1492 0 485 378 820 568 385 1406 636 564 471 712 503 888 478 285  
 219 901 349 1151 606 329 1258 466 492 968 695 1213 423 828 1309 821 779 962  
 1316 654 776 1163 1083 165 1484 455 582 556  
 963 1097 382 1366 630 633 377 1044 626 1412 1180 546 252 400 1398 1087  
 157 1307 230 499 111 362 407 633 526 1265 179 296 1162 995 1182 267 1093  
 896 1069 1651 1479 485 0 444 972 351 706 1390 473 716 480 256 600 1128 31  
 580 279 1046 388 1391 875 607 1280 670 730 845 972 1090 101 705 1480 642  
 850 1202 1557 163 743 1040 1130 372 1554 48 591 231  
 986 1236 343 1653 699 917 586 1065 464 1697 1504 140 187 806 1533 1372  
 436 1444 636 392 329 304 761 837 252 1401 585 539 1299 1401 1588 517 1113  
 1302 1354 1810 1885 378 444 0 1107 770 691 1685 853 844 687 511 780  
 1185 442 565 352 1180 43 1448 901 205 1539 748 797 1170 992 1509 517  
 1031 1606 1024 1059 1259 1613 472 979 1366 1364 216 1766 396 798  
 649  
 146 180 775 821 408 495 596 162 767 872 853 1036 1100 861 503 664 870 483 747  
 816 959 1210 587 479 912 407 802 1265 338 558 651 776 81 607 646 782 937 820  
 972 1107 0 709 478 854 691 259 486 1214 382 142 951 545 830 225 1078 418 255  
 1125 583 337 330 578 140 771 871 623 576 752 327 229 583 1121 385 721 408 893  
 772 980 420 868  
 772 838 440 1097 424 362 201 851 664 1141 910 768 566 100 1137 816 347 979  
 126 596 425 676 136 311 633 1004 182 645 874 724 911 408 785 625 798 1380 1208  
 568 351 770 709 0 594 1121 129 453 223 596 337 858 333 462 466 784 703 1121

668 866 1003 424 512 576 757 821 253 429 1200 308 566 932 1286 503 473 771  
 861 568 1284 363 334 152  
 351 602 351 1118 190 568 408 430 353 1161 1040 611 683 672 899 851 533 884  
 558 382 622 873 495 496 487 803 614 991 739 745 931 439 479 794 826 1175 1228  
 385 706 691 478 594 0 1138 584 296 368 934 276 546 699 134 445 607 662 809 240  
 692 986 236 165 761 353 954 627 689 967 755 510 620 974 858 568 905 811 477 1163  
 673 375 685  
 997 734 1342 170 988 786 1115 998 1459 89 268 1616 1605 1168 546 335  
 1386 393 1165 1399 1464 1715 1009 868 1492 536 1221 1684 513 401  
 214 1310 839 592 434 553 136 1406 1390 1685 854 1121 1138 0 1009 847  
 1002 1635 961 870 1372 1122 1385 627 1742 616 1098 1716 334 944 973  
 596 972 377 739 738 511 903 631 721 624 1542 699 400 485 1463 349 1402 887  
 1289  
 752 809 508 986 419 251 255 831 732 948 773 836 687 159 1052 705 469 869 248 664  
 546 797 90 199 712 919 304 766 790 613 800 476 779 488 650 1269 1097 636 473  
 853 691 129 584 1009 0 447 218 717 332 853 454 451 549 705 824 1037 663 938  
 893 424 512 444 757 689 374 304 1070 182 482 927 1212 624 362 639 751 635 1174  
 483 298 267  
 237 387 504 824 146 286 340 416 617 868 746 774 844 600 687 557 614 595 485 556 703  
 954 354 261 650 553 541 1009 451 451 638 520 334 501 540 1071 935 564 716 844  
 259 453 296 847 447 0 230 958 126 408 695 279 574 334 811 671 270 864 691 100 129  
 474 359 665 615 449 760 533 221 474 837 865 279 616 516 627 918 724 164 611  
 534 615 343 979 201 242 116 613 549 1022 829 672 635 318 914 697 406 823 255 494  
 494 746 130 162 536 781 310 773 678 605 801 311 562 544 679 1298 1089 471 480  
 687 486 223 368 1002 218 230 0 728 115 636 465 237 388 561 659 899 446 772 919  
 202 290 495 539 740 385 355 987 389 383 710 1064 635 303 690 744 474 1145 494  
 118 382  
 1209 1342 627 1612 864 876 638 1288 818 1655 1424 607 313 643 1642  
 1330 400 1550 473 760 321 265 650 877 611 1508 422 58 1406 1239 1425  
 510 1336 1139 1313 2033 1722 712 256 511 1214 596 934 1635 717 958 728  
 0 852 1410 276 807 511 1299 452 1673 1128 670 1525 923 982 1097 1217  
 1342 351 957 1832 895 1103 1484 1838 81 994 1293 1383 507 1806 293  
 843 483

421 511 425 938 88 279 224 500 560 981 860 717 728 432 810 671 498 719 369 499 586  
838 244 220 592 677 425 893 575 565 751 404 458 601 653 1194 1048 503 600 780  
382 337 276 961 332 126 115 852 0 532 579 222 458 457 738 795 332 808 815 87 176  
532 421 779 499 412 883 496 335 606 960 749 320 729 640 544 1041 609 98 496  
269 207 843 790 532 644 745 127 399 895 826 1114 1176 1010 458 798 1010 483  
896 884 1099 1350 765 628 989 362 951 1451 386 580 674 916 62 756 795 735 890  
888 1128 1185 142 858 546 870 853 408 636 1410 532 0 1143 625 939 267 1154  
369 321 1195 555 475 434 717 195 842 1008 761 528 890 465 180 535 1313 523  
859 517 969 724 1118 551 1005  
944 1078 368 1347 595 612 358 1025 618 1393 1161 538 244 381 1379 1068  
149 1287 210 491 103 354 388 614 518 1245 159 317 1143 976 1162 259 1073  
877 1050 1632 1460 478 31 442 951 333 699 1372 454 695 465 276 579 1143 0  
577 295 1028 386 1374 857 605 1263 652 712 827 955 1072 83 687 1463 624  
825 1185 1539 186 725 1022 1112 369 1535 46 572 212  
428 678 225 1100 137 475 276 510 339 1144 1023 495 557 540 973 834 407 880  
426 282 496 747 363 396 371 839 482 865 736 726 913 313 555 777 807 1252 1210  
285 580 565 545 462 134 1122 451 279 237 807 222 625 577 0 321 616 536 891 346  
591 973 181 230 729 435 937 501 589 1042 623 493 702 1056 732 537 888 798 351  
1199 548 320 554  
742 992 99 1353 455 617 286 821 341 1397 1204 345 252 514 1289 1072 128  
1167 346 233 191 436 461 537 242 1124 292 586 1022 980 1175 103 869 919  
1054 1566 1464 219 279 352 830 466 445 1385 549 574 388 511 458 939 295 321  
0 924 322 1202 656 428 1281 501 550 867 745 1112 219 727 1360 721 756 1012  
1367 432 676 1063 1096 137 1508 266 496 352  
333 108 840 601 482 453 671 334 953 652 683 1110 1174 862 354 454 945 262 821  
893 1033 1284 654 531 986 220 877 1339 118 319 431 850 205 494 460 738 715  
901 1046 1180 225 784 607 627 705 334 561 1299 457 267 1028 616 924 0 1152  
342 434 1212 358 440 444 490 304 575 945 584 427 712 248 216 516 1196 345 523  
183 968 584 1055 485 942  
957 1207 314 1624 669 887 557 1036 429 1667 1474 148 142 750 1504 1342  
395 1414 580 362 288 319 731 808 222 1372 529 674 1270 1345 1532 461 1084  
1246 1419 1781 1829 349 388 43 1078 703 662 1742 824 811 659 452 738  
1154 386 536 322 1152 0 1418 873 212 1509 718 767 1141 962 1386 454

1001 1576 1004 1029 1229 1583 544 949 1403 1335 187 1736 359 769  
 593  
 531 290 1106 516 795 786 1008 497 1162 640 671 1376 1439 1194 158 543  
 1273 229 1159 1147 1362 1613 1028 864 1252 124 1214 1714 253 460  
 419 1179 339 737 581 383 614 1151 1391 1448 418 1121 809 616 1037 671  
 899 1673 795 369 1374 891 1202 342 1418 0 598 1456 303 749 711 795 472  
 632 1282 899 200 1028 581 196 184 1575 679 648 318 1231 448 1392 819 1279  
 138 389 561 1018 250 554 500 218 517 1123 1023 831 893 763 686 834 728 711  
 650 602 816 1067 575 504 707 590 706 1168 552 731 902 633 266 780 812 963 1117  
 606 875 901 255 668 240 1098 663 270 446 1128 332 321 857 346 656 434 873 598 0  
 913 784 259 168 751 143 934 782 699 757 783 498 410 764 1032 556 885 618 688 953  
 870 407 779  
 997 1248 428 1693 722 1001 661 1076 372 1718 1570 126 365 926 1545  
 1407 541 1456 799 318 507 506 827 903 209 1449 748 893 1329 1320 1551  
 679 1125 1465 1439 1822 1803 329 607 205 1125 866 692 1716 938 864 772  
 670 808 1195 605 591 428 1212 212 1456 913 0 1641 775 796 1252 999 1497  
 677 1113 1614 1106 1087 1266 1620 760 1061 1448 1391 291 1793  
 575 881 810  
 717 419 1198 244 840 669 997 683 1311 391 465 1468 1532 1050 190 337 1302  
 122 1047 1250 1391 1642 891 750 1343 221 1103 1566 242 283 214 1208  
 524 594 436 380 389 1258 1280 1539 583 1003 986 334 893 691 919 1525 815  
 555 1263 973 1281 358 1509 303 784 1641 0 798 811 601 659 423 1172 727  
 178 856 560 408 292 1423 579 439 234 1325 227 1282 808 1169  
 338 476 421 903 63 319 295 422 519 965 843 676 752 519 794 654 523 702 445 459 611  
 863 331 294 552 660 501 963 558 548 735 428 471 598 629 1167 1032 466 670 748  
 337 424 236 944 424 100 202 923 87 475 652 181 501 440 718 749 259 775 798 0 98  
 535 346 751 574 487 863 571 315 585 940 824 341 701 619 530 1021 662 186 571  
 251 502 455 950 98 397 354 331 518 994 872 714 788 607 799 683 582 716 505 488 671  
 922 419 348 594 663 560 1023 571 577 764 488 379 627 658 1076 1061 492 730  
 797 330 512 165 973 512 129 290 982 176 434 712 230 550 444 767 711 168 796 811 98  
 0 595 255 779 635 542 869 626 343 522 876 885 401 730 622 580 1045 723 250 632  
 622 614 825 579 579 277 596 736 1042 509 334 1154 1059 599 777 270 840 572  
 620 981 918 1169 463 336 1029 643 675 1138 521 326 392 794 655 50 211 971 689  
 968 845 1170 578 576 761 596 444 474 495 1097 532 717 827 729 867 490 1141

795 751 1252 601 535 595 0 715 246 748 143 801 309 304 708 909 999 199 196 443  
957 799 858 431 731

91 263 650 892 339 590 589 90 642 997 987 920 983 854 560 757 817 566 740 691 906  
1157 665 613 796 464 795 1258 421 692 776 723 140 741 781 837 991 695 972 992  
140 757 353 972 757 359 539 1217 421 195 955 435 745 304 962 472 143 999 659 346  
255 715 0 884 867 744 630 887 462 283 637 1118 520 823 491 774 826 956 493 864  
810 678 1071 400 800 523 841 920 1267 290 115 1399 1304 845 599 124 1085  
408 865 1226 1163 1414 708 581 1275 551 920 1383 528 259 214 1039 854  
202 177 793 511 1213 1090 1509 771 821 954 377 689 665 740 1342 779 842  
1072 937 1112 575 1386 632 934 1497 423 751 779 246 884 0 993 388 605 554  
442 738 719 1243 444 51 411 1202 589 1103 659 975

864 998 303 1268 532 532 295 945 546 1312 1081 614 313 274 1299 987 98 1207  
130 437 172 423 307 534 446 1165 78 392 1063 896 1082 186 993 796 970 1552  
1357 423 101 517 871 253 627 739 374 615 385 351 499 1008 83 501 219 945 454  
1282 782 677 1172 574 635 748 867 993 0 608 1482 545 753 1134 1489 259  
646 943 1033 341 1456 112 493 133

666 686 686 721 501 148 456 780 902 651 476 1014 919 460 897 412 701 705 480 842  
778 1029 323 196 890 777 536 998 655 451 534 654 699 191 353 1107 831 828 705  
1031 623 429 689 738 304 449 355 957 412 761 687 589 727 584 1001 899 699 1113  
727 487 542 143 744 388 608 0 908 168 358 806 1071 858 244 339 563 816 957 717  
315 590

690 390 1264 331 908 846 1097 656 1320 568 642 1535 1597 1279 77 515  
1432 240 1242 1305 1520 1710 1072 927 1410 205 1303 1872 338 460  
391 1337 497 771 614 292 430 1309 1480 1606 576 1200 967 511 1070 760  
987 1832 883 528 1463 1042 1360 427 1576 200 757 1614 178 863 869 801  
630 605 1482 908 0 1034 669 375 116 1624 766 617 364 1397 268 1483 906  
1370

795 814 679 886 585 266 426 909 903 817 642 1007 856 338 1026 578 638 834 417  
835 715 967 261 280 883 906 473 935 784 579 700 647 827 357 519 1235 997 821 642  
1024 752 308 755 903 182 533 389 895 496 890 624 623 721 712 1004 1028 783  
1106 856 571 626 309 887 554 545 168 1034 0 488 936 1201 796 366 509 718 806  
1087 655 400 445

369 350 714 608 360 230 487 484 831 651 530 988 1006 639 593 341 777 486 625 771  
865 1116 431 308 864 460 681 1144 331 234 421 682 402 284 323 950 718 779 850

1059 327 566 510 631 482 221 383 1103 335 465 825 493 756 248 1029 581 498  
1087 560 315 343 304 462 442 753 358 669 488 0 473 755 1005 109 391 372 844  
795 864 260 752  
342 114 917 640 606 662 819 307 973 745 776 1187 1250 1084 308 646 1084 333  
970 958 1173 1424 839 702 1063 212 1062 1525 304 511 524 990 150 703 632  
561 740 962 1202 1259 229 932 620 721 927 474 710 1484 606 180 1185 702  
1012 216 1229 196 410 1266 408 585 522 708 283 738 1134 806 375 936 473 0  
366 1379 568 722 373 1035 579 1217 628 1082  
697 455 1271 445 960 960 1173 662 1327 682 756 1542 1604 1369 190 628  
1438 348 1324 1312 1527 1778 1194 1038 1417 289 1416 1879 427  
634 504 1344 504 885 727 210 544 1316 1557 1613 583 1286 974 624 1212  
837 1064 1838 960 535 1539 1056 1367 516 1583 184 764 1620 292 940 876  
909 637 719 1489 1071 116 1201 755 366 0 1740 853 730 482 1396 380 1600  
990 1444  
1116 1250 535 1519 783 784 541 1195 777 1564 1331 568 274 551 1549  
1238 308 1458 381 668 228 247 558 784 677 1416 330 151 1313 1146 1333  
418 1244 1047 1220 1940 1630 654 163 472 1121 503 858 1542 624 865 635  
81 749 1313 186 732 432 1196 544 1575 1032 760 1423 824 885 999 1118  
1243 259 858 1624 796 1005 1379 1740 0 897 1195 1285 526 1708 165  
745 379  
427 447 634 676 386 111 407 542 850 707 532 962 926 519 691 394 697 558 517 790 785  
1036 322 192 838 557 572 1035 428 303 489 602 460 250 377 1047 786 776 743  
979 385 473 568 699 362 279 303 994 320 523 725 537 676 345 949 679 556 1061 579  
341 401 199 520 444 646 244 766 366 109 568 853 897 0 396 440 766 809 756 217 643  
760 621 1021 417 751 473 742 870 1218 314 139 1350 1255 795 615 108 1036  
424 815 1177 1114 1365 659 532 1225 543 871 1384 430 242 230 989 797 153  
127 809 527 1163 1040 1366 721 771 905 400 639 616 690 1293 729 859 1022  
888 1063 523 1403 648 885 1448 439 701 730 196 823 51 943 339 617 509 391 722  
730 1195 396 0 380 1152 603 1053 610 926  
527 280 1023 458 665 526 853 517 1136 505 499 1293 1345 908 318 276 1126  
139 905 1075 1203 1455 749 608 1169 190 961 1423 126 141 275 1050 387  
418 262 613 572 1083 1130 1364 408 861 811 485 751 516 744 1383 640 517  
1112 798 1096 183 1335 318 618 1391 234 619 622 443 491 411 1033 563 364  
718 372 373 482 1285 440 380 0 1150 476 1157 629 1034



772 1023 130 1440 485 703 372 851 287 1483 1290 207 212 646 1320 1158  
 250 1230 465 178 255 407 547 623 154 1224 414 550 1085 1066 1262 224 900  
 1005 1140 1596 1550 165 372 216 893 568 477 1463 635 627 474 507 544 969  
 369 351 137 968 187 1231 688 291 1325 530 580 957 774 1202 341 816 1397 806  
 844 1035 1396 526 766 1152 1150 0 1549 339 583 477  
 885 587 1460 221 1066 895 1255 851 1516 434 613 1694 1758 1276 305 499  
 1549 348 1329 1477 1617 1868 1117 977 1570 396 1384 1847 468 564  
 377 1473 693 755 598 207 217 1484 1554 1766 772 1284 1163 349 1174 918  
 1145 1806 1041 724 1535 1199 1508 584 1736 448 953 1793 227 1021  
 1045 799 826 589 1456 957 268 1087 795 579 380 1708 809 603 476 1549 0  
 1566 1034 1453  
 975 1109 350 1379 643 643 388 1033 592 1422 1191 512 218 387 1408 1097  
 117 1317 240 469 77 328 417 644 478 1275 189 334 1173 1005 1192 233 1103  
 906 1079 1661 1489 455 48 396 980 363 673 1402 483 724 494 293 609 1118  
 46 548 266 1055 359 1392 870 575 1282 662 723 858 956 1103 112 717 1483  
 655 864 1217 1600 165 756 1053 1157 339 1566 0 601 250  
 417 587 454 864 187 181 227 570 658 907 786 783 746 428 829 597 517 724 366 598 605  
 856 215 122 658 697 421 884 568 490 677 422 495 503 579 1187 974 582 591 798 420  
 334 375 887 298 164 118 843 98 551 572 320 496 485 769 819 407 881 808 186 250 431  
 493 659 493 315 906 400 260 628 990 745 217 610 629 583 1034 601 0 492  
 862 996 440 1266 530 530 292 941 682 1309 1078 747 453 138 1295 984 234 1204 129  
 573 312 563 290 481 579 1162 78 524 1060 892 1079 320 990 793 966 1549 1376 556  
 231 649 868 152 685 1289 267 611 382 483 496 1005 212 554 352 942 593 1279 779 810  
 1169 571 632 731 864 975 133 590 1370 445 752 1082 1444 379 643 926 1034 477 1453  
 250 492 0];

model.N=N;

model.x=x;

model.y=y;

model.D=D;

end

## EK-4. PSO Mutasyon Algoritması

```

function xnew=Mutate(x)
    [~, Tour]=sort(x);
    M=randi([1 3]);
    switch M
        case 1
            NewTour=DoSwap(Tour);
        case 2
            NewTour=DoReversion(Tour);
        case 3
            NewTour=DoInsertion(Tour);
    end
    xnew=zeros(size(x));
    xnew(NewTour)=x(Tour);
end
function NewTour=DoSwap(Tour)
    n=numel(Tour);
    i=randsample(n,2);
    i1=i(1);
    i2=i(2);
    NewTour=Tour;
    NewTour([i1 i2])=Tour([i2 i1]);
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
function NewTour=DoInsertion(Tour)
    n=numel(Tour);

```

```
i=randsample(n,2);  
i1=i(1);  
i2=i(2);  
if i1<i2  
    NewTour=[Tour(1:i1) Tour(i2) Tour(i1+1:i2-1) Tour(i2+1:end)];  
else  
    NewTour=[Tour(1:i2-1) Tour(i2+1:i1) Tour(i2) Tour(i1+1:end)];  
end  
end
```



## EK-5. PSO Hesaplama Algoritması

```
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-6. PSO Yazdırma Algoritması

```
function PlotSolution(tour,model)
    x=model.x;
    y=model.y;
    tour=[tour tour(1)];
    plot(x(tour),y(tour),'b-o',...
        'LineWidth',1,...
        'MarkerSize',5,...
        'MarkerFaceColor','y');
end
```



## EK-7. GA Ana Algoritması

```

cities = 4;
locations = [37.461362 37.772182 38.796038 39.720242; 35.765552 38.213367 30.55943
43.032627]';
xp=[37.461362 37.772182 38.796038 39.720242]';
yp=[35.765552 38.213367 30.55943 43.032627]';
distances = zeros(cities);
for count1=1:cities,
    for count2=1:count1,
        x1 = locations(count1,1);
        y1 = locations(count1,2);
        x2 = locations(count2,1);
        y2 = locations(count2,2);
        distances(count1,count2)=sqrt((x1-x2)^2+(y1-y2)^2);
        distances(count2,count1)=distances(count1,count2);
    end;
end;
type create_permutations.m
type crossover_permutation.m
type mutate_permutation.m
type traveling_salesman_fitness.m
FitnessFcn = @(x) traveling_salesman_fitness(x,distances);
type traveling_salesman_plot.m
my_plot = @(options,state,flag) traveling_salesman_plot(options, ...
    state,flag,locations);
options = optimoptions(@ga, 'PopulationType', 'custom','InitialPopulationRange', ...
    [1;cities]);
options = optimoptions(options,'CreationFcn',@create_permutations, ...
    'CrossoverFcn',@crossover_permutation, ...
    'MutationFcn',@mutate_permutation, ...
    'PlotFcn', my_plot, ...
    'MaxGenerations',500,'PopulationSize',60, ...
    'MaxStallGenerations',200,'UseVectorized',true);

```

```
numberOfVariables = cities;  
[x,fval,reason,output] = ...  
    ga(FitnessFcn,numberOfVariables,[],[],[],[],[],[],options)  
displayEndOfDemoMessage(mfilename)
```



## EK-8. GA Model Algoritması

```
function pop = create_permutations(NVARS,FitnessFcn,options)
totalPopulationSize = sum(options.PopulationSize);
n = NVARS;
pop = cell(totalPopulationSize,1);
for i = 1:totalPopulationSize
    pop{i} = randperm(n);
end
```





## EK-9. GA Mutasyon Algoritması

```
function mutationChildren = mutate_permutation(parents ,options,NVARS, ...
    FitnessFcn, state, thisScore,thisPopulation,mutationRate)
mutationChildren = cell(length(parents),1);
for i=1:length(parents)
    parent = thisPopulation{parents(i)};
    p = ceil(length(parent) * rand(1,2));
    child = parent;
    child(p(1)) = parent(p(2));
    child(p(2)) = parent(p(1));
    mutationChildren{i} = child;
end
```

## EK-10. GA Hesaplama Algoritması

```
function xoverKids = crossover_permutation(parents,options,NVARS, ...
    FitnessFcn,thisScore,thisPopulation)
nKids = length(parents)/2;
xoverKids = cell(nKids,1);
index = 1;
for i=1:nKids
    parent = thisPopulation{parents(index)};
    index = index + 2;
    p1 = ceil((length(parent) - 1) * rand);
    p2 = p1 + ceil((length(parent) - p1 - 1) * rand);
    child = parent;
    child(p1:p2) = fliplr(child(p1:p2));
    xoverKids{i} = child;
end
```

## EK-11. GA Yazdırma Algoritması

```
function scores = traveling_salesman_fitness(xc,distances)
scores = zeros(size(xc,1),1);
for j = 1:size(xc,1)
    p = xc{j};
    f = distances(p(end),p(1));
    for i = 2:length(p)
        f = f + distances(p(i-1),p(i));
    end
    scores(j) = f;
end
function state = traveling_salesman_plot(options,state,flag,locations)
persistent x y xx yy xxx yyy
if strcmpi(flag,'init')
    load('usborder.mat','x','y','xx','yy','xxx','yyy');
end
plot(y,x,'Color','red');
hold on
plot(xxx,yyy,'Color','red');
end
```

## ÖZGEÇMİŞ

### Kişisel Bilgiler

Soyadı, adı : GÜNDOĞDU, Haydar  
 Uyruğu : T.C.  
 Doğum tarihi ve yeri : 1981 / Tunceli  
 Medeni hali : Evli  
 Telefon : 0533 486 21 96  
 Faks : -  
 e-mail : hdmgundogdu@gmail.com



### Eğitim

Derece	Eğitim Birimi	Mezuniyet Tarihi
Yüksek lisans	İskenderun Teknik Üniversitesi / İnşaat Mühendisliği	Devam ediyor
Lisans	Gaziantep Üniversitesi / İnşaat Mühendisliği	Tamamlandı / 2006
Lise	Tunceli Cumhuriyet Lisesi / Fen Bilimleri	Tamamlandı / 1999

### İş Deneyimi

Yıl	Yer	Görev
2013- Devam Ediyor	İskenderun Belediyesi	Fen İşleri Müdür V.
2012 - 2013	Hassa-Kırıkhan Devlet Yolu – DEHA-Özmert / Hatay	Şantiye Şefi
2009 - 2012	Feke I ve Gökkaya Barajları- Egemen İnşaat / Adana	Sanat Yapıları Şefi
2008 - 2009	Mısrata Serbest Bölgesi / Libya	Sanat Yapıları Şefi
2008 - 2009	Kulp I ve IV Barajları / Diyarbakır	Şantiye Şefi
2006 - 2007	Feke I Barajı / Gümüşhane	Saha Mühendisi
2005 -2005	Kavak – Havza Devletiyolu / Samsun	Stajyer
2004 - 2004	Bakü Ceyhan Boru Hattı PTI / Ardahan	Stajyer
2003 - 2003	Birecik- Suruç Otoyolu / Şanlıurfa	Stajyer

### Yabancı Dil

İngilizce

## DİZİN

---

### A

Abstract · v  
 Analiz · iv, 1, 2, 9, 10, 11, 12, 13,  
 14, 15, 16, 17, 18, 19, 20, 21,  
 23, 42, 58  
 Algoritma · iv, vii, viii, ix, xiii, 2,  
 4, 5, 6, 7, 8, 9, 10, 11, 12, 13,  
 14, 15, 16, 17, 18, 19, 20, 21,  
 22, 23, 32, 33, 34, 37, 38, 39,  
 40, 42, 45, 46, 47, 50, 51, 52,  
 53, 54, 55, 56, 57, 58

---

### C-Ç

Çok türlü · iv, 1, 4, 9, 10, 12, 13,  
 15, 16, 18, 19, 20, 58

---

### D

Demiryolu · 1, 12, 13, 16, 17, 18,  
 20  
 Denizyolu · 1, 18, 20

---

### G

Genetik · iv, vii, viii, xi, xiii, 5, 7,  
 8, 9, 10, 11, 12, 13, 14, 15, 16,  
 17, 18, 19, 20, 21, 22, 42, 45,  
 46, 47, 50, 56, 57  
 Gezgin satıcı · iv, vii, xiii, 1, 4, 9,  
 10, 11, 13, 14, 16, 17, 18, 19,  
 20, 21, 22, 43, 44, 45  
 Güzergah · viii, x, xii, 4, 9, 12, 13,  
 14, 15, 16, 17, 18, 19, 20, 21,  
 22, 38, 39, 40, 41, 42, 54, 55,  
 56, 57

---

### H

Harita · viii, x, xii, 13, 22, 32, 33,  
 39, 40, 42, 45, 50, 53, 54, 55,  
 56, 57

---

### K

Karayolu · iv, vii, viii, xii, 1, 2, 3,  
 4, 10, 12, 13, 16, 17, 18, 20, 23,  
 54, 55, 56, 57, 58

---

### L

Lineer · iv, vii, viii, xiii, 2, 4, 6,  
 10, 16, 21, 23, 32, 38, 42, 48,  
 54, 58  
 Lojistik · vii, 1, 3, 10, 12, 16, 19,  
 21, 58

---

### M-N

Mesafe · iv, viii, x, xii, 1, 2, 4, 9,  
 10, 16, 17, 19, 20, 21, 22, 23,  
 32, 33, 37, 38, 39, 40, 41, 42,  
 43, 44, 45, 46, 47, 48, 49, 50,  
 51, 52, 53, 54, 55, 56, 57, 58  
 Maliyet · iv, 1, 4, 6, 9, 12, 13, 14,  
 15, 16, 18, 19, 21, 38, 40

---

### O-Ö

Optimum · iv, v, x, 2, 6, 7, 9, 11,  
 12, 16, 17, 18, 20, 21, 23, 38,  
 39, 40, 41, 42, 43, 46, 47, 49,  
 50, 51, 52, 53, 54, 55, 56, 57,  
 58  
 Optimizasyon · iv, vii, viii, xiii, 1,  
 2, 3, 5, 6, 7, 9, 10, 11, 12, 13,  
 14, 16, 17, 19, 20, 21, 23, 32,  
 38, 42, 54, 55, 56, 58  
 Özet · iv

---

### T

Taşımacılık · iv, vii 1, 2, 3, 4, 9,  
 10, 12, 13, 15, 16, 18, 19, 20,  
 23, 58  
 Tek türlü · 9, 10, 13, 16, 18, 19, 20

---

### U

Ulaşım · 3, 10, 12, 13, 14, 15, 19,  
 20

---

### Y

Yük · iv, vi, 1, 4, 5, 6, 8, 9, 10, 12,  
 13, 14, 15, 16, 18, 19, 20, 21,  
 23, 50, 51, 58

Yolcu · iv 1, 4, 12, 13



**TEKNOVERSİTE**



teknoversite **AYRICALIĞINDASINIZ**

**İSTE**

