



**İSKENDERUN TEKNİK**  
ÜNİVERSİTESİ  
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

**YÜKSEK  
LİSANS  
TEZİ**

**YÜK TAŞIMACILIĞI  
OPTİMİZASYONUNDA KARINCA  
KOLONİSİ VE BENZETİLMİŞ  
TAVLAMA ALGORİTMASI  
YÖNTEMLERİNİN KULLANILMASI**

**Aydın KAŞ**

**İNŞAAT MÜHENDİSLİĞİ  
ANABİLİM DALI**

**OCAK 2023**



**YÜK TAŞIMACILIĞI OPTİMİZASYONUNDA KARINCA KOLONİSİ VE  
BENZETİLMİŞ TAVLAMA ALGORİTMASI YÖNTEMLERİNİN  
KULLANILMASI**

**Aydın KAŞ**

**YÜKSEK LİSANS  
İNŞAAT MÜHENDİSLİĞİ ANABİLİM DALI**

**İSKENDERUN TEKNİK ÜNİVERSİTESİ  
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ**

**OCAK 2023**

**Aydın KAŞ** tarafından hazırlanan “YÜK TAŞIMACILIĞI OPTİMİZASYONUNDA KARINCA KOLONİSİ VE BENZETİLMİŞ TAVLAMA ALGORİTMASI YÖNTEMLERİNİN KULLANILMASI” 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. Fatih ÜNEŞ .....  
İnşaat Mühendisliği Anabilim Dalı, İskenderun Teknik Üniversitesi .....

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

**Üye:** Prof. Dr. Fatih ÜNEŞ .....  
İnşaat Mühendisliği Anabilim Dalı, İskenderun Teknik Üniversitesi .....

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

**Üye:** Prof. Dr. Mehmet ÇAKIROĞLU .....  
İnşaat Mühendisliği Anabilim Dalı, Toros Üniversitesi .....

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

Tez Savunma Tarihi: 12/01/23

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

.....

Doç. Dr. Ersin BAHÇECİ

Lisansüstü Eğitim Enstitüsü Müdürü

## ETİK BEYAN

İskenderun Teknik Üniversitesi Lisansüstü Eğitim Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmada;

- Yükseköğretim Kuruluna gönderilen kopya ile tarafından Lisansüstü Eğitim 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  
Aydın KAŞ  
12/01/2023

# YÜK TAŞIMACILIĞI OPTİMİZASYONUNDA KARINCA KOLONİSİ VE BENZETİLMİŞ TAVLAMA ALGORİTMASI YÖNTEMLERİNİN KULLANILMASI (Yüksek Lisans Tezi)

Aydın KAŞ

İSKENDERUN TEKNİK ÜNİVERSİTESİ  
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

Ocak 2023

## ÖZET

Her geçen gün devam eden nüfus artışı kişisel ve toplumsal ihtiyaçları arttırmakla birlikte, birçok problemi de beraberinde getirmektedir. Artan ihtiyaçları karşılamakta lojistik önemli bir rol almaktadır. Lojistik, bir ürünün üretim noktasından tüketim noktasına ulaşana kadar geçen sürece denilmektedir. Hazırlanan bir ürünün olması istenilen yere zamanında ulaşması büyük önem arz etmektedir. Bu çalışmada örnek bir uygulama üzerinde Gezgin Satıcı Problemine (GSP) çözüm aranmaktadır. Yapılan çalışmada her ilden 1 tane olmak üzere 81 ildeki Organize Sanayi Bölgelerine (OSB) uğranıp, başlangıç noktasına dönülen en kısa yol güzergahı bulunmaktadır. Karayolu taşımacılığında yapılan analizler Karınca Kolonisi Optimizasyonu (KKO) ve Benzetilmiş Tavlama Algoritması (BTA) yöntemleri ile yapılmaktadır. Bulunan analiz sonuçları farklı çalışmalarda yapılan Parçacıklı Sürü Optimizasyonu (PSO), Genetik Algoritma (GA), Lineer Optimizasyon (LO) ve Tabu Arama (TA) algoritması yöntemleri ile karşılaştırılmaktadır. KKO ile 81 OSB için yapılan analiz sonucunda en kısa yol mesafesi 10198 km, BTA ile yapılan analiz sonucunda en kısa yol mesafesi 10133 km olarak bulunmaktadır. Çıkan sonuçlar ve daha önce yapılan analizler ile karşılaştırıldığında BTA ve KKO sonuçları TA, GA ve PSO sonuçlarına göre daha iyi, LO sonuçlarına göre daha kötü sonuç verdiği gözlemlenmektedir.

Anahtar Kelimeler : Karınca kolonisi optimizasyonu, benzetilmiş tavlama algoritması, yük taşımacılığı, gezgin satıcı problemi  
Sayfa Adedi : 104  
Danışman : Dr. Öğr. Üyesi Ömer Faruk CANSIZ

# USING ANT COLONY AND SIMULATED ANNEALING ALGORITHM METHODS IN LOAD TRANSPORT OPTIMIZATION

(M.Sc. Thesis)

Aydın KAŞ

ISKENDERUN TECHNICAL UNIVERSITY  
INSTITUTE OF GRADUATE STUDIES

January 2023

## ABSTRACT

Population growth, which continues day by day, increases personal and social needs and brings many problems with it. Logistics plays an important role in meeting the increasing needs. Logistics is the process that takes a product from the point of production to the point of consumption. It is of great importance that a prepared product reaches the desired place on time. In this study, a solution to the Traveling Salesman Problem (TSP) is sought on a sample application. In the study, there is the shortest route to the Organized Industrial Zones (OIZ) in 81 provinces, one from each province, and return to the starting point. Analysis in road transport is done with Ant Colony Optimization (ACO) and Simulated Annealing Algorithm (SAA). The results of the analysis are compared with the Particle Swarm Optimization (PSO), Genetic Algorithm (GA), Linear Optimization (LO) and Tabu Search (TS) algorithms made in different studies. As a result of the analysis made with ACO for 81 OIZs, the shortest road distance is 10198 km, and as a result of the analysis made with SAA, the shortest road distance is 10133 km. When the results and previous analyses are compared, it is observed that SAA and ACO results are better than TS, GA and PSO results, and worse than LO results.

Keywords : Ant colony optimization, simulated annealing algorithm,  
freight transport, traveling salesman problem  
Page Number : 104  
Supervisor : Asst. Prof. Ömer Faruk CANSIZ

## TEŐEKKÖR

İlk olarak gerek eđitim hayatım gerek normal hayatım boyunca yanımda olan, maddi manevi hiçbir zaman desteđini esirgemeyip her zaman arkamda duran babam Sırrı KAŐ'a sonsuz sevgi, saygı ve teŐekkÖrlerimi sunuyorum. Őu an fiziksel olarak yanımda olmasa da her zaman yanımda hissettiđim, bana hayatı öđreten, üzerimde büyük emeđi olan ve her zaman kalbimde olacak olan annem Güler KAŐ'a sonsuz sevgi, saygı ve teŐekkÖrlerimi sunuyorum. Yüksek lisans tez döneminin boyunca bana sürekli destek olan tecrÜbesi ve bilgi birikiminden her zaman yararlanmamı sađlayan danıŐman hocam Dr. Öđr. Üyesi Ömer Faruk CANSIZ'a sonsuz saygı ve teŐekkÖrlerimi sunarım.



## İÇİNDEKİLER

	Sayfa
ÖZET .....	iv
ABSTRACT.....	v
TEŞEKKÜR.....	vi
İÇİNDEKİLER .....	vii
ÇİZELGELERİN LİSTESİ.....	ix
ŞEKİLLERİN LİSTESİ.....	x
SİMGELER VE KISALTMALAR.....	xi
1. GİRİŞ.....	1
2. ÖNCEKİ ÇALIŞMALAR.....	9
3. MATERYAL ve YÖNTEM.....	19
3.1. Karınca Kolonisi Optimizasyonu ile 5 OSB için Algoritma Oluşturulması .....	34
3.2. Benzetilmiş Tavlama Algoritması ile 5 OSB için Algoritma Oluşturulması .....	46
4. ARAŞTIRMA BULGULARI ve TARTIŞMA.....	57
4.1. Karınca Kolonisi Optimizasyonu ile OSB Koordinatlarına Göre Çözümü .....	57
4.1.1. Tur güzergahının koordinat düzlemi üzerinde gösterimi .....	58
4.1.2. Tur güzergahının Türkiye haritası üzerinde gösterimi .....	58
4.2. Karınca Kolonisi Optimizasyonu ile OSB Gerçek Uzunluklarına Göre Çözümü .....	59
4.2.1. Tur güzergahının koordinat düzlemi üzerinde gösterimi .....	60
4.2.2. Tur güzergahının Türkiye haritası üzerinde gösterimi .....	60
4.3. Benzetilmiş Tavlama Algoritması ile OSB Koordinatlarına Göre Çözümü.....	61
4.3.1. Tur güzergahının koordinat düzlemi üzerinde gösterimi .....	62
4.3.2. Tur güzergahının Türkiye haritası üzerinde gösterimi .....	62



4.3.3. İterasyonlar sonrası sonuçlardaki deęişimin grafiksel gösterimi.....	63
4.4. BTA ile OSB Gerçek Uzunluklarına Göre Çözümü .....	64
4.4.1. Tur güzergahının koordinat düzlemi üzerinde gösterimi .....	64
4.4.2. Tur güzergahının Türkiye haritası üzerinde gösterimi .....	65
4.4.3. İterasyonlar sonrası elde edilen sonuçların grafiksel gösterimi.....	65
5. SONUÇLAR ve ÖNERİLER.....	67
KAYNAKLAR.....	70
EKLER.....	75
EK-1. KKO ModelOlusturma.m Bölümü Kodu.....	75
EK-2. KKO Turmaliyet.m Bölümü Kodu.....	84
EK-3. KKO KKOkomutlari.m Bölümü Kodu .....	85
EK-4. KKO RuletCarki.m Bölümü Kodu.....	87
EK-5. KKO SonuclarinCizimi.m Bölümü Kodu .....	88
EK-6. BTA ModelOlusturma.m Bölümü Kodu.....	89
EK-7. BTA MesafeHesabi.m Bölümü Kodu .....	98
EK-8. BTA RastgeleCozumOlusturma.m Bölümü Kodu.....	99
EK-9. BTA BTAkomutlari.m Bölümü Kodu .....	100
EK-10. BTA KomsuCevapOlusturma.m Bölümü Kodu .....	102
EK-11. BTA TurCizimi.m Bölümü Kodu .....	103
DİZİN.....	104

## ÇİZELGELERİN LİSTESİ

<b>Çizelge</b>	<b>Sayfa</b>
Çizelge 3.1. OSB koordinatları.....	20
Çizelge 3.2. OSB'ler arasındaki karayolu mesafe (km) değerleri .....	22
Çizelge 3.3. ModelOlusturma.m bölümündeki kod yazımı .....	34
Çizelge 3.4. ModelOlusturma.m çıktısındaki gerçek karayolu mesafeleri .....	35
Çizelge 3.5. Turmaliyet.m bölümündeki kod yazımı .....	36
Çizelge 3.6. KKOkomutlari.m bölümündeki kod yazımı .....	36
Çizelge 3.7. RuletCarki.m bölümündeki kod yazımı.....	42
Çizelge 3.8. SonuclarinCizimi.m bölümündeki kod yazımı .....	46
Çizelge 3.9. ModelOlusturma.m bölümündeki kod yazımı .....	47
Çizelge 3.10. MesafeHesabi.m bölümündeki kod yazımı .....	48
Çizelge 3.11. RastgeleCozumOlusturma.m bölümündeki kod yazımı .....	49
Çizelge 3.12. BTAkomutlari.m bölümünün kod yazımı.....	49
Çizelge 3.13. KomsuCevapOlusturma.m bölümünün kod yazımı .....	53
Çizelge 3.14. TurCizimi.m bölümünün kod yazımı .....	56
Çizelge 5.1. Tez çalışmasının sonuçları ve farklı çalışmalardaki yöntemlerin sonuçları.....	68

## ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 1.1. Tavlama işlemindeki süreç .....	6
Şekil 3.1. ModelOluşturma.m bölümünün çıktı sonuçlarının dosya görünümü .....	35
Şekil 3.2. Yerel minimum ile global minimum gösterimi .....	54
Şekil 4.1. Koordinatlara göre KKO yönteminde çözümün koordinat düzlemi üzerinde gösterimi.....	58
Şekil 4.2. KKO yönteminde koordinatlara göre çözümün Türkiye haritası üzerinde gösterimi.....	58
Şekil 4.3. KKO yönteminde gerçek mesafelere göre çözümün koordinat düzleminde gösterimi.....	60
Şekil 4.4. KKO yönteminde gerçek mesafelere göre çözümün Türkiye haritası üzerinde gösterimi .....	60
Şekil 4.5. Koordinatlara göre BTA yönteminde çözümün koordinat düzlemi üzerinde gösterimi.....	62
Şekil 4.6. BTA yönteminde koordinatlara göre çözümün Türkiye haritası üzerinde gösterimi.....	62
Şekil 4.7. Koordinatlara göre çözümün iterasyonlara göre elde edilen sonuçlardaki değişim grafiği.....	63
Şekil 4.8. BTA yönteminde gerçek mesafelere göre çözümün koordinat düzlemi üzerinde gösterimi .....	64
Şekil 4.9. BTA yönteminde gerçek mesafelere göre çözümün Türkiye haritası üzerinde gösterimi.....	65
Şekil 4.10. İterasyonlara göre elde edilen sonuçlardaki değişim grafiği .....	65

## SİMGELER ve KISALTMALAR

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

### **Simgeler**                      **Açıklamalar**

**km**                                      Kilometre

### **Kısaltmalar**                      **Açıklamalar**

**BTA**                                      Benzetilmiş Tavlama Algoritması

**GA**                                      Genetik Algoritma

**GSP**                                      Gezgin Satıcı Problemi

**KKO**                                      Karınca Kolonisi Optimizasyonu

**LO**                                      Lineer Optimizasyon

**OSB**                                      Organize Sanayi Bölgesi

**PSO**                                      Parçacıklı Sürü Optimizasyonu

**TA**                                      Tabu Arama

## 1. GİRİŞ

Her geçen gün artış gösteren nüfus sayısı ve değişen yaşam koşulları nedeniyle ihtiyaçlarda artış göstermektedir. Artan ihtiyaçların karşılanması, istenilen bir ürünün zamanında ulaştırılması vb. konular ele alındığında lojistik sektörü ön plana çıkmaktadır. Lojistik insanlar için her zaman önemli bir faktör olmuştur. Lojistik faaliyetleri içerisinde taşımacılık sektörü büyük öneme sahiptir. Denizyolu taşımacılığı, karayolu taşımacılığı, denizyolu taşımacılığı, demiryolu taşımacılığı olmak üzere çeşitli taşımacılık türleri vardır. Ülkemizde taşımacılık türleri arasından karayolu taşımacılığı daha çok tercih edilmektedir. Taşımacılığın karmaşık yapılması yerine düzenli bir şekilde yapılması; zaman, mesafe ve maliyet açısından büyük yarar sağlayacaktır. Gezgin satıcı problemi (GSP) lojistik sektörü için önemli bir problemdir. GSP n tane şehrin her birine bir kere uğrayıp tekrar başlangıç noktasına en kısa mesafede dönmeyi hedefleyen bir problemdir. Bu kapsamda GSP baz alınarak bir çalışma yapılmaktadır. Bu çalışmada ülkemizdeki her ilden 1 tane olmak üzere toplam 81 adet Organize Sanayi Bölgesi (OSB) kullanılıp yük taşımacılığında en kısa mesafe analizi yapılmaktadır. En kısa mesafe analizi tek türlü taşımacılık yöntemi ile karayolu taşımacılığında yapılmaktadır. GSP çözümü için MATLAB programında Karınca Kolonisi Optimizasyonu (KKO) yöntemi ve Benzetilmiş Tavlama Algoritması (BTA) yöntemi kullanılmaktadır. Bulunan sonuçlar ise; farklı çalışmalarda aynı organize sanayi bölgeleri ile yapılmış olan Genetik Algoritma (GA), Parçacıklı Sürü Optimizasyonu (PSO), Lineer Optimizasyon (LO) ve ArcGIS programında kullanılan Tabu Arama (TA) algoritması sonuçları ile karşılaştırılmaktadır. Bu çalışmanın amacı karınca kolonisi optimizasyonu ve benzetilmiş tavlama algoritması kullanılarak GSP problemine çözüm bulmak ve en kısa yol analizi yapılarak diğer yöntemler ile karşılaştırmaktır.

Günümüzde sürekli artan ihtiyaçlar ve gelişen sektörler kapsamında lojistiğin önemi artarak devam etmektedir. Dilimize yerleşmiş olan lojistik kavramının anlamı ve kökeni ile ilgili farklı bilgiler bulunmaktadır (Koçak, 2020). Türk Dil Kurumu (TDK, 2022) verilerine göre lojistik kavramı; “Kişilerin ihtiyaçlarını karşılamak üzere her türlü ürünün, hizmetin ve bilgi akışının çıkış noktasından varış noktasına kadar taşınmasının etkili ve verimli bir biçimde planlanması ve uygulanması” şeklinde tanımlanmaktadır.

Lojistik sözcüğü aslında orta çağa kadar dayanmaktadır. Latince’de kullanılan logisticus ve Yunanca’da kullanılan logistikos sözcüklerinden üretildiği söylenmektedir. Bu sözcükler

logic ve statics kelimelerinden gelmektedir. Logic kelimesi mantık, statics kelimesi ise istatistik anlamına gelmektedir. Yunanca’da kullanılan Logistikos sözcüğüne bakıldığında anlam olarak “hesap yapma bilimi” veya “hesapta becerikli” anlamlarına geldiği söylenmektedir. Bu kapsamda bakıldığı zaman “mantıklı hesaplama” olarak tanımlanabilir (Van Wassenhove, 2006).

Lojistik sözcüğü aslında bir askeri terim olarak kullanılmıştır. İlk olarak lojistik sözcüğü 1905 senesinde ordu ile alakalı araç gereçlerin ve görevlilerin taşınması, malzemeleri tedarik etme, bakım ve yenilenmesi amacıyla kullanılmıştır. II. Dünya savaşı esnasında lojistiğin önemi fark edilip, bilimsel konu olarak benimsenip çalışmalar yapılmaya başlanmıştır. Savaş esnasında fark edilen doğrulardan birisi de yüksek sayıda asker sayısı bulundurmanın savaş galibiyeti için yeterli olmayıp; silah, araç ve diğer askeri ihtiyaçların sayısal üstünlükten daha önemli olduğu anlaşılmıştır. Bu kapsamda savaşı etkileyecek faktörlerin zamanında doğru yere ulaştırılması ve uzman personeller tarafından sürekli faal şekilde kullanım sağlanmasında lojistiğin büyük önem taşıdığı görülmüştür. Genel olarak bakıldığında herhangi bir hareket durumunda düzgün lojistik destek sağlanmadığı takdirde başarı beklemenin çok gerçekçi bir durum olmayacağını söylenebilir (Bakkal ve Demir, 2011).

20. yüzyılın başlarında lojistiğin çok önemli olduğu, çok ilgi gördüğü söylenemez. İş alanı olarak lojistik II. Dünya savaşına kadar kullanılmamıştır (Tekin, 2014). Lojistik sektöründeki gelişme II. Dünya savaşı sonrasında dönemlerde artış gösterip hızla devam etmiştir. Lojistik sektörü işletmeler için 1950-1960 yıllarından sonra önem kazanmaya başlamış ve sektördeki büyüme devam etmiştir. İletişim ve bilişim alanlarındaki ilerlemeler 2000’li yıllarda çoğu sektörü etkilemiş, lojistik sektörü de bu durumdan etkilenmiştir (Sarıcan, 2016).

Lojistik sektörü için en önemli problemlerden başlıca olanı gezgin satıcı problemidir. GSP herhangi bir şehirden (düğüm, nokta) başlayarak geriye kalan her şehre 1 kere uğramak şartıyla tüm şehirleri dolaşıp, tekrar başladığı noktaya en kısa mesafede dönmeyi amaçlayan problemidir. Gezgin Satıcı Problemini ilk olarak İngiliz matematikçi Thomas Penyngton Kirkman ve İrlandalı matematikçi Sir William Rowan Hamilton 1800’lü senelerde ele almış sonrasında bu zamana kadar konu üzerinde sayısız çalışmalar yapılmıştır. Tanımı kolay gibi görünen gezgin satıcı probleminin çözülmesi kolay değildir

(Koç, 2007). Şehir (düğüm, nokta) sayısı arttıkça problemin çözümü de gittikçe zorlaşmaktadır. Bu sebepten dolayı üzerinde fazlasıyla çalışma yapılan problemlerden biridir (Yıldırım, 2014).

Literatürde “Çizgi üzerinde, en kısa Hamilton turunun bulunması” gibi tanımlar bulunmaktadır. Hamilton turu diye adlandırılmasının nedeni GSP için ilk çalışmalarını yapmış olan William Rowan Hamilton adından gelmektedir. Burada bahsedilen Hamilton turu çizgi üzerindeki her noktaya uğranılıp, tekrar başladığı noktaya dönen tura denilmektedir. Gezgin satıcı probleminde Hamilton turunun bulunabilmesi için verilen noktalar arasındaki uzaklıklarının bilinmesi gerekmektedir. Burada Hamilton turları içerisindeki en kısa mesafe gezgin satıcı probleminin çözümü olacaktır. Verilerimizdeki düğüm, nokta veya şehir sayıları az miktarda olursa problemi çözümü kısa sürede gerçekleştirmek fakat bu düğüm, nokta veya şehir sayıları arttıkça problemin çözümü zorlaşarak uzun zaman zarfında çözülebilmektedir. Problemde  $[(x-1)!/2]$  Hamilton turu bulunmaktadır. En kısa mesafede bu turların arasında bulunmaktadır. Burada  $x$  problemde belirtilen nokta, düğüm veya şehir sayısını; bölen kısmındaki 2 ise simetrik çözüm bölenidir. Bu formülü de basit örneklerle açıklamak gerekirse 6 şehirli bir problem kapsamında Hamilton turunun sayısı 60 tane, 12 şehirli bir çalışmada Hamilton turunun sayısı 19 958 400 tane, bu çalışma kapsamında kullanılan 81 il için Hamilton turu sayısı  $3,57847285231319 \cdot 10^{18}$  tane olarak bulunmaktadır. Gezgin satıcı probleminin çözümü için birbirinden farklı birçok yöntem, birçok çözüm yolu bulunmaktadır. Yöntemlerin bazıları sezgisel, bazıları ise meta sezgisel yöntemlerdir. Bu çalışmada ise en iyi sonuca ulaşmak için bir meta sezgisel yöntem olan karınca kolonisi optimizasyonu ve benzetilmiş tavlama algoritması yöntemleri kullanılmıştır.

Taşımacılık birçok sektörün ihtiyacı olan önemli bir faktördür. Turizm, ticaret ve sanayi gibi sektörlerde taşımacılık önemli yer almaktadır. Birçok taşımacılık türü mevcuttur. Bunlardan başlıca olanları havayolu taşımacılığı, demiryolu taşımacılığı, denizyolu taşımacılığı ve karayolu taşımacılığıdır. Ülkemizde taşımacılık türleri arasından en çok kullanılan taşımacılık türü karayolu taşımacılığıdır. Karayolu taşımacılığı, belirli bir para miktarı karşılığında bir malzemenin herhangi bir konumdan alınıp, gideceği konuma bırakılması olayının karayolu ile yapılarak taşıyan ve gönderen arasında bir sözleşme oluşturulmasını gerektiren bir taşıma türü şeklinde tanımlanmaktadır (Demir, 2006). Karayolu yük taşımacılığı verilerine bakıldığında 2018 yılında %89,2 2019 yılında %91,9

gibi bir kullanım oranıyla diğer taşımacılık türlerine göre daha çok kullanılmaktadır (Çevre ve Şehircilik Bakanlığı). Ulaşım bağlantılarının geniş olması, transit yollardaki artış miktarı, eşyaların yükleme boşaltma kolaylığı, kapıya kadar hizmet verilebilmesi, hızlı ulaştırma imkanı vb. nedenlerden dolayı karayolu yük taşımacılığının diğer taşımacılık türlerine göre tercih edilmesi artış göstermektedir (Tuzkaya, 2007).

Ülkemizde geçmişten itibaren ulaştırma politikaları incelendiğinde Cumhuriyet ile birlikte ulaştırmanın önemi artarak yüksek miktarda yatırımlar yapılmıştır. Fakat denizyolu ve demiryoluna ayrı bir önem verilerek diğer türlere göre öncelik verilmiştir. 1950’li yıllara gelindiğinde ulaştırma politikaları ile birlikte karayolunun önemi artmıştır (Bulut, 2007). Günümüz zamanına bakıldığında ise yapılan yatırımlar, tüneller, köprüler vb. etkenlerden dolayı karayolu taşımacılığına olan talep artmış; bu durumda taşımacılık türleri arasında daha çok tercih edilmesini sağlamaktadır. Bu çalışmada da 81 ildeki organize sanayi bölgeleri ele alınarak karayolu yük taşımacılığında, MATLAB programı kullanılarak karınca kolonisi optimizasyonu ve benzetilmiş tavlama algoritması yöntemleri ile en kısa yol analizi yapılmaktadır.

MATLAB, ilk olarak Cleve Moler tarafından 1985 yılında yazılmış olan bir programdır. Programın yazıldığı zamanlarda kullandıkları sürümün dili FORTRAN olup sonraki sürümlerinde C, C++ dilleri kullanılmıştır (Bezek, 2009). MATLAB, teknik hesap işlemleri ve matematiksel problemlere çözüm bulmak amacıyla kullanılan bir programdır. MATLAB adını “MATrix ve LABoratory” kelimelerinin ilk 3 harflerinin birleşimi sonucundan almıştır. Çalışma prensibi olarak MATLAB adından da anlaşılacağı gibi matrisleri ya da farklı bir söylemle dizileri (array) kullanmaktadır. Günümüze bakıldığında kullanımı geniş alanlara yayılan MATLAB programı gerek akademisyenler, gerek öğrenciler ve araştırmacılar tarafından yüksek kullanım sayısına sahiptir. MATLAB genellikle mühendislik konularını içeren alanlarda kullanılmaktadır (Özbilge, 2010).

Sürekli gelişen teknoloji ile birlikte kullanılan ekipmanlar da gelişmektedir. Kullanılan programlar, yapılan algoritmalar değişim ve gelişim göstermektedir. Günümüzde oluşan en kısa güzergah problemlerinin çözümü için yeni algoritmalar geliştirilmiş, bazı çalışmalar için doğal hayattaki canlıların hareketleri bir örnek oluşturmuştur. İncelenen canlıların hareketlerinden esinlenilip modeller oluşturularak yeni algoritmalar geliştirilmiştir. Geliştirilen algoritmalarından en çok tercih edilenlerden birisi de KKO’dur (Dorigo, 1992).

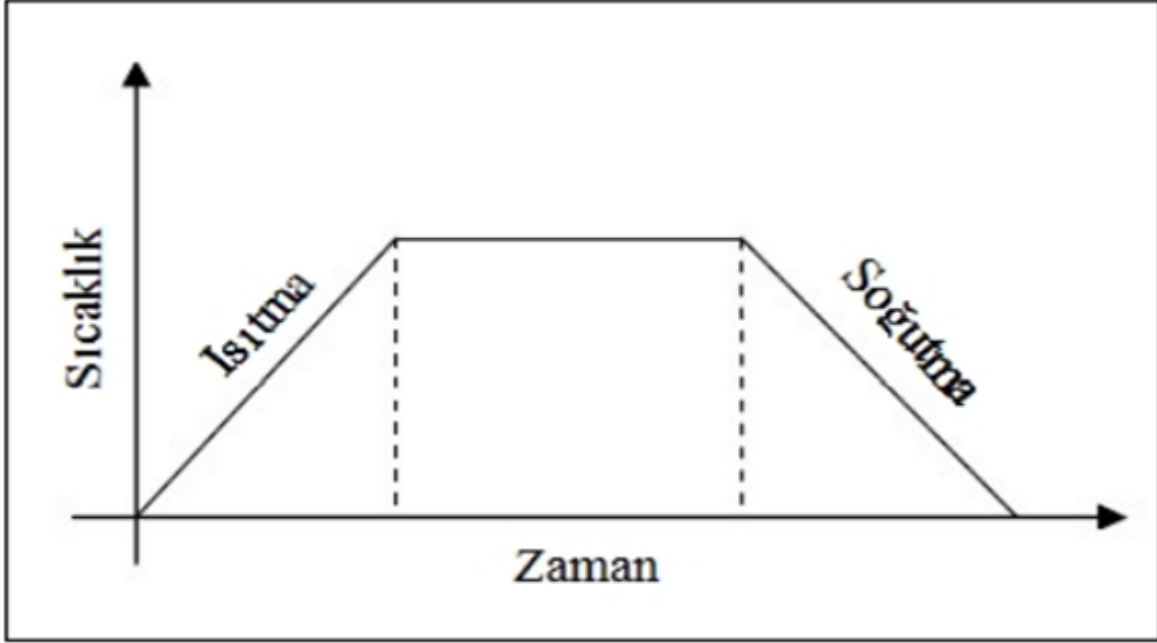


Marco Dorigo, 1992 yılında ilk olarak karınca kolonisi optimizasyonu üzerinde uygulama yapmıştır. Karıncaların hareketleri örnek alınan bu algoritmanın gelişimi Marco Dorigo ile gerçekleşmiştir (Erdoğan, 2008). Karınca kolonisi optimizasyonu için gerçek karıncalar kullanılarak deneyler yapılmıştır. Bu deneyler ise Goss ve arkadaşları tarafından yapılmıştır (Goss, Aron, Deneubourg ve Pasteels, 1989).

Karıncalar yuvaları ile yiyecekleri arasındaki yolları belirlerken ilk olarak rastgele yol güzergahı kullanırlar. Karıncalar geçtikleri yol güzergahlarına salgıladıkları feromon adında bir kimyasal madde bırakırlar (Sim ve Sun, 2003). Bırakılan bu kimyasal feromon maddesi yiyeceği araştıran karıncanın tekrar yuvasına dönmesine olanak sağlayacağı gibi, diğer karıncaların da yuva ile yiyecek arasındaki yolları bulmasını sağlayacaktır. Bırakılan feromon maddesi buharlaşabilen bir madde olup, zamana bağlı olarak buharlaşmaktadır. Karıncaların en kısa yol güzergahını bulma durumu bu buharlaşma miktarları sayesinde olmaktadır. Her yol güzergahına eşit miktarda feromon maddesi bırakılır ve yol güzergahları arasından yollara bırakılan feromon maddesinin daha yoğun olduğu güzergah daha az buharlaştığı için daha kısa yol güzergahı olacaktır. Karıncaların yiyecek olayından esinlenen karınca kolonisi optimizasyonunda gezgin satıcı problemlerinde aynı yiyecek bulma mantığıyla kullanılmaktadır. Problemin çözümünde karıncalar, ilk olarak rastgele seçilen bir şehirden başlarlar ve geçtikleri her şehre tekrar uğramamak şartıyla tüm şehirlere uğrarlar. Geçtikleri her yola da feromon maddesi bırakırlar (Ünal, 2008). Yiyecek ile yuva arasındaki en kısa yol durumunda olduğu gibi buharlaşma miktarının daha az olduğu güzergah en kısa güzergah olarak bulunacaktır.

Benzetilmiş tavlama algoritması, adını metalleri tavlama tanımına gelen tavlama sözcüğünden almıştır. Tavlama genel tanım olarak ısıtma ve soğutma işlemidir. Malzemenin fiziksel ve kimyasal özelliklerinde istenilen değişimi oluşturmak için kullanılan bir yöntemdir. Buna bağlı olarak tavlama işleminin asıl uygulanma durumu cisme ısı verilip daha sonra yavaşça soğutarak istenilen şekle, yapıya, modele ulaşmasını sağlamaktır. Yani tavlama iki basamaktan oluşan bir sistemdir. Önce sistem ısıtılarak sistemin enerjisi yükseltilir ve düzensiz bir yapı içerisinde atomlar serbestçe dağılır. İşlemin devamında istenilen şekil, yapı veya istenilen düzene getirmek için soğutma işlemi gerçekleşir. Bu işlem sonucunda atomların farklı enerji seviyelerine geçmesi sağlanarak daha düzgün, kararlı bir yapı oluşmasına olanak sağlayacaktır. Yapılan bu soğutma

uygulaması sistemdeki enerji minimize edilene kadar devam etmektedir. Cisme uygulanan tavlama işlemindeki ısı değişim grafiği Şekil 1.1’de gösterilmektedir.



Şekil 1.1. Tavlama işlemindeki süreç (Büyüközer, 2019)

BTA temeli Nicholas Metropolis ve arkadaşları tarafından 1953 senesinde atılmıştır. Metropolis’in temellerini attığı bu durum malzemelerin tavlama ile en iyileme problemlerinin çözüm sürecine benzerliğinden gelmektedir. BTA, katı bir malzemenin enerji durumunu minimize eden tavlama işlemi ile en iyileme probleminin çözüm işlemleri arasındaki benzerlikleri ile bağlantılı olarak geliştiren bir algoritmadır. Benzetilmiş tavlama algoritmasının gelişimi 1983 yılında Kirkpatrick ve arkadaşları tarafından yapılmıştır (Söke, 2003).

Genel olarak bakıldığında benzetilmiş tavlama algoritması optimizasyon problemlerinin çözümü için katı malzemelerin tavlama sırasındaki minimum enerji elde etmesi ile olan benzerliğinden geliştirilmiştir. GSP’de kullanılan optimizasyon tekniklerindeki amaç; tavlama sürecindeki enerjiyi minimize etmek olduğu gibi optimizasyon tekniklerinde de aynı şekilde maliyeti, mesafeyi minimize etmek için kullanılmaktadır. Optimizasyon tekniğinde kullanılan değişkenler, sistemdeki atomların dağılımına denk gelmektedir. Sistemdeki ısı, enerji değişimi ile optimizasyon probleminin çözümleri de benzerlik göstermektedir. Yani sistemdeki olan veya olabilecek işlemler ile problemlerdeki olan veya oluşabilecek çözümlerin çakıştığını göstermektedir. Optimizasyon problemlerinde kontrol işlevi olarak

kullanılan sıcaklık; tavlama istenilen sonuca ulaşma durumunu belirlemek için kullanılmaktadır (Büyüközer, 2019). Benzetilmiş tavlama algoritması ile ilgili formüller ve açıklamaları ilerleyen sayfalarda verilmektedir.

Lineer programlama birçok iş alanını yakından ilgilendirmektedir. Maliyeti azaltmaya çalışan lineer algoritma, aynı zamanda en yüksek miktarda kar elde etmeyi amaçlamaktadır. Problemler için Lineer optimizasyon önemli bir çözüm yöntemi olmaktadır. George Dantzig, 1947 yılında problemlerin çözüm sürecinde önemli rol oynayan Simpleks algoritmasını bulmuştur ve bulunan bu algoritma sonucunda lineer programlama çoğu sektörlerde kullanılan önemli bir program olmuştur (Bozkurt, 2019).

Parçacıklı sürü optimizasyonu 1995-1996 senelerinde Kennedy ve Eberhart tarafından ileri sürülmüştür. Bazı hayvan sürülerinin davranışlarını örnek alarak geliştiren bir yöntemdir. Örnek alınan hayvan sürüleri ise balık ve kuş sürüleridir (Eberhart ve Kennedy, 1995). Parçacıklı sürü optimizasyonunun temelini bu kuş ve balık sürülerinin aralarında iletişim kurarak yiyecek ararken olan davranışları oluşturmaktadır. Kuş sürülerinden örnek verecek olursak; kuşların her biri sistemde parçacık olarak adlandırılmaktadır. Tüm parçacıkların sistemde serbest olarak dağılımıyla başlamaktadır. Sistem üzerindeki adım sayısına göre de parçacıkların durumu, konumu güncellenmektedir. Belirli hızla hareketlerini devam ettiren parçacıkların aramalarını devam ettirerek en ideal sonucu bulmaya çalışmasına dayanan bir algoritmadır (Zhang, Zhang, Lok ve Lyu, 2007).

Genetik algoritma, John Holland tarafından 1960 yıllarında öne sürülmüş olup sonraki yıllarda gelişimini sürdürmeye devam etmiştir (Mitchell, 1998). GA, çözüme ulaşmak amacıyla evrim sürecini bilgisayar ortamında taklit eden bir yöntemdir. Problemlere çözüm üreten diğer yöntemlere benzer olarak yapılardan oluşan bir kümeleme yaparlar. Bu küme yöntemi kapsamında nüfus olarak tanımlanmaktadır. Birey ya da kromozom olarak isimlendirilen parametreler nüfusu oluşturmaktadır. Kromozom ya da birey olarak adlandırılan bu parametrelerin içerisindeki elemanlara da gen adı verilmektedir. Evrim sürecinde algoritmadaki işlemciler tarafından kromozomlar belirlenir. Çözüm sürecinde genetik algoritmaların problemler için birçok çözüm üreterek çözüm kümesi oluşturur. Bu sayede aynı zaman diliminde farklı noktaların değerlendirilmesi yapıp bütünsel çözüme ulaşma durumu artmaktadır (Gündoğdu, 2020).

Tabu arama algoritmasının geliřimi Glover (1986) tarafından yapılmıřtır. Diđer problem çözümlerinde de kullanılan bu tabu arama algoritması en ideal çözümler veya en ideal çözüme yakın sonuçlar elde edebilmek amacıyla esnek yapıdadır. Tabu arama algoritması bir çözüml üretir ve dönüşüm tekniklerini kullanarak en ideal çözümlü elde etmeye çalıřır (Chelouah ve Siarry, 2000). Bu algoritmada optimum çözümlü elde etmek için kullanılan dönüşüm teknikleri; elemanları ekleyip çıkarma, deęişkenin deęeri üzerinde yapılan deęişiklik, yapılan iki farklı çözümlü deęerlerinin birbiri ile deęişiklięi olarak söylenebilir. Buna baęlı olarak esnek olan yapısı sayesinde tabu arama algoritmasının farklı çözümler elde etmek ve en ideal çözümlü yaklařmak için bu algoritmayı etkin duruma ulařtırmaktadır. Bu özellikler çerçevesinde TA algoritması farklı problemlerde de kullanılabilir (Gürbüz, 2015).

## 2. ÖNCEKİ ÇALIŞMALAR

Serin (2009) yaptığı çalışmada, karınca kolonisi optimizasyonu yöntemini kullanarak üstyapı yönetim kavramı için bir optimizasyon modeli geliştirmiştir. Üstyapı yönetim kavramı, inşa edilen karayolu üstyapıların artması sonucu daha fazla kaynağın inşa edilen üstyapıların bakım ve iyileştirmesinde kullanılması sonucu ortaya çıkan bir planlamadır. Yapılan çalışma bu kavram çerçevesinde belirlenen bakım ihtiyacı gereksinimleri, bütçe tahsisi, insan gücü, malzeme ve teçhizat kullanılabilirliği gibi girdilerin en iyi şekilde bir araya getirerek maksimum verim alınmaya çalışılmıştır. Karınca kolonisi optimizasyonu yöntemi kullanılarak yapılan çalışmada bu girdiler baz alınarak modeller geliştirilmiş ve kaynakların verimli kullanılması sağlanmıştır.

Pamukçu (2022) yaptığı çalışmada, karınca kolonisi optimizasyonu yöntemini kullanarak gezgin satıcı problemine örnek bir uygulama ile çözüm getirmiş ve bu çözümü farklı yöntemler ile karşılaştırmıştır. Yapılan çalışma Diyarbakır Kayapınar ilçesindeki barış mahallesinde yer alan atıkların toplanması ve taşınması işleminin optimizasyonu yapılarak daha iyi bir sonuç elde edilmeye çalışılmıştır. Yapılan çalışmada coğrafi bilgi sistemi, genetik algoritma ve karınca kolonisi optimizasyonu yöntemi kullanılmıştır. Yapılan çalışma sonucunda gerçek hayat verisi ile karşılaştırıldığında coğrafi bilgi sistemi %15,1576, genetik algoritma %29,8104 ve karınca kolonisi optimizasyonu ise %40,5171 bir iyileşme sağlamıştır. Sonuçlara bakıldığında karınca kolonisi optimizasyonu yöntemi diğer yöntemlere göre daha iyi bir sonuç verdiği gözlemlenmiştir.

Özalp (2020) yaptığı çalışmada, karınca kolonisi optimizasyonu yöntemi ve genetik algoritma yöntemini beraber kullanarak örnek bir uygulama ile gezgin satıcı problemine çözüm getirmiştir. Yapılan çalışma bir kargo firmasına ait 5 aracın bir güne ait 605 kargonun dağıtım işleri baz alınarak yapılmıştır. Yapılan çalışmada ilk olarak karınca kolonisi optimizasyonu yöntemi ile kargoların dağıtım rotası belirlenmiş, daha sonra genetik algoritma yöntemi ile çaprazlama yapılmıştır. Çaprazlama sonucu atanan kargolara tekrar karınca kolonisi optimizasyonu yöntemi ile yeni rotalar bulunmuştur. Yapılan çalışma sonucunda karınca sayısını arttırmak çözümü iyileştirmediği gözlemlenmiş ve bunun sonucunda karınca sayısı azaltılarak başlangıç çözümünden %105,69 daha iyi bir sonuca ulaşılmıştır. Yapılan melez çözümün ise iyi bir sonuç verdiği gözlemlenmiştir.

Topal (2015) yaptığı çalışmada, karınca kolonisi optimizasyonu ve yapay arı kolonisi algoritması yöntemlerini kullanarak afet lojistik sistemleri için bir model geliştirmiştir. Yapılan çalışmada, afet bölgesine en kısa ve en hızlı şekilde nasıl gidileceği ve afet yardım malzemelerinin hangi bölgeden ne kadar alınacağı sorularına cevap vermiştir. Yapılan çalışma sonucunda karınca kolonisi optimizasyonu yönteminin yapay arı kolonisi algoritması yöntemine göre daha iyi bir sonuç verdiği gözlemlenmiştir. Bunun sebebi olarak karınca kolonisi optimizasyonu yönteminin yerel minimum noktalarda takılmayarak sonucu sürekli olarak iyileştirdiği vurgulanmıştır. Bu çalışma sayesinde afet bölgesine hızlı ve doğru şekilde müdahale edilerek can ve mal kayıplarının en aza indirgenmesi amaçlanmıştır.

Özkök (2013) yaptığı çalışmada, kan bankasının günlük ve haftalık sevkiyat rotalarının belirlenmesinde benzetilmiş tavlama algoritması yöntemini kullanarak model geliştirmiştir. Karışık tam sayılı doğrusal olmayan programlama modelinin kullanıldığı çalışmada optimum yerel kan bankası sayısı ve yerleri, ana kan bankasından yerel kan bankasına ve yerel kan bankasından hastanelere yapılacak olan sevkiyat rotaları gibi karmaşık bir problem çözülmeye çalışılmıştır. Problemin çözümünde benzetilmiş tavlama algoritması yöntemi kullanılmış ve analiz edilmiştir. Yapılan çalışma sonucunda sevkiyat rotaları en hızlı ve en uygun şekilde oluşturulmuş, yerel kan bankalarının sayısı ve yerleri belirlenmiştir.

Çelebi (2021) yaptığı çalışmada, gezgin satıcı problemine örnek bir uygulama ile çözüm getirmiştir. Yapılan uygulama 81 ildeki organize sanayi bölgeleri arasında yük taşımacılığını hem tek türlü hem de çok türlü taşımacılık ile en kısa rota seçilerek yapılmıştır. En kısa rota analizi tabu arama algoritması yöntemi kullanılarak yapılmış ve diğer teknikler ile karşılaştırılmıştır. Yapılan çalışma sonucunda tabu arama algoritması yöntemi en kısa rotayı tek türlü taşımacılık ile 10277 km, çok türlü taşımacılık ile 10187 km bulmuştur. Diğer teknikler ile karşılaştırıldığında ise tabu arama algoritması lineer optimizasyona göre kötü, genetik algoritma ve parçacıklı sürü optimizasyonu yöntemine göre ise daha iyi sonuç verdiği görülmüştür.

Gündoğdu (2020) yaptığı çalışmada, taşımacılık maliyetlerini azaltmaya yönelik bir uygulama gerçekleştirmiştir. Yapılan uygulama gezgin satıcı problemini temel alarak ülkemizde bulunan 81 ildeki organize sanayi bölgeleri arasında en kısa güzergah üzerinde

yük taşımacılığı yapılarak gerçekleştirilmiştir. En kısa güzergah analizi genetik algoritma, lineer optimizasyon ve parçacıklı sürü optimizasyonu yöntemleri kullanılarak yapılmıştır. Yapılan çalışma sonucunda optimum mesafeyi lineer optimizasyon ile 9923 km, genetik algoritma ile 10689 km ve parçacıklı sürü optimizasyonu ile ise 11311 km bulunmuştur. Çalışma sonucunda en iyi sonucun lineer optimizasyon yönteminin verdiği vurgulanmıştır.

Açıl (2008) yaptığı çalışmada, gezgin satıcı probleminin farklı uzantıları olduğunda bahsetmiş ve çalışmasında karlı tur problemini ele almıştır. Karlı tur probleminin gezgin satıcı probleminden farklı olarak tüm şehirleri gezme zorunluluğu olmadığını ve temel amacın karı maksimum tutarken yol maliyetini minimum tutmak olduğunu belirtmiştir. Karlı tur problemine daha önce karınca kolonisi algoritmasının uygulanmadığını belirtmiş ve çalışmasında karınca kolonisi algoritması kullanmıştır. Hibrit karınca kolonisi algoritmasının kullanıldığı çalışmada farklı modeller oluşturulmuş ve algoritma bu modellere uyarlanmıştır. Yapılan çalışma sonucunda karlı tur probleminin çözümünde karınca kolonisi algoritması ile çözülebildiği belirtilmiş ve iyi sonuçlar elde edilmiştir.

Özbay (2017) yaptığı çalışmada, zaman pencereli çoklu gezgin satıcı problemine yönelik bir çözüm geliştirmiştir. Bu çözüm evde bakım hizmetleri kapsamında günlük hasta ziyaretleri için çizelgeler ve rotalar belirlenerek yapılmıştır. Çözüm belirlenirken temel olarak hasta ziyaretini maksimum, hemşirelerin bekleme süresini ve toplam rotayı minimum yapmak amaçlanmıştır. Zaman pencereli çoklu gezgin satıcı problemini çözerken değişken komşuluk arama algoritması kullanılmıştır. Yapılan çalışma sonucunda algoritmanın gayet iyi sonuçlar verdiği gözlemlenmiştir.

Ahmadpur (2016) yaptığı çalışmada, trafik ışıklarının sinyal zamanlaması üzerine model geliştirmiştir. Çalışmanın ve modelin temel amacı trafik işaretlerini verimli kullanarak trafik sıkışıklığını azaltmaktır. Çalışmada genetik algoritma, benzetilmiş tavlama algoritması ve tepe tırmanan algoritması yöntemleri kullanılarak karşılaştırma yapılmıştır. Yapılan çalışma sonucunda genetik algoritma diğer algoritmalara göre arama süreci daha fazla olmasına karşın en optimum sonucu veren algoritmadır. Çalışma sonucunda trafik sıkışıklığı azaltılarak hem süreden hem de yakıttan tasarruf edilecektir.

Tafida (2014) yaptığı çalışmada, betonarme bir köprünün yapım maliyetini düşürmek için benzetilmiş tavlama algoritması yöntemini kullanmıştır. Köprünün maliyetini düşürmek

için uygun köprü açıklığı, açıklık sayısı, köprü ayak boyutları, köprü ayak sayısı gibi temel elemanların optimum şekilde belirlenmesi gerekmektedir. Bu temel elemanların optimum şekilde belirlenmesinde matlab programı üzerinde benzetilmiş tavlama algoritması yöntemi kullanılarak bir model geliştirmiştir. Yapılan çalışma sonucunda benzetilmiş tavlama algoritması optimum maliyet ve tasarım için en uygun modeli bulmuştur ve iyi sonuçlar alınmıştır.

Ekşioğlu (2022) yaptığı çalışmada, dijital dönüşümünün lojistik üzerindeki etkileri üzerine araştırma yapmıştır. Yapılan çalışmada lojistik işletmelerinin performansını arttırmak, müşteri memnuniyetini arttırmak, müşterilerin ihtiyaç ve beklentilerine daha hızlı cevap verebilmek ve maliyetleri düşürmek için dijital dönüşümün etkilerini araştırmıştır. Çalışmanın diğer bir amacı ise dijital dönüşümün lojistik işletmelerinin performansına etkisi ve müşterilerin tekrar satın alma niyetini ölçmüştür. Yapılan çalışma ülkemizde lojistik hizmet sağlayan 36 işletme ve bu işletmelerin müşterisi olan 37 işletme üzerinde anket çalışması yaparak gerçekleştirilmiştir. Yapılan çalışma sonucunda dijital dönüşümün işletme performansı ve tekrar satın alma niyetinde olumlu etkilerinin olduğu vurgulanmıştır.

Reda, Onsi, Elhosseini, Mahmoud ve Badawy (2022) yaptığı çalışmada, gezgin satıcı problemi temelli bir problem olan sipariş toplama rotası problemine çözüm getirmiştir. Problemin çözümünde meta sezgisel optimizasyon yöntemi olan guguk kuşu arama algoritması kullanılmıştır. Mevcut algoritma değişiklikler yapılarak yeni varyantlar geliştirilmiş ve problemler üzerinde denenmiştir. Yapılan çalışma sonucunda gayet iyi sonuçlar verdiği gözlemlenmiştir. Ayrıca genetik algoritma, parçacıklı sürü optimizasyonu ve karınca kolonisi optimizasyonu yöntemleri ile karşılaştırmalar yapılmış ve bu algoritmalara göre çok daha iyi sonuç verdiği gözlemlenmiştir.

Lu, Kuo, Ho ve Nguyen (2022) yaptığı çalışmada, son teslimat sürelerini iyileştirmek için esnek drone gezgin satıcı problemi modelini kullanmıştır. Bu modeli çözmek için ise k-ortalamlarını ikiye bölme ve meta sezgisel yöntemlerini formülize edilerek bir yöntem geliştirmişlerdir. Yapılan çalışma sonucunda bu yöntem %7,13 ile %24,17 arasında zaman tasarrufu ve %13,01 ile %19,77 arasında hizmet seviyesini iyileştirdiği gözlemlenmiştir. Ayrıca drone kullanılarak yapılan teslimatlar da müşteri sayısını %56,47'ye kadar arttırdığı



gözlenmiştir. Yapılan çalışmanın son derece iyi sonuçlar verdiği ve drone özelliklerine göre de daha iyi sonuçlar vereceği vurgulanmıştır.

Panwar ve Deep (2021) yaptığı çalışmada, ayrık gezgin satıcı problemine gri kurt optimizasyonu ile MATLAB üzerinde çözüm geliştirmiştir. Gri kurt optimizasyonu, gri kurtların hayatta kalmak için yaptığı davranışları inceleyerek ortaya çıkan meta sezgisel bir algoritmadır. Yapılan çalışmada gri kurt optimizasyonuna 2-opt optimizasyonu dahil edilerek performansı artırılmıştır. Yapılan çalışmanın sonuçları yarasa algoritması ve ateşböceği algoritması gibi algoritmalar ile karşılaştırılmıştır. Karşılaştırma ortalama ve standart sapma gibi tanımlayıcı istatistikler kullanıldığı gibi Friedman ve Holm testleri gibi istatistiksel testler de kullanılmıştır. Yapılan karşılaştırma sonucunda gri kurt optimizasyonunun diğer algoritmalara göre daha iyi performans gösterdiği gözlemlenmiştir.

Zhang ve Han (2022) yaptığı çalışmada, gezgin satıcı problemine bir sürü zekası yaklaşımı olan serçe arama algoritmasını kullanarak çözüm getirmiştir. Başlangıç çözümü olarak popülasyondaki rulet tekerleği ile seçim yapılır. Daha sonra sıraya dayalı kod çözümü ile serçe konumunun güncellenmesi tamamlanır. Son olarak da çözümün kalitesini arttırmak için 2 seçeneğe yerel arama eklenir ve çözüm tamamlanır. Yapılan çalışma bir kıyas veri seti üzerinde diğer yöntemler ile karşılaştırılmıştır. İstatistiksel testler kullanarak da karşılaştırma yapılan çalışma sonucunda serçe arama algoritması diğer yöntemlere göre daha rekabetçi ve sağlam olduğu gözlemlenmiştir.

Stodola, Otrisal ve Hasilova (2022) yaptığı çalışmada, gezgin satıcı problemine karınca kolonisi optimizasyonu yöntemi kullanarak çözüm getirmiştir. Yapılan çalışmanın amacı olarak algoritmanın genel performansını arttırmak, yürütme süresini azaltmak ve yerel bir optimuma düşmemesini sağlamaktır. Bunun amaca ulaşmak için düğüm kümeleme, çeşitliliğe dayalı sonlandırma koşulu formülasyonu ve kontrol edilebilir uyarlanabilir feromon buharlaşması gibi teknikler kullanılmıştır. Geliştirilen algoritma bir dizi örnekler üzerinde test edilmiş ve diğer yöntemler ile karşılaştırma yapılmıştır. Yapılan çalışma sonucunda geliştirilen algoritma daha iyi performans gösterdiği gözlemlenmiştir.

Kanna, Sivakumar ve Lingaraj (2021) yaptığı çalışmada, gezgin satıcı problemine hibrit bir yaklaşımla çözüm getirmiştir. Gezgin satıcı probleminin hala karmaşık bir optimizasyon

problemi olduğuna dikkat çekilen çalışmada hibrit bir yaklaşımla problemi çözmeye çalışmıştır. Geyik avı optimizasyonu ve solucan optimizasyonu algoritmaları birbirlerine entegre edilerek hibrit bir yaklaşım oluşturmuştur. Yapılan çalışmanın amacı olarak da satıcının tüm şehirleri için kat ettiği mesafeyi en aza indirmek olduğu belirtilmiştir. Yapılan çalışma sonucunda hibrit yaklaşımının gezgin satıcı problemini çözerken daha az hesaplama ile çözüme gittiği gayet iyi sonuçlar verdiği gözlenmiştir.

Al-Amyal, Szamel ve Hamouda (2022) yaptığı çalışmada, anahtarlamalı relüktans motor sürücülerindeki anlık tork kontrolünü karınca kolonisi optimizasyonu yöntemi ile kontrol etmeye çalışmışlardır. Kullanılan yöntem karınca kolonisi optimizasyonu yönteminin gelişmiş bir versiyonu olan çok aşamalı karınca kolonisi optimizasyonu yöntemidir. Bu gelişmiş yöntemle birlikte arama yeteneği gelişmiş ve belirlenmiş kısıtlamaları ihlal etmeden en iyi sonucu gitmeye çalışmıştır. Çalışmanın temel amacı, tork dalgalanmasını azaltmak, motor verimliliğini arttırmak ve negatif torku önlemeye yöneliktir. Yapılan deneysel çalışmalar sonucunda gelişmiş yöntemin gayet iyi sonuç verdiği ve geleneksel yöntemlere göre daha iyi performans gösterdiği gözlemlenmiştir.

Xu, Huang, Liu, Li ve Chen (2022) yaptığı çalışmada, yangın durumunda planlanan tahliye yollarının gelişmiş karınca kolonisi optimizasyonu yöntemi ile daha akıllı bir şekilde yapılmaya çalışılmıştır. Yangında tahliye yollarının sabit olduğu ve anlık duruma göre değişmediği için zaman ve can kayıplarının yaşandığını belirtilen çalışmada, gelişmiş karınca kolonisi optimizasyonu yöntemi ile tahliye yollarının anlık durumlara göre değiştirdiğini ve bununla birlikte tahliye zamanını azalttığı belirtilmiştir. Sıcaklık ve CO gazlarının bir simülasyon yazılımı ile anlık olarak takip edilerek tahliye yollarının anlık olarak değişmesini sağlamıştır. Gelişmiş karınca kolonisi optimizasyonu ile tahliye yolları belirlenen çalışmada, geleneksel yöntem ve ticari bir yazılım tarafından planlanan tahliye yolları ile karşılaştırıldığında gelişmiş yöntemin daha iyi sonuç verdiği gözlemlenmiştir. Bu çalışma ile birlikte başarısız tahliye yolları azaltılır ve can ve mal kayıpları daha aza indirilebilir vurgusu yapılmıştır.

Skinderowicz (2022) yaptığı çalışmada, karınca kolonisi optimizasyonuna yeni bir varyant geliştirerek gezgin satıcı problemine bir çözüm getirmiştir. Karınca kolonisi optimizasyonun zor optimizasyon problemlerine karşı çözüm getirdiğini ama özellikle temel sorunlarda yavaş kaldığını belirtmiştir. Odaklanmış karınca kolonisi optimizasyonu

isimli yeni varyant, mevcut çözümün kalitesini korurken iyileştirme yapmaya devam etmesi ve özel yerel arama ile çok daha verimli bir optimizasyon haline gelmiştir. Bir örnek uygulama ile odaklanmış karınca kolonisi optimizasyonu ile karınca kolonisi optimizasyonunu karşılaştıran çalışma, odaklanmış karınca kolonisi optimizasyonun çok daha iyi performans gösterdiği ve daha az süreyle işlemi bitirdiği vurgulanmıştır.

Guo, Qian, Na, Hu ve Mao (2022) yaptığı çalışmada, üç boyutlu bir karınca kolonisi optimizasyonu kullanarak çok bölmeli araç rotalama problemine çözüm getirmiştir. Çok bölmeli araç rotalama probleminin birçok sektörde ortaya çıktığı vurgulanan çalışmada karbon emisyonları da dahil olmak üzere toplam taşıma maliyetlerinin azaltılması amaçlanmıştır. Geleneksel karınca kolonisi optimizasyonunda farklı olarak feromon konsantrasyon matrisini başlatmak ve ilk koloniyi oluşturmak için 5 sezgisel kural ve arama stratejisini hızlandırmak için uyarlanabilir yerel arama tasarlanmıştır. 3 boyutlu karınca kolonisi optimizasyonun kullanıldığı çalışmada gayet iyi sonuçlar elde edilmiş ve diğer yöntemler ile karşılaştırıldığında daha iyi performans verdiği görülmüştür.

Deng ve diğerleri (2022) yaptığı çalışmada, havaalanlarında taksi yollarının planlanması için bir hibrit optimizasyonu geliştirmiştir. Çok stratejili parçacık sürüsü algoritması ve karınca kolonisi optimizasyonunda oluşan hibrit optimizasyon, havaalanı taksi yollarının çakışmalarını önleme, taksi gelme süresinin azaltılması ve ilk gelen ilk hizmet fikrine dayalı bir strateji üzerine çözüm getirmiştir. İlk olarak matematiksel model oluşturulmuş daha sonra ise yeni bir feromon başlatma yaklaşımı ile model geliştirilmiştir. Geliştirilen hibrit optimizasyon, farklı modellerdeki 10 gezgin satıcı problemi üzerinde ve gerçek bir havaalanı taksi yolu planlama problemi üzerinde deneyleri yapılmıştır. Yapılan çalışma sonucunda gezgin satıcı problemini etkin bir şekilde çözümlendiği ve diğer yöntemlere göre daha iyi bir performans gösterdiği vurgulanmıştır.

İlhan (2021) yaptığı çalışmada, kapasiteli araç rotalama problemine benzetilmiş tavlama algoritması yöntemi ile çözüm getirmiştir. Kapasiteli araç rotalama probleminin, günümüzde yaygın olarak çalışıldığını ve NP-zor problemler arasında yer aldığını belirtmiştir. Bu problemin çözümüne ise popülasyon tabanlı benzetilmiş tavlama algoritması yöntemi kullanmıştır. Popülasyondaki çözümler yerel arama operatörleri ile üretilmiş, daha sonra ise 2-opt algoritması ile bu çözümler geliştirilmiştir. Kısmi eşlemeli çaprazlama ve sıralı çaprazlama operatörleri ise çözümlere uygulanmıştır. Yapılan

çalışmanın sonucu 91 kıyaslama örneği ile test edilmiş ve gayet iyi sonuçlar verdiği gözlemlenmiştir.

Hanafiah ve diğerleri (2019) yaptığı çalışmada, gezgin satıcı problemine benzer bir probleme geliştirilmiş benzetilmiş tavlama algoritması yöntemi ile çözüm getirmiştir. Bilgiye ulaşmanın kolaylığı ile beraber seyahat planlamaları önceden yapılı hale gelmekte ve ziyaret edilecek noktalar belirlenmektedir. Ziyaret edilecek yerlerin sırası ise mesafe, konaklanacak yer ve yemek yerleri gibi faktörler ile belirlenmekte ve bir güzergah oluşturulmaktadır. Yapılan çalışmada bu güzergah benzetilmiş tavlama algoritması ve greedy algoritması kombinasyonu ile belirlenmiştir. Geliştirilmiş benzetilmiş tavlama algoritması yöntemi ile başlangıç çözümü oluşturulmuş, daha sonra ise greedy algoritmasının arama tekniği ile bu çözüm geliştirilmiştir. Yapılan çalışma sonucunda uygulanan yöntem, algoritmanın etkinliğini arttırdığı vurgulanmıştır.

Zidi, Omani ve Aldhafeeri (2019) yaptığı çalışmada, ambulans yönlendirme problemine hibrit bir yaklaşımla çözüm getirmiştir. Hibrit yaklaşımı, simüle edilmiş benzetilmiş tavlama algoritması ve tabu arama algoritması oluşturmaktadır. Yapılan çalışmanın amacı bir veya birden fazla afet durumunda etkin bir yönlendirme ile en kısa çözüme gitmek olarak vurgulanmıştır. Kullanılan hibrit yaklaşım ayrıca k-ortalamlar algoritması ile desteklenmiştir. Yapılan çalışma sonucu deneysel olarak test edilmiştir. Yapılan çalışma sonucunda iyi sonuçlar alınmıştır. Ayrıca deney sonuçları, parçacıklı sürü optimizasyonu ve genetik algoritma yöntemleri ile karşılaştırılmış ve sonuçların rekabetçi olduğu vurgulanmıştır.

Özkan (2021) yaptığı çalışmada, orman yangını risklerini azaltmak için insansız hava araçlarını kullanmak ve yönlendirmek için hibrit bir algoritma geliştirmiştir. Simüle edilmiş benzetilmiş tavlama, yerel arama meta sezgiselleri ve matematiksel algoritma hibrit algoritmayı oluşturmuştur. Yapılan hibrit algoritma gerçek bir vaka çalışması ile test edilmiş ve genetik algoritma ile karşılaştırılmıştır. Vaka çalışmasına örnek olarak ise, meteoroloji genel müdürlüğüne ait veriler kullanılmıştır. Bu verileri, müdürlük tarafından yayınlanan sonraki 3 gün içerisindeki orman yangını oluşabilecek riskli bölgeler haritası oluşturmaktadır. Haritalar üzerinde insansız hava araçlarının ziyaret edeceği bölgeler, ziyaret edeceği bölgelerin sırası ve kalkış-iniş yapacağı yerler geliştirilen hibrit

algoritmayla belirlenmiştir. Yapılan çalışma sonucunda gayet iyi sonuçlar alınmış ve sürdürülebilir olduğu vurgulanmıştır.

Zhang ve Xiong (2018) yaptığı çalışmada, tahıl dağıtım araçlarının rota optimizasyonu problemini hibrit bir yaklaşım ile çözmeye çalışmıştır. Tahıl talep noktalarındaki ihtiyaçların karşılanmasını en üst düzeye çıkarmak, dağıtım maliyetlerini düşürmek ve dağıtım süresini azaltmak şeklinde 3 ana hedef belirlenmiştir. Problem çözümünde kullanılan hibrit yaklaşım ise karınca kolonisi optimizasyonu algoritması ve yapay bağışıklık algoritmasından oluşmaktadır. Karınca kolonisi optimizasyonu algoritmasının arama yeteneği ve geri bildirim, yapay bağışıklık algoritmasının rastgeleliğinden yararlanılarak çözüm setleri üretilmiştir. Yapılan çalışma diğer algoritmalar ile karşılaştırılmış ve gayet iyi sonuçlar elde edildiği görülmüştür.

Kaabachi, Jriji, Madany ve Krichen (2017) yaptığı çalışmada, zaman pencereli yeşil gezgin satıcı problemine çözüm getirmektedir. Yapılan çalışmanın amacı yakıt tüketimini düşürmek, emisyon gazlarını azaltmak ve en kısa mesafede hedefe ulaşmak olarak belirlenmiştir. Problemin çözümü için iki kriterli karınca kolonisi optimizasyonu algoritması geliştirilmiş ve probleme uygulanmıştır. Geliştirilen algoritma yerel arama çözümü ile başlamış daha sonra ise çok sayıda komşuluk gerçekleştirerek çözüme ulaşmıştır. Yapılan çalışma Tunus'un güneyindeki Gafsa şehrinde gerçek bir olayla test edilmiştir. Yapılan çalışma sonucunda gayet iyi sonuçlar elde edildiği vurgulanmıştır.

Yang, Dong ve Yao (2017) yaptığı çalışmada, karınca kolonisi optimizasyonu algoritmasını kullanarak metrolarda yolcu dağılımı modellemesi yapmıştır. Geliştirilen model bekleme alanına olan mesafe, kuyruk uzunluğu, bekleme alanı uzunluğu ve tren seferleri olmak üzere 4 ana faktör üzerinde geliştirilmiştir. Geliştirilen modelin amacı acil durumlarda tahliye kolaylaştırmak ve metrolarda olan yoğunluğu modellemektir. Geliştirilen model, bir metrodan alınan verilerle test edilmiştir. Yapılan çalışmalar sonucunda acil durumun büyüklüğüne göre yolcu modellemesinin önemli olduğunu ve yolcu dağılımının incelenmesi gerektiği vurgulanmıştır.

Yogaswara ve Fatin (2019) yaptığı çalışmada, araç rotalama problemine tabu arama algoritması kullanarak çözüm getirmiştir. Yapılan çalışma, özel bir firmaya ait atık nakliyesi üzerinde yapılmıştır. Firmaya ait nakliye rotası çalışma öncesi 564,30 km olarak

belirlenmiştir. Tabu arama algoritması kullanarak nakliyede kullanılan yollar optimize edilmiş ve yeni rota oluşturulmuştur. Oluşturulan rota sonucunda yeni nakliye rotası 448,48 km olmuştur. Yapılan çalışmanın gayet iyi sonuç verdiği, 115,82 km'lik düşüş elde edildiği ve toplam mesafeden %20,53'lük tasarruf edildiği vurgulanmıştır.

Du ve He (2012) yaptığı çalışmada, araç rotalama problemi için hibrit bir algoritma ile çözüm getirmiştir. Çalışmanın amacı, coğrafi olarak büyük bir alanda hizmet veren filoların en düşük maliyetle hizmet vermesi gösterilmiştir. Geliştirilen hibrit algoritma en yakın komşu arama ve tabu arama algoritmalarından oluşmaktadır. Geliştirilen algoritma ilk olarak en yakın komşu arama algoritması ile ilk rotaları oluşturmakta daha sonra ise tabu arama algoritması ile rota içi ve rotalar arası mesafeyi optimize etmektedir. Yapılan çalışma sonucu 6772 müşteri yer alan veri setine uygulanmış ve gayet iyi sonuçlar alındığı vurgulanmıştır.

Huang, Blazquez, Huang, Paredes-Belmar and Latorre-Nuñez (2019) yaptığı çalışmada, araç rotalama probleminin farklı bir çeşidi olan besleyici araç rotalama problemini incelemiştir. Büyük araç (kamyon) ve küçük araç (motosiklet) kullanılarak yapılan çalışmada karınca kolonisi optimizasyonu yöntemi kullanılmıştır. Bu algoritmayı maliyeti minimize etmek ve en ideal yol güzergahını bulmak için kullanmıştır. Besleyici araç rotalama probleminin performans durumu için 3 veri seti oluşturulmuştur. Bu kapsamda KKO küçük örnekler için en uygun çözümler ile karşılaştırılmış, KKO yönteminin özellikle büyük çözümlerde kısa zamanda iyi ve umut veren sonuçlar elde ettiği belirtilmiştir.

Gao (2020) yaptığı çalışmada, gezgin satıcı probleminin çözümü için karınca kolonisi optimizasyonu yöntemini kullanmıştır. Geleneksel karınca kolonisi optimizasyonunun negatif yönlerini belirtmiş ve karıncaların çalışma alanları genişletilerek aynı zamanda oluşan çözümleri de çeşitlendirerek farklı bir karınca kolonisi optimizasyonu önerilmiştir. Yeni algoritmanın 20 tipik GSP'ye uygulanarak daha iyi sonuçlar elde edildiği belirtilmiştir. Ayrıca bu algoritmanın 16 algoritma ile karşılaştırıldığında elde edilen sonuçlar kapsamında algoritmanın birçok algoritmadan daha iyi performans sergilediği belirtilmiştir.

### 3. MATERYAL ve YÖNTEM

Bu tez çalışmasında ülkemizde 81 ilde bulunan her ilden bir tane olmak üzere organize sanayi bölgeleri belirlenmiştir. Sonrasında belirlenen OSB koordinatları bulunmuştur. Bu organize sanayi bölgelerinin koordinatları ile çözümü için GSP’de öklid bağıntısı kullanılarak karınca kolonisi optimizasyonu ve benzetilmiş tavlama algoritması yöntemleri ile çözüm algoritmaları oluşturulmuştur. Sonrasında eklenen koordinat değerleri yerine, gerçek mesafeler ile çözüm yapılabilmesi için OSB’ler arasındaki gerçek mesafe değerleri algoritma verilerine eklenmiştir. En ideal mesafe analizleri bu gerçek değerlere göre yapılmış, en ideal mesafenin güzergahı ise organize sanayi bölgelerinin koordinat değerlerine yazılan algoritma ile çizimi yapılmıştır. Bulunan en ideal sonuçlar farklı çalışmalarda yapılmış olan LO, PSO, GA ve TA algoritmaları ile karşılaştırılmıştır.

Çalışmanın amacı; GSP mantığı baz alınarak bir şehirden başlanıp verilerdeki tüm şehirlere uğrandıktan sonra tekrar başlangıç şehrine dönmek şartıyla karayolu yük taşımacılığında en ideal mesafenin bulunmasıdır. Algoritmada kullanılan organize sanayi bölgeleri arasındaki gerçek karayolu mesafe değerleri Google Earth’ten, OSB koordinatları ise Google haritalar ve OSB kendi sitelerinden bakılarak yapılmıştır. OSB koordinatları Çizelge 3.1’de, gerçek karayolu mesafeleri ise Çizelge 3.2’de gösterilmiştir.

Çizelge 3.1. OSB koordinatları

SIRA	ORGANİZE SANAYİ BÖLGELERİ	KOORDİNATLAR	
		X	Y
1	Adana Kozan Organize Sanayi Bölgesi	37,461362	35,765552
2	Adıyaman Organize Sanayi Bölgesi	37,772182	38,213367
3	Afyonkarahisar Organize Sanayi Bölgesi	38,796038	30,559430
4	Ağrı Organize Sanayi Bölgesi	39,720242	43,032627
5	Aksaray Organize Sanayi Bölgesi	38,279620	34,039940
6	Amasya Organize Sanayi Bölgesi	40,583659	35,643670
7	Ankara Sanayi Odası I. Organize Sanayi Bölgesi	39,985281	32,557627
8	Antakya Organize Sanayi Bölgesi	36,438213	36,266054
9	Antalya Organize Sanayi Bölgesi	37,081421	30,605369
10	Ardahan Organize Sanayi Bölgesi	41,111246	42,704906
11	Artvin Arhavi Organize Sanayi Bölgesi	41,331422	41,300299
12	Aydın Organize Sanayi Bölgesi	37,842823	27,890524
13	Balıkesir Organize Sanayi Bölgesi	39,594038	27,843575
14	Bartın Merkez I. Organize Sanayi Bölgesi	41,640073	32,328515
15	Batman Organize Sanayi Bölgesi	37,902854	41,133640
16	Bayburt Organize Sanayi Bölgesi	40,289232	40,164728
17	Bilecik I. Organize Sanayi Bölgesi	40,143815	29,974772
18	Bingöl Organize Sanayi Bölgesi	38,810304	40,527762
19	Bitlis Organize Sanayi Bölgesi	38,434203	42,138701
20	Bolu Karma Ve Tekstil İhtisas Organize Sanayi Bölgesi	40,738056	31,710704
21	Burdur II. Organize Sanayi Bölgesi	37,802546	30,387612
22	Bursa Organize Sanayi Bölgesi	40,244841	28,964237
23	Çanakkale Organize Sanayi Bölgesi	40,157048	26,464344
24	Çankırı Korgun Organize Sanayi Bölgesi	40,729186	33,516319
25	Çorum Organize Sanayi Bölgesi	40,478478	34,857005
26	Denizli Organize Sanayi Bölgesi	37,803913	29,241221
27	Diyarbakır Organize Sanayi Bölgesi	38,079130	40,068557
28	Düzce Organize Sanayi Bölgesi	40,841706	31,177041
29	Edirne Organize Sanayi Bölgesi	41,825306	26,832519
30	Elazığ Organize Sanayi Bölgesi	38,620488	39,300231
31	Erzincan Organize Sanayi Bölgesi	39,753335	39,398121
32	Erzurum Merkez I. Organize Sanayi Bölgesi	39,934872	41,186106
33	Eskişehir Sanayi Odası Organize Sanayi Bölgesi	39,743850	30,622113
34	Gaziantep Organize Sanayi Bölgesi	37,149188	37,295520
35	Giresun Organize Sanayi Bölgesi	40,892954	38,419926
36	Gümüşhane Organize Sanayi Bölgesi	40,432911	39,518272
37	Hakkâri Organize Sanayi Bölgesi	37,574256	43,737493
38	İğdır Organize Sanayi Bölgesi	39,934206	44,049613
39	Isparta Süleyman Demirel Organize Sanayi Bölgesi	37,888089	30,410037
40	İstanbul Anadolu Yakası Organize Sanayi Bölgesi	40,875825	29,341796
41	İzmir Atatürk Organize Sanayi Bölgesi	38,491909	27,037909



Çizelge 3.1. (Devam) OSB koordinatları

SIRA	ORGANİZE SANAYİ BÖLGELERİ	KOORDİNATLAR	
		X	Y
42	Kahramanmaraş Organize Sanayi Bölgesi	37,601103	36,772474
43	Karabük Organize Sanayi Bölgesi	41,137601	32,642985
44	Karaman Organize Sanayi Bölgesi	37,221808	33,308022
45	Kars Organize Sanayi Bölgesi	40,603598	43,097329
46	Kastamonu Organize Sanayi Bölgesi	41,522034	33,776064
47	Kayseri Organize Sanayi Bölgesi	38,729270	35,385758
48	Kilis Organize Sanayi Bölgesi	36,720531	37,217575
49	Kırıkkale Silah Sanayi İhtisas Organize Sanayi Bölgesi	39,813963	33,488427
50	Kırklareli Organize Sanayi Bölgesi	41,692118	27,336426
51	Kırşehir Organize Sanayi Bölgesi	39,078548	34,191881
52	Kocaeli Gebze Kömürçüler İhtisas Organize Sanayi Bölgesi	40,786472	29,528102
53	Konya Organize Sanayi Bölgesi	37,973946	32,610573
54	Kütahya Organize Sanayi Bölgesi	39,391802	30,117757
55	Malatya Merkez I. Organize Sanayi Bölgesi	38,340777	38,201391
56	Manisa Organize Sanayi Bölgesi	38,615848	27,365395
57	Mardin Organize Sanayi Bölgesi	37,283114	40,668681
58	Mersin Tarsus Organize Sanayi Bölgesi	36,899682	34,796679
59	Muğla Milas Organize Sanayi Bölgesi	37,303062	27,783070
60	Muş Organize Sanayi Bölgesi	38,747088	41,500370
61	Nevşehir Acıgöl Organize Sanayi Bölgesi	38,554517	34,517217
62	Niğde Organize Sanayi Bölgesi	37,947681	34,616439
63	Ordu Organize Sanayi Bölgesi	40,972422	37,931364
64	Osmaniye Organize Sanayi Bölgesi	37,008662	36,110558
65	Rize Organize Sanayi Bölgesi	40,917517	40,416974
66	Sakarya I. Organize Sanayi Bölgesi	40,721792	30,421702
67	Samsun Merkez Organize Sanayi Bölgesi	41,237923	36,433645
68	Siirt Organize Sanayi Bölgesi	37,924799	41,944743
69	Sinop Organize Sanayi Bölgesi	42,025929	35,151646
70	Sivas Merkez I. Organize Sanayi Bölgesi	39,771742	37,042907
71	Şanlıurfa Organize Sanayi Bölgesi	37,108623	38,649990
72	Şırnak Cizre Organize Sanayi Bölgesi	37,461253	42,297676
73	Tekirdağ Organize Sanayi Bölgesi	41,197187	27,534818
74	Tokat Merkez Organize Sanayi Bölgesi	40,329479	36,517194
75	Trabzon Arsin Organize Sanayi Bölgesi	40,958776	39,961717
76	Tunceli Organize Sanayi Bölgesi	39,075473	39,534661
77	Uşak Organize Sanayi Bölgesi	38,677248	29,220110
78	Van Organize Sanayi Bölgesi	38,576521	43,316720
79	Yalova Gemi İhtisas Organize Sanayi Bölgesi	40,656191	29,351163
80	Yozgat Kaleseramik Özel Organize Sanayi Bölgesi	39,623133	34,485649
81	Zonguldak Ereğli Organize Sanayi Bölgesi	41,282363	31,436680

Çizelge 3.2. OSB'ler arasındaki karayolu mesafe (km) değerleri

		1	2	3	4	5	6	7	8	9	10	11	12	13	14
		Adana	Adıyaman	Afyonkarahisar	Ağrı	Aksaray	Amasya	Ankara	Antakya	Antalya	Ardahan	Artvin	Aydın	Balıkesir	Bartın
1	Adana	0	325	646	950	335	537	578	176	700	1018	896	916	1043	849
2	Adıyaman	325	0	897	650	586	557	716	289	950	758	787	1166	1227	982
3	Afyonkarahisar	646	897	0	1318	360	572	255	725	265	1362	1157	327	337	517
4	Ağrı	950	650	1318	0	967	762	1082	918	1434	259	428	1637	1579	1139
5	Aksaray	335	586	360	967	0	370	245	413	471	1009	886	629	693	516
6	Amasya	537	557	572	762	370	0	341	653	784	794	609	901	844	403
7	Ankara	578	716	255	1082	245	341	0	655	473	1124	927	570	543	278
8	Antakya	176	289	725	918	413	653	655	0	779	1022	1006	995	1059	928
9	Antalya	700	950	265	1434	471	784	473	779	0	1480	1353	325	499	741
10	Ardahan	1018	758	1362	259	1009	794	1124	1022	1480	0	127	1636	1566	1099
11	Artvin	896	787	1157	428	886	609	927	1006	1353	127	0	1488	1391	924
12	Aydın	916	1166	327	1637	629	901	570	995	325	1636	1488	0	315	845
13	Balıkesir	1043	1227	337	1579	693	844	543	1059	499	1566	1391	315	0	609
14	Bartın	849	982	517	1139	516	403	278	928	741	1099	924	845	609	0
15	Batman	614	316	1189	369	829	792	1018	580	1245	579	638	1457	1522	1282
16	Bayburt	705	542	1025	310	696	479	787	819	1167	353	224	1324	1298	861
17	Bilecik	816	1000	212	1361	484	625	284	895	454	1348	1173	458	256	377
18	Bingöl	643	345	1102	365	744	629	924	609	1215	416	447	1372	1436	1011
19	Bitlis	731	429	1284	224	925	733	1110	696	1361	457	530	1575	1559	1110
20	Bolu	737	871	405	1140	404	405	166	816	630	1184	952	734	440	169
21	Burdur	687	938	153	1376	412	726	405	766	115	1419	1298	261	382	669
22	Bursa	902	1086	298	1439	569	703	402	981	540	1426	1251	444	150	447
23	Çanakkale	1152	1336	548	1689	818	953	714	1231	690	1676	1501	444	198	720
24	Çankırı	664	759	420	985	331	244	181	743	644	972	797	749	619	230
25	Çorum	509	635	493	844	306	82	266	636	709	845	670	821	785	356
26	Denizli	793	1044	205	1515	506	778	448	872	215	1513	1365	125	295	722
27	Diyarbakır	523	225	1098	461	702	664	890	489	1154	559	590	1330	1394	1098
28	Düzce	793	926	376	1196	460	460	222	872	618	1183	1008	683	389	210
29	Edirne	1249	1382	667	1652	916	916	678	1328	910	1695	1464	829	356	684
30	Elazığ	450	215	958	484	600	548	788	451	1071	535	566	1228	1292	856
31	Erzincan	599	421	944	376	590	388	717	652	1061	420	361	1218	1206	770
32	Erzurum	786	534	1130	189	777	575	903	799	1248	239	270	1405	1393	956
33	Eskişehir	722	906	129	1279	389	543	212	801	391	1323	1130	415	326	486
34	Gaziantep	200	155	775	758	463	571	672	153	830	863	893	1045	1107	936
35	Giresun	645	592	875	536	628	326	645	759	1091	466	291	1203	1108	648
36	Gümüşhane	674	539	994	374	665	448	767	788	1136	417	261	1293	1267	806
37	Hakkâri	896	655	1471	426	1159	1048	1408	862	1526	639	818	1741	1803	1429
38	Iğdır	1050	752	1428	150	1074	873	1201	1016	1546	222	401	1703	1691	1254
39	Isparta	689	940	141	1382	403	714	384	768	125	1425	1301	249	371	658
40	İstanbul	963	1097	382	1366	630	633	377	1044	626	1412	1180	546	252	400
41	İzmir	986	1236	343	1653	699	917	586	1065	464	1697	1504	140	187	806

Çizelge 3.2. (Devam) OSB'ler arasındaki karayolu mesafe (km) değerleri

		15	16	17	18	19	20	21	22	23	24	25	26	27	28
		Batman	Bayburt	Bilecik	Bingöl	Bitlis	Bolu	Burdur	Bursa	Çanakkale	Çankırı	Çorum	Denizli	Diyarbakır	Düzce
1	Adana	614	705	816	643	731	737	687	902	1152	664	509	793	523	793
2	Adıyaman	316	542	1000	345	429	871	938	1086	1336	759	635	1044	225	926
3	Afyonkarahisar	1189	1025	212	1102	1284	405	153	298	548	420	493	205	1098	376
4	Ağrı	369	310	1361	365	224	1140	1376	1439	1689	985	844	1515	461	1196
5	Aksaray	829	696	484	744	925	404	412	569	818	331	306	506	702	460
6	Amasya	792	479	625	629	733	405	726	703	953	244	82	778	664	460
7	Ankara	1018	787	284	924	1110	166	405	402	714	181	266	448	890	222
8	Antakya	580	819	895	609	696	816	766	981	1231	743	636	872	489	872
9	Antalya	1245	1167	454	1215	1361	630	115	540	690	644	709	215	1154	618
10	Ardahan	579	353	1348	416	457	1184	1419	1426	1676	972	845	1513	559	1183
11	Artvin	638	224	1173	447	530	952	1298	1251	1501	797	670	1365	590	1008
12	Aydın	1457	1324	458	1372	1575	734	261	444	444	749	821	125	1330	683
13	Balıkesir	1522	1298	256	1436	1559	440	382	150	198	619	785	295	1394	389
14	Bartın	1282	861	377	1011	1110	169	669	447	720	230	356	722	1098	210
15	Batman	0	508	1364	183	148	1170	1235	1450	1614	1002	877	1341	136	1226
16	Bayburt	508	0	1079	316	395	859	1109	1157	1407	703	562	1203	462	914
17	Bilecik	1364	1079	0	1206	1343	221	341	105	365	400	556	355	1164	170
18	Bingöl	183	316	1206	0	190	1079	1157	1273	1523	867	726	1251	147	1078
19	Bitlis	148	395	1343	190	0	1111	1344	1433	1683	956	814	1416	240	1167
20	Bolu	1170	859	221	1079	1111	0	555	300	550	179	404	607	1041	56
21	Burdur	1235	1109	341	1157	1344	555	0	431	584	576	649	141	1112	509
22	Bursa	1450	1157	105	1273	1433	300	431	0	258	478	644	417	1253	248
23	Çanakkale	1614	1407	365	1523	1683	550	584	258	0	729	895	491	1504	499
24	Çankırı	1002	703	400	867	956	179	576	478	729	0	181	622	866	234
25	Çorum	877	562	556	726	814	404	649	644	895	181	0	699	742	461
26	Denizli	1341	1203	355	1251	1416	607	141	417	491	622	699	0	1206	523
27	Diyarbakır	136	462	1164	147	240	1041	1112	1253	1504	866	742	1206	0	1092
28	Düzce	1226	914	170	1078	1167	56	509	248	499	234	461	523	1092	0
29	Edirne	1682	1370	440	1590	1624	513	801	361	246	690	917	654	1548	462
30	Elazığ	264	400	1062	146	311	939	1010	1151	1402	749	625	1103	130	994
31	Erzincan	450	127	988	258	347	767	1000	1065	1317	611	470	1094	350	823
32	Erzurum	385	124	1174	194	276	954	1187	1252	1503	797	657	1280	337	1009
33	Eskişehir	1269	998	97	1093	1253	312	282	185	436	387	463	312	1050	261
34	Gaziantep	426	654	942	451	537	822	816	1030	1281	662	554	920	330	878
35	Giresun	735	227	889	537	626	669	1030	967	1218	512	385	1079	621	724
36	Gümüşhane	572	65	1048	364	463	827	1075	1125	1377	671	530	1169	468	883
37	Hakkâri	366	703	1638	501	312	1523	1512	1726	1977	1270	1130	1616	498	1482
38	Iğdır	469	422	1472	479	324	1252	1485	1550	1801	1095	955	1578	561	1307
39	Isparta	1237	1115	329	1163	1324	544	15	392	566	558	635	125	1121	494
40	İstanbul	1398	1087	157	1307	1337	230	499	111	362	407	633	526	1265	179
41	İzmir	1533	1372	436	1444	1644	636	392	329	304	761	837	252	1401	585

Çizelge 3.2. (Devam) OSB'ler arasındaki karayolu mesafe (km) değerleri

		29	30	31	32	33	34	35	36	37	38	39	40	41	42
		Edirne	Elazığ	Erzincan	Erzurum	Eskişehir	Gaziantep	Giresun	Gümüşhane	Hakkâri	Iğdır	Isparta	İstanbul	İzmir	Kahramanmaraş
1	Adana	1249	450	599	786	722	200	645	674	896	1050	689	963	986	146
2	Adıyaman	1382	215	421	534	906	155	592	539	655	752	940	1097	1236	180
3	Afyonkarahisar	667	958	944	1130	129	775	875	994	1471	1428	141	382	343	775
4	Ağrı	1652	484	376	189	1279	758	536	374	426	150	1382	1366	1653	821
5	Aksaray	916	600	590	777	389	463	628	665	1159	1074	403	630	699	408
6	Amasya	916	548	388	575	543	571	326	448	1048	873	714	633	917	495
7	Ankara	678	788	717	903	212	672	645	767	1408	1201	384	377	586	596
8	Antakya	1328	451	652	799	801	153	759	788	862	1016	768	1044	1065	162
9	Antalya	910	1071	1061	1248	391	830	1091	1136	1526	1546	125	626	464	767
10	Ardahan	1695	535	420	239	1323	863	466	417	639	222	1425	1412	1697	872
11	Artvin	1464	566	361	270	1130	893	291	261	818	401	1301	1180	1504	853
12	Aydın	829	1228	1218	1405	415	1045	1203	1293	1741	1703	249	546	140	1036
13	Balıkesir	356	1292	1206	1393	326	1107	1108	1267	1803	1691	371	252	187	1100
14	Bartın	684	856	770	956	486	936	648	806	1429	1254	658	400	806	861
15	Batman	1682	264	450	385	1269	426	735	572	366	469	1237	1398	1533	503
16	Bayburt	1370	400	127	124	998	654	227	65	703	422	1115	1087	1372	664
17	Bilecik	440	1062	988	1174	97	942	889	1048	1638	1472	329	157	436	870
18	Bingöl	1590	146	258	194	1093	451	537	364	501	479	1163	1307	1444	483
19	Bitlis	1624	311	347	276	1253	537	626	463	312	324	1324	1337	1644	614
20	Bolu	513	939	767	954	312	822	669	827	1523	1252	544	230	636	747
21	Burdur	801	1010	1000	1187	282	816	1030	1075	1512	1485	15	499	392	816
22	Bursa	361	1151	1065	1252	185	1030	967	1125	1726	1550	392	111	329	959
23	Çanakkale	246	1402	1317	1503	436	1281	1218	1377	1977	1801	566	362	304	1210
24	Çankırı	690	749	611	797	387	662	512	671	1270	1095	558	407	761	587
25	Çorum	917	625	470	657	463	554	385	530	1130	955	635	633	837	479
26	Denizli	654	1103	1094	1280	312	920	1079	1169	1616	1578	125	526	252	912
27	Diyarbakır	1548	130	350	337	1050	330	621	468	498	561	1121	1265	1401	407
28	Düzce	462	994	823	1009	261	878	724	883	1482	1307	494	179	585	802
29	Edirne	0	1457	1285	1472	554	1382	1187	1345	2078	1770	787	296	539	1265
30	Elazığ	1457	0	237	314	948	318	499	355	621	599	1019	1162	1299	338
31	Erzincan	1285	237	0	189	906	519	283	120	662	487	1009	995	1401	558
32	Erzurum	1472	314	189	0	1093	642	349	187	593	301	1195	1182	1588	651
33	Eskişehir	554	948	906	1093	0	847	843	963	1543	1396	266	267	517	776
34	Gaziantep	1382	318	519	642	847	0	678	707	703	857	817	1093	1113	81
35	Giresun	1187	499	283	349	843	678	0	164	960	678	1017	896	1302	607
36	Gümüşhane	1345	355	120	187	963	707	164	0	802	521	1090	1069	1354	646
37	Hakkâri	2078	621	662	593	1543	703	960	802	0	422	1514	1651	1810	782
38	Iğdır	1770	599	487	301	1396	857	678	521	422	0	1492	1479	1885	937
39	Isparta	787	1019	1009	1195	266	817	1017	1090	1514	1492	0	485	378	820
40	İstanbul	296	1162	995	1182	267	1093	896	1069	1651	1479	485	0	444	972
41	İzmir	539	1299	1401	1588	517	1113	1302	1354	1810	1885	378	444	0	1107

Çizelge 3.2. (Devam) OSB'ler arasındaki karayolu mesafe (km) değerleri

		43	44	45	46	47	48	49	50	51	52	53	54	55	56
		Karabük	Karaman	Kars	Kastamonu	Kayseri	Kilis	Kırıkkale	Kırklareli	Kırşehir	Kocaeli	Konya	Kütahya	Malatya	Manisa
1	Adana	772	351	997	752	237	269	534	1209	421	944	428	742	333	957
2	Adıyaman	838	602	734	809	387	207	615	1342	511	1078	678	992	108	1207
3	Afyonkarahisar	440	351	1342	508	504	843	343	627	425	368	225	99	840	314
4	Ağrı	1097	1118	170	986	824	790	979	1612	938	1347	1100	1353	601	1624
5	Aksaray	424	190	988	419	146	532	201	864	88	595	137	455	482	669
6	Amasya	362	568	786	251	286	644	242	876	279	612	475	617	453	887
7	Ankara	201	408	1115	255	340	745	116	638	224	358	276	286	671	557
8	Antakya	851	430	998	831	416	127	613	1288	500	1025	510	821	334	1036
9	Antalya	664	353	1459	732	617	399	549	818	560	618	339	341	953	429
10	Ardahan	1141	1161	89	948	868	895	1022	1655	981	1393	1144	1397	652	1667
11	Artvin	910	1040	268	773	746	826	829	1424	860	1161	1023	1204	683	1474
12	Aydın	768	611	1616	836	774	1114	672	607	717	538	495	345	1110	148
13	Balıkesir	566	683	1605	687	844	1176	635	313	728	244	557	252	1174	142
14	Bartın	100	672	1168	159	600	1010	318	643	432	381	540	514	862	750
15	Batman	1137	899	546	1052	687	458	914	1642	810	1379	973	1289	354	1504
16	Bayburt	816	851	335	705	557	798	697	1330	671	1068	834	1072	454	1342
17	Bilecik	347	533	1386	469	614	1010	406	400	498	149	407	128	945	395
18	Bingöl	979	884	393	869	595	483	823	1550	719	1287	880	1167	262	1414
19	Bitlis	1067	1011	401	970	760	569	958	1584	883	1329	1047	1327	427	1612
20	Bolu	126	558	1165	248	485	896	255	473	369	210	426	346	821	580
21	Burdur	596	382	1399	664	556	884	494	760	499	491	282	233	893	362
22	Bursa	425	622	1464	546	703	1099	494	321	586	103	496	191	1033	288
23	Çanakkale	676	873	1715	797	954	1350	746	265	838	354	747	436	1284	319
24	Çankırı	136	495	1009	90	354	765	130	650	244	388	363	461	654	731
25	Çorum	311	496	868	199	261	628	162	877	220	614	396	537	531	808
26	Denizli	633	487	1492	712	650	989	536	611	592	518	371	242	986	222
27	Diyarbakır	1004	803	536	919	553	362	781	1508	677	1245	839	1124	220	1372
28	Düzce	182	614	1221	304	541	951	310	422	425	159	482	292	877	529
29	Edirne	645	991	1684	766	1009	1451	773	58	893	317	865	586	1339	674
30	Elazığ	874	739	513	790	451	386	678	1406	575	1143	736	1022	118	1270
31	Erzincan	724	745	401	613	451	580	605	1239	565	976	726	980	319	1345
32	Erzurum	911	931	214	800	638	674	801	1425	751	1162	913	1175	431	1532
33	Eskişehir	408	439	1310	476	520	916	311	510	404	259	313	103	850	461
34	Gaziantep	785	479	839	779	334	62	562	1336	458	1073	555	869	205	1084
35	Giresun	625	794	592	488	501	756	544	1139	601	877	777	919	494	1246
36	Gümüşhane	798	826	434	650	540	795	679	1313	653	1050	807	1054	460	1419
37	Hakkâri	1380	1175	553	1269	1071	735	1298	2033	1194	1632	1252	1566	738	1781
38	Iğdır	1208	1228	136	1097	935	890	1089	1722	1048	1460	1210	1464	715	1829
39	Isparta	568	385	1406	636	564	888	471	712	503	478	285	219	901	349
40	İstanbul	351	706	1390	473	716	1128	480	256	600	31	580	279	1046	388
41	İzmir	770	691	1685	853	844	1185	687	511	780	442	565	352	1180	43

Çizelge 3.2. (Devam) OSB'ler arasındaki karayolu mesafe (km) değerleri

		57	58	59	60	61	62	63	64	65	66	67	68	69	70
		Mardin	Mersin	Muğla	Muş	Neveşehir	Niğde	Ordu	Osmaniye	Rize	Sakarya	Samsun	Siirt	Sinop	Sivas
1	Adana	531	138	997	717	338	251	622	91	810	864	666	690	795	369
2	Adıyaman	290	389	1248	419	476	502	614	263	678	998	686	390	814	350
3	Afyonkarahisar	1106	561	428	1198	421	455	825	650	1071	303	686	1264	679	714
4	Ağrı	516	1018	1693	244	903	950	579	892	400	1268	721	331	886	608
5	Aksaray	795	250	722	840	63	98	579	339	800	532	501	908	585	360
6	Amasya	786	554	1001	669	319	397	277	590	523	532	148	846	266	230
7	Ankara	1008	500	661	997	295	354	596	589	841	295	456	1097	426	487
8	Antakya	497	218	1076	683	422	331	736	90	920	945	780	656	909	484
9	Antalya	1162	517	372	1311	519	518	1042	642	1267	546	902	1320	903	831
10	Ardahan	640	1123	1718	391	965	994	509	997	290	1312	651	568	817	651
11	Artvin	671	1023	1570	465	843	872	334	987	115	1081	476	642	642	530
12	Aydın	1376	831	126	1468	676	714	1154	920	1399	614	1014	1535	1007	988
13	Balıkesir	1439	893	365	1532	752	788	1059	983	1304	313	919	1597	856	1006
14	Bartın	1194	763	926	1050	519	607	599	854	845	274	460	1279	338	639
15	Batman	158	686	1545	190	794	799	777	560	599	1299	897	77	1026	593
16	Bayburt	543	834	1407	337	654	683	270	757	124	987	412	515	578	341
17	Bilecik	1273	728	541	1302	523	582	840	817	1085	98	701	1432	638	777
18	Bingöl	229	711	1456	122	702	716	572	566	408	1207	705	240	834	486
19	Bitlis	294	797	1631	71	859	884	658	671	488	1238	795	119	923	578
20	Bolu	1159	650	799	1047	445	505	620	740	865	130	480	1242	417	625
21	Burdur	1147	602	318	1250	459	488	981	691	1226	437	842	1305	835	771
22	Bursa	1362	816	507	1391	611	671	918	906	1163	172	778	1520	715	865
23	Çanakkale	1613	1067	506	1642	863	922	1169	1157	1414	423	1029	1710	967	1116
24	Çankırı	1028	575	827	891	331	419	463	665	708	307	323	1072	261	431
25	Çorum	864	504	903	750	294	348	336	613	581	534	196	927	280	308
26	Denizli	1252	707	209	1343	552	594	1029	796	1275	446	890	1410	883	864
27	Diyarbakır	124	590	1449	221	660	663	643	464	551	1165	777	205	906	460
28	Düzce	1214	706	748	1103	501	560	675	795	920	78	536	1303	473	681
29	Edirne	1714	1168	893	1566	963	1023	1138	1258	1383	392	998	1872	935	1144
30	Elazığ	253	552	1329	242	558	571	521	421	528	1063	655	338	784	331
31	Erzincan	460	731	1320	283	548	577	326	692	259	896	451	460	579	234
32	Erzurum	419	902	1551	214	735	764	392	776	214	1082	534	391	700	421
33	Eskişehir	1179	633	679	1208	428	488	794	723	1039	186	654	1337	647	682
34	Gaziantep	339	266	1125	524	471	379	655	140	854	993	699	497	827	402
35	Giresun	737	780	1465	594	598	627	50	741	202	796	191	771	357	284
36	Gümüşhane	581	812	1439	436	629	658	211	781	177	970	353	614	519	323
37	Hakkâri	383	963	1822	380	1167	1076	971	837	793	1552	1107	292	1235	950
38	Iğdır	614	1117	1803	389	1032	1061	689	991	511	1357	831	430	997	718
39	Isparta	1151	606	329	1258	466	492	968	695	1213	423	828	1309	821	779
40	İstanbul	1391	875	607	1280	670	730	845	972	1090	101	705	1480	642	850
41	İzmir	1448	901	205	1539	748	797	1170	992	1509	517	1031	1606	1024	1059

Çizelge 3.2. (Devam) OSB'ler arasındaki karayolu mesafe (km) değerleri

		71	72	73	74	75	76	77	78	79	80	81
		Şanlıurfa	Şırnak	Tekirdağ	Tokat	Trabzon	Tunceli	Uşak	Van	Yalova	Yozgat	Zonguldak
1	Adana	342	697	1116	427	760	527	772	885	975	417	862
2	Adıyaman	114	455	1250	447	621	280	1023	587	1109	587	996
3	Afyonkarahisar	917	1271	535	634	1021	1023	130	1460	350	454	440
4	Ağrı	640	445	1519	676	417	458	1440	221	1379	864	1266
5	Aksaray	606	960	783	386	751	665	485	1066	643	187	530
6	Amasya	662	960	784	111	473	526	703	895	643	181	530
7	Ankara	819	1173	541	407	742	853	372	1255	388	227	292
8	Antakya	307	662	1195	542	870	517	851	851	1033	570	941
9	Antalya	973	1327	777	850	1218	1136	287	1516	592	658	682
10	Ardahan	745	682	1564	707	314	505	1483	434	1422	907	1309
11	Artvin	776	756	1331	532	139	499	1290	613	1191	786	1078
12	Aydın	1187	1542	568	962	1350	1293	207	1694	512	783	747
13	Balıkesir	1250	1604	274	926	1255	1345	212	1758	218	746	453
14	Bartın	1084	1369	551	519	795	908	646	1276	387	428	138
15	Batman	308	190	1549	691	615	318	1320	305	1408	829	1295
16	Bayburt	646	628	1238	394	108	276	1158	499	1097	597	984
17	Bilecik	1084	1438	308	697	1036	1126	250	1549	117	517	234
18	Bingöl	333	348	1458	558	424	139	1230	348	1317	724	1204
19	Bitlis	419	223	1491	647	501	304	1413	159	1357	852	1195
20	Bolu	970	1324	381	517	815	905	465	1329	240	366	129
21	Burdur	958	1312	668	790	1177	1075	178	1477	469	598	573
22	Bursa	1173	1527	228	785	1114	1203	255	1617	77	605	312
23	Çanakkale	1424	1778	247	1036	1365	1455	407	1868	328	856	563
24	Çankırı	839	1194	558	322	659	749	547	1117	417	215	290
25	Çorum	702	1038	784	192	532	608	623	977	644	122	481
26	Denizli	1063	1417	677	838	1225	1169	154	1570	478	658	579
27	Diyarbakır	212	289	1416	557	543	190	1224	396	1275	697	1162
28	Düzce	1062	1416	330	572	871	961	414	1384	189	421	78
29	Edirne	1525	1879	151	1035	1384	1423	550	1847	334	884	524
30	Elazığ	304	427	1313	428	430	126	1085	468	1173	568	1060
31	Erzincan	511	634	1146	303	242	141	1066	564	1005	490	892
32	Erzurum	524	504	1333	489	230	275	1262	377	1192	677	1079
33	Eskişehir	990	1344	418	602	989	1050	224	1473	233	422	320
34	Gaziantep	150	504	1244	460	797	387	900	693	1103	495	990
35	Giresun	703	885	1047	250	153	418	1005	755	906	503	793
36	Gümüşhane	632	727	1220	377	127	262	1140	598	1079	579	966
37	Hakkâri	561	210	1940	1047	809	613	1596	207	1661	1187	1549
38	Iğdır	740	544	1630	786	527	572	1550	217	1489	974	1376
39	Isparta	962	1316	654	776	1163	1083	165	1484	455	582	556
40	İstanbul	1202	1557	163	743	1040	1130	372	1554	48	591	231
41	İzmir	1259	1613	472	979	1366	1364	216	1766	396	798	649

Çizelge 3.2. (Devam) OSB'ler arasındaki karayolu mesafe (km) değerleri

		1	2	3	4	5	6	7	8	9	10	11	12	13	14
		Adana	Adıyaman	Afyonkarahisar	Ağrı	Aksaray	Amasya	Ankara	Antakya	Antalya	Ardahan	Artvin	Aydın	Balıkesir	Bartın
42	Kahramanmaraş	146	180	775	821	408	495	596	162	767	872	853	1036	1100	861
43	Karabük	772	838	440	1097	424	362	201	851	664	1141	910	768	566	100
44	Karaman	351	602	351	1118	190	568	408	430	353	1161	1040	611	683	672
45	Kars	997	734	1342	170	988	786	1115	998	1459	89	268	1616	1605	1168
46	Kastamonu	752	809	508	986	419	251	255	831	732	948	773	836	687	159
47	Kayseri	237	387	504	824	146	286	340	416	617	868	746	774	844	600
48	Kilis	269	207	843	790	532	644	745	127	399	895	826	1114	1176	1010
49	Kırıkale	534	615	343	979	201	242	116	613	549	1022	829	672	635	318
50	Kırklareli	1209	1342	627	1612	864	876	638	1288	818	1655	1424	607	313	643
51	Kırşehir	421	511	425	938	88	279	224	500	560	981	860	717	728	432
52	Kocaeli	944	1078	368	1347	595	612	358	1025	618	1393	1161	538	244	381
53	Konya	428	678	225	1100	137	475	276	510	339	1144	1023	495	557	540
54	Kütahya	742	992	99	1353	455	617	286	821	341	1397	1204	345	252	514
55	Malatya	333	108	840	601	482	453	671	334	953	652	683	1110	1174	862
56	Manisa	957	1207	314	1624	669	887	557	1036	429	1667	1474	148	142	750
57	Mardin	531	290	1106	516	795	786	1008	497	1162	640	671	1376	1439	1194
58	Mersin	138	389	561	1018	250	554	500	218	517	1123	1023	831	893	763
59	Muğla	997	1248	428	1693	722	1001	661	1076	372	1718	1570	126	365	926
60	Muş	717	419	1198	244	840	669	997	683	1311	391	465	1468	1532	1050
61	Neşehir	338	476	421	903	63	319	295	422	519	965	843	676	752	519
62	Niğde	251	502	455	950	98	397	354	331	518	994	872	714	788	607
63	Ordu	622	614	825	579	579	277	596	736	1042	509	334	1154	1059	599
64	Osmaniye	91	263	650	892	339	590	589	90	642	997	987	920	983	854
65	Rize	810	678	1071	400	800	523	841	920	1267	290	115	1399	1304	845
66	Sakarya	864	998	303	1268	532	532	295	945	546	1312	1081	614	313	274
67	Samsun	666	686	686	721	501	148	456	780	902	651	476	1014	919	460
68	Siirt	690	390	1264	331	908	846	1097	656	1320	568	642	1535	1597	1279
69	Sinop	795	814	679	886	585	266	426	909	903	817	642	1007	856	338
70	Sivas	369	350	714	608	360	230	487	484	831	651	530	988	1006	639
71	Şanlıurfa	342	114	917	640	606	662	819	307	973	745	776	1187	1250	1084
72	Şırnak	697	455	1271	445	960	960	1173	662	1327	682	756	1542	1604	1369
73	Tekirdağ	1116	1250	535	1519	783	784	541	1195	777	1564	1331	568	274	551
74	Tokat	427	447	634	676	386	111	407	542	850	707	532	962	926	519
75	Trabzon	760	621	1021	417	751	473	742	870	1218	314	139	1350	1255	795
76	Tunceli	527	280	1023	458	665	526	853	517	1136	505	499	1293	1345	908
77	Uşak	772	1023	130	1440	485	703	372	851	287	1483	1290	207	212	646
78	Van	885	587	1460	221	1066	895	1255	851	1516	434	613	1694	1758	1276
79	Yalova	975	1109	350	1379	643	643	388	1033	592	1422	1191	512	218	387
80	Yozgat	417	587	454	864	187	181	227	570	658	907	786	783	746	428
81	Zonguldak	862	996	440	1266	530	530	292	941	682	1309	1078	747	453	138



Çizelge 3.2. (Devam) OSB'ler arasındaki karayolu mesafe (km) değerleri

		15	16	17	18	19	20	21	22	23	24	25	26	27	28
		Batman	Bayburt	Bilecik	Bingöl	Bitlis	Bolu	Burdur	Bursa	Çanakkale	Çankırı	Çorum	Denizli	Diyarbakır	Düzce
42	Kahramanmaraş	503	664	870	483	614	747	816	959	1210	587	479	912	407	802
43	Karabük	1137	816	347	979	1067	126	596	425	676	136	311	633	1004	182
44	Karaman	899	851	533	884	1011	558	382	622	873	495	496	487	803	614
45	Kars	546	335	1386	393	401	1165	1399	1464	1715	1009	868	1492	536	1221
46	Kastamonu	1052	705	469	869	970	248	664	546	797	90	199	712	919	304
47	Kayseri	687	557	614	595	760	485	556	703	954	354	261	650	553	541
48	Kilis	458	798	1010	483	569	896	884	1099	1350	765	628	989	362	951
49	Kırkkale	914	697	406	823	958	255	494	494	746	130	162	536	781	310
50	Kırklareli	1642	1330	400	1550	1584	473	760	321	265	650	877	611	1508	422
51	Kırşehir	810	671	498	719	883	369	499	586	838	244	220	592	677	425
52	Kocaeli	1379	1068	149	1287	1329	210	491	103	354	388	614	518	1245	159
53	Konya	973	834	407	880	1047	426	282	496	747	363	396	371	839	482
54	Kütahya	1289	1072	128	1167	1327	346	233	191	436	461	537	242	1124	292
55	Malatya	354	454	945	262	427	821	893	1033	1284	654	531	986	220	877
56	Manisa	1504	1342	395	1414	1612	580	362	288	319	731	808	222	1372	529
57	Mardin	158	543	1273	229	294	1159	1147	1362	1613	1028	864	1252	124	1214
58	Mersin	686	834	728	711	797	650	602	816	1067	575	504	707	590	706
59	Muğla	1545	1407	541	1456	1631	799	318	507	506	827	903	209	1449	748
60	Muş	190	337	1302	122	71	1047	1250	1391	1642	891	750	1343	221	1103
61	Neveşehir	794	654	523	702	859	445	459	611	863	331	294	552	660	501
62	Niğde	799	683	582	716	884	505	488	671	922	419	348	594	663	560
63	Ordu	777	270	840	572	658	620	981	918	1169	463	336	1029	643	675
64	Osmaniye	560	757	817	566	671	740	691	906	1157	665	613	796	464	795
65	Rize	599	124	1085	408	488	865	1226	1163	1414	708	581	1275	551	920
66	Sakarya	1299	987	98	1207	1238	130	437	172	423	307	534	446	1165	78
67	Samsun	897	412	701	705	795	480	842	778	1029	323	196	890	777	536
68	Siirt	77	515	1432	240	119	1242	1305	1520	1710	1072	927	1410	205	1303
69	Sinop	1026	578	638	834	923	417	835	715	967	261	280	883	906	473
70	Sivas	593	341	777	486	578	625	771	865	1116	431	308	864	460	681
71	Şanlıurfa	308	646	1084	333	419	970	958	1173	1424	839	702	1063	212	1062
72	Şırnak	190	628	1438	348	223	1324	1312	1527	1778	1194	1038	1417	289	1416
73	Tekirdağ	1549	1238	308	1458	1491	381	668	228	247	558	784	677	1416	330
74	Tokat	691	394	697	558	647	517	790	785	1036	322	192	838	557	572
75	Trabzon	615	108	1036	424	501	815	1177	1114	1365	659	532	1225	543	871
76	Tunceli	318	276	1126	139	304	905	1075	1203	1455	749	608	1169	190	961
77	Uşak	1320	1158	250	1230	1413	465	178	255	407	547	623	154	1224	414
78	Van	305	499	1549	348	159	1329	1477	1617	1868	1117	977	1570	396	1384
79	Yalova	1408	1097	117	1317	1357	240	469	77	328	417	644	478	1275	189
80	Yozgat	829	597	517	724	852	366	598	605	856	215	122	658	697	421
81	Zonguldak	1295	984	234	1204	1195	129	573	312	563	290	481	579	1162	78

Çizelge 3.2. (Devam) OSB'ler arasındaki karayolu mesafe (km) değerleri

		29	30	31	32	33	34	35	36	37	38	39	40	41	42
		Edirne	Elazığ	Erzincan	Erzurum	Eskişehir	Gaziantep	Giresun	Gümüşhane	Hakkâri	İğdır	Isparta	İstanbul	İzmir	Kahramanmaraş
42	Kahramanmaraş	1265	338	558	651	776	81	607	646	782	937	820	972	1107	0
43	Karabük	645	874	724	911	408	785	625	798	1380	1208	568	351	770	709
44	Karaman	991	739	745	931	439	479	794	826	1175	1228	385	706	691	478
45	Kars	1684	513	401	214	1310	839	592	434	553	136	1406	1390	1685	854
46	Kastamonu	766	790	613	800	476	779	488	650	1269	1097	636	473	853	691
47	Kayseri	1009	451	451	638	520	334	501	540	1071	935	564	716	844	259
48	Kilis	1451	386	580	674	916	62	756	795	735	890	888	1128	1185	142
49	Kırıkale	773	678	605	801	311	562	544	679	1298	1089	471	480	687	486
50	Kırklareli	58	1406	1239	1425	510	1336	1139	1313	2033	1722	712	256	511	1214
51	Kırşehir	893	575	565	751	404	458	601	653	1194	1048	503	600	780	382
52	Kocaeli	317	1143	976	1162	259	1073	877	1050	1632	1460	478	31	442	951
53	Konya	865	736	726	913	313	555	777	807	1252	1210	285	580	565	545
54	Kütahya	586	1022	980	1175	103	869	919	1054	1566	1464	219	279	352	830
55	Malatya	1339	118	319	431	850	205	494	460	738	715	901	1046	1180	225
56	Manisa	674	1270	1345	1532	461	1084	1246	1419	1781	1829	349	388	43	1078
57	Mardin	1714	253	460	419	1179	339	737	581	383	614	1151	1391	1448	418
58	Mersin	1168	552	731	902	633	266	780	812	963	1117	606	875	901	255
59	Muğla	893	1329	1320	1551	679	1125	1465	1439	1822	1803	329	607	205	1125
60	Muş	1566	242	283	214	1208	524	594	436	380	389	1258	1280	1539	583
61	Neveşehir	963	558	548	735	428	471	598	629	1167	1032	466	670	748	337
62	Niğde	1023	571	577	764	488	379	627	658	1076	1061	492	730	797	330
63	Ordu	1138	521	326	392	794	655	50	211	971	689	968	845	1170	578
64	Osmaniye	1258	421	692	776	723	140	741	781	837	991	695	972	992	140
65	Rize	1383	528	259	214	1039	854	202	177	793	511	1213	1090	1509	771
66	Sakarya	392	1063	896	1082	186	993	796	970	1552	1357	423	101	517	871
67	Samsun	998	655	451	534	654	699	191	353	1107	831	828	705	1031	623
68	Siirt	1872	338	460	391	1337	497	771	614	292	430	1309	1480	1606	576
69	Sinop	935	784	579	700	647	827	357	519	1235	997	821	642	1024	752
70	Sivas	1144	331	234	421	682	402	284	323	950	718	779	850	1059	327
71	Şanlıurfa	1525	304	511	524	990	150	703	632	561	740	962	1202	1259	229
72	Şırnak	1879	427	634	504	1344	504	885	727	210	544	1316	1557	1613	583
73	Tekirdağ	151	1313	1146	1333	418	1244	1047	1220	1940	1630	654	163	472	1121
74	Tokat	1035	428	303	489	602	460	250	377	1047	786	776	743	979	385
75	Trabzon	1384	430	242	230	989	797	153	127	809	527	1163	1040	1366	721
76	Tunceli	1423	126	141	275	1050	387	418	262	613	572	1083	1130	1364	408
77	Uşak	550	1085	1066	1262	224	900	1005	1140	1596	1550	165	372	216	893
78	Van	1847	468	564	377	1473	693	755	598	207	217	1484	1554	1766	772
79	Yalova	334	1173	1005	1192	233	1103	906	1079	1661	1489	455	48	396	980
80	Yozgat	884	568	490	677	422	495	503	579	1187	974	582	591	798	420
81	Zonguldak	524	1060	892	1079	320	990	793	966	1549	1376	556	231	649	868

Çizelge 3.2. (Devam) OSB'ler arasındaki karayolu mesafe (km) değerleri

		43	44	45	46	47	48	49	50	51	52	53	54	55	56
		Karabük	Karaman	Kars	Kastamonu	Kayseri	Kilis	Kırıkkale	Kırklareli	Kırşehir	Kocaeli	Konya	Kütahya	Malatya	Manisa
42	Kahramanmaraş	709	478	854	691	259	142	486	1214	382	951	545	830	225	1078
43	Karabük	0	594	1121	129	453	858	223	596	337	333	462	466	784	703
44	Karaman	594	0	1138	584	296	546	368	934	276	699	134	445	607	662
45	Kars	1121	1138	0	1009	847	870	1002	1635	961	1372	1122	1385	627	1742
46	Kastamonu	129	584	1009	0	447	853	218	717	332	454	451	549	705	824
47	Kayseri	453	296	847	447	0	408	230	958	126	695	279	574	334	811
48	Kilis	858	546	870	853	408	0	636	1410	532	1143	625	939	267	1154
49	Kırıkkale	223	368	1002	218	230	636	0	728	115	465	237	388	561	659
50	Kırklareli	596	934	1635	717	958	1410	728	0	852	276	807	511	1299	452
51	Kırşehir	337	276	961	332	126	532	115	852	0	579	222	458	457	738
52	Kocaeli	333	699	1372	454	695	1143	465	276	579	0	577	295	1028	386
53	Konya	462	134	1122	451	279	625	237	807	222	577	0	321	616	536
54	Kütahya	466	445	1385	549	574	939	388	511	458	295	321	0	924	322
55	Malatya	784	607	627	705	334	267	561	1299	457	1028	616	924	0	1152
56	Manisa	703	662	1742	824	811	1154	659	452	738	386	536	322	1152	0
57	Mardin	1121	809	616	1037	671	369	899	1673	795	1374	891	1202	342	1418
58	Mersin	668	240	1098	663	270	321	446	1128	332	857	346	656	434	873
59	Muğla	866	692	1716	938	864	1195	772	670	808	605	591	428	1212	212
60	Muş	1003	986	334	893	691	555	919	1525	815	1263	973	1281	358	1509
61	Neveşehir	424	236	944	424	100	475	202	923	87	652	181	501	440	718
62	Niğde	512	165	973	512	129	434	290	982	176	712	230	550	444	767
63	Ordu	576	761	596	444	474	717	495	1097	532	827	729	867	490	1141
64	Osmaniye	757	353	972	757	359	195	539	1217	421	955	435	745	304	962
65	Rize	821	954	377	689	665	842	740	1342	779	1072	937	1112	575	1386
66	Sakarya	253	627	739	374	615	1008	385	351	499	83	501	219	945	454
67	Samsun	429	689	738	304	449	761	355	957	412	687	589	727	584	1001
68	Siirt	1200	967	511	1070	760	528	987	1832	883	1463	1042	1360	427	1576
69	Sinop	308	755	903	182	533	890	389	895	496	624	623	721	712	1004
70	Sivas	566	510	631	482	221	465	383	1103	335	825	493	756	248	1029
71	Şanlıurfa	932	620	721	927	474	180	710	1484	606	1185	702	1012	216	1229
72	Şırnak	1286	974	624	1212	837	535	1064	1838	960	1539	1056	1367	516	1583
73	Tekirdağ	503	858	1542	624	865	1313	635	81	749	186	732	432	1196	544
74	Tokat	473	568	699	362	279	523	303	994	320	725	537	676	345	949
75	Trabzon	771	905	400	639	616	859	690	1293	729	1022	888	1063	523	1403
76	Tunceli	861	811	485	751	516	517	744	1383	640	1112	798	1096	183	1335
77	Uşak	568	477	1463	635	627	969	474	507	544	369	351	137	968	187
78	Van	1284	1163	349	1174	918	724	1145	1806	1041	1535	1199	1508	584	1736
79	Yalova	363	673	1402	483	724	1118	494	293	609	46	548	266	1055	359
80	Yozgat	334	375	887	298	164	551	118	843	98	572	320	496	485	769
81	Zonguldak	152	685	1289	267	611	1005	382	483	496	212	554	352	942	593

Çizelge 3.2. (Devam) OSB'ler arasındaki karayolu mesafe (km) değerleri

		57	58	59	60	61	62	63	64	65	66	67	68	69	70
		Mardin	Mersin	Muğla	Muş	Neveşehir	Niğde	Ordu	Osmaniye	Rize	Sakarya	Samsun	Siirt	Sinop	Sivas
42	Kahramanmaraş	418	255	1125	583	337	330	578	140	771	871	623	576	752	327
43	Karabük	1121	668	866	1003	424	512	576	757	821	253	429	1200	308	566
44	Karaman	809	240	692	986	236	165	761	353	954	627	689	967	755	510
45	Kars	616	1098	1716	334	944	973	596	972	377	739	738	511	903	631
46	Kastamonu	1037	663	938	893	424	512	444	757	689	374	304	1070	182	482
47	Kayseri	671	270	864	691	100	129	474	359	665	615	449	760	533	221
48	Kilis	369	321	1195	555	475	434	717	195	842	1008	761	528	890	465
49	Kırkkale	899	446	772	919	202	290	495	539	740	385	355	987	389	383
50	Kırklareli	1673	1128	670	1525	923	982	1097	1217	1342	351	957	1832	895	1103
51	Kırşehir	795	332	808	815	87	176	532	421	779	499	412	883	496	335
52	Kocaeli	1374	857	605	1263	652	712	827	955	1072	83	687	1463	624	825
53	Konya	891	346	591	973	181	230	729	435	937	501	589	1042	623	493
54	Kütahya	1202	656	428	1281	501	550	867	745	1112	219	727	1360	721	756
55	Malatya	342	434	1212	358	440	444	490	304	575	945	584	427	712	248
56	Manisa	1418	873	212	1509	718	767	1141	962	1386	454	1001	1576	1004	1029
57	Mardin	0	598	1456	303	749	711	795	472	632	1282	899	200	1028	581
58	Mersin	598	0	913	784	259	168	751	143	934	782	699	757	783	498
59	Muğla	1456	913	0	1641	775	796	1252	999	1497	677	1113	1614	1106	1087
60	Muş	303	784	1641	0	798	811	601	659	423	1172	727	178	856	560
61	Neveşehir	749	259	775	798	0	98	535	346	751	574	487	863	571	315
62	Niğde	711	168	796	811	98	0	595	255	779	635	542	869	626	343
63	Ordu	795	751	1252	601	535	595	0	715	246	748	143	801	309	304
64	Osmaniye	472	143	999	659	346	255	715	0	884	867	744	630	887	462
65	Rize	632	934	1497	423	751	779	246	884	0	993	388	605	554	442
66	Sakarya	1282	782	677	1172	574	635	748	867	993	0	608	1482	545	753
67	Samsun	899	699	1113	727	487	542	143	744	388	608	0	908	168	358
68	Siirt	200	757	1614	178	863	869	801	630	605	1482	908	0	1034	669
69	Sinop	1028	783	1106	856	571	626	309	887	554	545	168	1034	0	488
70	Sivas	581	498	1087	560	315	343	304	462	442	753	358	669	488	0
71	Şanlıurfa	196	410	1266	408	585	522	708	283	738	1134	806	375	936	473
72	Şırnak	184	764	1620	292	940	876	909	637	719	1489	1071	116	1201	755
73	Tekirdağ	1575	1032	760	1423	824	885	999	1118	1243	259	858	1624	796	1005
74	Tokat	679	556	1061	579	341	401	199	520	444	646	244	766	366	109
75	Trabzon	648	885	1448	439	701	730	196	823	51	943	339	617	509	391
76	Tunceli	318	618	1391	234	619	622	443	491	411	1033	563	364	718	372
77	Uşak	1231	688	291	1325	530	580	957	774	1202	341	816	1397	806	844
78	Van	448	953	1793	227	1021	1045	799	826	589	1456	957	268	1087	795
79	Yalova	1392	870	575	1282	662	723	858	956	1103	112	717	1483	655	864
80	Yozgat	819	407	881	808	186	250	431	493	659	493	315	906	400	260
81	Zonguldak	1279	779	810	1169	571	632	731	864	975	133	590	1370	445	752

Çizelge 3.2. (Devam) OSB'ler arasındaki karayolu mesafe (km) değerleri

		71	72	73	74	75	76	77	78	79	80	81
		Şanlıurfa	Şırnak	Tekirdağ	Tokat	Trabzon	Tunceli	Uşak	Van	Yalova	Yozgat	Zonguldak
42	Kahramanmaraş	229	583	1121	385	721	408	893	772	980	420	868
43	Karabük	932	1286	503	473	771	861	568	1284	363	334	152
44	Karaman	620	974	858	568	905	811	477	1163	673	375	685
45	Kars	721	624	1542	699	400	485	1463	349	1402	887	1289
46	Kastamonu	927	1212	624	362	639	751	635	1174	483	298	267
47	Kayseri	474	837	865	279	616	516	627	918	724	164	611
48	Kilis	180	535	1313	523	859	517	969	724	1118	551	1005
49	Kırkkale	710	1064	635	303	690	744	474	1145	494	118	382
50	Kırklareli	1484	1838	81	994	1293	1383	507	1806	293	843	483
51	Kırşehir	606	960	749	320	729	640	544	1041	609	98	496
52	Kocaeli	1185	1539	186	725	1022	1112	369	1535	46	572	212
53	Konya	702	1056	732	537	888	798	351	1199	548	320	554
54	Kütahya	1012	1367	432	676	1063	1096	137	1508	266	496	352
55	Malatya	216	516	1196	345	523	183	968	584	1055	485	942
56	Manisa	1229	1583	544	949	1403	1335	187	1736	359	769	593
57	Mardin	196	184	1575	679	648	318	1231	448	1392	819	1279
58	Mersin	410	764	1032	556	885	618	688	953	870	407	779
59	Muğla	1266	1620	760	1061	1448	1391	291	1793	575	881	810
60	Muş	408	292	1423	579	439	234	1325	227	1282	808	1169
61	Neveşehir	585	940	824	341	701	619	530	1021	662	186	571
62	Niğde	522	876	885	401	730	622	580	1045	723	250	632
63	Ordu	708	909	999	199	196	443	957	799	858	431	731
64	Osmaniye	283	637	1118	520	823	491	774	826	956	493	864
65	Rize	738	719	1243	444	51	411	1202	589	1103	659	975
66	Sakarya	1134	1489	259	646	943	1033	341	1456	112	493	133
67	Samsun	806	1071	858	244	339	563	816	957	717	315	590
68	Siirt	375	116	1624	766	617	364	1397	268	1483	906	1370
69	Sinop	936	1201	796	366	509	718	806	1087	655	400	445
70	Sivas	473	755	1005	109	391	372	844	795	864	260	752
71	Şanlıurfa	0	366	1379	568	722	373	1035	579	1217	628	1082
72	Şırnak	366	0	1740	853	730	482	1396	380	1600	990	1444
73	Tekirdağ	1379	1740	0	897	1195	1285	526	1708	165	745	379
74	Tokat	568	853	897	0	396	440	766	809	756	217	643
75	Trabzon	722	730	1195	396	0	380	1152	603	1053	610	926
76	Tunceli	373	482	1285	440	380	0	1150	476	1157	629	1034
77	Uşak	1035	1396	526	766	1152	1150	0	1549	339	583	477
78	Van	579	380	1708	809	603	476	1549	0	1566	1034	1453
79	Yalova	1217	1600	165	756	1053	1157	339	1566	0	601	250
80	Yozgat	628	990	745	217	610	629	583	1034	601	0	492
81	Zonguldak	1082	1444	379	643	926	1034	477	1453	250	492	0

### 3.1. Karınca Kolonisi Optimizasyonu ile 5 OSB için Algoritma Oluşturulması

Yapılan tez çalışmasında 81 adet organize sanayi bölgesi kullanılmıştır. Yapılan örnek çalışmada ise tez içerisinde kullanılan 5 organize sanayi bölgesi seçilerek algoritma hazırlanmaktadır. Bu rastgele seçilen organize sanayi bölgeleri; Manisa, Mardin, Mersin, Muğla ve Muş şehirlerinde bulunan organize sanayi bölgeleridir. Karınca kolonisi optimizasyonu kullanılarak yapılan bu örnek çalışmadaki amaç oluşturulan algoritmanın çalışma şeklini göstermektir. Burada yapılan karınca kolonisi optimizasyonunun kod yazımı 5 farklı bölümden oluşmaktadır. Bu bölüm dosyaları;

- ✓ ModelOlusturma.m
- ✓ Turmaliyet.m
- ✓ KKOkomutlari.m
- ✓ RuletCarki.m
- ✓ SonuclarinCizimi.m

olarak toplamda 5 tanedir. Bu dosyaların hepsinin birbirinden farklı olarak uzantıları ‘.m’ şeklinde olup bölümlerin isimlendirmesini kendimiz yapabilmekteyiz. “ModelOlusturma.m” uzantılı bölümde analizler için model oluşturulmaktadır. Verilerimizden programa işlediğimiz organize sanayi bölgelerinin x ve y koordinatları, toplam kaç şehir (nokta, düğüm) olduğunu ve girilen gerçek karayolu mesafe değerlerini matris formatında yaparak bünyesinde tutmaktadır. “ModelOlusturma.m” bölümüne ait kodlar Çizelge 3.3’te verilmektedir.

Çizelge 3.3. ModelOlusturma.m bölümündeki kod yazımı

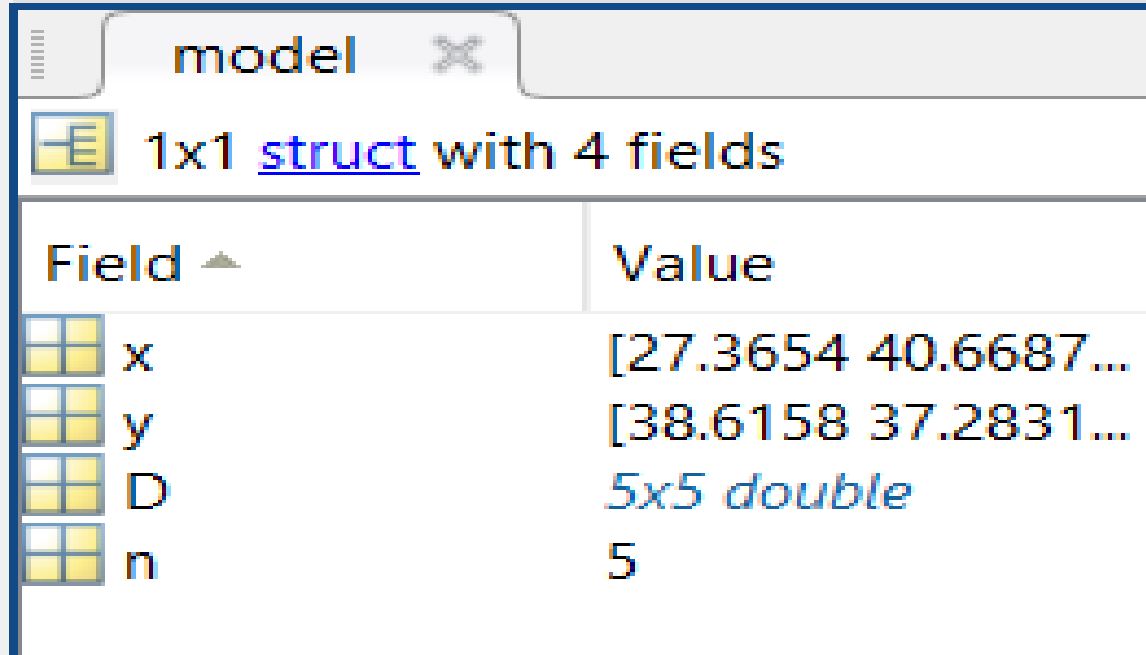
```
function model=ModelOlusturma()
x=[27.365395 40.668681 34.796679 27.783070 41.500370];
y=[38.615848 37.283114 36.899682 37.303062 38.747088];
n=numel(x);

D=[0 1418 873 212 1509;
  1418 0 598 1456 303;
  873 598 0 913 784;
  212 1456 913 0 1641;
  1509 303 784 1641 0]

model.x=x;
model.y=y;
model.D=D;
model.n=n;

end
```

Bu ‘‘ModelOlusturma’’ bölümündeki kod yazımını tamamlayıp çalıştırdıktan sonra sonuçlar model dosyası ana başlığı altında organize sanayi bölgelerinin x koordinatları (x), y koordinatları (y), toplam şehir sayısı (n) ve matris biçiminde yazılan gerçek karayolu mesafeleri (D) verilmektedir. Elde edilen bu çıktı Şekil 3.1’de, gerçek karayolu mesafelerinin matris biçiminde gösterimi ise Çizelge 3.4’te gösterilmektedir.



Şekil 3.1. ModelOlusturma.m bölümünün çıktı sonuçlarının dosya görünümü

Çizelge 3.4. ModelOlusturma.m çıktısındaki gerçek karayolu mesafeleri

Şehir	1	2	3	4	5
1	0	1418	873	212	1509
2	1418	0	598	1456	303
3	873	598	0	913	784
4	212	1456	913	0	1641
5	1509	303	784	1641	0

Turmaliyet.m isimli bölümde yazılan kodlamada bir şehirden başlanıp tüm şehirleri dolaştıktan sonra tekrar başlangıç şehrine gelinip turun tamamlanması ile oluşan toplam mesafenin (L) kod yazımı yapılmaktadır. Turmaliyet.m bölümdeki yazılan kodlama Çizelge 3.5’te gösterilmektedir.

Çizelge 3.5. Turmaliyet.m bölümündeki kod yazımı

```
function L=turmaliyet(model,tur)
n=numel(tur);
tur=[tur tur(1)];

L=0;
for i=1:n
    L=L+model.D(tur(i),tur(i+1));
end
end
```

Turmaliyet.m bölümündeki kodlama tamamlandıktan sonra KKO'nun 3. bölümü olan "KKOkomutlari.m" bölümüne geçilmektedir. Bu bölümle alakalı kodlar Çizelge 3.6'da gösterilmektedir. KKOkomutlari.m bölümü kısımlara ayrılarak çizelgeler şeklinde gösterilmektedir. Her çizelge sonrası o çizelge ile ilgili açıklama yapılmaktadır.

Çizelge 3.6. KKOkomutlari.m bölümündeki kod yazımı

```
%% Problemin Tanıtımı
model=ModelOlusturma();
mesafefonksiyonu=@(tur) turmaliyet(model,tur);
sehirsayisi =model.n;
```

Yazılan kodu çalıştırmak için bir model gereklidir ve bu model KKO'nun ilk bölümü olan "ModelOlusturma.m" bölümünde yapılmıştır. Burada modelimizi tekrar tanımlamak için "model=ModelOlusturma();" eklemesi yapıyoruz. Belirtilen kodlamada "mesafefonksiyonu" komutu maliyeti vermektedir. Maliyet ise çalışma içerisinde başlangıç şehirden itibaren her şehre bir kere uğrayarak tekrar başlanılan noktaya gelene kadar olan toplam mesafeye denilmektedir. Burada girdi 'tur' olup çıktısı ise maliyeti hesaplayan "turmaliyet(model,tur)" olmaktadır. Model için kaç şehir olduğu da kaydedilmesi de gerekmektedir. Veri setinde kaç şehir olduğu veya optimal değer kaç



şehir için bulunması gerektiği “sehirsayisi=model.n;” komutunda belirtilmektedir. Karınca kolonisi optimizasyonunun ilk bölümünde kaç şehir olduğu “model.n” komutunda kaydedilmişti. Burada “sehirsayisi” komutu oradan kaç şehir olduğu bilgisini alarak devam etmektedir. Bu kısım açıklandıktan sonra “KKOkomutlari.m” bölümü kod yazımının devamı Çizelge 3.6. (Devam) kısmında gösterilmektedir.

Çizelge 3.6. (Devam) KKOkomutlari.m bölümündeki kod yazımı

```
%% KKO Parametreleri

iterasyonsayisi=750;
karincasayisi=100;
eta=1./model.D;
alfa=1;
beta=1;
buharlasmaniktari=0.3;
feromonyogunlugu0=1;
feromonyogunlugu=feromonyogunlugu0*ones(sehirsayisi);
Q=1;
```

Karıncalar tur güzergahı için bir şehirden başladıktan sonra gideceği bir sonraki şehri de seçmeleri gerekmektedir. Hangi yoldan gitmesi gerektiğini, en kısa, en doğru yolun hangisi olduğuna karar vermelidir. Bu durumda karıncalar bir sonraki gidilecek olan şehrin yolunun seçilebilmesi için ilk olarak şehirlerarasındaki yollarda bulunan feromon yoğunluğuna bakmaktadır. Feromon yoğunluğunun fazla olması şehirlerarasındaki mesafenin daha kısa olduğu anlamına gelmektedir. Bu durumda karıncalar gideceği yolu seçerken ilk adım olarak feromon yoğunluğuna bakıp, komşu şehirlerarasındaki en yoğun olan yolu belirleyerek o yolu seçmektedir. Yani seçeceği şehir ile bulunduğu şehir arasındaki yoldaki feromon yoğunluğunun en fazla olması gerekmektedir. Feromon yoğunluğunu tanımlayan komut parametreler içerisinde “feromonyogunlugu” olarak ifade edilmektedir. Kodlama içerisinde diğer parametreler ise “iterasyonsayisi” çalışmada en

fazla kaç adet iterasyon yapılması gerektiğini, “karincasayisi” ise çalışma içerisinde bulunan karınca miktarını, Q ise sabit bir değer olarak tanımlanmaktadır.

Diğer bir faktör olarak karıncaların bulunduğu şehir ile gidebileceği komşu şehirler arasından en kısa mesafeye sahip olan şehri “model.D” kodundan oluşturduğumuz yerden alınarak belirlenmektedir. Karıncaların başka bir şehre doğru karar verip gidebilmesinde bu parametreler önemli yer almaktadır.

Kodlama içerisindeki diğer parametrelerden birisi de eta’dır. Eta yani sezgisel bilgiler, soru içerisinde elde edilen bilgilerdir. Sonuç itibarıyla karıncaların bir sonraki şehre geçme durumu bulunduğu şehirden geçeceği şehir arasındaki mesafe, gidebileceği diğer şehirlerden daha kısa olmalı ve aynı şekilde gideceği şehirle arasındaki yol güzergahında feromon yoğunluğu da daha fazlası olması gerekmektedir.

Bu kodlama içerisinde verilen parametrelerden birisi olan eta’nın formülü ise “[eta=(1/D)]” şeklindedir. Formülde belirtilen ‘D’ başlangıç ve gidilecek şehir arasındaki mesafeyi tanımlamaktadır. Bu durumda iki şehir arasındaki mesafe (D) ne kadar fazla olursa eta o kadar düşük, ‘D’ mesafesi ne kadar az olursa eta o kadar yüksek olmaktadır. Kodlama içerisinde tanımlanmayan diğer parametreler alfa ve beta’dır. Bu kodlama içerisinde alfa feromonun ağırlığı, beta ise eta sezgisel değişkeninin ağırlığı olarak tanımlanmaktadır.

Bir karınca daha önce farklı bir karıncanın kullandığı yoldan geçtiği zaman o karınca da aynı yola feromon maddesinden bırakarak ekleme yapmaktadır. Aynı zamanda bu kimyasal feromon maddesi buharlaşabilen bir madde olup zamana bağlı olarak azalmaktadır. Bu feromon maddesindeki buharlaşma miktarı kodlama içerisindeki parametrelerde “buharlaşmamiktari” olarak belirtilmektedir. Feromon yoğunluğunun ise kodlama içerisinde “feromonyogunlugu” olarak adlandırıldığı belirtilmiştir. Bu duruma bağlı olarak “feromonyogunlugu0” şehirlerarasındaki yollarda başlangıçta olan feromon değerini göstermektedir. Aynı zamanda feromon maddesi bütün yollarda ilk olarak eşit miktarda bırakılmaktadır. Bu eşit şekilde bırakılan miktara ek olarak karıncalar o yollardan geçtikçe onlarda feromon maddesi bırakarak genel miktarı arttırmaktadır. Kodlama da bu feromon miktarını gösteren komut ise;

“feromonyogunlugu=feromonyogunlugu0\*ones(sehirsayisi);” şeklinde tanımlanmaktadır. KKOkomutlari.m bölümündeki kod yazımının devamı Çizelge 3.6. (Devam) kısmında gösterilmektedir.

Çizelge 3.6. (Devam) KKOkomutlari.m bölümündeki kod yazımı

```
%% Komutları Başlatma Kısmı

%bos karınca

bos_karinca.tur=[];

bos_karinca.maliyet=[];

eniyikarinca.maliyet=inf;

karınca=repmat(bos_karinca,karincasayisi,1);
```

Bu bölümde “KKOkomutlari.m” bölümündeki komutların başlatma kısmının yazımı gösterilmektedir. Gezgini satıcı probleminde bir noktadan başlanıp tüm noktalara uğradıktan sonra tekrar başlangıç şehrine gelmesi ile tur tamamlanıyordu. Burada da aynı şekilde karıncalar bir şehirden başlayıp veri setindeki tüm şehirlere uğradıktan sonra tekrar ilk başladıkları şehre dönmektedir. Fakat tüm karıncalar aynı yol güzergahını kullanmamaktadır. Karıncalar çözüme ulaşıncaya kadar ki süreçte farklı çözümler farklı yol güzergahlarını kullanmaktadır. Buna bağlı olarak her karıncaya ait bir maliyet olmaktadır. Maliyet kavramı tamamlanan tur sonucunda elde edilen toplam mesafeye denilmektedir.

Az önce de belirtildiği gibi karıncalar farklı yol güzergahlarına sahiptirler. Yani her karıncanın farklı tur sıralaması olabilmektedir. Bu kapsamda “bos\_karinca.tur=[];” komutu karıncaların atacağı turun başlangıç şehirden itibaren sırasıyla hangi şehirleri ziyaret edeceğini gösterecek olan matrisin boş halidir. Diğer komut olan “bos\_karinca.maliyet=[];” ise atılacak bu turların toplam maliyetini yani toplam yol uzunluğunu gösterecek olan matrisin boş halidir. Bu durumda genel olarak tanımlamak gerekirse karıncalar için bu attıkları turlar ve toplam mesafe özellikleri kaydolacaktır. Bu

“bos\_karinca.tur=[];” ve “bos\_karinca.maliyet=[];” komutları sadece 1 karınca için yani 1 matris olarak oluşturulmaktadır. Bunun kodlamada belirlenen karınca sayısı kadar olması gerekmektedir. Her karınca için tur ve maliyet boş matrislerinin açılması “karınca=repmat(bos\_karinca,karincasayisi,1);” komutu ile yapılmaktadır. Bu komut sonrasında kodumuzu çalıştırdığımızda her karınca için tur ve maliyet boş matrisleri açılmış olacaktır. Bu boş olan matrisler atılan tur durumuna göre şehirlerin sırası ile kaydedilip daha önce kodu yazılmış olan “Turmaliyet.m” bölümündeki kod ile her karınca için maliyet hesaplanıp boş olan matrise yazılmaktadır. Kod kısmındaki “eniyikarinca.maliyet=inf;” bu komut ise karıncaların tur güzergahlarını tamamladıktan sonra elde edecekleri toplam mesafenin birbirleri ile karşılaştırıldığında en kısa yol mesafesine sahip olan karıncayı bünyesinde tutacak olan kodlamadır. Çizelge 3.6. (Devam) kısmında KKOkomutlari.m bölümündeki kod yazımının devam kısmı gösterilmektedir.

Çizelge 3.6. (Devam) KKOkomutlari.m bölümündeki kod yazımı

```

for it=1:iterasyonsayisi
    for k=1:karincasayisi
        karınca(k).tur=randi([1 sehirsayisi]);

        for l=2: sehirsayisi
            i=karınca(k).tur(end);
            P=feromonyogunlugu(i,:).^alfa.*eta(i,:).^beta;
            P(karınca(k).tur)=0;
            P=P/sum(P);
            j=RuletCarki(P);
            karınca(k).tur=[karınca(k).tur j];
        end
        karınca(k).maliyet=mesafefonksiyonu(karınca(k).tur);
        if karınca(k).maliyet< eniyikarınca.maliyet
            eniyikarınca =karınca(k);
        end
    end

    % Feromon Miktarlarının Güncellenmesi
    for k=1:karincasayisi
        tur=karınca(k).tur;
        tur=[tur tur(1)];
        for l=1: sehirsayisi
            i=tur(l);
            j=tur(l+1);
            feromonyogunlugu(i,j)=feromonyogunlugu(i,j)+Q/karınca(k).maliyet;
        end
    end

    % Buharlaşma
    feromonyogunlugu=(1-buharlasamiktari)*feromonyogunlugu;
    % En iyi sonuçların kaydedilmesi
    iterasyonuneniyisonucu(it)=eniyikarınca.maliyet;
    disp(['iterasyon :' num2str(it),' ', 'iterasyonuneniyisonucu ='
num2str(iterasyonuneniyisonucu(it))])
    SonuclarinCizimi(eniyikarınca.tur,model)
    pause=0.05;
end

```

Çizelge 3.7. RuletCarki.m bölümündeki kod yazımı

```
function j=RuletCarki(P)
r=rand;
C=cumsum(P);
j=find(r<=C,1,'first');

end
```

Boş matrisler hazırlanması sonucunda yani verilerin kaydedileceği yerler hazırlandıktan sonra o matrislere yazılacak olan sonuç verilerinin kod yazımına devam edilmektedir. Bu yapılan işlemlerden “for it=1:iterasyonsayisi” komutu yapılan işlemlerin parametrelerde belirlenen sayı tekrarı kadar yapılmasını söyleyen komuttur. Sistemde karıncaların başlangıç şehrini seçme durumu ile alakalı bir kısıtlama bulunmamaktadır. Yani karıncalar başlangıç şehirlerini seçerken herhangi bir duruma bağlı kalmadan rastgele seçmektedir. Yapılacak olan tura herhangi bir şehirden başlayan karıncalar bir sonraki gideceği şehri belirlemek için feromon yoğunluğu ve eta değişkenini benimseyerek ilerlemektedir. Veri setinde bulunan her karınca için anlamına gelen kod ise “for k=1:karincasayisi” şeklinde yazılmaktadır.

Karıncaların başlangıç şehrini herhangi bir faktöre bağlı kalmadan rastgele seçildiği belirtilmişti. Karıncaların rastgele belirleyeceği şehrin komutu ise “karınca(k).tur=randi([1 sehirsayisi]);” şeklinde tanımlanmaktadır. Burada kod içerisindeki yazılan “sehirsayisi” komutu veri setinde toplam kaç adet şehir olduğunu belirtmektedir. Diğer bir komut olan “randi” komutu ise tüm şehirlerarasından başlangıç şehri için rastgele 1 tane şehir seçilmesini isteyen komuttur. Buna bağlı olarak “karınca(k).tur=randi([1 sehirsayisi]);” kodu karıncaların ilk başlangıç şehirlerini belirleyen kod yazımıdır. Başlangıç şehri belirlenen karıncanın bir sonraki şehri de seçmesi gerekmektedir.

İlk şehrin seçilmesi sonrasında kalan şehirler için yazılan kodlama “for l=2: sehirsayisi” şeklinde olmaktadır. Herhangi bir karınca yani temsili olarak ‘k’ karıncası olarak tanımlayalım. Bulunduğu şehri ise ‘i’ olarak tanımlayalım. Bulunduğu bu ‘i’ şehirden karıncanın bir sonraki şehri seçmesi gerekmektedir. Karıncanın son olarak geldiği yani ‘i’

şehrine gelme durumunu belirten kodlama “ $i=karınca(k).tur(end);$ ” şeklinde olmaktadır. Yani bu durum karıncanın son olarak ‘i’ şehrine geldiği ve devamında gideceği şehrin bu şehirden farklı olması gerektiğini göstermektedir. Önceki açıklamalarda karıncaların her şehre 1 defa uğrama şartı belirtilmişti aynı zamanda bulunduğu şehri de seçmemesi gerektiği de belirtilmelidir. Yani bir sonraki seçim sırasında kendi bulunduğu şehri seçme olasılığı ‘0’ olmalıdır. Buna bağlı olarak “ $P(karınca(k).tur)=0;$ ” komutu karıncanın gideceği bir sonraki şehrin şu an bulunduğu şehir olma olasılığını sıfır yapan kodlama olarak yazılmaktadır.

Karıncaların gideceği bir sonraki şehirleri belirlemedeki en büyük 2 etken feromon yoğunluğu ve sezgisel bilgilerdir. Buna bağlı olarak eldeki verileri yazacak olursak; “ $P=feromonyogunlugu(i,:).^alfa.*eta(i,:).^beta;$ ” yani feromon ve eta için ‘i’ şehirden başlanıp, kalan tüm şehirler ile olan durumları ele alınarak katsayılarının da eklemesinden sonra birbirleri ile çarpılmaktadır. Bu işlemde normalizasyon yapmak için bu verilen feromon ve eta değişkenlerinin toplamlarına bölünmesi gerekmektedir. Bunu yapan kodlama ise; “ $P=P/sum(P);$ ” şeklinde olmaktadır. Burada ‘P’ karıncaların bulunduğu şehirden bir sonraki şehri seçme olasılığını ifade etmektedir.

Feromon yoğunluğu ile etanın fazla olması o şehre kesin olarak gidileceği anlamına gelmemektedir. Fakat bu iki değişkenin karıncanın gideceği bir sonraki şehri seçme olasılığını önemli ölçüde arttırmaktadır. Karıncaların bir sonraki gideceği şehri seçme durumu “RuletCarki.m” bölümündeki kodlama ile yapılmaktadır. Burada gideceği şehirlerarasında feromon yoğunluğu ve eta değişkenlerinin etkisiyle birlikte seçiminin olasılığa bağlı olduğu bir çözüm üretilmektedir. Feromon yoğunluğunun ve eta miktarının fazla olması her ne kadar o şehrin seçilme olasılığını arttırsa da kesin olarak seçileceği anlamı taşımamaktadır.

5 bölümden oluşan Karınca kolonisi optimizasyonunun 4. Sırasında “RuletCarki.m” bölümü bulunmaktadır. Bu bölüm genel olarak bakıldığında olasılıklar dahilinde karıncaların bulunduğu şehirden bir sonraki gideceği şehre karar verdiği kodlamanın olduğu bölümdür. Kodlama içerisinde ‘r’ değişkeninin ‘0’ ile ‘1’ aralığında olması gerekmektedir. Bunu sağlayan komut ise “ $r=rand;$ ” komutu olup, “ $C=cumsum(P);$ ” komutu ise P’lerin toplanıp ‘C’ içerisine yazıldığı komuttur. Kod içerisinde yazılan “ $j=find(r<=C,1,'first');$ ” komutu; r’nin j’den küçük olduğu ilk yerin karıncanın bir sonraki

gideceği şehri seçmesi olarak tanımlanmaktadır. Bu seçim gidilecek olan şehrin feromon yoğunluğu ve eta miktarına göre her şehrin olasılık değerleri alınıp sonrasında olasılık miktarları dahilinde rastgele yapılan seçimler ile oluşturulmuştur. Olasılığı yüksek şehirlerin seçilme durumu daha yüksek olsa da düşük olan şehirlerin de seçilme durumunun olduğu bir kodlamadır.

Kodlama sonucunda elde edilen j. şehir karıncanın bir sonraki gitmeye karar verdiği şehirdir. Karıncanın bir sonraki gideceği şehri başlangıç şehrine ekleyen yani o karıncanın turunu oluşturan kod “karınca(k).tur=[karınca(k).tur j]; ” olarak yazılmaktadır. Yazılan “j=RuletCarki(P)” kodu sonrası karıncalar bir sonraki gideceği şehre karar vermektedir. Yani gideceği şehirleri belli olan karınca belirli bir yol mesafesi ilerlemektedir. Karıncaların oluşturdukları turların toplam yol mesafesini belirleyen komut ise; “karınca(k).maliyet=mesafefonksiyonu(karınca(k).tur);” olarak yazılmaktadır. Bu kodlamalar sonucunda karıncalar belirtilen kurallar çerçevesinde tüm şehirlere uğrayarak, her bir karınca için tur sıralaması ve toplam yol uzunluğu belirlenmektedir. Her karıncanın farklı bir yol güzergahı ve buna bağlı olarak farklı bir toplam yol mesafesi mevcuttur. Her karınca için ayrı olacak şekilde sonuçlar bulunup farklı matrislerde kaydedilmektedir. Bu karıncalar arasından en kısa mesafe ve en iyi karıncayı bulabilmek için yazılan kod; “if karınca(k).maliyet< eniyikarınca.maliyet”, “eniyikarınca=karınca(k);” şeklinde olmaktadır. Bu veriler kapsamında ilk karıncanın elde ettiği mesafe başlangıçta en kısa mesafe olarak baz alınmaktadır. Sonrasında ise diğer karıncaların her birinin “eniyikarınca” verisinde kaydolun sonuç ile karşılaştırılması yapılmaktadır. Karşılaştırma sonucunda diğer karıncanın en kısa mesafesi “eniyikarınca” verisinde bulunan karıncadan daha fazla ise “eniyikarınca” verisinde bulunan karınca aynı şekilde kalır, eğer diğer karıncanın mesafesi “eniyikarınca” verisinde bulunan karıncadan daha kısa mesafe sonucu vermişse o karınca “eniyikarınca” olarak kabul edilir ve “eniyikarınca” veri kısmına o sonucu elde eden karınca yazılır. Devamında ise kalan karıncalar yine “eniyikarınca” kısmında bulunan karınca ile karşılaştırılmaya devam edilmektedir. Bu şekilde tüm karıncalar ile karşılaştırma yapıp en ideal sonuç bulunmaktadır.

Bu yapılan kod yerine kadar genel olarak karıncaların matris oluşturma, şehir seçme, tur atma, en iyi tur güzergahı ve en kısa mesafeyi belirleme vb. çözümlerin kod yazımı yapılmaktadır. Fakat karıncaların diğer şehirlere giderken rastgele değil, belirli değişkenlere bağlı olarak ilerlediği bilinmektedir. Karıncalar geçtiği yollara feromon



maddesi bırakılmaktadır. “%Feromon miktarlarının güncellenmesi” kısmında ise bu feromon maddesinin yazılan koda ekleme işlemi yapılmaktadır. Bu feromon maddesinin eklenmesi sistemde bulunan tüm karıncalar için yapılması gerekmektedir. Sistemde bulunan tüm karıncaları ele alan komut “k=1:karincasayisi” şeklinde yazılmaktadır. Karıncaların hangi güzergahı kullandığı, toplam tur durumlarını yapan kodlama “tur=karınca(k).tur;” şeklinde yazılmaktadır. Yapılan örnek çalışmada GSP mantığında olduğu gibi başlanılan şehre tekrar dönülmesi gerekmektedir. Karıncaların sistemde bulunan tüm şehirlere uğradıktan sonra, tekrar başlangıç noktasına dönmesini sağlayan kodlama “tur=[tur tur(1)];” olarak yazılmaktadır. Ekleme yapılan feromon maddesi her yola bırakılmaktadır. Kodlamada bu feromon maddesinin yollara bırakılmasını sağlamak için ilk bilgi olarak veri setinde kaydedilmiş olan şehir sayısının bilinmesi gerekmektedir. Toplam şehir sayısını veren komut ise; “l=1: sehirsayisi” şeklinde olmaktadır. Kod içerisinde gidilecek olan şehirlerle ilgili yapılan düzenleme kodları; “i=tur(l);” ve “j=tur(l+1);” şeklindedir. Bu bilgiler kapsamında şehirlerarasındaki feromon miktarı; “feromonyogunlugu(i,j)=feromonyogunlugu(i,j)+Q/karınca(k).maliyet;” kodu ile yazılmaktadır.

Kimyasal feromon maddesi buharlaşabilen bir maddedir. Bu kapsamda kimyasal feromon maddesinin buharlaşma durumunun da sisteme eklenmesi gerekmektedir. Kod yazımında bulunan “% Buharlaşma” kısmında bu kimyasal feromon maddesinin buharlaşma miktarının kodu yazılmaktadır. Feromon maddesinin buharlaşma miktarı “feromonyogunlugu=(1-buharlaşmamiktari)\*feromonyogunlugu;” kodu ile yapılmaktadır.

Kodlama içerisinde “iterasyonsayisi” yani en ideal sonucu bulmak için yapılan iterasyonların miktarını belirten komut vardır. Bu komut ile her iterasyon sonucunda o iterasyona ait en iyi sonuç bulunmaktadır. “%En iyi sonuçların kaydedilmesi” kısmında her iterasyonun bitimi için o iterasyonun en iyi sonucunun bir sonraki iterasyon yapıldıktan sonra da kaybedilmemesi gerekmektedir. Bunu sağlayan komut; “iterasyonuneniyisonucu(it)=eniyikarınca.maliyet;” şeklinde yazılmaktadır. İterasyonlar ile bulunan mesafeler minimize edilmeye çalışılıp, en ideal mesafenin bulunması amaçlanmaktadır. İterasyonlar sonucunda her iterasyona ait en ideal sonucu veren kod; “disp(['iterasyon :' num2str(it),' ', 'iterasyonuneniyisonucu =' num2str(iterasyonuneniyisonucu(it))])” olarak yazılmaktadır.

Bu yapılan iterasyonlar için her iterasyonun sonunda karıncanın elde edilen en iyi sonucun grafiksel çizimini yapan kod; “SonuclarinCizimi(eniyyikarinca.tur,model)” olarak yazılmaktadır. Devamında ise “KKOkomutlari.m” bölümündeki son komut “pause=0.05;” komutu; iterasyonların işlem hızını ayarlayan, daha yavaş yapılmasını sağlayan komuttur. Çizelge 3.8’de “SonuclarinCizimi.m” kısmının kod yazımı gösterilmektedir.

Çizelge 3.8. SonuclarinCizimi.m bölümündeki kod yazımı

```
function SonuclarinCizimi(tur,model)

tur=[tur tur];
plot(model.x(tur),model.y(tur),'k-s')

end
```

Bu bölüm karınca kolonisi optimizasyonu kod yazımının son bölümüdür. Bu bölümde karıncaların attıkları turlar baz alınarak gittiği güzergahlar, modele eklenmiş olan koordinatlardan alınarak grafik üzerinde yapılan tur güzergahı gösterilmektedir. Karıncalar tüm şehirlere uğrayıp turunu tamamladıktan sonra tekrar başladığı noktaya dönmektedir. Tur sonunda başladığı noktayı ekleyen kodlama; “tur=[tur tur];” olarak yazılmaktadır. Bu durumun grafiksel gösterimi için şehirlerin koordinatları alınıp karıncanın tur güzergahına göre çizim yapılmasını sağlayan kod; “plot(model.x(tur),model.y(tur),'k-s')” olarak yazılmaktadır.

### 3.2. Benzetilmiş Tavlama Algoritması ile 5 OSB için Algoritma Oluşturulması

Benzetilmiş tavlama algoritması, karınca kolonisi optimizasyonu yönteminde olduğu gibi 81 ildeki aynı organize sanayi bölgeleri kullanılarak en ideal mesafeyi bulmak için çalışma yapılmaktadır. Bu benzetilmiş tavlama algoritmasında, karınca kolonisi optimizasyonunda yapıldığı gibi 5 şehir alınarak bir çözüm algoritması oluşturulmaktadır. Alınan şehirler sırası ile Manisa, Mardin, Mersin, Muğla ve Muş şeklindedir. Yapılan örnek çalışma benzetilmiş tavlama algoritmasının kod yazımı ve çalışma şeklini tanıtmak için

yapılmaktadır. Yapılan bu benzetilmiş tavlama algoritması 6 bölümden oluşmaktadır. Bu bölümler;

- ✓ ModelOlusturma.m
- ✓ MesafeHesabi.m
- ✓ RastgeleCozumOlusturma.m
- ✓ BTAkomutlari.m
- ✓ KomsuCevapOlusturma.m
- ✓ TurCizimi.m

şeklinde isimlendirilmektedir. Bu BTA bölümlerinin tümü “.m” uzantılı olmaktadır. Bu bölümlerdeki kodlamalar çizelge şeklinde ilerleyen sayfalarda gösterilmektedir ve her çizelge sonrası o çizelgede bulunan kodlar ile ilgili açıklamalar yapılmaktadır. Bu kapsamda modelin oluşturulduğu “ModelOlusturma.m” bölümü Çizelge 3.9’da verilmektedir.

Çizelge 3.9. ModelOlusturma.m bölümündeki kod yazımı

```
function model=ModelOlusturma()
x=[27.365395 40.668681 34.796679 27.783070 41.500370];
y=[38.615848 37.283114 36.899682 37.303062 38.747088];
n=numel(x);
D=[0 1418 873 212 1509;
  1418 0 598 1456 303;
  873 598 0 913 784;
  212 1456 913 0 1641;
  1509 303 784 1641 0]

model.n=n;
model.D=D;
model.x=x;
model.y=y;

end
```

Bu “ModelOlusturma.m” bölümü benzetilmiş tavlama algoritmasında kod yazımının ilk bölümü olmaktadır. Burada çözüme ulaşabilmek için bir model oluşturulmaktadır. Bu bölümde model içerisinde verilen şehirler ile ilgili bilgiler yer almaktadır. Burada veri setinde bulunan şehirlerin x koordinatları (x), y koordinatları (y), modelde bulunan toplam şehir sayısı (n) ve şehirlerin birbirleri arasındaki mesafeleri (D) yer almaktadır. Kodlamada yazılan bu bilgiler model başlığı altında kaydedilmektedir. Bu bağlamda şehirlerin x koordinatları “model.x=x;”, şehirlerin y koordinatları “model.y=y;”, toplam şehir sayısı

“model.n=n;” ve şehirlerin birbirleri arasındaki mesafe ise “model.D=D;” komutları ile kaydedilmektedir.

Çizelge 3.10. MesafeHesabi.m bölümündeki kod yazımı

```
function z=MesafeHesabi(model,tur)

D=model.D;
n=numel(tur);
L=0;

tur=[tur tur(1)];
for i=1:n
    L=L+D(tur(i),tur(i+1));
end
z=L;
end
```

Bu “MesafeHesabi.m” bölümünde hazırlanan modele ve gidilen tur güzergahı verilerine göre toplam mesafe miktarı elde etmek için yapılmaktadır. Kodlarda yazılmış olan “D=model.D;” kısmı şehirlerin birbirleri ile olan mesafeleri olup, “n=numel(tur);” ise modelde bulunan toplam şehir sayısını ifade etmektedir. Tur başlangıcında herhangi bir ilerleme olmadığından dolayı gidilen mesafe ‘0’ olmaktadır. Bu durumu ifade eden komut ise “L=0;” olarak verilmektedir. Gezgin satıcı probleminde modelde bulunan tüm şehirlere uğranıp tur tamamlandıktan sonra tekrar başlangıç şehrine dönülmesi gerekmektedir. Bu kodlamada tüm şehirlere uğranıp, tur tamamlandıktan sonra tekrar başlangıç şehrine dönme durumunu ifade eden komut; “tur=[tur tur(1)];” olarak yazılmaktadır. Tura herhangi bir şehirden başlanıp modeldeki tüm şehirlere uğranacağını belirten komut ise “i=1:n” olarak ifade edilmektedir. Tur tamamlanması için şehirlere uğrandıkça gidilen toplam mesafe miktarı artmaktadır. Tüm şehirlere uğrandıktan sonra tur tamamlanıp ilk başlanılan şehre dönüldüğünde kat edilen toplam mesafe miktarı “L=L+D(tur(i),tur(i+1));”komutu ile elde edilmektedir. Buradaki “L” kodlamada toplam mesafe olarak ifade edilmektedir. Diğer bir komut olan “z=L;” ifadesi ise z’nin L ile aynı anlamda toplam tur mesafesini ifade ettiğini belirtmektedir.

Bu kodlamalar kapsamında genel anlamda bir model elde edilmiş olmaktadır. Bu model için öncelikle belirli bir kurala bağlı kalmadan herhangi rastgele bir tur güzergahında sonuç elde etmek için “RastgeleCozumOlusturma.m” bölümündeki komutlar kullanılmaktadır. Bu bölüme ait kodlar Çizelge 3.11’de gösterilmektedir.

Çizelge 3.11. RastgeleCozumOlusturma.m bölümündeki kod yazımı

```
function tur=RastgeleCozumOlusturma(model)

n=model.n;

tur=randperm(n);

end
```

Yazılan komutlar arasından “n=model.n;” modelde bulunan şehir sayısını ifade etmektedir. Rastgele bir tur güzergahı oluşturmak için ise “tur=randperm(n);” komutu kullanılmaktadır.

Bu bölümden sonra benzetilmiş tavlama algoritmasının bir diğer bölümü olan “BTAkomutlari.m” bölümüne geçilmektedir. Bu bölüme ait kodlar çizelgelere ayrılıp açıklanmaktadır. “BTAkomutlari.m” bölümü kodlamaları Çizelge 3.12’de gösterilmektedir. Her çizelge sonrası o çizelge ile ilgili açıklamalar yapılmaktadır.

Çizelge 3.12. BTAkomutlari.m bölümünün kod yazımı

```
%% Problemin Tanıtımı

model=ModelOlusturma()

mesafefonksiyonu=@(tur) MesafeHesabi(model,tur);
```

Gezgin satıcı problemi baz alınarak yapılan örnek çalışma kapsamında; bu bölümde problemin programa tanıtılma işlemi yapılmaktadır. Yazılan kodlama sonucunda elde edilen modelin komutu “model=ModelOlusturma()” şeklinde olmaktadır. Daha önce

oluşturulmuş olan model problemin tanıtımı bölümüne aktarılmaktadır. Çalışma kapsamında her tur güzergahı tamamlandığında o tur güzergahına ait toplam mesafe miktarını veren komut; “mesafefonksiyonu=@(tur) MesafeHesabi(model,tur);” olarak yazılmaktadır. Çizelge 3.12. (Devam) kısmında “BTAkomutlari.m” bölümü kod yazımının devam kısmı gösterilmektedir.

Çizelge 3.12. (Devam) BTAkomutlari.m bölümünün kod yazımı

```
%% BTA Parametreleri
iterasyonsayisi=750;
sicaklikdegisimisonrasitursayisi=20;
T0=1250;
```

BTAkomutlari.m bölümünün bu kısımda benzetilmiş tavlama algoritmasında örnek problem çözümü için hazırlanan algoritmadaki parametreler tanımlanmaktadır. Bu değişken parametrelerden “iterasyonsayisi” algoritma içerisinde en fazla kaç defa iterasyon yapılacağını belirten kodlama olarak tanımlanmaktadır. Sıcaklık değişimi sonrası atılacak tur sayısını belirten komut ise “sicaklikdegisimisonrasitursayisi” şeklinde tanımlanmaktadır. Algoritmadaki diğer parametrelerden birisi de sistemin başlangıç sıcaklığını göstermektedir. Bu kapsamda sistemin başlangıç sıcaklığını gösteren komut “T0” olarak yazılmaktadır. Çizelge 3.12. (Devam) kısmında “BTAkomutlari.m” bölümünün devam kısmı gösterilmektedir.

Çizelge 3.12. (Devam) BTAkomutlari.m bölümünün kod yazımı

```
%% Komutları Başlatma Kısmı

sonuc.turguzergahi=RastgeleCozumOlusturma(model);
sonuc.mesafe=mesafefonksiyonu(sonuc.turguzergahi);

eniyisonuc=sonuc;
iterasyonuneniyisonucu=zeros(iterasyonsayisi,1);

T=T0;
```

Bu kısım algoritma için çalıştırma, başlatma kısmı olarak tanımlanabilmektedir. Önceki komutlarda rastgele bir tur güzergahını elde edecek komut gösterilmektedir. Burada “sonuc.turguzergahi=RastgeleCozumOlusturma(model);” komutunda rastgele seçilen bir tur güzergahının “sonuc.turguzergahi” ile eşit olduğu söylenmektedir. Rastgele seçilen bu tür güzergahı tamamlandığında toplamda belirli bir mesafe kat edilmiş olmaktadır. Atılan turlar için kat edilen toplam mesafe; “sonuc.mesafe=mesafefonksiyonu(sonuc.turguzergahi);” komutu ile tanımlanmaktadır. Her turun güzergahı ve bu turlara göre alınan toplam mesafe miktarları birbirlerinden farklı olmaktadır. Bu turlar arasından en ideal turu kaydeden komut “eniyisonuc=sonuc;” olarak yazılmaktadır. Bu kapsamda en kısa mesafenin kaydedildiği komut ise; “iterasyonuneniyisonucu=zeros(iterasyonsayisi,1);” şeklinde olmaktadır. Benzetilmiş tavlama algoritmasına genel olarak bakıldığında sıcaklığın temel faktörlerden birisi olduğu görülmektedir. Bu kapsamda sistemde sıcaklık değişimleri olmakta ve buna bağlı olarak çözümler elde edilmektedir. Sistemin başlangıç sıcaklığını belirten parametre önceki kısımda belirtilmektedir. Sistemdeki sıcaklık ile başlangıç sıcaklığı ilk durumda herhangi bir işlem yapılmadığı için eşit olmaktadır. Bu durumu belirten komut ise “T=T0;” şeklinde olmaktadır. Çizelge 3.12. (Devam) kısmında “BTAkomutlari.m” bölümü kod yazımının devam kısmı gösterilmektedir.

Çizelge 3.12. (Devam) BTAkomutlari.m bölümünün kod yazımı

```

%% Döngü Kısmı
for it=1:iterasyonsayisi
    for it2=1:sicaklikdegisimisonrasitursayisi

yenisonuc.turguzergahi=KomsuCevapOlusturma(sonuc.turguzergahi);
    yenisonuc.mesafe=mesafefonksiyonu(yenisonuc.turguzergahi);
    if yenisonuc.mesafe<sonuc.mesafe
        sonuc=yenisonuc;
    else
        delta=yenisonuc.mesafe-sonuc.mesafe;
        p=exp(-delta/T);
        if rand<p
            sonuc=yenisonuc;
        end
    end
end

    if sonuc.mesafe<enyisonuc.mesafe
        enyisonuc=sonuc;
    end
end
T=0.99*T;
    iterasyonuneniyisonucu(it)=enyisonuc.mesafe;

    disp(['Iteration' num2str(it) ': iterasyonuneniyisonucu= '
num2str(iterasyonuneniyisonucu(it))]);

    figure(1);
    TurCizimi(eniyisonuc.turguzergahi,model)

end

%% Figürlerin Çizimi
figure;
plot(iterasyonuneniyisonucu);

```



Bu kısmı kodlamanın döngü kısmı olarak adlandırmak mümkündür. Sıcaklık ve iterasyon parametrelerine göre döngüler “for it=1:iterasyonsayisi” ve “for it2=1: sicaklikdegisimisonrasitursayisi” komutlarında yazılmaktadır. Döngülerin devamında sıcaklığın azalmasını sağlayan komut ise “T=0.99\*T;” şeklinde olmaktadır. Çizelge 3.13’te “KomsuCevapOlusturma.m” bölümünün kod yazımı gösterilmektedir.

Çizelge 3.13. KomsuCevapOlusturma.m bölümünün kod yazımı

```
function xyeni=KomsuCevapOlusturma(x)

n=numel(x);
i=randperm(n);
i1=i(1);
i2=i(2);

xyeni=x;
xyeni([i1 i2])=xyeni([i2 i1]);

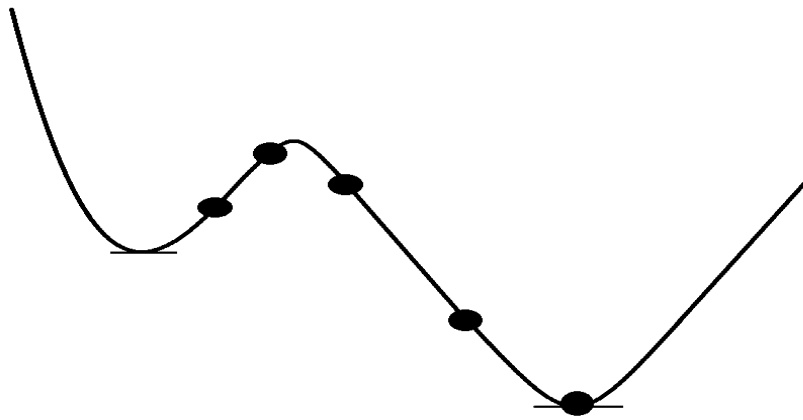
end
```

Elde edilmiş olan sonuçları en ideal hale getirmek için yöntemler bulunmaktadır. Bu yöntemlerin biri tanesi elde edilen yol güzergahındaki şehirlerin yer değiştirmesi ile elde edilmektedir. Bu yöntemde yol güzergahındaki 2 şehir yer değiştirilmektedir. Yani sırasıyla a b c d e şeklinde elde edilen bir güzergahın bu yöntem ile güzergah üzerinde 2 şehrin yer değiştirmesiyle a d c b e şeklinde yeni bir güzergah oluşturulup elde edilen yeni toplam mesafeye bakılarak işlem yapılmaktadır. Bu bağlamda elde edilen değişim sonucunda toplam tur güzergahı mesafesi daha iyi sonuçlar verebileceği gibi daha kötü sonuçlarda verebilmektedir. Bu durumun program üzerinde kodlamasının yapılması gerekmektedir. Bu durumun komutları Çizelge 3.13’te “KomsuCevapOlusturma.m” bölümünde gösterilmektedir. Veri setinde bulunan şehir sayısını gösteren komut; “n=numel(x);” şeklinde olmaktadır. Belirtilen tur güzergahı üzerindeki 2 şehrin yer değiştirmesini sağlayan komutlar; “i=randperm(n); i1=i(1); i2=i(2);” şeklinde yazılmaktadır. Başlangıçtaki x ile değişim sonrası x aynı olmaktadır. Sadece tur güzergahındaki yerlerin durumunun değiştiğini belirten komutlar; “xyeni=x; ve xyeni([i1

$i2]=xyeni([i2 i1]);$ ” olarak tanımlanmaktadır. Yani bu bölüm genel olarak oluşturulmuş olan tur güzergahında bulunan 2 şehrin yerinin değiştirilmesi olarak tanımlanmaktadır.

Bu bölümü tanımladıktan sonra “BTAkomutlari.m” bölümü komutları açıklamalarına devam edilmektedir. Çizelge 3.13’te belirtilen komutlarla tur güzergahında şehirlerin yer değiştirmesi ile yeni bir tur güzergahı oluşmaktadır. Bu güzergahın belirlenip kaydedilmesi; “yenisonuc.turguzergahi=KomsuCevapOlusturma(sonuc.turguzergahi);” komutu ile yapılmaktadır. Bu durum yeni bir tur güzergahın oluştuğunu ve yeni bir toplam mesafe elde edileceği anlamına gelmektedir. Bu yeni tur güzergahının toplam mesafesi ise “yenisonuc.mesafe=mesafefonksiyonu(yenisonuc.turguzergahi);” komutu ile bulunmaktadır. Tur güzergahındaki şehirlerin değişimi sonrasında daha iyi sonuçlar elde edilebileceği gibi daha kötü sonuçlarda elde edilebilmektedir. Bu durum için tur güzergahı sonrası toplam mesafelerinde karşılaştırılması gerekmektedir. Elde edilen bu yeni tur güzergahının toplam mesafesi önceki sonuçtan daha iyi çıkarsa bu sonucun kabul edilmesini belirten komut; “yenisonuc.mesafe<sonuc.mesafe ve sonuc=yenisonuc;” şeklinde olmaktadır.

Eğer bulunan yeni tur mesafesi daha önce elde edilen mesafeden daha yüksek çıkarsa; direkt reddedilmemesi gerekmektedir. Elde edilen bu sonuçlar şartlı bir şekilde kabul edilebilmektedir. Bir grafik üzerinde yerel minimum değeri ve global minimum değerinin gösterimi Şekil 3.2’de gösterilmektedir.



Şekil 3.2. Yerel minimum ile global minimum gösterimi

Bir yerel minimum değeri bulunduktan sonra elde edilen daha yüksek sonuçlar global minimum değerini bulabilmek için şartlı olarak kabul edilebilmektedir. Şekilde de görüldüğü gibi yerel minimum sonrası bulunan yüksek değerler bizi ideal sonuca yani global minimum değerine götüren faktörler olabilmektedir. Bu duruma bağlı olarak şartlı bir şekilde kabul edilebilmektedir. Sonucun önceki sonuçtan daha yüksek çıktığı durumda kabul edilebilme durumu Boltzmann dağılımı kullanarak belirlenmektedir. Belirtilen bu faktörleri algoritmaya eklemek gerekmektedir. Boltzmann dağılımını tanımlayan komutlar “ $\Delta = y_{enisonuc} \cdot mesafe - sonuc \cdot mesafe$ ; ve  $p = \exp(-\Delta/T)$ ;” şeklinde olmaktadır. Elde edilen sonuçlarda önceki çözümlerden daha yüksek çözüm sonucundaki durumlarda bu sonucun kabul edilebilme durumunu belirleyen komutlar; “ $rand < p$  ve  $sonuc = y_{enisonuc}$ ;” şeklinde yazılmaktadır. Elde edilen sonuçlar en iyi sonuçla da karşılaştırılmaktadır. Bu karşılaştırılma yapıldığında bulunan sonuç en iyi sonuçtan daha düşük bir sonuç değeri vermişse o en iyi sonuç olarak kabul edilmektedir. Bu durumu yapan komutlar ise; “ $sonuc \cdot mesafe < eniyisonuc \cdot mesafe$  ve  $eniyisonuc = sonuc$ ;” olarak yazılmaktadır. Çalışma içerisindeki parametrelerden birisi de iterasyon sayısıdır. Belirtilen “iterasyonsayisi” sayısı kadar algoritma içerisinde iterasyon yapılmaktadır. Her iterasyon bitiminde bir en iyi cevap elde edilmektedir. Her iterasyon sonrası en iyi cevabı belirten kod ise “ $iterasyonuneniyisonucu(it) = eniyisonuc \cdot mesafe$ ;” şeklinde yazılmaktadır. Yapılan iterasyonlardaki en iyi sonucu ve o sonucun hangi iterasyonda elde edildiği “ $disp(['Iteration' \quad num2str(it) \quad ': \quad iterasyonuneniyisonucu = \quad ' \quad num2str(iterasyonuneniyisonucu(it))]);$ ” komutu ile belirlenmektedir. Bu elde edilen sonuçlar grafiksel biçimde de gösterilebilmektedir. Bulunan en iyi sonuçlara göre grafiksel biçimde gösterim; “ $figure$ ; ve  $plot(iterasyonuneniyisonucu)$ ;” komutları ile yapılmaktadır. En iyi sonucun tur güzergahı çizimini yapan komut ise “ $TurCizimi(eniyisonuc.turguzergahi,model)$ ” şeklinde yazılmaktadır. Bu tur güzergahının çizim kodlaması Çizelge 3.14’te gösterilmektedir.

Çizelge 3.14. TurCizimi.m bölümünün kod yazımı

```
function TurCizimi(tur,model)

tur=[tur tur(1)];

plot(model.x(tur),model.y(tur),'k-s','MarkerFaceColor','y');

end
```

Bu bölüm benzetilmiş tavlama algoritmasında kod yazımının son bölümüdür. Bu bölümde elde edilen en ideal modelin tur güzergahını gösteren grafiksel çizim yapılmaktadır. Tur ve modele göre “TurCizimi(tur,model)” komutuyla çizim yapılıp, tur sonunda başlangıç şehrine tekrar döneceğini bildiren komut ise; “tur=[tur tur(1)];” şeklinde yazılmaktadır. Elde edilen en iyi sonucun tur güzergahına göre çizimi yapılırken, hazırlanmış olan modeldeki şehirler için verilen x, y koordinatları kullanılmaktadır. Her bir şehir için grafiksel çizim üzerinde karesel bir ifade ile gösterilmesini sağlayan komut ise; “plot(model.x(tur),model.y(tur),'k-s','MarkerFaceColor','y');” olarak yazılmaktadır.

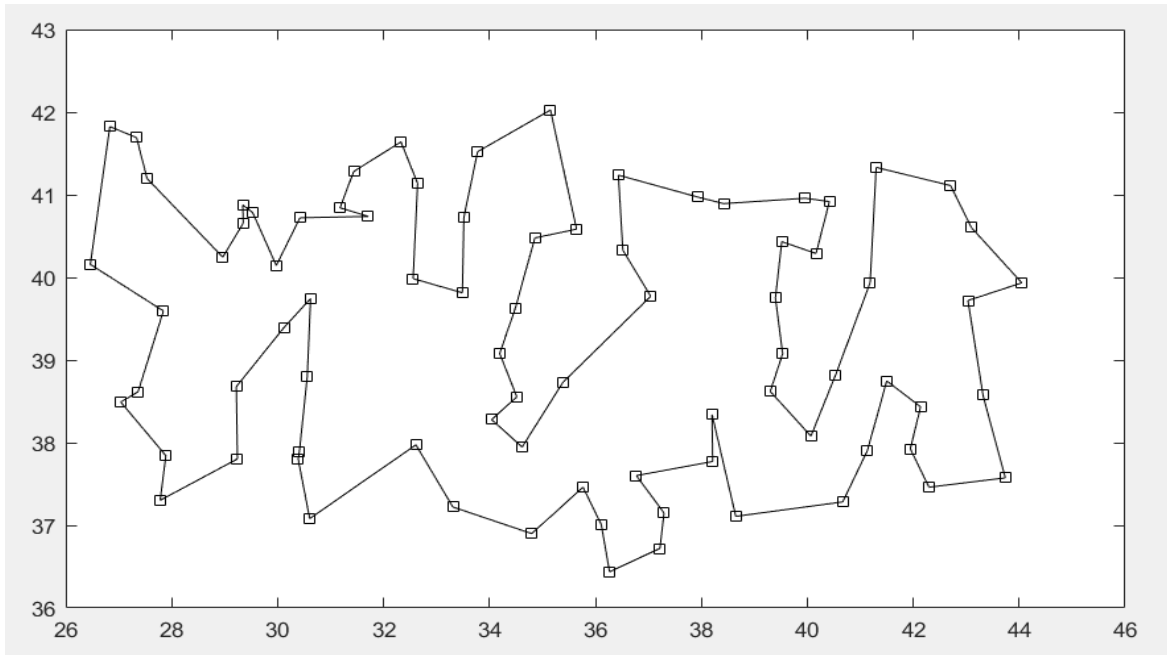
## 4. ARAŞTIRMA BULGULARI ve TARTIŞMA

Yapılan bu çalışmada her ilden 1 olmak üzere ülkemizdeki 81 ilden organize sanayi bölgeleri alınarak çalışma yapılmaktadır. Yapılan çalışma MATLAB programında benzetilmiş tavlama algoritması ve karınca kolonisi optimizasyonu yöntemleri kullanılarak analizler yapılmaktadır. Bu yöntemler kullanılarak yapılan çalışmalar organize sanayi bölgeleri için karayolu yük taşımacılığında gerçek karayolu mesafelerine göre ve organize sanayi bölgelerinin koordinatlarına göre kuş uçuşu mesafeleri için en ideal yol güzergahı bulunmaktadır. Karınca kolonisi optimizasyonunda OSB kuş uçuşu mesafelerine göre en ideal tur güzergahının koordinat düzlemi üzerinde gösterimi Şekil 4.1'de, Türkiye haritası üzerinde gösterimi ise Şekil 4.2'de gösterilmektedir. KKO için karayolu yük taşımacılığında gerçek mesafelere göre en ideal tur güzergahının koordinat düzlemi üzerinde gösterimi Şekil 4.3'te, Türkiye haritası üzerinde gösterimi ise Şekil 4.4'te gösterilmektedir. Çalışmadaki diğer bir analiz yöntemi olan BTA ile OSB'ler için kuş uçuşu mesafelerine göre en ideal tur güzergahının koordinat düzlemi üzerinde gösterimi Şekil 4.5'te, Türkiye haritası üzerinde gösterimi ise Şekil 4.6'da, yapılan iterasyonlara göre sonuçlardaki değişim grafiği Şekil 4.7'de gösterilmektedir. BTA kullanılarak elde edilen gerçek karayolu mesafelerine göre en ideal tur güzergahının koordinat düzlemi üzerinde gösterimi Şekil 4.8'de, Türkiye haritası üzerinde gösterimi ise Şekil 4.9'da, iterasyonlara göre sonuçlardaki değişimi ise Şekil 4.10'da gösterilmektedir.

### 4.1. Karınca Kolonisi Optimizasyonu ile OSB Koordinatlarına Göre Çözümü

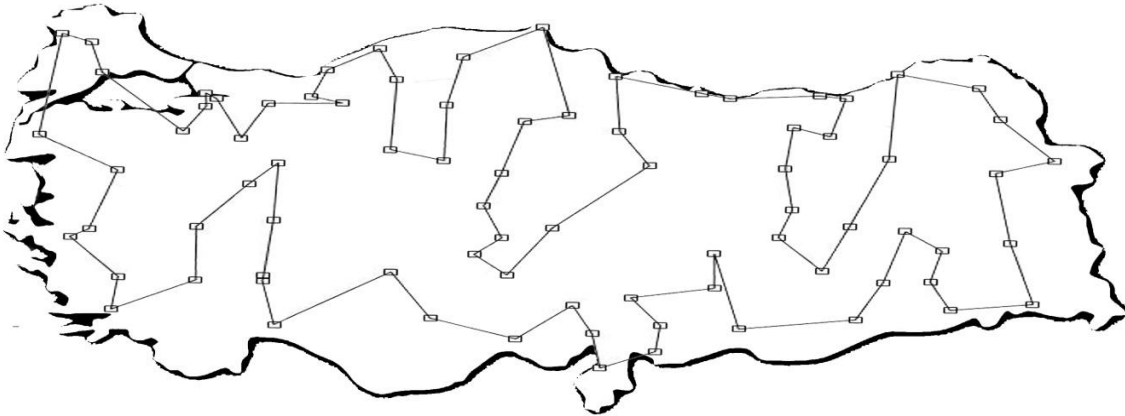
Ülkemizdeki 81 ilden her ilden 1 tane olmak üzere seçilen organize sanayi bölgeleri için Çizelge 3.1'de belirtilen koordinatlar kullanılarak analizler yapılmıştır. KKO yöntemi ile yapılan analiz sonuçlarına göre en ideal tur güzergahının koordinat düzlemi üzerinde gösterimi Şekil 4.1'de, Türkiye haritası üzerinde gösterimi ise Şekil 4.2'de gösterilmektedir.

#### 4.1.1. Tur güzergahının koordinat düzlemi üzerinde gösterimi



Şekil 4.1. Koordinatlara göre KKO yönteminde çözümün koordinat düzlemi üzerinde gösterimi

#### 4.1.2. Tur güzergahının Türkiye haritası üzerinde gösterimi



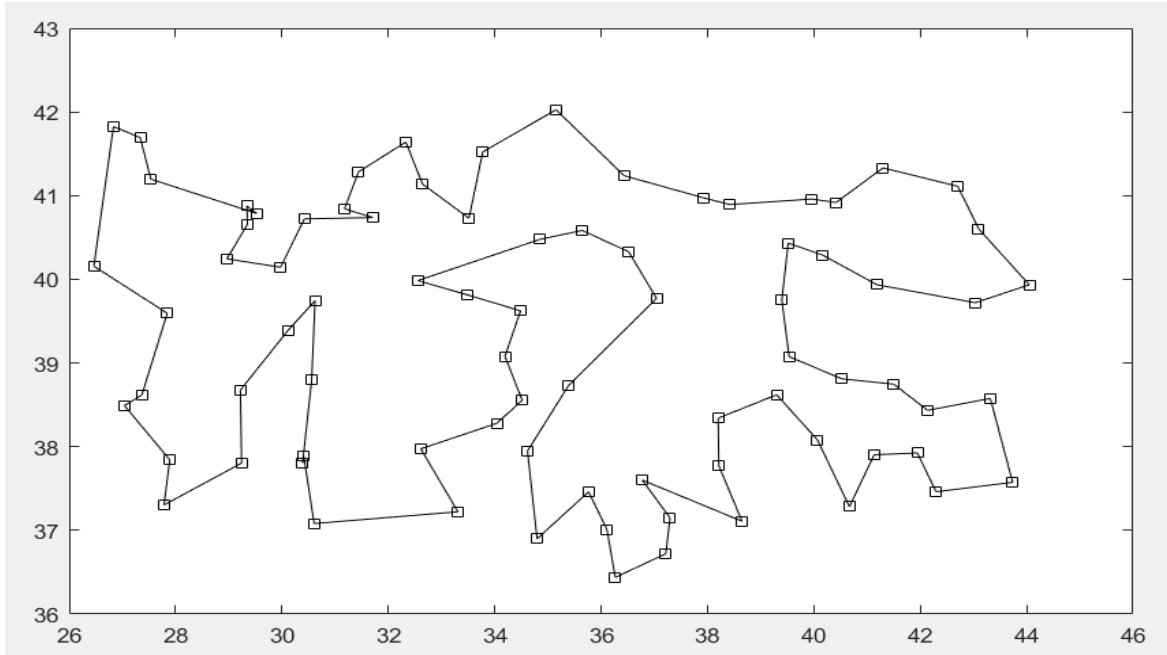
Şekil 4.2. KKO yönteminde koordinatlara göre çözümün Türkiye haritası üzerinde gösterimi

Karınca kolonisi optimizasyonunda organize sanayi bölgelerinin koordinatları kullanılarak en ideal kuş uçuşu mesafesi ve buna bağlı olarak en ideal tur güzergahı bulunmaktadır. Bu kapsamda koordinatlara göre KKO kullanılarak yapılan analiz sonuçlarında en ideal toplam mesafe değeri 7584,21 km olarak bulunmaktadır. Bulunan en ideal tur güzergahı sıralı bir şekilde; 72, 37, 78, 4, 38, 45, 10, 11, 32, 18, 27, 30, 76, 31, 36, 16, 65, 75, 35, 63, 67, 74, 70, 47, 62, 5, 61, 51, 80, 25, 6, 69, 46, 24, 49, 7, 43, 14, 81, 28, 20, 66, 17, 52, 40, 79, 22, 73, 50, 29, 23, 13, 56, 41, 12, 59, 26, 77, 54, 33, 3, 39, 21, 9, 53, 44, 58, 1, 64, 8, 48, 34, 42, 2, 55, 71, 57, 15, 60, 19, 68, 72 şeklinde olmaktadır. Yani başlangıçtaki OSB Şırnak Cizre OSB olup, sonrasında Hakkari OSB, Van OSB olarak ilerlenmektedir. Tur bitiminde ise Bitlis OSB ve Siirt OSB uğradıktan sonra turu tamamlamak için başlangıçtaki OSB olan Şırnak Cizre OSB gelinerek tur tamamlanmaktadır. Bu tur güzergahı koordinat düzleminde Şekil 4.1’de Türkiye haritası üzerinde gösterimi ise Şekil 4.2’de gösterilmektedir.

#### **4.2. Karınca Kolonisi Optimizasyonu ile OSB Gerçek Uzunluklarına Göre Çözümü**

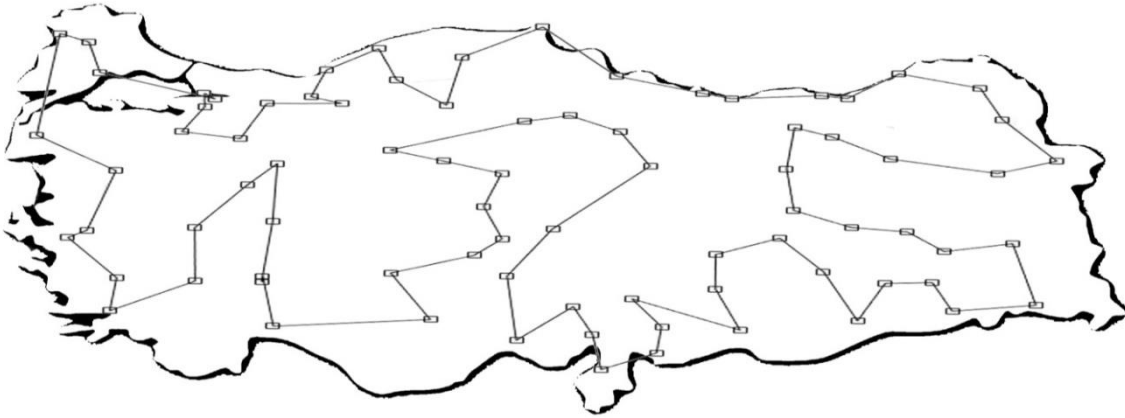
Ülkemizdeki 81 ilden her ilden 1 tane olmak üzere seçilen organize sanayi bölgeleri için Çizelge 3.2’de belirtilen gerçek karayolu mesafeleri kullanılarak analizler yapılmıştır. KKO yöntemi ile yapılan analiz sonuçlarına göre en ideal tur güzergahının koordinat düzlemi üzerinde gösterimi Şekil 4.3’te, Türkiye haritası üzerinde gösterimi ise Şekil 4.4’te gösterilmektedir.

#### 4.2.1. Tur güzergahının koordinat düzlemi üzerinde gösterimi



Şekil 4.3. KKO yönteminde gerçek mesafelere göre çözümün koordinat düzleminde gösterimi

#### 4.2.2. Tur güzergahının Türkiye haritası üzerinde gösterimi



Şekil 4.4. KKO yönteminde gerçek mesafelere göre çözümün Türkiye haritası üzerinde gösterimi

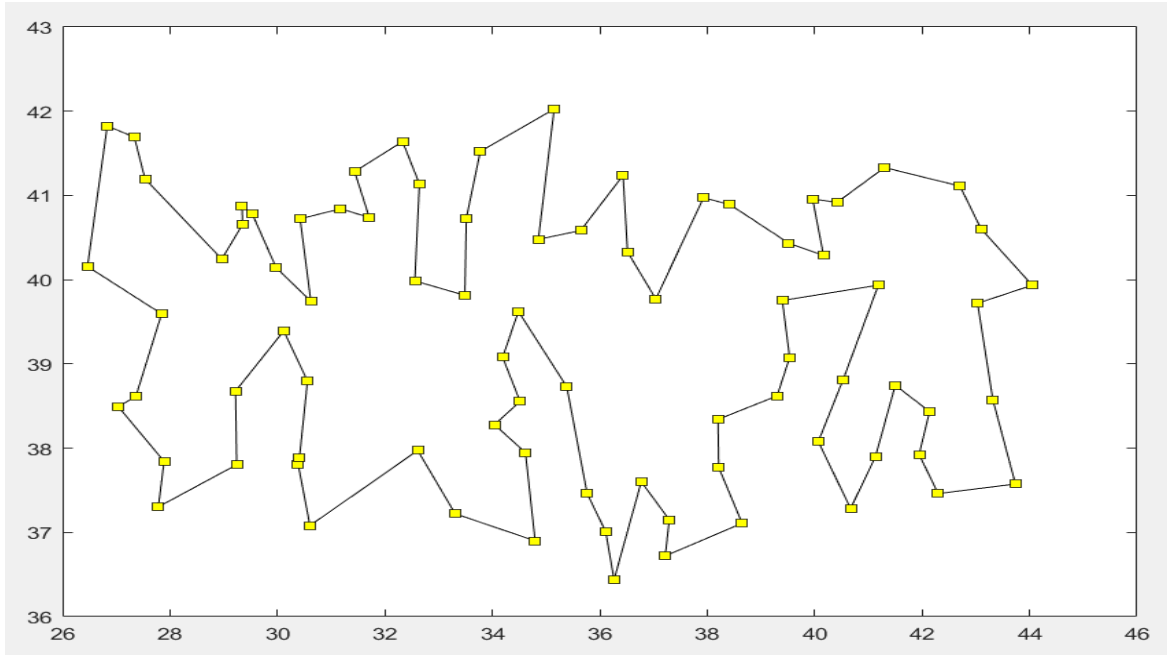


Karınca kolonisi optimizasyonu kullanılarak yapılan bir diğer analiz gerçek karayolu mesafelerine göre en ideal sonucu bulmak için yapılmaktadır. Yapılan analizler sonucunda elde edilen en ideal toplam mesafe miktarı 10198 km olarak bulunmaktadır. Bulunan en ideal tur güzergahı sıra bir şekilde; 2, 55, 30, 27, 57, 15, 68, 72, 37, 78, 19, 60, 18, 76, 31, 36, 16, 32, 4, 38, 45, 10, 11, 65, 75, 35, 63, 67, 69, 46, 24, 43, 14, 81, 28, 20, 66, 17, 22, 79, 40, 52, 73, 50, 29, 23, 13, 56, 41, 12, 59, 26, 77, 54, 33, 3, 21, 39, 9, 44, 53, 5, 61, 51, 80, 49, 7, 25, 6, 74, 70, 47, 62, 58, 1, 64, 8, 48, 34, 42, 71, 2 olarak bulunmaktadır. Yani tur başlangıcındaki OSB Adıyaman OSB olmakta ve sonrasında Malatya OSB, Elazığ OSB olarak ilerlenmektedir. Tur bitiminde ise Kahramanmaraş OSB, Şanlıurfa OSB uğrandıktan sonra başlangıçtaki Adıyaman OSB gelinerek tur tamamlanmaktadır. Bu tur güzergahının koordinat düzlemi üzerinde gösterimi Şekil 4.3'te, Türkiye haritası üzerinde gösterimi Şekil 4.4'te gösterilmektedir.

### **4.3. Benzetilmiş Tavlama Algoritması ile OSB Koordinatlarına Göre Çözümü**

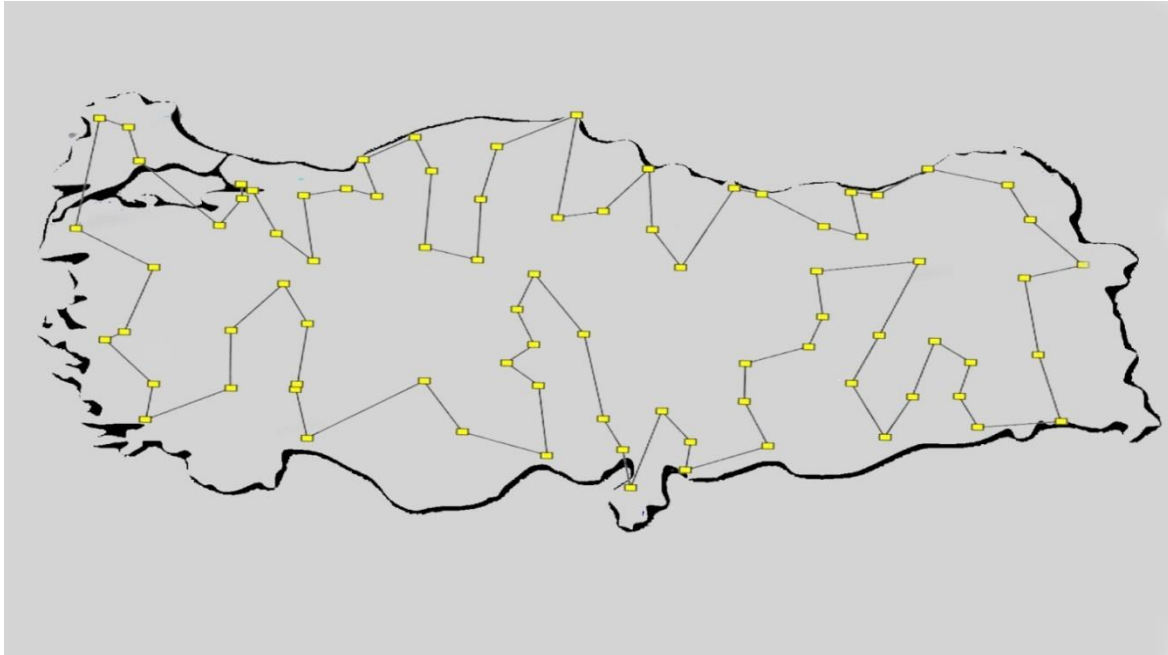
Ülkemizdeki 81 ilden her ilden 1 tane olmak üzere seçilen organize sanayi bölgeleri için Çizelge 3.1'de belirtilen koordinatlar kullanılarak analizler yapılmıştır. BTA yöntemi ile yapılan analiz sonuçlarına göre en ideal tur güzergahının koordinat düzlemi üzerinde gösterimi Şekil 4.5'te, Türkiye haritası üzerinde gösterimi Şekil 4.6'da, yapılan iterasyonlar sonrası sonuçlardaki değişimin grafiksel gösterimi ise Şekil 4.7'de gösterilmektedir.

### 4.3.1. Tur güzergahının koordinat düzlemi üzerinde gösterimi



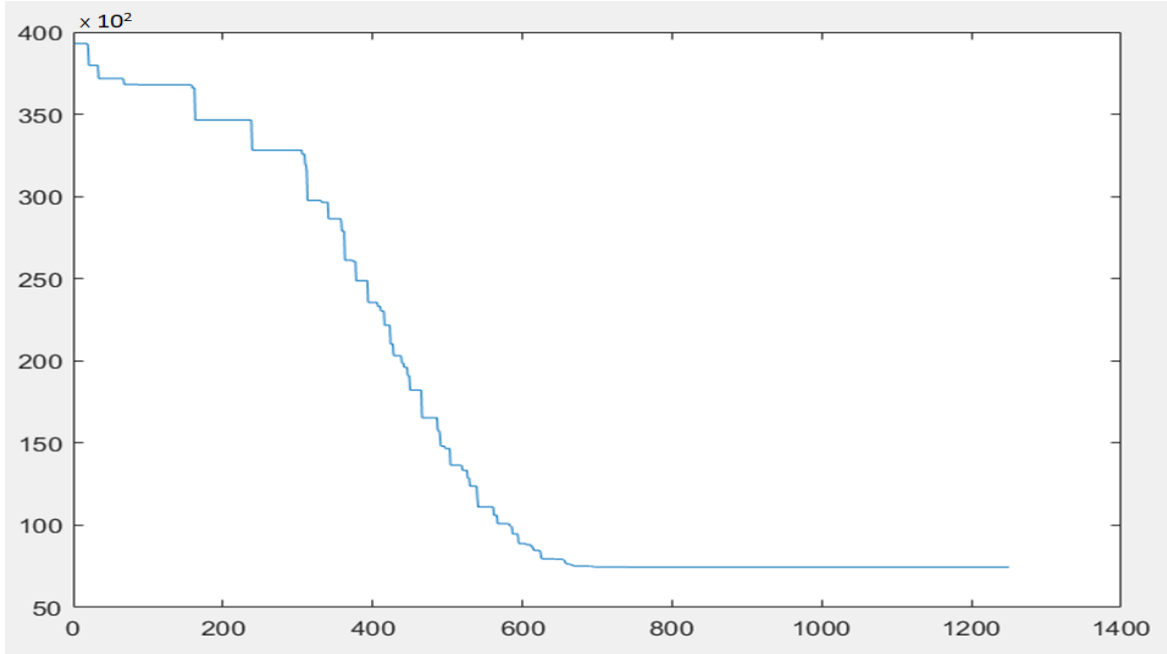
Şekil 4.5. Koordinatlara göre BTA yönteminde çözümün koordinat düzlemi üzerinde gösterimi

### 4.3.2. Tur güzergahının Türkiye haritası üzerinde gösterimi



Şekil 4.6. BTA yönteminde koordinatlara göre çözümün Türkiye haritası üzerinde gösterimi

### 4.3.3. İterasyonlar sonrası sonuçlardaki değişimin grafiksel gösterimi



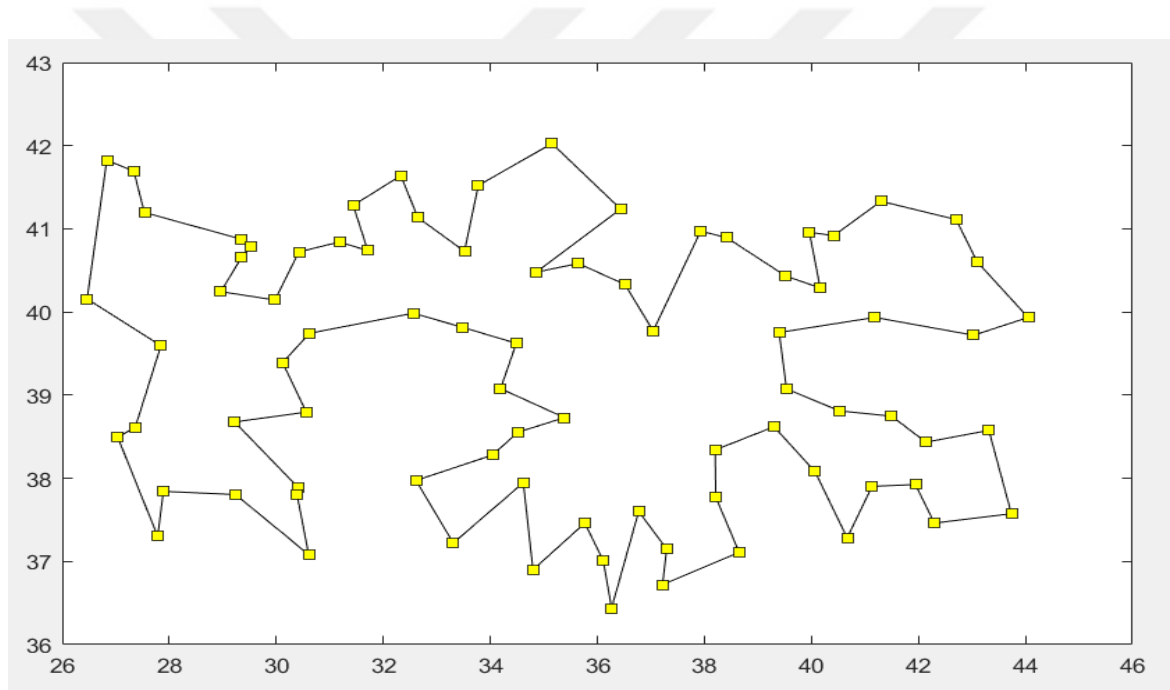
Şekil 4.7. Koordinatlara göre çözümün iterasyonlara göre elde edilen sonuçlardaki değişim grafiği

Benzetilmiş tavlama algoritması ile veri setinde bulunan organize sanayi bölgelerinin koordinatları kullanılarak en ideal kuş uçuşu mesafe değeri ve en ideal tur güzergahı bulunmaktadır. BTA ile yapılan analizler sonucunda kuş uçuşu olarak en ideal toplam mesafe değeri 7452,75 km olarak bulunmaktadır. En ideal tur güzergahı sıralı bir şekilde; 48, 34, 42, 8, 64, 1, 47, 80, 51, 61, 5, 62, 58, 44, 53, 9, 21, 39, 3, 54, 77, 26, 59, 12, 41, 56, 13, 23, 29, 50, 73, 22, 79, 40, 52, 17, 33, 66, 28, 20, 81, 14, 43, 7, 49, 24, 46, 69, 25, 6, 67, 74, 70, 63, 35, 36, 16, 75, 65, 11, 10, 45, 38, 4, 78, 37, 72, 68, 19, 60, 15, 57, 27, 18, 32, 31, 76, 30, 55, 2, 71, 48 olarak bulunmaktadır. Yani tur başlangıcındaki OSB, Kilis OSB olmakta ve sonrasında Gaziantep OSB, Kahramanmaraş OSB, Antakya OSB, Osmaniye OSB, Adana OSB olarak ilerlenmektedir. Tur bitiminde ise Elazığ OSB, Malatya OSB, Adıyaman OSB, Şanlıurfa organize sanayi bölgesine uğrandıktan sonra başlangıçtaki OSB olan Kilis organize sanayi bölgesine dönülerek tur tamamlanmaktadır. Elde edilen bu tur güzergahının koordinat düzlemi üzerinde gösterimi Şekil 4.5'te, Türkiye haritası üzerinde gösterimi ise Şekil 4.6'da gösterilmektedir. BTA'da kuş uçuşu mesafe için yapılan iterasyonlar ile en ideal mesafe bulunana kadar olan süreçte sonuçlarda olan değişimi gösteren grafik Şekil 4.7'de gösterilmektedir.

#### 4.4. BTA ile OSB Gerçek Uzunluklarına Göre Çözümü

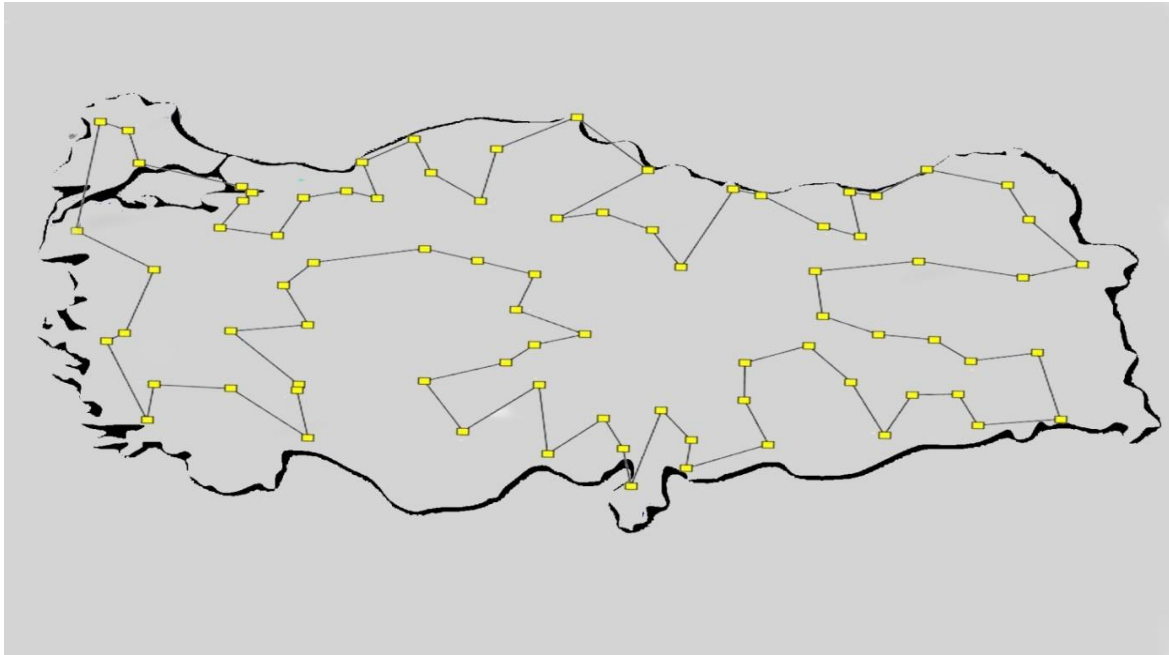
Ülkemizdeki 81 ilden her ilden 1 tane olmak üzere seçilen organize sanayi bölgeleri için Çizelge 3.2’de belirtilen gerçek karayolu mesafeleri kullanılarak analizler yapılmıştır. BTA yöntemi ile yapılan analiz sonuçlarına göre en ideal tur güzergahının koordinat düzlemi üzerinde gösterimi Şekil 4.8’de, Türkiye haritası üzerinde gösterimi Şekil 4.9’da, yapılan iterasyonlar sonrası sonuçlardaki değişimin grafiksel biçimde gösterimi ise Şekil 4.10’da gösterilmektedir.

##### 4.4.1. Tur güzergahının koordinat düzlemi üzerinde gösterimi



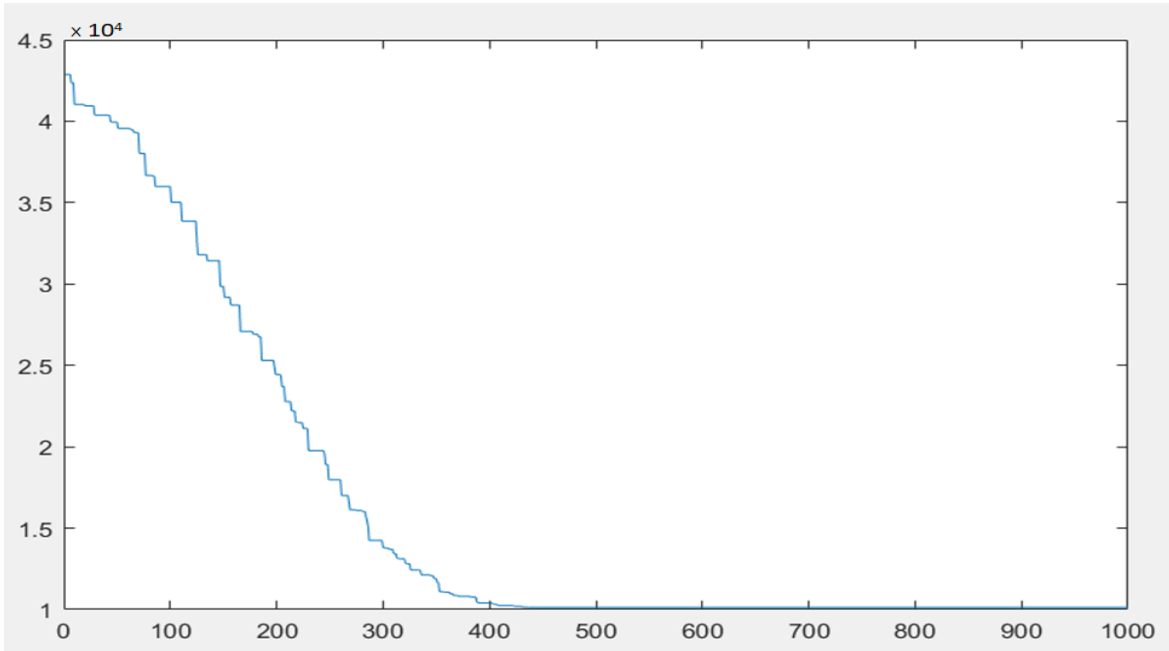
Şekil 4.8. BTA yönteminde gerçek mesafelere göre çözümün koordinat düzlemi üzerinde gösterimi

#### 4.4.2. Tur güzergahının Türkiye haritası üzerinde gösterimi



Şekil 4.9. BTA yönteminde gerçek mesafelere göre çözümün Türkiye haritası üzerinde gösterimi

#### 4.4.3. İterasyonlar sonrası elde edilen sonuçların grafiksel gösterimi



Şekil 4.10. İterasyonlara göre elde edilen sonuçlardaki değişim grafiği

Benzetilmiş tavlama algoritması kullanılarak en ideal toplam tur mesafesi değeri bulunmaktadır. BTA ile yapılan analizler sonucunda elde edilen en ideal toplam mesafe değeri 10133 km olarak bulunmaktadır. Bu sonucun elde edilmesini sağlayan tur güzergahı sırası ile; 63, 35, 36, 16, 75, 65, 11, 10, 45, 38, 4, 32, 31, 76, 18, 60, 19, 78, 37, 72, 68, 15, 57, 27, 30, 55, 2, 71, 48, 34, 42, 8, 64, 1, 58, 62, 44, 53, 5, 61, 47, 51, 80, 49, 7, 33, 54, 3, 77, 39, 21, 9, 26, 12, 59, 41, 56, 13, 23, 29, 50, 73, 40, 52, 79, 22, 17, 66, 28, 20, 81, 14, 43, 24, 46, 69, 67, 25, 6, 74, 70, 63 olarak bulunmaktadır. Bu durumda ilk olarak Ordu OSB ile başlanıp, Giresun OSB, Gümüşhane OSB, Bayburt OSB, Trabzon OSB, Rize OSB, Artvin OSB olarak ilerlenmektedir. Tur bitiminde ise Samsun OSB, Çorum OSB, Amasya OSB, Tokat OSB, Sivas OSB uğrandıktan sonra başlangıç noktası olan Ordu organize sanayi bölgesine dönülerek tur güzergahı tamamlanmaktadır. Bu tur güzergahının koordinat düzlemi üzerinde gösterimi Şekil 4.8’de, Türkiye haritası üzerinde gösterimi Şekil 4.9’da, iterasyonlar ile sonuçlardaki değişimi gösteren grafik ise Şekil 4.10’da gösterilmektedir.

## 5. SONUÇLAR ve ÖNERİLER

Yapılan bu çalışmada ülkemizde bulunan 81 ildeki organize sanayi bölgeleri ele alınmaktadır. Her ilden 1 tane alınmak şartıyla toplamda 81 adet organize sanayi bölgesi kullanılmaktadır. Organize sanayi bölgeleri kullanılarak yapılan çalışmalarda gezgin satıcı problemi örnek alınmaktadır. Bu çalışmada karınca kolonisi optimizasyonu ve benzetilmiş tavlama algoritması yöntemleri kullanılarak en ideal kuş uçuşu mesafeleri ve gerçek karayolu mesafelerine göre en ideal toplam mesafe değeri belirlenmektedir. KKO ve BTA yöntemleri kullanılarak elde edilen en ideal kuş uçuşu mesafesi daha önce farklı bir çalışmada yapılmış olan GA, PSO ve LO yöntemlerinin sonuçları ile karşılaştırılmaktadır. KKO ve BTA yöntemlerinin analiz sonuçları ile OSB gerçek karayolu mesafelerine göre en ideal sonuç daha önce farklı çalışmalarda yapılmış olan GA, PSO, LO ve TA yöntemlerinin sonuçları ile karşılaştırılmaktadır. Çalışma kapsamında KKO yöntemi kullanılarak kuş uçuşu mesafe için en ideal sonuç 7584,21 km, BTA yöntemi kullanılarak kuş uçuşu mesafe için en ideal sonuç 7452,75 km olarak bulunmaktadır. Daha önce farklı bir çalışmada aynı organize sanayi bölgeleri için en ideal kuş uçuşu mesafe sonuçları; LO yöntemi ile 7339 km, PSO yöntemi ile 7723 km, GA yöntemi ile 7605 km olarak bulunmaktadır.

Çalışmanın diğer bölümünde ise KKO ve BTA yöntemleri kullanılarak gerçek karayolu mesafelerine göre en ideal sonuçlar bulunmaktadır. KKO yöntemi ile bulunan en ideal sonuç 10198 km, BTA yöntemi ile bulunan ideal sonuç ise 10133 km olmaktadır. Daha önce farklı bir çalışmada aynı organize sanayi bölgeleri için karayolu gerçek mesafelerine göre en ideal sonuçlar; lineer optimizasyon yöntemi ile 9923 km, parçacıklı sürü optimizasyonu yöntemi ile 11313 km, genetik algoritma yöntemi ile 10689 km olarak bulunmaktadır. Daha önce farklı çalışmalarda elde edilen sonuçlar ile bu çalışmada yapılan analizler sonucunda elde edilen tüm sonuçlar Çizelge 5.1'de gösterilmektedir.

Çizelge 5.1. Tez çalışmasının sonuçları ve farklı çalışmalardaki yöntemlerin sonuçları

YÖNTEM	SONUÇLAR (km)	
	KOORDİNATLARA GÖRE (km)	GERÇEK KARAYOLU MESAFELERİNE GÖRE (km)
Karınca Kolonisi Optimizasyonu	7584,21	10198
Benzetilmiş Tavlama Algoritması	7452,75	10133
Lineer Optimizasyon	7339	9923
Parçacıklı Sürü Optimizasyonu	7723	11313
Genetik Algoritma	7605	10689
Tabu Arama Algoritması	---	10277

Çizelge 5.1’de verilen sonuçlar incelendiğinde 81 organize sanayi bölgesi kullanılarak aralarındaki en ideal kuş uçuşu mesafelerine göre incelendiğinde; KKO yöntemi, Lineer optimizasyon yöntemi sonucuna göre daha kötü, genetik algoritma ve parçacıklı sürü optimizasyonu yöntemlerine göre daha iyi sonuç elde edildiği gözlemlenmektedir. BTA yöntemi sonucu, lineer optimizasyon yöntemine göre daha kötü, parçacıklı sürü optimizasyonu ve genetik algoritma yöntemlerinin sonuçlarına göre daha iyi sonuç elde edildiği gözlemlenmektedir. Bu tez çalışmasında yapılan BTA ve KKO yöntemlerinin sonuçları birbirleri ile karşılaştırıldığında ise BTA yönteminin KKO yöntemine göre daha iyi sonuç verdiği gözlemlenmektedir.

Çizelge 5.1’de belirtilen sonuçlar incelendiğinde karayolu gerçek mesafelerine göre KKO yöntemi kullanılarak elde edilen sonuçlar ile farklı çalışmalarda elde edilen sonuçlar karşılaştırıldığında; KKO yönteminin lineer optimizasyon yöntemine göre daha kötü sonuç verdiği; parçacıklı sürü optimizasyonu, genetik algoritma ve tabu arama algoritması yöntemlerine göre daha iyi sonuç verdiği gözlemlenmektedir. BTA yöntemi ile diğer yöntem sonuçları karşılaştırıldıklarında ise; BTA yönteminin lineer optimizasyon yöntemine göre daha kötü; parçacıklı sürü optimizasyonu, genetik algoritma ve tabu arama algoritması yöntemlerine göre daha iyi sonuç verdiği gözlemlenmektedir. Bu tez kapsamında gerçek karayolu mesafelerine göre BTA yöntemi ile KKO yöntemi sonuçları karşılaştırıldıklarında ise; BTA yönteminin KKO yöntemine göre daha iyi sonuç verdiği gözlemlenmektedir.

Bu tez çalışmasında ülkemizdeki önemli sektörlerden birisi olan lojistik sektörü ve bu bağlamda organize sanayi bölgeleri arasındaki yol mesafeleri için çalışmalar yapılmaktadır. Kullanılan yöntemlerin amacı toplam mesafeyi azaltmak için



uygulanmaktadır. Çalışma kapsamında her ilden 1 tane olması şartı ile ülkemizdeki 81 il için organize sanayi bölgeleri kullanılmaktadır. Toplamda 81 adet olan organize sanayi bölgeleri arasındaki en ideal kuş uçuşu mesafeleri ve en ideal karayolu gerçek mesafeleri bulmak için çalışmalar yapılmaktadır. Bu çalışmalar KKO ve BTA yöntemleri kullanılarak yapılmakta ve farklı çalışmalarda yapılan diğer yöntemler ile karşılaştırılmaktadır. Çalışma içerisinde KKO ve BTA yöntemlerinin analiz detayları verilmektedir. Bu yöntemler kullanılarak elde edilen sonuçlar ile farklı çalışmalardaki kuş uçuşu mesafe ve gerçek karayolu mesafelerine göre elde edilen sonuçlar karşılaştırıldığında; BTA ve KKO yöntemlerinin LO yönteminden kötü, diğer yapılan yöntemlerden ise daha iyi sonuçlar verdiği gözlemlenmektedir. Bu analizler sonucunda en ideal mesafelerin bulunması ve bu duruma yönelik çalışmalar yapılmasına olanak sağlayacaktır. Buna bağlı olarak en kısa mesafelerin bulunması hem zaman hem de maliyet açısından fayda sağlayacaktır.

## KAYNAKLAR

- Açıl, A. (2008). Solving the profitable tour problem using ant colony system, Master's Thesis, *Boğaziçi University*, İstanbul.
- Ahmadpur, M. (2016). Evaluation of heuristic algorithms for traffic signal optimization along a corridor, Master's thesis, *Boğaziçi University Graduate School of Sciences and Engineering*, İstanbul.
- Al-Amyal, F., Számel, L. ve Hamouda, M. (2022). An enhanced direct instantaneous torque control of switched reluctance motor drives using ant colony optimization. *Ain Shams Engineering Journal*, 101967.
- Bakkal, M. ve Demir, U. (2011). *Lojistik yönetimi ve e-lojistik* (Vol. 30). Hiperlink eğitim. yay. san. tic. ve ltd. sti.
- Bezek, Ç. (2009). Sonlu elemanlar yöntemi ile asenkron motor analizinin Matlab'da gerçekleştirilmesi, Yüksek Lisans Tezi, *Fırat Üniversitesi Fen Bilimleri Enstitüsü*, Elazığ.
- Bozkurt, M. A. (2019). Çok türlü taşımacılık güzergâhlarında lineer optimizasyon ve parçacıklı sürü optimizasyonunun karşılaştırılması, Yüksek Lisans Tezi, *İskenderun Teknik Üniversitesi Mühendislik ve Fen Bilimleri Enstitüsü İnşaat Mühendisliği Anabilim Dalı*, Hatay.
- Bulut, Ö. (2007). Türkiye'de taşımacılık sektörünün lojistik olgusu içerisinde incelenmesi, Yüksek Lisans Tezi, *Kadir Has Üniversitesi Sosyal Bilimler Enstitüsü İşletme Anabilim Dalı*, İstanbul.
- Büyüközer, A. (2019). Benzetilmiş tavlmalı hibrit balina optimizasyon algoritmasının karesel atama problemine uygulanması, Yüksek Lisans Tezi, *Karabük Üniversitesi Fen Bilimleri Enstitüsü*, Karabük.
- Chelouah, R. ve Siarry, P. (2000). Tabu search applied to global optimization. *European journal of operational research*, 123(2), 256-270.
- Çelebi, S. B. (2021). Yük taşımacılığında coğrafi bilgi sistemi tabanlı en kısa güzergâh analizi, Yüksek Lisans Tezi, *İskenderun Teknik Üniversitesi Mühendislik ve Fen Bilimleri Enstitüsü İnşaat mühendisliği Anabilim Dalı*, Hatay.
- Demir, K. (2006). Karayolu uluslararası yük taşımacılığının durumu, yeni mevzuatın ve uluslararası gelişmelerin taşımacılık sektörüne etkisi, Yüksek Lisans Tezi, *Gazi Üniversitesi Fen Bilimleri Enstitüsü*, Ankara.
- Deng, W., Zhang, L., Zhou, X., Zhou, Y., Sun, Y., Zhu, W., Chen, H., Deng, W., Chen, H. ve Zhao, H. (2022). Multi-strategy particle swarm and ant colony hybrid optimization for airport taxiway planning problem. *Information Sciences*, 612, 576-593.
- Dorigo, M. (1992). Optimization, learning and natural algorithms. *Ph. D. Thesis*, *Politecnico di Milano*.

- Du, L. ve He, R. (2012). Combining nearest neighbor search with tabu search for large-scale vehicle routing problem. *Physics Procedia*, 25, 1536-1546.
- Eberhart, R. ve Kennedy, J. (1995). A new optimizer using particle swarm theory. In *MHS'95. Proceedings of the sixth international symposium on micro machine and human science* (pp. 39-43). Ieee.
- Ekşioğlu, D. (2022). Lojistik işletmelerde dijitalleşmenin lojistik performans ve satın alma açısından değerlendirilmesi, Yüksek Lisans Tezi, *Niğde Ömer Halisdemir Üniversitesi Sosyal Bilimler Enstitüsü Uluslararası Ticaret ve Lojistik Yönetimi Ana Bilim Dalı*, Niğde.
- Erdoğan, Ş. Z. (2008). Kendini klonlayan karınca kolonisi yaklaşımıyla optimal yolun bulunması, Doktora Tezi, *Trakya Üniversitesi Fen Bilimleri Enstitüsü*, Edirne.
- Gao, W. (2020). New ant colony optimization algorithm for the traveling salesman problem. *International Journal of Computational Intelligence Systems*, 13(1), 44-55.
- Glover F. ve Mcmillan, C. (1986). The General Employee Scheduling Problem: An Integration of Management Science and Artificial Intelligence. *Computers and Operations Research*, 13 (5), 563-593.
- Goss, S., Aron, S., Deneubourg, J. L. ve Pasteels, J. M. (1989). Self-organized shortcuts in the Argentine ant. *Naturwissenschaften*, 76(12), 579-581.
- Guo, N., Qian, B., Na, J., Hu, R. ve Mao, J. L. (2022). A three-dimensional ant colony optimization algorithm for multi-compartment vehicle routing problem considering carbon emissions. *Applied Soft Computing*, 127, 109326.
- Gündoğdu, H. (2020). Yük taşımacılığında optimizasyon tekniklerinin karşılaştırılması, Yüksek Lisans Tezi, *İskenderun Teknik Üniversitesi Mühendislik ve Fen Bilimleri Enstitüsü İnşaat Mühendisliği Anabilim Dalı*, Hatay.
- Gürbüz, Ö. (2015). Tabu arama algoritmasının kuyruk problemine uygulanması, Yüksek Lisans Tezi, *Hacettepe Üniversitesi Fen Bilimleri Enstitüsü*, Ankara.
- Hanafiah, N., Wijaya, I., Xavier, S., Young, C. G., Adrianto, D. ve Shodiq, M. (2019). Itinerary recommendation generation using enhanced simulated annealing algorithm. *Procedia Computer Science*, 157, 605-612.
- Huang, Y. H., Blazquez, C. A., Huang, S. H., Paredes-Belmar, G., ve Latorre-Nuñez, G. (2019). Solving the feeder vehicle routing problem using ant colony optimization. *Computers and Industrial Engineering*, 127, 520-535.
- İlhan, İ. (2021). An improved simulated annealing algorithm with crossover operator for capacitated vehicle routing problem. *Swarm and Evolutionary Computation*, 64, 100911.

İnternet: [https://cevreselgostergeler.csb.gov.tr/ulastirma-turlerine-gore-tasinan-yolcu-ve-yuk-miktari-i-85789#\\_edn1](https://cevreselgostergeler.csb.gov.tr/ulastirma-turlerine-gore-tasinan-yolcu-ve-yuk-miktari-i-85789#_edn1), Son Erişim Tarihi: 10.10.2022

İnternet: <https://sozluk.gov.tr/>, Son Erişim Tarihi: 10.06.2022

- Kaabachi, I., Jriji, D., Madany, F. ve Krichen, S. (2017). A Bi-criteria ant colony optimization for minimizing fuel consumption and cost of the traveling salesman problem with time windows. *Procedia computer science*, 112, 886-895.
- Kanna, S. R., Sivakumar, K. ve Lingaraj, N. (2021). Development of deer hunting linked earthworm optimization algorithm for solving large scale traveling salesman problem. *Knowledge-Based Systems*, 227, 107199.
- Koç, İ. O. (2007). Gezgin satıcı problemi için çok popülasyonlu paralel bir genetik algoritma tasarımı geliştirilmesi ve analizi, Doktora Tezi, *Eskişehir Osmangazi Üniversitesi Fen Bilimleri Enstitüsü*, Eskişehir.
- Koçak, R. D. (2020). Lojistiğin Tarihsel Gelişimi: Askeri Gereksinimden İşletme Lojistiğine ve Tedarik Zinciri Yönetimine Evrilme Süreci. *Yaşar Üniversitesi E-Dergisi*, 15(58), 246-258.
- Lu, S. H., Kuo, R. J., Ho, Y. T. ve Nguyen, A. T. (2022). Improving the efficiency of last-mile delivery with the flexible drones traveling salesman problem. *Expert Systems with Applications*, 209, 118351.
- Mitchell, M. (1998). *An introduction to genetic algorithms*. MIT press.
- Özalp, M. (2020). Uygun dağıtım rotası belirlenmesi probleminde hibrit sezgisel bir yöntem uygulanması, Yüksek Lisans Tezi, *Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü*, İstanbul.
- Özbay, T. (2017). Evde bakım hizmetlerinin değişken komşuluk arama algoritması ile çözeltilmesi, Yüksek Lisans Tezi, *Pamukkale Üniversitesi Fen Bilimleri Enstitüsü*, Denizli.
- Özbiçge, E. U. (2010). Digital kameraların kalibrasyonuna yönelik bir matlab yaklaşımı, Yüksek Lisans Tezi, *Gebze Yüksek Teknoloji Enstitüsü Mühendislik ve Fen Bilimleri Enstitüsü Jeodezi ve Fotogrametri Mühendisliği Ana Bilim Dalı*, Kocaeli.
- Özkan, Ö. (2021). Optimization of the distance-constrained multi-based multi-UAV routing problem with simulated annealing and local search-based metaheuristic to detect forest fires: The case of Turkey. *Applied Soft Computing*, 113, 108015.
- Özkök, D. (2013). Design of a blood bank network: considering facility location, inventory and routing decisions, Master's Thesis, *Koç University Graduate School of Sciences and Engineering*, İstanbul.
- Pamukçu, H. (2021). Katı atık toplama güzergah optimizasyonu: Kayapınar (Diyarbakır) ilçesi örneği, Yüksek Lisans Tezi, *Harran Üniversitesi Fen Bilimleri Enstitüsü*, Şanlıurfa.
- Panwar, K. ve Deep, K. (2021). Discrete Grey Wolf Optimizer for symmetric travelling salesman problem. *Applied Soft Computing*, 105, 107298.

- Reda, M., Onsy, A., Elhosseini, M. A., Haikal, A. Y. ve Badawy, M. (2022). A discrete variant of cuckoo search algorithm to solve the Travelling Salesman Problem and path planning for autonomous trolley inside warehouse. *Knowledge-Based Systems*, 252, 109290.
- Sarıcan, M. A. (2016). E-lojistikte kritik faaliyetlerin belirlenmesi ve Türkiye'deki e-lojistik uygulamaları, Yüksek Lisans Tezi, *Pamukkale Üniversitesi Sosyal Bilimler Enstitüsü*, Denizli.
- Serin, S. (2009). Karınca kolonisi yaklaşımıyla karayolu üstyapı rutin bakım çalışmalarının planlanması, *Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü*, Isparta.
- Sim, K. M. ve Sun, W. H. (2003). Ant colony optimization for routing and load-balancing: survey and new directions. *IEEE transactions on systems, man, and cybernetics-Part A: systems and humans*, 33(5), 560-572.
- Skinderowicz, R. (2022). Improving Ant Colony Optimization efficiency for solving large TSP instances. *Applied Soft Computing*, 120, 108653.
- Söke, A. (2003). Genetik algoritma ve benzetilmiş tavlama ile iki boyutlu giyotinsiz kesme problemlerine olasılıksal yaklaşım, Yüksek Lisans Tezi, *Kocaeli Üniversitesi Fen Bilimleri Enstitüsü*, Kocaeli.
- Stodola, P., Otrisal, P. ve Hasilová, K. (2022). Adaptive ant Colony optimization with node clustering applied to the travelling salesman problem. *Swarm and Evolutionary Computation*, 70, 101056.
- Tafida, A. K. (2014). Overall cost optimization of a concrete bridge using simulated annealing algorithm, Master's Thesis, *Atılım University Graduate School of Sciences and Engineering*, Ankara.
- Tekin, E. (2014). E-Lojistik ve ilaç dağıtımında e-lojistik uygulamaları. Yayımlanmamış Yüksek Lisans Tezi, *Selçuk Üniversitesi Sosyal Bilimler Enstitüsü*, Konya.
- Topal, B. (2015). Afet lojistik yönetim sistemlerinin incelenmesi ve yeni model tasarımı, Yüksek Lisans Tezi, *Sakarya Üniversitesi Fen Bilimleri Enstitüsü*, Sakarya.
- Tuzkaya, U. R. (2007). Çok modlu taşımacılık sistemlerinin stratejik planlamasında kritik faktörlerin modellenmesine yönelik bir çözüm yaklaşımı, Doktora Tezi, *Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü*, İstanbul.
- Ünal, M. (2008). Pid kontrolörün karınca kolonisi genetik algoritma tabanlı optimizasyonu ve gunt rt 532 basınç prosesinin kontrolü, Yüksek Lisans Tezi, *Marmara Üniversitesi Fen Bilimleri Enstitüsü*, İstanbul.
- Van Wassenhove, L. (2006). Blackett memorial lecture humanitarian aid logistics: supply chain. *Journal of the operational research society*, 57(5), 475-489.
- Xu, L., Huang, K., Liu, J., Li, D. ve Chen, Y. F. (2022). Intelligent planning of fire evacuation routes using an improved ant colony optimization algorithm. *Journal of Building Engineering*, 61, 105208.

- Yang, X., Dong, H. ve Yao, X. (2017). Passenger distribution modelling at the subway platform based on ant colony optimization algorithm. *Simulation Modelling Practice and Theory*, 77, 228-244.
- Yıldırım, T. (2014). Simetrik gezgin satıcı problemi için yeni bir meta-sezgisel: kör fare algoritması, Yüksek Lisans Tezi, *Pamukkale Üniversitesi Fen Bilimleri Enstitüsü*, Denizli.
- Yogaswara, Y. ve Fatin, F. F. (2020). Tabu Search for Route Determination and Scheduling Waste Transportation Bandung City West Bandung Region (Case Study: PD. Kebersihan Kota Bandung). *In IOP Conference Series: Earth and Environmental Science* (Vol. 448, No. 1, p. 012125). IOP Publishing.
- Zhang, J. R., Zhang, J., Lok, T. M. ve Lyu, M. R. (2007). A hybrid particle swarm optimization–back-propagation algorithm for feedforward neural network training. *Applied mathematics and computation*, 185(2), 1026-1037.
- Zhang, Q. ve Xiong, S. (2018). Routing optimization of emergency grain distribution vehicles using the immune ant colony optimization algorithm. *Applied Soft Computing*, 71, 917-925.
- Zhang, Z. ve Han, Y. (2022). Discrete sparrow search algorithm for symmetric traveling salesman problem. *Applied Soft Computing*, 118, 108469.
- Zidi I., Al-Omani, M. ve Aldhafeeri, K. (2019). A new approach based on the hybridization of simulated annealing algorithm and tabu search to solve the static ambulance routing problem. *Procedia Computer Science*, 159, 1216-1228.

## EKLER

### EK-1. KKO ModelOlusturma.m Bölümü Kodu

```

function model=ModelOlusturma()
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 42.138701 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 37.217575 33.488427
27.336426 34.191881 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];
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 38.434203 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 36.720531 39.813963
41.692118 39.078548 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];
n=numel(x);

D=[0 325 646 950 335 537 578 176 700 1018 896 916 1043 849 614 705 816 643 731 737
687 902 1152 664 509 793 523 793 1249 450 599 786 722 200 645 674 896 1050 689 963
986 146 772 351 997 752 237 269 534 1209 421 944 428 742 333 957 531 138 997 717
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 345 429 871
938 1086 1336 759 635 1044 225 926 1382 215 421 534 906 155 592 539 655 752 940
1097 1236 180 838 602 734 809 387 207 615 1342 511 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 1284
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 508 504 843 343 627 425 368 225 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 365 224
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 170 986 824 790 979 1612 938 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 886 629 693 516 829 696 484 744 925 404 412  
569 818 331 306 506 702 460 916 600 590 777 389 463 628 665 1159 1074 403 630 699  
408 424 190 988 419 146 532 201 864 88 595 137 455 482 669 795 250 722 840 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 0 341 653 784 794 609 901 844 403 792 479 625 629 733 405 726  
703 953 244 82 778 664 460 916 548 388 575 543 571 326 448 1048 873 714 633 917 495  
362 568 786 251 286 644 242 876 279 612 475 617 453 887 786 554 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 1110 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 255 340 745 116 638 224 358 276 286 671 557 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 928 580 819 895 609 696 816  
766 981 1231 743 636 872 489 872 1328 451 652 799 801 153 759 788 862 1016 768  
1044 1065 162 851 430 998 831 416 127 613 1288 500 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 1361  
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 732 617 399 549 818 560 618 339 341 953 429 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  
457 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 89 948 868 895 1022 1655 981 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 447 530  
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 268 773 746 826 829 1424 860 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 1575  
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 1114 672 607 717 538 495 345 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  
1559 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 1176 635 313 728 244 557 252 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 1110 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 1010 318 643 432 381 540 514 862 750 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 183 148 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 687 458 914 1642 810 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 316 395 859 1109 1157 1407 703 562 1203 462 914 1370 400 127 124 998 654 227 65 703 422 1115 1087 1372 664 816 851 335 705 557 798 697 1330 671 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 1343 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 1010 406 400 498 149 407 128 945 395 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 0 190 1079 1157 1273 1523 867 726 1251 147 1078 1590 146 258 194 1093 451 537 364 501 479 1163 1307 1444 483 979 884 393 869 595 483 823 1550 719 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;

731 429 1284 224 925 733 1110 696 1361 457 530 1575 1559 1110 148 395 1343 190 0 1111 1344 1433 1683 956 814 1416 240 1167 1624 311 347 276 1253 537 626 463 312 324 1324 1337 1644 614 1067 1011 401 970 760 569 958 1584 883 1329 1047 1327 427 1612 294 797 1631 71 859 884 658 671 488 1238 795 119 923 578 419 223 1491 647 501 304 1413 159 1357 852 1195

737 871 405 1140 404 405 166 816 630 1184 952 734 440 169 1170 859 221 1079 1111 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 248 485 896 255 473 369 210 426 346 821 580 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 1344 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 664 556 884 494 760 499 491 282 233 893 362 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 1433 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 1099 494 321 586 103 496 191 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 1683 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 1350 746 265 838 354 747 436 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 703 400 867 956 179 576 478 729 0 181 622 866 234 690 749 611 797 387 662 512 671 1270 1095 558 407 761

587 136 495 1009 90 354 765 130 650 244 388 363 461 654 731 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 82 266 636 709 845 670 821 785 356 877 562 556 726 814 404 649  
644 895 181 0 699 742 461 917 625 470 657 463 554 385 530 1130 955 635 633 837 479  
311 496 868 199 261 628 162 877 220 614 396 537 531 808 864 504 903 750 294 348 336  
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  
1416 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 712 650 989 536 611 592 518 371 242 986 222 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 147 240  
1041 1112 1253 1504 866 742 1206 0 1092 1548 130 350 337 1050 330 621 468 498 561  
1121 1265 1401 407 1004 803 536 919 553 362 781 1508 677 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 1167  
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 304 541 951 310 422 425 159 482 292 877 529 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  
1624 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 1451 773 58 893 317 865 586 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 146 311  
939 1010 1151 1402 749 625 1103 130 994 1457 0 237 314 948 318 499 355 621 599  
1019 1162 1299 338 874 739 513 790 451 386 678 1406 575 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 770 450 127 988 258 347 767  
1000 1065 1317 611 470 1094 350 823 1285 237 0 189 906 519 283 120 662 487 1009  
995 1401 558 724 745 401 613 451 580 605 1239 565 976 726 980 319 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 194 276  
954 1187 1252 1503 797 657 1280 337 1009 1472 314 189 0 1093 642 349 187 593 301  
1195 1182 1588 651 911 931 214 800 638 674 801 1425 751 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 1253  
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 476 520 916 311 510 404 259 313 103 850 461 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 936 426 654 942 451 537 822  
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 62 562 1336 458 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 648 735 227 889 537 626 669 1030 967 1218 512 385 1079 621 724 1187 499 283 349 843 678 0 164 960 678 1017 896 1302 607 625 794 592 488 501 756 544 1139 601 877 777 919 494 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 364 463 827 1075 1125 1377 671 530 1169 468 883 1345 355 120 187 963 707 164 0 802 521 1090 1069 1354 646 798 826 434 650 540 795 679 1313 653 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 501 312 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 735 1298 2033 1194 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 479 324 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 935 890 1089 1722 1048 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 1324 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 636 564 888 471 712 503 478 285 219 901 349 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 1337 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 1128 480 256 600 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 1644 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 1185 687 511 780 442 565 352 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 861 503 664 870 483 614 747 816 959 1210 587 479 912 407 802 1265 338 558 651 776 81 607 646 782 937 820 972 1107 0 709 478 854 691 259 142 486 1214 382 951 545 830 225 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 1067 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 129 453 858 223 596 337 333 462 466 784 703 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 1011 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 584 296 546 368 934 276 699 134 445 607 662 809 240 692 986

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 393 401  
 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 847 870 1002 1635 961 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 705 469 869 970 248  
 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 0 447 853 218 717 332 454 451 549 705 824 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;  
 237 387 504 824 146 286 340 416 617 868 746 774 844 600 687 557 614 595 760 485 556  
 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 0 408 230 958 126 695 279 574 334 811 671 270 864 691 100 129  
 474 359 665 615 449 760 533 221 474 837 865 279 616 516 627 918 724 164 611;  
 269 207 843 790 532 644 745 127 399 895 826 1114 1176 1010 458 798 1010 483 569  
 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 0 636 1410 532 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;  
 534 615 343 979 201 242 116 613 549 1022 829 672 635 318 914 697 406 823 958 255  
 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 218 230 636 0 728 115 465 237 388 561 659 899 446 772 919 202  
 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  
 1584 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 1410 728 0 852 276 807 511 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 88 279 224 500 560 981 860 717 728 432 810 671 498 719 883 369 499  
 586 838 244 220 592 677 425 893 575 565 751 404 458 601 653 1194 1048 503 600 780  
 382 337 276 961 332 126 532 115 852 0 579 222 458 457 738 795 332 808 815 87 176  
 532 421 779 499 412 883 496 335 606 960 749 320 729 640 544 1041 609 98 496;  
 944 1078 368 1347 595 612 358 1025 618 1393 1161 538 244 381 1379 1068 149 1287  
 1329 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 1143 465 276 579 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 1047  
 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 451 279 625 237 807 222 577 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 1327  
 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 549 574 939 388 511 458 295 321 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 862 354 454 945 262 427 821  
 893 1033 1284 654 531 986 220 877 1339 118 319 431 850 205 494 460 738 715 901

1046 1180 225 784 607 627 705 334 267 561 1299 457 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  
1612 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 1154 659 452 738 386 536 322 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 229 294  
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 671 369 899 1673 795 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 831 893 763 686 834 728 711 797 650  
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 663 270 321 446 1128 332 857 346 656 434 873 598 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  
1631 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 1195 772 670 808 605 591 428 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 71  
1047 1250 1391 1642 891 750 1343 221 1103 1566 242 283 214 1208 524 594 436 380  
389 1258 1280 1539 583 1003 986 334 893 691 555 919 1525 815 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 63 319 295 422 519 965 843 676 752 519 794 654 523 702 859 445 459  
611 863 331 294 552 660 501 963 558 548 735 428 471 598 629 1167 1032 466 670 748  
337 424 236 944 424 100 475 202 923 87 652 181 501 440 718 749 259 775 798 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 98 397 354 331 518 994 872 714 788 607 799 683 582 716 884 505 488  
671 922 419 348 594 663 560 1023 571 577 764 488 379 627 658 1076 1061 492 730 797  
330 512 165 973 512 129 434 290 982 176 712 230 550 444 767 711 168 796 811 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 599 777 270 840 572 658 620  
981 918 1169 463 336 1029 643 675 1138 521 326 392 794 655 50 211 971 689 968 845  
1170 578 576 761 596 444 474 717 495 1097 532 827 729 867 490 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 90 642 997 987 920 983 854 560 757 817 566 671 740 691  
906 1157 665 613 796 464 795 1258 421 692 776 723 140 741 781 837 991 695 972 992  
140 757 353 972 757 359 195 539 1217 421 955 435 745 304 962 472 143 999 659 346  
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 408 488  
865 1226 1163 1414 708 581 1275 551 920 1383 528 259 214 1039 854 202 177 793 511  
1213 1090 1509 771 821 954 377 689 665 842 740 1342 779 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 1238  
 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 1008 385 351 499 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 919 460 897 412 701 705 795 480  
 842 778 1029 323 196 890 777 536 998 655 451 534 654 699 191 353 1107 831 828 705  
 1031 623 429 689 738 304 449 761 355 957 412 687 589 727 584 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 240 119  
 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 760 528 987 1832 883 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 578 638 834 923 417  
 835 715 967 261 280 883 906 473 935 784 579 700 647 827 357 519 1235 997 821 642  
 1024 752 308 755 903 182 533 890 389 895 496 624 623 721 712 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 639 593 341 777 486 578 625  
 771 865 1116 431 308 864 460 681 1144 331 234 421 682 402 284 323 950 718 779 850  
 1059 327 566 510 631 482 221 465 383 1103 335 825 493 756 248 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 333 419  
 970 958 1173 1424 839 702 1063 212 1062 1525 304 511 524 990 150 703 632 561 740  
 962 1202 1259 229 932 620 721 927 474 180 710 1484 606 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 348 223  
 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 837 535 1064 1838 960 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  
 1491 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 1313 635 81 749 186 732 432 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;  
 427 447 634 676 386 111 407 542 850 707 532 962 926 519 691 394 697 558 647 517 790  
 785 1036 322 192 838 557 572 1035 428 303 489 602 460 250 377 1047 786 776 743 979  
 385 473 568 699 362 279 523 303 994 320 725 537 676 345 949 679 556 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 424 501  
 815 1177 1114 1365 659 532 1225 543 871 1384 430 242 230 989 797 153 127 809 527  
 1163 1040 1366 721 771 905 400 639 616 859 690 1293 729 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 139 304  
 905 1075 1203 1455 749 608 1169 190 961 1423 126 141 275 1050 387 418 262 613 572

1083 1130 1364 408 861 811 485 751 516 517 744 1383 640 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  
1413 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 635 627 969 474 507 544 369 351 137 968 187  
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  
159 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 918 724 1145 1806 1041 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  
1357 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 1118 494 293 609 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;  
417 587 454 864 187 181 227 570 658 907 786 783 746 428 829 597 517 724 852 366 598  
605 856 215 122 658 697 421 884 568 490 677 422 495 503 579 1187 974 582 591 798  
420 334 375 887 298 164 551 118 843 98 572 320 496 485 769 819 407 881 808 186 250  
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 1195  
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 1005 382 483 496 212 554 352 942 593 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.x=x;  
model.y=y;  
model.D=D;  
model.n=n;

end

**EK-2. KKO Turmaliyet.m Bölümü Kodu**

```
function L=turmaliyet(model,tur)
n=numel(tur);
tur=[tur tur(1)];

L=0;
for i=1:n
    L=L+model.D(tur(i),tur(i+1));
end

end
```





### EK-3. KKO Komutları.m Bölümü Kodu

```

clc;
clear;
close all;
%% Problemin tanıtımı
model=ModelOlusturma();
mesafefonksiyonu=@(tur) turmaliyet(model,tur);
sehirsayisi=model.n;

%% KKO Parametreleri
iterasayonsayisi=1333;
karincasayisi=10;
eta=1./model.D;
alfa=1;
beta=1;
buharlasamiktari=0.2;
feromonyogunlugu0=1;
feromonyogunlugu=feromonyogunlugu0*ones(sehirsayisi);
Q=1;

%% Komutları Başlatma Kısmı
%bos karınca
bos_karinca.tur=[];
bos_karinca.maliyet=[];
enyikarinca.maliyet=inf;
karınca=repmat(bos_karinca,karincasayisi,1);

for it=1:iterasayonsayisi
    for k=1:karincasayisi
        karınca(k).tur=randi([1 sehirsayisi]);

        for l=2:sehirsayisi
            i=karınca(k).tur(end);
            P=feromonyogunlugu(i,:).^alfa.*eta(i,:).^beta;
            P(karınca(k).tur)=0;
            P=P/sum(P);
            j=RuletCarki(P);
            karınca(k).tur=[karınca(k).tur j];
        end
        karınca(k).maliyet=mesafefonksiyonu(karınca(k).tur);
        if karınca(k).maliyet<enyikarinca.maliyet
            enyikarinca=karınca(k);
        end
    end
end
end

```

```
% Feromon Miktarlarının Güncellenmesi
for k=1:karincasayisi
    tur=karınca(k).tur;
    tur=[tur tur(1)];
    for l=1:sehirsayisi
        i=tur(l);
        j=tur(l+1);
        feromonyogunlugu(i,j)=feromonyogunlugu(i,j)+Q/karınca(k).maliyet;
    end
end

% Buharlaşma
feromonyogunlugu=(1-buharlasamiktari)*feromonyogunlugu;
%En iyi sonuçların kaydedilmesi
iterasyonuneniyisonucu(it)=eniyikarınca.maliyet;
disp(['iterasyon :' num2str(it), ' ', 'iterasyonuneniyisonucu ='
num2str(iterasyonuneniyisonucu(it))])
SonuclarinCizimi(eniyikarınca.tur,model)
pause=0.05;
end
```

**EK-4. KKO RuletCarki.m Bölümü Kodu**

```
function j=RuletCarki(P)
r=rand;
C=cumsum(P);
j=find(r<=C,1,'first');

end
```



**EK-5. KKO SonuclarinCizimi.m Bölümü Kodu**

```
function SonuclarinCizimi(tur,model)
```

```
tur=[tur tur];  
plot(model.x(tur),model.y(tur),'k-s')
```

```
end
```



**EK-6. BTA ModelOlusturma.m Bölümü Kodu**

```

function model=ModelOlusturma()
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 42.138701 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 37.217575 33.488427
27.336426 34.191881 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];
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 38.434203 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 36.720531 39.813963
41.692118 39.078548 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];
n=numel(x);

D=[0 325 646 950 335 537 578 176 700 1018 896 916 1043 849 614 705 816 643 731 737
687 902 1152 664 509 793 523 793 1249 450 599 786 722 200 645 674 896 1050 689 963
986 146 772 351 997 752 237 269 534 1209 421 944 428 742 333 957 531 138 997 717
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 345 429 871
938 1086 1336 759 635 1044 225 926 1382 215 421 534 906 155 592 539 655 752 940
1097 1236 180 838 602 734 809 387 207 615 1342 511 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 1284
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 508 504 843 343 627 425 368 225 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 365 224
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 170 986 824 790 979 1612 938 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 886 629 693 516 829 696 484 744 925 404 412  
569 818 331 306 506 702 460 916 600 590 777 389 463 628 665 1159 1074 403 630 699  
408 424 190 988 419 146 532 201 864 88 595 137 455 482 669 795 250 722 840 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 0 341 653 784 794 609 901 844 403 792 479 625 629 733 405 726  
703 953 244 82 778 664 460 916 548 388 575 543 571 326 448 1048 873 714 633 917 495  
362 568 786 251 286 644 242 876 279 612 475 617 453 887 786 554 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 1110 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 255 340 745 116 638 224 358 276 286 671 557 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 928 580 819 895 609 696 816  
766 981 1231 743 636 872 489 872 1328 451 652 799 801 153 759 788 862 1016 768  
1044 1065 162 851 430 998 831 416 127 613 1288 500 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 1361  
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 732 617 399 549 818 560 618 339 341 953 429 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  
457 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 89 948 868 895 1022 1655 981 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 447 530  
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 268 773 746 826 829 1424 860 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 1575  
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 1114 672 607 717 538 495 345 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  
1559 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 1176 635 313 728 244 557 252 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 1110 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 1010 318 643 432 381 540 514 862 750 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 183 148  
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 687 458 914 1642 810 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 316 395 859  
1109 1157 1407 703 562 1203 462 914 1370 400 127 124 998 654 227 65 703 422 1115  
1087 1372 664 816 851 335 705 557 798 697 1330 671 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 1343  
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 1010 406 400 498 149 407 128 945 395 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 0 190  
1079 1157 1273 1523 867 726 1251 147 1078 1590 146 258 194 1093 451 537 364 501  
479 1163 1307 1444 483 979 884 393 869 595 483 823 1550 719 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;

731 429 1284 224 925 733 1110 696 1361 457 530 1575 1559 1110 148 395 1343 190 0  
1111 1344 1433 1683 956 814 1416 240 1167 1624 311 347 276 1253 537 626 463 312  
324 1324 1337 1644 614 1067 1011 401 970 760 569 958 1584 883 1329 1047 1327 427  
1612 294 797 1631 71 859 884 658 671 488 1238 795 119 923 578 419 223 1491 647 501  
304 1413 159 1357 852 1195

737 871 405 1140 404 405 166 816 630 1184 952 734 440 169 1170 859 221 1079 1111 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 248 485 896 255 473 369 210 426 346 821 580 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 1344  
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 664 556 884 494 760 499 491 282 233 893 362 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  
1433 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 1099 494 321 586 103 496 191 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  
1683 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 1350 746 265 838 354 747 436 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 703 400 867 956 179  
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 90 354 765 130 650 244 388 363 461 654 731 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 82 266 636 709 845 670 821 785 356 877 562 556 726 814 404 649  
644 895 181 0 699 742 461 917 625 470 657 463 554 385 530 1130 955 635 633 837 479

311 496 868 199 261 628 162 877 220 614 396 537 531 808 864 504 903 750 294 348 336  
 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  
 1416 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 712 650 989 536 611 592 518 371 242 986 222 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 147 240  
 1041 1112 1253 1504 866 742 1206 0 1092 1548 130 350 337 1050 330 621 468 498 561  
 1121 1265 1401 407 1004 803 536 919 553 362 781 1508 677 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 1167  
 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 304 541 951 310 422 425 159 482 292 877 529 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  
 1624 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 1451 773 58 893 317 865 586 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 146 311  
 939 1010 1151 1402 749 625 1103 130 994 1457 0 237 314 948 318 499 355 621 599  
 1019 1162 1299 338 874 739 513 790 451 386 678 1406 575 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 770 450 127 988 258 347 767  
 1000 1065 1317 611 470 1094 350 823 1285 237 0 189 906 519 283 120 662 487 1009  
 995 1401 558 724 745 401 613 451 580 605 1239 565 976 726 980 319 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 194 276  
 954 1187 1252 1503 797 657 1280 337 1009 1472 314 189 0 1093 642 349 187 593 301  
 1195 1182 1588 651 911 931 214 800 638 674 801 1425 751 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 1253  
 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 476 520 916 311 510 404 259 313 103 850 461 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 936 426 654 942 451 537 822  
 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 62 562 1336 458 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 648 735 227 889 537 626 669  
 1030 967 1218 512 385 1079 621 724 1187 499 283 349 843 678 0 164 960 678 1017 896  
 1302 607 625 794 592 488 501 756 544 1139 601 877 777 919 494 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 364 463 827  
1075 1125 1377 671 530 1169 468 883 1345 355 120 187 963 707 164 0 802 521 1090  
1069 1354 646 798 826 434 650 540 795 679 1313 653 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  
312 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 735 1298 2033 1194 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  
324 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 935 890 1089 1722 1048 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 1324  
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 636 564 888 471 712 503 478 285 219 901 349 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  
1337 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 1128 480 256 600 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  
1644 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 1185 687 511 780 442 565 352 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 861 503 664 870 483 614 747  
816 959 1210 587 479 912 407 802 1265 338 558 651 776 81 607 646 782 937 820 972  
1107 0 709 478 854 691 259 142 486 1214 382 951 545 830 225 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 1067  
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 129 453 858 223 596 337 333 462 466 784 703 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 1011  
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 584 296 546 368 934 276 699 134 445 607 662 809 240 692 986  
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 393 401  
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 847 870 1002 1635 961 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 705 469 869 970 248  
 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 0 447 853 218 717 332 454 451 549 705 824 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;  
 237 387 504 824 146 286 340 416 617 868 746 774 844 600 687 557 614 595 760 485 556  
 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 0 408 230 958 126 695 279 574 334 811 671 270 864 691 100 129  
 474 359 665 615 449 760 533 221 474 837 865 279 616 516 627 918 724 164 611;  
 269 207 843 790 532 644 745 127 399 895 826 1114 1176 1010 458 798 1010 483 569  
 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 0 636 1410 532 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;  
 534 615 343 979 201 242 116 613 549 1022 829 672 635 318 914 697 406 823 958 255  
 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 218 230 636 0 728 115 465 237 388 561 659 899 446 772 919 202  
 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  
 1584 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 1410 728 0 852 276 807 511 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 88 279 224 500 560 981 860 717 728 432 810 671 498 719 883 369 499  
 586 838 244 220 592 677 425 893 575 565 751 404 458 601 653 1194 1048 503 600 780  
 382 337 276 961 332 126 532 115 852 0 579 222 458 457 738 795 332 808 815 87 176  
 532 421 779 499 412 883 496 335 606 960 749 320 729 640 544 1041 609 98 496;  
 944 1078 368 1347 595 612 358 1025 618 1393 1161 538 244 381 1379 1068 149 1287  
 1329 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 1143 465 276 579 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 1047  
 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 451 279 625 237 807 222 577 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 1327  
 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 549 574 939 388 511 458 295 321 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 862 354 454 945 262 427 821  
 893 1033 1284 654 531 986 220 877 1339 118 319 431 850 205 494 460 738 715 901  
 1046 1180 225 784 607 627 705 334 267 561 1299 457 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  
 1612 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 1154 659 452 738 386 536 322 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 229 294  
 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 671 369 899 1673 795 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 831 893 763 686 834 728 711 797 650  
 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 663 270 321 446 1128 332 857 346 656 434 873 598 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  
 1631 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 1195 772 670 808 605 591 428 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 71  
 1047 1250 1391 1642 891 750 1343 221 1103 1566 242 283 214 1208 524 594 436 380  
 389 1258 1280 1539 583 1003 986 334 893 691 555 919 1525 815 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 63 319 295 422 519 965 843 676 752 519 794 654 523 702 859 445 459  
 611 863 331 294 552 660 501 963 558 548 735 428 471 598 629 1167 1032 466 670 748  
 337 424 236 944 424 100 475 202 923 87 652 181 501 440 718 749 259 775 798 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 98 397 354 331 518 994 872 714 788 607 799 683 582 716 884 505 488  
 671 922 419 348 594 663 560 1023 571 577 764 488 379 627 658 1076 1061 492 730 797  
 330 512 165 973 512 129 434 290 982 176 712 230 550 444 767 711 168 796 811 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 599 777 270 840 572 658 620  
 981 918 1169 463 336 1029 643 675 1138 521 326 392 794 655 50 211 971 689 968 845  
 1170 578 576 761 596 444 474 717 495 1097 532 827 729 867 490 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 90 642 997 987 920 983 854 560 757 817 566 671 740 691  
 906 1157 665 613 796 464 795 1258 421 692 776 723 140 741 781 837 991 695 972 992  
 140 757 353 972 757 359 195 539 1217 421 955 435 745 304 962 472 143 999 659 346  
 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 408 488  
 865 1226 1163 1414 708 581 1275 551 920 1383 528 259 214 1039 854 202 177 793 511  
 1213 1090 1509 771 821 954 377 689 665 842 740 1342 779 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 1238  
 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 1008 385 351 499 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 919 460 897 412 701 705 795 480  
 842 778 1029 323 196 890 777 536 998 655 451 534 654 699 191 353 1107 831 828 705  
 1031 623 429 689 738 304 449 761 355 957 412 687 589 727 584 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 240 119  
 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 760 528 987 1832 883 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 578 638 834 923 417  
 835 715 967 261 280 883 906 473 935 784 579 700 647 827 357 519 1235 997 821 642  
 1024 752 308 755 903 182 533 890 389 895 496 624 623 721 712 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 639 593 341 777 486 578 625  
 771 865 1116 431 308 864 460 681 1144 331 234 421 682 402 284 323 950 718 779 850  
 1059 327 566 510 631 482 221 465 383 1103 335 825 493 756 248 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 333 419  
 970 958 1173 1424 839 702 1063 212 1062 1525 304 511 524 990 150 703 632 561 740  
 962 1202 1259 229 932 620 721 927 474 180 710 1484 606 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 348 223  
 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 837 535 1064 1838 960 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  
 1491 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 1313 635 81 749 186 732 432 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;  
 427 447 634 676 386 111 407 542 850 707 532 962 926 519 691 394 697 558 647 517 790  
 785 1036 322 192 838 557 572 1035 428 303 489 602 460 250 377 1047 786 776 743 979  
 385 473 568 699 362 279 523 303 994 320 725 537 676 345 949 679 556 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 424 501  
 815 1177 1114 1365 659 532 1225 543 871 1384 430 242 230 989 797 153 127 809 527  
 1163 1040 1366 721 771 905 400 639 616 859 690 1293 729 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 139 304  
 905 1075 1203 1455 749 608 1169 190 961 1423 126 141 275 1050 387 418 262 613 572  
 1083 1130 1364 408 861 811 485 751 516 517 744 1383 640 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  
 1413 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 635 627 969 474 507 544 369 351 137 968 187  
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  
159 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 918 724 1145 1806 1041 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  
1357 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 1118 494 293 609 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;  
417 587 454 864 187 181 227 570 658 907 786 783 746 428 829 597 517 724 852 366 598  
605 856 215 122 658 697 421 884 568 490 677 422 495 503 579 1187 974 582 591 798  
420 334 375 887 298 164 551 118 843 98 572 320 496 485 769 819 407 881 808 186 250  
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 1195  
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 1005 382 483 496 212 554 352 942 593 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.D=D;  
model.x=x;  
model.y=y;

end

**EK-7. BTA MesafeHesabi.m Bölümü Kodu**

```
function z=MesafeHesabi(model,tur)
```

```
D=model.D;  
n=numel(tur);  
L=0;
```

```
tur=[tur tur(1)];
```

```
for i=1:n
```

```
    L=L+D(tur(i),tur(i+1));
```

```
end
```

```
z=L;
```

```
end
```

**EK-8. BTA RastgeleCozumOlusturma.m Bölümü Kodu**

```
function tur=RastgeleCozumOlusturma(model)
n=model.n;
tur=randperm(n);
end
```



## EK-9. BTA BTAkomutlari.m Bölümü Kodu

```

clc;
clear;
close all;
%% Problemin Tanıtımı
model=ModelOlusturma()
mesafefonksiyonu=@(tur) MesafeHesabi(model,tur);

%% BTA Parametreleri
iterasyonsayisi=1000;
sicaklikdegisimisonrasitursayisi=95000;
T0=2000;

%% Komutları Başlatma Kısım1

sonuc.turguzergahi=RastgeleCozumOlusturma(model);
sonuc.mesafe=mesafefonksiyonu(sonuc.turguzergahi);

eniyisonuc=sonuc;
iterasyonuneniyisonucu=zeros(iterasyonsayisi,1);

T=T0;

%% Döngü Kısım1
for it=1:iterasyonsayisi
    for it2=1:sicaklikdegisimisonrasitursayisi

        yenisonuc.turguzergahi=KomsuCevapOlusturma(sonuc.turguzergahi);
        yenisonuc.mesafe=mesafefonksiyonu(yenisonuc.turguzergahi);
        if yenisonuc.mesafe<sonuc.mesafe
            sonuc=yenisonuc;
        else
            delta=yenisonuc.mesafe-sonuc.mesafe;
            p=exp(-delta/T);
            if rand<p
                sonuc=yenisonuc;
            end
        end
    end

    if sonuc.mesafe<eniyisonuc.mesafe
        eniyisonuc=sonuc;
    end
end
T=0.99*T;
iterasyonuneniyisonucu(it)=eniyisonuc.mesafe;

disp(['Iteration' num2str(it) ': iterasyonuneniyisonucu= '
num2str(iterasyonuneniyisonucu(it))]);

```



```
figure(1);  
TurCizimi(eniyisonuc.turguzergahi,model)  
  
end  
%% figürlerin Çizimi  
figure;  
plot(iterasyonuneniyisonucu);
```



**EK-10. BTA KomsuCevapOlusturma.m Bölümü Kodu**

```
function xyeni=KomsuCevapOlusturma(x)
```

```
n=numel(x);
```

```
i=randperm(n);
```

```
i1=i(1);
```

```
i2=i(2);
```

```
xyeni=x;
```

```
xyeni([i1 i2])=xyeni([i2 i1]);
```

```
end
```

**EK-11. BTA TurCizimi.m Bölümü Kodu**

```
function TurCizimi(tur,model)
```

```
tur=[tur tur(1)];
```

```
plot(model.x(tur),model.y(tur),'k-s','MarkerFaceColor','y');
```

```
end
```



## DİZİN

---

### A

ABSTRACT · v

Algoritma · iv, xi, 1, 34, 46, 68

Analiz · iv, 10, 57, 59, 61, 64, 69

---

### B

Benzetilmiş Tavlama Algoritması · iv, v, x, xi, 1, 6, 46  
47, 50, 57, 61, 62, 63, 64, 65, 66, 67, 68, 69

---

### G

Genetik algoritma · iv, v, xi, 1, 9, 10, 11, 12, 16, 19, 67,  
68, 72, 73

Gezgin Satıcı Problemi · iv, v, xi, 1, 2, 3, 6, 19, 45

Güzergah · 4, 5, 10, 16, 53

---

### K

Karınca Kolonisi Optimizasyonu · iv, x, xi, 1, 4, 34, 36,  
37, 57, 58, 59, 60, 67, 68, 69

---

### L

Lineer Optimizasyon · iv, v, xi, 1, 19, 67, 68, 69

Lojistik · iv, 1, 2, 70, 71, 73

---

### M

MATLAB · 1, 4, 13, 57

Mesafe · 1, 3, 16, 17, 19, 22, 23, 24, 25, 26, 27, 28, 29,  
30, 31, 32, 33, 34, 38, 39, 44, 48, 50, 51, 54, 59, 61,  
63, 66, 67, 69

---

### O

Organize Sanayi Bölgesi · iv, xi, 1, 19, 20, 21, 22, 23,  
24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 46, 57, 59,  
61, 63, 64, 66, 67

---

### Ö

ÖZET · iv

---

### P

Parçacıklı Sürü Optimizasyonu · iv, v, xi, 1, 19, 67, 68

---

### T

Tabu Arama · iv, v, xi, 1, 8, 19, 67, 68

Taşımacılık · 1, 3, 4, 10, 70, 73

---

### Y

Yöntem · 3, 7, 12, 14, 16, 53, 68



# TEKNOVERSITE



teknoversite **AYRICALIĞINDASINIZ**

**İSTE**

