

T.C.
GAZİ ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ

DOKTORA TEZİ

KATI ATIK TOPLAMA ROTA OPTİMİZASYONUNA SEZGİSEL BİR YAKLAŞIM: ÇANKAYA ÖRNEĞİ

İBRAHİM KESKİN

EKONOMETRİ ANABİLİM DALI
YÖNEYLEM ARAŞTIRMASI BİLİM DALI

KASIM 2023


GAZİ ÜNİVERSİTESİ
Gazi
1926

KATI ATIK TOPLAMA ROTA OPTİMİZASYONUNA SEZGİSEL BİR YAKLAŞIM: ÇANKAYA ÖRNEĞİ

İbrahim KESKİN

DOKTORA TEZİ
EKONOMETRİ ANABİLİM DALI
YÖNEYLEM ARAŞTIRMASI BİLİM DALI

GAZİ ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ

KASIM 2023

ETİK BEYAN

Gazi Üniversitesi Sosyal Bilimler Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmasında; 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 çalışmasında 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.

İbrahim KESKİN

06.11.2023

TEZ ONAY SAYFASI

Ankara Hacı Bayram Veli Üniversitesi Lisansüstü Eğitim Enstitüsü Ekonometri Anabilim Dalı Yöneylem Araştırması Bilim Dalı Doktora öğrencisi İbrahim KESKİN tarafından hazırlanan "Katı Atık Toplama Rota Optimizasyonuna Sezgisel Bir Yaklaşım: Çankaya Örneği" başlıklı tez çalışması, 06/11/2023 tarih ve 11:30 saatinde yapılan tez savunması sınavında aşağıdaki jüri tarafından OY BİRLİĞİ ile DOKTORA TEZİ olarak **KABUL** edilmiştir.

	Kabul	**Ret**
Başkan: Prof. Dr. Latif ÖZTÜRK	☑	☐
Danışman: Prof. Dr. Şenol ALTAN	☑	☐
Üye: Prof. Dr. Murat ATAN	☑	☐
Üye: Prof. Dr. Mehmet KABAK	☑	☐
Üye: Prof. Dr. H. Hasan ÖRKCÜ	☑	☐

KATI ATIK TOPLAMA ROTA OPTİMİZASYONUNA SEZGİSEL BİR YAKLAŞIM: ÇANKAYA ÖRNEĞİ

(Doktora Tezi)

İbrahim KESKİN

GAZİ ÜNİVERSİTESİ

SOSYAL BİLİMLER ENSTİTÜSÜ

Kasım 2023

ÖZET

Nüfus ve toplumun ekonomik düzeyindeki artışa paralel, tüketim miktarının artması ile oluşan katı atıkların, çevre ve insan sağlığına zarar vermeksizin ve en az maliyetle yönetimi yerel yönetimler için çözülmesi gereken öncelikli sorunlardan biridir. Yerel yönetimlerin en büyük maliyet kalemlerinden birisini atık yönetimi, bunun içindeki en büyük payı da ulaşım giderleri oluşturmaktadır. Bu yüzden yerel yönetimler bir yandan katı atıkları insan ve çevreye zarar vermeden en etkili bir şekilde toplamaya çalışırken diğer yandan da oluşan ulaşım maliyetini en aza indirmeye çalışmaktadır. Bu çalışmada Ankara ili Çankaya ilçesinde bulunan bir kısım çöp konteynerinin yeri belirlenerek söz konusu bölgede icra edilen katı atık toplama sürecinde toplam mesafenin minimize edilmesi amaçlanmaktadır. Kat edilen toplam mesafeyi minimize etmek maksadıyla bir matematiksel model önerilmiş, problemin büyüklüğünden ötürü sezgisel yöntemlerden Demon algoritmasını esas alan bir çözüm yöntemi geliştirilmiştir. Geliştirilen algoritmanın çözümleri ile önerilen matematiksel modelin GAMS kullanılarak çözülebilen küçük düğüm sayılı çözümleri karşılaştırılmış, müteakiben geliştirilen algoritma ile çok büyük boyutlu çözümler yapılmıştır. GAMS kullanılarak çözülebilen küçük problem örneklerindeki optimal çözüm sonuçlarına çok kısa sürede ulaşılmıştır.

Bilim Kodu : 112304
Anahtar Kelimeler : Katı Atık Toplama, Araç Rotalama Problemi, Demon Algoritması
Sayfa Adedi : 67
Tez Danışmanı : Prof. Dr. Şenol ALTAN

HEURISTIC APPROACH TO SOLID WASTE ROUTE OPTIMİZATION: ÇANKAYA EXAMPLE

(Ph. D. Thesis)

İbrahim KESKİN

GAZİ UNIVERSITY

GRADUATE SCHOOL OF SOCIAL SCIENCES

November 2023

ABSTRACT

Parallel to the increase in the population and economic level of the society, the management of solid wastes generated by the increase in consumption amount, without harming the environment and human health and at the least cost, for municipals is one of the priority problems. One of the biggest cost items of municipals is waste management, and the largest share in this is carriage freight. Therefore, municipals, on the one hand, try to collect solid waste in the most effective way without harming people and the environment, and on the other hand, they try to minimize the carriage freight. In this study, it is aimed to minimize the total distance during the solid waste collection process in the that region by determining the location of some garbage containers in Çankaya district of Ankara province. A mathematical model was proposed in order to minimize the total distance traveled, and due to the size of the problem, a solution method based on the demon algorithm, one of the heuristic methods, was developed. The solutions of the developed algorithm and the solutions of the proposed mathematical model with small number of nodes that can be solved using GAMS were compared, subsequently very large scale solutions were made with the developed algorithm. Optimal solution results in small problem examples that can be solved using GAMS have been achieved in a very short time.

Science Code : 112304
Key Words : Solid Waste Collection, Vehicle Routing Problem, Demon Algorithm
Page Number : 67
Supervisor : Prof. Dr. Şenol ALTAN

TEŞEKKÜR

Doktora çalışmalarım boyunca değerli katkılarıyla beni yönlendiren danışmanım Prof.Dr. Şenol ALTAN'a, tez izleme komitesi üyeleri Prof.Dr. Mehmet KABAK ve Prof.Dr. Murat ATAN'a , çalışmalarım sırasında desteğini benden esirgemeyen eşim Elçin ÖZTÜRK KESKİN ve oğlum Kuzey KESKİN'e teşekkürü bir borç bilirim.

İÇİNDEKİLER

Sayfa

ÖZET iv

ABSTRACT v

TEŞEKKÜR vi

İÇİNDEKİLER vii

ÇİZELGELERİN LİSTESİ ix

ŞEKİLLERİN LİSTESİ x

KISALTMALAR xi

1. GİRİŞ 1

2. ARAÇ ROTALAMA PROBLEMİ VE ÇÖZÜM YÖNTEMLERİ 3

2.1. Araç Rotalama Problemi 3

2.2. Araç Rotalama Proleminin Türleri 4

2.2.1. Dinamik ve Statik Çevre Durumuna Göre ARP 5

2.2.2. Rotalama Durumuna Göre ARP 5

2.2.3. Yolların Durumuna Göre ARP 6

2.2.4. Kısıtlarına Göre ARP 6

2.3. Araç Rotalama Problemi Çözüm Yaklaşımları 8

2.3.1. Kesin Çözüm Yöntemleri 9

2.3.2. Sezgisel Çözüm Yöntemleri 9

2.3.2.1. Klasik sezgisel çözüm yöntemleri 10

2.3.2.2. Metasezgisel çözüm yöntemleri 10

2.3.2.2.1. Tavlama benzetimi 11

2.3.2.2.2. Genetik algoritma 12

2.3.2.2.3. Tabu arama 13

2.3.2.2.4. Demon algoritması 14

Sayfa

3. KATI ATIK TOPLAMA ROTA OPTİMİZASYONU PROBLEMİ 19
3.1. Literatür Taraması 19
3.2. Problemin Tanımı 24
3.3. Problemin Varsayımları 25
3.4. Problemin Matematiksel Modeli 25
3.5. Problemin Küçük Veri Setinin Matematiksel Model ile Çözümü 28
4. SEZGİSEL YÖNTEM 33
4.1. Geliştirilen Algoritmanın Adımları 33
4.2. Algoritmada Kullanılan Yerel Arama Operatörleri 34
4.2.1. Takas Operatörü Uygulanışı 34
4.2.2. Taşı Operatörü Uygulanışı 36
4.2.3. 2-Opt Operatörü Uygulanışı 37
4.3. Problemin Küçük Veri Setinin Geliştirilen Sezgisel Yöntem ile Çözümü 39
4.4. Problemin Geliştirilen Sezgisel Yöntem ile Çözümü 41
5. SONUÇ VE ÖNERİLER 45
KAYNAKLAR 47
EKLER 51
EK-1. Problemin çözüm sonuçları 52
ÖZGEÇMİŞ 67

ÇİZELGELERİN LİSTESİ

Çizelge **Sayfa**

Çizelge 2.1. Demon algoritması adımları 14

Çizelge 2.2. Örnek Demon algoritma çözümü..... 15

Çizelge 2.3. Tavlamalı Sınırlandırılmış Demon Algoritması adımları..... 17

Çizelge 2.4. Örnek Tavlamalı Sınırlandırılmış Demon Algoritma çözümü 18

Çizelge 3.1. Veri setinde kullanılan noktaların koordinatları 30

Çizelge 3.2. Matematiksel modelin çözümünden elde edilen sonuçlar..... 32

Çizelge 4.1. Geliştirilen algoritmanın adımları..... 33

Çizelge 4.2. Takas operatörünün uygulanışı (Rota içi)..... 35

Çizelge 4.3. Takas operatörünün uygulanışı (Rotalar arası)..... 35

Çizelge 4.4. Taşı operatörünün uygulanışı (Rota İçi)..... 36

Çizelge 4.5. Taşı operatörünün uygulanışı (Rotalar arası) 37

Çizelge 4.6. 2-Opt operatörünün uygulanışı (Rota içi)..... 38

Çizelge 4.7. 2-Opt operatörünün uygulanışı (Rotalar arası)..... 38

Çizelge 4.8. Sezgisel algoritma kullanılarak elde edilen çözüm sonuçları..... 40

Çizelge 4.9. Farklı iterasyon ve Demon değerleri için ortalama ve en iyi çözüm sonuçları..... 42

Çizelge 4.10. Yüksek iterasyon sayılı en iyi çözüm sonuçları 43

ŞEKİLLERİN LİSTESİ

Şekil **Sayfa**

Şekil 2.1. ARP İçin örnek bir çözüm gösterimi 4

Şekil 2.2. Açık ve kapalı uçlu ARP görseli 5

Şekil 2.3. ARP çözüm yöntemleri sınıflandırması..... 9

Şekil 3.1. Katı atık toplama problemi 24

Şekil 3.2. Veri setinde kullanılan noktalar..... 29

Şekil 3.3. GAMS sonuç ekranı 30

Şekil 4.1. Sezgisel algoritma sonuç ekranı 39

KISALTMALAR

Bu çalışmada kullanılmış kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Kısaltmalar	Açıklamalar
ARP	Araç Rotalama Problemi
AUARP	Açık Uçlu Araç Rotalama Problemi
AYARP	Asimetrik Yollu Araç Rotalama Problemi
AYY	Atık Yönetimi Yönetmeliği
CBS	Coğrafi Bilgi Sistemi
D	Demon Değeri
DA	Demon Algoritması
DARP	Dinamik Araç Rotalama Problemi
GA	Genetik Algoritma
KKARP	Kapasite Kısıtlı Araç Rotalama Problemi
KUARP	Kapalı Uçlu Araç Rotalama Problemi
MKARP	Mesafe Kısıtlı Araç Rotalama Problemi
PSO	Parçacık Sürü Optimizasyonu
SARP	Statik Araç Rotalama Problemi
SYARP	Simetrik Araç Rotalama Problemi
TA	Tabu Arama
TB	Tavlama Benzetimi
TDARP	Toplama Dağıtım Kısıtlı Araç Rotalama Problemi
ZPARP	Zaman Pencereli Araç Rotalama Problemi

1. GİRİŞ

Atık kavramı; Resmi Gazetede 02 Nisan 2015 tarihinde yayımlanan Atık Yönetimi Yönetmeliği'ne (AYY) göre en geniş tanımıyla "Üreticisi veya fiilen elinde bulunduran gerçek veya tüzel kişi tarafından çevreye atılan veya bırakılan ya da atılması zorunlu olan herhangi bir madde veya materyal olarak tanımlanmaktadır. Ülkemizde atıkların toplanması, taşınması sorumluluğu Belediyelere ve Büyükşehir Belediyelerine verilmiştir. AYY'nin Belediyelerin görev ve sorumlulukları başlıklı *8.maddesinin Büyükşehir Belediyeleri için geçen 1.fıkrasının a bendinde;"Sorumlulukları çerçevesinde atık işleme tesislerini kurmak/kurdurmakla, işletmek/işlettirmekle, ilgili tesislere çevre lisansı almak/aldırmakla," şeklinde açıklanmış ve Büyükşehir İlçe Belediyeleri için yine 8. maddenin 3.fıkrasının ç bendinde: "Belediye atıkları ile ilgili mevzuat kapsamında yönetiminden sorumlu olduğu atıkları kaynağında ayrı toplamak/toplattırmakla, aktarma istasyonuna taşımakla ve ikili toplama sistemi ile atık getirme merkezi kurmak/kurdurtmakla, toplanan atıklara ilişkin bilgi ve belgeleri Bakanlığa sunmakla*, yükümlü oldukları belirtilmiştir.

Nüfus artışı ve toplumun ekonomik düzeyindeki artışa paralel tüketim miktarının artması ile oluşan katı atıkların çevre ve insan sağlığına zarar vermeksizin ve en az maliyetle yönetimi yerel yönetimler için çözülmesi gereken öncelikli sorunlardan biridir.

Türkiye İstatistik Kurumu Adrese Dayalı Nüfus Kayıt Sistemi Sonuçlarına göre, 2010 yılında Türkiye nüfusu 73 milyon 722 bin 988 kişi, 2014 yılında 77 milyon 695 bin 904 kişi ve 2020 yılında 83 milyon 614 bin 362 kişi iken söz konusu yıllarda belediyeler tarafından toplanan toplam atık miktarı da sırasıyla 25,28 milyon ton, 28 milyon ton ve 32,3 milyon ton olarak hesaplanmıştır. Bu veriler ışığında nüfus artışına paralel olarak belediyeler tarafından toplanan toplam atık miktarının da aynı oranda arttığı gözlemlenmektedir.

Yerel yönetimlerin en büyük maliyet kalemlerinden birisi atık yönetimi ve buna bağlı olarak atıkları toplamak ve toplanan atıkları taşımak maksadıyla kullanılan araçların giderleridir. Bu giderlerin içindeki en büyük payı da ulaşım giderleri oluşturmaktadır. Atık miktarının artmasına paralel olarak söz konusu maliyetlerde artmaktadır.

Bu yüzden yerel yönetimler bir yandan katı atıkları insan ve çevreye zarar vermeden en etkili bir şekilde toplamaya çalışırken diğer yandan da oluşan ulaşım maliyetini en aza indirmeye çalışmaktadır.

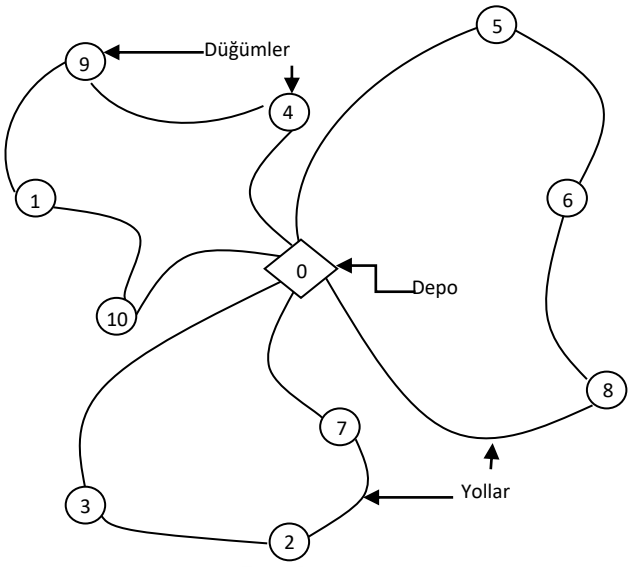
Bu çalışma kapsamında, Ankara ili Çankaya ilçesinde bulunan çöp konteynerlerinin yerleri çıkarılarak katı atık toplama süreci incelenmiş ve toplam mesafenin minimize edilerek rota optimizasyonunun sağlanması amaçlanmıştır. Ulaşım maliyetini minimize edebilmek amacıyla matematiksel model önerilmiş, problemin büyüklüğünden ötürü sezgisel yöntemlerden Demon algoritmasını esas alan bir çözüm yöntemi geliştirilmiştir. Geliştirilen algoritmanın çözümleri ile önerilen matematiksel modelin GAMS kullanılarak çözülebilen küçük düğüm sayılı çözümleri karşılaştırılmış, en iyi çözüm sonuçlarına çok kısa sürede ulaşılmıştır. Müteakiben geliştirilen algoritma ile çok büyük boyutlu çözümler yapılmıştır. Çalışmanın ikinci bölümünde, ARP ve çözüm yöntemleri ile ilgili bilgi verilmiş, çalışmanın üçüncü bölümünde, literartür taraması yapılmış, problemin matematiksel modeli verilmiş, söz konusu matematiksel model ile problemin küçük boyutlu çözümleri yapılmıştır. Dördüncü bölümde sezgisel yöntem kapsamında geliştirilen algoritma detaylı şekilde anlatılmış, geliştirilen algoritmanın çözüm sonuçları incelenmiş ve son bölümde çalışmaya yönelik değerlendirme yapılmış ve gelecek çalışmalar hakkında bilgi verilmiştir.

2. ARAÇ ROTALAMA PROBLEMİ VE ÇÖZÜM YÖNTEMLERİ

Bu bölümde Araç Rotalama Problemi (ARP), türleri ve çözüm yöntemleri (Kesin Çözüm Yöntemleri, Sezgisel Çözüm Yöntemleri ve Metasezgisel Çözüm Yöntemleri) hakkında bilgi verilmiştir. Problemi çözmek için önerilen sezgisel yöntemin temelini oluşturan Demon Algoritması çözüm örnekleri ile birlikte anlatılmıştır.

2.1. Araç Rotalama Problemi

Araç Rotalama Problemi ilk olarak Dantzig ve Ramser tarafından 1959 yılında çalışılmıştır. Yazarlar çalışmalarında gerçek hayat problemi olan benzin istasyonlarına benzin dağıtımını ele almışlardır. ARP, sorunu ve problemi çözmede gerekli olan hesaplama gücünün problemin boyutuyla birlikte üssel olarak arttığı NP-zor problemler sınıfına giren ve literatürde yaygın olarak çalışılan bir tamsayılı programlama problemidir. ARP'de belirli müşterilere hizmet eden araçların izleyeceği güzergahların en iyisinin belirlenmesi için çalışılır. Bu tarz problemlerde doğru sonuca hızlı bir şekilde erişebilmek için yaklaşık sonuçların bulunması hedeflenmektedir. Bu yüzden çözüm için genellikle, problemin doğasını anlamaya çalışan çeşitli sezgisel yöntemler kullanılır. Clark ve Wright 1964 yılında yaptıkları çalışmada ARP'ye sezgisel yöntem önermişlerdir (Şeker, 2007). ARP, konum olarak merkezleri dağınık olan depolar kapsamında, bir veya birden fazla depodan servis oluşturulmasında görevlendirilen araçlar için oluşturulmuş optimum dağıtım ve toplama rotalarının planlanması problemleridir (Güngör ve Ergülen, 2006). ARP için örnek bir çözüm gösterimi şekil 2.1'de gösterilmiştir.



Şekil 2.1. ARP için örnek bir çözüm gösterimi (Bell ve McMullen, 2004)

Araç Rotalama Problemleri ile ilgili literatürde bulunan birçok çalışma aşağıdaki genel varsayımlara sahiptir (Caccetta ve Hill, 2001).

- İşletmenin elinde tek tip ticari mal bulunmaktadır, müşterilerin talepleri önceden bilinmektedir ve dağıtımlar tek depodan sağlanmaktadır.
- Her müşterinin talebi yalnızca bir tane araç tarafından karşılanır.
- Tüm araçların kapasiteleri aynıdır ve rotada tek tur yaparlar.
- Her bir aracın kat edebileceği mesafe bellidir ve bu mesafe aşılamaz.
- Her müşteri belirli bir zaman aralığında ziyaret edilmelidir.
- Amaç bünyede bulunan araçlar tarafından gidilen toplam yolu minimize etmektir.

2.2. Araç Rotalama Proleminin Türleri

Literatürde çok farklı ARP türü bulunmaktadır. Problem türü problemin kısıtlarına göre farklılık göstermektedir. ARP, günümüzde gelişen koşullarla beraber birçok kısıt içermektedir. Belirli zaman aralıkları (servis zamanı, kalkış zamanı, bekleme zamanı vb.), kapasite farklılıklarına sahip araçlar, işlem görecekleri rota üzerinde izin verilen seyahat süresi, gidilen rotalara göre değişen hızlar, şoförler için gereken dinlenme zamanları bu kısıtlar arasına girmektedir (Erel, 1995). Bunun yanında literatürde farklı sınıflandırma çalışmaları da mevcuttur. Genel olarak dinamik ve statik çevre durumuna göre, rotalama durumuna göre, kısıtlara göre ve yolların durumuna göre ARP olmak üzere ayrılmaktadır.

2.2.1. Dinamik ve Statik Çevre Durumuna Göre ARP

Statik Araç Rotalama Problemlerinde (SARP), problemle ilgili tüm bilgiler (talepler, kısıtlar, maliyet bilgileri vb.) rota oluşturulmadan yani problem çözülmeden önce bilinmektedir. Bu veriler sabit kabul edilir ve problemin çözümü sırasında da değişmez. Dinamik Araç Rotalama Problemlerinde (DARP) ise söz konusu bilgiler yeni talep noktalarının ortaya çıkması, trafik yoğunluğu, bazı yolların kapanması, müşterinin talep durumunda meydana gelen değişiklik gibi durumların ortaya çıkmasıyla, faaliyet devam ederken dinamik olarak değişmeye devam eder. SARP, müşteri ve tedarikçilerin miktarı ve zamanı belli olan talepleri için, DARP ise müşterilerine kurye yoluyla anlık olarak değişkenlik gösterebilen işletmelerde kullanılır (Erol, 2006).

2.2.2. Rotalama Durumuna Göre ARP

Rotaların başlangıç ve bitiş durumlarına göre açık uçlu ve kapalı uçlu ARP olarak iki farklı biçimde incelenir. Açık uçlu ve kapalı uçlu ARP görseli şekil 2.2'de verilmiştir.



Şekil 2.2. Açık ve kapalı uçlu ARP görseli

Açık Uçlu Araç Rotalama Problemi (AUARP); rotaların başlangıç ve bitiş noktalarının farklı olduğu problem türleridir. Araçların tura belirli bir depodan başlayıp turu farklı bir depoda veya müşteri noktasında bitirme durumudur.

Kapalı Uçlu Araç Rotalama Problemi (KUARP) ise rotaların başlangıç ve bitiş noktasının aynı olduğu problem türleridir. Araçların tura belirli bir depodan başlayıp turu aynı depoda bitirme durumudur. Çöp toplama araçlarının belirli bir depodan başlayarak çöpleri toplamaları, müteakiben çöpleri boşaltım noktasına dökerek başladıkları depoya dönmeleri KUARP olarak örneklendirilebilir (Dişkaya, 2018).

2.2.3. Yolların Durumuna Göre ARP

Yolların durumuna göre ARP, geliş gidiş mesafelerinin eşit olup olmamasına göre ikiye ayrılır. Simetrik Araç Rotalama Problemlerinde (SYARP) bir noktadan diğer noktaya olan gidiş dönüş mesafeleri birbirine eşittir ($d_{ij} = d_{ji}$). Bazı durumlarda ise bir noktadan diğer noktaya olan gidiş dönüş mesafesi eşit olmayabilir. Bu tip problemler de Asimetrik Yollu Araç Rotalama Problemi (AYARP) olarak adlandırılmaktadır. Bu tip problemlerde araçların hangi müşteriye ilk olarak hizmet edeceği önem kazanmakta, böylece rota mesafesinin belirlenmesini sağlamaktadır (Erol, 2006).

2.2.4. Kısıtlarına Göre ARP

ARP, kısıtlar göz önüne alındığında araçların yükleme kısıtı, yolların mesafe kısıtı, teslim sürelerine ve çalışanların çalışma sürelerine ilişkin zaman kısıtı, yükleme ve boşaltmaya yönelik kısıtlara göre sınıflandırılmaktadır.

Kapasite kısıtlı ARP (KKARP); belirli bir sayıda deposu bulunan işletmede, taleplerin belirli sayıdaki müşteriye ulaştırılabilmesi için rota planlaması problemidir. Müşterilerin ihtiyaçlarını karşılamak için kullanılan araçların hepsi aynı kapasitedir. Aynı rota üzerinde bulunan müşterilerin talepleri toplamı araç kapasitesini aşamaz. Müşterilerin talepleri bölünemez. KKARP'nde amaç toplam rota maliyetinin minimize edilmesidir (Takan, 2019). KKARP için literatürde farklı modeller bulunmaktadır. Yılmaz (2018) tarafından yapılan çalışmada;

İndisler:
i,j = müşteriler
parametreler:
Q = Araç yükleme kapasitesi.
q_i = i müşterisinin talep miktarı.
M = Araç sayısı.
N = Müşteri sayısı.
C_{ij} =i müşterisi ile j müşterisi arasındaki mesafe
y_i = Alt turları engellemek için kullanılan rastgele değişken

Karar Değişkeni:

$$X_{ijk} = \begin{cases} 1, \ k \ aracı \ i \ müşteri \sin den \ j \ müşteri \sin e \ giderse \\ 0, \ d.d. \end{cases}$$

Model:

$$Min\ Z = \sum_{i=0}^{N}\sum_{j=0}^{N}(c_{ij}.\sum_{k=1}^{M}x_{ijk}) \tag{2.1}$$

st.

$$\sum_{i=0}^{N}\sum_{k=1}^{M}x_{ijk} = 1, \qquad \forall j \ (j \neq 0) \tag{2.2}$$

$$\sum_{i=0}^{N}x_{ipk} - \sum_{j=0}^{N}x_{pjk} = 0 \qquad \forall k, p \ (p \neq 0) \tag{2.3}$$

$$\sum_{i=1}^{N}(q_i.\sum_{j=0}^{N}x_{ijk}) \leq Q \qquad \forall k \tag{2.4}$$

$$\sum_{j=1}^{N}x_{0jk} = 1 \qquad \forall k \tag{2.5}$$

$$y_i - y_j + N.\sum_{k=1}^{M}x_{ijk} \leq N-1 \qquad \forall i, j \ (i \neq j \neq 0) \tag{2.6}$$

$$x_{ijk} \in (0,1) \qquad \forall i, j, k \tag{2.7}$$

Eş. 2.1'de toplam maliyetin minimize edilmesi amaçlanmaktadır. Eş. 2.2'de her müşterinin sadece bir defa ziyaret edilmesi sağlanmaktadır. Eş. 2.3'te ziyaret edilen müşteriden tekrar hareket edileceğini ifade eder. Eş. 2.4 ile araçların yük taşıma kapasitesinin aşılmaması sağlanmaktadır. Eş. 2.5'te araçların sadece bir rotada kullanılabileceği belirtilmektedir.

Eş. 2.6 ile alt turların oluşması engellenmektedir. Eş. 2.7'de karar değişkenleri tanımlanmaktadır.

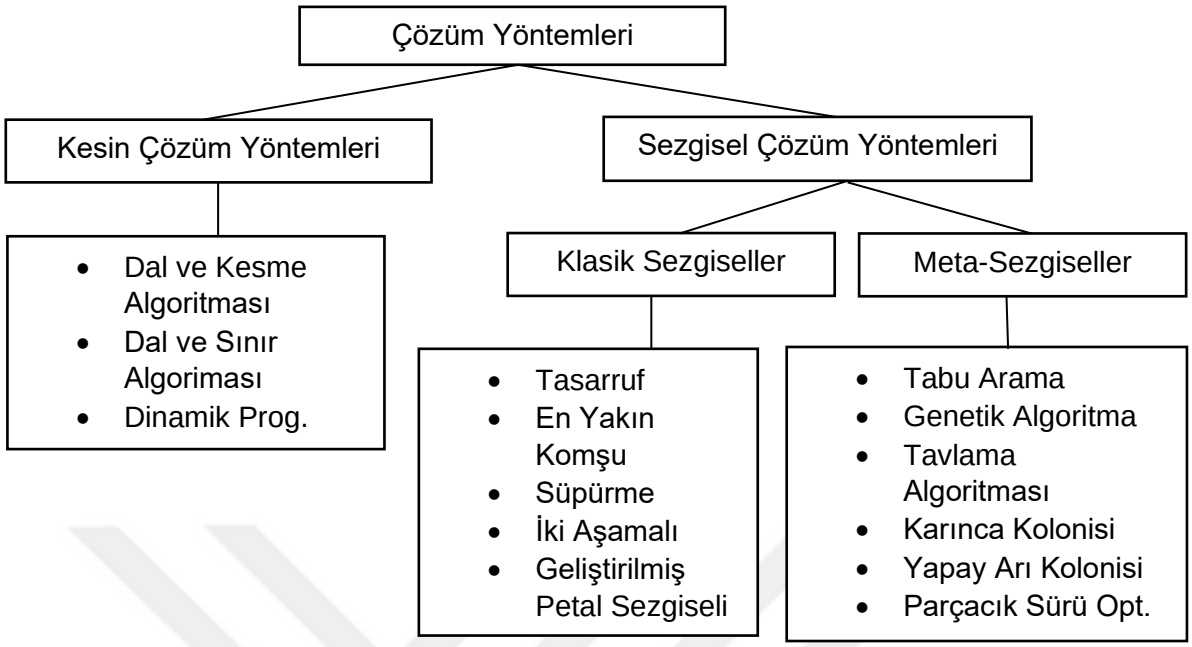
Mesafe kısıtlı ARP (MKARP); gidilecek her rota için kapasite kısıtının yerine, her bir rotada bulunan araçların maksimum mesafe kısıtının bulunduğu problemlerdir.

Zaman pencereli ARP (ZPARP); müşteriler taleplerinin belirli bir zaman aralığında karşılanmasını isteyebilirler. Söz konusu zaman aralığı "zaman penceresi" olarak adlandırılmaktadır. ZPARP'de temel amaç belirtilen zaman penceresi diliminde müşteri taleplerini minimum maliyetle karşılayacak rotaların oluşturulmasıdır.

Toplama dağıtım kısıtlı ARP (TDARP); toplama ve dağıtma işlemlerinin birlikte yapıldığı rotalama problemidir. Dağıtımı yapılacak ürünler depodan başlarken toplanacak ürünler depoya getirilir. Ancak müşteriler arasında herhangi bir ürün alış verişi bulunmaz. Temel amaç araç akışı ve sürenin minimize edilmesidir. Aynı araçta toplanacak ve dağıtılacak ürün miktarının araç kapasitesini aşmaması gerekir (Yılmaz, 2018). Gıda sektöründe marketlere yeni ürün dağıtılırken, tarihi geçmiş ürünlerin toplanması bu problem türüne örnek verilebilir.

2.3. Araç Rotalama Problemi Çözüm Yaklaşımları

Araç rotalama problemi NP-zor problem sınıfına dahildir. Bu yüzden değişken sayısı fazla olan problemler için çözüme ulaşmak oldukça zor olmaktadır. Problemlerde kısıt sayısı arttıkça problemde daha karmaşık hale gelmekte ve istenilen optimum sonucu elde etmek zorlaşmaktadır. Bu nedenle kesin sonuçtan ziyade kesin sonuca en yakın olanı hızlı bir şekilde verebilen yöntemler kullanılmaktadır (Ekizler, 2011). ARP çözüm yöntemleri genellikle optimal sonuca ulaşıp ulaşmamalarına göre değerlendirilmekte, kesin çözüm yöntemleri ve sezgisel çözüm yöntemleri olarak ikiye ayrılmaktadır. ARP çözüm yöntemleri sınıflandırması Şekil 2.3 'te sunulmuştur.



Şekil 2.3. ARP çözüm yöntemleri sınıflandırması

2.3.1. Kesin Çözüm Yöntemleri

Kesin çözüm yöntemleri optimum sonuçlar elde edebilen matematiksel programlama tabanlı yöntemlerdir. Fakat büyük ölçekli problem türlerini bu çözüm yöntemlerini kullanarak çözmek oldukça uzun zaman almaktadır. Bu yöntemlerin ortak özellikleri optimum sonucu vermeleridir.

Laporte (1992)'ye göre, tamsayılı model olarak formüle edilen araç rotalama probleminin çözümü üç genel kategoride sınıflandırılabilir:

- Doğrudan ağaç arama metotları; dal-ve-sınır, dal-ve-kesme, dal-kesme-değer,
- Dinamik programlama,
- Tamsayılı doğrusal programlama, küme ayrıştırma (bölümleme) ve kolon üretimi, üç indeksli araç akış formülasyonu, iki indeksli araç akış formülasyonu, iki-ürün ağ akış formülasyonu.

2.3.2. Sezgisel Çözüm Yöntemleri

Araç Rotalama Problemleri büyük ve karmaşık yapıdaki problemler olduğundan optimum sonuca ulaşmak zorlaşmakta ve uzun uğraşlar gerektirmektedir. Bu yüzden sezgisel

yöntemlerin kullanımı bir gereklilik haline gelmiştir. Sezgisel yöntemler, Klasik Sezgiseller ve Meta Sezgiseller olarak iki gruba ayrılmaktadır.

2.3.2.1. Klasik sezgisel çözüm yöntemleri

ARP için geliştirilen ve optimal sonucu veren tamsayılı modeller küçük ölçekli problemlerde uygulanabilir. Ancak günlük hayatta karşılaşılan problemler büyük ölçekli problemlerdir. Hizmet edilen müşteri sayısı arttıkça ve coğrafi alan büyüdükçe hesaplama zorlaşmaktadır. Bu yüzden optimum olmasa da yakın çözümlere hızlı bir şekilde ulaşılması istenir. Bu nedenle sezgisel yöntemler geliştirilmiştir (Yılmaz, 2018). Eryavuz ve Gencer (2001)'e göre Klasik Sezgisel algoritmalar üçe ayrılmaktadır.

- Tur Kurucu Sezgiseller
- Tur Geliştirici Sezgiseller
- İki Aşamalı Metotlar

2.3.2.2. Metasezgisel çözüm yöntemleri

Metasezgisel yöntemler diğer çözüm yöntemlerine göre daha pratiktirler. Büyük ölçekli problem örneklerinin çözümü klasik sezgisel yöntemler ile daha uzun zaman alırken metasezgisel yöntemler ile daha kabul edilebilir sürelerde çözüme ulaşılır (Dişkaya, 2018).

Metasezgisel yöntemlerde amaç; umut vadeden bölgelerde ayrıntılı arama yaparak optimal sonuca yakın sonuçlar elde etmektir. Klasik sezgisel yöntemler yerel minimum noktalarına takılırken, Metasezgisel yöntemler yerel minimum noktalarına takılmazlar. Bu yöntemler karmaşık komşuluk arama kurallarından ve çözüm kombinasyonlarından çözüm üretirler. Bu yöntemlerle elde edilen sonuçlar, klasik sezgisel yöntemlerle elde edilen sonuçlardan daha iyidir (Erol, 2006).

Blum ve Roli (2003) yaptıkları çalışmada metasezgisel yöntemlerin karakteristik özelliklerini belirtmişlerdir. Buna göre;

- Metasezgisel yöntemler arama sürecini yönlendiren stratejilerdir.
- Amaç, optimal veya optimale yakın sonuçlar bulmak için arama uzayını etkili bir şekilde keşfetmektir.

- Metasezgisel algoritmaları oluşturan teknikler lokal arama yöntemlerinden, karmaşık öğrenme prosedürlerine uzanır.
- Metasezgisel algoritmalar yaklaşık yöntemlerdir ve genellikle deterministik değildir.
- Arama uzayının dar alanlarında sıkışmaktan kaçınmak maksadıyla mekanizmalar içerebilirler.
- Metasezgisel yöntemler, soyut düzeyde tanımlamaya izin verir.
- Metasezgisel yöntemler belirli bir probleme özgü değildir.

En sık kullanılan metasezgisel yöntemlerden bazıları Tavlama Benzetimi, Genetik Algoritma ve Tabu Arama algoritmalarıdır.

2.3.2.2.1. Tavlama benzetimi

Tavlama Benzetimi (TB), Kirkpatrick, Gerlatt ve Vechi (1983) tarafından önerilmiştir. TB, metasezgisel yöntemler arasında bilinen en eski algoritma olup, yerel minimumlara takılmayan stratejiye sahiptir. Algoritmada temel fikir, yerel minimumlardan kurtulmak için mevcut çözümden daha kötü çözümlere hareket edilmesidir. Kötü çözümlerin kabul edilme olasılığı arama süresince düşürülür. Düşürme süreci camların ve metallerin tavlama işlemine benzer (Blum ve Roli, 2003). TB algoritmasının adımları şu şekildedir (Breedam, 2001; Erol, 2006).

Adım 1. Bir başlangıç çözümü üret ve bu çözümü en iyi çözüm "S" olarak ata. Aynı zamanda iterasyon sayacını (t), "0" olarak ata.

Adım 2. Bir başlangıç sıcaklık değeri "T_B" belirle ve bunu mevcut sıcaklık değeri "T_0" olarak ata. ($T_0 = T_B$)

Adım 3. Mevcut çözümden rastgele hareketle komşu bir çözüm "S^1" oluştur. $S^1 \in N(S)$

Adım 4. Yeni üretilen "S^1" çözümüyle "S" çözümünün amaç fonksiyonu değerleri arasındaki farkı bul. $\delta = C(S^1) - C(S)$

Adım 5. Eğer S^1 çözümü, S çözümünden daha iyi ise ($\delta < 0$), "S" çözümünü "S^1"çözümüne ata. Eğer "S^1" çözümü, S çözümünden daha kötü ise ($\delta > 0$) ancak T_t

mevcut sıcaklıkta ($e^{-\delta/T}$) > Q, koşulu sağlanıyorsa, “S” çözümünü “S^1”çözümüne ata. Sağlanmıyorsa mevcut "*S*" çözümünü koru.

Adım 6. T_t sıcaklık değerini aşağıdaki formüllerden birine göre revize et.

$T_t = R \cdot T_t\text{-}1$ $(0 < R < 1)$

$T_t = t / (1 + \beta \cdot t)$ (β uygun küçük bir değerdir)

Adım 7. Eğer, durdurma kriteri sağlanırsa araştırmayı durdur, sağlanmazsa “t” iterasyon sayacını 1 artırarak adım 3’e git.

2.3.2.2.2. Genetik algoritma

Genetik Algoritma (GA), canlıların doğadaki değişikliklere adapte olabilme yeteneğinden esinlenerek geliştirilen bir algoritmadır. Her bir iterasyonda seçilen çözümlerin bazılarına, yeni çözüm oluşturmak için genetik operatörler uygulanır. Genetik Algoritmada amaç fonksiyonu değeri daha iyi çözümlerin, diğer çözümlere göre seçilme olasılığı daha fazladır (Blum ve Roli, 2003). GA, problemde tek çözüm üretmekten ziyade farklı çözümlerin birleşiminden oluşan çözüm kümesi üretir. Böylece arama bölgesinde farklı birçok nokta değerlendirilerek çözüme ulaşılma olasılığı artmaktadır. Algoritmada çözüm grubu popülasyon, kodlar kromozom olarak adlandırılır. GA adımları şöyledir (Goldberg, 1989; Yılmaz, 2018).

Adım 1. Mümkün çözümlerin kodlandığı bir çözüm grubu oluşturulur.

Adım 2. Uygunluk fonksiyonu kullanılarak her bir kromozomun ne kadar iyi olduğu belirlenir. Bu işlem evrimleşme olarak adlandırılır.

Adım 3. Popülasyondaki çözümler belirlenen uygunluk değerlerine göre eşlenir, çaprazlama ile mutasyon işlemlerine tabii tutulur. Böylece, yeni bir toplum oluşturulur. Çaprazlama işlemi çeşitlilik, mutasyon işlemi yerel optimumlardan kurtulmayı sağlar.

Adım 4. Eski kromozomların çıkartılmasıyla yeni kromozomlara yer açılır, böylece sabit büyüklükte toplum sağlanır. Müteakiben tüm kromozomların uygunluk fonksiyonu değerleri tekrar hesaplanır böylece yeni toplumun başarısı bulunur.

Adım 5. Adımlar tekrarlanarak daha iyi yeni nesillerin oluşturulması sağlanır.

Adım 6. En iyi bireyler bulununca çözüm elde edilmiş olunur.

2.3.2.2.3. Tabu arama

Tabu Arama (TA), Glover tarafından 1986 yılında geliştirilmiştir. Bu yöntemde lokal optimumlara takılmamak için daha önce yapılan aramalarda elde edilen bilgiler kullanılır. Tabu Arama yönteminde komşu çözümlerde arama yapılır. Amaç fonksiyonuna göre komşular değerlendirmeye tutulur. Değerlendirilen komşu Tabu listesinde yoksa daha önce bulunan en iyi çözümle karşılaştırılır ve listeye eklenir. Çözümün en iyi çözümden daha iyi olması durumunda bir sonraki çözümde başlangıç çözümü olarak kabul edilir. Sonlandırma koşulu sağlanıncaya kadar tekrar edilir. TA yönteminin adımları şöyledir (Glover, 1989; Şeker, 2007).

Adım 1. Başlangıç çözümü belirle bu çözümü, mevcut çözüm aynı zamanda en iyi çözüm olarak kaydet.

Adım 2. Bir komşuluk yapısı kullanarak mevcut çözümün komşularını üret.

- Tabu olmayan ya da aspirasyon ölçütünü sağlayan bir komşu çözüm seç ve bu çözümü yeni çözüm olarak al.
- Mevcut çözümden yeni çözüme taşıma özelliğini tabu listesine ekleyerek listeyi güncelle.
- Yeni çözümün o ana kadar bulunan en iyi çözümden daha iyi olması durumunda bu çözümü yeni en iyi çözüm olarak kaydet.

Adım 3. Durdurma kriteri sağlanıncaya kadar Adım 2`yi tekrarla

2.3.2.2.4. Demon algoritması

Demon Algoritması (DA), kabul fonksiyonu demon'un enerji değerine dayanan tavlama benzetimi tabanlı bir algoritmadır. Algoritma önceden belirlenmiş pozitif bir demon (D) değeri ile başlatılır. İyileşme sağlamayan yeni bir çözümde eğer demon enerji değeri amaç fonksiyonunda meydana gelen değişimden büyük ise kabul edilir ve demon enerji değeri düşürülür. Benzer şekilde iyileşme sağlayan yeni çözüm için demon enerji değeri yükseltilir. Böylece demon değeri elde edilen çözüme bağlı olarak dinamik bir şekilde değişir (Talbi, 2009). Demon Algoritmasının adımları Çizelge 2.1'de gösterilmiştir.

Çizelge 2.1. Demon algoritması adımları (Pepper ve diğerleri, 2002)

Adım 1:Başlangıç çözümünü bul, S = S0
Adım 2: Demon değerini belirle, D > 0
Adım 3: Sonlandırma koşulu sağlanana kadar tekrarla
a. Yeni çözümü bul, S^1
b. $\Delta E = E(S^1) - E(S)$
c. Eğer $\Delta E < D$ ise
yeni çözümü kabul et $S = S^1$ ve
demon değerini güncelle $D = D - \Delta E$
Adım 4: Sonlandırma koşulu sağlanınca bitir.

Demon Algoritmasının daha iyi anlaşılabilmesi için örnek çözüm Çizelge 2.2'de sunulmuştur. Örnek çözümde;

Başlangıç çözümü (S): 420
Demon değeri (D) : 10
Yeni çözüm ($S^{1)}$: 422 olarak belirlenmiştir. Buna bağlı olarak,
$\Delta E = 422 - 420 = 2$ olarak hesaplanır.
$\Delta E = 2 < D$ olduğu için yeni çözüm kabul edilmiştir ve yeni $D = 10 - 2 = 8$ olarak hesaplanmıştır.
Diğer adımlarda da benzer hesaplamalar yapılarak 412 değerine ulaşılmıştır.

Çizelge 2.2. Örnek Demon algoritma çözümü

Adım	Çözüm (S)	Yeni Çözüm ($S^{1)}$	$\Delta E = S^1 - S$	D	Kabul/Ret ($\Delta E<D$)
1	420	422	2	10	Kabul
2	422	426	4	8	Kabul
3	426	440	14	4	Ret
4	426	418	- 8	4	Kabul
5	418	400	- 18	12	Kabul
6	400	405	5	30	Kabul
7	405	415	10	25	Kabul
8	415	435	20	15	Ret
9	415	390	- 25	15	Kabul
10	390	412	22	40	Kabul
Sonuç	412				

Demon algoritması ilk olarak Wood ve Down (1998) tarafından literatüre kazandırılmıştır. Yazarlar yaptıkları çalışmada Sınırlandırılmış Demon Algoritması, Rassal Sınırlandırılmış Demon Algoritması, Tavlamalı Demon Algoritması ve Rassal Tavlamalı Demon Algoritması olmak üzere Demon Algoritmasının dört farklı versiyonunu önermişler, söz konusu algoritmaları Gezgin Satıcı Problemlerinde test etmişler ve sonuçları karşılaştırmışlardır.

Pepper, Golden ve Wasil (2002), tavlama benzetimi tabanlı 11 sezgisel algoritma içeren bir çalışma yapmışlardır. Söz konusu çalışmada Wood ve Down (1998), literatüre kazandırılan Demon Algoritmasının dört versiyonuna ilave olarak dört farklı versiyon daha önermiş ve sekiz farklı versiyonu test etmişlerdir. Her bir sezgisel algoritmayı 29 Gezgin Satıcı Problemine uygulamışlar, sonuçları ve çalışma sürelerini karşılaştırmışlardır. Çalışma sonucunda Tavlamalı Sınırlandırılmış Demon Algoritmasının, Rassal Tavlamalı Sınırlandırılmış Demon Algoritmasının ve Hibrid Tavlamalı Sınırlandırılmış Demon Algoritmasının Tavlama Benzetimine yakın sonuçlar çıkardığını, genel olarak DA'nın uygulama ve tasarımının kolay olmasının yanında hesaplama süresinin nispeten hızlı olduğunu belirtmişlerdir.

Chandran, Golden ve Wasil, (2003) yaptıkları çalışmada Uniform Demon Algoritması, Triangular Demon Algoritması ve Tavlamalı Uniform Demon Algoritması olmak üzere Demon Algoritmasının üç farklı versiyonunu önermişlerdir. Söz konusu üç algoritmayı, Benzetimli Tavlama Algoritması, Tavlamalı Sınırlandırılmış Demon Algoritması ve Rassal Tavlamalı Sınırlandırılmış Demon Algoritması ile birlikte 36 Gezgin Satıcı Problemi üzerinde test etmişler ve çözüm sonuçlarını karşılaştırmışlardır. Karşılaştırma sonucunda en iyi sonuçları Tavlamalı Sınırlandırılmış Demon Algoritmasının verdiğini belirtmişlerdir.

Yurtdaş (2018) Çapraz Sevkiyatta Araç Rotalama Problemini Tavlamalı Sınırlandırılmış Demon Algoritmasını kullanarak çözmüştür. Çalışmasında literatürde yer alan 36 problem örneğini çözerek çözüm sonuçlarını diğer araştırmacıların çözüm sonuçları ve süreleri ile karşılaştırmıştır. Çözüm sonucunda Tavlamalı Sınırlandırılmış Demon Algoritmasının bazı problem örneklerinde en iyi ortalama ve en iyi çözüm sonuçlarına daha kısa sürede ulaştığını belirtmiştir.

Bu çalışmada Tavlamalı Sınırlandırılmış Demon Algoritması kullanılmıştır. Bu algoritmanın kullanılmasının ana sebepleri;

1. Pepper ve diğerleri (2002) tarafından Tavlama Benzetimine yakın sonuçlar çıkarmasının yanında hesaplama süresinin daha kısa olduğunun belirtilmesi,
2. Chandran, Golden ve Wasil, (2003) tarafından yapılan çalışmada, Demon Algoritmasının altı farklı versiyonunun incelenerek en iyi çözüm sonuçlarını Tavlamalı Sınırlandırılmış Demon Algoritmasının verdiğinin belirtilmesi,
3. Yurtdaş (2018) tarafından bazı problem örneklerinde en iyi ortalama ve en iyi çözüm sonuçlarına daha kısa sürede ulaşıldığının belirtilmesidir.

Demon Algoritması ile Tavlamalı Sınırlandırılmış Demon Algoritmasının adımları arasında bir takım farklılıklar bulunmaktadır. Tavlamalı Sınırlandırılmış Demon Algoritmasının adımları Çizelge 2.3’te gösterilmiştir.

Çizelge 2.3. Tavlamalı Sınırlandırılmış Demon algoritması adımları (Pepper ve diğerleri, 2002)

Adım 1:Başlangıç çözümünü bul, $S = S_0$

Adım 2: Demon değerini belirle, $D = D_{sınır} > 0$

Adım 3: Sonlandırma koşulu sağlanana kadar tekrarla

a. Yeni çözümü bul, S^1

b. $\Delta E = E(S^1) - E(S)$

c. Eğer $\Delta E < D$ ise

yeni çözümü kabul et $S = S^1$ ve

demon değerini güncelle $D = D - \Delta E$

d. Eğer $D > D_{sınır}$ ise demon değerini güncelle $D = D_{sınır}$

e. $D_{sınır}$ değerini Tavlama programına göre düşür

Adım 4: Sonlandırma koşulu sağlanınca bitir.

Tavlamalı Sınırlandırılmış Demon Algoritmasının daha iyi anlaşılabilmesi için örnek çözüm Çizelge 2.4'te sunulmuştur. Örnek çözümde;

Başlangıç çözümü (S): 420

Demon değeri (D) : 10

Demon sınır ($D_{sınır}$) : 10

Yeni çözüm (S_1) : 422 olarak belirlenmiştir. Buna bağlı olarak,

$\Delta E = 422 - 420 = 2$ olarak hesaplanır.

$\Delta E = 2 < D$ olduğu için yeni çözüm kabul edilmiştir ve yeni $D = 10 - 2 = 8$ olarak hesaplanmıştır.

Diğer adımlarda da benzer hesaplamalar yapılarak 390 değerine ulaşılmıştır. $D_{sınır} = 10$ olarak belirlendiğinden problemin bütün adımlarında Demon değeri bu sayıdan küçük veya bu sayıya eşit çıkmıştır. 7'nci adımda da Demon sınır değeri düşürülmüştür.

Çizelge 2.4. Örnek Tavlamalı Sınırlandırılmış Demon Algoritma çözümü

Adım	Çözüm (S)	Yeni Çözüm (S^1)	$\Delta E = S^1 - S$	D	$D_{sınır}$	Kabul/Ret ($\Delta E < D$)
1	420	422	2	10	10	Kabul
2	422	426	4	8	10	Kabul
3	426	440	14	4	10	Ret
4	426	418	- 8	4	10	Kabul
5	418	400	- 18	10	10	Kabul
6	400	405	5	10	10	Kabul
7	405	415	10	5	10	Ret
8	405	435	30	5	9	Ret
9	405	390	- 25	5	9	Kabul
10	390	412	22	9	9	Ret
Sonuç	390					

3. KATI ATIK TOPLAMA ROTA OPTİMİZASYONU PROBLEMİ

Bu bölümde problemin tanımı ve varsayımları belirtilerek literatürde yapılan çalışmalar incelenmiştir. Ayrıca problemin matematiksel modeli oluşturularak küçük problem örnekleri için çözülmüş ve çözüm sonuçları değerlendirilmiştir.

3.1. Literatür Taraması

Genel olarak bir atık toplama sistemi, katı atıkların toplanmasını ve bertaraf edileceği tesislere kadar taşınmasını içerir. Türkiye'de belediyeler tarafından yürütülen bu temel hizmet özellikle kentsel alanlarda, çevre ve nüfus artışı ile beraber kamu kaygısı üzerindeki etkisi nedeniyle birçok araştırmacının ilgisini çekmektedir. Atıkların toplanması üzerine ilk çalışma Beltrami ve Bodin tarafından 1974 yılında yapılmıştır. Atıkların toplanması hizmeti çok yüksek bir işletme maliyeti içerdiğinden, araştırmacılar atık toplama araçlarının rotasını iyileştirerek, bertaraf tesislerinin ve çöp kutularının en uygun yerini bularak ve kullanılan araç sayısını en aza indirerek maliyeti düşürmeye çalışmaktadır.

Benjamin ve Beasley (2010) yaptıkları çalışmada, atıkların toplanması üzerine çalışmışlar ve problemin çözümü için Tabu Arama ve Değişken Komşu Arama algoritmasını kullanan iki metahüristik algoritma ile Değişken Komşu Aramanın, Tabu Arama kullanılarak yapıldığı hibrit bir algoritma önermişlerdir. Çalışma sonucunda önerilen üç algoritmanın çözüm sonuçları ile Kim ve diğerleri (2006) tarafından önerilen algoritmaların çözüm sonuçlarını karşılaştırmışlardır. Karşılaştırma sonucunda Değişken Komşu Arama algoritmasının en etkili yöntem olduğunu belirtmişlerdir. Ayrıca önerdikleri algoritmaların da literatürdeki çalışmalara göre daha iyi sonuçlar verdiğini ifade etmişlerdir.

Islam ve Rahman (2012) yaptıkları çalışmada, atık toplama problemi üzerine çalışmışlar, kullanılan araç sayısının ve toplam katedilen mesafenin minimize edilmesini amaçlamışlardır. Problemin çözümü için Karınca Kolonisi algoritmasına dayanan bir algoritma önermişlerdir. Önerdikleri algoritmayı literatürdeki bazı algoritmalar ile karşılaştırmışlar ve önerdikleri algoritmanın daha iyi sonuçlar verdiğini belirtmişlerdir. Önerilen algoritmanın atık toplama problemleri için uygun olduğunu ifade etmişlerdir.

Buhrkal, Larsen ve Ropke (2012) yaptıkları çalışmada, atıkların toplanması ve taşınması için en uygun maliyetli rotanın bulunmasını amaçlamışlardır. Problemin çözümü için Adaptive Large Neighborhood Search (ALNS) algoritmasını önermişler ve problem çözümlerini literatürdeki örneklerle karşılaştırmışlardır. Önerilen algoritmayla yeni bir gerçek hayat örneğini test etmişler ve önemli iyileştirmelerin sağlandığını belirtmişlerdir.

Markov, Varone ve Bierlaire (2014) yaptıkları çalışmada, geri dönüştürülebilir atıkların toplanması üzerine çalışmışlar, mekansal ve zamansal maliyetleride dikkate alarak toplam maliyeti minimize etmeyi amaçlamışlardır. Problemin çözümü için matematiksel model oluşturmuşlardır. Ayrıca Lokal Arama algoritması geliştirerek sezgisel yöntem önermişlerdir. Küçük ölçekli problem örneklerinde matematiksel modelin çözüm sonuçları ile önerilen sezgisel algoritmanın çözüm sonuçlarını karşılaştırmışlardır. Önerilen algoritmanın çözüm sonuçları ile kesin çözüm sonucu arasındaki farkın %2 den az olduğunu, çalışma süresinin de 1 sn gibi kısa bir sürenin altında olduğunu belirtmişlerdir.

Rabbani, Farrokhi-asl ve Rafiei (2016) yaptıkları çalışmada, atıkların toplanması üzerine çalışmışlar ve toplam maliyetin minimize edilmesini amaçlamışlardır. Problemin çözümü için öncelikle yeni Karışık Tamsayılı Programlama Modeli oluşturmuşlar müteakiben Genetik algoritma ve Genetik algoritma ile En Yakın Komşu algoritmasının birlikte kullanıldığı iki Hibrit Genetik algoritma önermişlerdir. Yapılan küçük ölçekli problem örneklerinde, matematiksel modelin optimal sonucu ile önerilen Hibrit Genetik Algoritmaların çözüm sonuçlarını karşılaştırmışlardır. Matematiksel modelin çözüm sonucu ile Hibrit Genetik algoritma-1'in çözüm sonucu arasında % 2, Hibrit Genetik algoritma-2'nin çözüm sonucu arasında % 4 den daha az bir fark olduğunu belirtmişlerdir. Büyük ölçekli problemlerin çözüm sonuçları karşılaştırıldığında ise Hibrit Genetik algoritmaların, Genetik algoritmaya göre daha iyi sonuçlar verdiğini ifade etmişlerdir.

Assaf ve Saleh (2017) yaptıkları çalışmada, Nablus (Palastine) şehrinde katı atıkların toplanması üzerine çalışmışlar ve toplam katedilen mesafenin minimize edilmesini amaçlamışlardır. Problemin çözümü için matematiksel model ve Genetik algoritma önermişlerdir. Yapılan çalışma sonuçları ile mevcut durumu karşılaştırmışlar ve yapılan çalışma ile toplam katedilen mesafe yönünden % 66 lık bir iyileşme sağlandığını, katı atık toplama zamanının ise 7 saatten 2,3 saate indiğini belirtmişlerdir.

Hannan ve diğerleri (2017) yaptıkları çalışmada, en optimal atık toplama için rota güzergahlarını belirlemeye çalışmışlardır. Problemin çözümü için Geliştirilmiş Parçacıklı Sürü Optimizasyonu (PSO) algoritmasını önermişler ve çözüm sonuçlarını PSO algoritması ve Genetik algoritma ile karşılaştırmışlardır. Önerilen algoritma genel olarak PSO ve GA'ya göre daha iyi perormans göstermesede, 50 düğüme kadar olan veri setlerinde daha iyi sonuçlar verdiğini belirtmişlerdir.

Elgarej, Mansourı, Youssfı, Benmoussa ve El Fazazı (2017) yaptıkları çalışmada, belirli bir bölgede atıkların toplanması üzerine çalışmışlar ve kat edilen toplam mesafenin minimize edilmesini amaçlamışlardır. Problemin çözümü için Karınca Kolonisi algoritmasına dayanan bir algoritma önermişlerdir. Söz konusu algoritmayı küçük ölçekli problem örneklerinde denemişler ve çözüm sonuçlarını Genetik algoritma ile karşılaştırmışlardır. Küçük ölçekli problemlerde genetik algoritmaya göre daha iyi sonuçlar verdiğini belirtmişlerdir.

Erfani, Danesh, Karrabi ve Shad (2017) yaptıkları çalışmada, Ahmadabad (İran) şehrindeki çöp kutularının yerlerinin belirlenmesi ve çöplerin toplanması üzerine çalışmışlardır. Kapasite Kısıtlı Araç Rotalama Problemini Coğrafi Bilgi Sistemlerinde Network Analizi yaparak çözmüşlerdir. Çalışma sonuçlarını mevcut sistemle karşılaştırmışlar ve önerilen yöntemle toplam çalışan personel sayısında % 41,7 , çöpleri toplamak için yapılan günlük tur sayısında % 12,5 ve toplam kat edilen mesafede de % 53 oranında bir iyileşme sağlandığını belirtmişlerdir.

Rızvanoğlu (2018) yaptığı çalışmada, Şanlıurfa ilinin Haliliye ilçesinde örnek bir mahalle olarak seçtiği Veysel Karani Mahallesinde mevcut çöp toplama sistemini gözden geçirmiş, problemin çözümü için hem matematiksel model hem de Coğrafi Bilgi Sistemi (CBS) yazılımı kullanarak katı atık toplama güzergâhının optimizasyonu sağlamaya çalışmıştır. Çalışma sonucunda elde ettiği çözüm sonuçlarını halihazırda uygulanan verilerle karşılaştırmış ve karşılaştırma sonucunda CBS çözümüyle, günlük olarak kat edilen toplam mesafeden % 31, GAMS ile % 33 tasarruf sağladığını belirtmiştir.

Asefi, Lim, Maghrebi ve Shahparvari (2018) yaptıkları çalışmada, katı atıkların toplanması üzerine çalışmışlar ve kat edilen toplam mesafe ile tesis kurulum ücretini de içeren toplam maliyetin minimize edilmesini amaçlamışlardır. Öncelikli olarak problemi Karışık

Tamsayılı Doğrusal Programlama modeli oluşturarak çözmüşler müteakiben Değişken Komşu Arama algoritması ile Değişken Komşu Arama algoritması ve Benzetimli Tavlama algoritmasının birlikte kullanıldığı hibrit bir algoritma önermişlerdir. Önerdikleri iki algoritmanın çözüm sonuçlarını optimal çözümle karşılaştırmışlar ve önerilen iki algoritmanında optimale yakın sonuçlar verdiğini belirtmişlerdir.

Alakaş, Kızıltaş, Eren ve Özcan (2018) yaptıkları çalışmada "Sıfır Atık Projesi" kapsamında Kırıkkale Merkez, Yahşihan ve Bahşili ilçesindeki kamu kuruluşlarından ve okullardan yapılacak atık toplama üzerine çalışmışlar ve toplam maliyeti minimize eden araç rotalarını bulmayı hedeflemişlerdir. İlk olarak çözümü yapılacak 224 kurum binası belirlemişler, daha sonra atık toplanacak noktaları, birbirlerine olan mesafeleri göz önüne alarak 95 noktaya indirmişlerdir. Her bir nokta için talep tahmini yaparak toplanacak atık miktarını belirlemişlerdir. Ardından her nokta için Pareto Analizi yaparak, A, B ve C grubu noktaları belirlemişler. Çalışmada A grubu noktalar her gün gidilmesi gereken noktalar, B grubu noktalar iki günde bir ve C grubu noktalar haftada bir gidilmesi gereken noktalar olarak ele alınmıştır. Bu duruma göre haftalık yolcu planı oluşturmuşlardır. Sonraki aşamada doğrusal programlama modeli oluşturarak problemi çözmüşler ve az maliyetli rotaları bulmuşlardır.

Tirkolaee, Abbasian, Soltani ve Ghaffarian (2019) yaptıkları çalışmada, atıkları toplarken izin verilen zaman pencerelerinden çıkış cezasını da içeren toplam maliyeti en aza indirmeyi amaçlamışlardır. Söz konusu problemin çözümü için matematiksel model oluşturmuşlar ve Benzetimli Tavlama algoritmasını önermişlerdir. Problemi hem matematiksel model hem de önerdikleri algoritmayla çözmüşler ve çözüm sonuçlarını karşılaştırmışlardır. Önerdikleri Benzetimli Tavlama algoritası ile kesin çözüme % 1 yakın sonuçlar bulduklarını, hesaplama zamanı olarak önerdikleri algoritmanın çok daha hızlı olduğunu belirtmişlerdir.

Molina, Eguia ve Racero (2019) yaptıkları çalışmada, atıkları toplama problemini çözerken carbon emisyonu, yakıt tüketimi vb. hususları göz önünde bulunduran matematiksel model önermişlerdir. Problemin çözümü için Değişken Komşu Arama algoritması ve Tabu Arama algoritmasının birlikte kullanıldığı hibrit bir algoritma önermişler ve öncelikle bu algoritmayla elde edilen çözüm sonuçlarını Değişken Komşu Arama ve Tabu Arama algoritmaları ile karşılaştırmışlardır. Önerilen algoritmanın diğer iki algoritmaya göre daha

iyi sonuçlar verdiğini, hesaplama süresi bakımından ise performansının düşük olduğunu belirtmişlerdir. Müteakiben önerdikleri algoritmayı Sevilla (İspanya) şehrinde gerçek bir problemde uygulamışlardır. Uygulama sonucunda önerdikleri algoritmanın kat edilen mesafe, yakıt tüketimi ve karbon emisyon miktarında iyileşme sağladığını belirtmişlerdir.

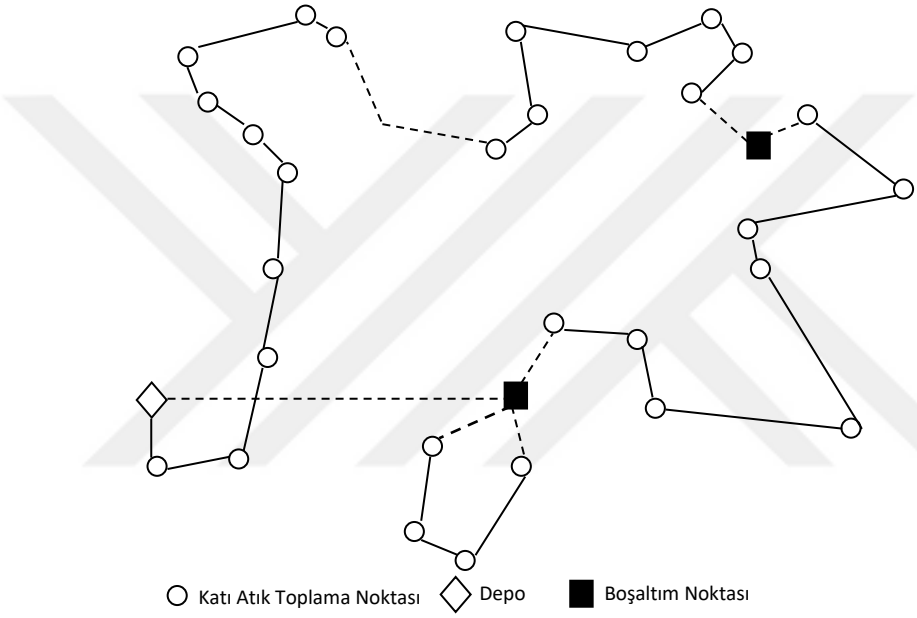
Kızıltaş, Alakaş ve Eren (2020) yaptıkları çalışmada, Sıfır Atık Projesi kapsamında Kırıkkale ili ve sekiz ilçesindeki kamu kurumları ile alışveriş merkezleri ve okullardan yapılacak geri dönüştürülebilir atıkların toplama sürecini ele almış, toplam maliyetin minimize edilmesini amaçlamışlardır. Problemin çözümü için Periyodik Heterojen Çok Araçlı Araç Rotalama Problemi oluşturmuşlar ve üç farklı senaryo altında çözmüşler. Çalışma sonucunda en az maliyetli rotaları belirlemişlerdir.

Qiao, Tao, Wu, Yu ve Zhang (2020) yaptıkları çalışmada, katı atık toplama sistemi üzerine çalışmışlar, araçların sabit maliyeti, yakıt tüketimi, karbon emisyon miktarı ve boşaltım istasyonuna atanan araç sayısının, sistemin işleme kapasitesini aşmasından kaynaklanabilecek ceza maliyetinide göz önünde bulunduran ve bu maliyetleri minimize etmeyi amaçlayan matematiksel model oluşturmuşlardır. Problemin çözümü için Parçacık Sürü Optimizasyonu ve Tabu Arama algoritmasının birlikte kullanıldığı iki aşamalı hibrit bir algoritma önermişlerdir. Bazı problem örneklerinde önerilen algoritma ile Parçacık Sürü Optimizasyonu algoritmasını karşılaştırmışlar ve önerilen algoritmanın çok iyi performans gösterdiğini belirtmişlerdir.

Dereci (2022) yaptığı çalışmada, İstanbul ili Ümraniye ilçesinin katı atık toplama sürecini incelemiş ve oluşan rotaların minimize edilmesini amaçlamıştır. Problemin çözümünde; En Yakın Komşu, Tasarruf, Benzetimli Tavlama, Aç Gözlü İniş, Rehberli Yerel Arama ve Tabu Arama gibi sezgisel ve metasezgisel algoritmaları kullanmıştır. Ayrıca önce kümele sonra rotala prensibine dayanan bir çözümde incelemiştir. Çalışma sonucunda incelenen algoritmaların gerçek hayat problemlerinde hızlı ve olurlu sonuçlar üretebildiğini belirtmiştir.

3.2. Problemin Tanımı

Araç Rotalama Problemi (ARP) ; belirli bir depodan başlayarak yine aynı depoda bitecek şekilde tüm talep noktalarının dolaşıldığı zor optimizasyon modelleridir. Temel amaç kat edilen toplam mesafenin minimize edilmesidir. Bu çalışmanın klasik ARP'den farkı ise katı atık toplayan araçların kapasiteleri dolduğunda yüklerini boşaltım istasyonuna dökmeleri ve ana depoya döndüklerinde boş olmaları gerektiğidir. Söz konusu problem şekil 3.1'deki gibidir.



Şekil 3.1. Katı atık toplama problemi

Bu çalışmada; Ankara ili, Çankaya ilçesinde bulunan çöp konteynerlerindeki katı atıkların toplanması ve boşaltılması işleminin minimum mesafe kat ederek yapılması problemi ele alınmaktadır.

Çalışmada 3850 adet çöp konteyneri ele alınmıştır. Söz konusu çöp konteynerlerinin bulunduğu noktalara ait koordinat verileri saha çalışması, Çankaya Belediyesi Temizlik İşleri Müdürlüğü personelinin yardımı ve Arvento programı kullanılarak elde edilmiştir. Her bir katı atık konteynerinin kapasitesi 500 kg dır. Ancak konteynerler tam olarak dolu olmadığından problemde her bir konteynerde 350 kg katı atık olduğu değerlendirilmiştir.

Çalışmada homojen araç filosu kullanılmıştır. Araçların kapasite kısıtı bulunmaktadır ve kapasitesi 13 m^3 olarak alınmıştır.

Zaman kısıtı olarak da araçlar en fazla 10 saat çalışabilmektedir. Ayrıca araçların her bir katı atık toplama noktasında toplama işlemini 3 dk. da, katı atık boşaltım noktasında ise 30 dk. da boşaltım işlemini gerçekleştirdiği değerlendirilmiştir. Araçların Sincan ve Mamak ilçelerinde katı atık boşaltım işlemini gerçekleştirdiği 2 boşaltım istasyonu bulunmaktadır.

Araçlar ana depodan hareketle belirlenen çöp konteynerlerindeki katı atıkları alacak, kapasitesi dolduğunda 2 adet katı atık boşaltma istasyonundan birisine atıkları boşaltacak ve kaldığı yerden katı atıkları toplamaya devam edecektir. En sonunda araçta mevcut olan atıkları da boşaltım istasyonuna boşalttıktan sonra boş olarak ana depoya dönüş yapacaktır.

3.3. Problemin Varsayımları

1. Düğümler arası mesafe Öklid mesafesi olarak hesaplanmaktadır.
2. Her bir düğüme sadece bir araç hizmet vermekte ve katı atıklar bölünememektedir.
3. Katı atık toplayan araçların rotaları ana depodan başlamakta ve toplama işlemi bitiminde ana depoda sonlanmaktadır.
4. Araç filosu homojendir ve araçlar sabit belli bir hızla hareket etmektedir.
5. Çöp boşaltma tesisinde yeteri kadar işgücü ve teçhizat bulunmaktadır, araçlar tesiste sıra beklememektedir.
6. Çöp boşaltma tesisinde katı atıkların boşaltma işlemi belli bir süre almaktadır.
7. Katı atık toplayan araçların belli bir kapasitesi vardır ve toplam yük bu kapasiteyi aşmamaktadır.
8. Araçların günlük çalışma zamanı vardır ve araçlar bu zamanı aşmamaktadır.
9. Katı atıkların toplandığı düğüm noktalarında konteynerlerin araçlara boşaltılması ve bırakılması belirli bir süre almaktadır.
10. Kullanılan katı atık toplama noktalarının yerleri statiktir.
11. Tüm araçlar ana depoya dönmeden önce çöp boşaltım tesisine uğramakta ve boş olarak ana depoya gitmektedir.

3.4. Problemin Matematiksel Modeli

İndisler:

i , j = Katı Atık Toplama Noktaları

k = Araç Sayısı

Parametreler:

$y_{i,j}$ = i düğümü ile j düğümü arasındaki mesafe

$t_{i,j}$ = i düğümü ile j düğümü arasındaki süre

$çöp_i$ = i noktasındaki çöp miktarı

s_i = i noktasındaki çöp yükleme süresi

vehcap = Araç kapasitesi

N = Düğüm noktası sayısı

M = Boşaltım Noktası sayısı

Karar Değişkeni:

$$x_{i,j,k} = \begin{cases} 1, i \text{ düğümünden } j \text{ düğümüne } k \text{ aracı ile gidilirse} \\ 0, d.d \end{cases}$$

Model:

$$\text{Min } Z = \sum_{k}^{K} \sum_{(i,j)}^{N} x_{i,j,k} \cdot y_{i,j} \tag{3.1}$$

Kısıtlar:

$$\sum_{k}^{K} \sum_{j}^{N} x_{i,j,k} = 1 \qquad \forall i \in 1,2,\ldots N \tag{3.2}$$

$$\sum_{j=1}^{N} x_{"0",j,k} \leq 1 \qquad \forall k \in K \tag{3.3}$$

$$D_{i,k} \leq vehcap \qquad i \in 1,2,\ldots N, \quad k \in K \tag{3.4}$$

$$D_{m,k} = 0 \qquad \forall k \in K, \quad m = 0, N+1, N+2 \ldots N+M \tag{3.5}$$

$$D_{i,k} + çöp_j - D_{j,k} \le (1 - x_{i,j,k}) * Big\ M \qquad i, j \in 0,1,2,...N, N+M \tag{3.6}$$

$$\sum_{j=1}^{N} çöp_{(j)} \sum_{i=1}^{N+M} x_{i,j,k} \le vehcap * \mathrm{N}_k \qquad \forall k \in K \tag{3.7}$$

$$\sum_{i=1}^{N} \sum_{m=N+1}^{M} X_{i,m,k} = N_k \qquad \forall k \in K \tag{3.8}$$

$$\sum_{m=N+1}^{N+M} \sum_{i=1}^{N} X_{m,i,k} = N_k - 1 \qquad \forall k \in K \tag{3.9}$$

$$\sum_{m=N+1}^{N+M} X_{m,"0",k} = 1 \qquad \forall k \in K \tag{3.10}$$

$$\sum_{j=0}^{N+M} X_{i,j,k} = \sum_{j=0}^{N+M} X_{j,h,k} \qquad i, h \in 0,1,...N+M, \quad \forall k \in K \tag{3.11}$$

$$W_{i,k} + S_i + t_{i,j} - W_{j,k} \le (1 - X_{i,j,k}) * \text{Big M} \quad \forall k \in K,\ \mathrm{i} \neq j,\ \mathrm{j}>1 \tag{3.12}$$

$$a_i * \sum_{j=1}^{N+M} X_{i,j,k} \le W_{i,k} \qquad \forall k \in K, \ \forall i \in 1,2,...N+M \tag{3.13}$$

$$b_i \ge W_{i,k} - Big\ M\ * (1 - \sum_{j=1}^{N+M} X_{i,j,k}) \quad \forall k \in K, \quad \forall i \in 1,2,...N+M \tag{3.14}$$

$$E \le W_{i,k} \le L \qquad \forall k \in K, \quad i \in 0,1,...N+M \tag{3.15}$$

$$X_{i,j,k} \in \{0,1\}, \quad \forall k \in K \tag{3.16}$$

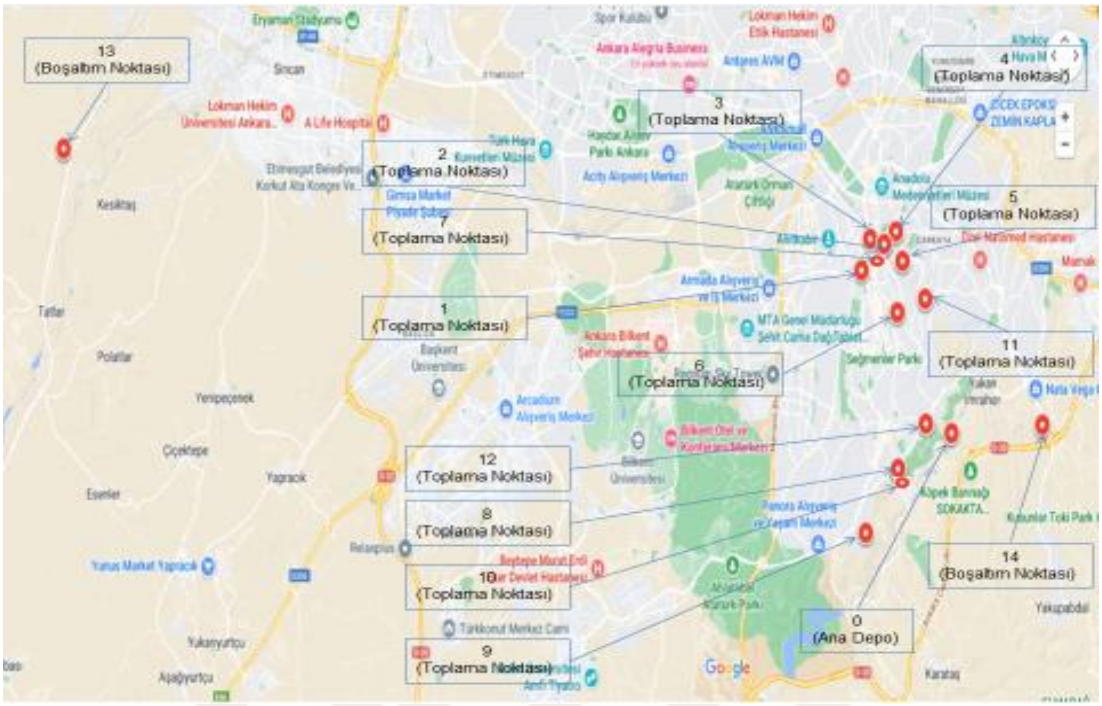
$$N_k \ge 0, \ \text{int}\, eger, \ \forall k \in K \tag{3.17}$$

$$D_{i,k} \ge 0, \ \forall k \in K, \ i \in 0,1,...N+M \tag{3.18}$$

Eş. 3.1'de yer alan amaç fonksiyonu, toplam kat edilen mesafenin minimize edilmesini amaçlamaktadır. Eş. 3.2 her atık toplama noktasına yalnız bir araç tarafından gidilmesini sağlar. Eş. 3.3 her rotanın ana depodan başlaması gerektiğini belirtir. Eş. 3.4 toplanan çöp miktarının araç kapasitesini aşmasını engeller. Eş. 3.5 araçlar depodan ayrıldığında ve çöp boşaltım istasyonunu her ziyaret ettiğinde toplam çöp miktarının sıfıra eşitlenmesini sağlar. Eş. 3.6 bir araç herhangi bir atık toplama noktasını ziyaret ettiğinde toplanan atık miktarını gösterir. Eş. 3.7 her bir aracın çöp boşaltım istasyonuna kaç sefer gideceğini hesaplar. Eş. 3.8 atık toplama noktalarından boşaltım istasyonuna hesaplanan değer kadar gidilmesini Eş. 3.9'da boşaltım istasyonundan atık toplama noktalarına hesaplanan değerin bir eksiği kadar dönülmesini sağlar. Eş. 3.10 boşaltım istasyonundan yapılacak son turun ana depoya olması gerektiğini gösterir. Eş. 3.11 aracın herhangi bir noktaya gittiğinde o noktadan tekrar ayrılmasını sağlar. Eş. 3.12 toplam çalışma süresini hesaplar. Eş. 3.13 ve Eş. 3.14 zaman kısıtlarıdır. Eş. 3.15 toplam çalışma süresinin belirlenen aralıkta olmasını garantiler.

3.5. Problemin Küçük Veri Setinin Matematiksel Model ile Çözümü

Oluşturulan matematiksel model ile katı atıkların toplanması ve boşaltılması sürecinde toplam kat edilen mesafenin minimize edilecek şekilde rotaların oluşturulması amaçlanmaktadır. Araç Rotalama Problemleri NP-zor problem kategorisinde yer aldığından ve veri sayısı fazla olduğundan mevcut paket programlar ile çözülmesi zordur. Bu nedenle oluşturulan matematiksel model küçük veri seti kullanılarak çözülmüştür. Veri setinde kullanılan noktalar Şekil 3.2'de gösterilmiştir.



Şekil 3.2. Veri setinde kullanılan noktalar

Resimde 0 noktası; araçların çöp toplamaya başladığı ve bitişte boş olarak dönmeleri gereken ana depoyu, 1-12 arasındaki noktalar; çöp konteynerlerinin bulunduğu katı atık toplama noktalarını, 13-14 numaralı noktalar ise çöp boşaltım istasyonlarının yerlerini ifade etmektedir. Veri setinde kullanılan noktaların koordinatları Çizelge 3.1'de verilmiştir.

Çizelge 3.1. Veri setinde kullanılan noktaların koordinatları

Noktalar	X Koordinatı	Y Koordinatı
0 noktası (Ana Depo)	490993	4414048
1'nci nokta (Atık Toplama Noktası)	487384	4418574
2'nci nokta (Atık Toplama Noktası)	488287	4419300
3'üncü nokta (Atık Toplama Noktası)	487739	4419460
4'üncü nokta (Atık Toplama Noktası)	488750	4419665
5'inci nokta (Atık Toplama Noktası)	489016	4418832
6'ıncı nokta (Atık Toplama Noktası)	488808	4417402
7'inci nokta (Atık Toplama Noktası)	488288	4419293
8'inci nokta (Atık Toplama Noktası)	488839	4413052
9'uncu nokta (Atık Toplama Noktası)	487532	4411257
10'uncu nokta (Atık Toplama Noktası)	488840	4413058
11'inci nokta (Atık Toplama Noktası)	489923	4417772
12'nci nokta (Atık Toplama Noktası)	489950	4414297
13'üncü nokta (Boşaltım noktası)	455448	4422090
14'üncü nokta (Boşaltım Noktası)	494631	4414263

Matematiksel modelin çözümünde İntel (R) Core (TM) İ5-3427U HQ CPU@ 1.80 GHz işlemcili 4 GB Ram bellek özellikli dizüstü bilgisayar ve GAMS IDE 37 uygulaması kullanılmıştır. 3 numaralı problemin (7 atık toplama noktası, 2 boşaltım noktası ve 1 araç) GAMS programı kullanılarak elde edilen çözüm sonucu Şekil 3.3'te sunulmuştur.

```
Proven optimal solution
MIP Solution:        83302.436286    (179927 iterations, 19398 nodes)
Final Solve:         83302.436286    (0 iterations)

Best possible:       83302.436286
Absolute gap:            0.000000
Relative gap:            0.000000

--- Reading solution for model WasteManagement
--- Executing after solve: elapsed 0:00:07.975
```

Şekil 3.3. GAMS sonuç ekranı

Elde edilen optimal sonuç değeri 83 302,4 m, çözüm süresi 7,97 sn'dir. Söz konusu problemde oluşan araç rotası;

0 - 6 - 1- 8 (Boşaltım Noktası) - 3 - 7 - 2 - 4 - 5 - 9 (Boşaltım Noktası) - 0 dır.

Matematiksel model kullanılarak elde edilen tüm sonuçlar Çizelge 3.2'de gösterilmiştir. Problemin çözüm sonuçları incelendiğinde; 4 numaralı çözümde (8 atık toplama noktası, 2 boşaltım noktası ve 1 araç) optimal çözüme ulaşılma zamanı 210,66 sn, 6 numaralı çözümde (9 atık toplama noktası, 2 boşaltım noktası ve 1araç) optimal çözüme ulaşma zamanı 2 264,26 sn iken 8 numaralı çözümde (10 atık toplama noktası, 2 boşaltım noktası ve 1 araç) optimal çözüme ulaşılma zamanı 23 726,25 sn olarak gerçekleşmiştir.

Boşaltım noktası sayısı (düğüm noktası sayısı) artırıldığında arama uzayının büyümesine sebep olmakta bu durum da GAMS paket programında çözüm süresinin artmasına sebep olmaktadır. Gerçek hayat problemlerinde kullanılan düğüm sayısı çok daha fazla olduğundan optimal yada geçerli bir çözümün bulunması çok daha fazla zaman almakta yada bir çözüm bulunamamaktadır. Bu yüzden problemin çözümü için kısa sürede çözüm veren Tavlamalı Sınırlandırılmış Demon Algoritmasının kullanıldığı sezgisel bir yöntem önerilmiştir.

Çizelge 3.2. Matematiksel modelin çözümünden elde edilen sonuçlar

S.No	Toplama Noktası Sayısı	Boşaltım Noktası Sayısı	Araç Sayısı	Çözüm Sonucu Oluşan Rotalar	Amaç Fonksiyonu Değeri (m)	Çözüm Süresi (sn)
1.	5	1	1	0-5-4-2-3-1-6-0	76 738	0,13
2.	6	2	1	0-6-1-7-3-2-4-5-8-0	83 292	3,48
3.	7	2	1	0-6-1-8-3-7-2-4-5-9-0	83 302	7,97
4.	8	2	1	0-8-6-1-9-3-7-2-4-5-10-0	86 022	210,66
5.	8	1	2	0-1-3-7-2-4-9-0 0-8-6-5-9-0	38 595	4,31
6.	9	2	1	0-8-9-6-1-10-3-7-2-4-5-11-0	90 169	2 264,26
7.	9	1	2	0-1-3-7-2-4-10-0 0-8-9-6-5-10-0	42 741	9,05
8.	10	2	1	0-10-8-9-6-1-11-3-7-2-4-5-12-0	90 171	23 726,25
9.	10	1	2	0-10-8-9-6-5-11-0 0-1-3-7-2-4-11-0	42 744	22,14
10.	11	1	3	0-6-5-11-12-0 0-1-3-7-2-4-12-0 0-10-8-9-12-0	51 852	9.92
11.	12	1	3	0-1-3-7-2-4-13-0 0-12-10-8-9-13-0 0-6-5-11-13-0	52 218	112,62

4. SEZGİSEL YÖNTEM

Bu bölümde, Yurtdaş (2018) tarafından çapraz sevkiyatta araç rotalama problemi için geliştirilen sezgisel yöntem açıklanmıştır. Geliştirilen algoritmanın çözümleri ile matematiksel modelin GAMS kullanılarak çözülebilen küçük düğüm sayılı çözümleri karşılaştırılmış, müteakiben geliştirilen algoritma ile çok büyük boyutlu çözümler yapılmıştır.

4.1. Geliştirilen Algoritmanın Adımları

Problemin çözümünde Tavlamalı Sınırlandırılmış Demon algoritması temel alınmıştır. Algoritmada demon sınır değeri ve tavlama programı ısı düşürme sabiti girdi olarak kullanılmıştır. Isı düşürme sabiti de bir olarak alınmıştır. Algoritmanın adımları Çizelge 4.1'de gösterilmiştir.

Çizelge 4.1. Geliştirilen algoritmanın adımları

1'inci adım	:	Demon değerini, demon sınır olarak ata $D = D_{sınır} > 0$
2'nci adım	:	Başlangıç çözümünü bul $S = S_0$
3'üncü adım	:	Sonlandırma koşulları sağlanana kadar tekrarla
4'üncü adım	:	Yerel arama operatörlerinin uygulanacağı düğüm çiftini bul (x,y)
5'inci adım	:	Minimum operatör maliyet farkını bul $\Delta c = \min \{ \Delta c_{takas}(x,y), \Delta c_{taşı}(x,y), \Delta c_{2opt}(x,y) \}$
6'ncı adım	:	Eğer $\Delta c < D$ ise
7'nci adım	:	Minimum maliyet farkını sağlayan operatörü uygula ve yeni çözümü bul $S_1 = S + \Delta c$
8'inci adım	:	Eğer uygun çözüm oluşur ise
9'uncu adım	:	Yeni çözümü kabul et $S = S^1$
10'uncu adım	:	Demon değerini güncelle $D = D - \Delta c$
11'inci adım	:	Eğer $D > D_{sınır}$ ise $D = D_{sınır}$
12'nci adım	:	Eğer uygun çözüm oluşmaz bir önceki çözüme geri dön
13'üncü adım	:	Demon sınır değerini tavlama programına göre düşür
14'üncü adım	:	Bitir

Algoritmanın birinci adımında sıfırdan büyük bir sayı olarak belirlenen demon değeri, demon sınır değeri olarak atanır. İkinci adımda başlangıç çözümü bulunur, başlangıç çözümüne göre toplam kat edilen mesafe hesaplanır. Üçüncü adım ana döngünün başladığı

ve sonlandırma koşulu sağlanıncaya kadar tekrarlanan bölümdür. Çalışmada döngü iterasyon sayısı kadar tekrarlanmaktadır. Dördüncü ve beşinci adım, algoritmada kullanılan yerel arama operatörlerinin kullanılacağı düğümlerin ve bu düğümlere göre minimum maliyetli operatörün belirlendiği adımlardır. Çalışmada takas operatörü (swap, 1-1 exchange), taşı operatörü (relocate, 1-0 exchange) ve 2-opt operatörü olmak üzere üç farklı yerel arama operatörü kullanılmıştır. Maliyet farkı matematiksel olarak hesaplanırken düğümlerin aynı rotada olup olmadığı kontrol edilmektedir. Eğer düğümler farklı rotalarda ise araçların kapasite kısıtları göz önünde bulundurularak hesaplama yapılır, kapasite kısıtı sağlanmıyorsa maliyet farkı çok büyük bir sayıya eşitlenerek operatör uygulanmaz. Altıncı adımda, hesaplanan maliyet farkının demon değerinden küçük olup olmadığı kontrol edilir. Yedinci adımda yeni çözüm bulunur ve uygun çözüm oluşması durumunda sırasıyla dokuz, on ve on birinci adımlarda, yeni çözüm kabul edilerek demon değeri güncellenir. Demon değerinin demon sınır değerinden büyük olması durumunda demon değeri sınırlandırılarak demon sınır değerine eşitlenir. On ikinci adımda, eğer uygun çözüm oluşmaz ise bir önceki uygun çözüme dönülür. Uygun çözümün oluşup oluşmamasında zaman penceresi kısıtı da göz önünde bulundurulur. On üçüncü adımda demon sınır değeri doğrusal olarak düşürülür.

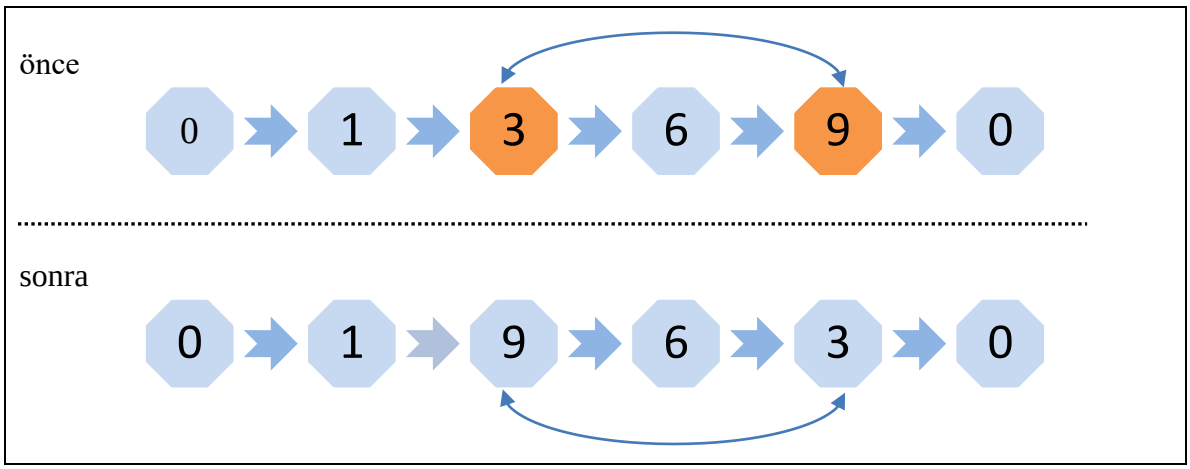
4.2. Algoritmada Kullanılan Yerel Arama Operatörleri

Algoritmada takas operatörü, taşı operatörü ve 2-opt operatörü olmak üzere üç yerel arama operatörü kullanılmıştır. Her bir operatörün uygulanmadan önceki ve uygulandıktan sonraki rota durumları gösterilmiştir.

4.2.1. Takas Operatörü Uygulanışı

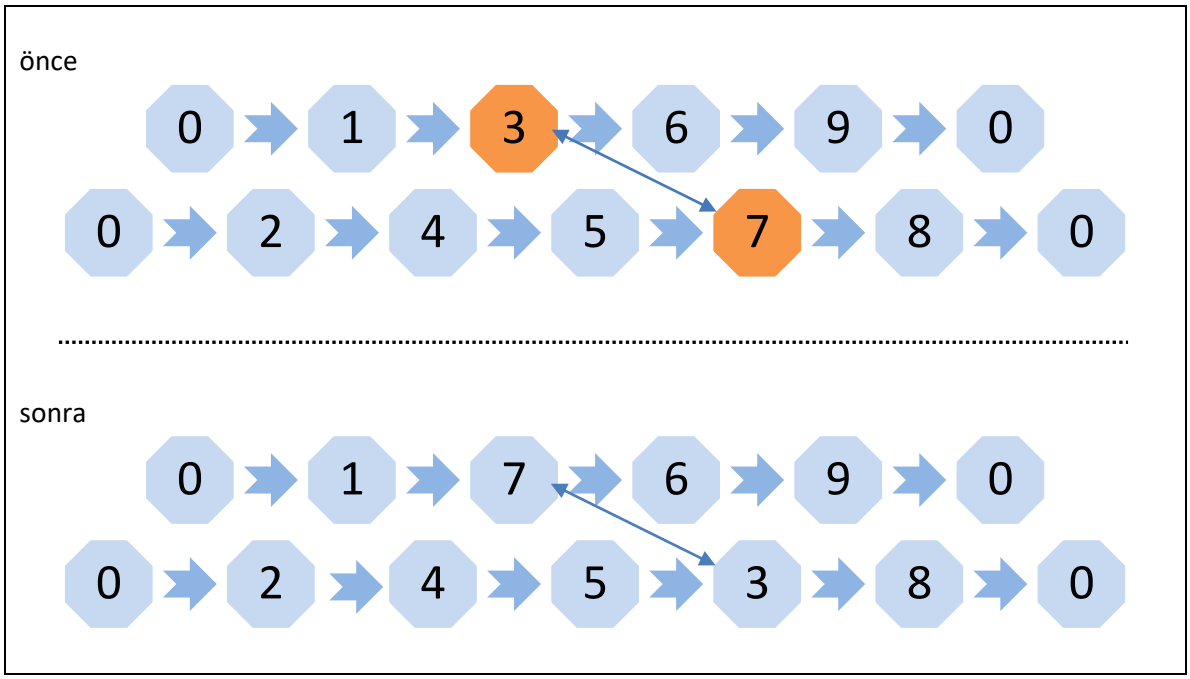
Operatör uygulanırken öncelikli olarak iki farklı düğüm çifti belirlenir ve belirlenen düğüm çiftine göre maliyet farkı hesaplanır. Takas operatöründe belirlenen iki düğüm karşılıklı olarak yer değiştirmektedir. Örnekte 3 ve 9 numaralı düğümler operatörün uygulanacağı düğüm çifti olarak belirlenmiştir. Takas operatörü uygulanmadan önceki ve uygulandıktan sonraki rota durumu Çizelge 4.2’de gösterilmiştir.

Çizelge 4.2. Takas operatörünün uygulanışı (Rota içi)



Operatörün uygulanışı belirlenen düğüm çiftinin aynı veya farklı rotada olmasına göre farklılık göstermektedir. Düğüm çiftinin farklı rotalarda olması durumunda araçların kapasite kısıtının da göz önünde bulundurulması gerekmektedir. Eğer kapasite kısıtı sağlanmazsa operatör uygulanmaz. Örnekte 3 ve 7 numaralı düğümler operatörün uygulanacağı farklı rotalardaki düğüm çifti olarak belirlenmiştir. Takas operatörü uygulanmadan önceki ve uygulandıktan sonraki rota durumu Çizelge 4.3'te gösterilmiştir.

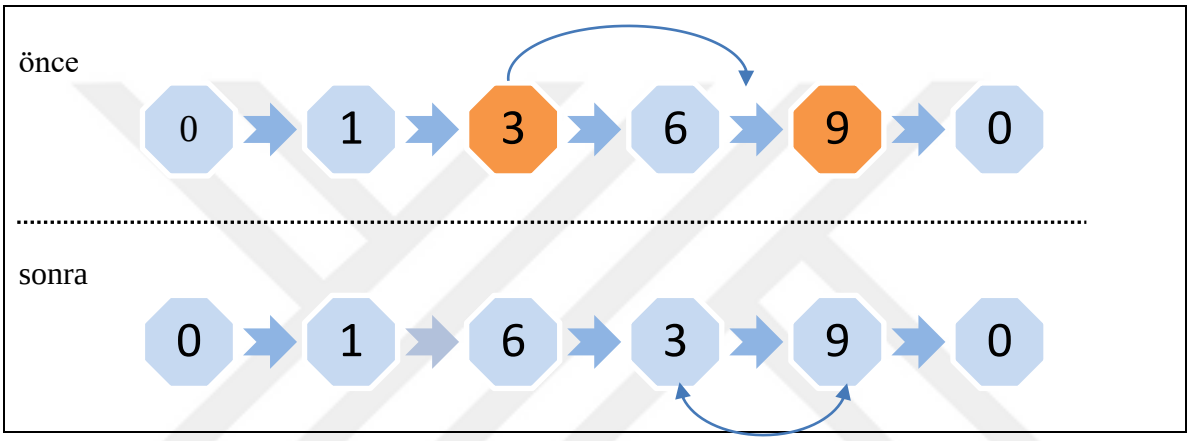
Çizelge 4.3. Takas operatörünün uygulanışı (Rotalar arası)



4.2.2. Taşı Operatörü Uygulanışı

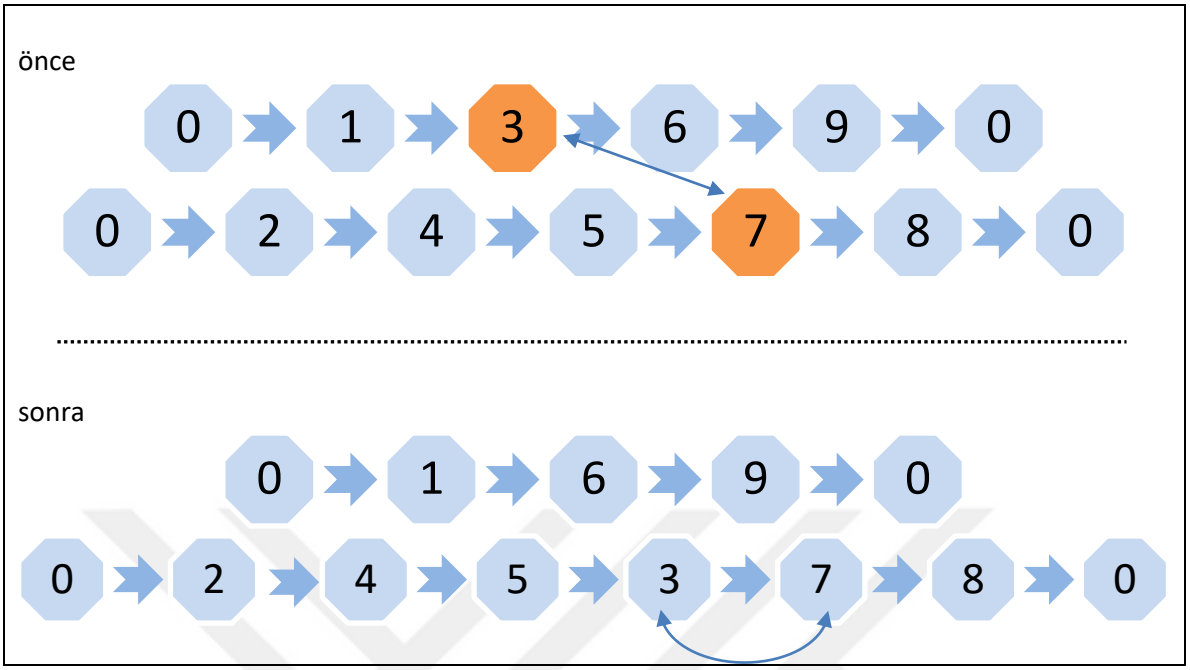
Operatör uygulanırken öncelikli olarak iki farklı düğüm çifti belirlenir ve belirlenen düğüm çiftine göre maliyet farkı hesaplanır. Taşı operatöründe belirlenen iki düğümden birisi diğer düğümün önüne alınmaktadır. Örnekte 3 ve 9 numaralı düğümler operatörün uygulanacağı düğüm çifti olarak belirlenmiştir. Taşı operatörü uygulanmadan önceki ve uygulandıktan sonraki rota durumu Çizelge 4.4'te gösterilmiştir.

Çizelge 4.4. Taşı operatörünün uygulanışı (Rota İçi)



Operatörün uygulanışı belirlenen düğüm çiftinin aynı veya farklı rotada olmasına göre farklılık göstermektedir. Düğüm çiftinin farklı rotalarda olması durumunda araçların kapasite kısıtının da göz önünde bulundurulması gerekmektedir. Eğer kapasite kısıtı sağlanmazsa operatör uygulanmaz. Örnekte 3 ve 7 numaralı düğümler operatörün uygulanacağı farklı rotalardaki düğüm çifti olarak belirlenmiştir. Taşı operatörü uygulanmadan önceki ve uygulandıktan sonraki rota durumu Çizelge 4.5'te gösterilmiştir

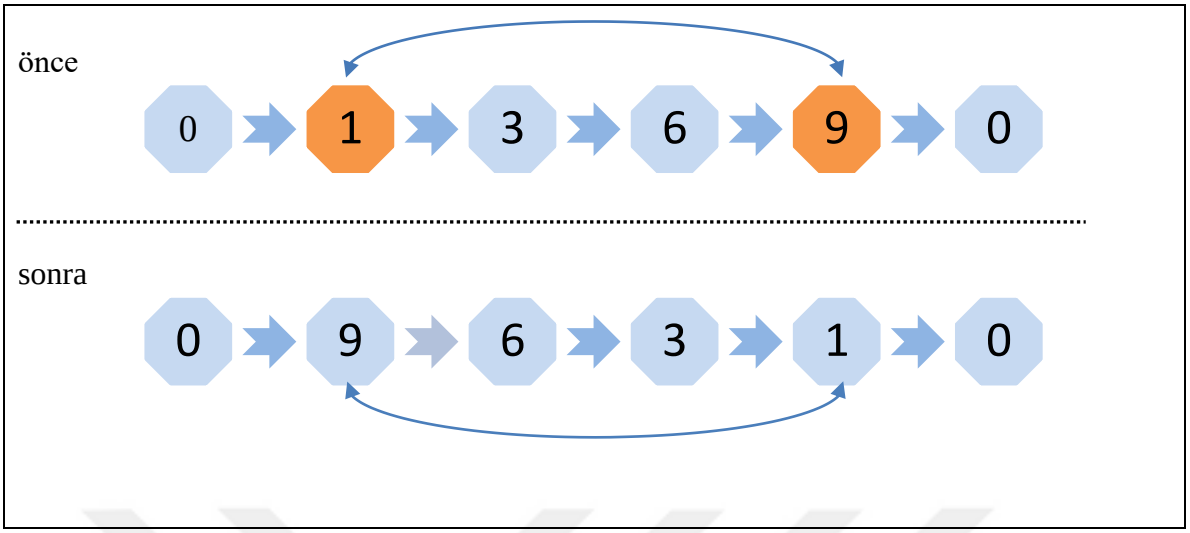
Çizelge 4.5. Taşı operatörünün uygulanışı (Rotalar arası)



4.2.3. 2-Opt Operatörü Uygulanışı

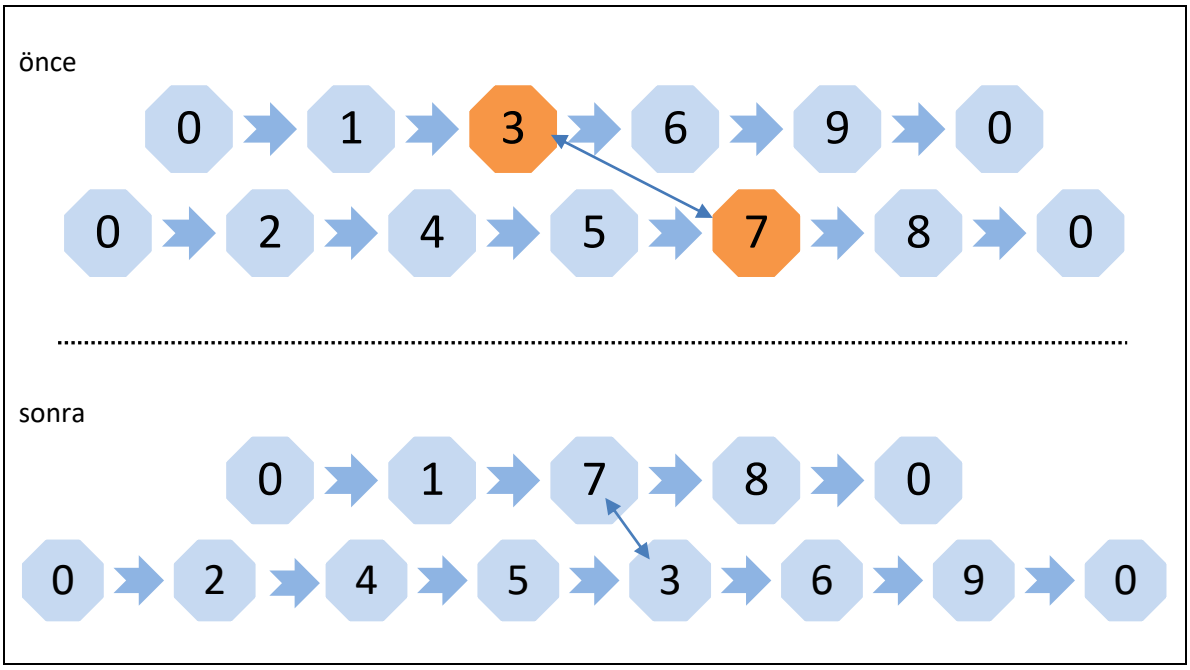
Operatör uygulanırken öncelikli olarak iki farklı düğüm çifti belirlenir ve belirlenen düğüm çiftine göre maliyet farkı hesaplanır. 2-Opt operatöründe belirlenen düğüm çifti yer değiştirmekte, belirlenen düğüm çiftinin arasında kalan düğümler ise tersine dizilmektedir. Örnekte 1 ve 9 numaralı düğümler operatörün uygulanacağı düğüm çifti olarak belirlenmiştir. 2-Opt operatörü uygulanmadan önceki ve uygulandıktan sonraki rota durumu Çizelge 4.6'da gösterilmiştir.

Çizelge 4.6. 2-Opt operatörünün uygulanışı (Rota içi)



Operatörün uygulanışı belirlenen düğüm çiftinin aynı veya farklı rotada olmasına göre farklılık göstermektedir. Düğüm çiftinin farklı rotalarda olması durumunda belirlenen düğüm çifti ve sonraki düğümler karşılıklı olarak yer değiştirmektedir. Ayrıca araçların kapasite kısıtının da göz önünde bulundurulması gerekmektedir. Eğer kapasite kısıtı sağlanmazsa operatör uygulanmaz. Örnekte 3 ve 7 numaralı düğümler operatörün uygulanacağı farklı rotalardaki düğüm çifti olarak belirlenmiştir. 2-Opt operatörü uygulanmadan önceki ve uygulandıktan sonraki rota durumu Çizelge 4.7'de gösterilmiştir.

Çizelge 4.7. 2-Opt operatörünün uygulanışı (Rotalar arası)



4.3. Problemin Küçük Veri Setinin Geliştirilen Sezgisel Yöntem ile Çözümü

Katı atık toplama rota optimizasyonu probleminin sezgisel çözümünde kullanılan Tavlamalı Sınırlandırılmış Demon algoritması C^{++} programlama dili kullanılarak kodlanmıştır.

Kodlanan Tavlamalı Sınırlandırılmış Demon algoritması, katı atık toplama rota optimizasyonu problemi için oluşturulan matematiksel model test edilirken kullanılan aynı parametre değerleri ve aynı veri seti kullanılarak denenmiştir. 3 numaralı problemin (7 atık toplama noktası, 2 boşaltım noktası ve 1 araç) önerilen sezgisel algoritma kullanılarak elde edilen çözüm sonucu Şekil 4.1'de sunulmuştur.

```
Elapsed(ms)=1083
_SOLUTION_
Seed:49823839
Iterasyon:3000000
Demon Bound:40000
_ROUTES_
ROUTE[Depot┬┐P6┬┐P1┬┐Disposal1┬┐P3┬┐P7┬┐P2┬┐P4┬┐P5┬┐Disposal2┬┐Depot][Free Capacity: 1500kg
Distance: 83302.4m Travel Time: 83min Load Time: 105min Unload Time: 60min Total Time: 248min]
TotalDistance:83302.4
```

Şekil 4.1. Sezgisel algoritma sonuç ekranı

Çözüm sonucu incelendiğinde optimal sonuç değeri 83 302,4 m, çözüm süresi 1,083 sn'dir. Söz konusu problemde oluşan araç rotası;

0 - 6 – 1 - Boşaltım Noktası - 3 - 7 - 2 - 4 – 5 - Boşaltım Noktası - 0 dır.

Bu sonuç, GAMS programı kullanılarak elde edilen optimal sonucun aynısıdır. Matematiksel model test edilirken kullanılan diğer problem örneklerinin sezgisel algoritma kullanılarak elde edilen çözüm sonuçları Çizelge 4.8'de gösterilmiştir. Test problemlerinin matematiksel model kullanılarak elde edilen çözüm sonuçları ile Tavlamalı Sınırlandırılmış Demon algoritması kullanılarak elde edilen çözüm sonuçları toplam kat edilen mesafe ve sn olarak çözümlere ulaşma süresi (CPU) bakımından mukayese edildiğinde, sezgisel algoritma ile çözümlere daha kısa sürede ulaşıldığı görülmüştür.

Çizelge 4.8. Sezgisel algoritma kullanılarak elde edilen çözüm sonuçları

S.No	**Toplama Noktası Sayısı**	**Boşaltım Noktası Sayısı**	**Araç Sayısı**	**Çözüm Sonucu Oluşan Rotalar**	**GAMS Çözümü Sonucu (m)**	**Sezgisel Yöntem Çözüm Sonucu (m)**	**Çözüm Süresi (sn)**
1.	5	1	1	0-5-4-2-3-1-Disposal_1-0	76 738	76 738	1,281
2.	6	2	1	0-6-1-Disposal_1-3-2-4-5-Disposal_2-0	83 292	83 292	1,671
3.	7	2	1	0-6-1-Disposal_1-3-7-2-4-5-Disposal_2-0	83 302	83 302	1,925
4.	8	2	1	0-8-6-1-Disposal_1-3-7-2-4-5-Disposal_2-0	86 022	86 022	1,904
5.	8	1	2	0-1-3-7-2-4-Disposal_2-0 0-8-6-5-Disposal_2-0	38 595	38 595	1,679
6.	9	2	1	0-8-9-6-1-Disposal_1-3-7-2-4-5-Disposal_2-0	90 169	90 169	2,127
7.	9	1	2	0-1-3-7-2-4-Disposal_2-0 0-8-9-6-5-Disposal_2-0	42 741	42 741	1,764
8.	10	2	1	0-10-8-9-6-1-Disposal_1-3-7-2-4-5- Disposal_2-0	90 171	90 171	2,248
9.	10	1	2	0-10-8-9-6-5-Disposal_2-0 0-1-3-7-2-4-Disposal_2-0	42 744	42 744	1,957
10.	11	1	3	0-6-5-11-Disposal_2-0 0-1-3-7-2-4-Disposal_2-0 0-10-8-9-Disposal_2-0	51 852	51 852	1,748
11.	12	1	3	0-1-3-7-2-4-Disposal_2-0 0-12-10-8-9-Disposal_2-0 0-6-5-11-Disposal_2-0	52 218	52 218	1,803

4.4. Problemin Geliştirilen Sezgisel Yöntem ile Çözümü

Katı atık toplama rota optimizasyonu problemi için geliştirilen sezgisel algoritma C^{++} programlama dili kullanılarak kodlanmış ve 3.850 atık toplama noktası ve iki boşaltım istasyonundan oluşan problem geliştirilen algoritma kullanılarak çözülmüştür.

Söz konusu problem çözülürken intel (R) core (TM) i7-8750 H CPU@ 2,20 GHz işlemcili 16 GB Ram bellek özellikli Windows 11 işletim sistemli bilgisayar kullanılmıştır.

Problem geliştirilen algoritma ile farklı iterasyon sayılarında, 4 farklı demon sınır değeri için ($D_{sınır}$ = 40, $D_{sınır}$ = 60, $D_{sınır}$ = 80, $D_{sınır}$ = 100) ve farklı rasgele sayı değerleri kullanılarak çözülmüştür. Çözümlerde iki parametre sabit tutulmuş, $D_{sınır}$ değeri değiştirilmiş ve 640 farklı çözüm elde edilmiştir. Elde edilen tüm çözüm sonuçları Ek-1'de sunulmuştur.

Farklı iterasyon sayısı ve farklı $D_{sınır}$ değerleri için çözülen problemin, her bir iterasyon ve $D_{sınır}$ değeri için elde edilen en iyi ve ortalama çözüm sonuçları Çizelge 4.9'da gösterilmiştir.

Çizelge 4.9. Farklı iterasyon ve Demon değerleri için ortalama ve en iyi çözüm sonuçları

Sıra Nu.	Katı Atık Toplama Noktası Sayısı	İterasyon Sayısı	Demon Değeri	Ortalama		Bulunan En İyi	
				Değer (km)	Çözüm Süresi (sn)	Değer (km)	Çözüm Süresi (sn)
1	3 850	3 000 000	40	5 051,380	8,689	4 893,986	8,997
2	3 850	3 000 000	60	5 546,191	8,511	5 383,990	8,463
3	3 850	3 000 000	80	5 998,873	8,130	5 802,664	8,057
4	3 850	3 000 000	100	6 287,468	8,662	6 061,869	11,278
5	3 850	6 000 000	40	4 317,589	19,071	4 175,680	19,487
6	3 850	6 000 000	60	4 715,435	17,682	4 611,859	17,579
7	3 850	6 000 000	80	5 030,017	19,116	4 842,886	23,608
8	3 850	6 000 000	100	5 289,597	16,511	5 153,519	16,324
9	3 850	12 000 000	40	3 788,268	37,039	3 692,844	36,748
10	3 850	12 000 000	60	4 079,181	37,870	3 958,501	50,798
11	3 850	12 000 000	80	4 314,106	38,426	4 198,191	37,797
12	3 850	12 000 000	100	4 541,608	38,031	4 420,059	37,280
13	3 850	24 000 000	40	3 379,809	78,860	3 311,211	73,790
14	3 850	24 000 000	60	3 594,298	74,755	3 506,618	73,405
15	3 850	24 000 000	80	3 784,720	78,442	3 679,744	71,923
16	3 850	24 000 000	100	3 939,155	76,825	3 870,180	81,870

En iyi ve ortalama çözüm sonuçları incelendiğinde;

- Her iterasyon sayısı için farklı Demon sınır değerleriyle yapılan çözümlerde, Demon sınır değeri küçüldükçe daha iyi sonuçlar elde edildiği,
- İterasyon sayısı arttıkça çözüm süresinin arttığı, aynı zamanda çözüm sonucunun iyileştiği,
- Elde edilen en iyi çözümlerin ortalama çözümden iyi olduğu görülmektedir.

Bu kapsamda; $D_{sınır} = 40$ alınarak ve iterasyon sayısı artırılarak çözümler yapılmış, belirli iterasyon sayısından sonra çözüm sonuçlarının birbirine yakınsadığı belirlenmiştir. Yüksek iterasyon sayılı en iyi çözüm sonuçları Çizelge 4.10'da gösterilmiştir.

Çizelge 4.10. Yüksek iterasyon sayılı en iyi çözüm sonuçları

Sıra Nu.	Katı Atık Toplama Noktası Sayısı	İterasyon Sayısı	Demon Değeri	Bulunan Değer (km)	Çözüm Süresi (sn)
1	3850	150 000 000	40	2 731,109	507,785
2	3850	160 000 000	40	2 690,199	528,899
3	3850	170 000 000	40	2 685,720	566,622
4	3850	180 000 000	40	2 677,014	616,670
5	3850	190 000 000	40	2 664,735	627,937
6	3850	200 000 000	40	2 639,897	673,631

5. SONUÇ VE ÖNERİLER

Bu çalışmada Ankara ili Çankaya ilçesinde bulunan bir kısım çöp konteynerinin yeri belirlenerek, söz konusu bölgede icra edilen katı atık toplama sürecinde toplam mesafenin minimize edilmesi amaçlanmıştır.

Toplam kat edilen mesafeyi minimize etmek maksadıyla bir matematiksel model önerilmiş, problemin büyüklüğünden ötürü sezgisel yöntemlerden demon algoritmasını esas alan bir çözüm yöntemi geliştirilmiştir. Geliştirilen algoritmanın çözümleri ile önerilen matematiksel modelin GAMS kullanılarak çözülebilen küçük düğüm sayılı çözümleri karşılaştırılmış, müteakiben geliştirilen algoritma ile çok büyük boyutlu çözümler yapılmıştır.

GAMS programı kullanılarak 2 264,26 sn'de çözülebilen, dokuz atık toplama noktalı, tek araçlı problem, geliştirilen sezgisel algoritma kullanılarak 2,127 sn gibi çok kısa bir sürede çözülmüş ve en iyi çözüme ulaşılmıştır. Bu da GAMS programı kullanılarak çözülebilen küçük problem örneklerindeki en iyi çözüm sonuçlarına geliştirilen sezgisel algoritma ile çok kısa sürede ulaşılabildiğini göstermiştir.

Bu kapsamda; geliştirilen sezgisel algoritmanın kesinlik, hız ve basitlik kriterlerini sağladığı, literatürde bulunan diğer Araç Rotalama Problemlerinin çözümüne uyarlanabileceği değerlendirilmektedir.

Bu çalışmada, iki boşaltım noktalı ve homojen araç filolu katı atık toplama optimizasyon problemi ele alınmıştır. Problemde kullanılan araçların hızlarının ve yolun eğiminin sabit olduğu varsayılmıştır. İlerleyen çalışmalarda heterojen araç filolu, araç hızı ve yol eğimi gibi kısıtları içeren algoritmalar çalışılabilir.

KAYNAKLAR

Alakaş, M., Kızıltaş, Ş., Eren, T., & Özcan, E. (2018). Sıfır Atık Projesi Kapsamında Atıkların Toplanması: Kırıkkale İlinde Homojen Çok Araçlı Araç Rotalama Uygulaması. *Harran Üniversitesi Mühendislik Dergisi*, 190-196.

Asefi, H., Lim, S., Maghrebi, M., & Shahparvari, S. (2018). Mathematical Modelling and Heuristic Approaches to the Location-Routing Problem of a Cost-Effective Integrated Solid Waste Management. *Annals of Operation Research*, 1-38.

Assaf, R., & Saleh, Y. (2017). Vehicle-Routing Optimization for Municipal Solid Waste Collection using Genetic Algorithm: The Case of Southern Nablus City. *Civil and Environmental Engineering Reports*, 43-57.

Bell, J. E., & McMullen, P. R. (2004). Ant Colony Optimization Techniques for The Vehicle Routing Problem. *Advanced Engineering Informatic* , 41-48.

Benjamin, A., & Beasley, J. (2010). Metaheuristics for the Waste Collection Vehicle Routing Problem with Time Windows, Driver Rest Period and Multiple Disposal Facilities. *Computers &Operations Research*, 2270-2280.

Blum, C., & Roli, A. (2003). Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison. *ACM Computing Surveys*, 268-308.

Breedam, A. (2001). Comparing Descent Heuristics and Metaheuristics for the Vehicle. *Computers & Operations Research*, 289-315.

Breedam, A. V. (2001). Comparing Descent Heuristics and Metaheuristics for The Vehicle Routing Problem. *Computers and Operations Research*, 289-315.

Buhrkal, K., Larsen, A., & Ropke, S. (2012). The waste collection vehicle routing problem with time windows in a city logistics context. *Procedia - Social and Behavioral Sciences*, 241-254.

Caccetta, L., & Hill, S. (2001). Branch and Cut Methods for Network Optimization. *Mathematical and Computer Modelling*, 517-532.

Chandran, B., Golden, B., & Wasil, E. (2003). A Computational Study of Three Demon Algorithm Variants for Solving the Traveling Salesman Problem. H. K. Bhargava, & N. Ye içinde, *Computational Modeling and Problem Solving in the Networked World* (s. 156-174). Boston/London: Kluwer Academic Publishers.

Christofides, N., Mingozzi, A., & Toth, P. (1981). Exact Algorithms for The Vehicle Routing Problem, Based On Spanning Tree and Shortest Path Relaxations. *Mathematical Programming*, 255-282.

Dantzig, G., & Ramser, J. (1959). The Truck Dispatching Problem. *Management Science*, 80-91.

Dereci, U. (2022). *Katı Atık Toplama Rota Optimizasyonu: Ümraniye İlçesi Örneği.* Yüksek Lisans Tezi, İstanbul Medeniyet Üniversitesi Lisansüstü Eğitim Enstitüsü: İstanbul.

Dişkaya, F. (2018). *Yeşil Lojistik Yönetiminde Ulusal Karayolu Yük Taşımacılığı Araç Rotalama Optimizasyonu.* Doktora Tezi, Marmara Üniversitesi Sosyal Bilimler Enstitüsü: İstanbul.

Ekizler, H. (2011). *Araç Rotalama Probleminin Çözümünde Karınca Kolonisi Optimizasyonu Algoritmasının Kullanılması.* Yüksek Lisans Tezi, İstanbul Üniversitesi Fen Bilimleri Entitüsü: İstanbul.

Elgarej, M., Mansourı, K., Youssfı, M., Benmoussa, N., & El Fazazı, H. (2017). Distributed Swarm Optimization Modeling for Waste Collection Vehicle Routing Problem. *International Journal of Advanced Computer Science and Applications*, 306-312.

Erel, R. (1995). *Taşıt Rotalaması ve Çizelgelemesi:Otobüsle Kentlerarası Yolcu Taşımacılığı için Bir Model.* Doktora Tezi, Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü: İstanbul.

Erfani, S. M., Danesh, S., Karrabi, S. M., & Shad, R. (2017). A Novel Approach to Find and Optimize Bin Locations and Collection Routes using a Geographic Information System. *Waste Management & Research*, 1-10.

Ergülen, A., & Güngör, İ. (2006). Bulanık Araç Rotalama Problemlerine bir Model Önerisi ve bir Uygulama. *Celal Bayar Üniversitesi Yönetim ve Ekonomi Dergisi*, Cilt:13, Sayı:1.

Erol, V. (2006). *Araç Rotalama Problemleri için Populasyon ve Komşuluk Tabanlı Metasezgisel bir Algoritmanın Tasarımı ve Uygulaması.* Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü: İstanbul.

Eryavuz, M., & Gencer, C. (2001). Araç Rotalama Problemine ait Bir Uygulama. *Süleyman Demirel Üniversitesi İİBF Dergisi*, 139-155.

Glover, F. (1989). Tabu Search - Part 1. *ORSA Journal on Computing*, 190-206.

Goldberg, D. (1989). *Genetic Algorithms in Search Optimization and Machine Learning.* USA: Addison Wesley Publishing Company.

Hannan, M., Akhtar, M., Begum, R., Basri, H., Hussain, A., & Scavino, E. (2017). Capacitated Vehicle-Routing Problem Model for Scheduled Solid Waste Collection and Route Optimization Using PSO Algorithm. *Waste Management*, 1-11.

Islam, R., & Rahman, M. S. (2012). An Ant Colony Optimization Algorithm for Waste Collection Vehicle Routing with Time Windows, Driver Rest Period and Multiple Disposal Facilities. *IEEE/OSA/IAPR International Conference on Informatics, Electronics & Vision*, 774-779.

Kirkpatrick, S., Gelatt, C., & Vecchi, M. (1983). Optimization by Simulated Annealing. *Science*, 671-680.

Kızıltaş, Ş., Alakaş, H. M., & Eren, T. (2020). Collection of Recyclable Wastes within the Scope of the Zero Waste Project: Heterogeneous Multi-Vehicle Routing Case in Kirikkale. *Environ Monit Assess*, 192-214.

Laporte, G. (1992). The Vehicle Routing Problem: An Overview of Exact and Approximate Algorithms. *European Journal of Operational Research*, 345-358.

Laporte, G. (2007). What You Should Know about the Vehicle Routing Problem. *Naval Research Logistics*, 811-819.

Markov, I., Varone, S., & Bierlaire, M. (2014). Vehicle Routing for a Complex Waste Collection Problem. *Vehicle Routing for a Complex Waste Collection* (s. 1-25). Monte Verita/Ascona: Commission for Technology and Innovation.

Molina, J. C., Eguia, I., & Racero, J. (2019). Reducing Pollutant Emissions in a Waste Collection Vehicle Routing Problem using a Variable Neighborhood Tabu Search Algorithm: A Case Study. *An Official Journal of the Spanish Society of Statistics and Operations Research*, 345-380.

Pepper, J. W., Golden, B. L., & Wassil, E. A. (2002). Solving the Traveling Salesman Problem With Annealing-Based Heuristics: A Computational Study. *IEEE Transactions On Systems, Man, And Cybernetics—Part A: Systems And Humans*, 72-77.

Qiao, Q., Tao, F., Wu, H., Yu, X., & Zhang, M. (2020). Optimization of a Capacitated Vehicle Routing Problem for Sustainable Municipal Solid Waste Collection Management Using the PSO-TS Algorithm. *International Journal of Environmental Research and Public Health*, 1-22.

Rabbani, M., Farrokhi-asl, H., & Rafiei, H. (2016). A Hybrid Genetic Algorithm for Waste Collection Problem by Heterogeneous Fleet of Vehicles with Multiple Separated Compartments. *Journal of Intelligent & Fuzzy Systems*, 1817-1830.

Rızvanoğlu, O. (2018). *Katı Atık Toplama Güzergah Optimizasyonu: Haliliye (Şanlıurfa) İlçesi Örneği.* Yüksek Lisans Tezi, Harran Üniversitesi Fen Bilimleri Enstitüsü: Şanlıurfa.

Şeker, Ş. (2007). *Araç Rotalama Problemleri ve Zaman Pencereli Stokastik Araç Rotalama Problemine Genetik Algoritma Yaklaşımı.* Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü: İstanbul.

Takan, M. A. (2019). *Araç Rotalama Problemleri için Matematiksel Modeller ve Subgradyant Temelli Çözüm Yaklaşımı.* Doktora Tezi, Eskişehir Teknik Üniversitesi Lisansüstü Eğitim Enstitüsü: Eskişehir.

Talbi, E.-G. (2009). *Metaheuristic From Design to Implementation.* New Jersey: John Wiley&Sons.

Tirkolaee, E. B., Abbasian, P., Soltani, M., & Ghaffarian, S. A. (2019). Developing an Applied Algorithm for Multi-Trip Vehicle Routing Problem with Time Windows in Urban Waste Collection: A Case Study. *Waste Management & Research*, 4-13.

Wood, I. A., & Downs, T. (1998). Fast optimization by Demon Algorithms. *Proceedings Australian Conference on Neural Networks*, 245-249.

Yılmaz, N. (2018). *Etkin Keşif Gözetleme Sistemi için Kablo Güdümlü Balonlar ve İHA Sistemlerine Yönelik Yer Seçimi ve Rotalama Model Önerileri.* Doktora Tezi, Gazi Üniversitesi Fen Bilimleri Enstitüsü: Ankara.

Yurtdaş, H. (2018). *Çapraz Sevkiyatta Araç Rotalama Problemi için bir Sezgisel Çözüm Yaklaşımı.* Doktora Tezi, Gazi Üniversitesi Fen Bilimleri Enstitüsü: Ankara.

EKLER

EK-1. Problemin çözüm sonuçları

Çözüm	İterasyon Sayısı	$D_{sınır}$ Değeri	D^- Değeri	Rastgele Sayı	Çözüm Süresi (milisaniye)	Toplam Mesafe (km)
1	3 000 000	40	1	5	8 350	4 935,168
2	3 000 000	40	1	26	8 528	5 051,201
3	3 000 000	40	1	38	8 386	5 168,027
4	3 000 000	40	1	47	8 364	5 026,189
5	3 000 000	40	1	71	8 318	5 073,454
6	3 000 000	40	1	89	8 351	5 001,540
7	3 000 000	40	1	101	8 465	5 253,265
8	3 000 000	40	1	118	8 340	4 993,481
9	3 000 000	40	1	128	8 550	5 004,962
10	3 000 000	40	1	136	8 399	5 007,498
11	3 000 000	40	1	149	8 480	5 028,721
12	3 000 000	40	1	157	8 560	5 019,122
13	3 000 000	40	1	168	8 323	5 104,825
14	3 000 000	40	1	179	8 295	4 969,200
15	3 000 000	40	1	211	8 421	4 959,963
16	3 000 000	40	1	218	8 439	5 029,564
17	3 000 000	40	1	236	8 614	5 074,684
18	3 000 000	40	1	251	8 514	4 996,730
19	3 000 000	40	1	257	8 485	5 127,604
20	3 000 000	40	1	280	8 379	5 165,957
21	3 000 000	40	1	293	8 309	4 944,316
22	3 000 000	40	1	313	8 976	5 202,671
23	3 000 000	40	1	336	9 000	4 992,127
24	3 000 000	40	1	352	9 021	5 074,856
25	3 000 000	40	1	379	9 152	4 996,347
26	3 000 000	40	1	381	8 755	5 078,425
27	3 000 000	40	1	409	9 328	4 960,141
28	3 000 000	40	1	417	8 912	5 093,881
29	3 000 000	40	1	438	8 921	5 138,983
30	3 000 000	40	1	461	8 801	4 920,872
31	3 000 000	40	1	484	8 968	5 042,736
32	3 000 000	40	1	503	8 957	5 130,999
33	3 000 000	40	1	547	8 883	5 049,217
34	3 000 000	40	1	560	9 015	4 898,720
35	3 000 000	40	1	587	8 981	4 947,335
36	3 000 000	40	1	607	9 028	5 284,627
37	3 000 000	40	1	619	8 852	5 107,025
38	3 000 000	40	1	653	8 997	4 893,986
39	3 000 000	40	1	680	8 939	5 136,983
40	3 000 000	40	1	701	9 238	5 169,814
Ortalama					8 689,85	5 051,380

Çözüm	İterasyon Sayısı	$D_{sınır}$ Değeri	D^- Değeri	Rastgele Sayı	Çözüm Süresi (milisaniye)	Toplam Mesafe (km)
41	3 000 000	60	1	5	9 038	5 389,396
42	3 000 000	60	1	26	8 357	5 567,599
43	3 000 000	60	1	38	8 565	5 543,592
44	3 000 000	60	1	47	8 462	5 472,984
45	3 000 000	60	1	71	8 605	5 501,047
46	3 000 000	60	1	89	8 453	5 641,801
47	3 000 000	60	1	101	8 684	5 650,750
48	3 000 000	60	1	118	8 521	5 612,699
49	3 000 000	60	1	128	8 562	5 458,162
50	3 000 000	60	1	136	8 286	5 678,461
51	3 000 000	60	1	149	8 582	5 452,537
52	3 000 000	60	1	157	8 545	5 611,562
53	3 000 000	60	1	168	8 568	5 630,035
54	3 000 000	60	1	179	8 454	5 485,302
55	3 000 000	60	1	211	8 506	5 515,967
56	3 000 000	60	1	218	8 457	5 717,270
57	3 000 000	60	1	236	8 444	5 398,548
58	3 000 000	60	1	251	8 454	5 408,173
59	3 000 000	60	1	257	8 493	5 558,569
60	3 000 000	60	1	280	8 422	5 724,009
61	3 000 000	60	1	293	8 511	5 560,745
62	3 000 000	60	1	313	8 463	5 383,990
63	3 000 000	60	1	336	8 489	5 470,303
64	3 000 000	60	1	352	8 456	5 740,009
65	3 000 000	60	1	379	8 332	5 480,637
66	3 000 000	60	1	381	8 429	5 576,972
67	3 000 000	60	1	409	8 697	5 508,925
68	3 000 000	60	1	417	8 476	5 525,022
69	3 000 000	60	1	438	8 484	5 486,497
70	3 000 000	60	1	461	8 632	5 544,634
71	3 000 000	60	1	484	8 491	5 600,505
72	3 000 000	60	1	503	8 526	5 644,536
73	3 000 000	60	1	547	8 468	5 582,508
74	3 000 000	60	1	560	8 568	5 536,963
75	3 000 000	60	1	587	8 615	5 417,868
76	3 000 000	60	1	607	8 499	5 572,085
77	3 000 000	60	1	619	8 326	5 707,251
78	3 000 000	60	1	653	8 416	5 490,148
79	3 000 000	60	1	680	8 403	5 541,480
80	3 000 000	60	1	701	8 706	5 458,101
Ortalama					8 511,13	5 546,191

Çözüm	İterasyon Sayısı	$D_{sınır}$ Değeri	D^- Değeri	Rastgele Sayı	Çözüm Süresi (milisaniye)	Toplam Mesafe (km)
81	3 000 000	80	1	5	8 014	6 142,499
82	3 000 000	80	1	26	8 127	5 927,110
83	3 000 000	80	1	38	8 077	5 870,696
84	3 000 000	80	1	47	8 047	5 909,199
85	3 000 000	80	1	71	8 055	5 865,288
86	3 000 000	80	1	89	8 135	5 996,615
87	3 000 000	80	1	101	8 096	5 988,413
88	3 000 000	80	1	118	8 089	6 163,388
89	3 000 000	80	1	128	8 050	5 956,056
90	3 000 000	80	1	136	7 945	6 086,595
91	3 000 000	80	1	149	7 951	5 937,147
92	3 000 000	80	1	157	8 115	6 048,064
93	3 000 000	80	1	168	8 017	6 095,837
94	3 000 000	80	1	179	8 154	5 901,858
95	3 000 000	80	1	211	8 138	5 933,340
96	3 000 000	80	1	218	8 115	6 013,953
97	3 000 000	80	1	236	7 902	5 994,263
98	3 000 000	80	1	251	8 123	5 976,358
99	3 000 000	80	1	257	8 075	5 953,921
100	3 000 000	80	1	280	8 068	6 029,429
101	3 000 000	80	1	293	7 989	6 060,393
102	3 000 000	80	1	313	8 007	5 928,750
103	3 000 000	80	1	336	8 004	6 121,752
104	3 000 000	80	1	352	8 130	6 115,360
105	3 000 000	80	1	379	8 166	6 052,490
106	3 000 000	80	1	381	8 057	5 802,664
107	3 000 000	80	1	409	8 071	6 024,532
108	3 000 000	80	1	417	8 097	5 964,854
109	3 000 000	80	1	438	8 117	5 949,770
110	3 000 000	80	1	461	8 231	6 138,309
111	3 000 000	80	1	484	8 174	5 857,830
112	3 000 000	80	1	503	9 240	5 975,320
113	3 000 000	80	1	547	8 733	6 132,499
114	3 000 000	80	1	560	8 211	6 201,262
115	3 000 000	80	1	587	8 200	6 040,767
116	3 000 000	80	1	607	8 257	5 866,462
117	3 000 000	80	1	619	7 996	5 991,230
118	3 000 000	80	1	653	8 126	5 878,573
119	3 000 000	80	1	680	8 036	6 094,859
120	3 000 000	80	1	701	8 104	5 967,209
Ortalama					8 130,98	5 998,873

Çözüm	İterasyon Sayısı	$D_{sınır}$ Değeri	D^- Değeri	Rastgele Sayı	Çözüm Süresi (milisaniye)	Toplam Mesafe (km)
121	3 000 000	100	1	5	8 144	6 378,178
122	3 000 000	100	1	26	7 963	6 299,618
123	3 000 000	100	1	38	7 958	6 141,481
124	3 000 000	100	1	47	8 134	6 397,645
125	3 000 000	100	1	71	8 074	6 259,835
126	3 000 000	100	1	89	8 151	6 230,901
127	3 000 000	100	1	101	8 076	6 437,721
128	3 000 000	100	1	118	8 129	6 564,560
129	3 000 000	100	1	128	8 008	6 481,652
130	3 000 000	100	1	136	7 968	6 323,623
131	3 000 000	100	1	149	7 869	6 154,264
132	3 000 000	100	1	157	8 021	6 308,712
133	3 000 000	100	1	168	8 097	6 406,905
134	3 000 000	100	1	179	8 077	6 148,112
135	3 000 000	100	1	211	8 086	6 244,262
136	3 000 000	100	1	218	8 015	6 154,540
137	3 000 000	100	1	236	10 856	6 326,034
138	3 000 000	100	1	251	11 278	6 061,869
139	3 000 000	100	1	257	11 173	6 306,428
140	3 000 000	100	1	280	11 278	6 278,761
141	3 000 000	100	1	293	11 111	6 295,183
142	3 000 000	100	1	313	11 251	6 265,640
143	3 000 000	100	1	336	11 337	5 962,675
144	3 000 000	100	1	352	11 048	6 200,414
145	3 000 000	100	1	379	8 028	6 320,767
146	3 000 000	100	1	381	8 216	6 359,616
147	3 000 000	100	1	409	8 136	6 345,965
148	3 000 000	100	1	417	8 065	6 239,714
149	3 000 000	100	1	438	8 002	6 235,807
150	3 000 000	100	1	461	7 891	6 374,030
151	3 000 000	100	1	484	8 074	6 341,347
152	3 000 000	100	1	503	7 978	6 296,834
153	3 000 000	100	1	547	8 019	6 317,614
154	3 000 000	100	1	560	7 873	6 388,463
155	3 000 000	100	1	587	8 055	6 203,734
156	3 000 000	100	1	607	8 039	6 215,351
157	3 000 000	100	1	619	8 047	6 329,961
158	3 000 000	100	1	653	8 119	6 314,631
159	3 000 000	100	1	680	7 929	6 316,262
160	3 000 000	100	1	701	7 927	6 269,591
Ortalama					8 662,50	6 287,468

Çözüm	İterasyon Sayısı	$D_{sınır}$ Değeri	D^- Değeri	Rastgele Sayı	Çözüm Süresi (milisaniye)	Toplam Mesafe (km)
161	6 000 000	40	1	5	19 372	4 406,338
162	6 000 000	40	1	26	19 581	4 294,615
163	6 000 000	40	1	38	19 452	4 222,828
164	6 000 000	40	1	47	18 870	4 337,023
165	6 000 000	40	1	71	19 077	4 297,303
166	6 000 000	40	1	89	19 162	4 319,401
167	6 000 000	40	1	101	18 911	4 244,657
168	6 000 000	40	1	118	18 983	4 440,212
169	6 000 000	40	1	128	18 911	4 288,118
170	6 000 000	40	1	136	19 172	4 378,449
171	6 000 000	40	1	149	19 026	4 365,458
172	6 000 000	40	1	157	19 081	4 346,362
173	6 000 000	40	1	168	19 771	4 244,347
174	6 000 000	40	1	179	18 917	4 326,351
175	6 000 000	40	1	211	19 121	4 384,707
176	6 000 000	40	1	218	18 899	4 235,770
177	6 000 000	40	1	236	18 894	4 313,815
178	6 000 000	40	1	251	19 271	4 310,255
179	6 000 000	40	1	257	18 952	4 328,483
180	6 000 000	40	1	280	18 799	4 319,798
181	6 000 000	40	1	293	19 000	4 363,716
182	6 000 000	40	1	313	19 022	4 434,660
183	6 000 000	40	1	336	18 855	4 273,708
184	6 000 000	40	1	352	19 090	4 300,489
185	6 000 000	40	1	379	19 486	4 434,759
186	6 000 000	40	1	381	18 802	4 290,757
187	6 000 000	40	1	409	19 487	4 175,680
188	6 000 000	40	1	417	19 368	4 232,016
189	6 000 000	40	1	438	18 958	4 264,706
190	6 000 000	40	1	461	18 782	4 290,942
191	6 000 000	40	1	484	18 945	4 248,451
192	6 000 000	40	1	503	18 776	4 307,186
193	6 000 000	40	1	547	19 074	4 340,455
194	6 000 000	40	1	560	19 009	4 252,364
195	6 000 000	40	1	587	18 901	4 324,654
196	6 000 000	40	1	607	19 200	4 335,355
197	6 000 000	40	1	619	18 922	4 446,516
198	6 000 000	40	1	653	18 867	4 369,170
199	6 000 000	40	1	680	19 044	4 335,395
200	6 000 000	40	1	701	19 035	4 278,299
Ortalama					19 071,13	4 317,589

Çözüm	İterasyon Sayısı	$D_{sınır}$ Değeri	D^- Değeri	Rastgele Sayı	Çözüm Süresi (milisaniye)	Toplam Mesafe (km)
201	6 000 000	60	1	5	17 872	4 665,580
202	6 000 000	60	1	26	17 795	4 735,531
203	6 000 000	60	1	38	17 734	4 705,802
204	6 000 000	60	1	47	17 535	4 717,749
205	6 000 000	60	1	71	17 699	4 677,091
206	6 000 000	60	1	89	17 636	4 671,625
207	6 000 000	60	1	101	17 847	4 705,731
208	6 000 000	60	1	118	17 472	4 658,951
209	6 000 000	60	1	128	17 933	4 868,592
210	6 000 000	60	1	136	17 420	4 643,832
211	6 000 000	60	1	149	17 438	4 778,203
212	6 000 000	60	1	157	17 630	4 729,291
213	6 000 000	60	1	168	17 763	4 688,058
214	6 000 000	60	1	179	17 902	4 782,512
215	6 000 000	60	1	211	17 893	4 738,944
216	6 000 000	60	1	218	17 717	4 712,157
217	6 000 000	60	1	236	17 573	4 790,695
218	6 000 000	60	1	251	17 734	4 749,178
219	6 000 000	60	1	257	17 606	4 719,507
220	6 000 000	60	1	280	17 872	4 758,570
221	6 000 000	60	1	293	17 753	4 671,850
222	6 000 000	60	1	313	17 499	4 626,641
223	6 000 000	60	1	336	17 662	4 810,664
224	6 000 000	60	1	352	17 799	4 638,436
225	6 000 000	60	1	379	17 753	4 684,232
226	6 000 000	60	1	381	17 700	4 682,195
227	6 000 000	60	1	409	17 562	4 764,228
228	6 000 000	60	1	417	17 638	4 787,218
229	6 000 000	60	1	438	17 518	4 722,292
230	6 000 000	60	1	461	17 352	4 701,653
231	6 000 000	60	1	484	17 730	4 560,058
232	6 000 000	60	1	503	17 745	4 690,863
233	6 000 000	60	1	547	17 568	4 746,528
234	6 000 000	60	1	560	17 797	4 641,336
235	6 000 000	60	1	587	17 930	4 779,890
236	6 000 000	60	1	607	17 579	4 611,859
237	6 000 000	60	1	619	17 691	4 704,980
238	6 000 000	60	1	653	17 718	4 794,119
239	6 000 000	60	1	680	17 677	4 789,843
240	6 000 000	60	1	701	17 562	4 710,920
Ortalama					17 682,60	4 715,435

Çözüm	İterasyon Sayısı	$D_{sınır}$ Değeri	D^- Değeri	Rastgele Sayı	Çözüm Süresi (milisaniye)	Toplam Mesafe (km)
241	6 000 000	80	1	5	21 921	5 138,259
242	6 000 000	80	1	26	17 113	5 015,013
243	6 000 000	80	1	38	16 850	5 133,101
244	6 000 000	80	1	47	16 812	5 014,503
245	6 000 000	80	1	71	24 036	4 943,151
246	6 000 000	80	1	89	23 855	5 040,533
247	6 000 000	80	1	101	23 608	4 842,886
248	6 000 000	80	1	118	23 568	4 979,055
249	6 000 000	80	1	128	23 388	5 045,838
250	6 000 000	80	1	136	23 283	5 137,487
251	6 000 000	80	1	149	23 293	5 038,634
252	6 000 000	80	1	157	24 269	5 146,648
253	6 000 000	80	1	168	23 823	4 946,057
254	6 000 000	80	1	179	23 576	4 983,559
255	6 000 000	80	1	211	24 111	5 006,814
256	6 000 000	80	1	218	28 739	5 000,912
257	6 000 000	80	1	236	17 126	5 063,313
258	6 000 000	80	1	251	16 762	4 910,727
259	6 000 000	80	1	257	16 922	4 949,424
260	6 000 000	80	1	280	16 781	5 049,408
261	6 000 000	80	1	293	16 750	5 026,216
262	6 000 000	80	1	313	16 927	5 039,829
263	6 000 000	80	1	336	16 660	5 010,541
264	6 000 000	80	1	352	16 633	5 025,471
265	6 000 000	80	1	379	16 736	5 109,354
266	6 000 000	80	1	381	16 650	4 980,166
267	6 000 000	80	1	409	16 721	5 116,975
268	6 000 000	80	1	417	16 933	5 054,111
269	6 000 000	80	1	438	16 737	4 991,605
270	6 000 000	80	1	461	16 706	4 988,958
271	6 000 000	80	1	484	16 658	5 253,398
272	6 000 000	80	1	503	16 499	5 045,199
273	6 000 000	80	1	547	16 613	5 130,814
274	6 000 000	80	1	560	17 242	4 879,895
275	6 000 000	80	1	587	16 801	5 051,386
276	6 000 000	80	1	607	16 573	5 035,372
277	6 000 000	80	1	619	16 619	5 152,559
278	6 000 000	80	1	653	16 384	4 928,010
279	6 000 000	80	1	680	16 809	5 073,338
280	6 000 000	80	1	701	17 153	4 922,164
Ortalama					19 116	5 030,017

Çözüm	İterasyon Sayısı	$D_{sınır}$ Değeri	D^- Değeri	Rastgele Sayı	Çözüm Süresi (milisaniye)	Toplam Mesafe (km)
281	6 000 000	100	1	5	17 098	5 423,897
282	6 000 000	100	1	26	16 647	5 337,919
283	6 000 000	100	1	38	16 652	5 228,175
284	6 000 000	100	1	47	16 781	5 166,064
285	6 000 000	100	1	71	17 127	5 236,552
286	6 000 000	100	1	89	16 164	5 298,326
287	6 000 000	100	1	101	15 953	5 389,994
288	6 000 000	100	1	118	16 489	5 219,414
289	6 000 000	100	1	128	16 598	5 373,036
290	6 000 000	100	1	136	16 366	5 315,264
291	6 000 000	100	1	149	16 461	5 288,139
292	6 000 000	100	1	157	16 705	5 287,204
293	6 000 000	100	1	168	16 131	5 160,333
294	6 000 000	100	1	179	16 211	5 402,570
295	6 000 000	100	1	211	16 322	5 280,716
296	6 000 000	100	1	218	16 322	5 368,941
297	6 000 000	100	1	236	16 567	5 341,141
298	6 000 000	100	1	251	16 317	5 284,087
299	6 000 000	100	1	257	16 214	5 208,691
300	6 000 000	100	1	280	16 342	5 379,676
301	6 000 000	100	1	293	16 577	5 269,404
302	6 000 000	100	1	313	16 139	5 307,364
303	6 000 000	100	1	336	16 699	5 273,111
304	6 000 000	100	1	352	16 202	5 288,541
305	6 000 000	100	1	379	16 406	5 379,764
306	6 000 000	100	1	381	16 324	5 153,519
307	6 000 000	100	1	409	16 450	5 225,868
308	6 000 000	100	1	417	16 293	5 231,544
309	6 000 000	100	1	438	16 571	5 198,908
310	6 000 000	100	1	461	16 362	5 307,286
311	6 000 000	100	1	484	16 515	5 354,529
312	6 000 000	100	1	503	16 590	5 430,263
313	6 000 000	100	1	547	16 681	5 214,107
314	6 000 000	100	1	560	16 547	5 335,659
315	6 000 000	100	1	587	16 698	5 294,449
316	6 000 000	100	1	607	16 688	5 322,148
317	6 000 000	100	1	619	16 652	5 160,384
318	6 000 000	100	1	653	17 009	5 200,862
319	6 000 000	100	1	680	17 002	5 319,082
320	6 000 000	100	1	701	16 579	5 326,929
Ortalama					16 511,28	5 289,597

Çözüm	İterasyon Sayısı	$D_{sınır}$ Değeri	D^{-} Değeri	Rastgele Sayı	Çözüm Süresi (milisaniye)	Toplam Mesafe (km)
321	12 000 000	40	1	5	38 947	3 896,527
322	12 000 000	40	1	26	39 336	3 789,448
323	12 000 000	40	1	38	39 348	3 730,729
324	12 000 000	40	1	47	39 688	3 827,374
325	12 000 000	40	1	71	39 285	3 808,637
326	12 000 000	40	1	89	39 568	3 849,437
327	12 000 000	40	1	101	37 198	3 766,347
328	12 000 000	40	1	118	36 492	3 809,882
329	12 000 000	40	1	128	36 748	3 692,844
330	12 000 000	40	1	136	36 687	3 722,614
331	12 000 000	40	1	149	36 997	3 796,261
332	12 000 000	40	1	157	36 889	3 765,554
333	12 000 000	40	1	168	37 341	3 789,298
334	12 000 000	40	1	179	35 945	3 810,612
335	12 000 000	40	1	211	36 731	3 791,913
336	12 000 000	40	1	218	36 381	3 758,555
337	12 000 000	40	1	236	36 407	3 767,765
338	12 000 000	40	1	251	36 860	3 785,639
339	12 000 000	40	1	257	36 626	3 837,755
340	12 000 000	40	1	280	36 406	3 765,537
341	12 000 000	40	1	293	36 747	3 770,066
342	12 000 000	40	1	313	36 520	3 796,523
343	12 000 000	40	1	336	36 099	3 779,125
344	12 000 000	40	1	352	36 556	3 798,905
345	12 000 000	40	1	379	37 108	3 828,650
346	12 000 000	40	1	381	36 293	3 805,045
347	12 000 000	40	1	409	36 353	3 722,112
348	12 000 000	40	1	417	36 836	3 837,081
349	12 000 000	40	1	438	36 616	3 786,666
350	12 000 000	40	1	461	36 089	3 841,232
351	12 000 000	40	1	484	36 456	3 808,213
352	12 000 000	40	1	503	36 229	3 804,818
353	12 000 000	40	1	547	36 992	3 733,103
354	12 000 000	40	1	560	36 409	3 791,440
355	12 000 000	40	1	587	36 177	3 809,762
356	12 000 000	40	1	607	37 377	3 791,710
357	12 000 000	40	1	619	36 569	3 752,588
358	12 000 000	40	1	653	36 645	3 850,672
359	12 000 000	40	1	680	36 873	3 749,459
360	12 000 000	40	1	701	36 757	3 710,819
Ortalama					37 039,53	3 788,268

Çözüm	İterasyon Sayısı	$D_{sınır}$ Değeri	D^- Değeri	Rastgele Sayı	Çözüm Süresi (milisaniye)	Toplam Mesafe (km)
361	12 000 000	60	1	5	36 480	4 107,474
362	12 000 000	60	1	26	36 455	4 140,747
363	12 000 000	60	1	38	36 151	4 045,448
364	12 000 000	60	1	47	36 358	4 090,481
365	12 000 000	60	1	71	36 374	4 069,329
366	12 000 000	60	1	89	36 439	4 122,256
367	12 000 000	60	1	101	36 569	4 021,129
368	12 000 000	60	1	118	36 151	4 126,484
369	12 000 000	60	1	128	36 331	4 136,242
370	12 000 000	60	1	136	36 427	4 083,299
371	12 000 000	60	1	149	36 486	4 071,056
372	12 000 000	60	1	157	36 011	4 083,030
373	12 000 000	60	1	168	36 305	4 093,262
374	12 000 000	60	1	179	37 994	4 048,348
375	12 000 000	60	1	211	38 752	4 074,383
376	12 000 000	60	1	218	38 624	3 964,897
377	12 000 000	60	1	236	36 632	4 087,302
378	12 000 000	60	1	251	38 473	4 045,771
379	12 000 000	60	1	257	38 613	4 070,764
380	12 000 000	60	1	280	38 121	4 094,446
381	12 000 000	60	1	293	38 174	4 127,587
382	12 000 000	60	1	313	35 885	4 179,208
383	12 000 000	60	1	336	36 835	4 079,983
384	12 000 000	60	1	352	36 492	4 124,286
385	12 000 000	60	1	379	36 796	4 120,883
386	12 000 000	60	1	381	36 291	4 126,398
387	12 000 000	60	1	409	36 662	4 118,793
388	12 000 000	60	1	417	36 427	4 029,817
389	12 000 000	60	1	438	36 880	3 963,226
390	12 000 000	60	1	461	36 299	4 110,078
391	12 000 000	60	1	484	36 018	4 073,334
392	12 000 000	60	1	503	36 554	4 039,327
393	12 000 000	60	1	547	36 377	4 056,303
394	12 000 000	60	1	560	36 605	4 165,714
395	12 000 000	60	1	587	36 539	4 085,375
396	12 000 000	60	1	607	36 355	4 084,632
397	12 000 000	60	1	619	38 391	4 009,135
398	12 000 000	60	1	653	51 001	4 121,196
399	12 000 000	60	1	680	50 798	3 958,501
400	12 000 000	60	1	701	49 678	4 017,315
Ortalama					37 870,08	4 079,181

Çözüm	İterasyon Sayısı	$D_{sınır}$ Değeri	D^- Değeri	Rastgele Sayı	Çözüm Süresi (milisaniye)	Toplam Mesafe (km)
441	12 000 000	100	1	5	37 380	4 540,491
442	12 000 000	100	1	26	37 630	4 578,465
443	12 000 000	100	1	38	37 772	4 549,163
444	12 000 000	100	1	47	37 880	4 511,461
445	12 000 000	100	1	71	37 507	4 690,820
446	12 000 000	100	1	89	37 643	4 561,277
447	12 000 000	100	1	101	37 600	4 732,664
448	12 000 000	100	1	118	37 655	4 558,191
449	12 000 000	100	1	128	37 574	4 424,453
450	12 000 000	100	1	136	38 197	4 478,292
451	12 000 000	100	1	149	37 613	4 478,789
452	12 000 000	100	1	157	37 267	4 626,684
453	12 000 000	100	1	168	37 363	4 636,523
454	12 000 000	100	1	179	37 398	4 478,865
455	12 000 000	100	1	211	38 004	4 458,028
456	12 000 000	100	1	218	37 768	4 510,314
457	12 000 000	100	1	236	37 712	4 569,008
458	12 000 000	100	1	251	37 308	4 521,757
459	12 000 000	100	1	257	37 418	4 595,293
460	12 000 000	100	1	280	37 373	4 486,765
461	12 000 000	100	1	293	38 798	4 573,297
462	12 000 000	100	1	313	37 514	4 564,878
463	12 000 000	100	1	336	37 774	4 578,716
464	12 000 000	100	1	352	37 785	4 503,792
465	12 000 000	100	1	379	38 255	4 554,227
466	12 000 000	100	1	381	37 156	4 495,106
467	12 000 000	100	1	409	37 498	4 438,537
468	12 000 000	100	1	417	37 577	4 547,846
469	12 000 000	100	1	438	37 555	4 575,164
470	12 000 000	100	1	461	37 256	4 489,992
471	12 000 000	100	1	484	37 779	4 594,524
472	12 000 000	100	1	503	37 280	4 420,059
473	12 000 000	100	1	547	37 383	4 539,306
474	12 000 000	100	1	560	37 668	4 627,510
475	12 000 000	100	1	587	37 879	4 470,791
476	12 000 000	100	1	607	37 715	4 613,026
477	12 000 000	100	1	619	37 470	4 537,279
478	12 000 000	100	1	653	38 017	4 543,570
479	12 000 000	100	1	680	39 532	4 458,827
480	12 000 000	100	1	701	51 321	4 550,587
Ortalama					38 031,85	4 541,608

Çözüm	İterasyon Sayısı	$D_{sınır}$ Değeri	D^- Değeri	Rastgele Sayı	Çözüm Süresi (milisaniye)	Toplam Mesafe (km)
481	24 000 000	40	1	5	106 115	3 349,219
482	24 000 000	40	1	26	106 081	3 363,734
483	24 000 000	40	1	38	106 150	3 358,972
484	24 000 000	40	1	47	106 049	3 318,201
485	24 000 000	40	1	71	111 547	3 368,017
486	24 000 000	40	1	89	80 446	3 420,091
487	24 000 000	40	1	101	75 074	3 348,608
488	24 000 000	40	1	118	76 376	3 348,430
489	24 000 000	40	1	128	75 533	3 415,963
490	24 000 000	40	1	136	75 073	3 359,851
491	24 000 000	40	1	149	74 727	3 387,628
492	24 000 000	40	1	157	75 904	3 364,810
493	24 000 000	40	1	168	74 214	3 404,629
494	24 000 000	40	1	179	75 549	3 410,649
495	24 000 000	40	1	211	75 929	3 330,599
496	24 000 000	40	1	218	75 299	3 387,460
497	24 000 000	40	1	236	74 905	3 345,159
498	24 000 000	40	1	251	75 085	3 365,394
499	24 000 000	40	1	257	75 761	3 401,565
500	24 000 000	40	1	280	75 623	3 337,535
501	24 000 000	40	1	293	74 244	3 419,560
502	24 000 000	40	1	313	74 410	3 361,674
503	24 000 000	40	1	336	75 804	3 365,786
504	24 000 000	40	1	352	73 790	3 311,211
505	24 000 000	40	1	379	73 459	3 409,494
506	24 000 000	40	1	381	73 236	3 335,667
507	24 000 000	40	1	409	74 495	3 445,480
508	24 000 000	40	1	417	74 223	3 440,974
509	24 000 000	40	1	438	73 783	3 415,289
510	24 000 000	40	1	461	73 669	3 337,834
511	24 000 000	40	1	484	72 930	3 465,935
512	24 000 000	40	1	503	74 147	3 338,871
513	24 000 000	40	1	547	72 730	3 404,365
514	24 000 000	40	1	560	73 362	3 417,460
515	24 000 000	40	1	587	73 769	3 355,732
516	24 000 000	40	1	607	73 501	3 370,152
517	24 000 000	40	1	619	73 810	3 415,219
518	24 000 000	40	1	653	74 456	3 439,636
519	24 000 000	40	1	680	76 695	3 369,051
520	24 000 000	40	1	701	76 484	3 386,460
Ortalama					78 860,93	3 379,809

Çözüm	İterasyon Sayısı	$D_{sınır}$ Değeri	D^- Değeri	Rastgele Sayı	Çözüm Süresi (milisaniye)	Toplam Mesafe (km)
521	24 000 000	60	1	5	74 982	3 597,740
522	24 000 000	60	1	26	79 948	3 603,123
523	24 000 000	60	1	38	74 813	3 548,064
524	24 000 000	60	1	47	75 951	3 609,714
525	24 000 000	60	1	71	74 041	3 614,463
526	24 000 000	60	1	89	74 459	3 537,135
527	24 000 000	60	1	101	75 441	3 632,329
528	24 000 000	60	1	118	74 360	3 531,223
529	24 000 000	60	1	128	75 955	3 580,925
530	24 000 000	60	1	136	75 273	3 584,456
531	24 000 000	60	1	149	74 680	3 644,034
532	24 000 000	60	1	157	73 716	3 655,189
533	24 000 000	60	1	168	74 078	3 627,059
534	24 000 000	60	1	179	75 013	3 558,686
535	24 000 000	60	1	211	74 486	3 545,473
536	24 000 000	60	1	218	74 328	3 581,994
537	24 000 000	60	1	236	73 655	3 589,954
538	24 000 000	60	1	251	75 061	3 596,343
539	24 000 000	60	1	257	74 806	3 611,834
540	24 000 000	60	1	280	75 111	3 610,290
541	24 000 000	60	1	293	74 540	3 608,721
542	24 000 000	60	1	313	74 573	3 633,274
543	24 000 000	60	1	336	74 637	3 637,106
544	24 000 000	60	1	352	74 032	3 546,000
545	24 000 000	60	1	379	73 889	3 583,930
546	24 000 000	60	1	381	73 806	3 627,969
547	24 000 000	60	1	409	80 088	3 614,449
548	24 000 000	60	1	417	74 681	3 605,396
549	24 000 000	60	1	438	75 195	3 589,050
550	24 000 000	60	1	461	73 405	3 506,618
551	24 000 000	60	1	484	73 634	3 623,303
552	24 000 000	60	1	503	74 369	3 578,361
553	24 000 000	60	1	547	74 063	3 584,589
554	24 000 000	60	1	560	74 636	3 534,618
555	24 000 000	60	1	587	73 975	3 542,691
556	24 000 000	60	1	607	74 505	3 611,193
557	24 000 000	60	1	619	74 030	3 621,112
558	24 000 000	60	1	653	74 133	3 602,162
559	24 000 000	60	1	680	73 565	3 622,798
560	24 000 000	60	1	701	74 320	3 638,547
Ortalama					74 755,83	3 594,298

Çözüm	İterasyon Sayısı	$D_{sınır}$ Değeri	D^- Değeri	Rastgele Sayı	Çözüm Süresi (milisaniye)	Toplam Mesafe (km)
561	24 000 000	80	1	5	102 796	3 774,399
562	24 000 000	80	1	26	102 627	3 851,848
563	24 000 000	80	1	38	104 409	3 771,377
564	24 000 000	80	1	47	70 863	3 764,418
565	24 000 000	80	1	71	72 862	3 847,591
566	24 000 000	80	1	89	71 359	3 809,072
567	24 000 000	80	1	101	72 073	3 817,910
568	24 000 000	80	1	118	70 602	3 756,668
569	24 000 000	80	1	128	71 973	3 790,315
570	24 000 000	80	1	136	70 791	3 781,592
571	24 000 000	80	1	149	72 504	3 798,453
572	24 000 000	80	1	157	71 624	3 847,790
573	24 000 000	80	1	168	72 118	3 744,451
574	24 000 000	80	1	179	70 826	3 828,594
575	24 000 000	80	1	211	71 796	3 768,530
576	24 000 000	80	1	218	71 839	3 820,587
577	24 000 000	80	1	236	72 939	3 841,420
578	24 000 000	80	1	251	71 834	3 801,770
579	24 000 000	80	1	257	71 407	3 790,716
580	24 000 000	80	1	280	72 000	3 734,683
581	24 000 000	80	1	293	71 157	3 716,628
582	24 000 000	80	1	313	71 948	3 811,487
583	24 000 000	80	1	336	71 837	3 772,781
584	24 000 000	80	1	352	71 790	3 760,946
585	24 000 000	80	1	379	71 220	3 800,758
586	24 000 000	80	1	381	71 923	3 679,744
587	24 000 000	80	1	409	71 816	3 835,900
588	24 000 000	80	1	417	71 395	3 734,040
589	24 000 000	80	1	438	72 178	3 802,843
590	24 000 000	80	1	461	71 974	3 773,237
591	24 000 000	80	1	484	70 936	3 752,394
592	24 000 000	80	1	503	72 090	3 773,624
593	24 000 000	80	1	547	71 203	3 734,984
594	24 000 000	80	1	560	71 527	3 855,573
595	24 000 000	80	1	587	101 404	3 748,924
596	24 000 000	80	1	607	101 645	3 771,467
597	24 000 000	80	1	619	101 314	3 761,995
598	24 000 000	80	1	653	101 531	3 851,590
599	24 000 000	80	1	680	99 546	3 761,839
600	24 000 000	80	1	701	100 009	3 745,845
Ortalama					78 442,13	3 784,720

Çözüm	İterasyon Sayısı	$D_{sınır}$ Değeri	D^{-} Değeri	Rastgele Sayı	Çözüm Süresi (milisaniye)	Toplam Mesafe (km)
601	24 000 000	100	1	5	71 498	3 900,734
602	24 000 000	100	1	26	70 301	3 903,313
603	24 000 000	100	1	38	71 685	3 935,585
604	24 000 000	100	1	47	70 447	3 981,534
605	24 000 000	100	1	71	70 196	4 000,528
606	24 000 000	100	1	89	70 537	3 966,251
607	24 000 000	100	1	101	69 721	3 901,934
608	24 000 000	100	1	118	70 413	3 943,967
609	24 000 000	100	1	128	70 150	3 910,175
610	24 000 000	100	1	136	71 453	3 905,651
611	24 000 000	100	1	149	70 434	3 924,230
612	24 000 000	100	1	157	70 760	3 986,234
613	24 000 000	100	1	168	71 315	3 899,666
614	24 000 000	100	1	179	70 216	3 979,710
615	24 000 000	100	1	211	77 646	3 971,081
616	24 000 000	100	1	218	70 018	3 970,972
617	24 000 000	100	1	236	70 893	3 901,765
618	24 000 000	100	1	251	69 427	3 947,050
619	24 000 000	100	1	257	70 059	3 955,859
620	24 000 000	100	1	280	70 440	4 004,272
621	24 000 000	100	1	293	70 488	3 974,540
622	24 000 000	100	1	313	70 988	3 942,388
623	24 000 000	100	1	336	70 990	3 912,315
624	24 000 000	100	1	352	70 277	3 879,556
625	24 000 000	100	1	379	70 559	3 891,868
626	24 000 000	100	1	381	69 832	3 941,060
627	24 000 000	100	1	409	71 124	3 892,029
628	24 000 000	100	1	417	70 508	3 968,377
629	24 000 000	100	1	438	70 405	3 959,229
630	24 000 000	100	1	461	71 323	3 982,474
631	24 000 000	100	1	484	70 742	4 022,422
632	24 000 000	100	1	503	81 870	3 870,180
633	24 000 000	100	1	547	98 017	3 931,601
634	24 000 000	100	1	560	97 994	3 892,227
635	24 000 000	100	1	587	99 580	3 919,202
636	24 000 000	100	1	607	101 308	3 911,312
637	24 000 000	100	1	619	100 779	3 928,340
638	24 000 000	100	1	653	100 104	3 927,170
639	24 000 000	100	1	680	97 797	3 967,139
640	24 000 000	100	1	701	100 736	3 962,267
Ortalama					76 825,75	3 939,155

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : KESKİN, İbrahim
Uyruğu : T.C.

Eğitim

Derece	Eğitim Birimi	Mezuniyet tarihi
Doktora	Gazi Üniversitesi / Yöneylem Araştırması	Devam ediyor
Yüksek lisans	KHO / Harekât Araştırması	2016
Lisans	KHO / Sistem Mühendisliği	2005
Lise	Maltepe Askeri Lisesi	2001

İş Deneyimi

Yıl	Yer	Görev
-	-	-

Yabancı Dil

İngilizce

Yayınlar

Hobiler

Yüzme, Doğa Yürüyüşü



GAZİLİ OLMAK AYRICALIKTIR…

