

**KARINCA KOLONİSİ OPTİMİZASYONU İLE ARAÇ ROTALAMA
PROBLEMİNİN MALİYETLERİNİN KÜMELEME TEKNİĞİ İLE
İYİLEŞTİRİLMESİ**

KAMİL ÇALIŞKAN

**YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ**

**TOBB EKONOMİ VE TEKNOLOJİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

EYLÜL 2011

ANKARA

Fen Bilimleri Enstitü onayı

Prof. Dr. Ünver KAYNAK
Müdür

Bu tezin Yüksek Lisans derecesinin tüm gereksinimlerini sağladığını onaylarım.

Doç. Dr. Erdoğan DOĞDU
Anabilim Dalı Başkanı

Kamil ÇALIŞKAN tarafından hazırlanan KARINCA KOLONİSİ
OPTİMİZASYONU İLE ARAÇ ROTALAMA PROBLEMİNİN
MALİYETLERİNİN KÜMELEME TEKNİĞİ İLE İYİLEŞTİRİLMESİ adlı bu tezin
Yüksek Lisans tezi olarak uygun olduğunu onaylarım.

Yrd. Doç. Dr. Tansel ÖZYER

Tez Danışmanı

Tez Jüri Üyeleri

Başkan :Yrd. Doç. Dr. Mehmet TAN

Üye :Yrd. Doç. Dr. Tansel ÖZYER

Üye :Doç. Dr. Bülent TAVLI

TEZ BİLDİRİMİ

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

KAMİL ÇALIŞKAN

Üniversitesi : TOBB Ekonomi ve Teknoloji Üniversitesi
Enstitüsü : Fen Bilimleri
Anabilim Dalı : Bilgisayar Mühendisliği
Tez Danışmanı : Yrd. Doç. Dr. Tansel ÖZYER
Tez Türü ve Tarihi : Yüksek Lisans – Eylül 2011

Kamil ÇALIŞKAN

**KARINCA KOLONİSİ OPTİMİZASYONU İLE ARAÇ ROTALAMA
PROBLEMİNİN MALİYETLERİNİN KÜMELEME TEKNİĞİ İLE
İYİLEŞTİRİLMESİ**

ÖZET

Çok depolu araç rotalama problemlerinde müşteri taleplerinin en kısa sürede, en kısa yoldan karşılanması ve taleplerin karşılanması için kullanılacak olan araç sayısının en uygun sayıda seçilmesi oldukça önemlidir.

NP zor bir problem olan araç rotalama probleminin çözümü için pratik yaklaşımlar ve birçok algoritmalar oluşturulmuştur, Bu algoritmalar kesin ve yaklaşımsal olmak üzere genel olarak iki gruba ayrılabilirler. Kesin algoritmaların kötü tarafı düşük performans göstermesidir fakat birçok yaklaşımsal algoritma kombinasyonel problemlerde kesin algoritmaların aksine kısa zaman dilimlerinde yüksek kalitede çözümler ortaya koyarlar. Bilinen bu tekniklerden biriside karınca kolonisi optimizasyonudur. Bu tez çalışmasında araç rotalama problemine karınca kolonisi optimizasyonu k ortalama kümeleme tekniği ile birlikte uygulanarak taleplerin toplam gerçekleşme süresi, toplam mesafe ve kullanılan toplam araç sayısının değişimleri izlenmiştir.

Anahtar Kelimeler: Sürü Zekâsı, Karınca Kolonisi Optimizasyonu, K Ortalama Kümeleme, Araç Rotalama

University : TOBB University of Economics and Technology
Institute : Institute of Natural and Applied Sciences
Science Programme : Computer Engineering
Supervisor : Assistant Professor Tansel ÖZYER
Degree Awarded and Date : M.Sc. – September 2011

Kamil ÇALIŞKAN

**IMPROVING THE COST OF VEHICLE ROUTING PROBLEM BY USING
ANT COLONY OPTIMIZATION WITH CLUSTERING TECHNIQUES**

ABSTRACT

In multi-depot vehicle routing problems meeting the customer demand in the shortest time with the shortest path and appropriate number of vehicles is the major point for the solution of these problems. Many practical approaches and algorithms have been established for the solution of a kind of hard NP vehicle routing problems. These algorithms can be divided into two groups in general, to be precise and approximate. The bad side of the exact algorithms is that they show poor performance. But unlike the exact algorithms, many approximate algorithms reveal high-quality solutions in short time periods for combinatorial problems. One of these high-quality solution techniques is known as ant colony optimization. In this thesis ant colony optimization with k-means clustering technique is applied for the vehicle routing problem. The changes in the total realization of demands in time, total distance and total number of vehicles are observed for the applications.

Keywords: Swarm intelligence, Ant colony optimization, K-means clustering, Vehicle routing

TEŐEKKÜR

Çalıőmalarım boyunca yardım ve katkılarıyla beni yönlendiren deęerli hocam Yrd. Doç. Dr. Tansel ÖZYER'e ve yine kıymetli tecrübelerinden faydalandığım TOBB Ekonomi ve Teknoloji Üniversitesi Bilgisayar Mühendislięi öğretim üyelerine,

Her zaman arkamda olan, beni sürekli destekleyen ve bugünlere gelmemde en büyük emeęe sahip olan deęerli aileme teşekkürlerimi sunarım.

İÇİNDEKİLER

ÖZET	v
ABSTRACT	vi
TEŞEKKÜR	vii
İÇİNDEKİLER	viii
1. GİRİŞ	1
1.1. Araç Rotalama Problemi	2
1.2. Araç Rotalama Problemi Temel Bileşenleri	3
1.3. Araç Rotalama Problemi Genel Matematiksel Modeli	4
1.4. Rotalama Türleri	6
1.4.1. Rota Süresi Sınırsız Rotalama Problemleri	6
1.4.2. Rota Süresi Sınırlı Rotalama Problemleri	6
1.4.3. Zaman Pencereleli Rotalama Problemleri	7
1.4.4. Eşzamanlı Rotalama ve Çizelgeleme Problemleri	8
1.4.5. Kapasiteli Araç Rotalama	8
1.5. Araç Rotalama Yöntemleri	9
1.5.1. En Kısa Yol Yöntemi	9
1.5.2. Süpürme (Sweep) Yöntemi	10
1.5.3. Gezgin Satıcı Problemleri	11
1.5.4. Tasarruf Algoritması	13
1.5.5. Doğrusal Programlama Yardımıyla Araç Rotalama	15
2. KARINCA KOLONİSİ OPTİMİZASYONU	18
2.1. Giriş	18
2.2. Sürü Zekası	19
2.3. Karıncaların Yiyecek Arama Mekanizması	20
2.4. Yapay Karıncaların Özellikleri	21
2.5. Normal Karıncalar ve Yapay Karıncalar'ın Karşılaştırılması	22
2.6. Orginal Karınca Sistemi	23
2.7. Karınca Algoritmaları ve Hesaplamalı Optimizasyon Teknikleri	24
2.7.1. Karınca Kolonisi Optimizasyonu Sezgiseli	25

2.7.2. Max-Min Ant System	26
2.7.3. Karınca Kolonisi Sistemi	26
2.7.4. Sıra Tabanlı Karınca Sistemi	27
2.8. Karınca Algoritmaları Karşılaştırması	28
2.9. Karınca Algoritmalarının Geçmişteki ve Günümüzdeki Uygulamaları	29
2.10. Gelişmiş Karınca Algoritmaları	30
3. KÜMELEME ANALİZİ	32
3.1. Giriş	32
3.2. Kümeleme Analizi	32
3.3. Kümeleme Analizi'nin Kullanım Alanları	34
3.4. Kümeleme Analizi ile Veriler Üzerinde Yapılan İşlemler	35
3.5. Kümeleme Türleri	37
3.5.1. Hiyerarşik(İç içe) Kümeleme ve İç içe olmayan Kümeleme	37
3.5.2. Seçkin, Örtüşen ve Bulanık Kümeleme	38
3.5.3. Tam Kümeleme ve Kısmi Kümeleme	38
3.6. Küme Türleri	38
3.6.1. İyi Ayrılmış Kümeler	39
3.6.2. Örnek Tabanlı Kümeler	39
3.6.3. Çizge Tabanlı Kümeler	39
3.6.4. Yoğunluk Tabanlı Kümeler	40
3.6.5. Paylaşılan Nitelik ve Kavramsal Kümeler	40
3.7. Kümeleme Teknikleri	42
3.7.1. K-Ortalama Kümeleme	42
3.7.2. K-Ortalama İşlem Basamakları	43
3.7.3. Noktaların En Yakın Merkeze Atanması	44
3.8. K-Ortalama Kümelemesi Özellikleri, Avantaj ve Dezavantajları	45
3.9. K-Ortalama Kümelemesi Uygulama Alanları	46
4. YAPILAN İŞLEMLER	47
4.1. Giriş	47
4.2. İşlem Basamakları	47
4.3. Uygulama Ekran Görüntüleri	53
5. UYGULAMA ÇIKTILARI	57

5.1. Depo ve Müşteri Sayısı Sabitken İterasyon Sayısının Arttırılması	57
5.2. Depo ve İterasyon Sayısı Sabitken Müşteri Sayısının Arttırılması	63
5.3. Müşteri ve İterasyon Sayısı Sabitken Depo Sayısının Arttırılması	69
6. SONUÇ	75
6.1. Yorumlar	75
6.2. Gelecek Çalışmalar	77
KAYNAKLAR	79
ÖZGEÇMİŞ	84

ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 2.1. Karınca Algoritmaları Karşılaştırılması	27

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 1.1. Tasarruf Yöntemindeki Müşteri Birleştirilmesi	14
Şekil 2.1. Karıncaların yol bulma işleminin şematik gösterimi (Şekil [27] den alınmıştır.)	20
Şekil 3.1 Aynı Veri Seti Üzerindeki Farklı Küme Sayıları (Şekiller [42] den alınmıştır)	36
Şekil 3.2 İyi ayrılmış ve Merkez Tabanlı Kümeler (Şekiller [42] den alınmıştır)	41
Şekil 3.3. Komşuluk Tabanlı ve Yoğunluk Tabanlı Kümeler (Şekiller [42] den alınmıştır)	41
Şekil 3.4. Kavramsal Kümeler (Şekiller [42] den alınmıştır)	42
Şekil 3.5. K Ortalama işlem basamakları	43
Şekil 3.6. Örnek bir veride k-ortalama algoritması ile kümelerin bulunması (Şekiller [42] den alınmıştır)	45
Şekil 4.1. Uygulama ekranının genel görüntüsü	48
Şekil 4.2. Algoritma işlem basamakları	49
Şekil 4.3. Alt Kümelerin ve alt küme merkezlerinin dağılımı	50
Şekil 4.4. Alt kümelerin alt küme merkezleri etrafında birleştirilmesi	51
Şekil 4.5. Rotalama işleminin ekran görüntüsü	51
Şekil 4.6. Kümelerin alt kümelere ayrılması	53
Şekil 4.7. Alt küme merkezleri arasındaki en kısa mesafelerin hesaplanması	54
Şekil 4.8. Alt kümelerin merkez noktaları etrafında birleştirilmesi	55
Şekil 4.9. Bulunan en kısa yollar üzerinden araç rotalama işleminin gerçekleştirilmesi	56

KISALTMALAR

Kısaltma	Açıklama
VRP	Vehicle Routing Problem – Araç Rotalama Problemi
CVRP	Capacitated Vehicle Routing Problem-Kapasiteli Araç Rotalama Problemi
TSP	Travelling Salesman Problem – Gezgin Satıcı Problemi
GSP	Gezgin Satıcı Problemi
ARP	Araç Rotalama Problemi
CO	Combinational Optimisation-Kombinasyonel Optimizasyon
AS	Ant System – Karınca Sistemi
MMAS	Max-Min Ant System –Max-Min Karınca Sistemi
QAP	Quadratic Assignment Problem

BÖLÜM 1

1. GİRİŞ

Dağıtım sistemleri, müşterilerin sürekli artan beklentileri ve sürekli değişen ürün nitelikleri ile son derece karmaşık bir hale gelmiştir. Bu da bu alanda hizmet veren şirketlerin daha etkili dağıtım planları yapmasını gerektirmektedir. Dağıtım ağı içerisinde bulunan müşterilerin taleplerinin en az maliyet ile karşılanması bu dağıtım planlarının merkezinde yer almaktadır.

Araç rotalama problemi de dağıtım sistemi içerisindeki problemlerin bir örneğidir. Günlük hayatta ürün dağıtımını, banka para transfer araçları, öğrenci servisleri, yakıt dağıtımını ve daha birçok dağıtım işlemi araç rotalama probleminin örnekleri olarak gösterilebilir.

Bu tarz karmaşık problemlerin çözümünde genel olarak kesin ve yaklaşımsal olmak üzere iki tarz çözüm geliştirilmiştir. Kesin algoritmalar en iyi çözümü bulmak ve bunun en iyi çözüm olduğunu ispatlamak üzerine konumlandırılmıştır ancak bu algoritmalar kombinasyonel problemlerde düşük performans göstermiştir. Yaklaşımsal algoritmalar ise kesin algoritmaların aksine kısa zaman dilimlerinde yüksek kalitede çözümler ortaya koymuştur. Bu algoritmalarından biriside karınca kolonisi optimizasyonu algoritmasıdır ve araç rotalama, gezgin satıcı problemi gibi birçok kombinasyonel problemin çözümünde başarılı sonuçlar ortaya koymuştur. Bu tez çalışmasında kapasiteli araç rotalama probleminin karınca kolonisi optimizasyonu ile çözümünde kümeleme tekniği de çözüme dahil edilerek, araç rotalama probleminin kısıtlarının (toplam yol, toplam mesafe ve toplam araç sayısı) değişimi incelenmiştir.

Kümeleme işlemi teorik ve pratik birçok alanda yaygın olarak kullanılmaktadır. Kümeleme ile veriler anlamlı nesne grupları haline dönüştürülmektedirler. Verileri anlamlandırma ve çok büyük verilerden bilgi çıkarsama noktasında kümeler sınıfları gösterir ve kümeleme analizi de olası bu sınıfların otomatik olarak bulunmasına yönelik tekniklerin bütünüdür.

Tezin ilk bölümünde araç rotalama problemi açıklanmış, araç rotalama probleminin temel bileşenleri, matematiksel modeli, rotalama türleri ve araç rotalama yöntemleri anlatılmıştır. İkinci bölümde karınca kolonisi optimizasyonu, normal ve yapay karıncalar, sürü zekası, karınca kolonisi algoritmaları, karınca algoritmalarının geçmişte ve günümüzdeki uygulamaları anlatılmıştır. Üçüncü bölümde kümeleme analizi, kullanım alanları, küme ve kümeleme türleri, kümeleme teknikleri, k ortalama kümelemenin avantaj ve dezavantajları anlatılmıştır. Dördüncü bölümde yapılan işlemler, uygulama işlem basamakları anlatılmıştır. Beşinci bölümde uygulama çıktıları ve grafiklere yer verilmiştir. Altıncı bölümde ise elde edilen çıktılar yorumlanmış ve gelecek çalışmalar anlatılmıştır.

1.1. Araç Rotalama Problemi

Araç Rotalama Problemi, coğrafi olarak dağınık müşterilere bir veya birden fazla depodan hizmet için görevlendirilen araçların en ideal şekilde dağıtım ve toplama rotalarının tasarlanması problemine denir. [1] ARP ilk olarak 1959 yılında Dantzig ve Ramser tarafından bulunan ve üzerinde en fazla optimizasyon geliştirilen yöntemdir.[6]

Standart bir araç rotalama probleminde depolardan araçlar ile farklı noktalardaki müşterilerin talepleri karşılanmaya çalışılmaktadır. Bu işlemi gerçekleştirirken amaç verimli bir şekilde müşteri ihtiyaçlarını mümkün olan en kısa zamanda, en kısa yoldan ve en az maliyetle karşılayan rotayı belirlemektir. Araç rotalama yapılırken aşağıdaki unsurlara dikkat edilmelidir:

- Şebeke içersinde bulunan müşterilerin talepleri tamamıyla karşılanmalıdır.
- Şebekede bulunan her varış noktası tek bir araç tarafından sadece bir kez ziyaret edilmelidir.
- Rota depodan başlamalı ve tekrar depoda sonlanmalıdır.
- Rota üzerinde bulunan müşterilerin toplam talep miktarı aracın toplam kapasitesinden fazla olmamalıdır.
- Her bir araç sadece bir rota üzerinde faaliyet göstermelidir.

- Araç rotalamanın temel amacı araçların kat edecekleri toplam mesafenin minimize edilmesi olmalıdır.[5]

1.2. Araç Rotalama Problemi Temel Bileşenleri

Araç rotalama problemlerinin temel bileşenlerini; talep yapısı, taşınacak malzemenin tipi, dağıtım/ toplama noktaları ve araç filosu oluşturur.

1-Talep Yapısı: Araç rotalama problemlerinde talep statik veya dinamik olabilir. Statik talep durumunda talep önceden bilinir. Dinamik durumda ise bazı düğümlerdeki talep bilinmekte bazıları ise araç rotasında devam ederken belirli olmaktadır. Bu çalışmada statik talep yapısı kullanılmıştır.

2-Malzeme Tipi: Araçlarla çok çeşitli malzemeler taşınır. Tehlikeli maddeler, gıda maddeleri, gazete dağıtımı, çöp toplama bütün bunlar basit paketler olarak adlandırılır ve problem ilave bir karmaşıklık getirmezler. Diğer taraftan öğrenci servisleri; güvenlik, etkinlik, eşitlik gibi ilave bazı amaçlardan ötürü daha karmaşık bir yapıya sahiptir. Tehlikeli maddeleri taşıyan araçların rotalarının belirlenmesinde ise coğrafi özellikler büyük önem kazanır.[7]

3-Dağıtım/Toplama noktaları: Birçok araç rotalama probleminde, dağıtım noktaları müşterilerin bulunduğu yer, toplama noktaları ise depodur. Tüketim mallarının fabrikalardan toptancılara dağıtımına buna iyi bir örnektir.

Depo genellikle aracın rotasına başladığı ve geri döndüğü noktadır. Depo sayısına göre problem, tek depolu ve çok depolu diye adlandırılabilir. Çok depolu problemlerde, depoların her biri kendi araçlarıyla işlerini yürütebilir, bu durumda problem birkaç bağımsız tek depolu araç rotalama problemine dönüşür. Araç bir depodan çıkıp, başka bir depoda yükleme/boşaltma yapabilir. Bu durumda problem bir bütün olarak ele alınmalıdır.

Dağıtım noktaları sabit ve önceden biliniyorsa hangi noktalara, hangi araçların hizmet vereceği belirlenmelidir. Diğer durumda dağıtım noktaları potansiyel yerler arasından seçileceği için ilave bir yerleştirme kararı gerekir.

Bazı araç rotalama problemlerinde dağıtım ve toplama noktaları aynıdır. Örneğin öğrenci servislerinde okul, gidişte dağıtım noktası, duraklar toplama noktası; öğrenciler evlerine dönerken ise okul depo, duraklar ise dağıtım noktalarıdır. [7]

4- **Araç Filosu:** Bütün araç rotalama problemlerinde araçların kapasitesinin bilindiği ve çoğunlukla araçların homojen (aynı kapasitede) olduğu varsayılır. Filo heterojen ise filodaki araçların taşıma kapasiteleri farklıdır. Bu durum hangi araç tipinin, hangi rotaya hizmet vereceğinin belirlenmesini, yani ilave bir kararı gerektirir. Araçların diğer özellikleri arasında hız, yakıt tüketimi, taşınacak malzemeye uygunluğu sayabiliriz. Bu özelliklerin rotalama kararlarına doğrudan etkisi yoktur.

1.3. Araç Rotalama Problemi Genel Matematiksel Modeli

Standard Araç Rotalama Probleminin formülasyonu şu şekilde belirtilir. [3]

Parametreler:

Q_k : araç kapasitesi,

N : müşteri veya durak sayısı,

q_i : i ($i > 0$)müşterisinin talebi,

d_{ij} : i ve j müşterileri arasındaki uzaklık,

Değişkenler:

x_{ij} , $i \neq j$: araç i " den j " ye gidiyorsa "1", yoksa "0" olmak üzere,

($i, j \in \{ 0, \dots, n \}$ ve "0" başlangıç deposu iken)

Amaç Fonksiyonu:

$$\text{Min} \sum_{i=0}^n \sum_{\substack{j=0, \\ i \neq j}}^n d_{ij} x_{ij} \quad (1.1)$$

Kısıtlar:

$$\sum_{\substack{i=1 \\ i \neq j}}^n x_{ij} = 1 \quad \forall_{i,j} \in \{1, \dots, n\} \quad (1.2)$$

$$\sum_{\substack{i=1 \\ i \neq j}}^n x_{ij} = 1 \quad \forall_{i,j} \in \{1, \dots, n\} \quad (1.3)$$

$$\sum_{i=0}^n \sum_{\substack{j=0 \\ j \in S}}^n x_{ij} \leq |S| - 1 \quad (1.4)$$

$$\sum_{i=0}^n \sum_{\substack{j=0 \\ j \in T}}^n x_{ij} \leq |T| - n \quad (1.5)$$

Burada (1.2) ve (1.3) kısıtları ziyaret edilen müşteriden ayrılma kısıtlarıdır. (1.4) kısıtı depodan başlamayan ve depoda bitmeyen turları elemekte kullanılır.

Son olarak, (1.5) kısıdı araçlardaki yük durumunu kontrol etmektedir. Bu kısıt, depo dahil olmak üzere her T müşteri kümesine eklenir. Bu kümelerin her biri,

$$\sum_{i \in T} q_i < Q_k$$

Şartını sağlamaktadır.

Ayrıca n ise aşırı yüklemeyi engellemek için T kümesinden çıkarılması gereken minimum sayıda müşteri sayısıdır [3].

1.4. Rotalama Türleri

1.4.1. Rota Süresi Sınırsız Rotalama Problemleri

Rota süresi, depodan ayrılan bir aracın depoya dönmesi için gereken süre olarak tanımlanabilir. Rota süresi sınırsız rotalama problemlerinin belki de en iyi örneği, tek araçlı gezgin satıcı problemleridir. Bu problem, tek bir araç ile bir "Hamiltonion tur" oluşturulması esasına dayanmaktadır. Bir Hamiltonion tur, bir $G = [F, A]$ ağında, her bir düğümden kesinlikle bir kez geçen turu göstermektedir. Bu ağda, (F) , düğümler kümesidir; (A) kollar (adresleri birbirine bağlayan yollar) kümesini göstermektedir. Problemin amacı, toplam tur uzunluğunu ya da maliyetini en aza indirmektir. Bu problemde, adreslere uğranılacak zamanlar ve rota süresi ile ilgili girdiler ve kısıtlar yoktur. [8]

1.4.2. Rota Süresi Sınırlı Rotalama Problemleri

Bu problem türünde, araçların kat ettikleri toplam uzunluk en aza indirgenmektedir. Depodan ayrılan araçlar belirli adreslere (düğümlere) uğradıktan sonra aynı depoya dönmek zorundadırlar. Bir aracın depodan depoya olan rota süresi, en büyük rota süresini aşmamalıdır. Bu kısıt, adresler kümesine hizmet verecek olan araç sayısının belirlenmesinde önemli bir rol oynar.

Rota süresi ile birlikte araç sayısının da sınırlı olduğu bir problemin çözümünde, bazı adreslere uğranılamaması durumu ile karşılaşılabilir. Eğer bütün adreslere mutlaka uğranılması gerekiyorsa ve araç sayısını arttırmak olanaksızsa, en büyük rota süresi aşılacak zorunda kalınabilir. Bu durumda aşılacak her birim süre için bir ceza maliyeti verilerek, toplam ceza maliyeti en aza indirgenebilir. Böyle bir yaklaşım araç sayısının sınırlı olmasını bir "katı kısıt", rota süresinin sınırlı olmasını

bir "gevşek kısıt" yapar. Bir "katı kısıt" kesinlikle uyulması gereken, bir "gevşek kısıt" ise uyulması istenilen kısıtlardır. Görüldüğü üzere problemin yapısı, rota süresinin sınırlandırılmasından önemli ölçüde etkilenmektedir.[8]

1.4.3. Zaman Pencereli Rotalama Problemleri

Zaman pencereli araç rotalama problemleri, en küçük maliyetli rota kümelerinin tasarımı, merkez bir depo yaratmayı ve bilinen belirli bir talebe bağlı filo rotalamayı kapsamaktadır. İlk etapta, yolcular, kapasiteler göz önünde bulundurulmadan araçlara görevlendirilmektedir.

Zaman penceresi, sert veya yumuşak olabilmektedir. Sert bazlı vakalarda, eğer bir araç müşteriye erken bir zamanda ulaşırsa, araç müşteri gelinceye kadar beklemelidir. Ancak, aracın bir düğüme, hizmete başlaması için gereken en geç zamandan (latest time) sonra varılmasına izin verilmez. Tersine, yumuşak vaka bazlı çalışmalarda, zaman bazları bir maliyetle bozulabilmektedir.[9]

Zaman bazı, genellikle doğal olarak esnek zaman çizelgeleriyle çalışan organizasyonlarda karşılaşılan bir durumdur. Sert zaman bazlı spesifik örnekler; banka sevkiyatları, posta iletileri, okul servisleri rotalama ve çizelgeleme şeklindedir.[9]

Zaman pencereleri iki sınıfta incelenebilir:

- Çift sınırlı zaman pencereleri,
- Tek sınırlı zaman pencereleri

Çift sınırlı bir zaman penceresinde, adrese uğranabilecek en erken zaman ve en geç zaman belirlidir. Bir araç bir adrese zaman sınırları içinde uğramalıdır. Zamanından önce adrese erişen bir aracın, bu zamana kadar bekleyeceği kabul edilir. Aracın, sınırından sonra adrese uğramasına izin verilmez. Tek sınırlı zaman pencereleri, uğranılabilecek "ilk zamanı belirli olanlar" ve "son zamanı belirli olanlar" şeklinde iki türde ifade edilmektedir.[8]

1.4.4. Eşzamanlı Rotalama ve Çizelgeleme Problemleri

Bir toplu taşıma seferi başlangıç zamanı /yeri ve bitiş zamanı/yeri ile tanımlanabilir. Her bir seferin, filodaki hangi araç ile yapılacağıın belirlenmesi işlemine, "eşzamanlı rotalama ve çizelgeleme" adı verilmektedir. Eşzamanlı araç rotalama ve çizelgeleme, her bir aracın gerçekleştireceği görevlerin "sıralamasını" veren bir optimizasyon problemi olarak ele alınmaktadır. [8]

Bir eşzamanlı rotalama ve çizelgeleme probleminde, en yaygın olarak dikkate alınan kısıtlar şunlardır:

- Her bir görev, belirli bir zaman periyodunda tek bir araç ile ve sadece bir kez gerçekleştirilmelidir.
- Bir depodan ayrılan her bir araç, görevlerini gerçekleştirdikten sonra ait olduğu depoya geri dönmelidir.
- Depodaki veya depolardaki araçlar ile görevlerin tümü yerine getirilmelidir.
- Görevlere, kendileriyle özelleştirilen araç tipleri ile hizmet verilmelidir.
- Her bir araç, belirli bir süre sonunda depoya dönmelidir.

Problemler, burada verilen kısıtların tümünü ya da bazılarını içerebilir. Ayrıca, problemin yapısı değişik kısıtların dikkate alınmasını gerektirebilir. Bu kısıtların bazıları katı, bazıları ise gevşek kısıtlar olabilir.[8]

1.4.5. Kapasiteli Araç Rotalama

ARP'nin en genel versiyonu (Kapasiteli Araç Rotalama Problemi (KARP)) olan bu problem; merkezi bir depoda teslimat işini gerçekleştiren k tane K kapasiteli V hızına sahip araçların o coğrafi alanda düzgün bir şekilde farklı noktalara dağılmış n tane müşterinin taleplerini en az M maliyetle karşılaması için kullanması gereken en uygun R rotalarının bulunması olarak tanımlanır. Müşteriler arasındaki mesafeler karşılıklı gidiş-geliş şeklinde simetrik olarak varsayılmaktadır. Tüm araçların talepleri karşılamak için kullanılması durumunda k tane rota bulunur. Yolların açık

ya da kapalı olması durumu göz önünde bulundurulmak istenirse kapalı olan yolları ifade eden kenarlar pasif duruma getirilerek o anki duruma göre rotalar belirlenebilir. Ayrıca yollar her ne kadar çift yönlü olduğu kabul edilse de tek yönlü olarak da ayarlanabilir.[10]

1.5. Araç Rotalama Yöntemleri

Araç rotalama problemlerinin çözümü değişen çevre faktörleri, artan rekabet koşulları ile birlikte gelen kısıtlarla giderek zorlaşmaktadır. Zaman aralıkları, farklı kapasitelere sahip çok sayıda araç, rota üzerinde müsaade edilen seyahat zamanı, araç sürücülerinin dinlenme zamanları araç rotalama problemlerinde karşımıza çıkan kısıtlar içinde yer almaktadır. [8]

Bir şebeke için araç rotalama problemi o probleme özel tasarlanan bir yöntem ile çözülebilir. En sık kullanılan çözüm yöntemlerinden birisi de en kısa yol yöntemidir. Ayrıca sıkça kullanılan diğer yöntemler süpürge metodu, gezgin satıcı yöntemi ve tasarruf metodudur. Bu yöntemlerin her biri kendine özgü amaçlar ve kısıtlar taşımaktadır. Ayrıca karmaşık yapılı, çoklu orijin ve varış noktalarına sahip araç rotalama problemleri doğrusal programlama metodu ile çözülmektedir.

1.5.1. En Kısa Yol Yöntemi

En kısa yol yönteminin temel amacı müşteri noktaları arasındaki yakınlık ve uzaklık durumu ve araç yükleme kapasitelerine göre araçlara müşteri atanmasıdır. En kısa yol yöntemi süreç aşamaları aşağıdaki gibidir (Breedam, 2002):

1. Merkezi birimden rotalamaya başlanır. İlk araca merkezi noktaya en yakın müşteri atanır.
2. Rotaya atanan müşteriye en yakın, daha önce rotaya eklenmemiş noktalar incelenir. Eğer müşteriye en yakın iki nokta varsa her biri için süreç oluşturularak ayrı çözüm dalları yaratılır.

3. Eğer müşteri direkt olarak merkezi birim ile bağlantılı değilse (Oluşturulan rotada müşteri merkezi birimden sonra gelmiyorsa) ve müşteriye en yakın başka müşteri ile merkezi birim aynı mesafede ise süreç yine ikiye ayrılarak yeni çözümler oluşturulur.

Rotalamada ilk önce müşteri başka müşteriye araç kapasite kısıtı sağlıyorsa bağlanır. Ardından çözüm ağacında yeni bir dal oluşturularak müşteri direkt olarak merkezi birime bağlanır.

4. Çözümler ayrı ayrı hesaplanarak en uygun çözüm seçilir.

Ancak bir orijin noktasından çok sayıda varış noktasına ya da çok sayıda orijin noktasından çok sayıda varış noktasına en kısa mesafenin bulunması durumu söz konusu olduğu zaman yapılan işlemler daha da karmaşık hale gelmektedir. Bu gibi durumlarda doğrusal programlama yardımıyla çözüm yolu bulunmaktadır.[11]

1.5.2. Süpürme (Sweep) Yöntemi

Gilbert ve Miller tarafından 1974 yılında geliştirilen Süpürme algoritmasında rotalarda yer alacak müşteriler, depo merkezli bir doğrunun döndürülmesi ile elde edilirler. Döndürme esnasında doğrunun üzerinden geçtiği müşteriler bir gruba ayrılır ve kapasite veya mesafe kısıtı aşıldığında grup kapatılarak yeni bir grup ile devam edilir. Oluşturulan nokta gruplarına merkez depo da eklenerek, genel olarak GSP gibi çözümlere rotalar belirlenir.[12]

Süpürme Yönteminde müşterilerin koordinatları Öklidyen forma (x, y) yerine, θ açı ve ρ doğru uzunluğu olmak üzere polar formatta (θ_i, ρ_i) tutulur. Böylece talep noktaları θ açısına göre küçükten büyüğe sıralanarak, kapasite ve mesafe kısıtları dikkate alınarak gruplanır. Genel olarak Süpürme Algoritmasının çözüm adımları aşağıdaki belirtilmektedir.[13]

1- Bir harita üzerinde depo ve müşteri noktalarının yeri tespit edilir ve koordinatlar polar formata (θ_i, ρ_i) çevrilir. Rotaya atanmamış herhangi bir araç belirlenir.

2- Depodan yatayla 0° açı ile başlanarak saat yönünün tersine doğru taranmaya başlanır. Eğer bir müşteri ile karşılaşılırsa ve eğer müşterinin talep miktarı aracın

kapasitesini geçmiyorsa müşteri araca atanır. Aksi durumda saat yönünün tersi yönde hareket edilir. Eğer araca her iki yönde de müşteri atanamıyorsa, diğer araca geçilir.

3- Eski aracın kaldığı yerden taramaya devam edilir. Eğer daha önce rotalanmayan müşteri ile karşılaşırsa ve müşterinin araca atanması kapasite kısıtının aşılmasına sebep olmuyorsa, müşteri araca atanır. Bu süreç açıkta bir talep noktası kalmayınca kadar (tüm noktalar rotalanıncaya kadar) sürer.

4- Tüm noktalar araçlara atandıktan sonra gruplar uygun bir şekilde optimize edilerek rotalar belirlenir.

Süpürme yöntemi Tasarruf algoritmasına nazaran uygulanması daha basit olmasına rağmen, çoğu ARP için kesinlik ve hız bakımından Tasarruf metodundan geridedir. Bunun sebebi gruplama yapılırken noktalar arasındaki uzaklıkların dikkate alınmaması ve rotaların belirli bir alandan başlayarak oluşturulmasıdır. Ayrıca bu tekniğin yeni kısıtlarla entegre olması zor olmakla birlikte sadece iki boyutlu Öklidyen sistemde çözümler üretmesi uygulanabilirliğini kısıtlamaktadır.[12]

1.5.3. Gezgin Satıcı Problemleri

Bir satıcının n adet şehri en kısa zamanda dolanımını sağlayacak şekilde şehirleri ziyaret etme sırasını bulma problemine gezgin satıcı problemi denir. Gezgin satıcı probleminde her bir şehir yalnız bir defa en kısa yol takip edilerek ziyaret edilir ve tekrar başlangıç noktasına geri dönülür.

Gezgin satıcı problemi yöneylem araştırmalarında ve teorik bilgisayar bilimlerinde çalışılmıştır, kombinasyonel optimizasyon bir NP-Zor bir problemdir. Şehirler ve ikili mesafelerin bir listesi verildiğinde, görev tam olarak bir kere her şehirde ziyaret eden en kısa turu bulmaktır.

Sorun ilk olarak 1930 yılında bir matematik problemi olarak formüle edilmiş ve optimizasyon konusunda en çok çalışılan problemlerden biri olmuştur. Problem hesaplama olarak zor olsa da birçok sezgisel ya da kesin algoritma ile probleme çözümler üretilmiştir.

Hesaplama karmaşıklığı ise Gezgin satıcı problemi ve Araç Rotalama probleminde şehir sayısı arttıkça problemin zorluğu süratle artar. n şehirli bir problemde olurlu turların sayısı $(n-1)!$ 'dir. İlk şehir için $(n-1)$ tane, ikinci şehir için $(n-2)$ tane seçenek vardır.

Çoğu GSP sezgisellerinden ARP için geliştirilen sezgiseller oluşturulmuştur. Gezgin satıcı sezgiselleri; tur kurucu algoritmalar, tur geliştirici algoritmalar ve karışık algoritmalar olmak üzere üç tipte belirtilir. Tur kurucu algoritmalar, mesafe matrisinden optimale yakın turlar oluştururlar. Tur geliştirici algoritmalar mevcut bir mümkün çözümü geliştirirler. Karışık algoritmalar ise tur kurucu algoritmalarından biri ile bir başlangıç çözümü bulup, tur geliştirici algoritmaların bir veya birkaçı ile mevcut çözümü geliştirirler.[7]

GSP'nin matematiksel ifadeleri aşağıdaki gibidir.

$$\min \sum_{i=1}^n \sum_{j=1}^n d_{ij} \cdot x_{ij} \quad (1.6)$$

$$\sum_{i=1}^n x_{ij} = 1 \quad j = \{1, \dots, n\} \quad (1.7)$$

$$\sum_{j=1}^n x_{ij} = 1 \quad i = \{1, \dots, n\} \quad (1.8)$$

$$x = (x_{ij}) \in S \quad (1.9)$$

$$x_{ij} = 0 \text{ veya } 1 \quad j = \{1, \dots, n\} \quad i = \{1, \dots, n\} \quad (1.10)$$

n : Dağıtım ağı içindeki düğüm sayısı

d_{ij} : (i, j) yayının uzunluğu

S : Alt tur eliminasyon kısıtı

x_{ij} : $\begin{cases} 1, \text{ eğer } (i, j) \text{ yayı turun içinde ise} \\ 0 \text{ aksi takdirde} \end{cases}$

Eğer yalnız ilk iki küme kısıtı hesaba katılırsa problem bilinen atama problemine dönüşmektedir. Esas zorluk oluşturan kısıt üçüncü kısıttır. Bu kısıt, atama kısıtlarını yeterli kılan çözümlerini engellemek için kullanılan alt tur eliminasyon kısıtından meydana gelmektedir.

$$S = (x_{ij}) : \sum_{i \in T} \sum_{j \in T} x_{ij} \geq 1 \quad (V' \text{ nin boş olmayan her bir } T \text{ alt kümesi için}) \quad (1.11)$$

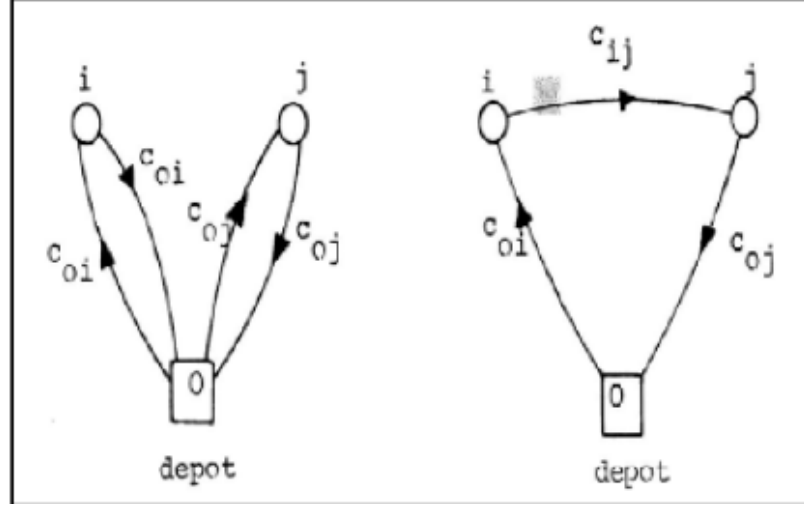
V : Düğüm Seti

Alt tur eliminasyon kısıtlarının ilk seti, düğümlerin her bir uygun alt kümesinin, aynı alt kümenin tamamlayıcısı ile ilişkili olduğunu garanti etmektedir. Bu da, sistemin diğer düğümleri işe ilişkili olmayan herhangi bir uygun alt kümenin, diğer düğümlerden bağımsız olarak bir alt tur oluşturmasını engeller.[14]

1.5.4. Tasarruf Algoritması

ARP problemlerini çözmek için geliştirilen yöntemlerden birisi, Clarke ve Wright tarafından 1964 yılında geliştirilen ve belki de bilinen en iyi tur oluşturma sezgiseli olan Tasarruf yöntemidir. Bu yöntem, her bir müşteri ikilisi arasındaki maliyet tasarrufunu hesaplayarak başlar. Maliyet tasarrufları hesaplanarak iki müşteri arasında

bir müşteri eklenir. Şekil –1.1’de görüldüğü gibi i ve j. müşteri ayrı turlardadır, i. müşteriden sonra j. müşteri eklenerek turlar birleştirilir.



Şekil 1.1. Tasarruf Yöntemindeki Müşteri Birleştirilmesi

$$s_{ij} = (c_{0i} + c_{i0} + c_{0j} + c_{j0}) - (c_{0i} + c_{ij} + c_{j0}) \quad (1.12)$$

$$s_{ij} = c_{i0} + c_{0j} - c_{ij} \quad (1.13)$$

Denklem (1.13)'deki tasarruf miktarı (s_{ij}) müşteri i ve j. müşterinin ayrı turlarda değil aynı turda hizmet almasından kaynaklanan bir maliyet tasarrufudur. Bu maliyet tasarrufu iki bağımsız turun birleştirilmesi ile ortaya çıkmaktadır. Her zaman tasarruf yönteminde, en büyük tasarrufu sağlayan (i,j) ikilisi, müşteri talebi ve araç kapasitesi kısıtları dikkate alınarak seçilir. Bütün müşterilerin araçlara atanmasına kadar bu işlem tekrarlanır.[15]

Clarke-Wright'ın tasarruf yöntemi, kolay anlaşılabilirliği ve diğer ARP yöntemlerine göre esnek olması sayesinde geniş bir kullanım alanına sahiptir. Bu yöntem Gaskell (1967) ve Yellow (1970) gibi pek çok araştırmacı tarafından günümüze kadar uzanan zaman diliminde geliştirilmiştir.[16]

1.5.5. Doğrusal Programlama Yardımıyla Araç Rotalama

Karışık yapılı şebekelerde araç rotalama işlemi için yukarıda belirtilen rotalama yöntemleri yetersiz kalmaktadır. Bu nedenle de araç rotalamanın temel prensibi olan toplam seyahat mesafesinin minimize edilmesi mümkün olmamakta ve dolayısıyla yüksek seviyede maliyetler oluşmakta, servis seviyesi düşmektedir. Bu gibi durumlarda araç rotalamada optimal bir sonuca ulaşmak için doğrusal programlama kullanılmaktadır.

Doğrusal programlama yardımıyla araç rotalamada kısıtlar müşterilerden, araçlardan, depodan ve müşteriler ile depo arasındaki şebeke bağlantılarından oluşur. $N+1$ adet müşteri noktası ve K adet araç bulunmaktadır. Depo 0. müşteri olarak ifade edilir. Araçlar rotaya depodan başlar, müşteri noktalarını ziyaret eder ve tekrar depoya geri dönerler. Şebeke içersindeki rotaların sayısı kullanılan araçların sayısına eşittir. Bir araç sadece bir rota üzerinde faaliyet gösterebilir. Rotalama yaparken şebeke içersindeki her bir müşteri araçlardan sadece biri tarafından ziyaret edilebilir. Tüm araçlar aynı kapasiteye sahiptir ve müşteri talepleri değişebilmektedir. Aracın kapasitesi rota üzerinde bulunan tüm müşterilerin taleplerinin toplamından küçük ya da eşit olmalıdır. Bunun anlamı araçların kapasitelerinin üzerinde yüklenememesidir. Doğrusal programlama yardımıyla bir araç rotalama problemi şu şekilde formüle edilebilir:

Parametreler:

K : Toplam araç sayısı

N : Toplam müşteri sayısı

C_{ij} : i noktasından j noktasına taşıma maliyeti

M_i : i müşteri noktasındaki talep

q_k : k aracının kapasitesi

Değişkenler:

x_{ij} : eğer k aracı i noktasından j noktasına giderse 1, aksi halde 0

$$\min \sum_{i=0}^N \sum_{j=0, j \neq i}^N \sum_{k=1}^K C_{ij} \cdot x_{ijk} \quad (1.14)$$

$$\sum_{k=1}^K \sum_{j=1}^N x_{ijk} \leq K \quad i = 0 \text{ için} \quad (1.15)$$

$$\sum_{j=1}^N x_{ijk} = \sum_{j=1}^N x_{ijk} \leq 1 \quad i = 0 \text{ ve } k \in \{1, \dots, K\} \quad (1.16)$$

$$\sum_{k=1}^K \sum_{j=0, j \neq i}^N x_{ijk} = 1 \quad i \in \{1, \dots, N\} \quad (1.17)$$

$$\sum_{k=1}^K \sum_{i=0, j \neq i}^N x_{ijk} = 1 \quad j \in \{1, \dots, N\} \quad (1.18)$$

$$\sum_{i=1}^N M_i \sum_{j=0, j \neq i}^N x_{ijk} \leq q_k \quad k \in \{1, \dots, K\} \quad (1.19)$$

$$x_{ijk} \in \{0,1\} \quad (1.20)$$

Yukarıdaki doğrusal programlama modelinde (1.8) modelin amacını yani toplam seyahat mesafesinin ve toplam rotalama maliyetinin minimize edilmesini ifade etmektedir. (1.9) oluşabilecek maksimum K rota sayısını ifade etmektedir. (1.10) her bir rotanın depodan başlayıp yine depoda bittiğini göstermektedir. (1.11) müşteri noktalarının her birinin sadece bir araç tarafından yalnızca bir defa ziyaret edildiğini göstermektedir. (1.14) üzerinde bulunan müşterilerin toplam talep miktarının rotaya çıkacak olan aracın toplam kapasitesinden fazla olmayacağını ifade etmektedir. (1.15) X_{ijk} değişkeninin 0 ya da 1 değerini alabileceğini göstermektedir.[17]

Dağıtım sistemlerinde karşılaşılan en önemli problemlerden biri dağıtım faaliyeti için belli sayıda noktaya uğraması gereken araçların etkin ve verimli bir şekilde kullanılmasıdır. Araç rotalama olarak bilinen bu tür problemlerde araç kapasitesi kısıtı altında toplam seyahat mesafesini ve dolayısıyla da toplam maliyeti minimize edecek uygun rotayı belirlemek gerekmektedir. Araç rotalama problemi uzun yıllar boyunca hem araştırmacıların hem de uygulamacıların ilgisini çekmiştir. Bunun nedeni problemin zor ve aynı zamanda uygulama alanının geniş olmasıdır.[5]

BÖLÜM 2

2. KARINCA KOLONİSİ OPTİMİZASYONU

2.1. Giriş

Karınca kolonisi optimizasyonu doğadaki gerçek karınca kolonilerinden etkilenilerek ortaya çıkarılmış meta sezgisel bir metottur. Karmaşık kombinasyonel problemlerin çözümünde kullanılır. Karmaşık kombinasyonel problemlere günlük hayatta birçok alanda karşılaşılabılır. Ekonomi, ticaret, endüstri, mühendislik kimya bu alanlardan bazıları olarak sayılabilir. Fakat bu problemlerin uygulamalardaki çözümü çok zordur. Bu tür problemler bilişim alanında “NP Zor” problemler olarak bilinirler yani bu tür problemleri polinomsal zamanda çözebilen bir algoritmanın olmadığını gösterir.

Bu tarz problemlerin çözümünde birçok pratik yaklaşımlar ve birçok algoritmalar oluşturulmuştur, Bu algoritmalar kesin ve yaklaşımsal olmak üzere genel olarak iki gruba ayrılabilirler. Kesin algoritmalar problemin en iyi çözümünü bulmak ve bunun en iyi çözüm olduğunu ispatlamak üzerine konumlandırılmıştır. Bu algoritmalar backtracking, branch and bound, dynamic programming gibi örneklendirilebilir. Bu kesin algoritmaların kötü tarafı kullanılan tekniklerden de anlaşılacağı gibi birçok problemde düşük performans göstermesidir.

Fakat birçok yaklaşımsal algoritma kombinasyonel problemlerde kesin algoritmaların aksine kısa zaman dilimlerinde yüksek kalitede çözümler ortaya koyarlar. Optimizasyon problemleri ile bilim alanında olduğu gibi sanayinin birçok alanında da karşılaşılabılır. Örnek olarak lojistik trafik yönlendirme örnek olarak verilebilir; burada amaç verimliliği arttırmak, araç kümelerinin müşterilerin taleplerini optimum düzeyde karşılamalarını sağlamaktır. Biyoloji alanında kendi döngüsünde oluşan proteinlerin yapısının belirlenmesi de önemli bir problemidir. Buna benzer problemler farklı alanlarda çoğaltılabilir. İşte optimizasyon bu tarz problemlerin çözümünde geniş kapsamlı araştırmaları ve çözüm algoritmalarını içerir.

Kombinasyonel Optimizasyon(CO) Problemleri[18] optimizasyon problemlerinin özel bir alt kümesidir. Bu problemde problemdeki nesnelere bir maliyet değeri atanır amaç fonksiyonu ise arama alanında en az maliyetle nesnelere ulaşmaktır.

Kombinasyonel Optimizasyon problemlerinde optimum çözümleri garanti eden algoritmalar mevcuttur fakat bu algoritmalar genellikle çok büyük hesaplama maliyetlerine sahiptir. Özellikle NP-Zor Optimizasyon problemlerinde optimum çözümleri bulmak için geçen süre oldukça fazladır. Bu yüzden bu tarz problemlerin çözümünde yaklaşımsal algoritmalar kullanılır. Bu algoritmalar önemli ölçüde daha kısa sürede doğru sonuçlar sunar.

Karınca algoritmaları optimizasyon alanındaki son gelişmeleri yansıtır. Bu algoritmalar doğadaki vahşi karıncalardan ilham alır.[23] Bu algoritma Koloni içindeki karıncalar arasında feromon salgılanması ile dolaylı iletişimin sağlanması işlemidir.

2.2. Sürü Zekası

Karınca algoritmaları yapay zekâ sistemi içerisinde sürü zekâsı içerisinde yer alır. [20] Sürü zekâsı özerk yapıdaki basit bireyler grubunun kolektif bir zekâ geliştirmesidir. Bu ise “stigmergy” [21] (dolaylı iletişim olarak ta bilinir) yani ortamdaki etmenlerin ortama müdahale ederek iletişim kurmaları ve birbirlerinin hareketlerini düzenlemeleri ve “Self-Organization”(kendinden organizasyon) denen iki mekanizma üzerine kuruludur. Stigmergy vasıtasıyla iletişim, bireyler yaptıkları işlerle ortamda değişikliğe sebep olarak sağlanırken kendinden organizasyon yardımıyla önceden yapılmış herhangi bir plan olmadan sonuç üretebilmelerini, esnek ve sağlam merkezi bir birim olmadan yapılanmalarını sağlar.

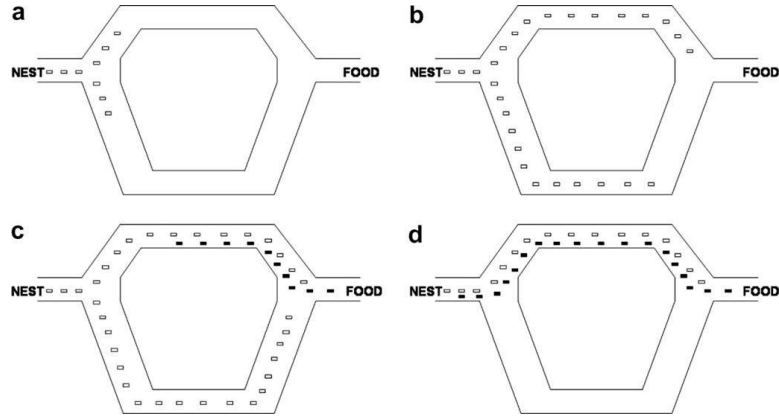
Bu gerçek yaşamdaki ortak zeka üzerine kurulu bir çok algoritma bulunmaktadır, örnek olarak böcek ve arıların toplu olarak geliştirdikleri çözümler verilebilir.

Orijinal karınca kolonisi algoritmalarının başarısı ilk olarak Dorigo ve arkadaşları tarafından önerilmiştir(1996).Bu başarının ardından bilinen ilk genel amaçlı optimizasyon tekniği Karınca kolonisi optimizasyonudur. Daha sonradan algoritma

çeşitli sezgisel yaklaşımlarla geliştirilmiştir. [22] Bu sezgisel yaklaşımlara benzetimli tavlama(simulated annealing) [24],tabu arama [25] ve yerel arama[26] da dahil edilmiştir.

2.3. Karıncaların Yiyecek Arama Mekanizması

Birçok karınca türü yiyecek yuva ve diğer karıncalar arasındaki en kısa mesafeyi bulmada oldukça yeteneklidir. Birçok karınca türü kördür. Yuva ile yiyecek kaynağı arasında ilerlerken yola feromon adı verilen kimyasal bir sıvı sürerler. Eğer yolda hiç feromon kimyasalı yok ise karıncalar rastsal olarak ilerlerler. Ama koku aldıklarında da bu kokuyu takip etme eğilimi gösterirler. Biyologların yaptıkları araştırmalarda karıncaların ilerledikleri yol üzerindeki feromon miktarının çok fazla olduğu tespit edilmiştir ve karıncalar yollarını bu feromon kokusuna göre tayin etmektedirler feromon miktarı ne kadar fazla ise yolun cazibesi o kadar artış göstermektedir. Bu davranış karıncanın yuva ile yiyecek kaynağı arasındaki en kısa yolu tanımlamasına olanak sağlar. Peki, bu mekanizma karıncaların en kısa yolu bulmasına nasıl izin verir.



Şekil 2.1. Karıncaların yol bulma işleminin şematik gösterimi (Şekil [27] den alınmıştır.)

Şekil 2.1 deki diyagramda yuva ile besin kaynağı arasında iki yol gözükmektedir. Üstteki yol daha kısa alttaki yol daha ona göre daha uzun bir yoldur. Yapılan deneyde karıncalar yiyecek aramaları için bırakıldığında karıncaların yarısı üstteki kısa yolu diğer yarısı da alttaki uzun yolu tercih etmişlerdir.(Şekil 2.1 a)

Üstteki yolun daha kısa olması nedeniyle karıncalar yiyeceğe daha kısa sürede ulaşmaktadır.(Şekil 2.1 b)

Karıncalar besin kaynağına ulaştıktan sonra bu yola feromen sürerek yuvaya tekrar geri dönerler ve bu feromen izlerini takip ederek yiyeceğe ulaşırlar.(Şekil 2.1 c)

Bu şekilde üst yolda biriken feromen miktarı daha fazla olacaktır. Bu yüzden karıncaların üstteki yolu takip etme olasılıkları daha fazla olacaktır. Bu işlem sonunda en kısa yol için üst kısımda bulunan yola bir yakınsama elde edilir.

Algoritma ilk olarak Travelling Salesman Problem (TSP) Gezgin Satıcı Problemine uygulanmış ve iyi sonuçlar elde edilmiştir.

2.4. Yapay Karıncaların Özellikleri

Yapay karıncalarla ilgili temel özellikler şu şekilde sıralanabilir. [28]

- Verilen problemin olası çözümlerinde maliyeti en düşük olan çözümü araştırır.
- Her yapay karınca geçtiği yolları tutan dahili bir hafızaya sahiptir.
- Başlangıç durumundan itibaren verilen problem için her karınca uygun sonuçlar üretmeye çalışır. Arama uzayı üzerinde yinelemeli olarak sonuçlar üretirler.
- Bir karıncanın S_i durumundan S_j durumuna hareketi sırasında bir geçiş kuralı rehberlik etmektedir. Bu kural probleme özgü olarak bir takım kısıtlar içerebilir.
- Her karıncanın sahip olduğu feromen miktarı probleme özgü feromen güncelleme kuralına göre yapılır.
- Karıncaların depolayacakları feromen miktarı durum ve durum geçişleri ile ilgilidir.

- Feromen depolanması çözüm oluşturulurken her durum geçişinde gerçekleştirilir. Buna adım adım feromen izi güncellemesi denilir.
- Algoritma hedeflenen duruma ulaşılnca sonlandırılır.

Bu özelliklere ek olarak yapay karıncaların performanslarını arttırmak doğal karıncalarda olmayan ek özelliklere sahip olabilirler. Örnek olarak Yerel Arama[26] ve Aday Listesi [22] örnek olarak verilebilir.

Problem çözümünde Daemon (Arka Planda Çalışan) Uygulamalar algoritmalara eklenmiştir. Bu uygulamalar karıncalara algoritmaların çalışma zamanında yönlendirme etkisi yapar ve algoritmayı hızlandırır. Bu uygulamalara örnek olarak her iterasyonun sonunda en iyi sonuca feromen eklenmesi verilebilir.

2.5. Normal Karıncalar ve Yapay Karıncalar'ın Karşılaştırılması

- Koloni elemanları verilmiş olan görevi birbirleri ile etkileşimli olarak işbirliği yaparak çözüme ulaştırır.
- Hem yapay hem de doğal karıncalar feromen kokusuna bağlı olan iletişim ile çevrelerini değişikliğe uğrattırlar.
- Hem doğal hem de yapay karıncalar yuvaya ulaşma, yiyeceğe ulaşma gibi ortak bir görevi paylaşırlar.
- Hem doğal hem de yapay karıncalar çözüme tekrarlı yerel aramalarla ulaşmaya çalışır.
- Yapay karıncalarda feromen bilgisi dışında ek problem parametreleri de algoritmanın içerisine dahil edilir.
- Yapay karıncalar doğal karıncalardan farklı olarak izledikleri yolu tutan bir hafızaya sahiptir.
- Yapay karıncalarda bırakılan feromen miktarı sabit kabul edilirken bu değer doğal karıncalarda farklı olabilir.
- Feromen buharlaşması doğada yapay karıncalardan farklı olarak gerçekleşir.

2.6. Orginal Karınca Sistemi

1996 yılında Dorigo tarafından geliştirilmiştir. TSP Gezgin satıcı problemi üzerinde iyi sonuçlar üretmiştir. Gezgin satıcı problemi M adet şehir için tüm şehirleri sadece bir kez gezerek en kısa mesafeyi bulma problemidir. Öklit teoremi Gezgin Satıcı Probleminde i ve j şehirleri arasındaki mesafeyi bulmak için kullanılır.

$$d_{ij} = \left[(x_i - x_j)^2 + (y_i - y_j)^2 \right]^{1/2} \quad (2.1)$$

Sistemde bulunan her karınca şu özelliklere sahiptir.

Karıncanın bulunduğu şehirden hangi şehre gideceği bir geçiş kuralı ile belirlenir. Bu kural mevcut şehirle bağlı şehirlerarasındaki uzaklık ve mevcut şehirle bağlı şehirlerarasındaki feromen miktarının bir fonksiyonudur.

Geçişlerin yapıldığı şehirler tabu listesinde tutulur. Bu şekilde bu şehirlerin tekrar ziyaret edilmesi önlenir.

Bir tur tamamlandığında karınca feromen izini ziyaret ettiği her yola sürer.

Karıncalar kesikli sürelerde hareket ederler.(t,t+1) Aralığında tüm N karınca bir şehirden diğer bir şehre hareket ederler. Karıncalar n iterasyonun ardından feromen güncellemesi gerçekleştirirler. Bu güncelleme şu şekilde ifade edilebilir.

$$\tau_{ij}(t+n) = (1-\rho) \cdot \tau_{ij} + \sum_{k=1}^N \Delta\tau_{ij}^k \quad (2.2)$$

$\rho \in (0,1]$ değer alır ve buharlaşma miktarını gösterir. $\Delta\tau_{ij}^k$ k. Karıncanın t ve t+n zamanda (i,j) yoluna sürdüğü feromen miktarıdır.

$$\Delta\tau_{ij}^k = \begin{cases} Q/L_K & \text{eğer karınca k kenar(i,j) turunda kullandıysa,} \\ 0 & \text{diğer türlü} \end{cases} \quad (2.3)$$

Q:sabit bir deęer

L_k : k. Karıncanın bulduęu tur uzunluęu

n_{ij} : Sezgisel bilgi grnrlk olarak bilinir $1/d_{ij}$ feromen izinden farklı olarak bu deęer algoritma boyunca deęiřiklięe uęratılmaz.

$tabu(s)_k$: Tabu listesi nceki ve řu andaki ziyaret edilen řehirlerin listesini tutar.

k. karıncanın i řehrinden j řehrine hareketini belirleyen olasılık yoęunluk fonksiyonu řu řekildedir,

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{k \in allowed_k} [\tau_{ik}(t)]^\alpha \cdot [\eta_{ik}]^\beta} & \text{eęer } j \in allowed_k \\ 0 & \text{dięer trl,} \end{cases} \quad (2.4)$$

$allowed_k : \{M-tabu_k\}$ Tabu listesinde gidilmeyen řehirlerin listesini ifade eder. α ve β ayar parametreleridir.

2.7. Karınca Algoritmaları ve Hesaplamalı Optimizasyon Teknikleri

Gezgin Satıcı Probleminde başarılı sonuçlar reten AS algoritması ACO Karınca kolonisi Optimizasyonu sezgisellerinin geliřmesinde katkıda bulunmuřtur. [22]

Karınca Kolonisi Optimizasyonu sezgiseli Kombinasyonel Optimizasyon Problemlerinin zmnde genel yntemler ortaya koymuřtur. Bu sebeple Karınca Kolonisi Optimizasyonu sezgiselinin tm zel uygulamaları karınca algoritmaları olarak adlandırılmasına raęmen karınca algoritmaları karınca kolonisi optimizasyonunu tam olarak karřılamaz.

2.7.1. Karınca Kolonisi Optimizasyonu Sezgiseli

Karınca Kolonisi Optimizasyonu üç ana fonksiyon üzerine kurulmuştur(Algoritma 1).

KarıncaCozumuOlustur(), karıncalar problemdeki komşu düğümlere belirlenen geçiş kuralına göre sıralı bir biçimde sonuç üretirler.

FeromonGuncelle(), feromon izi ya her iterasyonda ya da çözüm oluşturulduktan sonra güncellenir. Feromon izi güncellemesinin yanı sıra Karınca kolonisi optimizasyonu feromon izi buharlaşması da gerçekleştirmektedir. Bu işlemle bulunan kötü sonuçların olası çözümlerden çıkarılması sağlanmıştır. Buharlaşma işleminden kasıt, her iterasyon işlemi sonrası yollar üzerindeki feromon miktarının belli bir oranda azaltılması kastedilmektedir.

ArkaPlanEylemleri(), küresel bir bakış açısıyla ek güncelleştirmeleri içerir. İsteğe bağlı bir adımdır. Bu işleme üretilen en iyi yol çözümüne ek feromon ekleme işlemi verilebilir. Bu işleme çevrim dışı feromon güncelleme denilir.

Algoritma 1

ParametreleriBelirle

While sonuc durumuna ulasana kadar

Zamanlama Etkinlikleri

 KarıncaCozumuOlustur()

 FeromonGuncelle()

 ArkaPlanEylemleri()

Bitir Zamanlama Etkinlikleri

Bitir While

Gelişmiş ACO Algoritmaları

Karınca Sistemi algoritmasının ardından Karınca Kolonisi Optimizasyonu algoritmasına birçok geliştirmeler eklenmiştir. Bunlar şu şekilde sıralanabilir.

2.7.2. Max-Min Ant System

Max-Min Karınca Sistemi (MMAS) [30] , karınca sistemi algoritmasından iki yönüyle ayrılır.

-Sadece en iyi yolu bulan karınca feromen izini günceller ve feromen güncelleme fonksiyonu şu şekilde değiştirilmiştir.

$$\tau_{ij}(t+n) = \left[(1-\rho) \cdot \tau_{ij}(t) + \sum_{k=1}^N \Delta\tau_{ij}^{best} \right]_{\tau_{min}}^{\tau_{max}}, \quad (2.5)$$

$$\Delta\tau_{ij}^{best} = \begin{cases} 1/L_{best} & \text{if } (i,j) \text{ belongs to the best tour,} \\ 0 & \text{otherwise,} \end{cases} \quad (2.6)$$

τ_{max} , τ_{min} alt ve üst feromen sınırlarını gösterir.

Güncelleme fonksiyonu x olmak üzere bu fonksiyonun alt ve üst limitleri şu şekilde belirlenir.

$$[X]_b^a = \begin{cases} a & \text{if } x > a, \\ b & \text{if } x < b, \\ x & \text{otherwise} \end{cases} \quad (2.7)$$

L_{best} :Karıncanın bulduğu en iyi tur uzaklığıdır.

2.7.3. Karınca Kolonisi Sistemi

Karınca kolonisi sisteminde [29] algoritma diğer algoritmalarından feromen güncelleme fonksiyonuyla farklılık gösterir. Doğal karıncaların davranışından farklı olarak yerel feromen güncelleme işlemi uygulanır. Yerel feromen güncellemesi şu şekildedir.

$$\tau_{ij} = (1 - \varphi) \cdot \tau_{ij} + \varphi \cdot \tau_0 , \quad (2.8)$$

$\varphi \in (0,1]$ bir değer almaktadır.(pheromone decay coefficient)

τ_0 : Feromen başlangıç değeridir.

Max-Min Karınca Sisteminde olduğu gibi bu algortmada da gizli feromen güncellemesi görülür. Bu güncelleme en iyi yolu bulan karıncanın bulduğu yola feromen eklenmesi ile gerçekleştirilir. Bu feromen güncellemesi fonksiyonu şu şekilde ifade edilir,

$$\tau_{ij} = \begin{cases} (1 - \rho) \cdot \tau_{ij} + \rho \cdot \Delta\tau_{ij} & \text{eğer karınca } (i,j) \text{ en iyi tura aitse} \\ 0 & \text{diğer türlü,} \end{cases} \quad (2.9)$$

$\Delta\tau_{ij} = 1/L_{best}$ ve L_{best} : En iyi yol uzunluğu

Karınca kolonisi sistemi farklı bir geçiş fonksiyonu olan rastgele orantı kuralını kullanır.

k karıncası i durumunda ve $q_0 \in [0,1]$ arasında değer alan bir parametre, q [0,1] arasında değer alan rastgele değer olmak üzere sonraki durum olan j durumuna geçiş için şu olasılık kuralı uygulanır,

If $q \leq q_0$:

$$p_{ij}^k = \begin{cases} 1 & \text{eğer } j = \text{argmax } j \in N_k(i) \tau_{ij} \cdot \eta_{ij}^\beta \\ 0 & \text{diğer türlü} \end{cases} \quad (2.10)$$

else ($q > q_0$) ve denklem (2.4) kullanılır

2.7.4. Sıra Tabanlı Karınca Sistemi

Sıra tabanlı karınca sistemi (AS_{rank}) [32] feromen güncelleme işlemi üzerinde bir sıralama işlemi gerçekleştirir. N adet karınca buldukları çözümlerin kalitesine göre sıralanır. Örnek olarak ($L_1 \leq L_2 \leq \dots \leq L_N$).Feromen izi çevrimdışı olarak arka planda $\sigma - 1$ en iyi karınca yolu üzerinde çalışan eylemler tarafından gerçekleştirilir.

Depolanan feromen miktarı direk olarak ilgili karıncanın sırasına(μ) ve bulduğu çözümün kalitesine bağlıdır. Ayrıca global en iyi sonucu üreten karıncanın bulduğu yola (L^{gb}) bu yolun kalitesi miktarınca feromen eklenir.

Feromen güncelleme fonksiyonu şu şekildedir,

$$\tau_{ij} = \rho\tau_{ij} + \Delta\tau_{ij}^{gb} + \Delta\tau_{ij}^{rank}, \quad (2.11)$$

$$\Delta\tau_{ij}^{rank} = \begin{cases} \sum_{\mu=1}^{\sigma-1} (\sigma - \mu) \frac{Q}{L_{\mu}} & \text{eğer the } \mu\text{th best ant travels} \\ & \text{on edge}(i, j), \\ 0 & \text{diğer türlü,} \end{cases} \quad (2.12)$$

$$\Delta\tau_{ij}^{gb} = \begin{cases} \sigma \frac{Q}{L^{gb}} & \text{if edge}(i, j)\text{ is part of the best solution found} \\ 0 & \text{diğer türlü.} \end{cases} \quad (2.13)$$

2.8. Karınca Algoritmaları Karşılaştırması

Tablo 1 de gösterilen sonuçlar [30] dört karınca algoritmasının performanslarının karşılaştırmasını göstermektedir. Bu performanslar her bir algoritmanın farklı boyuttaki gezgin satıcı problemi üzerindeki sonuçları üzerinden hazırlanmıştır. Bu karşılaştırmadan da görüldüğü gibi MMAS algoritması en iyi sonucu üretmektedir. Bu algoritmayı Karınca kolonisi sistemi(ACS), AS_{rank} üçüncü sırada yer alırken karınca sistemi (AS) algoritması bu algoritmalar içerisinde en kötü sonucu üretir.

Çizelge 2.1. Karınca Algoritmaları Karşılaştırılması

Örnek	En İyi	MMAS	ACS	AS_{rank}	AS
eil51	426	427.6	428.1	434.5	437.3
kroA100	21282	21320.3	21420.0	21746.0	22471.4
d198	15780	15972.5	16054.0	16199.1	16702.1

2.9. Karınca Algoritmalarının Geçmişteki ve Günümüzdeki Uygulamaları

Karınca Algoritmaları farklı alanlardaki birçok farklı probleme başarıyla uygulanmıştır. Endüstri alanında da giderek yaygınlaşan bu çözüm yöntemleri gerçek dünyada karşılaşılan problemlerde de başarılı sonuçlar üretmiştir.

Daha öncede belirtildiği gibi karınca algoritmaları NP-Zor kombinasyonel optimizasyon problemleri için oldukça uygundur. İlk uygulaması Karınca Sistemi algoritması olup yine bir NP-Zor problem olan Gezgin Satıcı Problemi üzerinde uygulanmıştır. Gezgin satıcı probleminin kısıtı en kısa yol problemidir.

Gezgin Satıcı Probleminin ardından algoritma farklı problem türlerine de uygulanmıştır. Bu problemler NP-Zor problemlerden olan ikinci dereceden atama problemleri (Quadratic Assignment Problem (QAP)) [23] ve İş planlama problemlerine (Job-Shop Scheduling Problem) [19] uygulanmıştır.

Karınca Algoritmalarının çözümünde kullanıldığı bir diğer problem veri paketlerinin ağ üzerinde yönlendirme problemidir. Bu problem sistemin özellikleri zaman içerisinde değişebilen en kısa yol problemidir. [33]

Karınca algoritmalarının çözümünde kullanıldığı başlıca uygulamalar Araç Rotalama Problemi [34],Çizge Renklendirme [35],Küme Kaplama [36] problemleridir.

Özellikle araç rotalama problemi ile sanayi arasında güçlü bir ilişki vardır. Şirketler gerçek dünyadaki çeşitli araç rotalama senaryolarının çözümü için karınca algoritmaları üzerine dayandırdıkları araçlar geliştirmişlerdir. Bunlara iki özel uygulama olan DyvOil ve AntRoute örnek olarak verilebilir.[37]

DyvOil akaryakıt satış ve dağıtım planlama süreci destekleyen bir uygulamadır. DyvOil çevrim dışı bir modülü karınca algoritmalarını kullanarak statik araç rotalama problemlerinin çözümünde kullanılır. Bir sonraki günün araç turlarını planlamak için her akşam uygulama çalıştırılır. Bu uygulama İsviçre deki akaryakıt dağıtım şirketi olan Pina Petroli test edilmiştir. Bu teknikle insan gücü kullanılarak yapılan planlara göre %30 lara varan araç rotalama performansı elde edilmiştir. [37]

Bu verimlilik karınca kolonisi sistemi kullanılarak dinamik araç rotalama problemlerinde de gözlenmiştir.

AntRoute İsviçre merkezli süpermarket olan Migrosta tedarik zincirinde kullanılmak üzere tasarlanmış olan benzer bir uygulamadır. MACS-VRPTW [38] dayanan bir algoritmadır. İsviçre genelinde süpermarketler araç dağıtım turlarını hesaplamak için bu uygulamayı kullanırlar. Bu uygulama da insanların yaptığı planlama işlemlerine göre daha iyi sonuçlar üretmiştir. Günümüzde karınca algoritmalarının kullanıldığı uygulamaların kapsamı arttıkça yeni uygulamalar ile karınca algoritmalarının da önemi önemli ölçüde artmıştır.

Bunun yanında son zamanlarda güncel birçok problem uygulamasında karınca algoritmaları kullanılmaktadır. Bu uygulamalara sürekli optimizasyon problemleri [39], paralel işleme uygulamaları [40] örnek olarak verilebilir.

Sürekli optimizasyon problemleri karınca algoritmaları için farklı yapıdaki problemlerdir çünkü karınca algoritmaları genellikle kesikli optimizasyon problemleri için geliştirilmiştir.[22] Bu tarz problemler için karınca kolonisi optimizasyonu sezgiselleri geliştirilmiştir.

2.10. Gelişmiş Karınca Algoritmaları

Yakın zamanlarda karınca algoritmalarını içeren araştırma ve geliştirmelerin artması geniş kapsamlı daha özel karınca algoritmalarının oluşturulmasına zemin hazırlamıştır. Bu algoritmalar görüntü işleme, hesaplamalı biyoloji, veri madenciliği gibi birçok alanda kullanılmıştır.

Dijital görüntü işleme ile ilgili bilimsel alanda birçok yöntem mevcuttur. Buna rağmen makine öğrenmesi ve yapay zekâ teknikleri görüntü işleme yöntemleri içerisinde kullanılmaktadır. Karınca algoritmalarının görüntü işleme teknikleri içerisinde kullanılması yeni bir yöntemdir. Karınca algoritmaları temel düşük seviyeli resim bölümlenme ve sınır algılama yöntemlerinde kullanılmıştır. Bu işlem

için çeşitli sezgisel işlemler geliştirilmiştir. Karıncaların bir sonraki piksel seçimlerinde olasılıksal bir yoğunluk ağırlık fonksiyonundan faydalanılır. Buradaki amaç fonksiyonu pikseller arasındaki en büyük değişimin olduğu pikselleri bulmaktır.

Ek olarak kendi doğrusal dizisinde protein'in yapısının belirlenmesi hesaplamalı biyolojide NP-Zor bir problemdir ve çok fazla hesaplama maliyetine sahiptir. Bu hesaplama maliyeti sebebiyle arama alanını basitleştirmek için çeşitli modeller geliştirilmiştir. Bu modeller içerisinde de Karınca algoritmalarından faydalanılmaktadır.

Veri Madenciliğinde genel amaç bilgi çıkarsamadır. Çok büyük veri kümelerinden veriler arasındaki ilişkilerin bulunması ve çeşitli tasarım örüntülerinin çıkarılmasını sağlamaktadır. Veri madenciliği algoritmaları sınıflandırma, kümeleme, tahmin etme gibi işlemleri içerir. Karınca algoritmaları da bu işlemlerden sınıflandırma işlemleri içerisinde kullanılmaktadır.

Karınca Algoritmaları bilim ve sanayi gibi farklı alanlarda birçok teorik ve pratik problem üzerinde uygulanmaktadır.

BÖLÜM 3

3. KÜMELEME ANALİZİ

3.1. Giriş

Kümeleme işlemi verinin anlamlı ve faydalı gruplara ayrılması işlemidir. Büyük boyutlardaki verilerin özetlenmesinde sıklıkla kullanılır. Kümeleme işlemi teorik ve pratik birçok alanda yaygın olarak kullanılmaktadır. Biyoloji, örüntü tanıma, istatistik, bilgi çıkarsama, veri madenciliği ve makine öğrenmesi gibi pozitif bilimler de kullandığı gibi psikoloji gibi sosyal bilimlerde de kümeleme işleminden faydalanılmaktadır. Kümeleme analizinin birçok problem üzerinde uygulamalarına rastlanmaktadır.

Kümeleme ile veriler anlamlı nesne grupları haline dönüştürülmektedirler. Bu işlem insanların doğayı anlamalarında önemli katkılar sağlamıştır. Bu şekilde doğadaki canlıları ortak özelliklerine göre kümeleme yoluna gidilerek sınıfların araştırılması işlemi gerçekleştirilmiştir. İnsan nesnelere çeşitli gruplara ayırma da ve benzer nesnelere bu gruplara atamada doğal bir yeteneğe sahiptir. Bu işleme örnek olarak küçük bir çocuğun gördüğü resimdeki nesnelere insanlar, hayvanlar bitkiler gibi etiketlendirmesi verilebilir.

3.2. Kümeleme Analizi

Kümeleme analizi, 1955 yılında nesnelere sahip oldukları karakteristik özelliklerini temel alarak gruplamak amacıyla ortaya çıkmış yöntemler topluluğudur. Kümeleme analizinde amaç fonksiyonu aynı grup içerisindeki verilerin birbirleri ile ilişkili olması ve farklı gruplar altında tanımlanan verilerin birbirinden farklı olmasını sağlamaktır. Bu işlem verileri tanımlayan ve birbirleri ile olan ilişkilerini ortaya koyan parametreler ışığında gerçekleştirilir.

Eğer amaç anlamlı gruplar ise, bu durumda kümeler verinin doğal yapısını yakalamalıdır. Bu sebeple kümeleme analizinin uygulama adımları şöyle olmalıdır;

- 1) Değişkenlerin doğal gruplamaları hakkında kesin bilgilerin bulunmadığı veri setlerinden alınan n sayıda birimin p sayıda değişkenine ilişkin gözlemlerin elde edilmesi, yani veri matrislerinin belirlenmesi,
- 2) Değişkenlerin birbirleri ile olan benzerliklerini ya da farklılıklarını gösteren uygun bir benzerlik ölçüsü ile değişkenlerin birbirlerine uzaklıklarının hesaplanması yani benzerlik ve farklılık matrislerinin belirlenmesi,
- 3) Uygun küme yöntemi yardımı ile benzerlik/farklılık matrisine değişkenlerin uygun sayıda kümelere ayrılması,
- 4) Elde edilen kümelerin yorumlanması ve bu kümeleme yapısının doğrulanması için gerekli analitik yöntemlerin uygulanması.

Bu adımlamalardan da anlaşılacağı gibi kümeleme analizi ile çok sayıda değişik işlev yerine getiren farklı yöntemler kullanılabilir. Ayrıca değişkenlerin ölçü birimlerinin ve ölçümleme tekniklerinin farklı olmasından dolayı birimlerinin benzerliklerinin ortaya konmasında da değişik ölçüler kullanılır.

Kümeleme analizi, dört ana amaca yönelik işlemler gerçekleştirmektedir.

- 1) n sayıda nesnenin, p değişkene göre saptanan özellikleriyle, kendi içerisinde türdeş ve yine kendi aralarında farklı alt gruplara ayırmak,
- 2) p sayıda değişkeni, n sayıda birimde saptanan değerlere göre ortak özellikleri açıkladığı varsayılan alt kümelere ayırmak ve ortak faktör yapıları ortaya koymak,
- 3) Hem birimleri hem de değişkenleri birlikte ele alarak ortak n birimi p değişkene göre ortak özellikli alt kümelere ayırmak,
- 4) Birimleri, p değişkene göre saptanan değerlere göre, izledikleri biyolojik ve tipolojik sınıflamayı ortaya koymak.

3.3. Kümeleme Analizi'nin Kullanım Alanları

Verileri anlamlandırma ve çok büyük verilerden bilgi çıkarsama noktasında kümeler sınıfları gösterir ve kümeleme analizi de olası bu sınıfların otomatik olarak bulunmasına yönelik tekniklerin bütünüdür. Kümeleme analizinin Biyoloji, iklim etkileri, psikoloji, tıp, ticaret ve milyarlarca veri içeren internet gibi yoğun yapılardan bilgi çıkarsamasında kullanılmaktadır.

Özellikle internet üzerinde bulunan milyarlarca web sayfası üzerinde arama motorlarının daha hızlı ve doğru veriler döndürebilmesi için kümeleme analizinden faydalanılmaktadır. Kümeleme bu verilerin gruplara ayrılmasında kullanılır ve her grupta kendi içerisinde alt gruplara ayrılarak benzer verilerin bir arada bulunması sağlanmış olur. Örneğin son kullanıcının arama motoru üzerinde yaptığı bir film sorgusu için bu film yönetmenleri, oyuncular, film özetleri gibi küme grupları içerisinde son kullanıcıya daha hızlı ve doğru bilgi sunulmuş olacaktır.

Yine biyologlar yıllardır hiyerarşik sınıflandırmalarla uğraşmışlardır. Sınıf, aile, cins, tür gibi gruplar üzerinden canlıları sınıflandırma yoluna gitmiştir. Son zamanlarda kümeleme analizi biyologlar tarafından büyük boyuttaki gen bilgileri üzerinde uygulanmakta ve aynı özelliğe sahip genlerin bulunmasında kullanılmaktadır.

İklimlerin ve hava durumu tespitinde atmosferdeki ve yeryüzündeki hareketlerden bazı örüntülerin çıkarılması gerekmektedir. Yapılan kümeleme analizleri ile kutupsal bölgelerdeki atmosferik basınçlara ilişkin örüntülerin ve kara iklimine önemli etkisi bulunan okyanus alanlarının tespit edilmesinde kullanılmıştır.

Tıp ve Psikoloji alanında da kümeleme analizi sıkça kullanılmaktadır. Bir hastalığa ait çeşitliliklerin ortaya çıkarılmasında kullanılmıştır. Bu şekilde depresyonun birçok türü yapılan çalışmalarda ortaya çıkarılmıştır. Hastalıkların zaman veya belirli bir alan içindeki dağılımlarına göre çeşitli örüntülerin ortaya çıkarılmasında da kullanılmaktadır.

Kümeleme aynı zamanda potansiyel müşterilere ait büyük miktardaki verilerin küçük alt gruplara ayrılarak bilgiler üzerinde analiz ve pazarlama işlemleri yürütülmesine olanak sağlamaktadır.

3.4. Kümeleme Analizi ile Veriler Üzerinde Yapılan İşlemler

Kümeleme Analizi ile Özetleme, Veriyi sıkıştırma, Komşulukların verimli bir şekilde bulunması, Veriden faydalanma gibi işlemler yapılabilir.

Veri analizi yaparken birçok teknik kullanılmaktadır. Bunlar Regresyon, PCA gibi daha da çoğaltılabilecek yöntemlerdir. Ancak var olan bu yöntemler ile çok büyük veri tabanları üzerinde yapılacak işlemler yüksek karmaşıklığa sahiptirler. Bu algoritmaların kullandıkları bellekte fazla olmaktadır. $O(m^2)$ (m veri sayısını göstermektedir)

Bu yüzden bu yöntemler büyük miktardaki veri setleri için uygulanabilir değildirler. Bu nedenle algoritmayı tüm veri seti yerine uygulamak yerine veri setini temsil eden bir alt grubuna uygulamak hem daha hızlı hem de daha etkili sonuçlar üretecektir.

Elde edilecek analiz sonuçları algoritmanın uygulandığı alt grubun kümeyi temsil etme oranına, yapılan analizin türüne, veri setinden alınan ve bu veri setini temsil eden örnek sayısına göre değişiklik gösterir. Bu kısıtlar ne kadar iyi seçilirse alt gruba uygulanacak olan analizin sonucu gerçek veriye uygulanan analizin sonucuna o kadar yakın olur.

Kümeleme analizi veri seti üzerinde soyutlama yapmamıza da imkân sağlar. Bu şekilde bir kümeyi temsil eden örneklerin çıkarılmasını sağlar. Bu şekilde daha rahat analiz ve veri işleme tekniklerinin uygulanabilmesi sağlanır. Bu nedenle kümeleme analizini veri setini en iyi temsil eden örneklerin bulunmasında yaygın olarak kullanılır.

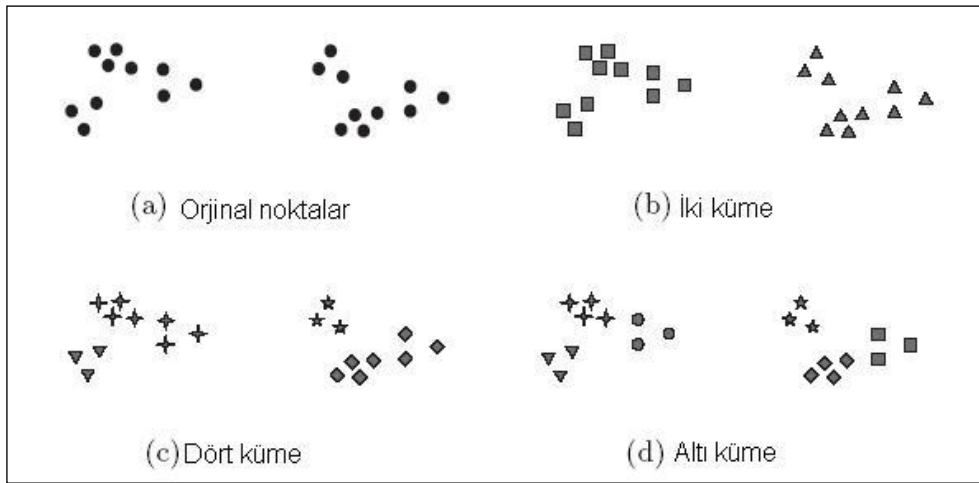
Kümeleme analizi ile yine karmaşık bir problem olan en yakın komşuluğun bulunması işlemi daha verimli hale getirilmiştir. En yakın komşuluk karşılıklı olarak tüm noktaların arasındaki mesafelerinin bulunmasını gerekli kılmaktadır. Bu işlem kümeleme analizi ile nesnelerin kümeyi temsil eden örneklere olan uzaklıklarına göre değerlendirerek yapılacak olan hesaplama sayısı azaltılabilir. Eğer iki kümeyi temsil eden küme örnekleri birbirinden sezgisel olarak uzakta ise bu iki nesnenin birbirinin komşuluğunda olma olasılığı çok düşüktür. Yani bir nesnenin en yakın komşuluğunu bulmak için yalnızca yakın kümeler içindeki nesnelere olan uzaklığını

bulmak yeterli olacaktır. Bu çıkarsamayı iki kümenin arasındaki uzaklığın kümeyi en iyi temsil eden örneklerinin arasındaki uzaklığa bağlı olmasından elde ediyoruz.

Kümeyi temsil eden örnekler verinin sıkıştırılması için kullanılmaktadır. Bu işlem için her kümeyi temsil eden örneklerin bir tablosu oluşturulur ve her örneğe tablodaki yerini gösteren bir indeks değeri verilir. Her veri, kümesi ile ilişkili temsilci örneğin indeks değeri ile temsil edilir. Bu işlem yoğun olarak ses ve görüntü verileri üzerinde uygulanmaktadır. Bu veriler de tekrar eden verilerin sayısı fazladır bu yüzden oluşan veri kaybı kabul edilebilir bir seviyede yani görüntü üzerinde göz ile algılanamayacak seviyededir. Özellikle internet üzerindeki veri aktarımı işlemlerinde genellikle tercih edilen bir durumdur.

Kümeleme işleminde oluşan kümelemenin doğruluğu kümeleme algoritmalarının üretilen kümelerin geçerliliği ile ilgilidir. Kümelemede genel olarak iki işlem üzerinde durulmaktadır. Bunlar bir veri setini kümelemenin yolları ve oluşan kümelerin türleridir.

Şekil-1 de aynı veri seti üzerinde farklı küme sayılarının dağılımı gösterilmiştir. Bir veri setindeki küme sayısını belirlemenin farklı yöntemleri bulunmaktadır. Şekilde 20 noktanın farklı sayıdaki küme örnekleri verilmiştir.



Şekil 3.1 Aynı Veri Seti Üzerindeki Farklı Küme Sayıları (Şekiller [42] den alınmıştır)

Şekil-1 (b) de veri öncelikle iki ve Şekil-1 (d) de altı alt kümeye ayrılmıştır. Şekil-1 (c) de veri dört alt kümeye ayrılmıştır. Görüldüğü gibi bir veri setinin sahip olduğu küme sayısı verilerin yapısına ve arzu edilen sonuçlara göre değişiklik gösterir.

Kümeleme bir çeşit sınıflandırma olarak düşünülebilir. Sınıfların etiketlerine göre nesnelerin etiketleri oluşturulur. Kümeleme bu bilgiyi veriden alır. Yapılan sınıflandırma bir öğreticili sınıflandırmadır. Yani yeni ve etiketlenmemiş veriler, daha önceden bilinen sınıf etiketlerinden oluşturulmuş bir model aracılığıyla birer sınıf etiketine sahip olurlar.

Kümeleme analizi kimi zamanda öğreticisiz sınıflandırma olarak ta nitelendirilir. Yani veri setinden oluşacak olan küme sayısı daha önceden belirlenmeksizin verinin yapısına göre şekillenir. Aksi belirtilmediği şekliyle sınıflandırma işleminde öğreticili sınıflandırma işaret edilmektedir.

3.5. Kümeleme Türleri

Kümeleme Hiyerarşik(iç içe),İç içe olmayan, Örtüşen, Bulanık, Tam, Kısmi, Seçkin gibi türlere sahiptir. Bu türler daha iyi anlatılabilmesi için birbirleri ile karşılaştırmalı olarak anlatılma yoluna gidilmiştir.

3.5.1. Hiyerarşik(İç içe) Kümeleme ve İç içe olmayan Kümeleme

İç içe olmayan kümeleme bir veri setinin üyelerinin örtüşmeyen alt kümelere ayrılması işlemidir. Yani her veri sadece bir kümeye aittir.

Kümelerin alt kümelere ayrılması durumunda hiyerarşik kümeleme yapmış oluruz.

Nesnelerin analiz edilmeden önce, hiyerarşik bir yapıya göre düzenlenmesidir. Hiyerarşik kümeler ağaçlar şeklinde organize edilmiş iç içe geçmiş alt kümelerden oluşur. Yaprak düğümler dışında ağaçtaki her bir düğüm, kendi alt kümelerinin bir birliği ve ağacın kökü ise tüm nesnelere içeren bir kümedir Hiyerarşik kümeleri ağaç şeklinde tasarlanmış iç içe geçmiş kümelerin bir birleşimi olarak tanımlayabiliriz.

Yaprak düğümler dışındaki her bir düğüm kendi alt kümelerinin bir birliği ağacın kökü de tüm nesnelere içeren bir kümedir.

3.5.2. Seçkin, Örtüşen ve Bulanık Kümeleme

Bir verinin; sadece bir kümeyle ait olma durumunu seçkin kümeleme olarak, birden fazla kümeyle ait olma durumunu da örtüşen kümeleme olarak tanımlıyoruz. Örtüşen kümeyle ait olarak bir üniversite de hem öğrenci olan hem de üniversitede çalışan bir kişi örnek olarak verilebilir. Bulanık kümeleme de veri belli bir ağırlık değeriyle tüm kümelere ait olmaktadır. Bu değer $[0,1]$ arasında değerler almaktadır. Yani veri belirli bir olasılık değeri ile o kümeyle aittir. Bulanık kümelemede toplam ağırlık değeri 1'dir.

Bulanık kümeler seçkin kümelere dönüştürülebilir. Bu işlem verinin ağırlık değerinin en fazla olduğu kümeyle aktarılması ile gerçekleştirilir.

3.5.3. Tam Kümeleme ve Kısmi Kümeleme

Tam kümelemede veri setindeki her veri bir kümeyle atanırken, kısmi kümelemede veri seti içerisindeki bazı veriler hiçbir kümeyle ait olmaksızın dışlanırlar. Bu tür veriler veri seti içindeki gürültüler olarak bilinir. Örnek olarak üzerinde işlem yapılan resim verisi için anlamlı olmayan bir arka plan verisi hiçbir kümeyle dahil edilmeden dışlanır. Böylece kümelerin kalitesi arttırılmış olur.

3.6. Küme Türleri

Kümelemede amaç veriyi faydalı gruplara ayırmaktır. Buradaki fayda veri analizinin hedefleri tarafından tanımlanır. Bu fayda değerine göre kümeler farklı şekillerde görsellik gösterebilirler. Buda farklı küme türlerinin oluşmasına olanak sağlar. Bu küme türleri genel olarak iyi ayrılmış, örnek tabanlı, çizge tabanlı, yoğunluk tabanlı, kavramsal kümeler olarak sıralanabilir.

3.6.1. İyi Ayrılmış Kümeler

Bu tür kümelerde küme içerisindeki her bir veri aynı küme içerisindeki bir diğer veriye benzer ve yakın iken küme dışındaki verilerden uzak ya da bu verilerden farklıdır. Küme içindeki verilerin birbirleri ile olan ilişkilerinin belirlenmesinde yani bir verinin diğer bir veriye olan yakınlığının hesaplanmasında belirli bir eşik değeri kullanılır.

Farklı gruplar içinden rastgele alınan iki nokta arasındaki uzaklık aynı grup içerisinde alınmış herhangi iki nokta arasındaki uzaklıktan daha fazladır. İyi ayrılmış kümelerin belirli bir şekilleri yoktur. Küre gibi sabit bir şekle sahip olmak zorunda değildirler.

3.6.2. Örnek Tabanlı Kümeler

Bu tür kümelerde küme içerisinde bulunan her bir veri kümeyi temsil eden bir örnek özelliği taşır. Aynı küme içerisindeki verilerin karakterleri o kümeyi temsil eden verinin karakteri ile aynı iken farklı kümeleri temsil eden verilerin karakteri ile farklı bir karakter gösterir. Sürekli özellik gösteren veriler için kümeyi temsil eden veri o kümedeki değerlerin aritmetik ortalamalarının alınması ile elde edilir. Kümedeki verilerin aritmetik ortalamalarının anlamlı olmadığı durumlarda yani veri kategorik özelliklere sahip bir veri ise veri setini en iyi temsil eden veri bir medoid'dir. Yani kümeyi en iyi temsil edecek noktadır. Birçok veri seti için bu veri kümenin merkezi olarak düşünülebilir. Bu yüzden örnek tabanlı kümelere merkez tabanlı sınıflarda denilir. Bu kümeler küresel bir şekle sahip olma eğilimindedirler.

3.6.3. Çizge Tabanlı Kümeler

Eğer veri seti içinde bulunan her bir veri bu verinin kümedeki diğer verilerle olan ilişkisini gösteren bir çizge olarak temsil edilebiliyorsa küme bir bağlı bileşen olarak tanımlanabilir. Yani grup içindeki bir verinin yine grup içinde bulunan veri ile arasında bağlantısı olup grup dışındaki verilerle arasında herhangi bir bağlantı bulunmamaktadır. Komşuluk tabanlı kümeler çizge tabanlı kümelerin önemli bir

örneğidir. Bu tarz kümelerde verinin komşuluk içindeki veriye uzaklığı komşuluk dışındaki veriden daha kısadır. Bu kümelerdeki dezavantaj birbirleri ile farklı kümeler olsalar bile aradaki küçük bir gürültü verisi yüzünden aynı kümeymiş gibi birleşebilirler. Bu kümelerde örnek tabanlı kümeler gibi küresel olma eğilimindedirler.

3.6.4. Yoğunluk Tabanlı Kümeler

Nesne yoğunluklarına göre yapılan gruplama işlemidir. Yoğunluk tabanlı kümelemede alanlar veri yoğunluğunun fazla ve az olmasına göre belirlenir. Oluşan kümeler daha düşük yoğunluklu bölgelerden ayrılan daha yüksek yoğunluklu noktaların bir kümesidir. Üretilen demetler birbirinden farklı şekillerde olabilir.

Birçok kümeleme yöntemi nesnelere birbirleri arasındaki farklılıklarına göre kümeleme yaparken, bu metod nesnelere yoğunluğuna göre gruplama yapar. Yoğunluktan kasıt, analiz edilen nesnelere sayıdır. Yoğunluk bazlı metotlara örnek olarak DBSCAN verilebilir.

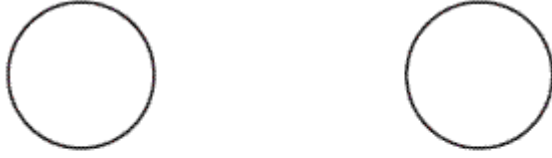
Yoğunluk tabanlı kümeleme güçlü bir yöntem olmasından ötürü istisna verilerden ve gürültülü verilerden etkilenmez. Yoğunluk tabanlı kümeleme işlemi belirlenen veri demetine son veri eklenene kadar devam eder bu aşamalar sürecinde yoğunluk parametresinin verilmesi gerekmektedir. Bir kümenin yoğunluk tabanlı tanımlanması daha çok kümeler düzensiz ya da birbirlerine geçmiş iken ve aynı zamanda veri setinde gürültü ve dışlayanlar var iken kullanılır.

3.6.5. Paylaşılan Nitelik ve Kavramsal Kümeler

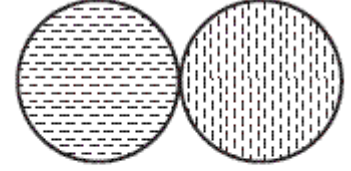
Bir kümeyle bazı özellikleri paylaşan veri setleri olarak tanımlayabiliriz. Paylaşılmış özellik yaklaşımı yeni küme türlerini de kapsar. Aynı ortak nitelikleri paylaşan veya kısmi bir kavram sunar. Kavramsal kümeleme 1980'li yıllarda öğreticisiz sınıflandırma teknikleri için makine öğrenmesi paradigması ortaya koyar.

Klasik kümeleme tekniklerinden her oluşturulan sınıf için bir kavram tanımlaması yapılması yönünden ayrılır. Bununla beraber paylaşılmış özellik yaklaşımı yeni küme türlerini de kapsar.

Şekil-2' deki kümeleri ele alalım. Bir üçgensel alan(küme), bir dikdörtgensel alana yakındır ve iki tane birbirine geçmiş daire(küme) vardır. Her iki durumda da, bir kümeleme algoritması kümeleri tespit edebilmek için özel bir küme kavramına ihtiyaç duyacaktır. Böylesi kümeleri bulma işlemi, kavramsal kümeleme olarak adlandırılır.

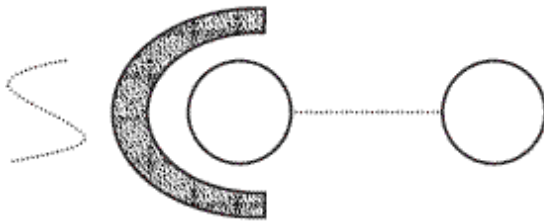


(a) İyi ayrılmış kümeler. Her bir nokta kendi kümesi içindeki noktalara diğer kümelerdeki noktalardan daha yakındır.

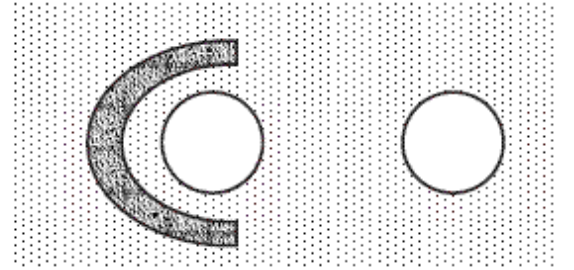


(b) Merkez tabanlı kümeler. Her bir nokta kendi küme merkezine diğer küme merkezlerinden daha yakındır.

Şekil 3.2 İyi ayrılmış ve Merkez Tabanlı Kümeler (Şekiller [42] den alınmıştır)

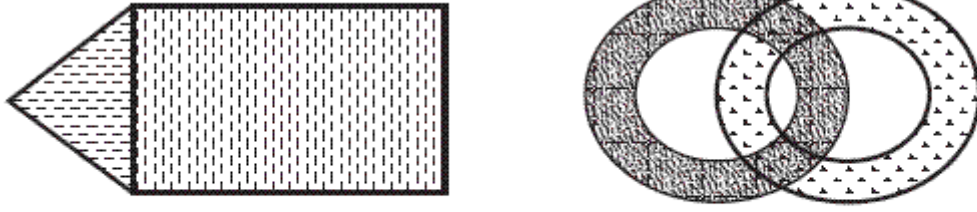


(c) Komşuluk tabanlı kümeler. Her nokta kendi kümesindeki en az bir noktaya diğer kümedeki herhangi bir noktadan daha yakındır.



(d) Yoğunluk tabanlı kümeler. Kümeler düşük yoğunluklu alanlarla ayrılmış yüksek yoğunluklu alanlardır.

Şekil 3.3. Komşuluk Tabanlı ve Yoğunluk Tabanlı Kümeler (Şekiller [42] den alınmıştır)



(e) Kavramsal kümeler. Bir kümedeki noktalar, tüm nokta seti tarafından türetilen bazı genel özellikleri paylaşırlar. Dairelerin arakesitindeki noktalar her ikisine de aittir.

Şekil 3.4. Kavramsal Kümeler (Şekiller [42] den alınmıştır)

3.7. Kümeleme Teknikleri

Başlıca kümeleme teknikleri şu şekilde sıralanabilir. K-Ortalama, Toplamsal Hiyerarşik Kümeleme, DBSCAN bu tekniklerden yapılan uygulamada K-Ortalama tekniği kullanılmıştır.

3.7.1. K-Ortalama Kümeleme

Öğreticisiz bir yöntem olan K-Ortalama kümeleme eldeki verileri özelliklerine göre hiçbir sınıf bilgisi olmadan K sayıda kümeye gruplama işlemidir. Bir prototip tabanlı ve kısmi kümeleme tekniğidir.

K-Ortalama kümelemede amaç, gerçekleştirilen bölümlene işlemi sonunda elde edilen kümelerin, küme içi benzerliklerinin maksimum ve kümeler arası benzerliklerinin minimum olmasını sağlamaktır.

Küme benzerliği, kümenin ağırlık merkezi olarak kabul edilen bir nesne ile kümedeki diğer nesneler arasındaki uzaklıkların ortalama değeri ile ölçülmektedir.

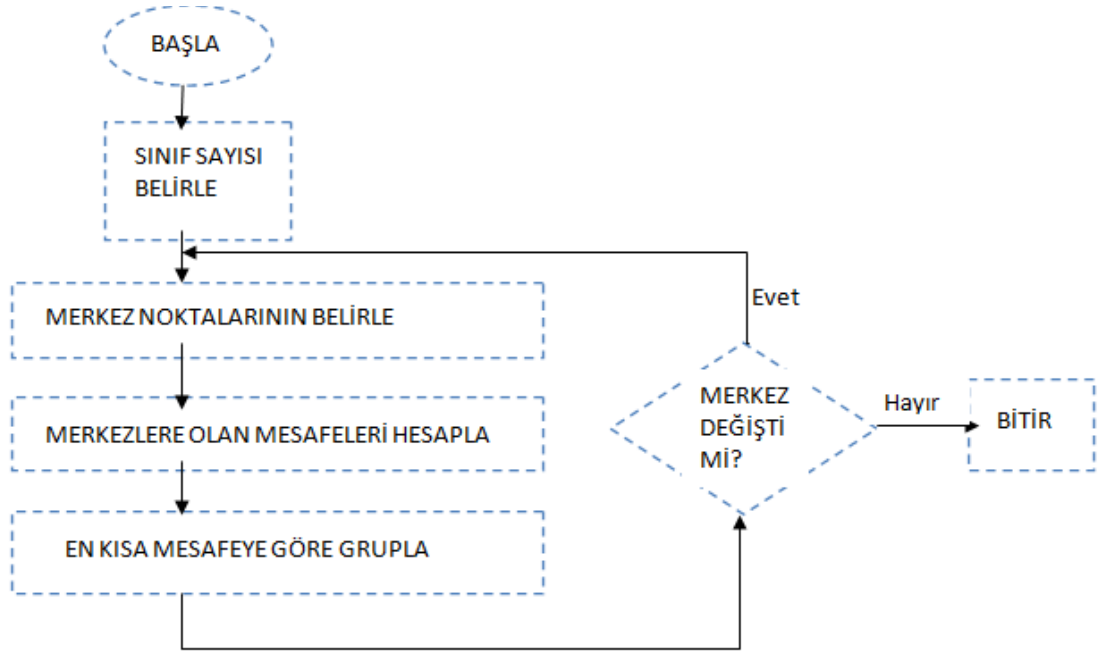
3.7.2. K-Ortalama İşlem Basamakları

Adım 1:Algoritma da ilk olarak küme merkezleri belirlenir. Bu işlem için iki farklı yol vardır. İlki veriler arasından küme sayısı kadar k adet rastgele nokta seçilir. İkinci yol merkez noktaları tüm verilerin ortalaması alınarak belirlenmesi işlemidir.

Adım 2:Her verinin seçilen merkez noktasına uzaklığı hesaplanarak verilerin hangi küme merkezine ait oldukları tespit edilerek o kümeye dahil edilir.

Adım 3:Oluşan kümedeki verilerin aritmetik ortalaması alınarak yeni küme merkezleri belirlenir ve bu yeni küme merkezleri eski küme merkezleri ile yer değiştirilir.

Adım 4:Merkez noktaları sabit kalıncaya kadar adım2 ve adım3 tekrar edilir.



Şekil 3.5. K Ortalama işlem basamakları

Bazı merkez ve yaklaşıklık fonksiyonlarının kombinasyonu için, K-ortalamları daima bir çözüme yakınsar, yani K-ortalamları öyle bir duruma gelir ki artık hiçbir nokta bir kümeden bir diğer kümeye kaymaz ve merkezler de hareket etmez.

3.7.3. Noktaların En Yakın Merkeze Atanması

Aralık ölçekli değişkenler arasındaki karşılaştırma oranıyla değil fark ile yapılır. Oransal ölçekli değişkenlerse; alan, uzunluk gibi aralarında kıyaslama yapılırken bir oran belirtebileceğimiz değişkenlerdir. Aralık ve oransal ölçekteki değişkenleri bulunan birimleri kümelerken, değişik uzaklık ölçüleri kullanılır. Bu ölçülerden bazıları şunlardır.

Öklid Mesafesi:

Öklid mesafesi kullanılarak iki birim arasındaki uzaklık,

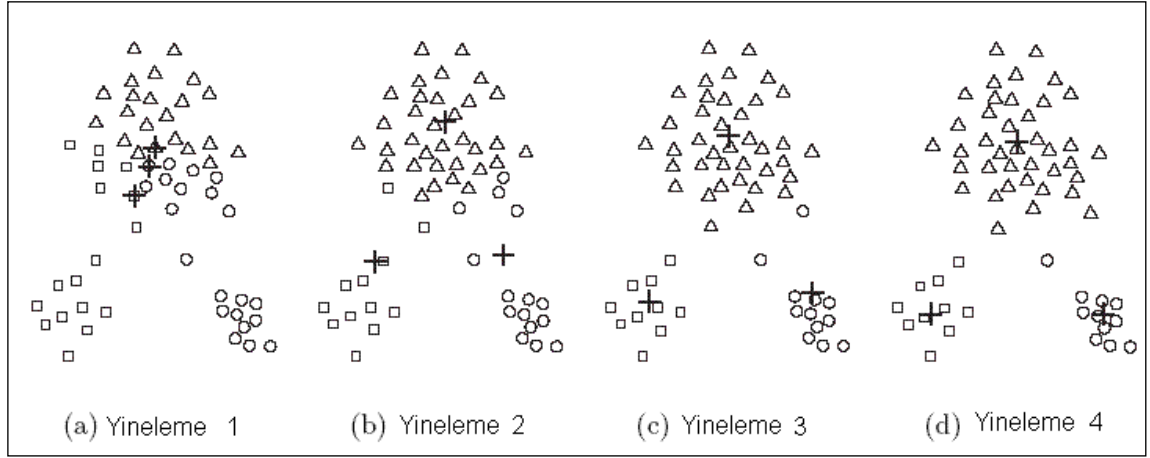
$$d(i, j) = \sqrt{(X_{i1} - X_{j1})^2 + (X_{i2} - X_{j2})^2 + \dots + (X_{ip} - X_{jp})^2} \quad (3.1)$$

Formülüyle hesaplanır. En çok kullanılan ölçü birimidir.

Eğer kümeleme analizinde kullanılacak olan değişkenler belirli bir önem derecesinde ağırlıklandırılmış ise, Öklid mesafesi formülü aşağıdaki gibi olur.

$$d(i, j) = \sqrt{w_1(X_{i1} - X_{j1})^2 + w_2(X_{i2} - X_{j2})^2 + \dots + w_p(X_{ip} - X_{jp})^2} \quad (3.2)$$

Şekildeki örnekte noktalar merkezlere atandıktan sonra merkez noktaları aritmetik ortalama işlemi ile güncellenir. Şekil (b) de noktalar güncellenmiş yeni merkezlere atanır ve merkezler tekrardan güncellenir ve son olarak merkez noktaları değişmeden sabit kalır ve yineleme işlemi sonlandırılır.



Şekil 3.6. Örnek bir veride k-ortalama algoritması ile kümelerin bulunması (Şekiller [42] den alınmıştır)

3.8. K-Ortalama Kümelemesi Özellikleri, Avantaj ve Dezavantajları

K-Ortalama kümelemesinin dört belirgin özelliği vardır. Bunlar;

- 1) Her zaman K sayıda küme olması
- 2) Her küme de en az bir nesne olması
- 3) Kümeler hiyerarşik olmamalı, ayrıca her hangi bir örtüşme de olmamalıdır.
- 4) Kümelerin her elemanı, kendine diğer kümelerden daha yakın olmalıdır. Çünkü yakınlık her zaman kümelerin merkezlerini kapsamaz.

Veri sayısı çok fazla olan hesaplamalarda, K-ortalama hesaplaması, eğer k küçük ise hesaplamaları hiyerarşik kümelemeden daha hızlı yapar. Yine k-ortalama hesaplaması eğer kümeler özellikle küresel ise hiyerarşik kümelemeden daha sıkı bir kümeleme yapacaktır.

Bunun yanında k-ortalama algoritmasının en büyük eksikliği k değerini tespit edememesidir. Bu nedenle başarılı bir kümeleme elde etmek için farklı k değerleri için deneme yanılma yönteminin uygulanması gerekmektedir. K-ortalama algoritmasının küresel kümelerde, her zaman doğru kümeleri bulamadığı ancak küme

sayısı doğru seçildiğinde ayırık ve sıkışık bulutlar şeklindeki kümeleri etkili bir şekilde bulabildiği söylenebilir.

3.9. K-Ortalama Kümeleme Uygulama Alanları

K-Ortalama kümeleme, görüntü bölümlene biçimi olarak bilgisayarla görü işlemlerinde kullanılmaktadır. Bölümlene sonucunda sınır tespiti ve nesne tanıma yardım için kullanılır. Bu çerçevede, standart Öklid hesaplaması genellikle yetersizdir. Bunun yerine, piksel kullanan ağırlıklı mesafe ölçümü koordinatları, RGB piksel, renkli ve yoğunluk ve görüntü dokusu yaygın olarak kullanılır. Bunun dışında K-Ortalama algoritması, veritabanlarında, market satışları kontrollerinde, ürün ayıklaması gibi işlemlerde de sıkça kullanılmaktadır.

BÖLÜM 4

4. YAPILAN İŞLEMLER

4.1. Giriş

Araç rotalama problemlerinde müşterilerin taleplerini gerçekleştirirken rotalamada kullanılacak araç sayısı, rotalama mesafesi, taleplerin karşılanma süresi ve buna benzer birçok maliyetin azaltılması amaçlanmaktadır.

NP zor bir problem olan araç rotalama probleminin karmaşıklığı problemde kullanılan girdilerin sayısı arttıkça artar. Bu nedenle aynı boyuttaki çok depolu araç rotalama problemlerine karınca kolonisi optimizasyonu kümeleme tekniği ile birlikte uygulanarak problem maliyetlerindeki değişim gözlenmiştir.

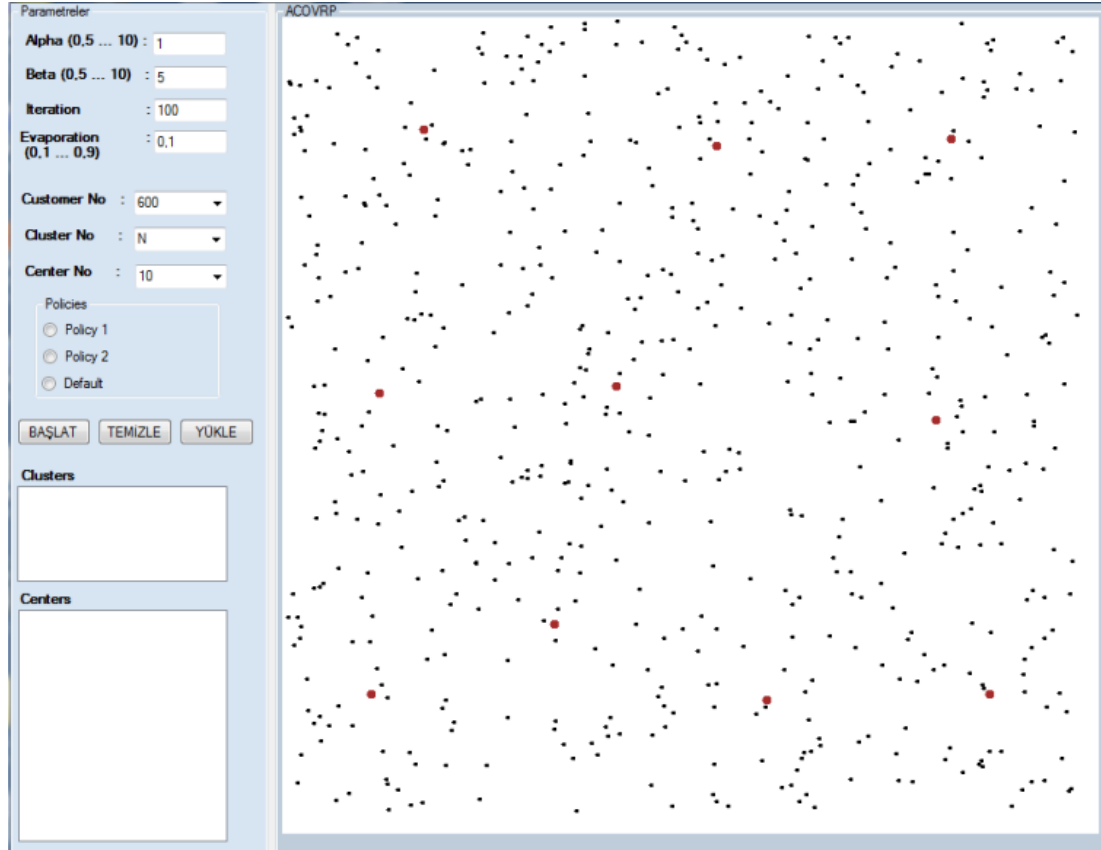
4.2. İşlem Basamakları

İşlem basamakları şu şekilde sıralanabilir.

1-İlk olarak algoritmada kullanılan ayar parametreleri seçilir. Uygulamada α , β , buharlaşma katsayısı gibi parametreler bulunmaktadır. Uygulamada $\alpha=1$, $\beta=5$ ve buharlaşma katsayısı 0,1 olarak seçilmiştir. Kapasiteli araç rotalama probleminde araç kapasitesi tüm araçlar için homojen ve 100 olarak belirlenmiştir.

2-Uygulamada belirlenen sayıdaki müşteri rastgele olarak dağıtılır ve her müşteriye (0,20] aralığında rastgele talep değerleri belirlenir. Talepler statiktir.

3-Belirlenen sayıdaki depo, k ortalama kümeleme işlemi kullanılarak müşterilere atanır ve uygulama çalıştırılır.



Şekil 4.1. Uygulama ekranının genel görüntüsü

Şekil 4.1 de görülen arayüzde kullanıcıdan Alpha, Beta, Iteration(iterasyon), Evaporation(buharlaştırma) değerleri istenir. Alpha ve Beta 1 ile 10 arasında değişen değerler alabilir. Evaporation(buharlaştırma) (0,1] arasında değerler alabilmektedir.

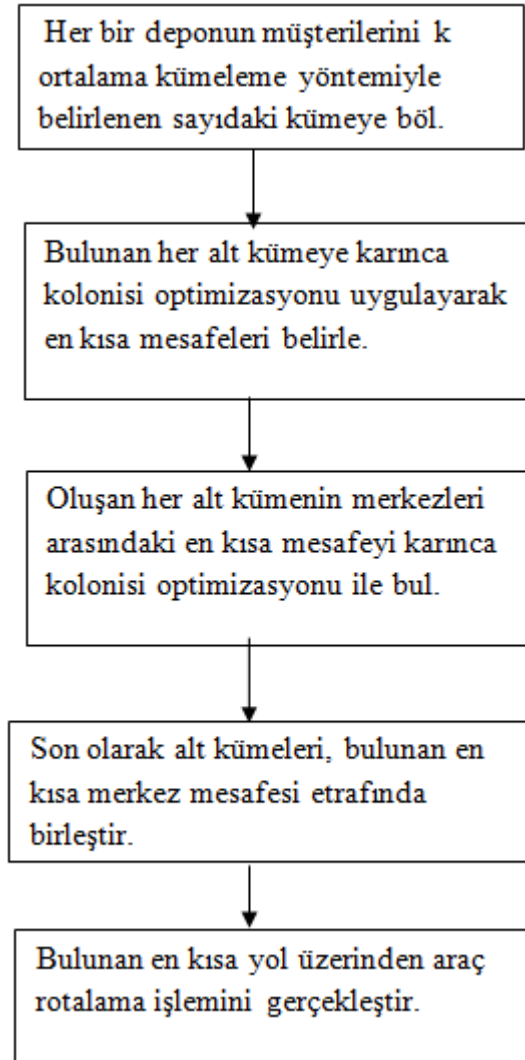
Customer No müşteri sayısını (600, 500, 400, 300, 200) temsil eder, kullanıcı bu değerlerden birisini seçtiğinde seçilen müşteri sayısı kadar nokta ekrana rastgele olarak dağıtılır. Bu noktalar ekranda siyah renkle gösterilmektedir.

Cluster No müşteri sayısının ayrılacağı küme sayısını ifade eder, bu alan N, N/2, N/4, N/8, SQRT(N), SQRT(N/2) değerlerini alır. Bu alanda herhangi bir değişiklik yapmaya gerek yoktur. Uygulama başladıktan sonra algoritma bu değerleri sırasıyla gezerek çözüm üretir.

Center No depo sayısını (10, 15, 20, 25) ifade eder. Depo sayısı seçildiğinde seçilen depo sayısı kadar nokta ekranda müşteri sayısına göre k ortalama kümeleme işlemi uygulanarak ekrana yerleştirilir. Bu noktalar ekranda kırmızı renkle gösterilmektedir.

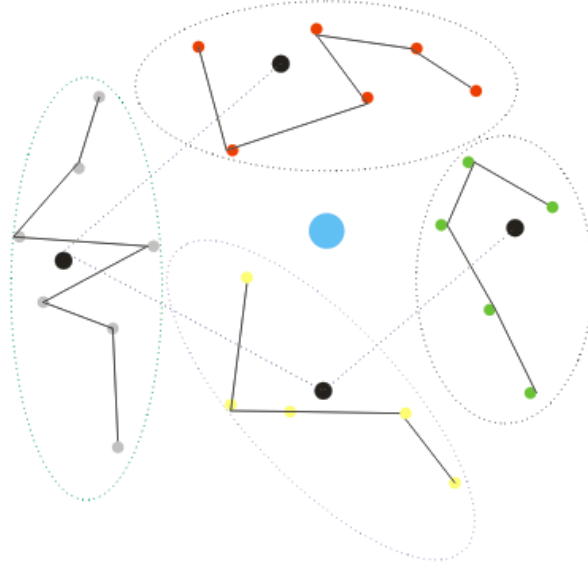
4-Uygulamada depolar ve onlara atanan müşteriler sırayla gezilerek karınca kolonisi algoritması uygulanır.

5-Tüm depolar sırayla gezilerek onlara atanan müşteriler sırasıyla $n, n/2, n/4, n/8, \sqrt{n}, \sqrt{n/2}$ kümeye ayrılarak her kümeye ayrı ayrı karınca kolonisi optimizasyonu uygulanmıştır. Burada n depoya atanan müşteri sayısını temsil etmektedir. Her bir depo ve ona atanan müşteriler için yapılan işlemler şu şekilde listelenebilir.



Şekil 4.2. Algoritma işlem basamakları

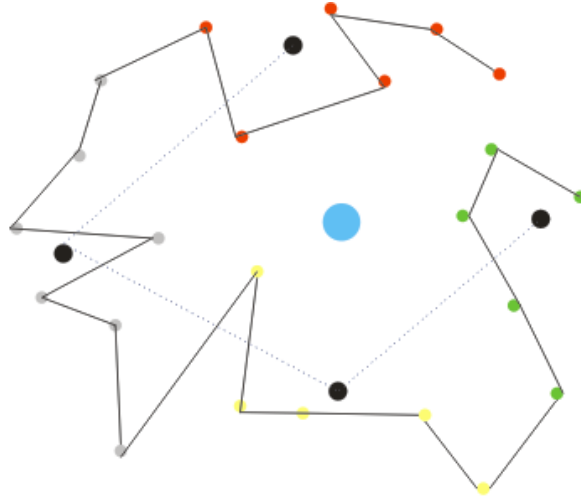
İlk olarak depoya atanmış müşteriler n , $n/2$, $n/4$, $n/8$, \sqrt{n} , $\sqrt{n/2}$ kümeye ayrılırlar. Yeni oluşan alt kümeler için karınca kolonisi optimizasyonu kullanılarak bu küme içinde kalan müşteriler arasındaki en kısa yol hesaplanır.



Şekil 4.3. Alt Kümelerin ve alt küme merkezlerinin dağılımı

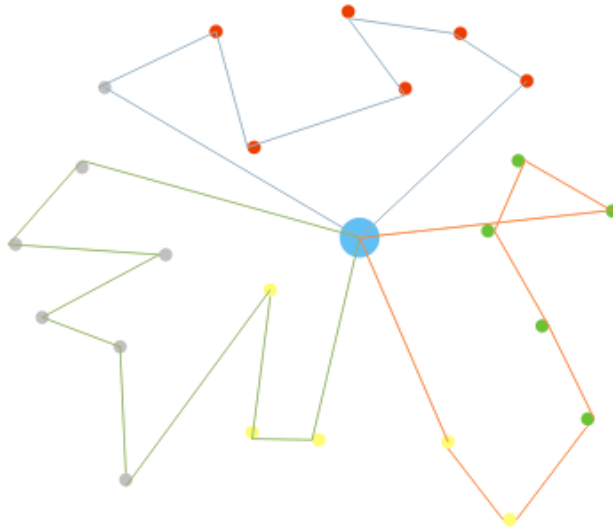
Şekil 4.3 de görülen örnekte veriler dört alt kümeye ayrılmış ve bu kümeler içerisinde kalan müşteriler arasındaki en kısa mesafeler gösterilmiştir. Daha sonra alt kümelerin merkezleri arasındaki en kısa rotalar belirlenmiştir.

Alt kümelerdeki en kısa yolun hesaplanmasının ardından bu alt kümelerin merkezleri arasındaki en kısa yollarda karınca kolonisi optimizasyonu ile hesaplanır ve alt kümelerde bulunan yollar, alt kümelerin merkez noktaları arasındaki en kısa yol doğrultusunda birleştirilir.



Şekil 4.4. Alt kümelerin alt küme merkezleri etrafında birleştirilmesi

Alt kümelerdeki yolların birleştirilmesinin ardından bulunan en kısa yol üzerinden araç kapasitesi dolana kadar ilerlenir ve araç kapasitesi dolduktan sonra depoya geri dönülerek, araç boşaltılır ve aynı işlem tüm müşterilerin talepleri toplanana kadar devam eder.



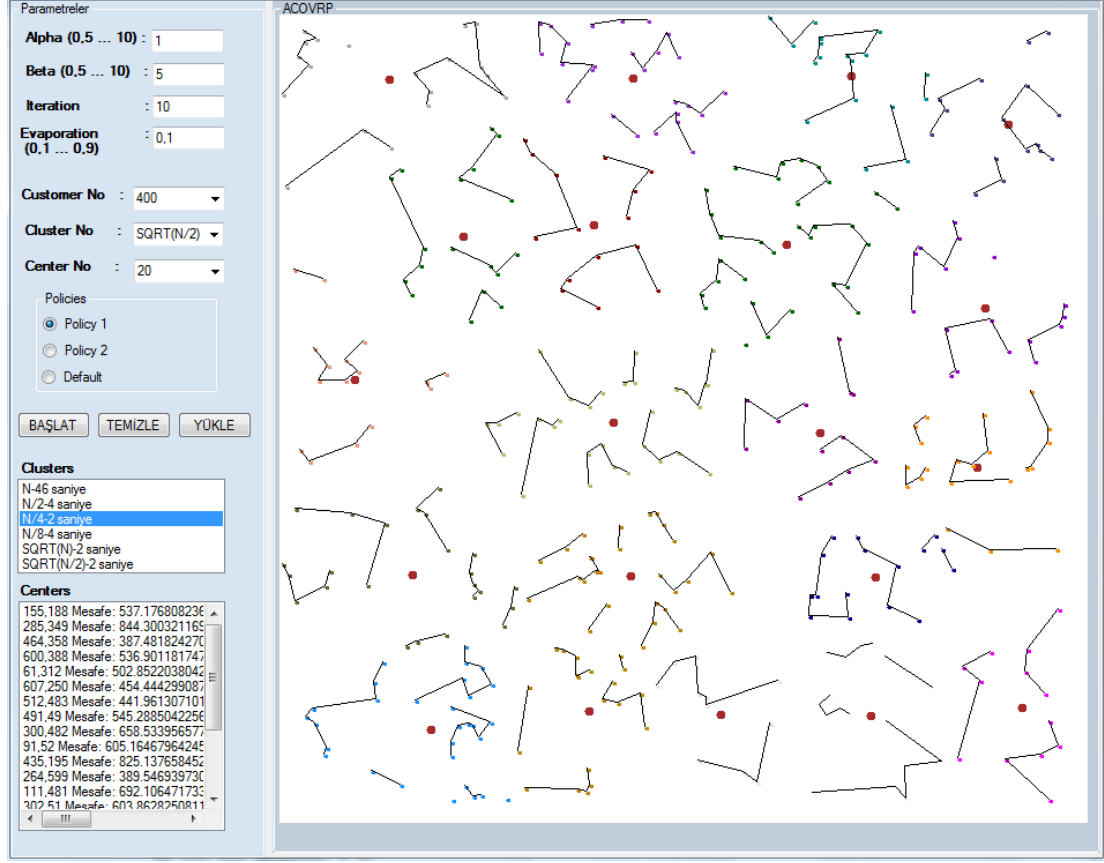
Şekil 4.5. Rotalama işleminin ekran görüntüsü

Bu işlem adımları tüm depolar ve onlara atanan müşteriler üzerinde gerçekleştirilir. Öncelikle depoya atanan müşteriler n adet kümeye (yani her bir müşteri bir kümeyi temsil edecek şekilde) ve sırasıyla $n/2, n/4, n/8, \sqrt{n}$ ve $\sqrt{n/2}$ kümeye

ayrılarak çözümler. Kümeleme işleminde müşteriler arasındaki mesafe hesaplanırken öklit mesafesi kullanılmıştır. Bu işlemlerin dışında bu kümeleme işlemlerinde bir depoya ait farklı kümeleme işlemlerinin bulunduğu sonuçlardan en iyisi de arka planda saklanır. Bu depo üzerinde kümeleme işlemlerinden en iyi sonuçları bulan kümelerin sonuçları hibrit bir yaklaşım olarak kullanılmıştır. Bu işlemler sonucunda depo ve müşteri sayısı sabitken iterasyon sayısı değiştirilmiş, depo ve iterasyon sayısı sabitken müşteri sayısı değiştirilmiş, müşteri ve iterasyon sayısı sabitken depo sayısı değiştirilmiştir ve ortalama toplam mesafe, tur için kullanılan toplam araç sayısı ve toplam işlem sürelerindeki değişimler kaydedilmiştir. Şekilde görüldüğü gibi uygulama üzerinden her kümelemede bulunan sonuçların ekran görüntülerine, depo bazında toplam mesafe, toplam araç sayısı ve toplam süre bilgilerine ulaşabilmektedir.

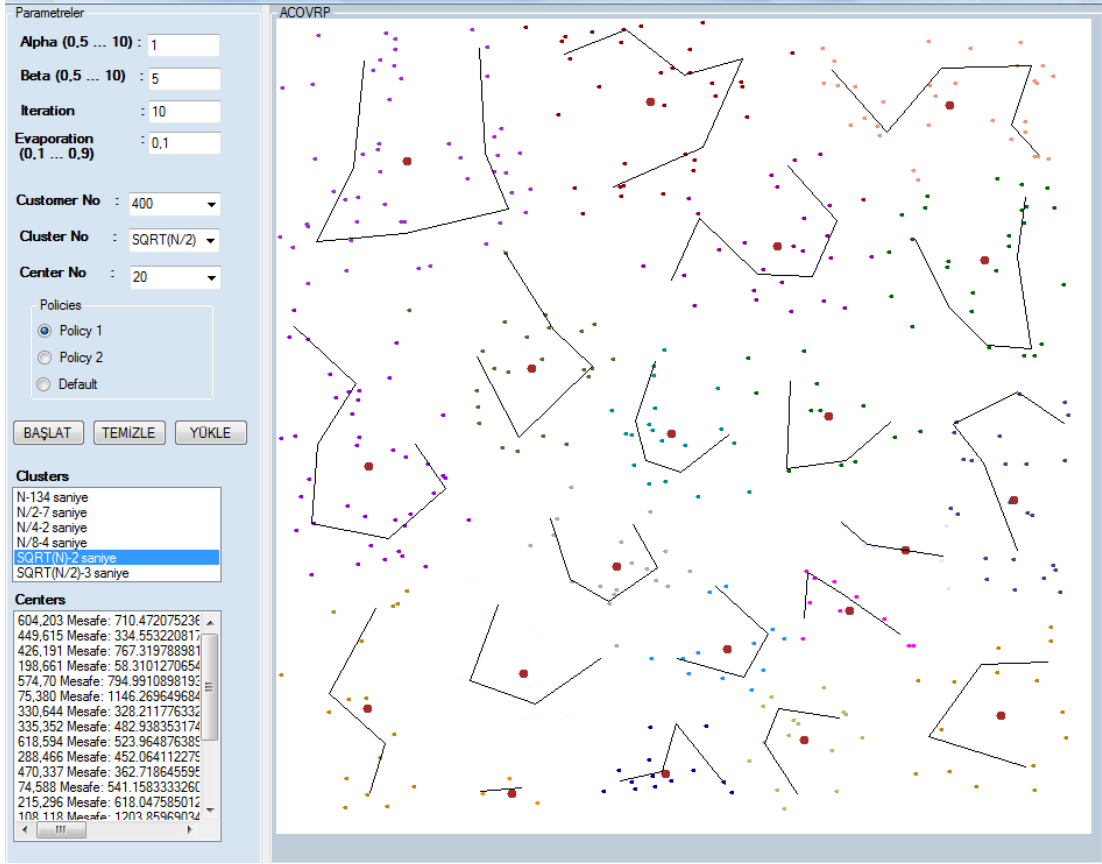
Seçilen depo ekran üzerinde farklı bir renkle gösterilmektedir. Seçilen depoya ait bilgilerde ekran üzerinde gözlemlenebilir. Uygulama kullanıcılara karşılaştırmalı bir ara yüz sağlamaktadır.

4.3. Uygulama Ekran Görüntüleri



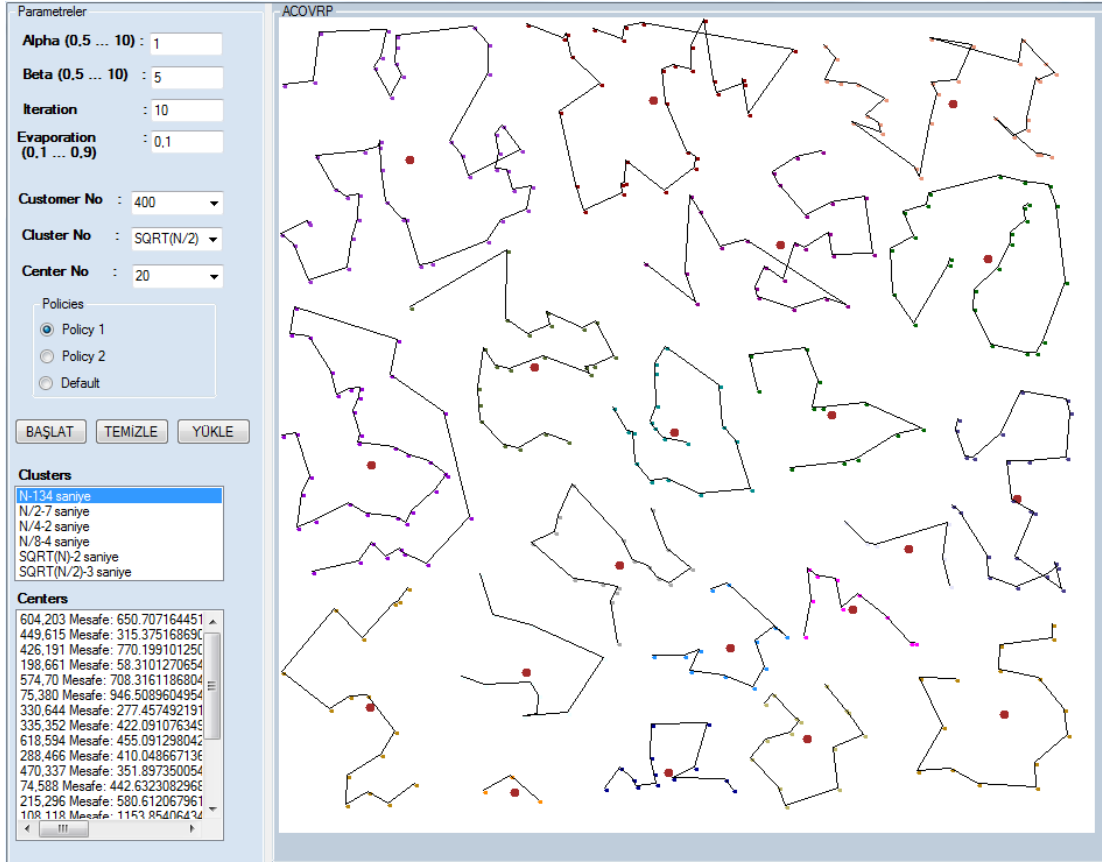
Şekil 4.6. Kümelerin alt kümelere ayrılması

Şekil 4.6 da 400 müşterili,20 depolu bir araç rotalama probleminde her depoya atanan müşteriler $\sqrt{n/2}$ kümeye bölünmüş ve her alt kümeye düşen müşteriler arasındaki en kısa mesafeler karınca kolonisi optimizasyonu ile hesaplanmıştır.



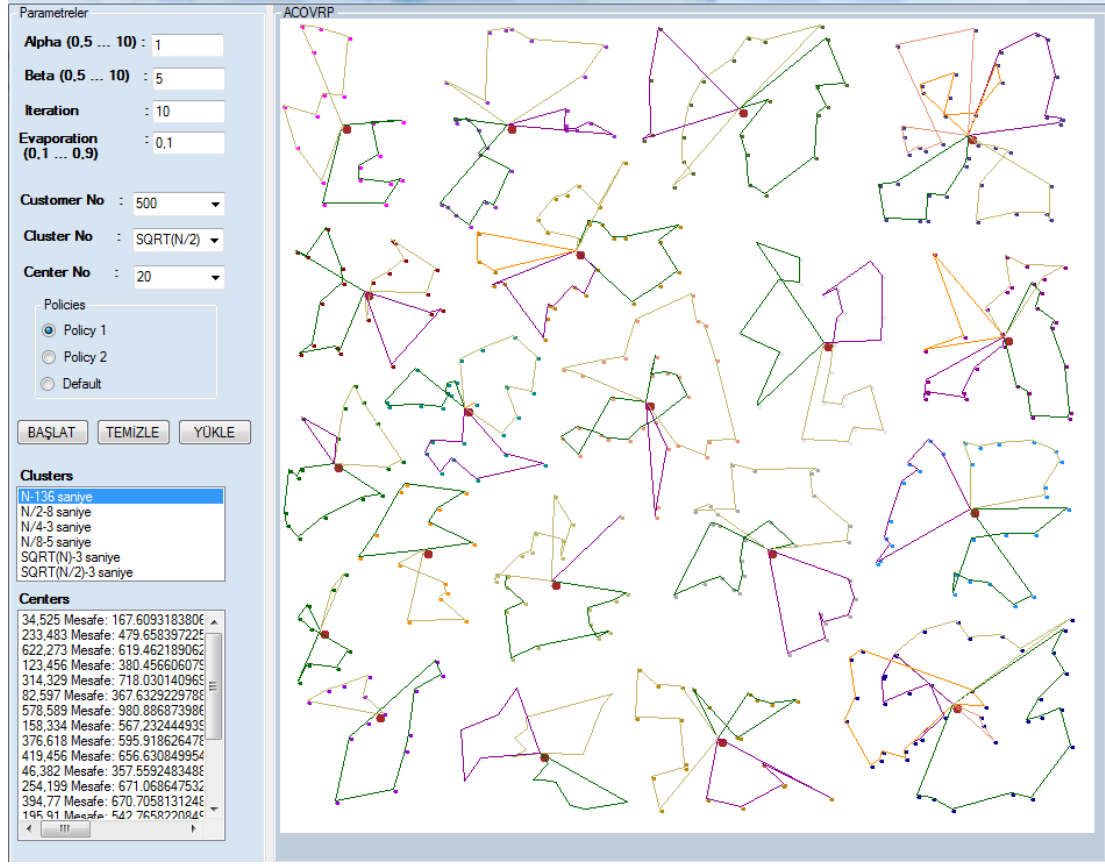
Şekil 4.7. Alt küme merkezleri arasındaki en kısa mesafelerin hesaplanması

Şekil 4.7 de 400 müşterili,20 depolu bir araç rotalama probleminde her depoya atanan müşteriler $\sqrt{n/2}$ kümeye bölünmüş ve küme merkezleri arasındaki mesafeler karınca kolonisi optimizasyonu ile hesaplanmıştır.



Şekil 4.8. Alt kümelerin merkez noktaları etrafında birleştirilmesi

Şekil 4.8 de 400 müşterili,20 depolu bir araç rotalama probleminde her depoya atanan müşteriler $\sqrt{n/2}$ kümeye bölünmüş ve küme merkezleri arasındaki mesafeler karınca kolonisi optimizasyonu ile hesaplanmıştır. Sonra bu alt kümelere bulunan en kısa yollar, bulunan alt küme merkez noktaları etrafında birleştirilmiştir.



Şekil 4.9. Bulunan en kısa yollar üzerinden araç rotalama işleminin gerçekleştirilmesi

Şekil 4.9 da 500 müşterili,20 depolu bir araç rotalama probleminde her depoya atanan müşteriler $\sqrt{n/2}$ kümeye bölünmüş ve küme merkezleri arasındaki mesafeler karınca kolonisi optimizasyonu ile hesaplanmıştır. Sonra bu alt kümelerde bulunan en kısa yollar, bulunan alt küme merkez noktaları etrafında birleştirilmiştir. Sonra da bulunan bu yollar üzerinden araç kapasitesi dolana kadar ilerlenir ve araç kapasitesi dolduktan sonra depoya geri dönülerek, araç boşaltılır ve aynı işlem tüm müşterilerin talepleri toplanana kadar devam eder.

BÖLÜM 5

5. UYGULAMA ÇIKTILARI

Yapılan çalışmalarda çok depolu araç rotalama problemlerine karınca kolonisi optimizasyonu k ortalama kümeleme tekniği ile birlikte uygulanarak elde edilen sonuçlar değerlendirilmiştir. Her depoya atanılan müşteriler sırasıyla n , $n/2$, $n/4$, $n/8$, \sqrt{n} , $\sqrt{n/2}$ kümeye ayrılarak, bu kümeleme işleminde bulunan kayıtlar grafikte gösterilmiştir. Ayrıca bu kümeleme işlemlerinin hibrit olarak uygulandığı en iyileme işleminde bulunan sonuçlar da grafikte gösterilmiştir.

Uygulama üzerinden

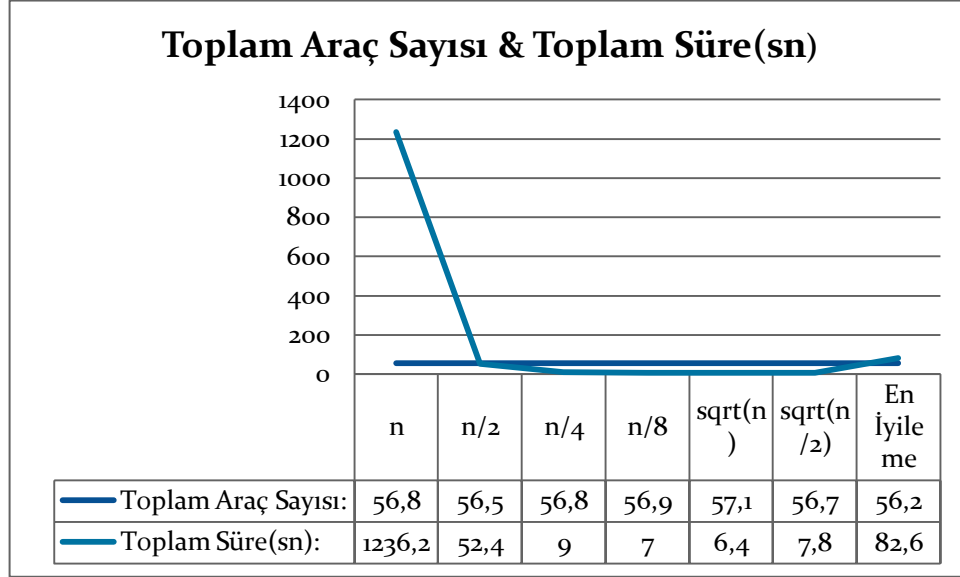
- Depo ve müşteri sayısı sabitken iterasyon sayısı arttırılmış,
- Depo ve iterasyon Sayısı sabitken müşteri sayısı arttırılmış,
- Müşteri sayısı ve iterasyon sayısı sabitken depo sayısı arttırılmıştır.

Elde edilen sonuçlar şu şekildedir.

5.1. Depo ve Müşteri Sayısı Sabitken İterasyon Sayısının Arttırılması

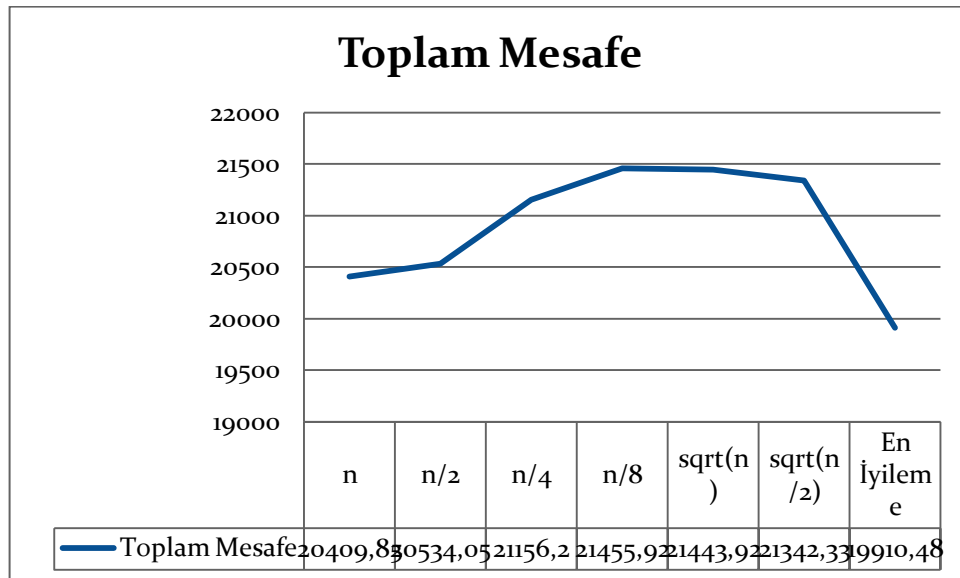
Uygulama üzerinden depo sayısı 10, müşteri sayısı 500 ve iterasyon sayısı 20, 40, 60, 80, 100 seçilerek toplam araç sayısı, toplam süre ve toplam mesafedeki değişimler kaydedilmiştir. Her bir işlem 10 kez arka arkaya gerçekleştirilmiş ve elde edilen sonuçların ortalaması alınarak grafiğe aktarılmıştır. Elde edilen sonuçların yapılan kümeleme işlemlerine göre dağılımları şu şekildedir.

İterasyon:20, Depo Sayısı:10 Müşteri Sayısı:500



Şekil 5.1 Araç Sayısı ve Sürenin Kümelemeye Göre Değişimi

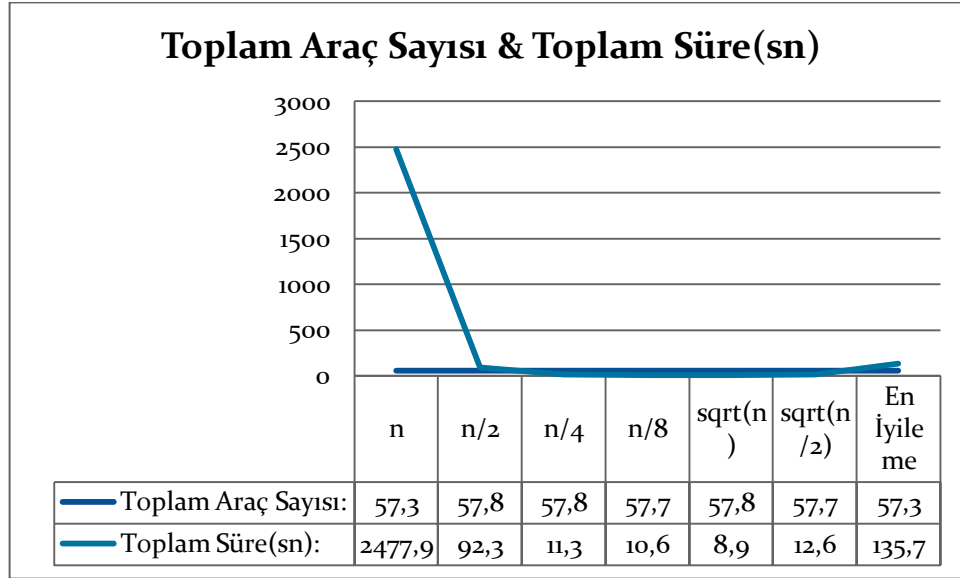
(20 iterasyon,10 depo,500 müşteri)



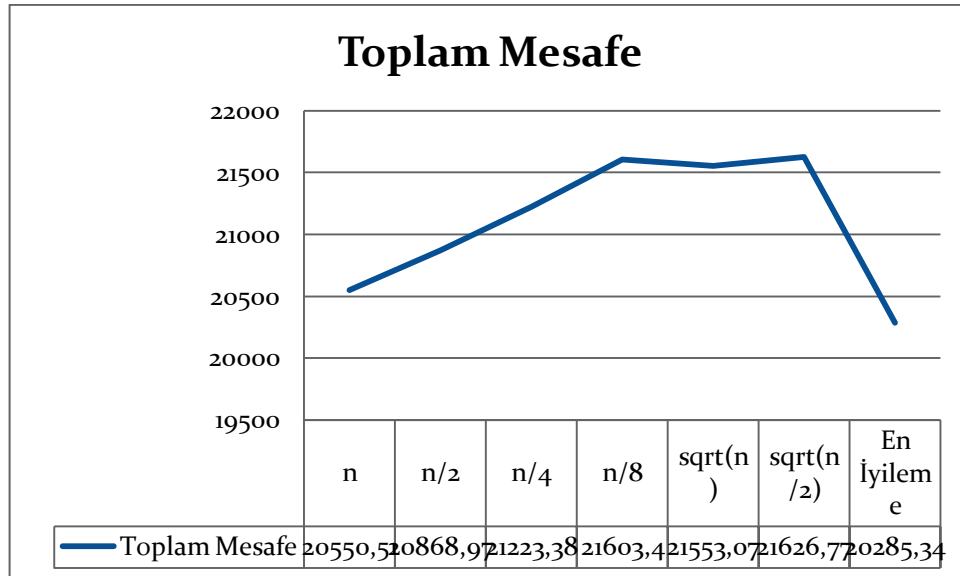
Şekil 5.2 Toplam Mesafenin Kümelemeye Göre Değişimi

(20 iterasyon,10 depo,500 müşteri)

İterasyon:40, Depo Sayısı:10 Müşteri Sayısı:500

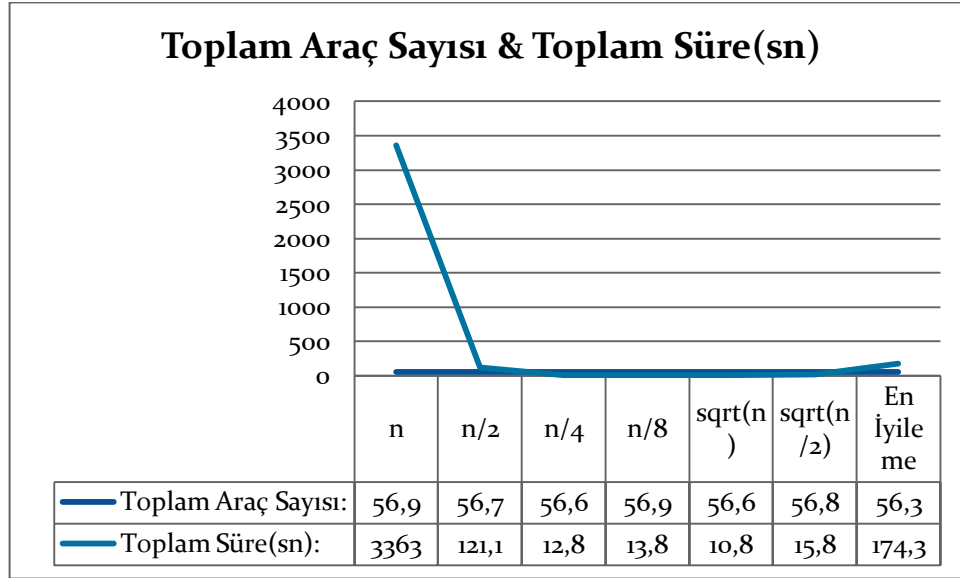


Şekil 5.3 Araç Sayısı ve Sürenin Kümelemeye Göre Değişimi
(40 iterasyon,10 depo,500 müşteri)

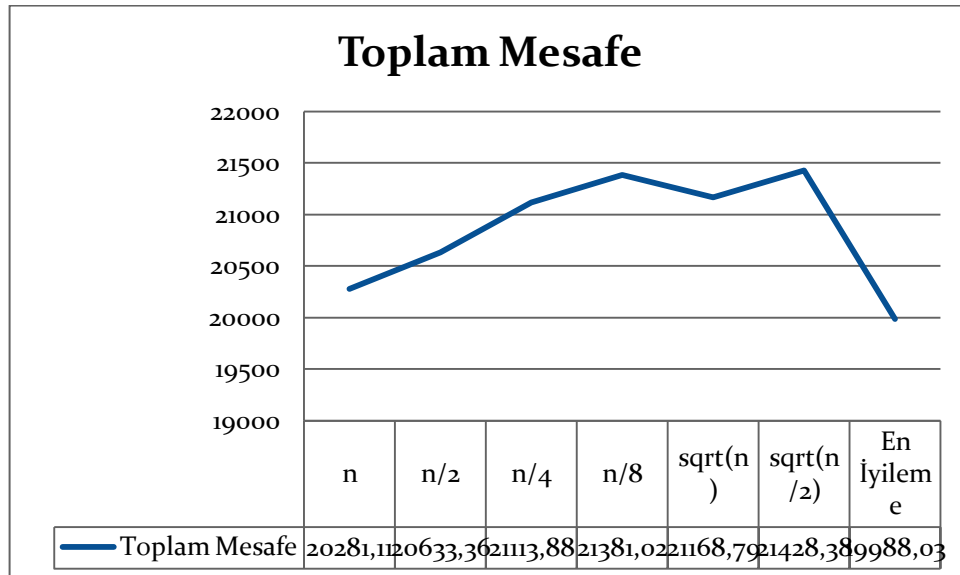


Şekil 5.4 Toplam Mesafenin Kümelemeye Göre Değişimi
(40 iterasyon,10 depo,500 müşteri)

İterasyon:60, Depo Sayısı:10 Müşteri Sayısı:500

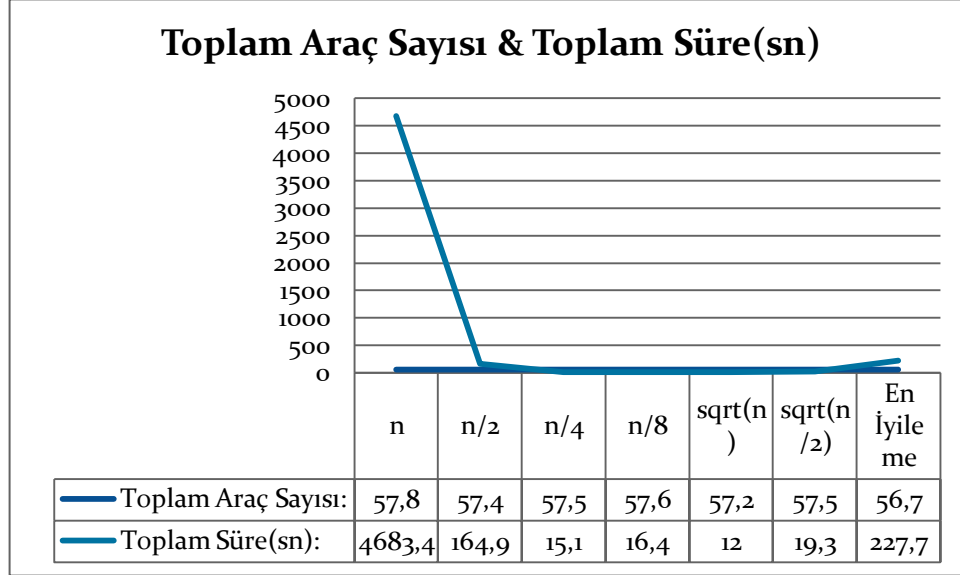


Şekil 5.5 Araç Sayısı ve Sürenin Kümelemeye Göre Değişimi
(60 iterasyon,10 depo,500 müşteri)

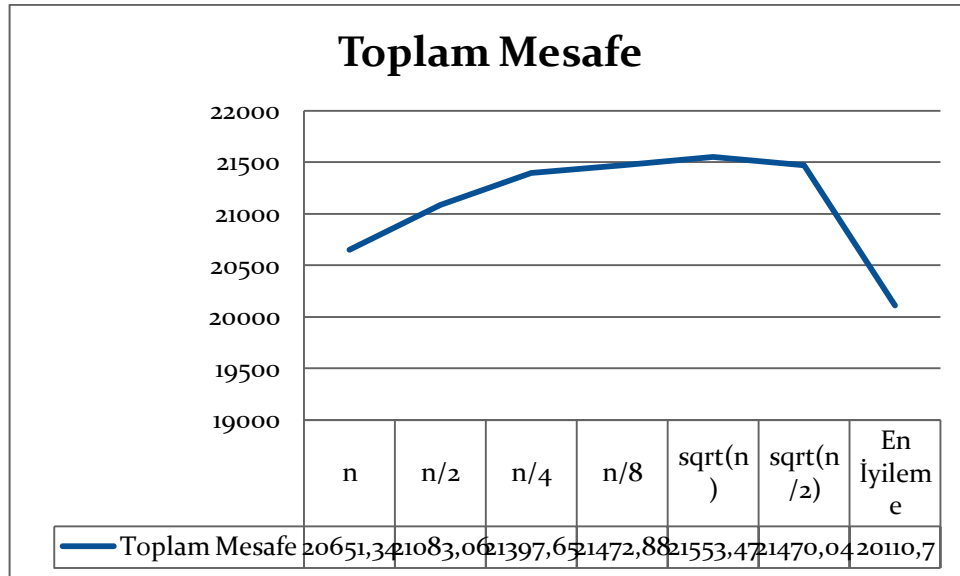


Şekil 5.6 Toplam Mesafenin Kümelemeye Göre Değişimi
(60 iterasyon,10 depo,500 müşteri)

İterasyon:80, Depo Sayısı:10 Müşteri Sayısı:500

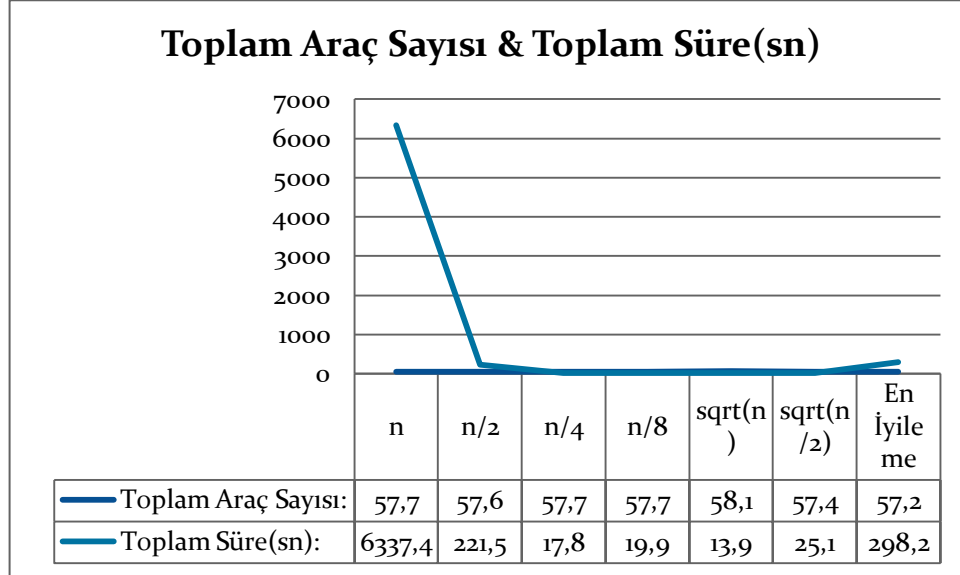


Şekil 5.7 Araç Sayısı ve Sürenin Kümelemeye Göre Değişimi
(80 iterasyon,10 depo,500 müşteri)

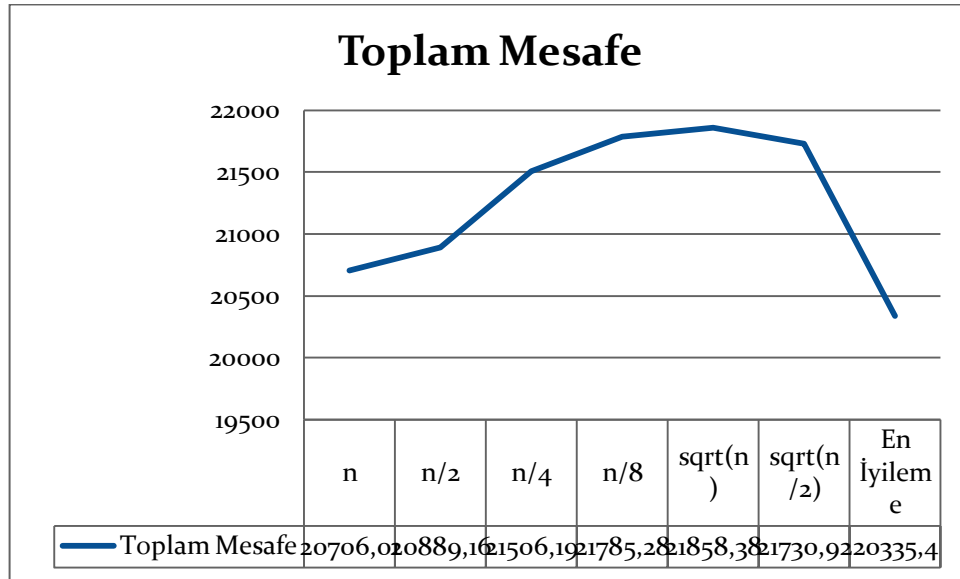


Şekil 5.8 Toplam Mesafenin Kümelemeye Göre Değişimi
(80 iterasyon,10 depo,500 müşteri)

İterasyon:100, Depo Sayısı:10 Müşteri Sayısı:500



Şekil 5.9 Araç Sayısı ve Sürenin Kümelemeye Göre Değişimi
(100 iterasyon,10 depo,500 müşteri)



Şekil 5.10 Toplam Mesafenin Kümelemeye Göre Değişimi
(100 iterasyon,10 depo,500 müşteri)

Depo sayısı ve müşteri sayısı sabitken iterasyon sayısı artırılmış toplam araç sayısı, toplam rotalama süresi ve toplam mesafenin değişimleri gözlenmiştir. Elde edilen verilere göre Toplam rotalama süresi en iyi \sqrt{n} kümelemede elde edilmiştir. Yani depoya atanmış müşterilerin karekökü kadar k ortalama kümeye ayırma işleminde rotalama süresinde optimum sonuç elde edilmiştir. Yine elde edilen verilere bakıldığında kümeleme yapılarak elde edilen rotalama süresi kümeleme yapılmadan elde edilen rotalama süresinden çok daha iyi sonuçlar verdiği görülmüştür.

Toplam araç sayısı en iyi hibrit olarak kullanılan en iyileme işlemiyle bulunmuştur. En iyileme işleminde her depo sırasıyla n , $n/2$, $n/4$, $n/8$, \sqrt{n} , $\sqrt{n/2}$ kümeye ayrılır ve o depo için en iyi sonucu üreten kümenin bulunduğu değer alınır. Yani çok depolu araç rotalama probleminde tüm depolar gezildiğinde en iyileme yönteminde bulunan en iyi sonuç farklı bir kümeleme yöntemi ile bulunmuş olur. En iyileme yöntemi kümeleme yapılmadan uygulanan karınca kolonisi optimizasyonunun bulunduğu toplam araç sayısına göre daha iyi sonuçlar üretmiştir.

En iyileme dışında toplam araç sayısı $n/2$ ve \sqrt{n} kümelemede diğer kümeleme işlemlerinden daha iyi sonuçlar üretmiştir.

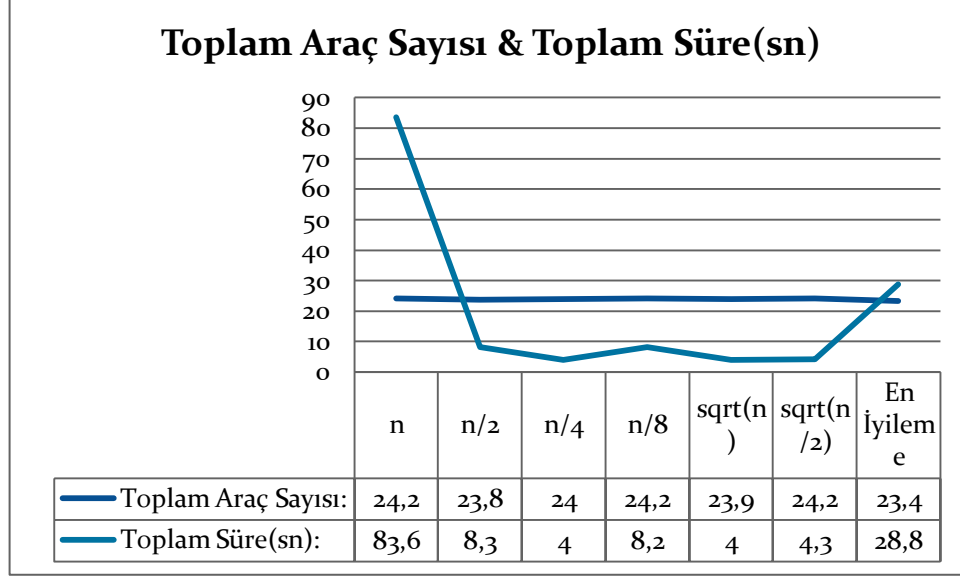
Toplam mesafe de en iyi yine hibrit olarak kullanılan en iyileme işlemi ile bulunmuştur. En iyileme yöntemi kümeleme yapılmadan uygulanan karınca kolonisi optimizasyonunun bulunduğu toplam mesafeye göre daha iyi sonuçlar üretmiştir.

En iyilemenin dışında en iyi sonuç $n/2$ kümeleme işleminde elde edilmiştir.

5.2. Depo ve İterasyon Sayısı Sabitken Müşteri Sayısının Arttırılması

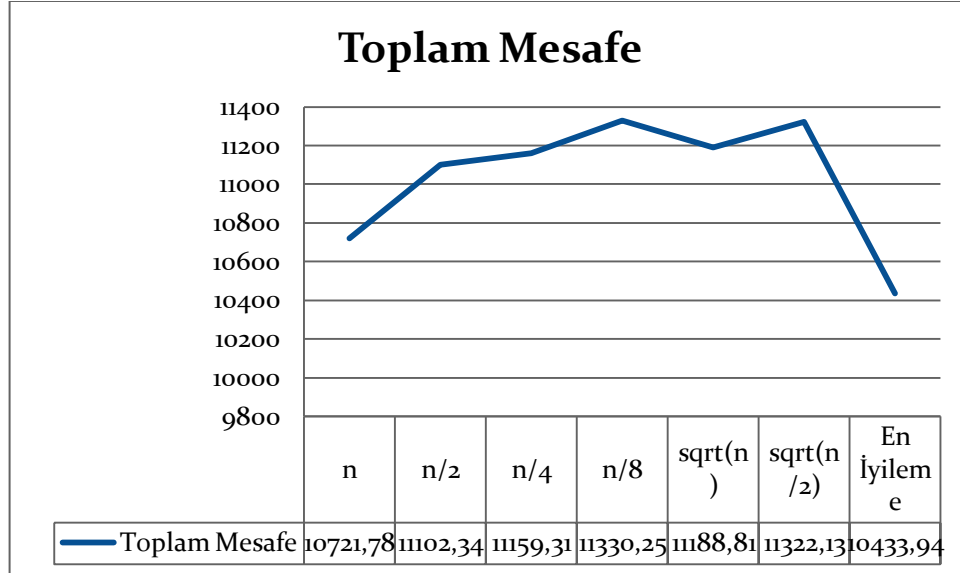
Uygulama üzerinden depo sayısı 10, iterasyon sayısı 100 ve müşteri sayısı 200, 300, 400, 500, 600 seçilerek toplam araç sayısı, toplam süre ve toplam mesafedeki değişimler kaydedilmiştir. Her bir işlem 10 kez arka arkaya gerçekleştirilmiş ve elde edilen sonuçların ortalaması alınarak grafiğe aktarılmıştır. Elde edilen sonuçların yapılan kümeleme işlemlerine göre dağılımları şu şekildedir.

İterasyon:100, Depo Sayısı:10 Müşteri Sayısı:200



Şekil 5.11 Araç Sayısı ve Sürenin Kümelemeye Göre Değişimi

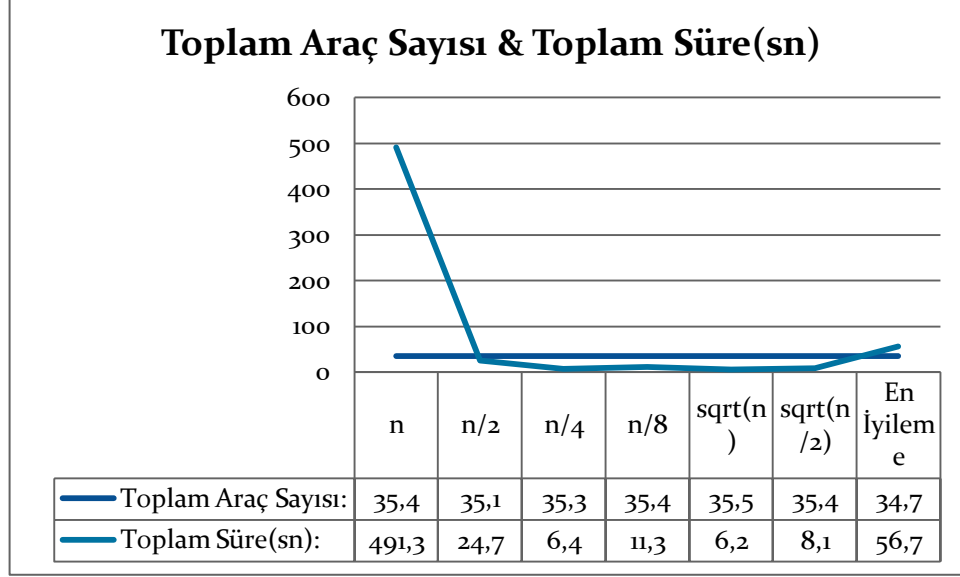
(100 iterasyon,10 depo,200 müşteri)



Şekil 5.12 Toplam Mesafenin Kümelemeye Göre Değişimi

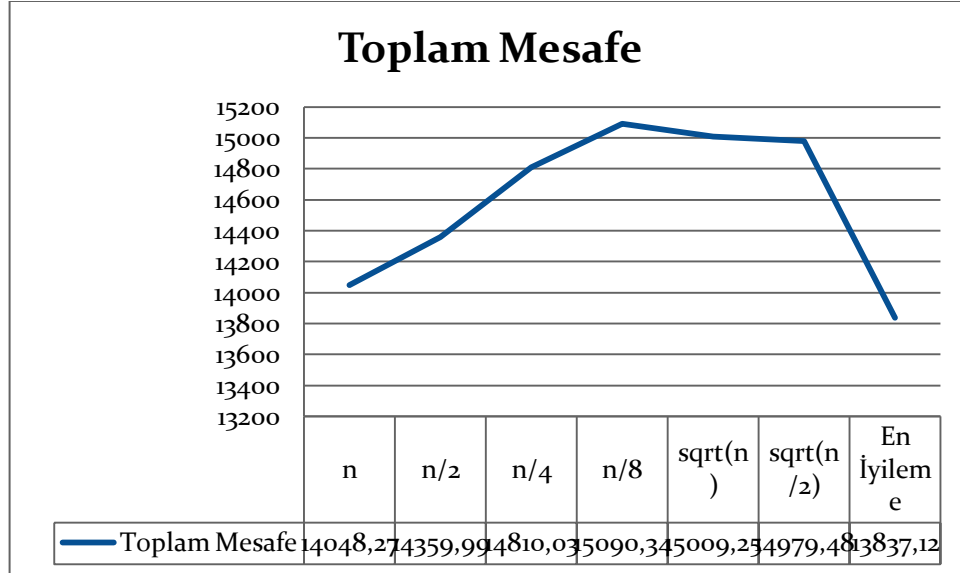
(100 iterasyon,10 depo,200 müşteri)

İterasyon:100, Depo Sayısı:10 Müşteri Sayısı:300



Şekil 5.13 Araç Sayısı ve Sürenin Kümelemeye Göre Değişimi

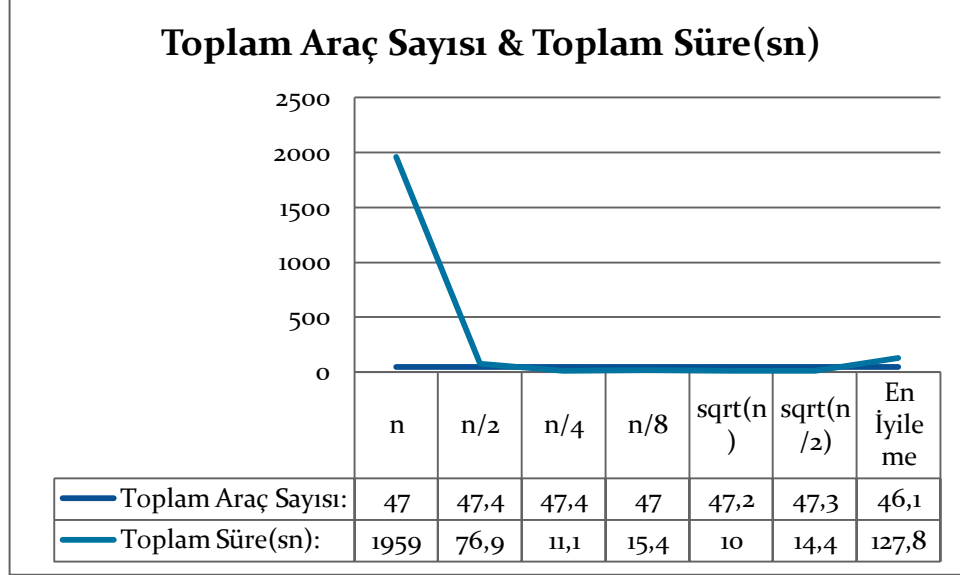
(100 iterasyon,10 depo,300 müşteri)



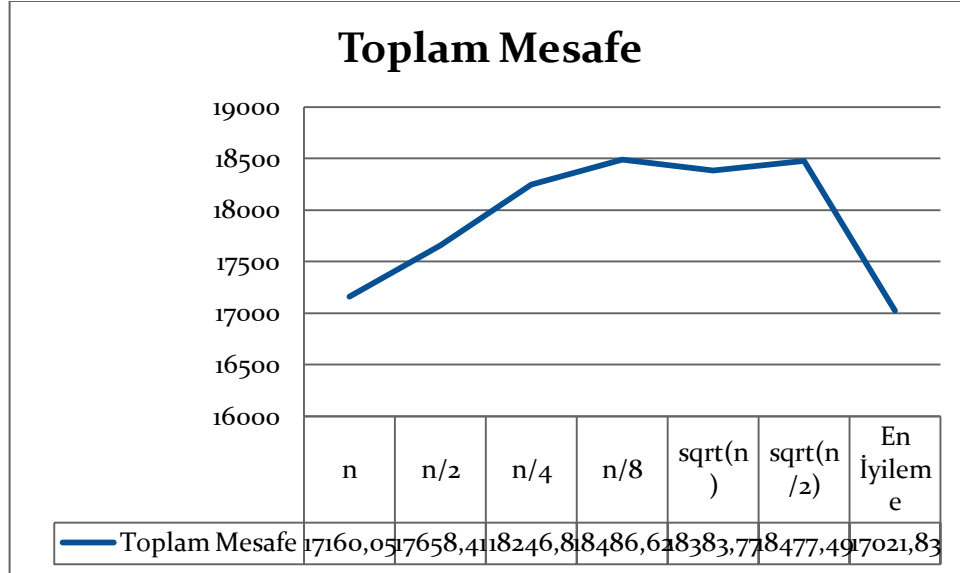
Şekil 5.14 Toplam Mesafenin Kümelemeye Göre Değişimi

(100 iterasyon,10 depo,300 müşteri)

İterasyon:100, Depo Sayısı:10 Müşteri Sayısı:400

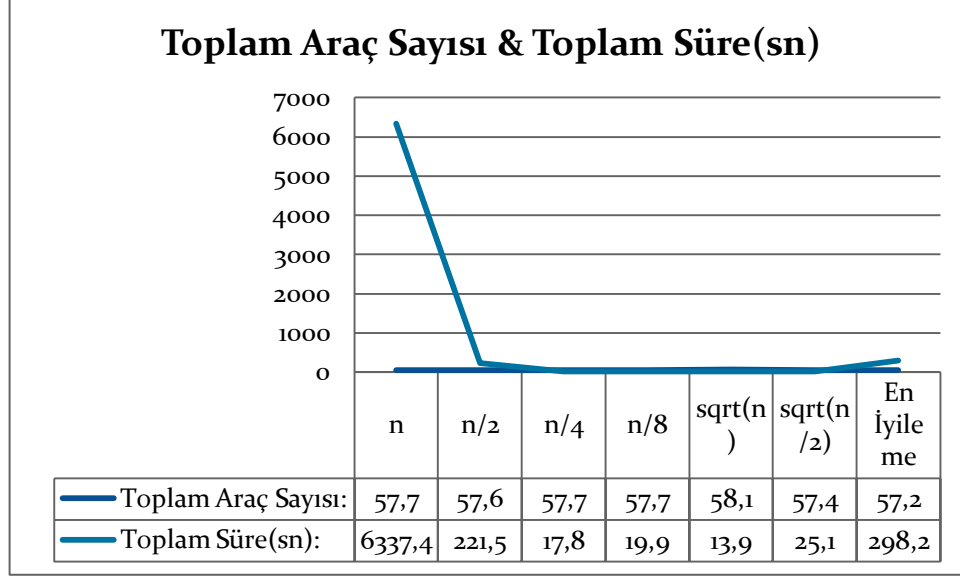


Şekil 5.15 Araç Sayısı ve Sürenin Kümelemeye Göre Değişimi
(100 iterasyon,10 depo,400 müşteri)

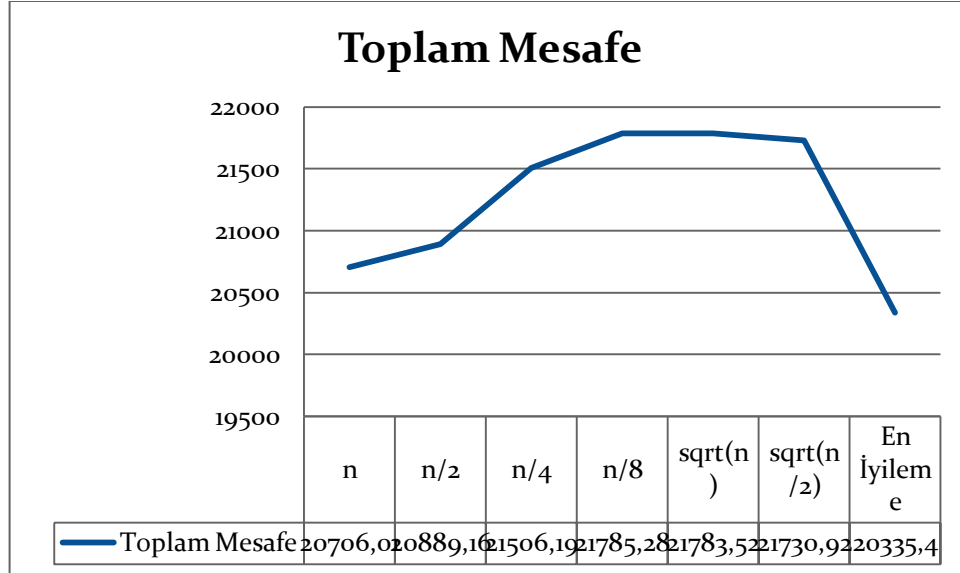


Şekil 5.16 Toplam Mesafenin Kümelemeye Göre Değişimi
(100 iterasyon,10 depo,400 müşteri)

İterasyon:100, Depo Sayısı:10 Müşteri Sayısı:500

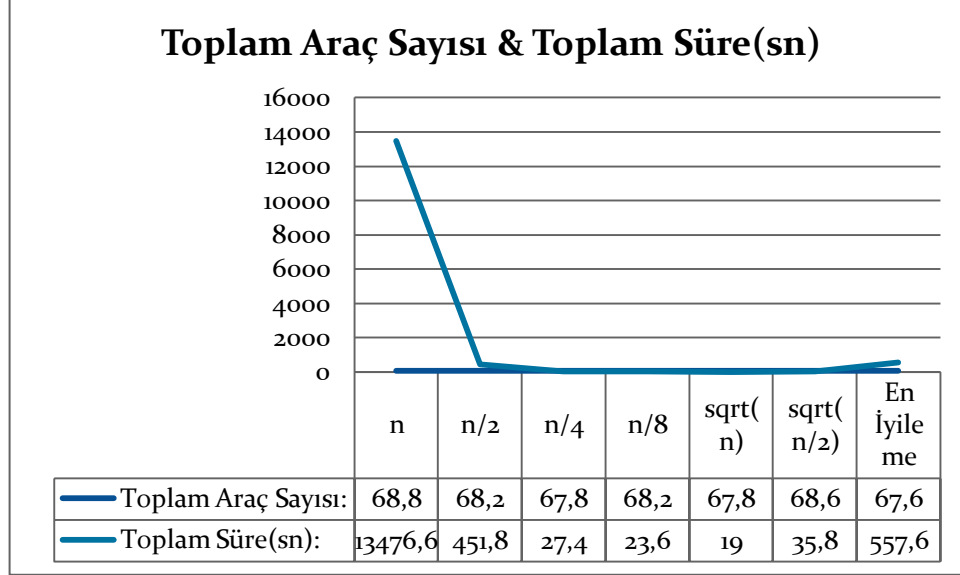


Şekil 5.17 Araç Sayısı ve Sürenin Kümelemeye Göre Değişimi
(100 iterasyon,10 depo,500 müşteri)

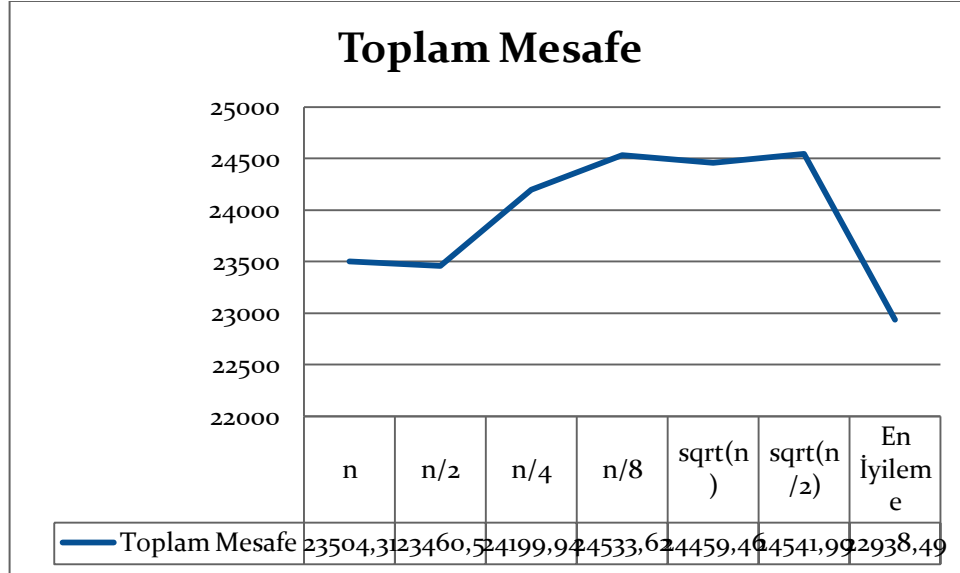


Şekil 5.18 Toplam Mesafenin Kümelemeye Göre Değişimi
(100 iterasyon,10 depo,500 müşteri)

İterasyon:100, Depo Sayısı:10 Müşteri Sayısı:600



Şekil 5.19 Araç Sayısı ve Sürenin Kümelemeye Göre Değişimi
(100 iterasyon,10 depo,600 müşteri)



Şekil 5.20 Toplam Mesafenin Kümelemeye Göre Değişimi
(100 iterasyon,10 depo,600 müşteri)

Depo sayısı ve iterasyon sayısı sabitken müşteri sayısı arttırılmış toplam araç sayısı, toplam rotalama süresi ve toplam mesafenin değişimleri gözlenmiştir. Elde edilen verilere göre Toplam rotalama süresi en iyi \sqrt{n} kümelemede elde edilmiştir. Yani depoya atanmış müşterilerin karekökü kadar k ortalama kümeye ayırma işleminde rotalama süresinde optimum sonuç elde edilmiştir. Yine elde edilen verilere bakıldığında kümeleme yapılarak elde edilen rotalama süresi kümeleme yapılmadan elde edilen rotalama süresinden çok daha iyi sonuçlar verdiği görülmüştür.

Toplam araç sayısı en iyi hibrit olarak kullanılan en iyileme işlemiyle bulunmuştur. En iyileme yöntemi kümeleme yapılmadan uygulanan karınca kolonisi optimizasyonunun bulduğu toplam araç sayısına göre daha iyi sonuçlar üretmiştir.

En iyileme dışında toplam araç sayısı $n/2$ ve $n/8$ kümelemede diğer kümeleme işlemlerinden daha iyi sonuçlar üretmiştir.

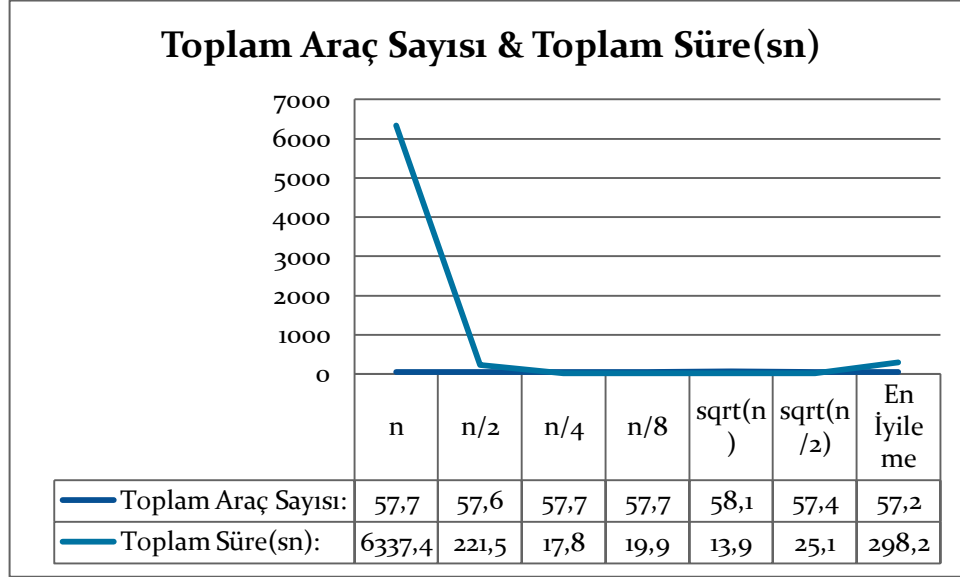
Toplam mesafe de en iyi yine hibrit olarak kullanılan en iyileme işlemi ile bulunmuştur. En iyileme yöntemi kümeleme yapılmadan uygulanan karınca kolonisi optimizasyonunun bulduğu toplam mesafeye göre daha iyi sonuçlar üretmiştir.

En iyilemenin dışında en iyi sonuç $n/2$ kümeleme işleminde elde edilmiştir.

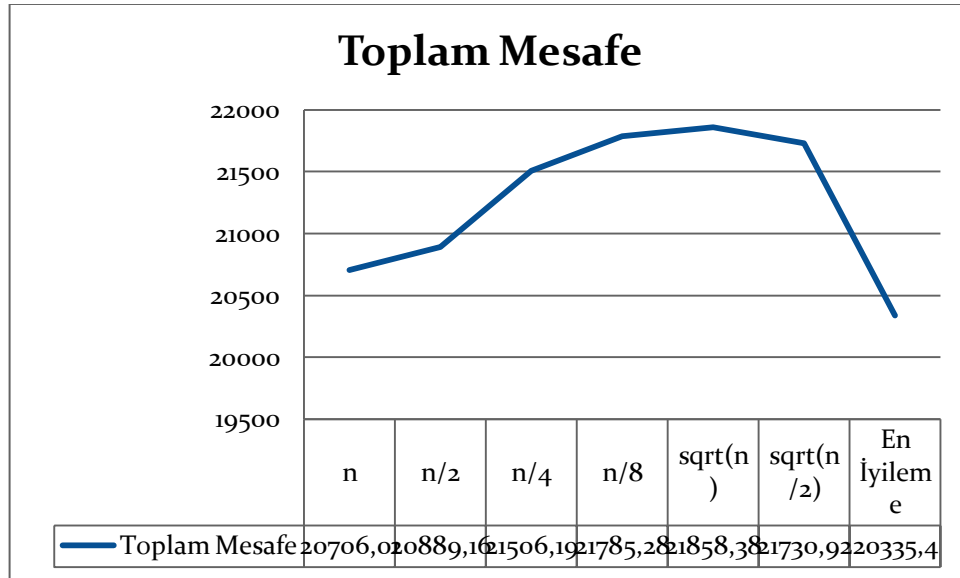
5.3. Müşteri ve İterasyon Sayısı Sabitken Depo Sayısının Arttırılması

Uygulama üzerinden iterasyon sayısı 100, müşteri sayısı 500 ve depo sayısı 10, 15, 20, 25 seçilerek toplam araç sayısı, toplam süre ve toplam mesafedeki değişimler kaydedilmiştir. Her bir işlem 10 kez arka arkaya gerçekleştirilmiş ve elde edilen sonuçların ortalaması alınarak grafiğe aktarılmıştır. Elde edilen sonuçların yapılan kümeleme işlemlerine göre dağılımları şu şekildedir.

İterasyon:100, Depo Sayısı:10 Müşteri Sayısı:500

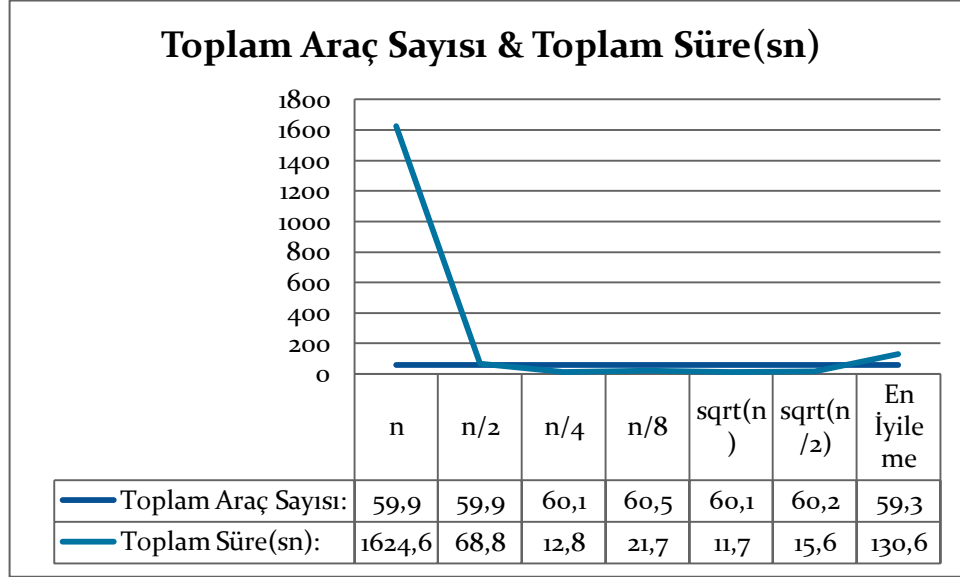


Şekil 5.21 Araç Sayısı ve Sürenin Kümelemeye Göre Değişimi
(100 iterasyon,10 depo,500 müşteri)

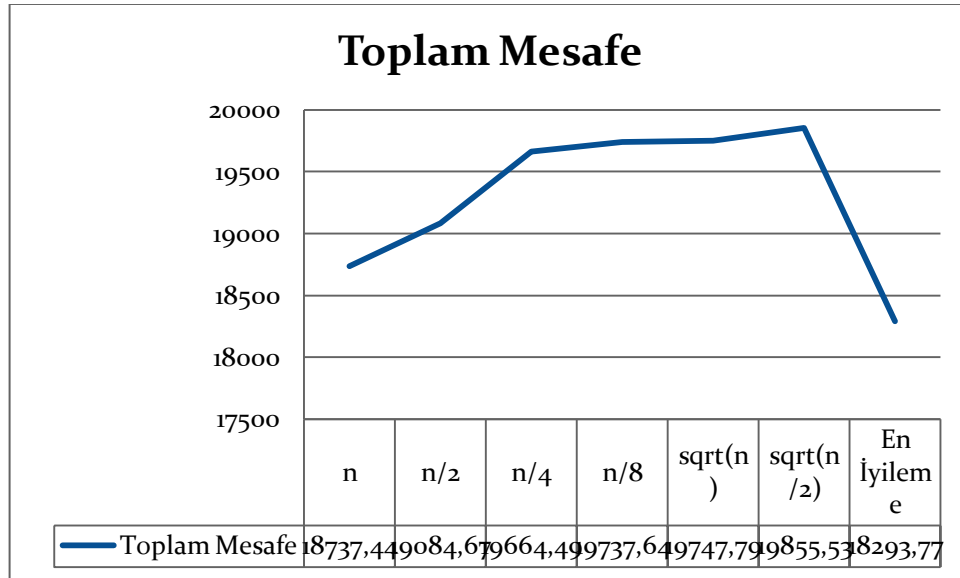


Şekil 5.22 Toplam Mesafenin Kümelemeye Göre Değişimi
(100 iterasyon,10 depo,500 müşteri)

İterasyon:100, Depo Sayısı:15 Müşteri Sayısı:500

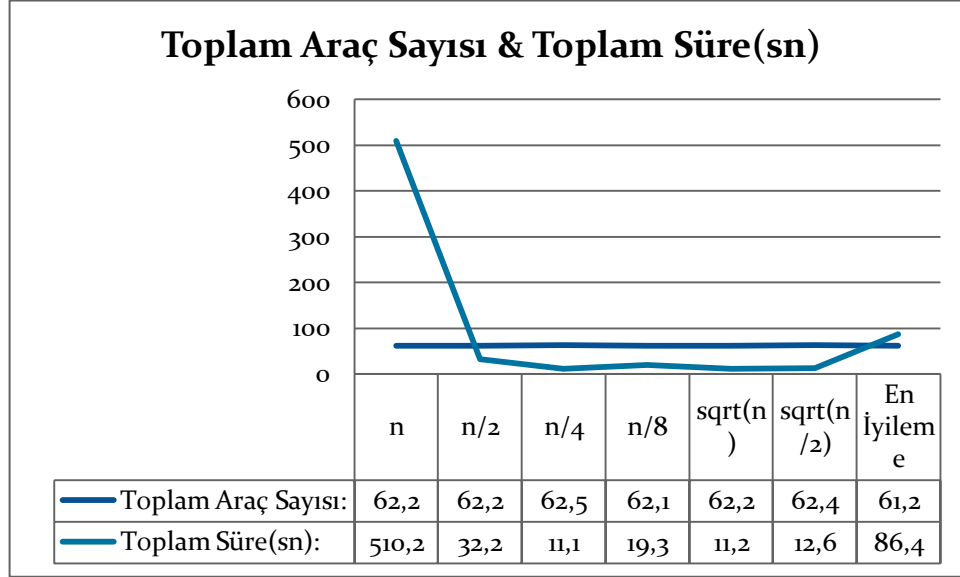


Şekil 5.23 Araç Sayısı ve Sürenin Kümelemeye Göre Değişimi
(100 iterasyon,15 depo,500 müşteri)

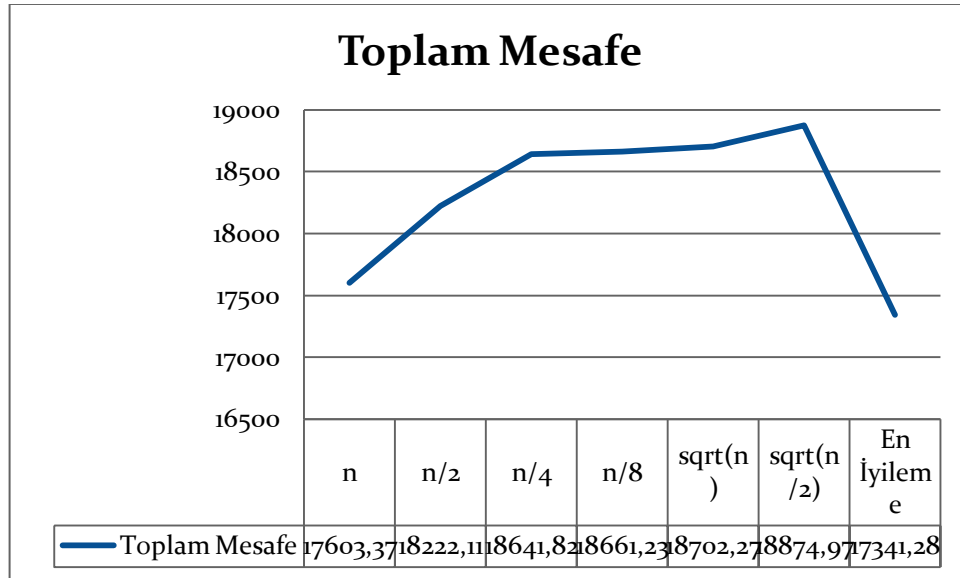


Şekil 5.24 Toplam Mesafenin Kümelemeye Göre Değişimi
(100 iterasyon,15 depo,500 müşteri)

İterasyon:100, Depo Sayısı:20 Müşteri Sayısı:500

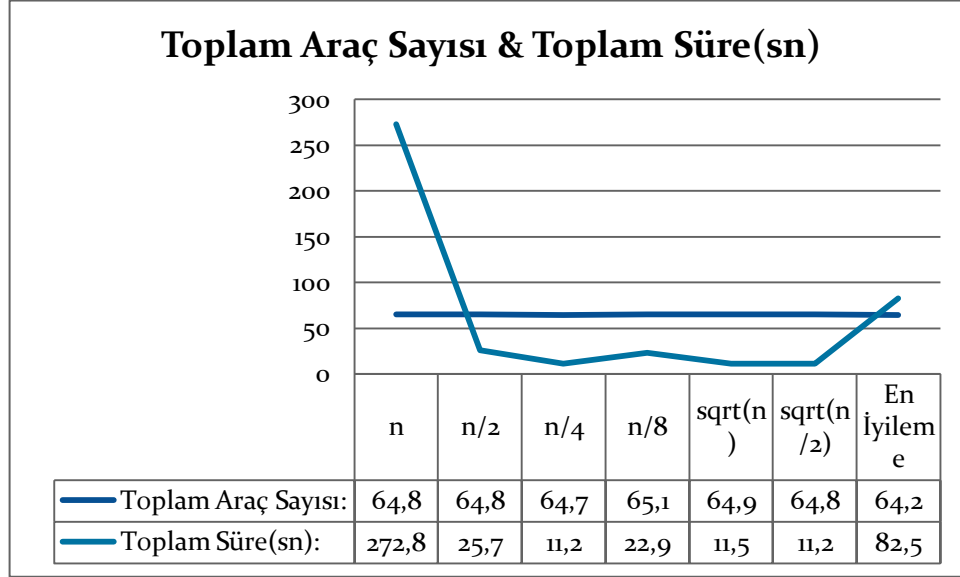


Şekil 5.25 Araç Sayısı ve Sürenin Kümelemeye Göre Değişimi
(100 iterasyon,20 depo,500 müşteri)

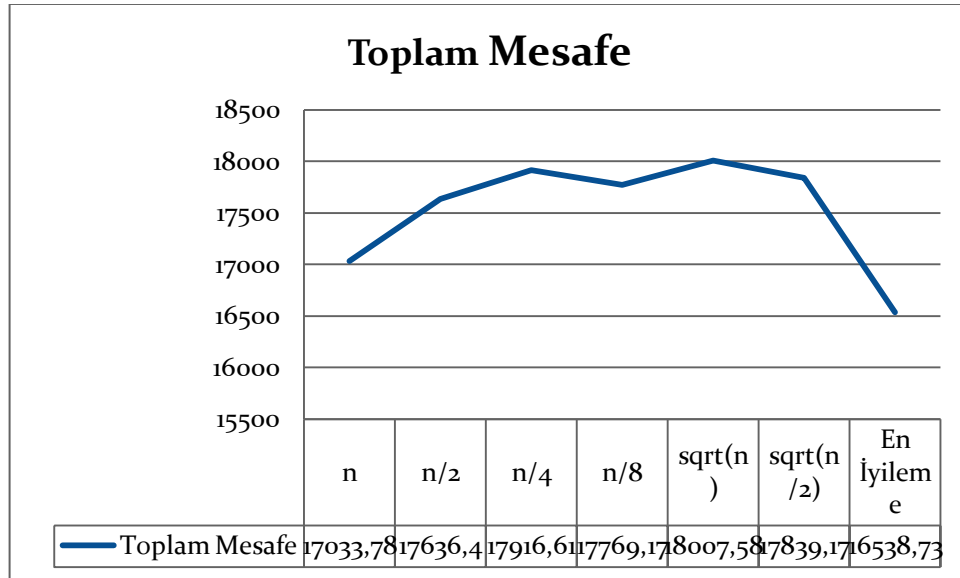


Şekil 5.26 Toplam Mesafenin Kümelemeye Göre Değişimi
(100 iterasyon,20 depo,500 müşteri)

İterasyon:100, Depo Sayısı:25 Müşteri Sayısı:500



Şekil 5.27 Araç Sayısı ve Sürenin Kümelemeye Göre Değişimi
(100 iterasyon,25 depo,500 müşteri)



Şekil 5.28 Toplam Mesafenin Kümelemeye Göre Değişimi
(100 iterasyon,25 depo,500 müşteri)

Müşteri sayısı ve iterasyon sayısı sabitken depo sayısı arttırılmış toplam araç sayısı, toplam rotalama süresi ve toplam mesafenin değişimleri gözlenmiştir. Elde edilen verilere göre Toplam rotalama süresi en iyi \sqrt{n} ve $n/4$ kümelemede elde edilmiştir. Yine elde edilen verilere bakıldığında kümeleme yapılarak elde edilen rotalama süresi kümeleme yapılmadan elde edilen rotalama süresinden çok daha iyi sonuçlar verdiği görülmüştür.

Toplam araç sayısı en iyi hibrit olarak kullanılan en iyileme işlemiyle bulunmuştur. En iyileme yöntemi kümeleme yapılmadan uygulanan karınca kolonisi optimizasyonunun bulduğu toplam araç sayısına göre daha iyi sonuçlar üretmiştir.

En iyileme dışında toplam araç sayısı $n/8$ kümelemede diğer kümeleme işlemlerinden daha iyi sonuçlar üretmiştir.

Toplam mesafe de en iyi yine hibrit olarak kullanılan en iyileme işlemi ile bulunmuştur. En iyileme yöntemi kümeleme yapılmadan uygulanan karınca kolonisi optimizasyonunun bulduğu toplam mesafeye göre daha iyi sonuçlar üretmiştir.

En iyilemenin dışında en iyi sonuç $n/2$ kümeleme işleminde elde edilmiştir.

BÖLÜM 6

6. SONUÇ

6.1. Yorumlar

Çok depolu araç rotalama problemlerinde müşteri taleplerinin en kısa sürede, en kısa yoldan karşılanması ve taleplerin karşılanması için kullanılacak olan araç sayısının en uygun sayıda seçilmesi oldukça önemlidir. NP zor bir problem olan araç rotalama problemlerinde müşteri sayısı arttıkça problemin çözümünde giderek zorlaşır.

Bu tez çalışmasında çok depolu araç rotalama problemlerinde müşteriler belirlenen sayıda (n , $n/2$, $n/4$, $n/8$, \sqrt{n} , $\sqrt{n/2}$) k ortalama kümeleme tekniği ile alt problemlere ayrılarak çözülmüş ve problem kısıtlarının (taleplerin karşılama süresi, kullanılacak araç sayısı, toplam mesafe) değişimleri incelenmiştir.

Yapılan çalışmada ilk olarak uygulama üzerinden depo sayısı 10, müşteri sayısı 500 ve iterasyon sayısı 20, 40, 60, 80, 100 seçilerek toplam araç sayısı, toplam süre ve toplam mesafedeki değişimler kaydedilmiştir. Her bir işlem 10 kez arka arkaya gerçekleştirilmiş ve elde edilen sonuçların ortalaması alınmıştır. Alınan sonuçlar incelendiğinde Şekil 5.1 ile Şekil 5.10 arasında görülen grafiklere göre toplam rotalama süresi en iyi \sqrt{n} kümelemede elde edilmiştir. Yani depoya atanmış müşterilerin karekökü kadar k ortalama kümeye ayırma işleminde rotalama süresinde optimum sonuç elde edilmiştir. Yine elde edilen verilere bakıldığında kümeleme yapılarak elde edilen rotalama süresi kümeleme yapılmadan elde edilen rotalama süresinden çok daha iyi sonuçlar verdiği görülmüştür.

Toplam araç sayısı en iyi hibrit olarak kullanılan en iyileme işlemiyle bulunmuştur. En iyileme işleminde her depo sırasıyla n , $n/2$, $n/4$, $n/8$, \sqrt{n} , $\sqrt{n/2}$ kümeye ayrılır ve o depo için en iyi sonucu üreten kümenin bulunduğu değer alınır. Yani çok depolu araç rotalama probleminde tüm depolar gezildiğinde en iyileme yönteminde bulunan en iyi sonuç farklı bir kümeleme yöntemi ile bulunmuş olur. En iyileme yöntemi kümeleme yapılmadan uygulanan karınca kolonisi optimizasyonunun bulunduğu toplam araç sayısına göre daha iyi sonuçlar üretmiştir.

En iyileme dışında toplam araç sayısı $n/2$ ve \sqrt{n} kümelemede diğer kümeleme işlemlerinden daha iyi sonuçlar üretmiştir.

Toplam mesafe de en iyi yine hibrit olarak kullanılan en iyileme işlemi ile bulunmuştur. En iyileme yöntemi kümeleme yapılmadan uygulanan karınca kolonisi optimizasyonunun bulunduğu toplam mesafeye göre daha iyi sonuçlar üretmiştir.

En iyilemenin dışında en iyi sonuç $n/2$ kümeleme işleminde elde edilmiştir.

Diğer bir işlem depo sayısı 10, iterasyon sayısı 100 ve müşteri sayısı 200, 300, 400, 500, 600 seçilerek toplam araç sayısı, toplam süre ve toplam mesafedeki değişimler kaydedilmiştir. Her bir işlem 10 kez arka arkaya gerçekleştirilmiş ve elde edilen sonuçların ortalaması alınmıştır. Alınan sonuçlar incelendiğinde Şekil 5.11 ile Şekil 5.20 arasında görülen grafiklere göre toplam rotalama süresi en iyi \sqrt{n} kümelemede elde edilmiştir. Yani depoya atanmış müşterilerin karekökü kadar k ortalama kümeleme ayırma işleminde rotalama süresinde optimum sonuç elde edilmiştir. Yine elde edilen verilere bakıldığında kümeleme yapılarak elde edilen rotalama süresi kümeleme yapılmadan elde edilen rotalama süresinden çok daha iyi sonuçlar verdiği görülmüştür.

Toplam araç sayısı en iyi hibrit olarak kullanılan en iyileme işlemiyle bulunmuştur. En iyileme yöntemi kümeleme yapılmadan uygulanan karınca kolonisi optimizasyonunun bulunduğu toplam araç sayısına göre daha iyi sonuçlar üretmiştir.

En iyileme dışında toplam araç sayısı $n/2$ ve $n/8$ kümelemede diğer kümeleme işlemlerinden daha iyi sonuçlar üretmiştir.

Toplam mesafe de en iyi yine hibrit olarak kullanılan en iyileme işlemi ile bulunmuştur. En iyileme yöntemi kümeleme yapılmadan uygulanan karınca kolonisi optimizasyonunun bulunduğu toplam mesafeye göre daha iyi sonuçlar üretmiştir.

En iyilemenin dışında en iyi sonuç $n/2$ kümeleme işleminde elde edilmiştir.

Diğer bir işlem de iterasyon sayısı 100, müşteri sayısı 500 ve depo sayısı 10, 15, 20, 25 seçilerek toplam araç sayısı, toplam süre ve toplam mesafedeki değişimler kaydedilmiştir. Her bir işlem 10 kez arka arkaya gerçekleştirilmiş ve elde edilen sonuçların ortalaması alınmıştır. Alınan sonuçlar incelendiğinde Şekil 5.21 ile Şekil 5.28 arasında görülen grafiklere göre toplam rotalama süresi en iyi \sqrt{n} ve $n/4$ kümelemede elde edilmiştir. Yine elde edilen verilere bakıldığında kümeleme yapılarak elde edilen rotalama süresi kümeleme yapılmadan elde edilen rotalama süresinden çok daha iyi sonuçlar verdiği görülmüştür.

Toplam araç sayısı en iyi hibrit olarak kullanılan en iyileme işlemiyle bulunmuştur. En iyileme yöntemi kümeleme yapılmadan uygulanan karınca kolonisi optimizasyonunun bulduğu toplam araç sayısına göre daha iyi sonuçlar üretmiştir.

En iyileme dışında toplam araç sayısı $n/8$ kümelemede diğer kümeleme işlemlerinden daha iyi sonuçlar üretmiştir.

Toplam mesafe de en iyi yine hibrit olarak kullanılan en iyileme işlemi ile bulunmuştur. En iyileme yöntemi kümeleme yapılmadan uygulanan karınca kolonisi optimizasyonunun bulduğu toplam mesafeye göre daha iyi sonuçlar üretmiştir.

En iyilemenin dışında en iyi sonuç $n/2$ kümeleme işleminde elde edilmiştir.

6.2. Gelecek Çalışmalar

Uygulamada statik talep yapısı kullanılmıştır yani düğümlerdeki talep miktarları daha öncesinden bilinmektedir. Gelecek çalışmalarda dinamik talep yapısı kullanılarak işlemin gerçekleştirilmesi düşünülmektedir. Dinamik talep yapısında bazı düğümlerin talepleri önceden bilinmekte bazılarının ise talepleri rotalama süresince değişebilmektedir.

Diğer bir çalışmada heterojen bir filonun kullanılmasıdır. Bu durum hangi araç tipinin, hangi rotaya hizmet vereceğinin belirlenmesini, yani ilave bir kararı

gerektirir Tasarlanan sistemde homojen yani kapasite miktarları aynı olan araç filoları kullanılmıştır.

Gerçekleştirilen sistemde rota süresi sınırsız araç rotalama problemi gerçekleştirilmiştir. Yani depodan ayrılan bir aracın tekrar depoya dönmesi için gerekli bir süre kısıdı yoktur. Gelecekteki çalışmalarda rotalama süresinin de probleme dahil edilmesi düşünülmektedir.

Gerçekleştirilen sistemde oluşacak olan küme sayısı önceden belirlenerek işlem gerçekleştirilmiştir. Gelecekteki çalışmalarda ise küme sayısı dinamik olarak sistem parametreleriyle bulunması amaçlanmaktadır.

KAYNAKLAR

- [1] **LAPORTE, G., NOBERT, Y. and TAILLEFER, S.**, “Solving a Family of Multi-Depot Vehicle Routing and Location Routing Problems”, Chaier du Gerad G-87-10, Ecole des Haltes Etudes Commercialse de Montreal, 1987.
- [2] **SAVELSBERGH, M. W. P. and SOL, M.**, “The General Pick-up and Delivery Problem”, Transportation Science 29, 17-29, 1995.
- [3] **M.A., Tokaylı**, Zaman Pencereleli Araç Rotalama Problemi İçin Bir Karar Destek Sistemi, Yüksek Lisans Tezi, Endüstri Mühendisliği, Gazi Üniversitesi Fen Bilimleri Enstitüsü, Ankara, (2005).
- [4] **FISHER, M. L. and JAIKUMAR, R.**, “A Generalised Assignment Heuristic for Vehicle Routing”, Networks 11, 109-124, 1981.
- [5] **Tan, K.C.**, A Framework of Supply Chain Management Literature ,European Journal of Purchasing Supply Management, 2000
- [6] **Toth,P., Vigo,D.**, The Vehicle Routing Problem, Siam, Bologna., 2002
- [7] **Eryavuz, M.**, Araç Rotalama Problemi ve Örnek Bir Uygulama, Yüksek Lisans Tezi, Gazi Üniversitesi, Fen Bilimleri Enstitüsü Endüstri Mühendisliği Ana Bilim Dalı, Ankara,2001
- [8] **Erel, R.**, Taşıt Rotalaması ve Çizelgelemesi:Otobüsle Kentlerarası Yolcu Taşımacılığı İçin Bir Model, Doktora Tezi ,Yıldız Teknik Üniversitesi, İnşaat Mühendisliği Bölümü, İstanbul, 1995
- [9] **Desrosiers, J.,Dumas, Y., Solomon, M.M., Soumis, F. ,** Time Constrained Routing and Scheduling, Forthcoming in Handbooks in Operations Research and Management Science, Volume on Networks, North-Holland, Amsterdam, 1993

- [10] **Bayzan, Ş.,Çetin, M., Uğur, A.,** Araç Rotalama Probleminde Araç Rotalarının Tespitinde En Kısa Yol Yaklaşımı: Denizli Örneği, Bilgisayar Mühendisliği Bölümü, Mühendislik Fakültesi Pamukkale Üniversitesi, Kınıklı, Denizli, 2002
- [11] **Ballou, R.H.,** Business Logistics Management:Planning, Organizing and Controlling The Supply Chain, McGraw-Hill, 1999
- [12] **Cordeau, J. F.,** Gendreau, M., Laporte, G., Potvin, J.Y., Semet, F., “A Guide To Vehicle Routing Heuristics”, Journal of the Operational Research Society, 53, 512-522, 2002
- [13] **Laporte, G.,** Gendreau, M., Potvin, J. Y., Semet, F., “Classical And Modern Heuristics For The Vehicle Routing Problem”, International Transactions in Operation Research, 7, 285-300, 2000
- [14] **Özalp, V.M.,** Gezgin Satıcı Problemi, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi, Fen BilimleriEnstitüsü, Endüstri Mühendisliği Ana Bilim Dalı, İstanbul, 1995
- [15] **Demircioğlu M.,** “Araç Rotalama Probleminin sezgisel bir yaklaşım ile çözümlenmesi üzerine bir uygulama”,Doktora Tezi,İşletme Anabilim Dalı, Çukurova Üniversitesi,Adana
- [16] **Laporte G.** “The Vehicle Routing Problem: An overview of exact and approximate algorithms”, European Journal of Operational Research, c. 59, sf. 345-358, 1992
- [17] **Khan, M.S. ve Siddiqui, A.S.,** The Radius Method: A Modified Heuristic For The Vehicle Routing Problem, Department of Industrial Engineering and Operations Research University of California Berkeley, 1998
- [18] **Papadimitriou, C. H., & Steiglitz, K.** Combinatorial optimization – algorithms and complexity. New York: Dover, 1982

- [19] **Colorni**, A., Dorigo, M., Maniezzo, V., & Trubian, M.. Ant System for Job-Shop Scheduling. *JORBEL – Belgian Journal of Operations Research, Statistics and Computer Science*, 34(1), 39–53, 1994
- [20] **Bonabeau**, B., Dorigo, M., & Theraulaz, G. Swarm intelligence: From natural to artificial systems. New York, NY: Oxford University Press, 1999
- [21] **Grasse**, P. P. La reconstruction du nid et les coordinations interindividuelles chez *bellicositermes natalensis* et *cubitermes* sp. la theorie de la stigmergie: Essai d'interpretation du comportement des termites constructeurs. *Insectes Sociaux*, 6, 41–81, 1959
- [22] **Dorigo**, M., & Di Caro, G. The Ant Colony optimization metaheuristic. *New Ideas in Optimization* (pp. 11–32), 1999
- [23] **Dorigo**, M., Maniezzo, V., & Colorni, A.. Ant System: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics – Part B*, 26, 29–41, 1996
- [24] **Kirkpatrick**, S., Gelatt, C. D., Jr, & Vecchi, M. P. Optimization by simulated annealing. *Science*, 220, 671–680, 1983
- [25] **Glover**, F. Tabu search – Part I. *ORSA Journal on Computing*, 1(3), 190–206, 1989
- [26] **Lourenco**, H. R., Martin, O., & Stutzle, T. Iterated local search. *Handbook of Metaheuristics*, 57, 321–353, 2002
- [27] **Goss**, S., Aron, S., Deneubourg, J. L., & Pasteels, J. M. Self-organized shortcuts in the argentine ant. *Naturwissenschaften*, 76, 579–581, 1989
- [28] **Cordon**, O., Herrera, F., & Stutzle, T. A review on the ant colony optimization metaheuristic: Basis, models and new trends. *Mathware and Soft Computing*, 9(2–3), :141–175, 2002

- [29] **Dorigo**, M., & Gambardella, L. M. Ant Colony System: A cooperating learning approach to the travelling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1(1), 53–66, 1997
- [30] **Stutzle**, T., & Hoos, H. H. Max–min ant system. *Future Generation Computer Systems*, 16(8), 889–914, 2000
- [31] **Dorigo**, M., & Gambardella, L. M. A study of some properties of Ant-Q. In *PPSN IV: Proceedings of the 4th international conference on parallel problem solving from nature* (pp. 656–665). Springer-Verlag, 1996
- [32] **Bullheimer**, B., Hartl, R. F., & Strauss, C. A new rank-based version of the Ant System: A computational study. *Central European Journal for Operations Research and Economics*, 7(1), 25–38, 1996
- [33] **Schoonderwoerd**, R., Holland, O., Bruten, J., & Rothkrantz, L. Ant-based load balancing in telecommunication networks. *Adaptive Behavior*, 5(2), 169–207, 1996
- [34] **Bullheimer**, B., Hartl, R. F., & Strauss, C. An improved Ant System algorithm for the vehicle routing problem. *Annals of Operations Research*, 89, 319–328, 1999
- [35] **Costa**, D., & Hertz, A. Ants can colour graphs. *Journal of the Operational Research Society*, 48, 295–305, 1997
- [36] **Lessing**, L., Dumitrescu, I., & Stutzle, T. A comparison between ACO algorithms for the set covering problem. *ANTS'2004, fourth international workshop on ant algorithms and swarm intelligence* (pp. 1–12). Springer-Verlag, 2004
- [37] **Gambardella**, L. M., Rizzoli, A. E., Oliverio, F., Casagrande, N., Donati, A. V., Montemanni, R., et al. Ant colony optimization for vehicle routing in advanced logistic systems. In *MAS 2003 – international workshop on modelling and applied simulation*, Bergeggi, Italy (pp. 3–9), 2003

- [38] **Gambardella**, L. M., Taillard, E. D., & Agazzi, G. MACS-VRPTW: A multiple ant colony system for vehicle routing problems with time windows. *New Ideas in Optimization* (pp. 63–76), 1999
- [39] **Socha**, K. ACO for continuous and mixed-variable optimization. In *Ant colony optimization and swarm intelligence, 4th international workshop, ANTS 2004* (pp. 25–36). Springer-Verlag, 2004
- [40] **Manfrin**, M., Birattari, M., Stutzle, T., & Dorigo, M. Parallel ant colony optimization for the travelling salesman problem. In *Proceedings of ANTS 2006* (pp. 224–234). Springer-Verlag, 2006
- [41] **Talbi**, E. G., Roux, O., Fonlupt, C., & Robillard, D. Parallel ant colonies for combinatorial optimization problems. In *Parallel and distributed processing, 11 IPPS/SPDP'99 workshops* (vol. 1586, pp. 239–247), 1999
- [42] **Pang-Ning Tan**, Michael Steinbach, Vipin Kumar, *Introduction to Data Mining*, Pearson Education (Addison Wesley), Page 488-497, 2006

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : ÇALIŞKAN, Kamil
Uyruğu : T.C.
Doğum tarihi ve yeri : 18.06.1985, AYDIN
Medeni hali : Bekâr
Telefon : 0 (256) 226 83 06
Faks :
e-mail : kcaliskan@etu.edu.tr

Eğitim

Derece	Eğitim Birimi	Mezuniyet tarihi
Lisans	TOBB ETÜ/Bilgisayar Müh.	2009

İş Deneyimi

Yıl	Yer	Görev
2009-2011	TOBB Yazılım Müdürlüğü	Mühendis

Yabancı Dil

İngilizce