



**ÇOKLU GEZGİN SATICI PROBLEMİNİN
ÇÖZÜMÜNDE KÜMELEME ALGORİTMASI İLE
ROTA OPTİMİZASYONU**

Burak AKSU

**Yüksek Lisans Tezi
Matematik Ana Bilim Dalı
Prof. Dr. Murat SUBAŞI
2025
(Her hakkı saklıdır.)**

T.C.
ATATÜRK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
MATEMATİK ANA BİLİM DALI

**ÇOKLU GEZGİN SATICI PROBLEMİNİN ÇÖZÜMÜNDE KÜMELEME
ALGORİTMASI İLE ROTA OPTİMİZASYONU**

(Route Optimization with Clustering Algorithm in Solving the Multiple Travelling
Salesman Problem)

YÜKSEK LİSANS TEZİ

Burak AKSU

Danışman: Prof. Dr. Murat SUBAŞI

Erzurum
Temmuz, 2025

TEZ KABUL VE ONAY TUTANAĞI

ÇOKLU GEZGİN SATICI PROBLEMİNİN ÇÖZÜMÜNDE KÜMELEME ALGORİTMASI İLE ROTA OPTİMİZASYONU

Prof. Dr. Murat SUBAŞI danışmanlığında, Burak AKSU tarafından hazırlanan bu çalışma, 25/07/2025 tarihinde aşağıdaki jüri tarafından Matematik Anabilim Dalı, Uygulamalı Matematik Bilim Dalı'nda yüksek lisans tezi olarak ile edilmiştir.

Jüri Başkanı: Doç. Dr. Muhammed YİĞİDER Aslı ıslak imzalıdır
Erzurum Teknik Üniversitesi

Danışman: Prof. Dr. Murat SUBAŞI Aslı ıslak imzalıdır
Atatürk Üniversitesi

Jüri Üyesi: Doç. Dr. Sıdıka Şule ŞENER KILIÇ Aslı ıslak imzalıdır
Atatürk Üniversitesi

Bu tezin Atatürk Üniversitesi Lisansüstü Eğitim ve Öğretim Yönetmeliği'nin ilgili maddelerinde belirtilen şartları yerine getirdiğini onaylarım.

Prof. Dr. Alper NUHOĞLU

Enstitü Müdürü

Aslı ıslak imzalıdır

ETİK BİLDİRİM VE İNTİHAL BEYAN FORMU

Yüksek Lisans tezi olarak Prof. Dr. Murat SUBAŞI danışmanlığında sunulan ‘‘ÇOKLU GEZGİN SATICI PROBLEMİNİN ÇÖZÜMÜNDE KÜMELEME ALGORİTMASI İLE ROTA OPTİMİZASYONU’’ başlıklı çalışmanın tarafımızdan bilimsel etik ilkelere uyularak yazıldığını, yararlanılan eserlerin kaynakçada gösterildiğini, Fen Bilimleri Enstitüsü tarafından belirlenmiş olan Turnitin Programı benzerlik oranlarının aşılmadığını ve aşağıdaki oranlarda olduğunu beyan ederiz.

Tez Bölümleri	Tezin Benzerlik Oranı (%)	Maksimum Oran (%)
Giriş	10	30
Kuramsal Temeller	3	30
Materyal ve Metot	14	35
Araştırma Bulguları	1	20
Tartışma ve Sonuç	0	20
Tezin Geneli	6	25

Not: Yedi kelimeye kadar benzerlikler ile Başlık, Kaynakça, İçindekiler, Teşekkür, Dizin ve Ekler kısımları tarama dışı bırakılabilir. Yukarıdaki azami benzerlik oranları yanında tek bir kaynaktan olan benzerlik oranlarının %5'den büyük olmaması gerekir.

Beyan edilen bilgilerin doğru olduğunu, aksi halde doğacak hukuk sorumlulukları kabul ve beyan ederiz.

Tez Yazarı (Öğrenci)	Tez Danışmanı
Burak AKSU	Prof. Dr. Murat SUBAŞI
25.7.2024	25.7.2024
İmza: Aslı ıslak imzalıdır	İmza: Aslı ıslak imzalıdır

* Tez ile ilgili YÖKTEZ’de yayımlanmasına ilişkin bir engelleme var ise aşağıdaki alanı doldurunuz.

Tezle ilgili patent başvurusu yapılması / patent alma sürecinin devam etmesi sebebiyle Enstitü Yönetim Kurulunun/.../.... tarih ve sayılı kararı ile teze erişim 2 (iki) yıl süreyle engellenmiştir.

Enstitü Yönetim Kurulunun/.../.... tarih ve sayılı kararı ile teze erişim 6 (altı) ay süreyle engellenmiştir.

TEŐEKKÜR

Yüksek Lisans tezi olarak sunduđum bu alıřmada, alıřma konusunun tayin edilmesinde ve alıřma sürecinin her ařamasında bilgi, tecrübe ve kıymetli zamanını esirgemeyerek sevgi, sabır ve destekle bana rehberlik eden Atatürk Üniversitesi Matematik Bölümü öğretim üyelerinden deđerli danışman hocam Prof. Dr. Murat SUBAŐI' ya en içten teşekkürlerimi sunmayı bir bor bilirim.

Bunun yanı sıra tez savunma jürisinde yer alarak deđerli görüşleri ile alıřmalarımıza destek olan saygıdeđer hocalarıma da ok teşekkür ediyorum.

Eđitim hayatımın her anında yanımda olan, desteđini ve sevgisini her daim hissettiren canım aileme; sevgili babama, anneme, ablama, kardeřime sonsuz teşekkür ederim. Yüksek lisansım sürecince bana destek olan deđerli arkadaşlarıma da sonsuz teşekkür ederim.

Ayrıca, TÜBİTAK Bilim İnsanı Destek Programları Başkanlığı (BİDEB) tarafından yürütölen 2210-A programı kapsamında verdikleri maddi destekler için TÜBİTAK nezdinde Devletime teşekkürlerimi sunarım.

Burak AKSU

ÖZET

YÜKSEK LİSANS TEZİ ÇOKLU GEZGİN SATICI PROBLEMİNİN ÇÖZÜMÜNDE KÜMELEME ALGORİTMASI İLE ROTA OPTİMİZASYONU

Burak AKSU

Danışman: Prof. Dr. Murat SUBAŞI

Amaç: Bu çalışmanın temel amacı, lojistik ve dağıtım ağlarında verimli rota optimizasyonu sağlamak amacıyla modellenen ve çoklu gezgin satıcı problemi olarak bilinen problemde farklı hibrit sezgisel algoritmaların performansını karşılaştırmaktır. Özellikle, farklı satıcı sayıları ve küme boyutları altında hangi hibrit yöntemin daha başarılı sonuçlar verdiğini analiz etmek ve bu algoritmaların güçlü ve zayıf yönlerini ortaya koymaktır.

Yöntem: Çalışmada, Parçacık Sürü Optimizasyonu (PSO), Karınca Kolonisi Optimizasyonu (KKO), Genetik Algoritma (GA) ve Yakın Komşu Algoritması (YKA) gibi sezgisel yöntemlerin kombinasyonlarından oluşan üç farklı hibrit algoritma (PSO+KKO, PSO+GA, GA+YKA) kullanılmıştır. PSO ve KKO metotları detaylı bir şekilde incelenmiş, diğer iki metot ise (GA ve YKA) detaylarına girilmeden karşılaştırma amacıyla dâhil edilmiştir. Algoritmaların performansı, literatürde yaygın olarak kullanılan veri setleri olan eil51, berlin52, eil76 ve rat99 test problemleri üzerinde değerlendirilmiştir. Kümeleme ve rotalama aşamaları birleştirilerek, her bir hibrit yöntemin etkinliği ölçülmüştür.

Bulgular: PSO+KKO algoritması, incelenen satıcı sayısı aralığında, kümeleme ve rotalama işlemlerinde diğer yöntemlere göre daha üstün performans sergilemiştir. Hibrit algoritmaların performansı, problem boyutuna ve satıcı sayısına bağlı olarak değişiklik göstermiştir.

Sonuç: Bu çalışma, farklı senaryolarda hangi hibrit algoritmanın daha etkili olduğunu ortaya koyarak, lojistik ve rota optimizasyon problemlerinde karar vericilere yol gösterici bilgiler sunmaktadır. İncelenen satıcı sayısı aralığında PSO+KKO'nun tercih edilmesi önerilmektedir. Gelecekteki çalışmalarda, farklı satıcı sayılarıyla farklı hibrit kombinasyonların denenmesi önerilebilir.

Anahtar Kelimeler: Çoklu Gezgin Satıcı Problemi, Meta-Sezgisel Optimizasyon, Parçacık Sürü Optimizasyonu, Karınca Kolonisi Optimizasyonu.

Temmuz 2025, 47 Sayfa

ABSTARCT

MASTER THESIS

ROUTE OPTIMIZATION WITH CLUSTERING ALGORITHM IN SOLVING THE MULTIPLE TRAVELLING SALESMAN PROBLEM

Burak AKSU

Danışman: Prof. Dr. Murat SUBAŞI

Purpose: The main objective of this study is to compare the performance of different hybrid heuristic algorithms on the problem known as the multiple traveling salesman problem, which is modeled to provide efficient route optimization in logistics and distribution networks. In particular, to analyze which hybrid method gives more successful results under different salesman numbers and cluster sizes and to reveal the strengths and weaknesses of these algorithms.

Method: In the study, three different hybrid algorithms (PSO+ACO, PSO+GA, GA+ NNA) consisting of combinations of heuristic methods such as Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Genetic Algorithm (GA) and Nearest Neighbor Algorithm (NNA) were used. PSO and ACO methods were examined in detail, while the other two methods (GA and NNA) were included for comparison purposes without going into detail. The performance of the algorithms was evaluated on the test problems eil51, berlin52, eil76 and rat99, which are widely used data sets in the literature. The effectiveness of each hybrid method was measured by combining the clustering and routing stages.

Findings: PSO+ACO algorithm has shown superior performance compared to other methods in clustering and routing operations in the examined salesman number range. The performance of hybrid algorithms has varied depending on the problem size and the number of salesmen.

Results: This study provides guidance to decision makers in logistics and route optimization problems by revealing which hybrid algorithm is more effective in different scenarios. It is recommended that PSO+ACO be preferred in the examined salesman number range. In future studies, it may be suggested to try different hybrid combinations with different salesman numbers.

Keywords: Multiple Traveling Salesman Problem (MTSP), Meta-Heuristic Optimization, Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO).

July 2025, 47 Pages

İÇİNDEKİLER

TEZ KABUL VE ONAY TUTANAĞI	i
ETİK BİLDİRİM VE İNTİHAL BEYAN FORMU	ii
TEŞEKKÜR	iii
ÖZET	iv
ABSTARCT	v
İÇİNDEKİLER.....	vi
ŞEKİLLER DİZİNİ	viii
TABLOLAR DİZİNİ.....	ix
SİMGELER VE KISALTMALAR DİZİNİ	x
GİRİŞ.....	11
Gezgin Satıcı Problemi (GSP)	11
Çoklu Gezgin Satıcı Problemi (ÇGSP).....	12
Matematiksel Formülasyon.....	12
Çoklu Gezgin Satıcı Probleminin Çözümü.....	12
KURAMSAL TEMELLER.....	15
Küme Ayrışması	15
Kümeleme Problemi.....	15
Çoklu Gezgin Satıcı Probleminde Kümeleme Algoritması	16
Çoklu GSP'de kümelenmenin tanımı	16
Kümeleme probleminin matematiksel modeli	16
Çoklu Gezgin Satıcı Probleminde Rotalama Algoritması	17
MATERYAL VE YÖNTEM	19
1. Kombinasyonel Patlama ve Hesaplama Karmaşıklığı	19
2. Yerel Optimum Tuzağından Kaçınma Yeteneği.....	19
3. Çok Modlu ve Dinamik Problemlerde Esneklik	19
4. Karmaşık Kısıtların Kolay Entegrasyonu	19
5. Hibrit Yapıların Sinerjik Avantajları	20
Sonuç.....	20
Parçacık Sürü Optimizasyonu (PSO).....	20
1. Eylemsizlik Faktörü (w):.....	21
2. Kendine Güven Katsayısı (c ₁):.....	21
3. Sürü Güven Katsayısı (c ₂):.....	22

PSO İin Literatür Taraması	22
Karıncı Kolonisi Optimizasyonu (KKO).....	23
KKO İin Literatür Taraması	25
ARAŐTIRMA BULGULARI	26
Paracık Sürü Optimizasyonu (PSO) ile oklu Gezgın Satıcı Problemi (GSP) iin Őehir Kümleme İŐlemi.....	26
PSO ile Kümleme İŐlemi iin Bir Simülasyon.....	28
Karıncı Kolonisi Optimizasyonu (KKO) ile oklu Gezgın Satıcı Problemi (GSP) iin Rota Bulma İŐlemi	30
Karıncı Kolonisi Optimizasyonu ile Optimal Rota Bulma Simülasyonu.....	34
Hibrit Yöntemlerin KarŐılaŐtırılmaları.....	36
SONU.....	41
KAYNAKA	42
ÖZGEMİŐ.....	45

ŞEKİLLER DİZİNİ

Şekil 1. Karıncaların yol seçimi	24
Şekil 2. eil51, 2 Satıcı.....	38
Şekil 3. eil51, 3 Satıcı.....	38
Şekil 4. eil51, 5 Satıcı.....	38
Şekil 5. eil51, 7 Satıcı.....	38
Şekil 6. berlin52, 2 satıcı.....	38
Şekil 7. berlin52, 3 satıcı.....	38
Şekil 8. berlin52, 5 satıcı.....	39
Şekil 9. berlin52, 7 satıcı.....	39
Şekil 10. eil76, 2 satıcı	39
Şekil 11. eil76, 3 satıcı	39
Şekil 12. eil76, 5 satıcı	39
Şekil 13. eil76, 7 satıcı	39
Şekil 14. rat99, 2 satıcı	40
Şekil 15. rat99, 3 satıcı	40
Şekil 16. rat99, 5 satıcı	40
Şekil 17. rat99, 7 satıcı	40

TABLolar DİZİNİ

Tablo 1. Hibrit Algoritmalar Karşılaştırma Tablosu	37
---	----



SİMGELER VE KISALTMALAR DİZİNİ

Simgeler

- c_1 : Kendine güven katsayısı, parçacığın kendi en iyi bulduğu konuma doğru olan hareketinin (bilişsel bileşenin) etkisini belirleyen katsayıdır.
- c_2 : Sürü güven katsayısı, parçacığın sürüdeki en iyi konuma (global best) yönelme eğilimini belirleyen parametredir.
- g_{best} : Tüm parçacıklar içindeki en iyi sonuca sahip parçacığın konumu.
- p_{best} : Parçacığın tüm iterasyonları içinde en iyi sonuç aldığı konum.
- r_1 ve r_2 : Çözüm çeşitliliği sağlamak amacıyla her iterasyonda $[0,1]$ aralığında rastgele program tarafından atanan değişkenler.
- w : Eylemsizlik faktörü, parçacığın mevcut hızının ne kadar etkili olacağını belirler.
- α : Feromonun etkisini kontrol eden katsayı.
- β : Sezgisel bilginin etkisini kontrol eden katsayı.
- η : Şehirler arasındaki mesafeye dayalı sezgisel bilgi, mesafeyle ters orantılı olarak belirlenir.
- τ : Karıncaların geçtiği yollardaki feromon yoğunluğu.

Kısaltmalar

- ÇGSP : Çoklu Gezgin Satıcı Problemi
- GA : Genetik Algoritma
- GSP : Gezgin Satıcı Problemi
- KKO : Karınca Koloni Optimizasyonu
- PSO : Parçacık Sürü Optimizasyonu
- YKA : Yakın Komşu Algoritması

GİRİŞ

Gezgin Satıcı Problemi (GSP)

Gezgin Satıcı Problemi (GSP), operasyonel araştırma ve bilgisayar bilimlerinde iyi bilinen bir optimizasyon problemidir. Belirli bir başlangıç noktasından hareket eden bir satıcının, bir dizi şehri yalnızca bir kez ziyaret ederek ve başlangıç noktasına geri dönerek en kısa mesafeyi katetmesi gereken bir rotayı bulmayı amaçlar (Gutin, et al. 2002; Lawler et al. 1985).

GSP'nin kökleri 1800'lere kadar uzanmaktadır, ancak matematiksel bir problem olarak ilk formülasyonu 20. yüzyılın başlarında Karl Menger tarafından yapılmıştır (Menger 1928). Problem, özellikle lojistik, ulaşım ve üretim planlama gibi alanlarda çok çeşitli uygulamalara sahiptir (Applegate et al 2006).

Matematiksel Formülasyon:

GSP, bir graf $G = (V, E)$ ile temsil edilebilir, burada V şehirlerin kümesini ve E şehirler arasındaki bağlantıları temsil eder. d_{ij} , i ve j şehirleri arasındaki mesafeyi gösterir. x_{ij} , satıcının i şehirden j şehrine gidip gitmediğini gösteren bir ikili değişkendir (1: gider, 0: gitmez). GSP'nin matematiksel formülasyonu aşağıdaki gibidir:

$$\min \sum_{i=1}^n \sum_{j=1}^n d_{ij} x_{ij}$$

Kısıtlar:

- $\sum_{i=1}^n x_{ij} = 1, \forall j \in V$ (Her şehre sadece bir kez girilir)
- $\sum_{j=1}^n x_{ij} = 1, \forall i \in V$ (Her şehirden sadece bir kez çıkılır)
- $\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1, \forall S \subset V, S \neq \emptyset$ (Alt tur oluşumunu engellemek için)
- $x_{ij} \in \{0, 1\}, \forall i, j \in V$

Bu formülasyon, toplam seyahat mesafesini en aza indirirken, her şehrin tam olarak bir kez ziyaret edilmesini ve alt turların (yani, tüm şehirleri içermeyen döngüler) oluşmasını engeller.

Çoklu Gezgin Satıcı Problemi (ÇGSP)

Çoklu Gezgin Satıcı Problemi (ÇGSP), GSP'nin bir genellemesidir ve birden fazla satıcının aynı anda tüm şehirleri ziyaret etmesini içerir. Her satıcı belirli bir şehirden başlar ve tüm şehirleri yalnızca bir kez ziyaret ettikten sonra aynı şehire geri döner. Amaç, tüm satıcıların kat ettiği toplam mesafeyi en aza indirmektir (Bektaş 2006).

ÇGSP, GSP'den daha karmaşık bir problemdir ve çözümü daha zordur. Bunun nedeni, satıcı sayısının artmasıyla çözüm uzayının boyutunun üstel olarak büyümesidir. ÇGSP, rota planlama, lojistik, dağıtım ve çizelgeleme gibi birçok alanda uygulama bulmaktadır (Toth and Vigo 2014; Golden et al 2008).

Matematiksel Formülasyon

m satıcı ve n şehir olduğunu varsayalım. ÇGSP'nin matematiksel formülasyonu aşağıdaki gibidir:

$$\min \sum_{k=1}^m \sum_{j=1}^n \sum_{i=1}^n d_{ij} x_{ijk}$$

Kısıtlar:

- $\sum_{k=1}^m \sum_{j=1}^n x_{ijk} = 1, \forall j \in V$ (Her şehir sadece bir satıcı tarafından ziyaret edilir)
- $\sum_{j=1}^n x_{ijk} = \sum_{j=1}^n x_{jik}, \forall i \in V, \forall k \in \{1, \dots, m\}$ (Her şehire bir satıcı girer ve çıkar)
- $\sum_{i \in S} \sum_{j \in S} x_{ijk} \leq |S| - 1, \forall S \subset V, S \neq \emptyset, \forall k \in \{1, \dots, m\}$ (Alt tur oluşumunu engellemek için)
- $x_{ijk} \in \{0, 1\}, \forall i, j \in V, \forall k \in \{1, \dots, m\}$

Bu formülasyon, her satıcının bir şehirden başlayıp aynı şehire dönmesini, her şehrin sadece bir satıcı tarafından ziyaret edilmesini ve alt turların oluşmasını engeller.

Çoklu Gezgin Satıcı Probleminin Çözümü

Çoklu Gezgin Satıcı Problemi (ÇGSP) çözümleri genellikle iki ana aşamadan oluşur:

1. Kümeleme:

Bu aşamada, şehirler (veya müşteriler) belirli bir kritere göre (örneğin; coğrafi yakınlık, talep miktarı) gezginler arasında gruplara ayrılır. Amaç, her gezginin sorumlu olduğu şehirlerin kümesini belirlemek ve böylece problemin daha küçük ve yönetilebilir alt problemlere bölünmesini sağlamaktır.

Literatürde Kullanılan Kümeleme Yöntemleri:

- **k-means kümeleme:** Bu, en yaygın kullanılan kümeleme algoritmalarından biridir. Şehirleri, belirli bir sayıda (k) kümeye ayırır, k sayısı kullanıcı tarafından belirlenir (MacQueen 1967).
- **Hiyerarşik kümeleme:** Bu yöntem, şehirler arasında bir hiyerarşi oluşturarak kümeler oluşturur (Ward 1963).
- **Fuzzy c-means kümeleme:** Bu yöntem, şehirlerin birden fazla kümeye ait olma olasılığını dikkate alarak daha esnek bir kümeleme sağlar (Bezdek 1981).
- **Genetik algoritmalar:** Genetik algoritmalar, kümeleme problemini çözmek için evrimsel bir yaklaşım kullanır .(Holland 1975).
- **Parçacık Sürü Optimizasyonu (PSO):** PSO, kümeleme problemini çözmek için sürü zekasını kullanır (Kennedy and Eberhart 1995).

2. Rota Optimizasyonu:

Bu aşamada, her gezgin için kendi şehir kümesi içindeki en kısa rotayı bulmak amaçlanır. Bu, klasik Gezgin Satıcı Problemine (GSP) benzer ve çeşitli yöntemlerle çözülebilir.

Literatürde Kullanılan Rota Optimizasyonu Yöntemleri şunlardır;

- **Kesin yöntemler:**
 - **Dal-sınır algoritması:** Bu yöntem, çözüm uzayını sistematik bir şekilde araştırarak en uygun çözümü bulmayı garanti eder. Ancak, büyük ölçekli problemler için hesaplama süresi çok uzun olabilir (Little et al 1963).
 - **Dinamik programlama:** Bu yöntem, problemi daha küçük alt problemlere bölerek ve bu alt problemlerin çözümlerini birleştirerek en uygun çözümü bulur (Bellman 1957).
- **Sezgisel yöntemler:** Bu yöntemler, en uygun çözümü garanti etmez, ancak makul bir sürede iyi çözümler üretebilir.
 - **En yakın komşu algoritması:** Bu algoritma her adımda gezgine en yakın ziyaret edilmemiş şehri seçer (Rosenkrantz et al 1977).
 - **Genetik algoritmalar:** Genetik algoritmalar, rota optimizasyonu problemini çözmek için evrimsel bir yaklaşım kullanır (Grefenstette 1985).
 - **Tabu arama:** Tabu arama, yerel optimumlara takılmamak için yasaklı bir arama alanı kullanır (Glover 1989).

- **Metasezgisel yöntemler:** Bu yöntemler, sezgisel yöntemlerden daha gelişmiş arama stratejileri kullanır.
 - **Karınca Kolonisi Optimizasyonu (KKO):** KKO, karıncaların yiyecek arama davranışlarından esinlenerek en kısa rotayı bulur (Dorigo et al 1996).
 - **Simüle edilmiş tavlama:** Bu yöntem, metalurjideki tavlama işleminden esinlenerek en kısa rotayı bulur (Kirkpatrick et al 1983).

Hibrit Yöntemler:

Son yıllarda, ÇGSP'nin çözümünde hibrit yöntemlerin kullanımı artmaktadır. Hibrit yöntemler, farklı algoritmaların güçlü yönlerini birleştirerek daha etkili çözümler üretmeyi amaçlar. Örneğin kümeleme için k-means algoritması ve rota optimizasyonu için KKO algoritması birleştirilebilir (Rego and Laguna 2004).

KURAMSAL TEMELLER

Küme Ayrışması

Küme ayrışması(küme parçalanışı) bir kümenin, elemanları arasındaki belirli kurallara göre, daha küçük alt kümelere ayrılması işlemidir. Bu işlemde, küme elemanları her bir alt kümeye yalnızca bir kez dahil olur ve bu alt kümelerin birleşimi kümenin tamamını verir.

Matematiksel olarak, bir A kümesi, $A_1, A_2, A_3, \dots, A_n$ şeklinde alt kümelere ayrıldığında, şu koşullar sağlanmalıdır:

1. Her alt küme, A'nın bir alt kümesi olmalıdır: Yani, her bir $A_i \subseteq A$ olmalıdır.
2. Alt kümelerin kesişimi boş küme olmalı $A_i \cap A_j = \emptyset, i \neq j$ ve $i, j \in \{1, 2, \dots, n\}$.
3. Alt kümelerin birleşimi A kümesini vermelidir $A_1 \cup A_2 \cup A_3 \cup \dots \cup A_n = A$ olmalıdır.

Küme ayrışması, kümeleme probleminin matematiksel ve teorik bir temelini oluşturur. Kümeleme, bu temeli kullanarak veriyi daha anlamlı gruplar oluşturacak şekilde organize eder.

Kümeleme Problemi

Kümeleme problemi belli nesne ve olguları türlerine göre sınıflandırmaktır. Bu tezde kullanılacak olan kümeleme probleminin temel yapı taşı ÇGSP'nin çözümüne odaklı olarak mesafeye dayalı bir seçimle yapılmaktadır. Amaç oluşacak her alt kümenin ortalama işleminden sonra rotalarının mesafeleri toplamını minimum yapmaktır

Kümeleme problemi, benzer nesnelere gruplama veya sınıflandırma amacıyla kullanılan bir veri analizi yöntemidir. Bu problem, genellikle aşağıdaki adımları içerir:

1. **Veri Toplama:** Analiz edilecek verilerin toplanması.
2. **Öznitelik Seçimi:** Gruplama için hangi özelliklerin (özniteliklerin) kullanılacağına karar verme.
3. **Mesafe Ölçümü:** Nesnelere arasındaki benzerlik veya farklılığı ölçmek için bir mesafe veya benzerlik ölçütü belirleme (örneğin, Öklidyen mesafe, Manhattan mesafesi).
4. **Kümeleme Algoritması:** Uygun bir kümeleme algoritması seçme (örneğin, K-means, Hiyerarşik kümeleme, DBSCAN, Sezgisel, Meta sezgisel).
5. **Kümeleme:** Algoritmanın uygulanarak verilerin kümelere ayrılması.

6. **Sonuçların Değerlendirilmesi:** Elde edilen kümelerin geçerliliğini ve anlamlılığını değerlendirme.

Kümeleme: lojistik, pazarlama, biyoinformatik, sosyal bilimler ve daha birçok alanda kullanılmaktadır. Amaç, verileri daha anlamlı hale getirmek ve içindeki yapıları ortaya çıkarmaktır.

Çoklu Gezgin Satıcı Probleminde Kümeleme Algoritması

ÇGSP, birden fazla satıcının belirli bir dizi noktayı ziyaret etmesi gereken bir optimizasyon problemidir. Kümelenme, bu problem bağlamında, satıcıların ziyaret etmesi gereken noktaları gruplara ayırarak her bir satıcının bu gruplardan birini ziyaret etmesini sağlamak anlamına gelir.

Çoklu GSP'de kümelenmenin tanımı

Nokta Kümeleri: Ziyaret edilmesi gereken noktalar, benzerliklerine veya coğrafi konumlarına göre gruplara ayrılır. Bu, her satıcının belirli bir küme içinde yer alan noktaları ziyaret etmesini sağlar.

Amaç: Her bir satıcının bir küme içerisindeki noktaları ziyaret ederken toplam mesafeyi minimize etmektir. Bu, toplam seyahat süresini veya maliyetini azaltmayı hedefler.

Kümeleme Yöntemleri: Çeşitli kümeleme algoritmaları (K-means, Hiyerarşik kümeleme, Meta sezgisel yöntemlerle kümeleme) kullanılarak gruplara ayrılabilir. Bu gruplar, satıcıların daha verimli bir şekilde çalışabilmesi için optimize edilir.

Çözüm Stratejileri: Her bir küme için en iyi rotayı bulmak amacıyla, klasik GSP algoritmaları (örneğin, Dal-sınır, Dinamik programlama, Meta sezgisel yöntemler) kullanılabilir.

Sonuç: Çoklu GSP'de kümelenme, satıcıların görevlerini daha verimli bir şekilde yerine getirebilmesi için kritik bir adımdır. Bu yaklaşım hem zaman hem de maliyet açısından önemli tasarruflar sağlayabilir.

Kümeleme probleminin matematiksel modeli

Kümeleme probleminin matematiksel modeli, verilerin belirli bir sayıda gruba (küme) ayrılması sürecini tanımlar. Bu model genellikle aşağıdaki temel bileşenleri içerir:

1. Veri Seti

Veri seti $X=\{x_1, x_2, \dots, x_n\}$ şeklinde tanımlanır. Burada n veri noktası, her biri d boyutlu bir uzayda temsil edilir.

2. Küme Sayısı

Kümeleme işlemi, verileri k sayıda kümeye ayırmayı amaçlar. k önceden belirlenmiş bir parametre olarak tanımlanır.

3. Küme Merkezleri

Her bir küme için bir merkez u_j ($j = 1, 2, \dots, k$) belirlenir. Bu merkezler, ilgili kümedeki veri noktalarının ortalama değerlerini temsil eder.

4. Uzaklık Ölçümü

Veri noktaları ile küme merkezleri arasındaki uzaklık genellikle Euclidean mesafesi kullanılarak ölçülür:

$$d(x_i, u_j) = \sqrt{\sum_{m=1}^d (x_{im} - u_{jm})^2}$$

5. Amaç Fonksiyonu

Amaç, her bir veri noktasını en yakın küme merkezine atamak ve toplam uzaklığı minimize etmektir. Bu, genellikle aşağıdaki şekilde ifade edilir:

$$\min_C \sum_{j=1}^k \sum_{x_i \in C_j} d(x_i, u_j)$$

Burada C kümelerin bir kümesini temsil eder.

6. Çözüm Algoritmaları

Kümeleme süreci genellikle iteratif bir algoritma ile gerçekleştirilir. K-means, Hiyerarşik Kümeleme, Spectral Kümeleme, Meta Sezgisel algoritmalarla kümeleme gibi çeşitli yöntemlerle çözülebilir.

7. Sonuç

Sonuç olarak, her bir veri noktası C_j kümesine atanır ve bu kümelerin analizi yapılır.

Çoklu Gezgin Satıcı Probleminde Rotalama Algoritması

Rotalama Probleminin Temel Teoremleri şunlardır:

Teorem 1 (Hamiltonian Teoremi)

Hamiltonian Teoremi, bir grafda Hamiltonian döngüsünün varlığı ile ilgili bir dizi koşulu tanımlar. Hamiltonian döngüsü, bir grafın tüm düğümlerini tam olarak bir kez ziyaret eden ve başlangıç düğümüne geri dönen bir döngüdür.

Hamiltonian Teoreminin genel hali, belirli koşullar altında bir grafın Hamiltonian döngüsüne sahip olup olmadığını belirlemek için kullanılır. Ancak, bu teorem için genel bir yeterli ve gerekli koşul yoktur. Fakat bazı özel durumlar ve koşullar aşağıda belirtilmiştir:

Teorem 2 (Dirac Teoremi) Eğer G grafı $n \geq 3$ düğümüne sahipse ve her düğümün derecesi $\deg(v) \geq \frac{n}{2}$ ise o zaman G Hamiltonian döngüsüne sahiptir. ($\deg(v)$, v düğümünün derecesini yani komşu olduğu düğüm sayısını gösterir.)

Teorem 3 (Ore Teoremi) Eğer G grafı $n \geq 3$ düğümüne sahipse ve herhangi iki düğüm u ve v için $\deg(u) + \deg(v) \geq n$ koşulunu sağlıyorsa, o zaman graf Hamiltonian döngüsüne sahiptir.

Hamiltonian Teoremi, grafiklerin analizinde ve çeşitli problem çözme senaryolarında kritik bir rol oynar. Örneğin: Gezgin Satıcı Problemi, robotik ve otonom araçlar, rotasyon planlaması ve görev dağılımı gibi konularda kullanılır.

Hamilton döngüsü, GSP'nin bir alt kümesi olarak düşünülebilir; GSP çözülürken Hamilton döngüsü kavramı ve özellikleri kullanılır. Bu nedenle, Hamilton döngüsü, GSP'nin anlaşılması ve çözülmesi açısından kritik bir öneme sahiptir.

Kısacası, Hamilton döngüsü, GSP'nin temelini oluşturur. GSP, Hamilton döngüsü bulma problemine ek olarak, bu döngünün maliyetini optimize etme amacını taşır.

MATERYAL VE YÖNTEM

Gezgin Satıcı Problemi (GSP) ve çoklu satıcılı varyasyonları gibi kombinyonel optimizasyon problemleri, NP-zor sınıfında yer alır. Bu tür problemlerde şehir sayısı arttıkça çözüm uzayı katlanarak büyür, dolayısıyla klasik yöntemlerle (kesin algoritmalarla) optimal çözümü bulmak pratikte mümkün olmayabilir. Sezgisel ve meta-sezgisel yöntemler ise bu tür karmaşık problemlerde yakınsama hızı, ölçeklenebilirlik ve esneklik açısından avantajlar sunar. İşte bu tercihin temel teorik gerekçeleri:

1. Kombinyonel Patlama ve Hesaplama Karmaşıklığı

Klasik yöntemler (dinamik programlama, tam sayılı lineer programlama, dal-sınır algoritmaları vb.) teorik olarak optimal çözümü bulabilir, ancak $O(n!)$ veya $O(2^n)$ gibi üstel karmaşıklığa sahiptir. Örneğin, 50 şehirli bir GSP’de $50! \approx 3 \times 10^{64}$ olası rota vardır. Bu da deterministik yöntemlerin gerçekçi sürelerde çözüm üretmesini imkansız kılar.

Sezgisel yöntemler (PSO, GA, YKA vb.) ise polinomial zamanlı ($O(n^2)$ veya $O(n^3)$) yaklaşımlarla kabul edilebilir çözümler üretebilir.

2. Yerel Optimum Tuzağından Kaçınma Yeteneği

Klasik gradient tabanlı yöntemler (gradient descent, Newton-Raphson vb.), dışbükey olmayan problemlerde yerel optimumlarda sıkışabilir.

Meta-sezgisel algoritmalar (GA’daki mutasyon, PSO’daki sosyal ve bilişsel bileşenler, YKA’daki rastgelelik) çoklu arama yönleri sayesinde global optimuma daha iyi yakınsar.

3. Çok Modlu ve Dinamik Problemlerde Esneklik

Gerçek hayat problemleri genellikle dinamiktir (trafik, talep değişimi vb.) ve klasik yöntemler her değişiklikte yeniden çözüm gerektirir.

Sezgisel yöntemler, adaptif mekanizmaları (GA’da popülasyon güncelleme, PSO’da hız ayarlama) sayesinde dinamik ortamlara daha hızlı uyum sağlar.

4. Karmaşık Kısıtların Kolay Entegrasyonu

Klasik yöntemler, kapasite kısıtları, zaman pencereleri veya çoklu amaç fonksiyonları gibi karmaşık kısıtlar ekledikçe çözülemeye hale gelebilir.

Sezgisel yöntemler, ceza fonksiyonları veya uygunluk değerlendirmeleri ile bu kısıtları doğal olarak modele dahil edebilir.

5. Hibrit Yapıların Sinerjik Avantajları

Tek bir algoritma, kümeleme ve rotalama gibi farklı alt problemlerde zayıf kalabilir.

Hibrit yöntemler (PSO+KKO, GA+YKA), kümeleme ve rotalamanın ayrı aşamalarında farklı algoritmaların güçlü yönlerini birleştirerek daha dengeli bir performans sunar.

Sonuç

Sezgisel yöntemler, özellikle yüksek boyutlu, çok kısıtlı ve dinamik problemlerde klasik yöntemlere kıyasla:

- Daha hızlı çözüm üretir,
- Global optimuma daha yakın sonuçlar verir,
- Karmaşık kısıtları daha esnek yönetir,
- Hibritleşmeye açık yapılarıyla ölçeklenebilir çözümler sunar.

Bu nedenle, bu çalışmada PSO, KKO gibi meta-sezgisel algoritmaların hibrit kombinasyonları tercih edilmiştir.

Parçacık Sürü Optimizasyonu (PSO)

Parçacık Sürü Optimizasyonu (PSO), Eberhart and Kennedy (1995) tarafından geliştirilmiş bir optimizasyon algoritmasıdır. Bu algoritma, kuş sürülerinin yiyecek bulma ve avcılardan kaçma gibi sosyal davranışlarından esinlenerek oluşturulmuştur. PSO, bir popülasyon (sürü) içindeki aday çözümleri (parçacıklar) iteratif olarak hareket ettirerek belirli bir kalite ölçütüne göre en iyi çözümü bulmayı amaçlar. Her parçacık, kendi en iyi konumunu (p_{best}) ve tüm sürünün en iyi konumunu (g_{best}) kullanarak hareket eder.

Her bir parçacığın konumu ve hızı

$$V_i^{(t+1)} = w \cdot V_i^{(t)} + c_1 \cdot r_1 \cdot (p_{best_i}^{(t)} - S_i^{(t)}) + c_2 \cdot r_2 \cdot (g_{best}^{(t)} - S_i^{(t)}) \quad (3.1)$$

$$S_i^{(t+1)} = S_i^{(t)} + V_i^{(t+1)} \quad (3.2)$$

eşitlikleri ile güncellenir. Burada w eylemsizlik faktörünü, c_1 ve c_2 ise parçacığın kendi ve sürünün en iyi konumlarına olan etkisini temsil eder.

Her parçacığın uygunluk değeri, belirli bir hedef fonksiyonu kullanılarak hesaplanır. Bu fonksiyon, parçacığın çözüm uzayındaki konumuna bağlı olarak belirlenir ve algoritmanın başarısını etkiler.

PSO, iki temel yaklaşım kullanır: küresel en iyi (g_{best}) ve yerel en iyi (p_{best}). Küresel en iyi yaklaşımda, tüm parçacıklar en iyi bulunan konumu paylaşırlarken, yerel en iyi yaklaşımda parçacıklar yalnızca komşuları ile bilgi alışverişinde bulunur. Bu, algoritmanın keşif ve sömürü dengesini etkiler.

PSO'nun performansı, algoritmanın parametrelerinin (örneğin, eylemsizlik faktörü ve katsayılar) doğru bir şekilde seçilmesine bağlıdır. Yanlış parametreler, algoritmanın yerel minimumda sıkışmasına veya yetersiz keşif yapmasına neden olabilir.

Parçacık Sürü Optimizasyonu (PSO) algoritmasının başarısı, büyük ölçüde kullanılan parametrelerin uygun bir şekilde seçilmesine bağlıdır. PSO'da temel parametreler şunlardır:

1. Eylemsizlik Faktörü (w):

Tanım: Parçacığın mevcut hızının ne kadar etkili olacağını belirler. Yüksek bir değer, daha fazla keşif yaparken, düşük bir değer daha fazla sömürü sağlar.

Eylemsizlik faktörü w, PSO algoritmasında her parçacığın hızının önceki hızına ne kadar bağlı kalacağını belirleyen parametredir.

Lineer Azaltmalı Eylemsizlik (Linearly Decreasing Inertia Weight):

$$w(t) = w_{max} - \left(\frac{w_{max} - w_{min}}{t_{max}} \right) \cdot t$$

Bu yaklaşımda w başlangıçta büyük tutulur ($w_{max} \approx 0,9$), sonra yavaş yavaş küçültülür ($w_{min} \approx 0,4$), böylece önce keşif (exploration) sonra sömürü (exploitation) davranışı sağlanır (Shi and Eberhart 2001; Clerc and Kennedy 2002).

Seçim Stratejisi: Genellikle 0.4 ile 0.9 arasında bir değer seçilir. Dinamik bir yaklaşımda, bu değer iterasyon boyunca azaltılabilir.

2. Kendine Güven Katsayısı (c_1):

Kendine güven katsayısı c_1 , parçacığın kendi en iyi bulduğu konuma doğru olan hareketinin (bilişsel bileşenin) etkisini belirleyen katsayıdır. Parçacığın bireysel hafızasını temsil eder ve keşif davranışını artırır (Eberhart and Kennedy 1995).

3. Sürü Güven Katsayısı (c_2):

Sürü güven katsayısı c_2 , parçacığın sürüdeki en iyi konuma (global best) yönelme eğilimini belirleyen parametredir. Bu katsayı, parçacığın sosyal öğrenme derecesini ve topluluğa olan güvenini temsil eder (Eberhart and Kennedy 1995).

Bu bağlamda önerilen tipik sabitler:

$$c_1 = 2.05$$

$$c_2 = 2.05$$

Bu değerlerle algoritma hem keşif hem sömürü bakımından dengeli çalışır.

Teorem (Kararlılık ve Yakınsama):

PSO algoritmasının kararlı biçimde yakınsamayı sağlaması için parametrelerin şu koşulu sağlaması gerekir:

$$\phi = c_1 + c_2 > 4 \text{ ve } w < 1$$

Sonuç olarak PSO algoritmasının performansını artırmak için parametrelerin dikkatli bir şekilde seçilmesi gerekir. Parametrelerin optimizasyonu, genellikle deneme-yanılma yöntemi veya otomatik optimizasyon teknikleri ile gerçekleştirilir (Clerc and Kennedy 2002).

PSO İçin Literatür Taraması

PSO'nun kümeleme işleminde kullanıldığı çalışmalardan bahsedelim.

(Çınaroğlu and Bulut 2018) çalışmasında; çok boyutlu verileri kümelendirme için çalışma kapsamında standart K-means ve PSO kümeleme algoritmaları için başlangıç küme merkezlerinin seçimine yönelik yeni yöntemler önerilmiştir. Ayrıca öbek (coreset) yaklaşımı PSO algoritmasına uyarlanmıştır. Geliştirilen yöntemlerin doğruluğu; literatürde sıkça kullanılan veri setleri üzerinde test edilmiş ve bu yaklaşımlar Colon Cancer mikrodizi veri seti üzerinde çalıştırılmıştır.

(Dolapçı et al 2024) çalışmasında; hiperspektral görüntülerin graf temelli boyuta indirgenerek sınıflandırılması ele alınmıştır. Hiperspektral görüntü verilerinden hem uzamsal hem de spektral öznitelik çıkarımı ile analiz işlemi için gerekli ayrıntılı bilgiler elde edilmektedir. Önerilen çalışma kapsamındaki parametre PSO ile optimize edilmiştir.

Metin içeriklerinden duyguların belirlenmesi işlemi, duygu analizi olarak adlandırılır. İnternet altyapısının dünya genelinde yaygınlaşması, bireylerin düşüncelerini çevrimiçi olarak

paylaşmalarını mümkün kılmıştır. Bu çevrimiçi ortamlarda biriken verilerden anlamlı bilgilerin elde edilmesi, birçok alanda büyük önem taşımaktadır. (Aydın et al, 2018) çalışmasında, kullanıcılar tarafından oluşturulan Twitter verileri kullanılarak duygu analizi gerçekleştirilmiştir. Metinlerdeki duygular; olumlu, olumsuz ve belirsiz olmak üzere üç sınıfa ayrılmıştır. Sınıflandırma öncesinde, veri kümeleri üzerinde metin madenciliğine yönelik ön işlemler uygulanmış ve ardından özellik çıkarımı yapılmıştır. Çalışmada, sınıflandırma süreci için optimizasyona dayalı yeni bir yöntem önerilmiş ve bu yöntemle elde edilen sınıflandırma başarımının literatürdeki mevcut çalışmalardan daha iyi olduğu deneysel olarak ortaya konmuştur.

Genlerdeki gizli bilgilerin temsil edilmesi, bu bilgilerin analizini kolaylaştırır da, gen sayısının fazlalığı ve veri setlerindeki yüksek gürültü seviyesi gen verilerinin yorumlanmasını güçleştirmektedir. Bu sorunu aşmak için genlerin belirlenmesini kolaylaştıracak kümeleme yöntemlerinden yararlanılmaktadır. (Çınaroğlu and Bulut, 2018) çalışmasında, çok boyutlu verilerin kümeleme işlemi için standart K-ortalamlar ve Parçacık Sürü Optimizasyonu (PSO) tabanlı algoritmalarda kullanılmak üzere yeni başlangıç küme merkezi seçme yöntemleri önerilmiştir. Ayrıca, PSO algoritmasına öbek (coreset) yaklaşımı entegre edilmiştir. Önerilen yöntemlerin doğruluğu, literatürde yaygın olarak kullanılan veri setleriyle test edilmiş ve Colon Cancer mikrodizi veri seti üzerinde uygulanarak analiz edilmiştir. Standart K-ortalamlar ve PSO tabanlı yöntemlerle geliştirilen yaklaşımlar karşılaştırılmış; başarıları ortalama iterasyon sayısı, Rand ve Silhouette indeksleri ile değerlendirilmiştir. Deneysel sonuçlar, geliştirilen yöntemlerin öznelik seçimi yapılmış ve normalize edilmiş veri setlerinde etkili olduğunu göstermiştir.

Karıncalar Kolonisi Optimizasyonu (KKO)

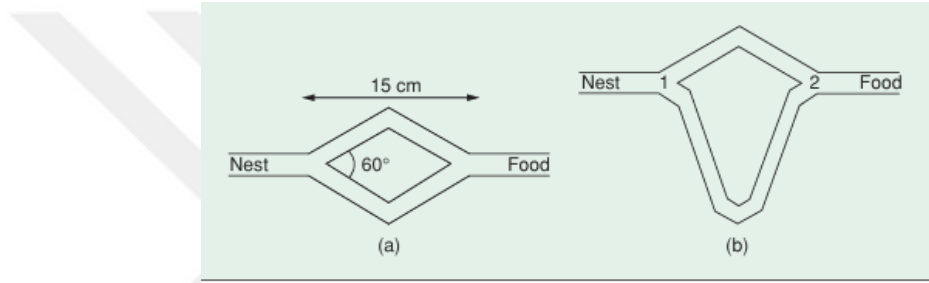
1940'lar ve 1950'lerde Fransız entomolog Pierre-Paul (1946), bazı termit türlerinin “önemli uyarılara” tepki verdiklerini gözlemlemiştir. Grassé, bu tepkilerin hem bu tepkileri üreten böcek hem de kolonideki diğer böcekler için yeni önemli uyarılar haline gelebileceğini belirtmiştir. Grassé, bu özel iletişim biçimini tanımlamak için “stigmerji” Pierre-Paul (1959) terimini kullanmıştır.

Stigmerjiyi diğer iletişim biçimlerinden ayıran iki temel özellik vardır:

- Stigmerji, çevre aracılığıyla dolaylı ve sembolik olmayan bir iletişim biçimidir: Böcekler, çevrelerini değiştirerek bilgi alışverişi yapar.
- Stigmerjik bilgi yereldir: Sadece, bilginin bırakıldığı yere (veya yakın çevresine) gelen böcekler tarafından erişilebilir.

Stigmerjiye karınca kolonilerinde sıkça rastlanır. Birçok karınca türü, bir yiyecek kaynağına giderken ve geri dönerken yere feromon adı verilen bir madde bırakır. Diğer karıncalar bu feromonu algılar ve feromon yoğunluğunun yüksek olduğu yolları takip etme eğilimindedir. Bu mekanizma sayesinde karıncalar, yiyeceği yuvalarına oldukça verimli bir şekilde taşırlar.

Deneubourg ve arkadaşları (1990), karıncaların feromon bırakma ve izleme davranışını ayrıntılı olarak incelemişlerdir. “Çift köprü deneyi” olarak bilinen bir deneyde, Arjantin karıncalarının yuvası, eşit uzunluktaki iki köprü aracılığıyla bir yiyecek kaynağına bağlanmıştır [Şekil 1(a)]. Karıncalar başlangıçta rastgele köprüleri seçerek çevreyi keşfeder. Yavaş yavaş, rastlantısal nedenlerle köprülerden biri daha fazla feromon biriktirir ve bu da daha fazla karıncayı o köprüye çeker. Böylece zamanla tüm koloni tek bir köprüye yönelir.



Şekil 1. Karıncaların yol seçimi

Bu koloni düzeyindeki davranış, yani pozitif geri bildirimden (autokataliz) faydalanma, karıncaların bir yiyecek kaynağı ile yuvaları arasındaki en kısa yolu bulmalarını sağlar.

Goss ve arkadaşları (1990), bu deneye bir varyasyon getirerek köprülerden birini diğerinden belirgin şekilde daha uzun yapmışlardır [Şekil 1(b)]. Bu durumda, kısa köprüyü seçen karıncalar daha önce yuvaya döner ve dolayısıyla kısa köprüye feromon daha önce bırakılır. Bu da diğer karıncaların kısa köprüyü tercih etme olasılığını artırır. Goss ve arkadaşları bu davranış için şu modeli geliştirmiştir:

$$p_1 = \frac{(m_1+k)^h}{(m_1+k)^h+(m_2+k)^h}$$

Burada

- m_1 , birinci köprüyü kullanan karınca sayısı
- m_2 , ikinci köprüyü kullanan karınca sayısı
- p_1 : Karıncanın 1. köprüyü seçme olasılığıdır.
- $p_2 = 1 - p_1$: Karıncanın 2. köprüyü seçme olasılığıdır.
- k : küçük sabit bir değer (başlangıç olasılıklarını sıfırdan kaçınmak için)

- h : sistemin duyarlılığını kontrol eden bir üstel parametre (feromon duyarlılığı gibi düşünülebilir)

k ve h , deneysel verilere göre ayarlanan parametrelerdir (Pasteels et al 1987).

Deneubourg ve çalışma arkadaşları tarafından karıncaların yiyecek arama davranışlarını açıklamak için önerilen model, karınca kolonisi optimizasyonunun geliştirilmesi için temel ilham kaynağı olmuştur.

KKO İçin Literatür Taraması

(Pala and Aksaraylı 2018) çalışmasında; ulaştırma sektöründe servis araçlarıyla uçuşlar arasındaki toplam tur sürelerini ve bir yolcunun ortalama ulaşımda kaldığı süreyi en aza indirme problemi ele alınmıştır. Araçlarda bulunan yolcu taşıma sınırlamaları nedeniyle problem Çok Amaçlı Kapasite Kısıtlı Araç Rotalama problemi olarak tanımlanmıştır. Problemin çözümünde Karınca Kolonisi Optimizasyonu Algoritması kullanılmıştır.

(Söyler and Keskindürk, 2007) çalışmasında; farklı KKO varyantlarına değinilmiş, örnek bir gezgin satıcı problemi üzerinden algoritmanın performansı, diğer yöntemlerle karşılaştırılmıştır.

(Alaykırın and Engin, 2005) çalışmasında; örnek gezgin satıcı problemleri, Visual Basic ile geliştirilen Karınca Programı yardımıyla çözümlenmiş ve elde edilen sonuçlar optimal değerlerle karşılaştırılmıştır.

ARAŞTIRMA BULGULARI

Parçacık Sürü Optimizasyonu (PSO) ile Çoklu Gezgin Satıcı Problemi (ÇGSP) için Şehir Kümeleme İşlemi

N adet şehir ve m adet satıcıdan oluşan bir çoklu gezgin satıcı probleminde bu şehirleri satıcılara dağıtarak her satıcının belirli bir şehir kümesinden sorumlu olduğu bir çözüm elde etmek amaçlanır.

$D = [d_{ij}]$ şehirler arasındaki mesafe matrisi (d_{ij} ; i . ve j . şehir arasındaki mesafe),

C_k ; k . kümeye atanmış şehirlerin indeksi ($k = 1, 2, \dots, m$),

S parçacığın durumu, her şehrin atanmış olduğu küme (S_i şehrin atanmış olduğu küme indeksini temsil eder, $S_i \in \{1, 2, \dots, m\}$). Böylece her şehir yalnızca bir satıcıya atanır. olmak üzere toplam mesafeyi minimize eden bir uygunluk fonksiyonu

$$f(S) = \sum_{k=1}^m \sum_{i \in C_k} \sum_{\substack{j \in C_k \\ j \neq i}} d_{ij} \quad (4.1)$$

şeklinde tanımlanabilir.

Burada

$C_k = \{i: S_i = k\}$; k . kümeye atanmış şehirlerin indeks kümesi (Burada her satıcıya en az bir şehir atanması şartı sağlanır),

$\sum_{i \in C_k} \sum_{\substack{j \in C_k \\ j \neq i}} d_{ij}$; k . küme içindeki şehirlerin toplam mesafesi

şeklindedir.

Her bir parçacığın konumu ve hızı

$$V_i^{(t+1)} = w \cdot V_i^{(t)} + c_1 \cdot r_1 \cdot (p_{best_i}^{(t)} - S_i^{(t)}) + c_2 \cdot r_2 \cdot (g_{best}^{(t)} - S_i^{(t)}) \quad (4.2)$$

$$S_i^{(t+1)} = S_i^{(t)} + V_i^{(t+1)} \quad (4.3)$$

eşitlikleri ile güncellenir. Burada,

w ; eylemsizlik katsayısı,

$V_i^{(t)}$; i . parçacığın t . iterasyondaki hızı,

c_1 ; yerel öğrenme katsayısı, c_2 ; global öğrenme katsayısı,

r_1, r_2 ; $[0,1]$ aralığında rastgele sayılar,

$p_{best_i}^{(t)}$; i . Parçacığın t . iterasyondaki kendi en iyi pozisyonu,

$g_{best}^{(t)}$; tüm parçacıkların t . iterasyondaki sürüdeki en iyi pozisyonu,

$S_i^{(t)}$; i . parçacığın t . iterasyondaki konumudur.

Ayrıca parçacıkların pozisyonları kümeler arasında kalmalıdır ($S_i \in [1, m]$). Bu nedenle pozisyon güncellemesi aşağıdaki şekilde sınırlandırılır ve yuvarlanır:

$$S_i^{(t+1)} = \text{round}(\max(1, \min(m, S_i^{(t+1)}))). \quad (4.4)$$

Bireysel ve Global En İyi Durum Güncellemeleri;

Her parçacık kendi geçmişindeki en iyi durumunu ($p_{best_i}^{(t)}$) ve o duruma ait uygunluk değerini ($f(p_{best_i}^{(t)})$) saklar.

Eğer $f(S_i^{(t+1)}) < f(p_{best_i}^{(t)})$, o zaman $p_{best_i}^{(t+1)} = S_i^{(t+1)}$, $f(p_{best_i}^{(t+1)}) = f(S_i^{(t+1)})$

ile kendi en iyi durumunu günceller.

Tüm parçacıkların en iyi durumları arasında en iyi olanı g_{best} olarak seçilir. Global en iyi durum güncelleme

$$g_{best}^{(t+1)} = \arg \min_i f(p_{best_i}^{(t+1)}), \quad f(g_{best}^{(t+1)}) = \min_i f(p_{best_i}^{(t+1)})$$

şeklinde yapılır.

Şimdi bu metodun yakınsamasını inceleyelim.

Öncelikle tek bir parçacığın (parçacık i) durumunu göz önüne alalım. (4.2) ile verilen eşitlik (4.3) ifadesinde yerine yazılırsa

$$S_i^{(t+1)} = (1 + w - (c_1 \cdot r_1 + c_2 \cdot r_2))S_i^{(t)} - wS_i^{(t-1)} + c_1 \cdot r_1 \cdot p_{best_i}^{(t)} + c_2 \cdot r_2 \cdot g_{best}^{(t)} \quad (4.5)$$

tekrarlama bağıntısı elde edilir.

(4.5) ifadesinde rastgele sayılar bulunur ve $S_i^{(0)}$, $S_i^{(1)}$ değerleri de rastgele sayılardır. Dolayısıyla her $S_i^{(t)}$ bir rastgele değişken olarak ele alınmalı ve $\{S_i^{(t)}\}$ yinelemeli süreci, stokastik bir süreç olarak düşünülmelidir. Bu aşamada önce parçacık pozisyonunun beklentisi ve varyansı incelenir, çünkü bu stokastik süreçten ziyade deterministik süreçtir, böylece karşılık gelen garantili yakınsama özellikleri doğrudan analiz edilebilir.

$ES_i^{(t)}$, rastgele $S_i^{(t)}$ değişkeninin beklentisi olmak üzere (4.5) ifadesinden hareketle $\{ES_i^{(t)}\}$ dizisinin iterasyon denklemi

$$ES_i^{(t+1)} = \left(1 + w - \frac{c_1 + c_2}{2}\right) ES_i^{(t)} - wES_i^{(t-1)} + \frac{c_1 p_{best_i}^{(t)} + c_2 g_{best}^{(t)}}{2} \quad (4.6)$$

şeklinde ifade edilebilir. (4.6) ifadesiyle gösterilen yinelemeli sürecin karakteristik denklemi

$$\lambda^2 - \left(1 + w - \frac{c_1 + c_2}{2}\right) \lambda + w = 0 \quad (4.7)$$

olur.

Teorem: Verilmiş $w, c_1, c_2 \geq 0$ sayıları için $0 \leq w < 1$ ve $0 < c_1 + c_2 < 4(1 + w)$ olduğunda $\{ES_i^{(t)}\}$ dizisi $\frac{c_1 p_{best_i}^{(t)} + c_2 g_{best}^{(t)}}{c_1 + c_2}$ değerine yakınsar.

Teoremin ispatı (Eberhart et al 2001) çalışmasındakine benzer olarak yapılabilir.

PSO ile Kümeleme İşlemi için Bir Simülasyon

Bu bölümde PSO ile kümelemeyi daha iyi anlamak için bir iterasyondan diğerine geçişi simüle edeceğiz.

$N = 6$ şehir ve $m = 2$ satıcıdan oluşan bir çoklu gezgin satıcı probleminde

$$D = \begin{bmatrix} 0 & 12 & 10 & 29 & 21 & 16 \\ 12 & 0 & 14 & 25 & 19 & 23 \\ 10 & 14 & 0 & 18 & 15 & 20 \\ 29 & 25 & 18 & 0 & 11 & 13 \\ 21 & 19 & 15 & 11 & 0 & 17 \\ 16 & 23 & 20 & 13 & 17 & 0 \end{bmatrix}$$

mesafe matrisini göz önüne alalım.

İki parçacık ile simülasyona başlayalım. Bu parçacıkların başlangıç konumları ve hızları sırasıyla şu şekilde olsun.

$$\text{Parçacık 1: } S_1^{(0)} = [1,1,2,2,1,2] \text{ ve } V_1^{(0)} = [0.2, -0.3, 0.1, -0.2, 0.3, -0.1]$$

$$\text{Parçacık 2: } S_2^{(0)} = [2,1,1,2,2,1] \text{ ve } V_2^{(0)} = [-0.2, 0.3, -0.1, 0.2, -0.3, 0.1]$$

Başlangıç Amaç Fonksiyonunu Hesaplama;

Amaç fonksiyonu $f(S)$, kümeler içindeki toplam mesafeyi minimize eder.

Parçacık 1:

- Küme 1: Şehirler $C_{11}^{(0)} = \{1,2,5\}$, Mesafe: $d_{12} + d_{15} + d_{25} = 12 + 21 + 19 = 52$

- Küme 2: Şehirler $C_{12}^{(0)} = \{3,4,6\}$, Mesafe: $d_{34} + d_{36} + d_{46} = 18 + 20 + 13 = 51$

Uygunluk Değeri: $f(S_1^{(0)}) = 52 + 51 = 103$

Parçacık 2:

- Küme 1: Şehirler $C_{21}^{(0)} = \{2,3,6\}$, Mesafe: $d_{23} + d_{26} + d_{36} = 14 + 23 + 20 = 57$

- Küme 2: Şehirler $C_{22}^{(0)} = \{1,4,5\}$, Mesafe: $d_{14} + d_{15} + d_{45} = 29 + 21 + 11 = 61$

Uygunluk Değeri: $f(S_2^{(0)}) = 57 + 61 = 118$

Başlangıç p_{best} ve g_{best}

- $f(p_{best_1}^{(0)}) = 103$, $S_{p_{best_1}}^{(0)} = [1,1,2,2,1,2]$

- $f(p_{best_2}^{(0)}) = 118$, $S_{p_{best_2}}^{(0)} = [2,1,1,2,2,1]$

- $f(g_{best}^{(0)}) = 103$, $S_{g_{best}}^{(0)} = S_{p_{best_1}}^{(0)} = [1,1,2,2,1,2]$

Hız Güncelleme

Hız güncelleme formülü

$$V_i^{(t+1)} = w \cdot V_i^{(t)} + c_1 \cdot r_1 \cdot (p_{best_i}^{(t)} - S_i^{(t)}) + c_2 \cdot r_2 \cdot (g_{best}^{(t)} - S_i^{(t)})$$

Parametreler: $w = 0.5, c_1 = 1.5, c_2 = 1.5$ ve Rastgele değerler $r_1 = 0.6, r_2 = 0.7$ olsun.

Parçacık 1 için Hız

$$\begin{aligned} V_1^{(1)} &= (0.5) \cdot [0.2, -0.3, 0.1, -0.2, 0.3, -0.1] + (1.5) \cdot (0.6) \cdot ([1,1,2,2,1,2] - [1,1,2,2,1,2]) \\ &+ (1.5) \cdot (0.7) \cdot ([1,1,2,2,1,2] - [1,1,2,2,1,2]) = \\ &[0.1, -0.15, 0.05, -0.1, 0.15, -0.05] \end{aligned}$$

Parçacık 2 için Hız

$$\begin{aligned} V_2^{(1)} &= (0.5) \cdot [-0.2, 0.3, -0.1, 0.2, -0.3, 0.1] + (1.5) \cdot (0.6) \cdot ([2,1,1,2,2,1] - [2,1,1,2,2,1]) \\ &+ (1.5) \cdot (0.7) \cdot ([1,1,2,2,1,2] - [2,1,1,2,2,1]) \end{aligned}$$

$$\begin{aligned} V_2^{(1)} &= [-0.1, 0.15, -0.05, 0.1, -0.15, 0.05] + (1.5) \cdot (0.7) \cdot [-1, 0, 1, 0, -1, 1] \\ &= [-1.15, 0.15, 1, 0, -1.2, 1.1] \end{aligned}$$

Konum Güncelleme

$$S_i^{(t+1)} = S_i^{(t)} + V_i^{(t+1)}$$

Konumlar, hızlara eklenir ve $[1,2]$ aralığında yuvarlanır.

Parçacık 1 için yeni konum

$$S_1^{(1)} = \text{round}([1,1,2,2,1,2] + [0.1, -0.15, 0.05, -0.1, 0.15, -0.05]) = [1,1,2,2,1,2]$$

Parçacık 2 için yeni konum

$$S_2^{(1)} = \text{round}([2,1,1,2,2,1] + [-1.15, 0.15, 1, 0, -1.2, 1.1]) = [1,1,2,2,1,2]$$

Uygunluk Fonksiyonlarını Güncelleme

Parçacık 1: $S_1^{(0)} = S_1^{(1)}$ olduğu için uygunluk değişmedi $f(S_1^{(1)}) = 103$

Parçacık 2: Yeni konum $S_2^{(1)} = [1,1,2,2,1,2]$

Böylece parçacık 2, parçacık 1' in konumuna geldi.

- Küme 1: Şehirler $C_{21}^{(1)} = \{1,2,5\}$, Mesafe: $d_{12} + d_{15} + d_{25} = 12 + 21 + 19 = 52$
- Küme 2: Şehirler $C_{22}^{(1)} = \{3,4,6\}$, Mesafe: $d_{34} + d_{36} + d_{46} = 18 + 20 + 13 = 51$

Uygunluk Değeri: $f(S_2^{(1)}) = 103$

$f(S_2^{(1)}) < f(p_{best_2}^{(0)})$ olduğundan $p_{best_2}^{(1)} = S_2^{(1)}$, $f(p_{best_2}^{(1)}) = f(S_2^{(1)})$

$$g_{best}^{(1)} = \arg \min_i f(p_{best_i}^{(1)}), \quad f(g_{best}^{(1)}) = \min_i f(p_{best_i}^{(1)})$$

Yeni p_{best} ve g_{best}

- $f(p_{best_1}^{(1)}) = 103$, $S_{p_{best_1}}^{(1)} = [1,1,2,2,1,2]$
- $f(p_{best_2}^{(1)}) = 103$, $S_{p_{best_2}}^{(1)} = [1,1,2,2,1,2]$
- $f(g_{best}^{(1)}) = 103$, $S_{g_{best}}^{(1)} = [1,1,2,2,1,2]$

Bu iterasyondan sonra 1. Satıcı için elde edilen şehirler kümesi $\{1,2,5\}$, 2. Satıcı için elde edilen şehirler kümesi $\{3,4,6\}$ olarak bulunmuştur.

Karınca Kolonisi Optimizasyonu (KKO) ile Çoklu Gezgin Satıcı Problemi (ÇGSP) için Rota Bulma İşlemi

Bu kısımda amaç her bir C_k kümesi için en kısa rotayı bulmak, dolayısıyla toplam mesafeyi minimize etmek amaçlanır. Amaç fonksiyonu

$$f_k(\text{rota}) = \sum_{i=1}^{|C_k|} d_{\text{rota}(i), \text{rota}(i+1)} \quad (4.8)$$

olup burada

- $f_k(\text{rota})$: k . küme için toplam rota uzunluğu
- $\text{rota}(i)$: Rotadaki i . şehir
- $|C_k|$: k . kümeye ait şehirlerin sayısı
- d_{ij} : Şehirler arasındaki mesafe

şeklindedir. Yöntemin amacı $f_k(\text{rota})$ değerini minimum yapan rota sırasını bulmaktır.

Feromon Matrisi (τ)

Feromon matrisi rotaların çekiciliğini belirler ve şu şekilde tanımlanır:

$\tau_{ij}(t)$: i ' den j ' ye olan yol üzerindeki feromon yoğunluğu.

Başlangıçta tüm yollar için feromon yoğunluğu eşit atanır:

$$\tau_{ij}(0) = \tau_{\text{init}}, \forall i, j$$

Olasılık Hesaplama

Bir karınca bulunduğu şehirden bir sonraki şehri seçerken olasılıkları hesaplar:

$$P_{ij}(t) = \begin{cases} \frac{\tau_{ij}(t)^\alpha \cdot \eta_{ij}^\beta}{\sum_{k \in \text{visited}} \tau_{ik}(t)^\alpha \cdot \eta_{ik}^\beta}, & \text{eğer } j \notin \text{visited} \\ 0, & \text{eğer } j \in \text{visited} \end{cases} \quad (4.9)$$

Burada:

- $P_{ij}(t)$: t . iterasyonda i ' den j ' ye geçiş olasılığı
- $\tau_{ij}(t)$: Feromon yoğunluğu
- $\eta_{ij} = \frac{1}{d_{ij}}$: şehirler arasındaki mesafeye dayalı sezgisel bilgi
- α : Feromonun etkisini kontrol eden katsayı
- β : Sezgisel bilginin etkisini kontrol eden katsayı
- visited: Karıncaların ziyaret ettiği şehirler kümesi

Rota Uzunluğu Hesaplama

Bir karınca tarafından izlenen rota için toplam uzunluk şu şekilde hesaplanır:

$$L_{\text{rota}} = \sum_{i=1}^{|C_k|} d_{\text{rota}(i), \text{rota}(i+1)} \quad (4.10)$$

Burada $\text{rota}(i + 1)$, $\text{rota}(i)$ 'den sonra ziyaret edilen şehri ifade eder.

Feromon Güncellenmesi

Feromon matrisi her iterasyon sonunda şu şekilde güncellenir.

- Feromon Buharlaşması: ρ feromon buharlaşma oranı olmak üzere

$$\tau_{ij}(t + 1) = (1 - \rho)\tau_{ij}(t) \quad (4.11)$$

ile ifade edilir.

- Feromon Artışı: Q Feromon miktarını belirleyen sabit ve L_{rota} ilgili karıncanın izlediği rotanın uzunluğu olmak üzere

$$\tau_{ij}(t + 1) = \tau_{ij}(t) + \sum_{\text{karıncalar}} \frac{Q}{L_{\text{rota}}} \quad (4.12)$$

ile ifade edilir.

En İyi Rota

Her iterasyonda en iyi rota bulunur:

$$\text{rota}_{\text{en iyi}} = \arg \min_{n \in \{1, 2, \dots, K\}} L_{\text{rota}_n} \quad (4.13)$$

Burada L_{rota_n} ; iterasyondaki n . karıncanın rota uzunluğudur.

Toplam Optimal Uzunluk

Her küme için en iyi rotaların uzunlukları toplanır ve toplam optimal uzunluk hesaplanır.

$$L_{\text{toplaml}} = \sum_{k=1}^m f_k(\text{rota}_{\text{en iyi}}) \quad (4.14)$$

Karıncı Kolonisi Optimizasyonu (KKO) algoritmalarının yakınsaklığı ve yakınsaklık hızı üzerine literatürde bazı teorik çalışmalar vardır, ancak bu konuda genel ve güçlü bir teorem (örneğin tüm KKO türleri için geçerli, net yakınsaklık garantisi sunan bir teorem) bulunmamaktadır. Bunun yerine şu yaklaşımlar öne çıkmaktadır:

Gutjahr (2000, 2002), KKO algoritmalarının yakınsaklığını olasılıksal olarak inceleyen en bilinen teorik çalışmalardandır. Bu çalışmada eğer belirli koşullar sağlanırsa (örneğin yeterince zaman verilir ve feromon buharlaşması sıfıra yakın tutulursa), algoritma optimum çözüme olasılık 1 ile yakınsadığı belirlenmiştir. Bu teoriler genelde Stochastic Convergence (olasılıksal yakınsaklık) türündedir ve "sonsuz iterasyonda" optimum çözüme varıldığını söyler.

Bu çalışmalarda elde edilen sonuçlar şöyle özetlenebilir:

Ω tüm çözümlerin kümesi ve $w^* \in \Omega$: global optimal çözüm olmak üzere KKO algoritması her iterasyonda bir $w_t \in \Omega$ çözümü üretir. $\tau_{ij} \in [\tau_{min}, \tau_{max}]$ olacak şekilde feromon sınırları belirlenmiş olsun. Bir karıncanın çözüm oluştururken, i . düğümünden j . düğümüne geçme olasılığı şu şekilde tanımlanır:

$$P_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}^\beta]}{\sum_{l \in N_i^k(t)} [\tau_{il}(t)]^\alpha \cdot [\eta_{il}^\beta]}, & j \in N_i^k(t) \text{ ise} \\ 0, & \text{aksi halde} \end{cases} \quad (4.15)$$

Burada $N_i^k(t)$: karınca k 'nin o anki geçerli komşuları (ziyaret etmediği düğümler) dir.

Varsayalım ki bir karınca k, n düğümden oluşan bir $w^k = (i_1, i_2, \dots, i_n)$ çözümü üretiyor. Bu çözümün oluşturulma olasılığı, her adımda yapılan geçişin olasılıklarının çarpımıdır.

$$P(w^k) = \prod_{r=1}^{n-1} P_{i_r i_{r+1}}^k(t)$$

Bu durumda, her iterasyonda her çözümün seçilme olasılığı sıfırdan büyüktür,

$$\forall w \in \Omega, P(w^k) \geq \varepsilon > 0$$

Bu, çözüm uzayının tamamen keşfedilebilir olduğunu garanti eder. Her iterasyon durumu (feromon seviyesi) bir sonraki adıma olasılıksal olarak bağlıdır. Sistemin özellikleri uygun şekilde tanımlanmışsa (ör. feromonlar sıfıra inmiyor, maksimumu geçmiyor), sistem kullanışlı olur. En iyi çözümler tekrarlandığında, onların kenarlarındaki feromonlar sürekli güçlenir:

$$\tau_{ij}^{t+1} = \min\{(1-p)\tau_{ij}^{t+1} + \Delta\tau_{ij}^{best}, \tau_{max}\}$$

Bu mekanizma sayesinde: En iyi çözümün kenarları daha yüksek olasılıkla tekrar seçilir ve zamanla tek bir çözüm (optimum) baskın hale gelir ve

$$\lim_{t \rightarrow \infty} P(w^k = w^*) = 1 \quad (4.16)$$

olur, yani eğer algoritma sonsuz iterasyon boyunca çalıştırılırsa, bulunan çözümün global optimum olma olasılığı 1'e (yani kesinliğe) yaklaşır.

Yakınsaklık hızına dair detaylı teorik sonuçlar oldukça sınırlıdır. Bazı çalışmalar Stützle and Dorigo (2004) parametrelerin (ör. feromon buharlaşma oranı α, β değerleri) yakınsaklık hızını etkilediğini deneysel olarak göstermektedir. Ancak bu konuda genellikle deneysel analizler yapılmış, katı matematiksel hız teoremleri sunulmamıştır.

Karınca Kolonisi Optimizasyonu ile Optimal Rota Bulma Simülasyonu

PSO ile kümeleme kısmındaki simülasyona geri dönelim. $N = 6$ şehir ve $m = 2$ satıcı için PSO sonucunda Küme 1 için atanan şehirler $C_1 = \{1,2,5\}$ olmuştu. D mesafe matrisinin Küme 1 için şehirler arası mesafeleri içeren alt matrisi

$$D_{C_1} = \begin{bmatrix} 0 & 12 & 21 \\ 12 & 0 & 19 \\ 21 & 19 & 0 \end{bmatrix}$$

şeklindedir.

KKO Parametreleri:

- Karınca sayısı: $\text{num_ants}(K) = 2$
- Feromon etkisi katsayısı: $\alpha = 1$
- Sezgisel Bilgi Katsayısı: $\beta = 2$
- Feromon buharlaşma oranı: $\rho = 0.1$
- Başlangıç feromon değeri: $\tau_{\text{init}} = 1$

Olasılık Hesaplama:

Bir karınca i . şehirdeyken j ' ye gitme olasılığı

$$P_{ij} = \frac{\tau_{ij}^\alpha \cdot \eta_{ij}^\beta}{\sum_{k \notin \text{visited}} \tau_{ik}^\alpha \cdot \eta_{ik}^\beta}$$

şeklindedir.

İlk Şehirden (1) Başlarken:

- Başlangıç feromon matrisi:

$$\tau = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

- Sezgisel bilgi ($\eta_{ij} = 1/d_{ij}$):

$$\eta = \begin{bmatrix} 0 & \frac{1}{12} & \frac{1}{21} \\ \frac{1}{12} & 0 & \frac{1}{19} \\ \frac{1}{21} & \frac{1}{19} & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0.0833 & 0.0476 \\ 0.0833 & 0 & 0.0526 \\ 0.0476 & 0.0526 & 0 \end{bmatrix}$$

Başlangıç İterasyonu

Karınca 1' in Rotası: Karınca 1, rastgele 1. şehirden başlasın.

1. Şehirden hareket (1 → j)

- Olasılık (P_{1j}) için:

$$P_{12} = \frac{\tau_{12}^{\alpha} \eta_{12}^{\beta}}{\tau_{12}^{\alpha} \eta_{12}^{\beta} + \tau_{15}^{\alpha} \eta_{15}^{\beta}} = \frac{1 \cdot (0.0833)^2}{1 \cdot (0.0833)^2 + 1 \cdot (0.0476)^2} = \frac{0.00694}{0.00694 + 0.00227} = 0.753$$

$$P_{15} = 1 - P_{12} = 0.247$$

Karınca 1 için İlk Seçim:

- Rastgele bir olasılık $r = 0.7$ olsun.
- $r < P_{12}$ olduğundan karınca 1, 2. şehri seçer.

2. Şehirden hareket (2 → j)

Ziyaret edilmeyen şehir 5, geçiş olasılığı: $P_{25} = 1$ (tek seçenek olduğu için). Karınca,

5. Şehre gider.

5. Şehirden hareket (5 → 1). Başlangıca dönerek döngüyü tamamlar.

Rota: [1,2,5,1]: Rota uzunluğu $L_{\text{rota}_1} = d_{12} + d_{25} + d_{51} = 12 + 19 + 21 = 52$

Karınca 2' nin Rotası: Karınca 2, rastgele 5. şehirden başlasın.

5. Şehirden hareket (5 → j)

- Olasılık (P_{5j}) için:

$$P_{52} = \frac{\tau_{52}^{\alpha} \eta_{52}^{\beta}}{\tau_{52}^{\alpha} \eta_{52}^{\beta} + \tau_{51}^{\alpha} \eta_{51}^{\beta}} = \frac{1 \cdot (0.0526)^2}{1 \cdot (0.0526)^2 + 1 \cdot (0.0476)^2} = \frac{0.00277}{0.00277 + 0.00227} = 0.550$$

$$P_{51} = 1 - P_{52} = 0.450$$

Karınca 2 için İlk Seçim:

- Rastgele bir olasılık $r = 0.6$ olsun.
- $r > P_{52}$ olduğundan karınca 2, 1. şehri seçer.

1. Şehirden hareket (1 → j)

Ziyaret edilmeyen şehir 2, geçiş olasılığı: $P_{12} = 1$ (tek seçenek olduğu için). Karınca,

2. Şehre gider.

2. Şehirden hareket (2 → 5). Başlangıca dönerek döngüyü tamamlar.

Rota: [5,1,2,5]: Rota uzunluğu $L_{\text{rota}_2} = d_{51} + d_{12} + d_{25} = 21 + 12 + 19 = 52$

Feromon Güncellemesi:

Bir iterasyon sonunda feromonlar şu şekilde güncellenir:

- Buharlaştırma:

$$\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) = (1 - 0.1)\tau = 0.9\tau = \begin{bmatrix} 0 & 0.9 & 0.9 \\ 0.9 & 0 & 0.9 \\ 0.9 & 0.9 & 0 \end{bmatrix}$$

- Feromon Artışı: Her karınca için feromon artışı:

$$\tau_{ij}(t+1) = \tau_{ij}(t) + \sum_{\text{karıncalar}} \frac{Q}{L_{\text{rota}}}$$

Burada $Q = 1, L_{\text{rota}} = 52$. Örneğin, rota $[1,2,5,1]$ için;

$$\tau_{12} = (0.9).1 + 2 \cdot \frac{1}{52} \cong 0.9384 \text{ ((1} \leftrightarrow \text{2): 1. ve 2. karınca bu yolu kullandı (2 karınca))}$$

$$\tau_{25} = (0.9).1 + 2 \cdot \frac{1}{52} \cong 0.9384 \text{ ((2} \leftrightarrow \text{5): 1. ve 2. karınca bu yolu kullandı (2 karınca))}$$

$$\tau_{51} = (0.9).1 + 2 \cdot \frac{1}{52} \cong 0.9384 \text{ ((5} \leftrightarrow \text{1): 1. ve 2. karınca bu yolu kullandı (2 karınca))}$$

olup güncellenmiş feromon matrisi

$$\tau = \begin{bmatrix} 0 & 0.9384 & 0.9384 \\ 0.9384 & 0 & 0.9384 \\ 0.9384 & 0.9384 & 0 \end{bmatrix}$$

olur. Minimum değer;

$$\text{rota}_{\text{en iyi}} = \arg \min \{L_{\text{rota}_1}, L_{\text{rota}_2}\}$$

Bu durumda iki rota da aynı uzunluğa sahiptir. $\text{rota}_{\text{en iyi}}$ herhangi birisi olabilir (örneğin, $[1,2,5,1]$)

Hibrit Yöntemlerin Karşılaştırılmaları

GitHub tsplib: Simetrik TSP verileri deposu sitesinden alınan eil51, berlin52, eil76, rat99 tsplib problemleri Matlab dilinde uygulamaları yapıldı ve PSO+KKO hibrit yöntemi PSO+GA ve GA+YKA hibrit yöntemleri ile mesafeye dayalı olarak karşılaştırıldı.

eil51: Gezgin satıcı problemi literatüründe kullanılan bir referans veri setidir. Adını, Christofides and Eilon (1969)'da yayınlanan bir makaleden alan ve 51 şehir (düğüm) içeren bir problemdir.

berlin52: Gezgin satıcı problemi literatüründe yaygın kullanılan bir referans veri setidir. Alman başkenti Berlin'deki 52 önemli noktanın (örneğin, Brandenburg Kapısı, Berlin Katedrali) koordinatlarından oluşur ve TSP algoritmalarının test edilmesinde standart olarak kabul edilir (Reinelt 1991).

eil76: Gezgin satıcı problemi literatüründe kullanılan bir referans veri setidir Adını, Christofides and Eilon (1979)'da yayınlanan bir makaleden alan ve 76 şehir (düğüm) içeren bir problemdir.

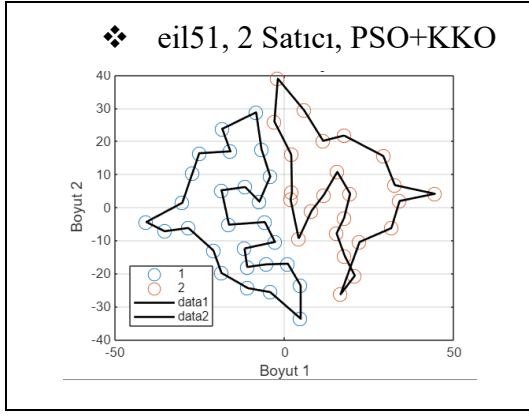
rat99: Gezgin satıcı problemi literatüründe kullanılan bir referans veri setidir. Adındaki "rat" ifadesi, problemin rastgele (random) üretilmiş koordinatlar içerdiğini belirtir. TSPLIB kütüphanesinde standartlaştırılmıştır ve algoritma testlerinde yaygın olarak kullanılır (Reinelt 1991).

Tablo 1 de 3 farklı hibrit model aynı problemler üzerinden 2, 3, 5 ve 7 satıcı için karşılaştırılmıştır. Her bir veri setinde her bir satıcı sayısı için 10 adet optimizasyon yapılmıştır. Bu optimizasyonlarda elde edilen en kısa, en uzun ve ortalama mesafeler aşağıdaki gibi elde edilmiştir.

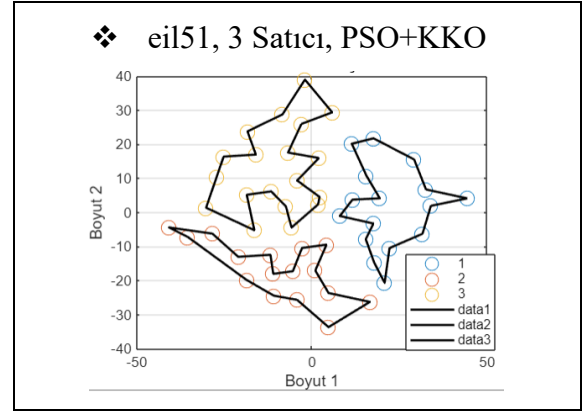
Tablo 1. Hibrit Algoritmalar Karşılaştırma Tablosu

Hibrit algoritma ismi		PSO+KKO				PSO+GA				GA+YKA			
Problem ismi	Satıcı sayısı	2	3	5	7	2	3	5	7	2	3	5	7
eil51	Ortalama mesafe	442	453,8	479,5	490,3	1209,7	1105,6	1045	1045,2	1115,9	959,9	745,9	631,3
	En kısa mesafe	432	444	463	469	1102	992	982	948	1073	951	721	607
	En uzun mesafe	454	471	502	508	1309	1195	1117	1155	1152	954	791	644
berlin52	Ortalama mesafe	7875	8205	8252	8564	21187	19695	19086	18454	18421	16273	11372	10069
	En kısa mesafe	7872	8723	7470	7862	19664	17996	17716	16895	18068	16043	10970	9684
	En uzun mesafe	7877	8723	9295	9399	21777	21492	20803	20042	20738	16419	11493	10659
eil76	Ortalama mesafe	566	578	594	616	1948	1862	1710	1630	1739	1417	1117	979
	En kısa mesafe	557	566	573	592	1790	1799	1597	1461	1687	1393	1070	964
	En uzun mesafe	574	580	615	655	2059	1922	1794	1775	1759	1426	1143	1001
rat99	Ortalama mesafe	1279	1282	1325	1389	6359	6095	5509	5263	2104	2113	1934	1837
	En kısa mesafe	1267	1253	1279	1343	6043	5769	5135	5002	2101	2113	1887	1809
	En uzun mesafe	1292	1304	1403	1407	6716	6306	5756	5629	2116	2113	2016	1879

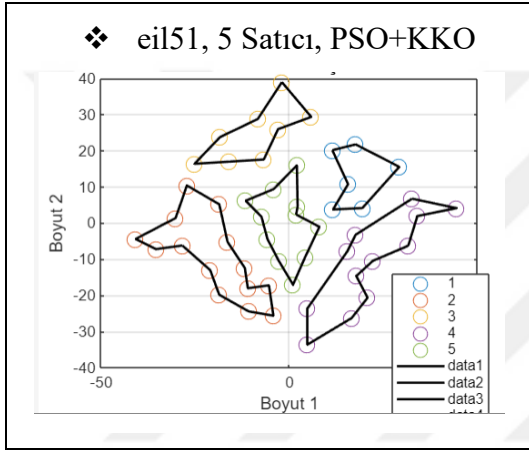
Ayrıca 3 hibrit algoritma içinden en iyi sonuçları veren PSO+KKO hibrit algoritmasının sonuçları aşağıdakiler gibi görselleştirilmiştir.



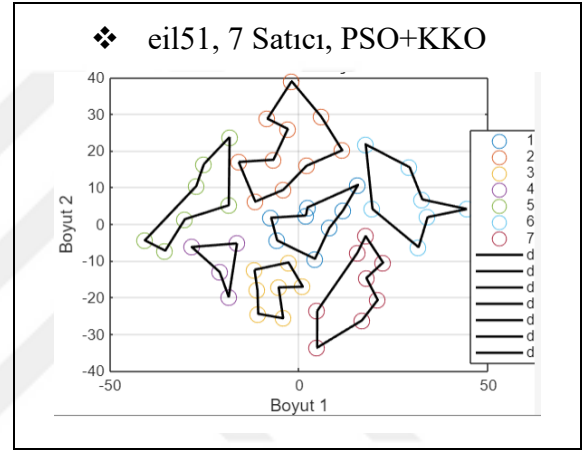
Şekil 2. eil51, 2 Satıcı
Tüm rotaların uzunluklarının toplamı: 452.43



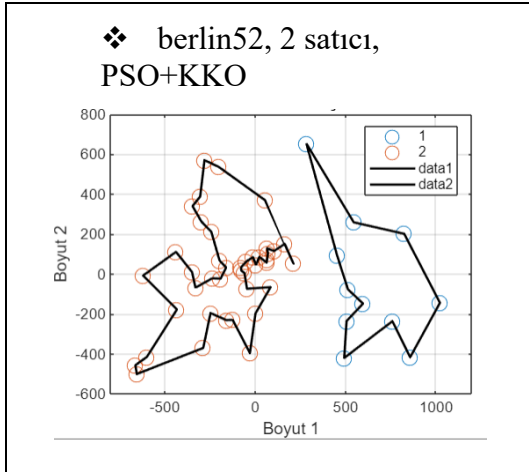
Şekil 3. eil51, 3 Satıcı
Tüm rotaların uzunluklarının toplamı: 466.07



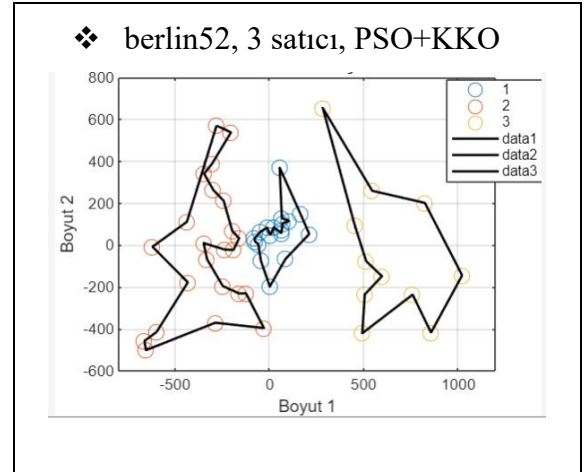
Şekil 4. eil51, 5 Satıcı
Tüm rotaların uzunluklarının toplamı: 469.80



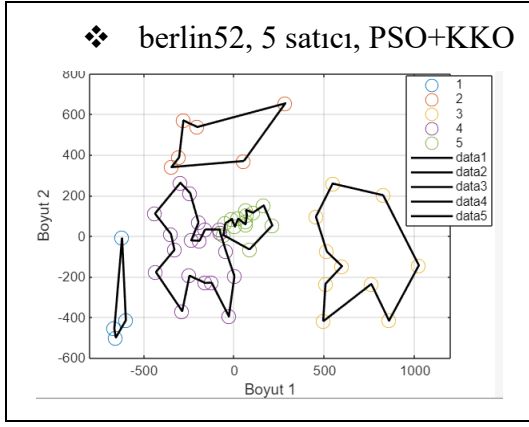
Şekil 5. eil51, 7 Satıcı
Tüm rotaların uzunluklarının toplamı: 507.77



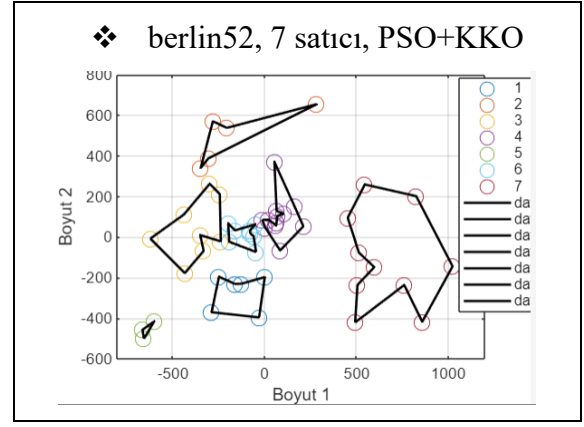
Şekil 6. berlin52, 2 satıcı
Tüm rotaların uzunluklarının toplamı: 7872.93



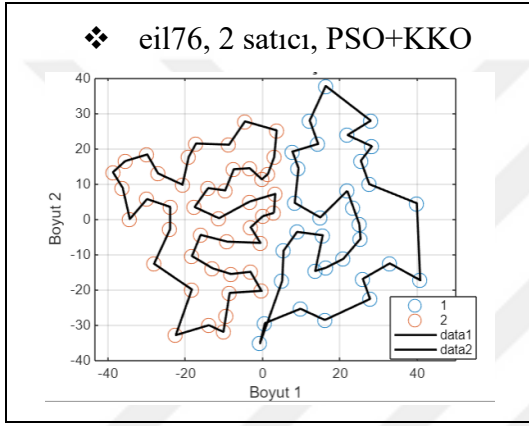
Şekil 7. berlin52, 3 satıcı
Tüm rotaların uzunluklarının toplamı: 8205.14



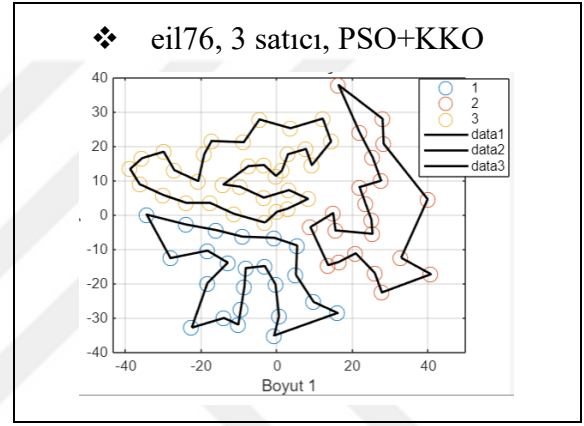
Şekil 8. berlin52, 5 satıcı
Tüm rotaların uzunluklarının toplamı: 8064.34



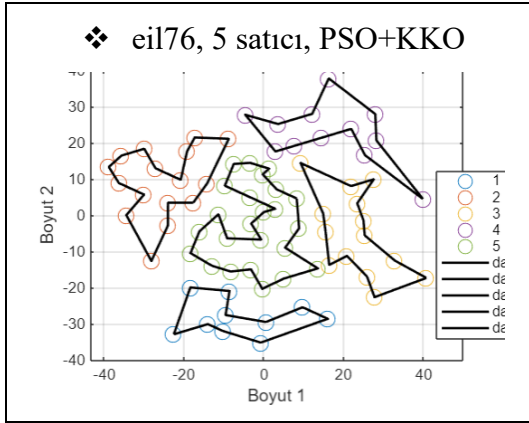
Şekil 9. berlin52, 7 satıcı.
Tüm rotaların uzunluklarının toplamı: 8039.62



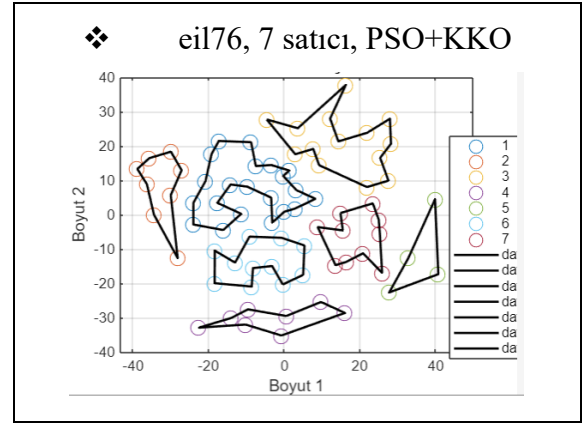
Şekil 10. eil76, 2 satıcı
Tüm rotaların uzunluklarının toplamı: 564.81



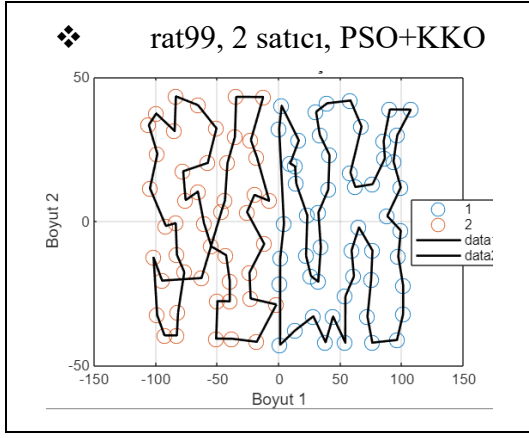
Şekil 11. eil76, 3 satıcı
Tüm rotaların uzunluklarının toplamı: 580.69



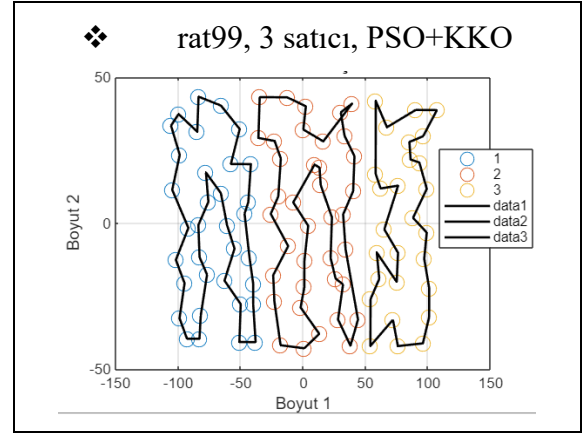
Şekil 12. eil76, 5 satıcı
Tüm rotaların uzunluklarının toplamı: 615.40



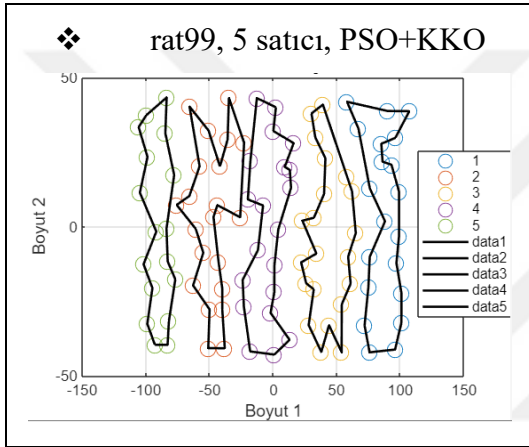
Şekil 13. eil76, 7 satıcı
Tüm rotaların uzunluklarının toplamı: 613.89



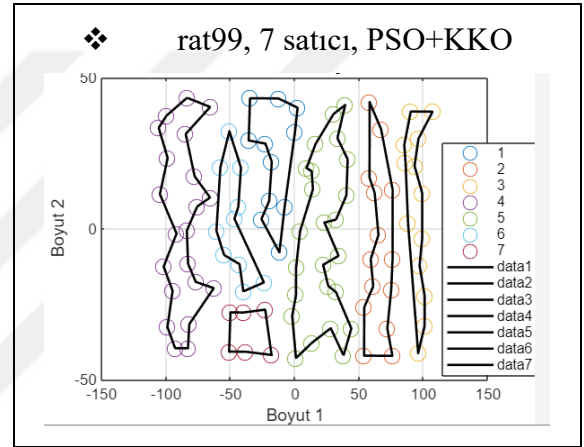
Şekil 14. rat99, 2 satıcı
Tüm rotaların uzunluklarının toplamı: 1292.35



Şekil 15. rat99, 3 satıcı
Tüm rotaların uzunluklarının toplamı: 1279.40



Şekil 16. rat99, 5 satıcı
Tüm rotaların uzunluklarının toplamı: 1296.01



Şekil 17. rat99, 7 satıcı
Tüm rotaların uzunluklarının toplamı: 1347.04

SONUÇ

Başarıları araştırılan bu 3 hibrit algoritmanın ilki kümeleme işlemi yaparken ikincisi rotalama işlemini yapmaktadır. Gezgin satıcı problemi literatüründe kullanılan referans veri setlerinin hepsinde PSO+KKO hibrit algoritmasının diğer iki hibrit algoritmaya göre üstünlüğü kanıtlanmıştır. PSO+KKO hibrit algoritmasında satıcı sayısı artıkça oluşan kümelemenin toplam mesafesinin küçük bir miktar arttığı gözlenmiştir. PSO+GA hibrit algoritmasında satıcı sayısı artıkça oluşan kümelemenin toplam mesafesinin azaldığı gözlenmiştir bu da gösterir ki GA eleman sayısı az olan kümelerde daha başarılı rotalar oluşturmaktadır. GA+YKA hibrit algoritmasında da satıcı sayısı artıkça oluşan kümelemenin toplam mesafesinin azaldığı gözlenmiştir bu da gösterir ki YKA eleman sayısı az olan kümelerde daha başarılı rotalar oluşturmaktadır. GA+YKA hibrit algoritması PSO+GA hibrit algoritmasına göre daha iyi sonuçlar vermiştir. Sonuç olarak PSO+KKO hibrit algoritması, 51-99 şehirli kümeler için 2'den 7 satıcıya kadar olan durumlarda başarısını kanıtlamıştır. Daha sonra yapılacak çalışmalarda satıcı sayıları ve kümelerdeki şehir sayıları artırılarak gerçek hayattaki daha başka problemlere çözüm bulmak için incelemeler yapılabilir, başka hibrit algoritmalar tasarlanıp üstün yönü olan algoritmalarından yeni hibrit algoritmalar oluşturulabilir.

KAYNAKÇA

- Alaykırın, K., & Engin, O. (2005). Karınca kolonileri metasezgiseli ve gezgin satıcı problemleri üzerinde bir uygulaması. *Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi*, 20(1), 69–76.
- Applegate, D. L., Bixby, R. E., Chvátal, V., & Cook, W. (2006). *The Traveling Salesman Problem: A Computational Study*. Princeton University Press.
- Aydın, Ö., Salur, M. U., & Başkaya, F. (2018). Duygu analizi için çoklu popülasyon tabanlı parçacık sürü optimizasyonu. *Türk Bilgisayar ve Bilişim Bilimleri Dergisi*. <https://dergipark.org.tr/en/download/article-file/484230>.
- Bektas, T. (2006). The Multiple Traveling Salesman Problem: An Overview of Formulations and Solution Procedures. *Omega*, 34(3), 209–219.
- Bellman, R. (1957). *Dynamic Programming*. Princeton University Press.
- Bezdek, J. C. (1981). *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press.
- Christofides, N., & Eilon, S. (1969). "An Algorithm for the Vehicle-Dispatching Problem." *Operational Research Quarterly*, 20(3), 309-318.
- Christofides, N., & Eilon, S. (1979). "An Algorithm for the Vehicle-Dispatching Problem." *Journal of the Operational Research Society*, 30(2), 93-103. <https://github.com/mastqe/tsplib>, Erişim Tarihi: 23.05.2025
- Clerc, M., & Kennedy, J. (2002). The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, 6(1), 58-73.
- Clerc, M., & Kennedy, J. (2002). *The particle swarm—Explosion, stability, and convergence in a multidimensional complex space*. *IEEE Transactions on Evolutionary Computation*, 6(1), 58–73. <https://doi.org/10.1109/4235.985692>
- Çınaroğlu, S., & Bulut, H. (2018). K-Ortalamlar ve parçacık sürü optimizasyonu tabanlı kümeleme algoritmaları için yeni ilkendirme yaklaşımları. *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, 33(2), 413-424. <https://doi.org/10.17341/gazimmfd.416350>.
- Çınaroğlu, S., & Bulut, H. (2018). New initialization approaches for the k-means and particle swarm optimization based clustering algorithms. *Journal of the Faculty of Engineering and Architecture of Gazi University*, 33(2), 413–423. <https://doi.org/10.17341/gazimmfd.416350>
- Deneubourg, J.-L., Aron, S., Goss, S., & Pasteels, J.-M. (1990). The self-organizing exploratory pattern of the Argentine ant. *Journal of Insect Behavior*, 3, 159.
- Dolapcı, B., Özcan, C. ve Özkaynak, E. (2024). Hiperspektral görüntülerin graf temelli boyuta indirgenerek sınıflandırılmasında sürü halinde ortaya çıkması yaklaşıyor. *Gümüşhane Üniversitesi Fen Bilimleri Dergisi*, 14(4),12191234. <https://doi.org/10.17714/gumusfenbil.1493821>.
- Dorigo, M., and Stützle, T. 2004. "Ant Colony Optimization." MIT Press.
- Dorigo, M., Maniezzo, V., & Colorni, A. (1996). Ant System: Optimization by a Colony of Cooperating Agents. *IEEE Transactions on Systems, Man, and Cybernetics*, 26(1), 29–41.

- Eberhart, R. C. and Kennedy, J. (1995). "A New Optimizer Using Particle Swarm Theory," in Symposium on Micro Machine and Human Science. Japan: Nagoya, Piscataway, NJ.
- Eberhart, R. C., & Shi, Y. (2001). *Particle swarm optimization: developments, applications and resources*. Proceedings of the 2001 Congress on Evolutionary Computation, Vol. 1, 81–86. <https://doi.org/10.1109/CEC.2001.934374>
- Glover, F. (1989). Tabu Search—Part I. *ORSA Journal on Computing*, 1(3), 190–206.
- Golden, B. L., Raghavan, S., & Wasil, E. A. (Eds.). (2008). *The Vehicle Routing Problem: Latest Advances and New Challenges*. Springer.
- Goss, S., Aron, S., Deneubourg, J.-L., & Pasteels, J.-M. (1989). Self-organized shortcuts in the Argentine ant. *Naturwissenschaften*, 76, 579–581.
- Grassé, P.-P. (1946). *Les insectes dans leur univers*. Paris, France: Éditions du Palais de la Découverte.
- Grassé, P.-P. (1959). La reconstruction du nid et les coordinations interindividuelles chez *Bellicositermes natalensis* et *Cubitermes* sp. La théorie de la stigmergie: Essai d'interprétation du comportement des termites constructeurs. *Insectes Sociaux*, 6, 41–81.
- Grefenstette, J. J. (1985). A Genetically Adaptive Heuristic for the Traveling Salesman Problem. *ORSA Journal on Computing*, 7(2), 125–133.
- Gutin, G., & Punnen, A. P. (2002). *The Traveling Salesman Problem and Its Variations*. Springer.
- Gutjahr, W.J. (2000). "A Graph-Based Ant System and its Convergence. Future Generation Computer Systems.
- Gutjahr, W.J. (2002). "ACO Algorithms with Guaranteed Convergence to the Optimal Solution." *Information Processing Letters*.
- Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press.
- Kennedy, J., & Eberhart, R. (1995). Particle Swarm Optimization. In *Proceedings of the IEEE International Conference on Neural Networks* (pp. 1942–1948).
- Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by Simulated Annealing. *Science*, 220(4598), 671–680.
- Lawler, E. L., Lenstra, J. K., Rinnooy Kan, A. H. G., & Shmoys, D. B. (1985). *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*. Wiley.
- Little, J. D. C., Murty, K. G., Sweeny, D., & Karel, C. (1963). An Algorithm for the Traveling Salesman Problem. *Journal of the ACM*, 10(4), 469–477.
- MacQueen, J. (1967). Some Methods for Classification and Analysis of Multivariate Observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability* (pp. 281–297).
- Menger, K. (1928). Über ein geometrisches Problem. *Mathematische Annalen*, 100, 75–86.
- Pala, O. ve Aksaraylı, M. (2018). Çok Amaçlı Kapasite Kısıtlı Araç Rotalama Problemi Çözümünde Bir Karınca Kolonisi Optimizasyon Algoritması Yaklaşımı. *Alfanümerik Dergi*, 6(1), 37-48. <https://doi.org/10.17093/alfphanumeric.366852>
- Pasteels, J.-M., Deneubourg, J.-L., & Goss, S. (1987). Self-organization mechanisms in ant societies (I): Trail recruitment to newly discovered food sources. *Experientia Supplementum*, 54, 155.

- Rego, C., Glover, F., & Laguna, M. (2004). A Tabu Search Algorithm for the Vehicle Routing Problem. *Journal of the Operational Research Society*, 55(2), 245–255.
- Reinelt, G. (1991). "TSPLIB—A Traveling Salesman Problem Library." *ORSA Journal on Computing*, 3(4), 376-384.
- Rosenkrantz, D. J., Stearns, R. E., & Lewis, P. M. (1977). An Analysis of Several Heuristics for the Traveling Salesman Problem. *SIAM Journal on Computing*, 6(3), 563–581.
- Shi, Y., & Eberhart, R. C. (1998). *A modified particle swarm optimizer*. Proceedings of the IEEE International Conference on Evolutionary Computation, 69–73. <https://doi.org/10.1109/ICEC.1998.699146>
- Söyler, H., & Keskindürk, T. (2007, Mayıs). *Karınca Kolonisi Algoritması ile Gezen Satıcı Probleminin Çözümü*. 8. Türkiye Ekonometri ve İstatistik Kongresi, İnönü Üniversitesi, Malatya, Türkiye.
- Toth, P., & Vigo, D. (Eds.). (2014). *Vehicle Routing: Problems, Methods, and Applications*. SIAM.
- Ward, J. H. (1963). Hierarchical Grouping to Optimize an Objective Function. *Journal of the American Statistical Association*, 58(301), 236–244.

ÖZGEÇMİŞ

Kişisel Bilgiler	
Adı Soyadı:	Burak AKSU
Doğum tarihi:	
Doğum Yeri:	
Uyruğu:	
Adres:	
E-mail:	
Eğitim	
Lisans:	Erzincan Binali Yıldırım Üniversitesi
Yüksek lisans:	Atatürk Üniversitesi
Yabancı Dil Bilgisi	
İngilizce:	Orta
Tezden Üretilmiş Yayınlar	
Burak Aksu, Murat Subaşı, ÇOKLU GEZGİN SATICI PROBLEMİNİN ÇÖZÜMÜNDE PSO TABANLI KÜMELEME VE KKO TABANLI ROTA OPTİMİZASYONU KULLANIMI, 16. Ankara Matematik Günleri (AMG 2025) - Çankaya Üniversitesi (19-20 Haziran 2025)	