



T.C

SAKARYA ÜNİVERSİTESİ

**BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

BSM 310 – YAPAY ZEKA

-KARINCA KOLONİSİ ALGORİTMASI-

Grup üyeleri:

**ADI SOYADI: Ahsen Ceyda ALTAŞ
NUMARASI: B181210102**

**ADI SOYADI: Burak ÇEVİK
NUMARASI: B171210034**

**ADI SOYADI: Gürkan KAYA
NUMARASI: G181210102**

**ADI SOYADI: Rüveyda HALICI
NUMARASI: B171210046**

**Sakarya
2021**

İÇİNDEKİLER

Kapak.....	1
İçindekiler.....	2
Genel Tanıtımı.....	3
Karınca Kolonisi Algoritmasının Amacı	3
Karıncaların Besin Arayışındaki Davranışları	3
Besin ve Feromon Miktarının İlişkisi	4
Sırt Çantası Problemi Nedir?	4
İşaretli İş Birliği Nedir?.....	4
Karıncaların Çevre Adaptasyonu	5
En Kısa Yolun Belirlenmesi	5
Optimum Karınca Sayısının Bulunması.....	6
Algoritmadaki Yapay Karıncaların Gerçek Karıncalarla Karşılaştırılması	6
Algoritmanın Avantajları ve Dezavantajları.....	6
Karınca Kolonisi Algoritması Akış Diyagramı	7
Karınca Kolonisi Algoritması Sözde Kodu.....	7
Geçiş Kuralı.....	8
Feromon Güncelleme	9
En İyi Çözümde Parametre Değerleri	9
Uygulama Alanları	10
Araç Rotalama Problemleri	10
Gezgin Satıcı Problemi.....	11-12
Gezgin Satıcı Probleminin Karınca Kolonisi Algoritması ile Çözümü.....	13
Çizelgeleme Problemleri	13
Atama Problemleri	13
Algoritmanın Simülatör Örneklemei	14-15
Kaynakça.....	16
Sorular	17-18

KARINCA KOLONİSİ ALGORİTMASI

Genel Tanıtımı:

Karınca kolonisi yaklaşımı literatürde ACO (Ant Colony Optimization) olarak genellikle kullanılır.

Sürü Zekası (swarm intelligence) veya sosyal zeka denen ve meta sezgisel yöntemlerin bir alt kümesini oluşturur.

1992’de Marco Dorigo’nun doktora tezinde önerilmiştir.

Karınca Kolonisi Algoritmasının Amacı:

Kombinasyonel hesaplar içeren problemlerde optimizasyon yapıp en uygun çözümü üretmeyi amaçlar. Bunu yaparken de gerçek karınca kolonilerinin davranışlarını baz alır.

Günümüze kadar Karınca Kolonisi Algoritmasının yeni modelleri ortaya çıkmış ve bu modellerin özellikle ayrık optimizasyon problemlerinin çözümüne uygulanması konusunda çeşitli çalışmalar yapılmıştır.

Karıncaların besin arama davranışlarından esinlenilmiştir. Bu davranış, karıncaların yuvalarını terk edip besin aramalarını ve daha sonra bu besini yuvalara taşımalarını içermektedir.

Karıncaların Besin Arayışındaki Davranışları:

Karıncalar yuvalarından çıkınca bir besine denk gelene kadar rastgele dolaşıp dururlar.

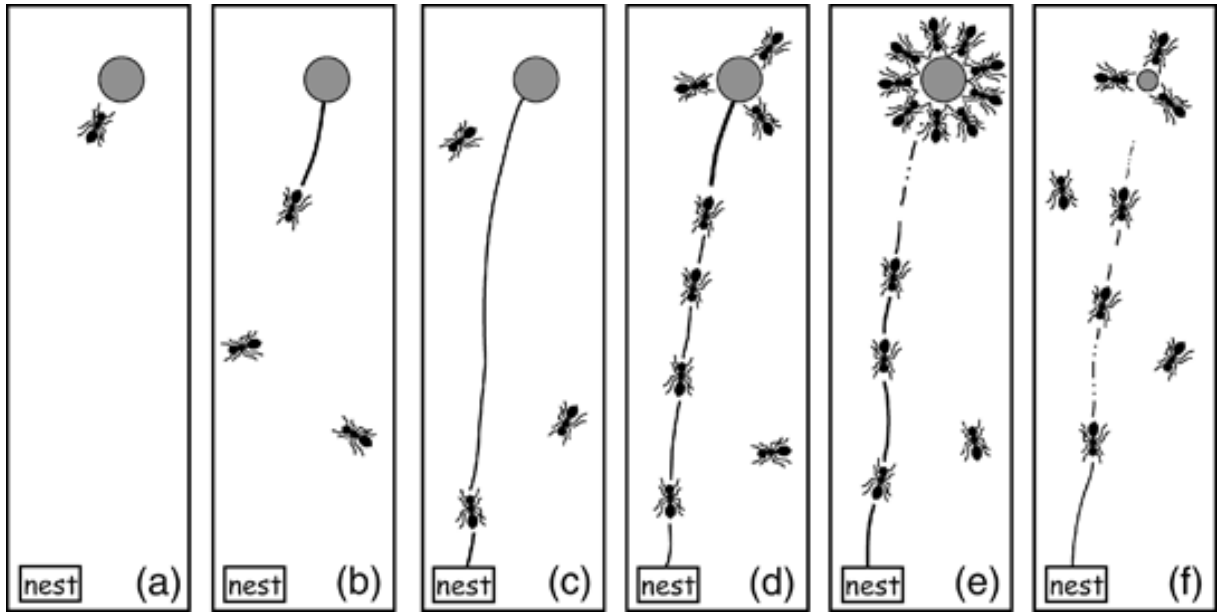
Besine denk gelen bir karınca , bu besin kaynağından bir miktarını alıp yuvaya taşıırken feromon denen bir kimyasal iz bırakır.

Bu izi fark eden diğer karıncalardan bir kısmı rastgele arayışlarını bırakıp bu izi takip etmeye başlarlar.

Feromon uçucu bir madde olduğundan bir süre sonra buharlaşmaya başlar ve bir süre sonra tamamen kaybolur.

Besinin yuvaya uzaklığı arttıkça buharlaşarak kaybolan feromon miktarı artar.

Besin ve Feromon Miktarının İlişkisi:



Besine gidip gelen karınca sayısı arttıkça, her biri feromon bıraktığından feromon miktarı hızla artar. Besin tükenince karıncaların besinle yuva arasındaki hareketleri yerini rastgele aramaya bırakır. Bu ise feromonun tamamen uçup gitmesi ile sonuçlanır. Bir karıncanın tespit ettiği besin miktarı çok ise, besin ve yuva arasındaki ize katılan karınca sayısı hızla artar. Bu artış feromon miktarını da artırır. Buna pozitif geribildirim denir. Bu şekilde karıncalar, bireysel bir zeka kullanmadan dolayı kolektif iletişim ile en kısa yol üzerinden besin toplama problemlerini çözebilmekteler. Kısacası karıncalar bir anlamda besin stoklarını sırt çantası problemi (knapsack problem) çözercesine doldururlar.

Sırt Çantası Problemi Nedir? :

Knapsack algoritması mevcut verilerden en optimum verimi alma üzerine kurulu bir algoritmadır. Bizim örneğimizde karıncalar stoklarını en değerli besinlerle optimum bir şekilde doldururlar.

İşaretili İş birliği Nedir? :

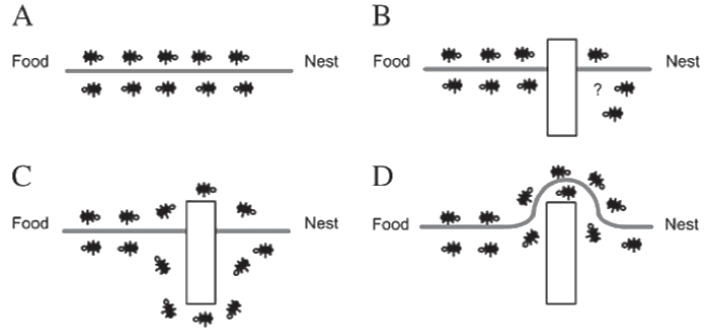
Karıncaların besin ve yuva arasındaki en kısa yolu ayırt etmek gibi bir özellikleri yok. (En azından şimdilik böyle olduğu sanılıyor) Ama kısa sürede besin ve yuva arasındaki en kısa yolu kullanmaya başlıyorlar. Bunun sebebi kısa yoldaki feromonların buharlaşıp uçma şansı bulamadan yenilerinin bırakılması ve artan feromonun daha fazla karıncayı cezbetmesidir.

Bu dolayı basit iş birliği mekanizmasına işaretili –iş birliği ‘‘ stigmergy ‘‘ denir. Burada hem feromon ile pozitif geribildirim hem de feromonun uçuculuğu sebebiyle negatif geri bildirim birlikte başarılmaktadır.

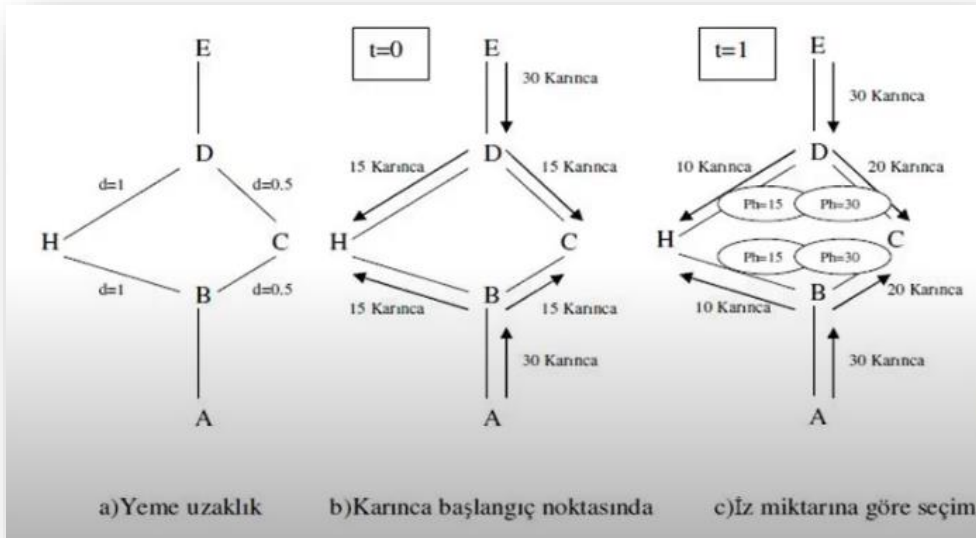
Karıncaların Çevre Adaptasyonu:

Karıncaların çevredeki değişime adapte olma yetenekleri vardır. Dış etkenler sonucu takip ettikleri mevcut yol artık en kısa yol değilse, yeni en kısa yolu bulabilmektedirler. Karıncaların en kısa yol bulmada kullandıkları bu yaklaşım, yol ve rota bulma problemlerine kolaylıkla

uygulanabilmektedir. Karınca Koloni Algoritmasına günlük hayatımızdan verebileceğimiz en iyi örneklerden bir tanesi ; lojistik firmalarının bize kısa zamanda siparişlerimizi iletmesi ve firmaya zaman, yakıt, çalışan eleman vs. tasarrufu sağlayabilmektedir.



En Kısa Yolun Belirlenmesi:



Özet olarak $t=0$ anında karıncalar uzun ve kısa yollara eşit sayıda dağılıyorken ; $t=1$ anında kısa yolu tercih eden karıncaların arttığı görülmektedir. Feromonun kısa yolda daha yoğun olmasına bağlı olarak iterasyon arttıkça kısa yoldaki karınca sayısı da artacaktır.

Optimum Karınca Sayısının Bulunması:

Karınca sayısının artırılması çözümde iyileşmeye neden olmaktadır. Fakat hesaplamaları arttırdığı için karınca sayısının fazla artırılması işlem zamanlarının uzamasına da neden olmaktadır.

Karınca sayısı, problem büyüklüğüne ve uygulama alanına bağlı olarak değişmektedir.

Algoritmadaki Yapay Karıncaların Gerçek Karıncalarla Karşılaştırılması:

Aynı Özellikleri :

- Karıncaların feromon aracılığı ile kurduğu iletişim,
- Feromon miktarının fazla olduğu yolların öncelikli seçilmesi,
- Kısa yollar üzerinde feromon miktarının daha hızlı artması

Ek Özellikleri :

- Zamanın ayrı olarak hesaplandığı bir ortamdadırlar,
- Tamamen kör olmayıp problem ile ilgili detaylara erişebilirler,
- Belli bir miktar hafıza ile, problemin çözümü için oluşturdukları bilgileri tutabilirler

Algoritmanın Avantajları ve Dezavantajları:

Avantajları :

- Pozitif geribildirim, iyi çözümlerin hızlı bir şekilde keşfedilmesini sağlar.
- Gezgin satıcı problemi ve benzeri problemler için verimli bir yöntemdir.
- Dinamik uygulamalarda kullanılabilir (yeni mesafeler gibi değişikliklere uyum sağlar)

Dezavantajları :

- Olasılık dağılımı her iterasyon için değişebilir.
- Yakınsama garantilidir, ancak yakınsama zamanı belirsizdir.

Karınca Kolonisi Algoritması Sözde Kodu:

Adım 1: Başlangıç feromon değerleri belirlenir.

Adım 2: Karıncalar her düğüme rastsal olarak yerleşir.

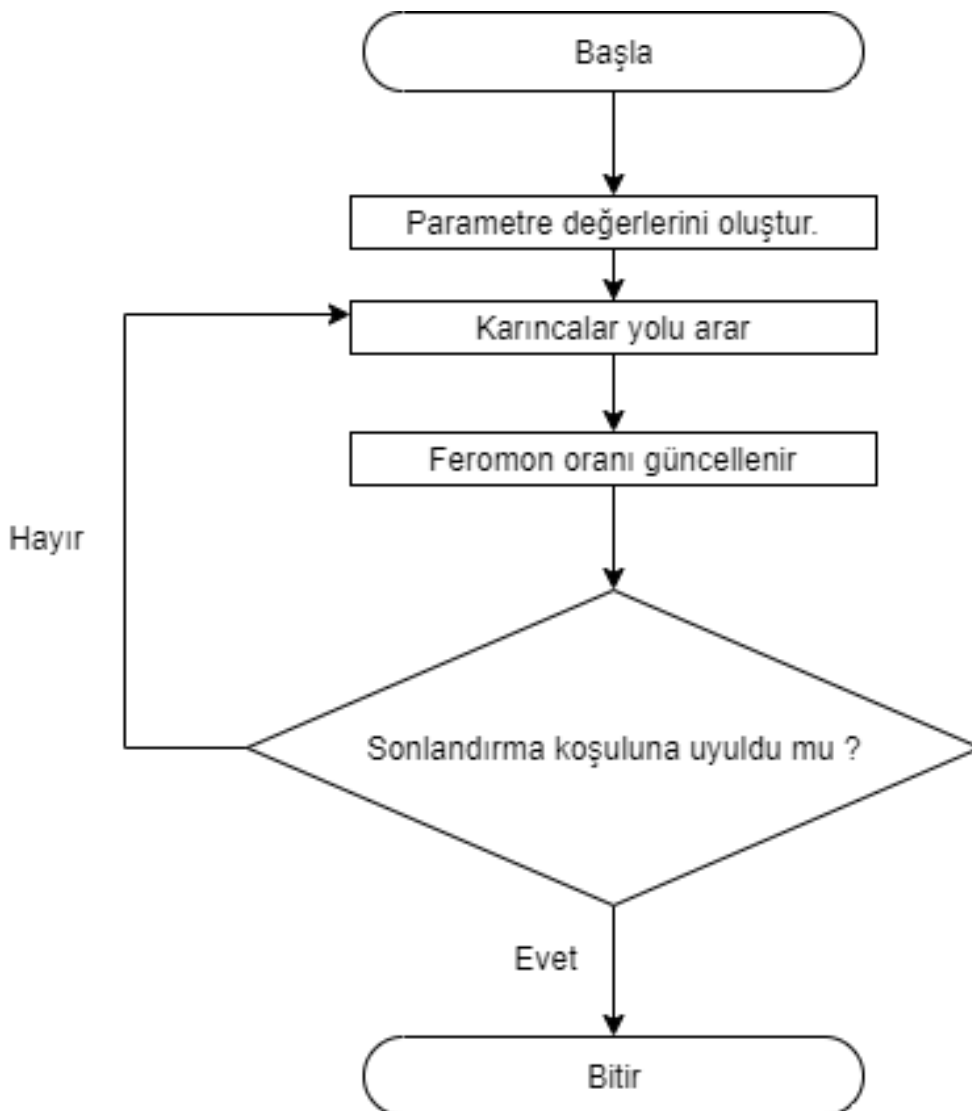
Adım 3: Her karınca, sonraki şehri denklemde verilen lokal arama olasılığına bağlı olarak seçmek suretiyle turunu tamamlar.

Adım 4: Her karınca tarafından katedilen yolların uzunluğunu hesaplanır ve lokal feromon güncellemesi yapılır.

Adım 5: En iyi çözüm hesaplanır ve global feromon yenilenmesinde kullanılır.

Adım 6: Maksimum iterasyon sayısı ya da yeterlilik kriteri sağlanana kadar Adım 2'ye gidilir.

Karınca Kolonisi Algoritması Akış Diyagramı:



Geçiş Kuralı:

Karınca kolonisi algoritmasında bir tur esnasında, i noktasında bulunan k karıncası için, sonraki j noktasını seçerken iki alternatif yol söz konusudur. İlk alternatif, gidebileceği yollar içerisinde feromon miktarlarına bağlı olarak hesaplanan seçim değerlerinden maksimum olanını seçmesidir. Genellikle bu yolla tercih yapma olasılığı (q_0) %90 olarak belirlenmektedir. İkinci alternatifte ise yollardaki feromon miktarı göz önüne alınarak oluşturulan olasılık dağılımına bağlı olarak yollar seçilir.

(q_0 = Karıncanın sezgisel bilgi ve öğrenilmiş feromon izlerinin gösterdiği sonuca göre ihtimal dahilindeki en iyi hareketini yapma olasılığı

q = 0-1 aralığında rastgele seçilen bir değişkendir.)

Aşağıda bu geçiş kuralına göre i noktasında bulunan k karıncasının u adet alternatif noktadan hangisine gideceğinin belirlendiği formül görülmektedir:

Eğer $q < q_0$;

$$j = \max_{u \in J_k(i)} \left\{ \left[\tau(i, u) \right]^\alpha \times \left[\eta(i, u) \right]^\beta \right\}$$

$\tau(i, u)$, (i,u) hattındaki feromon izidir.

$\eta(i, u) = 1/\delta(i, u)$, i noktasından u noktasına uzaklığın tersidir.

$J_k(i)$ i noktasındaki k karıncası tarafından henüz gidilmemiş şehirleri temsil etmektedir.

β ($\beta > 0$) feromonun güncellenmesinde, uzaklığın göreceli önemliliğini belirleyen parametredir.

Eğer $q > q_0$;

İkinci geçiş kuralı uygulanır. Bu kurala göre gidilecek bir sonraki nokta hesaplanan seçim değerlerine bağlı olarak rastlantısal olarak seçilmektedir. Dolayısıyla feromon miktarının daha yoğun olduğu yolların seçilme olasılığı daha fazla olacaktır. bir karıncanın i noktasından j noktasına gitmeyi seçme olasılığı :

$\tau(i, j)$, (i,j) yolundaki feromon miktarı

$\eta(i, j)$, (i,j) yolunun istenirliği

α : $\tau(i, j)$ ' nin etkisini kontrol parametresi

β : $\eta(i, j)$ ' nin etkisini kontrol parametresi

$$p_{ij} = \frac{(\tau_{ij}^\alpha)(\eta_{ij}^\beta)}{\sum (\tau_{ij}^\alpha)(\eta_{ij}^\beta)}$$

Feromon Güncelleme:

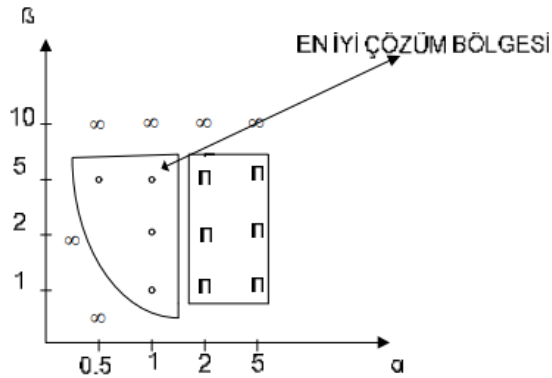
Tüm karıncalar turlarını tamamladıktan sonra feromon güncellemesi yapılır. Karıncaların her birinin toplam yol uzunlukları hesaplandıktan sonra en kısa yolu kullanan karınca bulunur.

- ρ : Feromonun buharlaşma oranı/iterasyon
- L_k : K'inci karıncanın tur (rota, yol) uzunluğu maliyeti
- τ_{ij} : (i,j) Yolundaki mevcut feromon miktarı
- $\Delta\tau_{ij}^k$: K'inci karıncanın (i,j) yoluna bıraktığı feromon miktarı
- $\Delta\tau_{ij}^k = \begin{cases} \frac{1}{L_k}; & \text{Eğer karınca } k, (i,j) \text{ yolundan geçtiyse} \\ 0; & \text{Aksi halde} \end{cases}$

Olarak tanımlandığında her iterasyon sonunda ayrıtlardaki feromonun miktarları bu formüle göre güncellenir :

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \Delta\tau_{ij}^k$$

En İyi Çözümde Parametre Değerleri:



α değeri, ilgili yolun feromon miktarının önemini belirlemektedir.

α değerinin yüksek olması feromonun yoğun olduğu yolların seçilme olasılığını arttırmaktadır.

β değeri ise yol uzunluklarının, bir sonraki noktanın seçimindeki etkisini belirlemektedir.

β değeri arttıkça tesadüfilik artmaktadır.

Aşağıdaki şekilde α ve β parametrelerinin aldığı çeşitli değerler karşısında çözümün nasıl etkilendiği gösterilmiştir

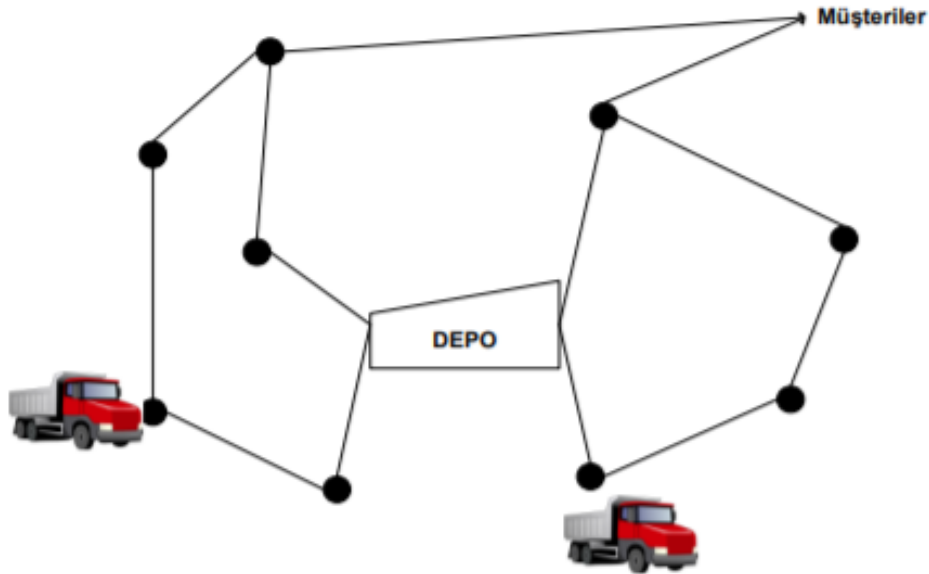
UYGULAMA ALANLARI:

Karınca Kolonisi Algoritması birçok uygulama alanında ve birçok problemin çözümünde kullanılmaktadır.

Araç Rotalama Problemleri:

Bir veya birkaç depodan müşterilerine hizmet verecek araçlar için en uygun rotaları belirlemeyi amaçlayan optimizasyon problemidir. Araç rotalama problemleri, gezgin satıcı probleminin birden fazla araç ve kısıtlar eklenerek genişletilmiş halidir. Bu yüzden çözümü gezgin satıcı problemine göre daha zordur. Araç rotalama probleminin bazı gerçek hayat uygulamaları ise okul servisleri, yakıt, gazete ve posta dağıtımı, perakende ürün dağıtımı, çöp toplanması gibi uygulamalardır. Problemin birçok çeşidi vardır.

Problemin çözümü için birçok algoritma vardır. 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 birisi de karınca kolonisi optimizasyonudur.



Araç Rotalama Problemi

Gezgin Satıcı Problemi:

Karınca Kolonisi Optimizasyonu Algoritmasının ilk uygulama alanlarından biri Gezgin Satıcı Problemidir. Amaç; en kısa turu bulmaktır. Bunun için bir başlangıç karınca kümesi oluşturulur. Her karınca kendine ait bir tur oluşturur. Her şehir tam olarak bir kez ziyaret edilebilir. Bir şehre erişim uzaklığı arttıkça seçilme olasılığı azalır. İki şehri birbirine bağlayan yol üzerindeki feromon yoğunlaştıkça o bağlantı yolunun seçilme olasılığı artar. Karınca rotayı tamamladığında yani başlangıç şehrine döndüğünde kat ettiği yol veya harcadığı zaman daha kısa ise bu yolun üzerindeki ayrıtlar daha çok feromon biriktirir. Her iterasyonda bir miktar feromon buharlaşarak kaybolur

Gezgin Satıcı Probleminin Karınca Kolonisi Algoritması ile Çözümü:

Problemimiz, Türkiye'nin herhangi bir ilinden yola çıkılarak ve tüm şehirler ziyaret edilerek tekrar başlangıçtaki noktaya dönmemizi ele almaktadır. Gerçekleştirilecek böyle bir Türkiye turunun, en kısa şekilde nasıl yapılabileceğini karınca kolonisi algoritması yardımıyla inceleyelim.

Öncelikle 81 ilimize ait mesafeler tablosu oluşturulur. Tüm illerimizin birbirlerine olan karayolu uzaklığı kilometre cinsinden tabloya yazılır.

	Adana	Adıyaman	Afyon	Ağrı	Amasya	Ankara
Adana	0	330	573	966	612	490
Adıyaman	330	0	903	648	636	757
Afyon	573	903	0	1314	593	257
Ağrı	966	648	1314	0	734	1057
Amasya	612	636	593	734	0	336
Ankara	490	757	257	1057	336	0

Mesafeler Tablosunun Bir Parçası

Problemin çözümünde kullanılacak parametreler ise şu şekilde belirlenmiştir.

Parametre	Değer
Global güncelleme paydası	100
Buharlaşma oranı	0,1
q değeri	15
Karınca sayısı	81
İterasyon sayısı	1000
Alfa	0,80
Beta	4

Parametre Değerleri

Daha sonra parametreler kullanılarak aşağıdaki algoritma adımları 1000 iterasyon için işletilir ve sonuç hesaplanır.

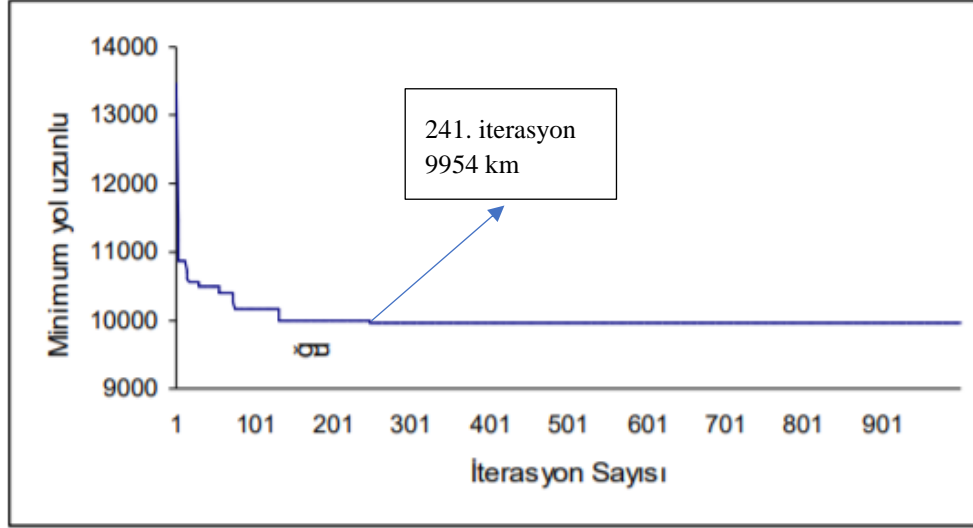
<i>Başlangıç</i>
<i>Her karınca için rassal bir başlangıç şehri belirle,</i>
<i>Her tur için;</i>
<i>Her karınca için;</i>
<i>Bir sonraki düğüm noktasını geçiş kurallarını kullanarak belirle;</i>
<i>Tur uzunluğunu belirle</i>
<i>Feromon güncelleme bağıntıları kullanarak feromon güncelleştirmesi yap</i>
<i>Dur.</i>

Bu algoritma 1000 iterasyon işletildiğinde optimum sonuç 241'inci iterasyonda 9954 km olarak bulunur. En kısa Türkiye turu sonuçları şu şekildedir.

Çıkış noktası	Varış noktası	Çıkış noktası	Varış noktası	Çıkış noktası	Varış noktası	Çıkış noktası	Varış noktası
Çorum	- Yozgat	Edirne	- Kırklareli	Rize	- Artvin	Batman	- Diyarbakır
Yozgat	- Kırşehir	Kırklareli	- Tekirdağ	Artvin	- Ardahan	Diyarbakır	- Mardin
Kırşehir	- Kırıkkale	Tekirdağ	- İstanbul	Ardahan	- Kars	Mardin	- Şanlıurfa
Kırıkkale	- Çankırı	İstanbul	- Kocaeli	Kars	- Iğdır	Şanlıurfa	- Adıyaman
Çankırı	- Ankara	Kocaeli	- Yalova	Iğdır	- Ağrı	Adıyaman	- K.Maraş
Ankara	- Eskişehir	Yalova	- Bursa	Ağrı	- Erzurum	K.Maraş	- Gaziantep
Eskişehir	- Kütahya	Bursa	- Bilecik	Erzurum	- Bayburt	Gaziantep	- Kilis
Kütahya	- Afyon	Bilecik	- Sakarya	Bayburt	- Gümüşhane	Kilis	- Hatay
Afyon	- Uşak	Sakarya	- Bolu	Gümüşhane	- Erzincan	Hatay	- Osmaniye
Uşak	- Isparta	Bolu	- Düzce	Erzincan	- Tunceli	Osmaniye	- Adana
Isparta	- Burdur	Düzce	- Zonguldak	Tunceli	- Malatya	Adana	- İçel
Burdur	- Antalya	Zonguldak	- Bartın	Malatya	- Elazığ	İçel	- Karaman
Antalya	- Denizli	Bartın	- Karabük	Elazığ	- Bingöl	Karaman	- Konya
Denizli	- Muğla	Karabük	- Kastamonu	Bingöl	- Muş	Konya	- Aksaray
Muğla	- Aydın	Kastamonu	- Sinop	Muş	- Bitlis	Aksaray	- Niğde
Aydın	- İzmir	Sinop	- Samsun	Bitlis	- Van	Niğde	- Nevşehir
İzmir	- Manisa	Samsun	- Ordu	Van	- Hakkari	Nevşehir	- Kayseri
Manisa	- Balıkesir	Ordu	- Giresun	Hakkari	- Şırnak	Kayseri	- Sivas
Balıkesir	- Çanakkale	Giresun	- Trabzon	Şırnak	- Siirt	Sivas	- Tokat
Çanakkale	- Edirne	Trabzon	- Rize	Siirt	- Batman	Tokat	- Amasya
						Amasya	- Çorum

İllere göre en kısa Türkiye turu

Her bir iterasyona ait en kısa yolları gösteren grafik aşağıdaki gibidir. Grafikte görüldüğü gibi karınca kolonisi optimizasyonu kullanılan problemde ilk iterasyonlarda çözüm değerinde çok yüksek oranda iyileşme mevcuttur. Daha sonra bu iyileşme oranı azalmaktadır.



İterasyonlar boyunca minimum yol uzunluğu

Çizelgeleme Problemleri:

Çizelgeleme, görevler ve kaynaklar arasında bir atama, sıralama ve zamanlama problemidir. Bu problem hizmet sistemlerinde görev ve nöbet çizelgeleme, ders programı, sınav programı çizelgeleme, tren ve uçak seferleri çizelgeleme gibi konularda ortaya çıkar. İmalat sistemlerinde ise genelde işlerin makinelere atanması şeklinde ortaya çıkan bir problemidir. Karınca kolonisi algoritması bu problemlerin çözümünde kullanılabilir.

Atama Problemleri:

Bu problemde amaç türlü kaynakların değişik görevlere en uygun dağıtımını yaparak toplam karı maksimum veya toplam maliyeti minimum kılmaktır. Karınca kolonisi algoritması kuadratik atama probleminde kullanılmaktadır.

Diğer Örnekler :

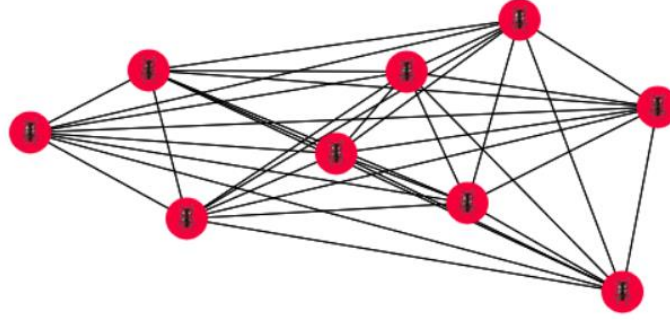
Kümeleme Problemleri

Sınıflandırma ve Kümeleme Problemleri

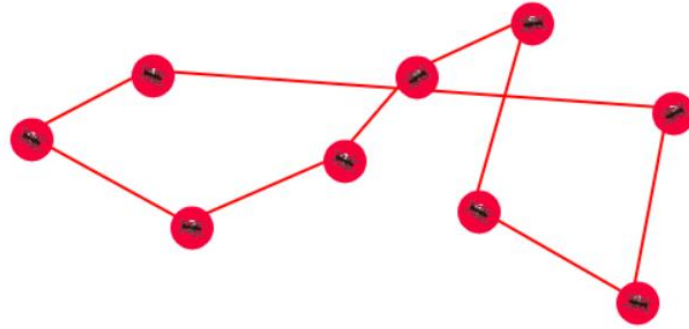
Simülasyon ile Karınca Kolonisi Algoritması Örneği:

<http://thiagodnf.github.io/aco-simulator/#> sitesi kullanılarak karınca kolonisi algoritmasını simüle edebiliriz. Karıncaların turlarda bıraktığı feromon miktarlarına göre siyah yollar belirginleşmektedir.

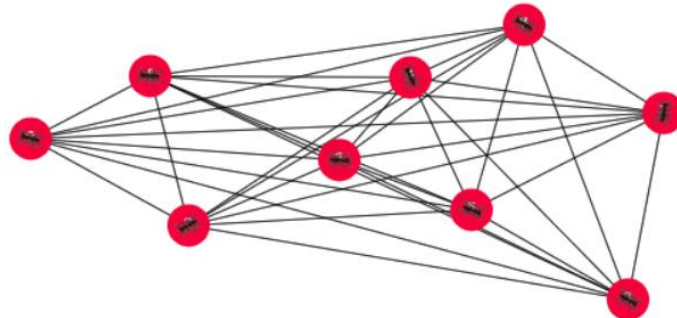
Örnek olarak 9 düğümlü bir problem oluşturalım.



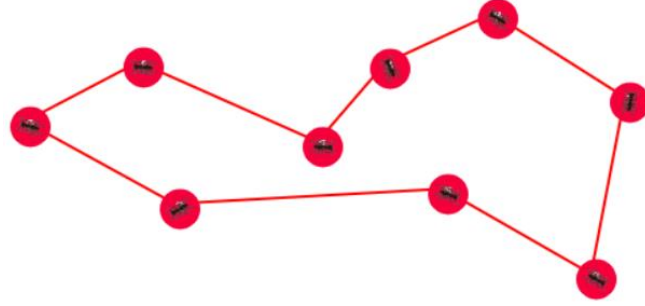
Parametrelerimizi $\alpha=1$, $\beta=2$ ve $\rho=0.1$ (buharlaştırma oranı/iterasyon) olacak şekilde seçelim. Karınca sayısı düğüm sayısı ile aynıdır yani 9'dur. Simülasyonu adım adım inceleyelim.



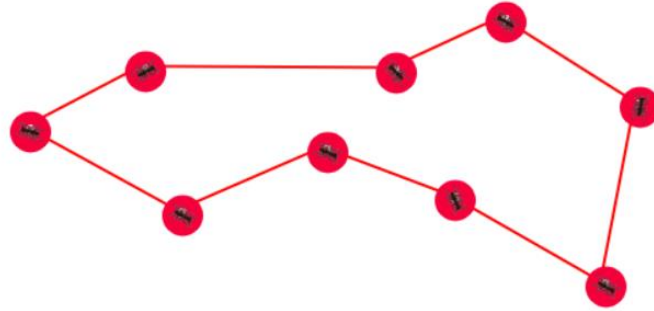
İlk tur sonucunda en iyi çözüm 1258 çıkmıştır. Fakat bu optimal çözüm değildir.



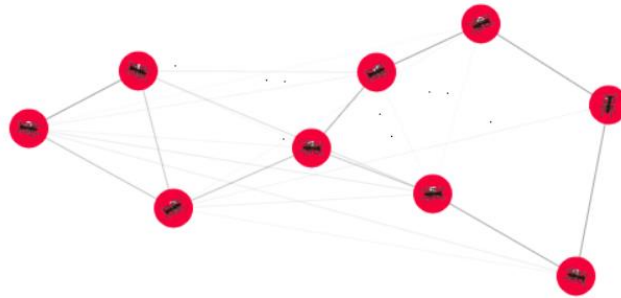
Siyah çizgiler gidilebilecek tüm yolları göstermektedir. Çizginin belirgin hale gelmesi feromon miktarının çok olduğu anlamına gelmektedir. 5. İterasyondan bir görüntü olduğundan dolayı fark belli olmamaktadır. Simülasyon ilerlediğinde siyah çizgiler en iyi çözüme yakın şekilde belirginleşecektir.



Simülasyon 5. tur sonunda en iyi çözüm olarak 1107 değerini bulmuştur. Bunun en iyi çözüm olup olmadığını anlamak için iterasyon sayısını arttıralım.



53. iterasyonda 1083 sonucu bulunmuştur.



1000'in üzerinde iterasyon sonucu çözümün 53. İterasyonda bulunan 1083 olduğu anlaşılmıştır. Görüldüğü üzere yollar üzerindeki feromon miktarlarına göre yollar daha belirgin ya da daha silik hale gelmiştir.

➤ Kaynakça

https://en.wikipedia.org/wiki/Ant_colony_optimization_algorithms

[M. Dorigo, V. Maniezzo & A. Coloni, 1996. "Ant System: Optimization by a Colony of Cooperating Agents", IEEE Transactions on Systems, Man, and Cybernetics–Part B, 26 \(1\): 29–41.](#)

[M. Dorigo, 2007. "Ant Colony Optimization"](#)

[Arzu EREN ŞENARAS, Şahin İNANÇ "GSP ÇÖZÜMÜ İÇİN KARINCA KOLONİSİ OPTİMİZASYONU"](#)

[Hasan SÖYLER, Timur KESKİNTÜRK "KARINCA KOLONİSİ ALGORİTMASI İLE GEZEN SATICI PROBLEMİNİN ÇÖZÜMÜ"](#)

[Gökhan DALKILIÇ "Karıncı Kolonisi Optimizasyonu"](#)

[8. Türkiye Ekonometri ve İstatistik Kongresi, Malatya, Türkiye, 1 - 04 Mayıs 2007, ss.1-11](#)

Sorular :

1) Aşağıdakilerden hangisi karınca kolonisi algoritması optimizasyonunda kullanılan yapay karıncaları gerçek karıncalardan ayıran özelliklerden birisidir?

- a) Feromon aracılığıyla iletişim kurmaları
- b) Feromon miktarı fazla olan yola yönelim
- c) Kısa yolda feromon miktarının daha fazla artması
- d) Tamamen kör olmaları
- e) Belirli bir miktar hafıza ile problemin çözümü için bilgi tutulması

Cevap E

2) Karıncalar arasındaki sosyal ilişkileri düzenleyen, karıncaların besin arama, yol bulma gibi problemlerini çözebilmelerini sağlayan kimyasal iz nedir?

- A) Biyom
- B) Feromon
- C) Biyosfer
- D) Popülasyon
- E) Habitat

Cevap: B

I) Besin Değeri

II) Besin Boyutu

III) Yol Uzunluğu

3) Yukarıdaki özelliklerden hangisi ya da hangileri bırakılan feromon miktarı ile doğru orantılıdır?

- A) Yalnız I
- B) Yalnız II
- C) II ve III
- D) I ve II
- E) I, II ve III

Cevap D

I) Çizelgeleme Problemleri

II) Araç Rotalama Problemleri

III) Atama Problemleri

4) Yukarıdaki problemlerden hangilerinin çözümünde Karınca Kolonisi Algoritmasından yararlanılabilir?

A) Yalnız I

C) I , II

B) Yalnız II

D) II , III

E) I , II , III

Cevap:E

I)Pozitif geribildirim, iyi çözümlerin hızlı bir şekilde keşfedilmesini sağlar.

II)Yakınsama zamanı belirsizdir.

III)Dinamik uygulamalarda kullanılabilir (yeni mesafeler gibi değişikliklere uyum sağlar)

5) Yukarıdaki özelliklerinden hangileri karınca kolonisi algoritmasının avantajlarındandır?

A) Yalnız I

B) Yalnız II

C) Yalnız III

D) I , III

E) I , II , III

Cevap: D