

T.C.
ESKİŞEHİR OSMANGAZİ ÜNİVERSİTESİ
Fen Bilimleri Enstitüsü



Bilgisayar Mühendisliği Ana Bilim Dalı

Algoritma Tasarımı ve Analizi Dersi

Karınca Kolonisi Optimizasyon Algoritması

(Ant Colony Optimization Algorithm)

Arif ALTIOK

Aralık 2021

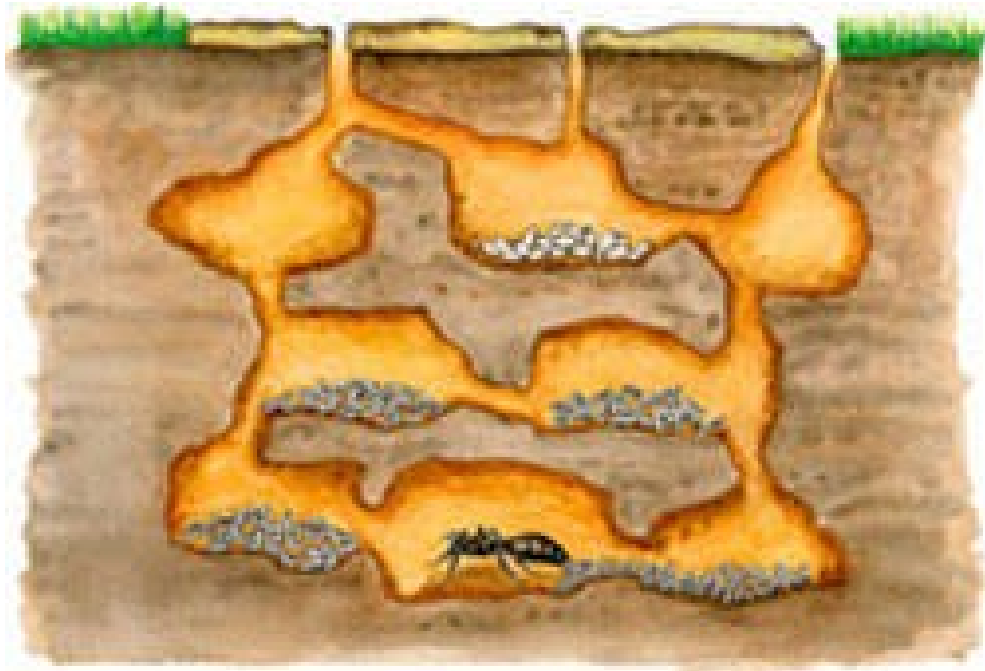
1. Giriş

Karınca Kolonisi Optimizasyon Algoritması (Ant Colony Optimization Algorithm) Marco Dorigo tarafından 1992 yılında doktora tezi olarak tanıtıldı. Algoritmanın ana fikri karıncaların yiyecek arama davranışının yapay olarak simüle edilmesi üzerinedir. Bu fikir ile Gezgin Satıcı Problemi ve graf tabanlı diğer problemlere farklı bir çözüm sunmaktadır. Yaklaşık 30 senelik bir geçmişi olmasına rağmen henüz geçtiğimiz on yılda popülerliğe ulaşmış bir algoritmadır.

2. Karıncalar ve Koloniler

Karınca, Karıncalar (Formicidae) familyasını oluşturan, yaban arıları ve arılarla birlikte zar kanatlılar (Hymenoptera) takımında yer alan sosyal yaşam gösteren böceklere verilen ortak bir addır. Günümüzde yaklaşık 14.000 türü olduğu tahmin edilmektedir. Dünya üzerinde hemen hemen her yerde bulunan karıncalar ekosistemin yaklaşık %15 – 25' sini oluştururlar. [1]

Karıncalar koloni şeklinde yaşamlarını sürdürmektedirler. Bir karınca kolonisi yaklaşık 300.000 ila 500.000 adet karıncadan oluşur.[2] Koloni de kraliçe, erkek, işçi ve asker karıncalar bulunur.[3] Kraliçe karıncanın türüne göre 7 ila 90 yıl arası yaşam ömrü vardır. Bir kolonide birden fazla Kraliçe karınca bulunabilir.

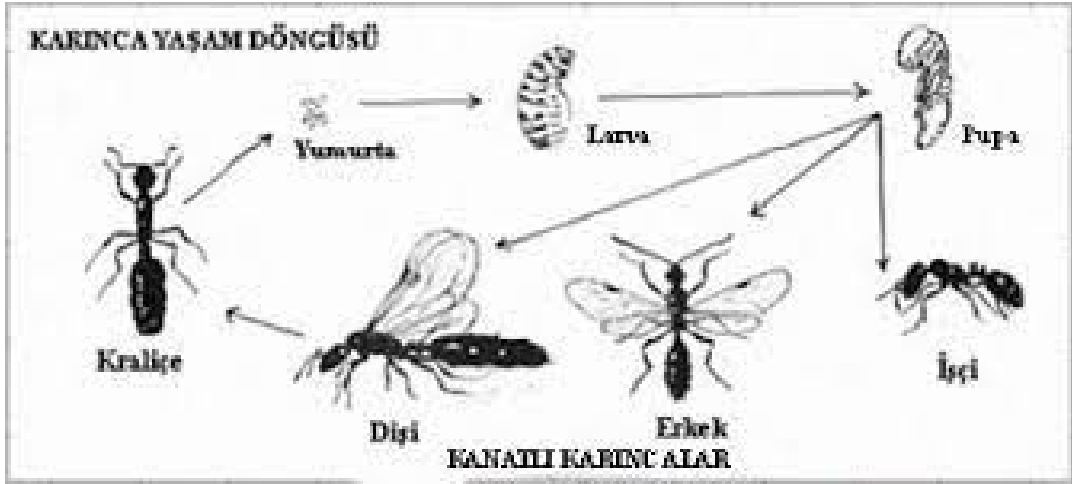


Şekil 1 Karınca Kolonisi

2.1 Karınca Yaşam Döngüsü

Kraliçe karınca yumurtlama döneminde erkek karıncalar ile eşleşir ve kolonide yumurtaları güvenli odacıklara bırakır. Bu yumurtalar yaklaşık 1 milimetrenin yarısı büyüklüğünde ve beyaz renktedir. Yumurtadan çıkan yavrular tırtıla benzerler ve “Larva” olarak adlandırılır. Larvalar tüy döküp deri değiştirerek “Pupa” adı verilen döneme geçer. Pupalar yetişkinlere benzemekle birlikte antenleri katlı ve beyaz renktedir. Pupalar olgunlaştıkça antenleri düzelir ve rengi koyulaşarak “Yetişkin” karınca halini alır. Bu süreç

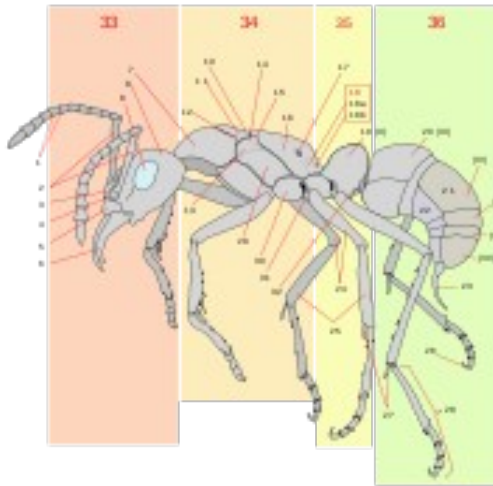
karıncaların türlerine ve yerlerine göre 2 hafta ile 2 ay arasında sürer. Eşleşme döneminde döllenen yumurtalar Kraliçe karınca, döllenmeyen yumurtalar işçi karınca olarak doğarlar. [2,3]



Şekil 2 Karınca Yaşam Döngüsü

2.2 Karıncaların İletişimleri

Karıncaların birçok eklem ve duyargaları bulunur. Birbirine bağlı sayısız lenslerden oluşan “petek göz”leri vardır. Petek gözler hızlı hareketleri tespit edebilseler de optik çözünürlüğü düşüktür. Bu sebeple türlerine göre değişmekle birlikte, karıncaların büyük bir çoğunluğunda görme yeteneği kısıtlıdır. Kafalarının üzerinde iki adet “anten”i bulunur. Bu antenler kimyasalları, hava akımlarını ve titreşimleri algılar. [1].

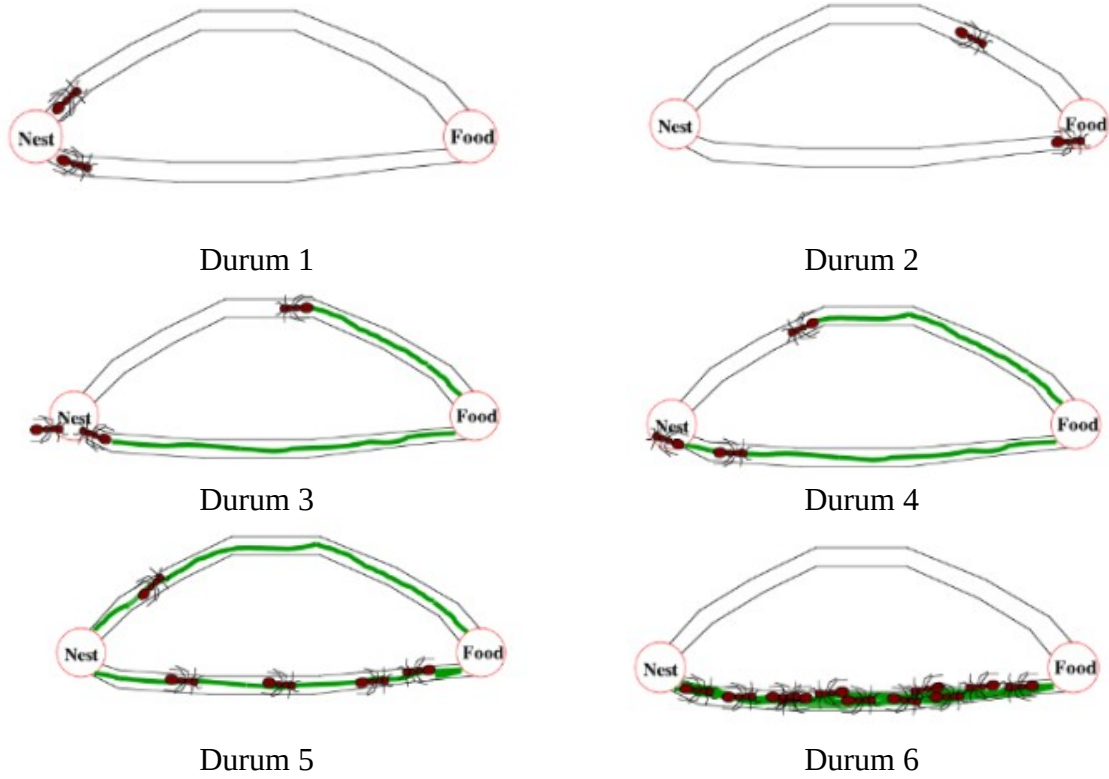


Şekil 3 Karıncanın Eklem ve Duyargaları

Koloni halinde yaşayan ancak kolonisini terk ettiğinde yiyecek aramak için rastgele dolaşan karınca diğer karıncalara bilgi bırakabilmek amacıyla geçtiği yollara “feromon” ismi verilen kimyasal salgıyı bırakır. Diğer karıncalar antenleri vasıtasıyla feromon salgısını algılar ve rastgele dolaşmak üzerine feromon çokluğu olan bölgede yiyecek olduğunu varsayarak bu yola yönelir. Böylelikle aynı zaman diliminde farklı konumda olan karıncalar birbirleriyle gizli bir iletişim kurmuş olur.

2.3 Karıncaların Yiyecek Arama Davranışı

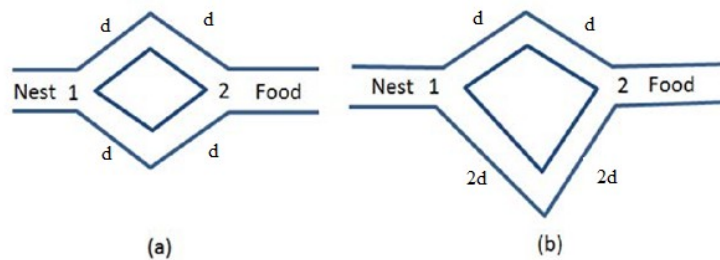
Gerçek hayatta karıncalar yiyecek aramak için yuvasını terk ettikten sonra rastgele olarak dolaşmaya başlar. Yiyecek bulduğunda ise geri döndüğü yollara feromon ismi verilen kimyasal salgıyı bırakır. Bu diğer karıncalarla dolaylı yoldan iletişim kurmak ve onlara yol göstermek amaçlıdır. Aynı yolda ne kadar çok feromon varsa yiyecek arayan diğer bir karınca rastgele dolaşmak yerine feromon yoğunluğu olan yola gitme eğilimi gösterir. Karıncalar ilk başlarda rastgele hareketler sergilese de bir müddet sonra neredeyse tümü aynı yolu kullanmaya başlar.



Şekil 4 Karıncaların Yiyecek Arama Davranışlarının Durumları

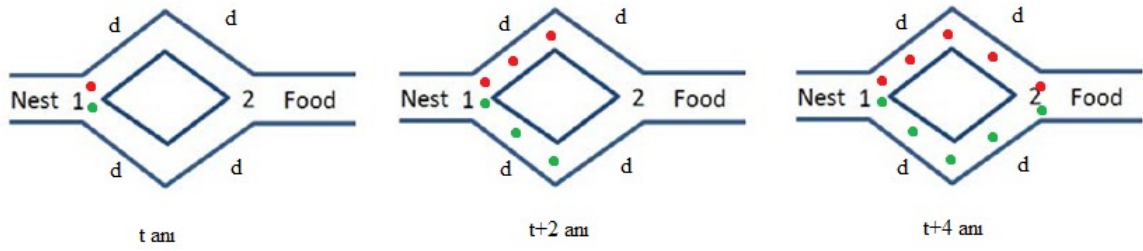
Yuvadan aynı yiyeceğe giden neredeyse sonsuz sayıda yol bulunur. Her karınca izlediği yola feromon bırakması, ortak kullanılan yollardaki feromon oranını artırır. Herhangi bir anda kavşağa gelen bir karınca hangi yola gireceğini, sadece bu kavşağa bağlı yollar üzerindeki feromon oranlarını değerlendirip en yüksek olanını tercih eder. Diğer kullanılmayan yolların feromon oranları zaman içinde buharlaşmaya uğrayarak azalır ve böylelikle karıncalar ortak yolu kullanmaya başlar. Bu genellikle yuvadan yiyeceğe giden en kısa yol olur.

2.3.1 Nasıl En Kısa Yol Seçilir?



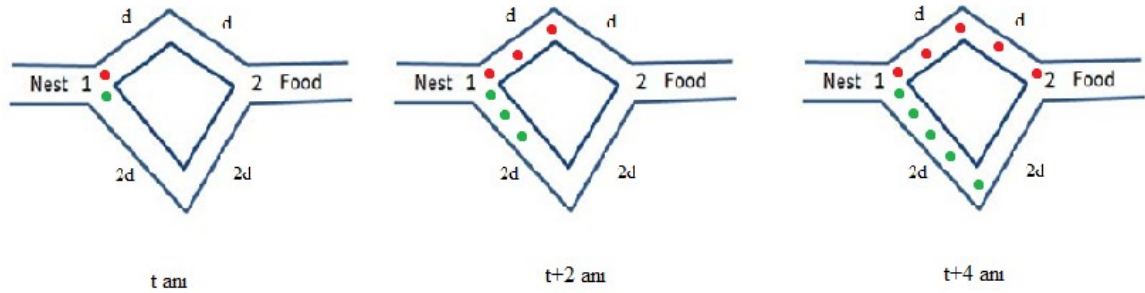
Şekil 5 Örnek Haritalar

Bu detayı açıklamak için yukarıdaki şekildeki gibi a ve b haritaları olduğu, yolların üzerlerinde mesafelerin d cinsinden yazıldığını düşünelim. Şekilden görülebileceği gibi, b haritasındaki aşağıya giden yol üst yolun toplamda 2 kat uzunluğundadır.



Şekil 6 A-Haritasındaki Muhtemel Karınca Hareketleri

Kırmızı ve Yeşil noktaları birbirinden farklı, ancak eşit özelliklere sahip iki karınca olarak ve her t anında bir birim hareket ettiğini kabul edelim. t anında karıncalar yuva yakınında bir kavşaktadır. Burada kırmızı karınca üst yolu, yeşil karınca alt yolu tercih ediyor. t+2 anında karıncalar 2 birim hareket ederek d kadar yol gitmişlerdir. t+4 anında yine iki birim hareket ederek yiyeceğe giden kavşakta buluşurlar ve yolu tamamlayarak yiyeceğe ulaşırlar. Burada yiyeceği alan karıncalar tekrar geldiği yolu geri döndüğünde t+8 anında yuvaya dönmüş olurlar. Aslında kullanılan yol mesafeleri aynı olduğundan alt ve üst yollarda bırakılan feromon miktarı eşit olacaktır.



Şekil 7 B-Haritasındaki Muhtemel Karınca Hareketleri

B haritasında t anında kavşakta bulunan karıncalar yine aynı şekilde kırmızı karınca üst yolu, yeşil karınca alt yolu tercih ettiğini düşünelim. t+2 anında karıncalar 2 birim hareket ederek kırmızı karınca gideceği yolun yarısını, yeşil karınca ise gideceği yolun çeyreğini bitirmiştir. t+4 anında ise kırmızı karınca gideceği yolu tamamlayarak yiyeceğe ulaşmış, ancak yeşil karınca halen gideceği yolun yarısını yani 2d lik kısmını bitirebilmiştir. (Aslında bu aşamaya kadar her iki karıncanın da kat ettiği yol mesafesi aynıdır, fakat yolların yiyeceğe olan uzaklıkları farklıdır.) Yiyeceğe ulaşan kırmızı karınca geldiği yolu geri dönerek feromon bırakmaya başlar ve yuvaya t+8 anında döner. t+8 anında yeşil karınca yiyeceğe henüz ulaşmıştır ve geri dönüş eylemine geçemediğinden alt yola henüz feromon bırakamamıştır. Bu aşamada diğer karıncalar yiyecek arama eylemine geçtiğinde üst yolda feromon olduğunu, ancak alt yolda henüz olmadığını görerek, üst yolu tercih edip aynı işlemleri tekrarlayacaktır. Alt yol zamanla hiç tercih edilmeyeceğinden yeşil karıncanın bu yol üzerinde bıraktığı feromonlar buharlaşarak yok olacak ve en kısa olan üst yol koloni tarafından kullanılan yol olacaktır.

3. Karınca Kolonisi Optimizasyon Algoritması (ACO)

Karınca Kolonisi Optimizasyon Algoritması (Ant Colony Optimization Algorithm), 1992 senesinde Marco Dorigo tarafından doktora tezi olarak tanıtıldı. Karıncaların doğal süreç ve ortamda yiyecek arama davranışını simüle eden algoritma, sürü zekasının bir uygulamasıdır. Algoritmada, karıncaların tekil olarak yiyecek arama davranışı bir anlam ifade etmez. Ancak kolonisiyle birlikte değerlendirildiğinde, feromonlar vasıtasıyla karıncaların birbirleri arasında kurduğu iletişim sürü zekasını ön plana çıkartır.

Yapay bir karınca gerçek bir karıncanın aksine hafızaya sahiptir. Bu hafıza yiyecek arayan bir karıncanın daha önceden ziyaret ettiği noktaları tekrar ziyaret etmemesi ve yiyeceği bulup yuvaya geri dönerken geçtiği yollara feromon bırakması için geri izleme amacıyla kullanılır.

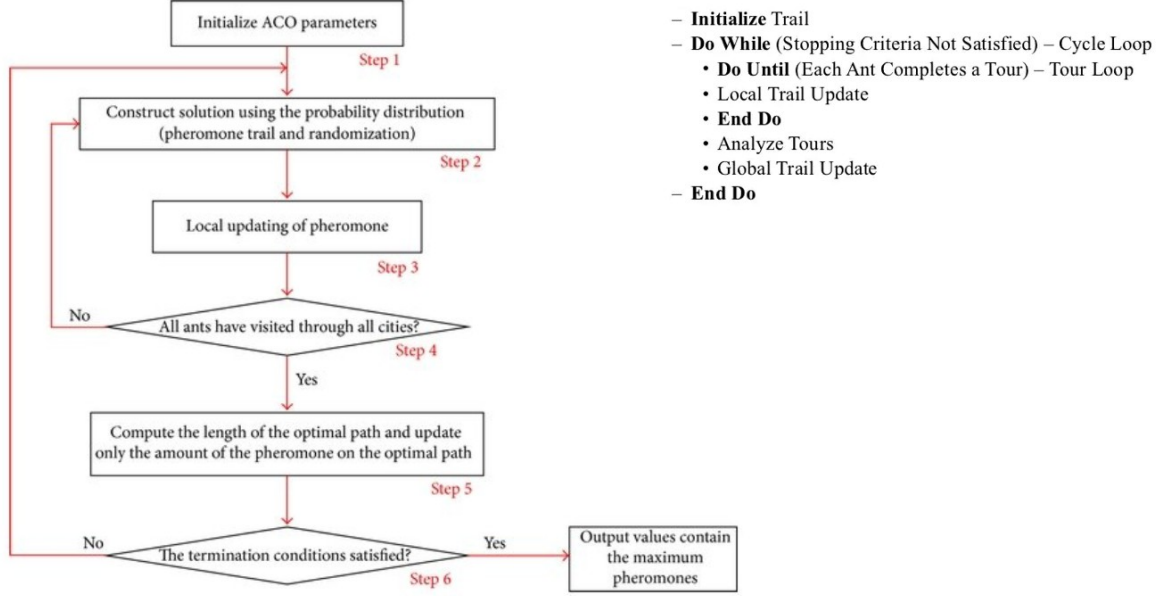
Yapay bir dünyada yapay karıncalar eş zamanlı olarak her bir t anında bir birim hareket eder. İterasyon başlangıcında karıncalar yuva konumundan bırakılır. Karıncaların yanlış yerde yiyecek aramalarını önlemek adına hareket sayısı kısıtlanır. İterasyon içerisinde yiyecek bulan karıncalar geri izleme ile feromon bırakarak geri döner. Bu geri izleme adımları da hareket sayısına dahildir. İterasyon bitiminde tüm yollardaki feromon miktarları buharlaşmaya tabi tutulur ve işlemler tekrarlanır.

Tüm karıncalar birbirlerinden bağımsızdır ve bireysel kararlarını alırlar. Bir yol üzerinde bulunan karınca kavşağa geldiğinde gidebileceği yollar üzerindeki feromon oranlarını olasılıksal olarak değerlendirerek yüksek olanı tercih eder. Tercih ettiği yolda yiyecek bulması, geri izleme adımlarında bu yola feromon bırakması anlamına gelir. Böylelikle yolun tercih edilme değeri artar ve aynı kavşağa gelen diğer bir karıncanın karar verme sürecinde etkili olur.

Algoritma önerildiği 90'lı yıllarda Gezgin Satıcı Problemi (TSP) problemine bir çözüm üretmek amacıyla tanıtılsa da günümüzde araç rotalama, ağ rotalama, görüntü bölütleme gibi pek çok optimizasyon içeren alanda kullanılmaktadır.

3.1 Algoritmanın Akış Diyagramı ve Sözde Kodu

Gezgin Satıcı Problemi, bir satıcının n adet şehri, her şehri en fazla bir kere ziyaret ederek tüm şehirleri en kısa yoldan turlayıp başladığı konuma geri dönmesi problemidir. Aşağıdaki akış diyagramı ve sözde kod Gezgin Satıcı Problemini çözen Karınca Kolonisi Algoritmasıdır.



Şekil 8 Algoritmanın Akış Diyagramı ve Sözde Kodu

3.2 Algoritmadaki Semboller ve Hesaplamalar

3.2.1 Semboller

t	feromon matrisi,
$t_{i,j}$	i noktasından j noktasına giden yoldaki toplam feromon miktarı,
$\Delta t_{i,j}^k$	k . karıncanın $t_{i,j}$ yoluna bıraktığı feromon miktarı,
L_k	k . karıncanın seçtiği yolun toplam uzunluğu,
$p_{i,j}$	bir karıncanın i noktasından j noktasına geçiş olasılığı,
e ,	buharlaşma katsayısı (0-1 aralığında),

olmak üzere hesaplama detayları alt bölümde verilmiştir.

3.2.2 Hesaplamalar

$$\Delta t_{i,j}^k = \begin{cases} \frac{Q}{L_k}, \wedge k. \text{karıncası } i, j \text{ kenarını ziyaret ederse,} \\ 0, \wedge \text{diğer durumda} \end{cases} \quad (1)$$

Denklem (1), her iterasyonda i noktasından j noktasına giden yolun feromon değişim miktarını gösterir. Aynı iterasyonda karıncalar farklı yollar tercih edebileceğinden tüm yolların feromon değişim miktarları farklı olabilir. Bir iterasyonda k -karıncası gezinmesi esnasında i, j yolunu ziyaret ederse gezdiği tüm yolun uzunluğunun oranı kadar i, j yoluna feromon bırakır. Gezinme esnasındaki toplam yol ne kadar kısa ise yola bırakılan feromon oranı yüksek olur. Aksi taktirde yol uzadıkça bırakılacak feromon miktarı düşer. Böylelikle

kısa yolun değeri daha fazla olur. Denklemdaki Q sayısı keyfi bir sabit değerdir. Bırakılan feromon miktarının probleme özgü ayarını sağlar.

$$t_{i,j} = t_{i,j} + \sum_{k=1}^{n_{karınca}} \Delta t_{i,j}^k \quad (2)$$

Denklem (2), iterasyon sonunda tüm yolların önceki iterasyondan kalan feromon miktarına iterasyondaki feromon değişim miktarı ekler. Denklem (1) de açıklandığı üzere i,j yolu hiçbir karınca tarafından ziyaret edilmemişse feromon miktarı değişmez.

$$p_{i,j} = \frac{t_{i,j}}{\sum_{k=1}^{i \text{ nin komşu sayısı}} t_{i,k}} \quad (3)$$

Denklem (3), bir karıncanın i noktasından j noktasına giden bir yolu tercih etme olasılığını hesaplar. İ noktasından komşularına geçiş olasılıklarının toplamı 1'dir.

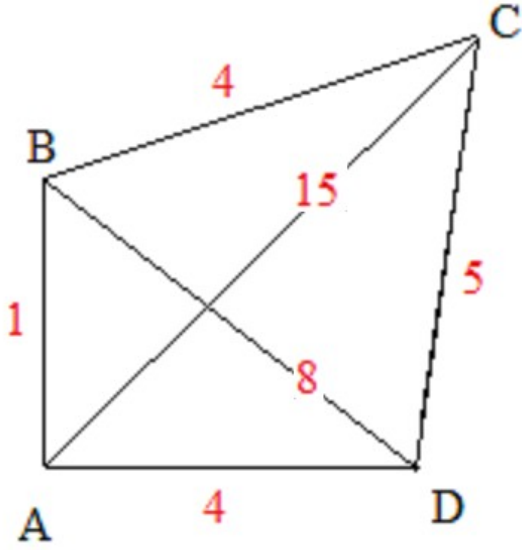
$$t_{i,j} = (1 - e) t_{i,j} \quad (4)$$

Denklem (4), iterasyon sonunda güncellenen feromon değerlerine buharlaşma uygular. Böylelikle tercih edilmeyen yolların değeri zamanla azalır. Denklemdaki e katsayısı buharlaşma katsayısıdır ve 0 ile 1 arasında seçilir.

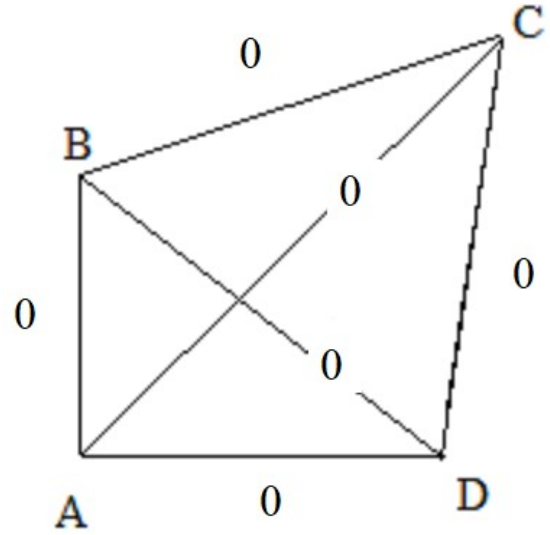
3.3 ACO Algoritması Örneği

A-B-C-D şehirlerinden oluşan ve şehirler arasındaki mesafeleri gösteren graf aşağıda verilmiştir. Başlangıç noktası A noktası olan Gezgin Satıcı Problemi için Karınca Kolonisi Optimizasyon Algoritmasını örneklendirelim.

Not: Bölüm içinde yapılacak hesaplamalarda ondalıklı sonuçlar elde edilmesi halinde ondalıklı kısım 4 basamak hassasiyetinde alınacaktır. Yani ondalıklı kısmın 5 basamağı 5 değerine eşit veya büyük ise 4. basamak yukarı, 5 değerinden küçük ise aşağı yuvarlanacaktır.

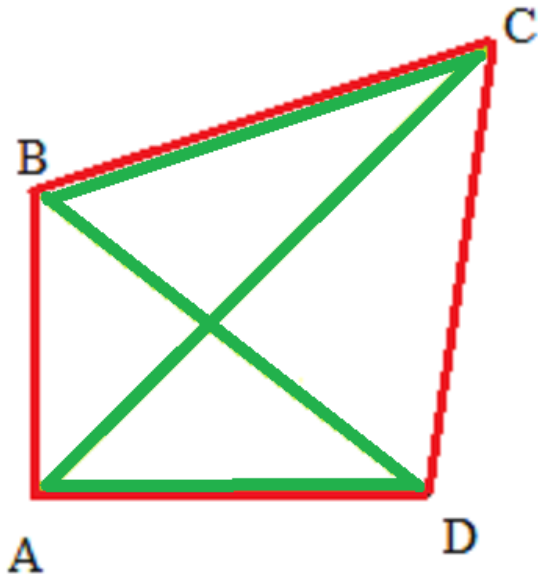


Şekil 9 Maliyet Grafi



Şekil 10 Feromon Grafi

Algoritmaya sadece maliyet grafini vermek yeterlidir. Feromon grafi, maliyet grafiyle aynı yapıya sahip şekilde algoritma tarafından 0 değerleriyle başlatılacaktır.

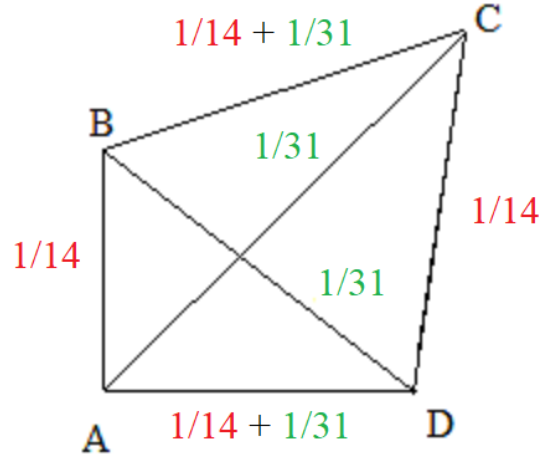


$$L_1 = 1 + 4 + 5 + 4 = 14$$

$$L_2 = 15 + 4 + 8 + 4 = 31$$

Şekil 11 Karıncaların Tercih Ettiği Yollar ve Uzunluk Hesaplaması

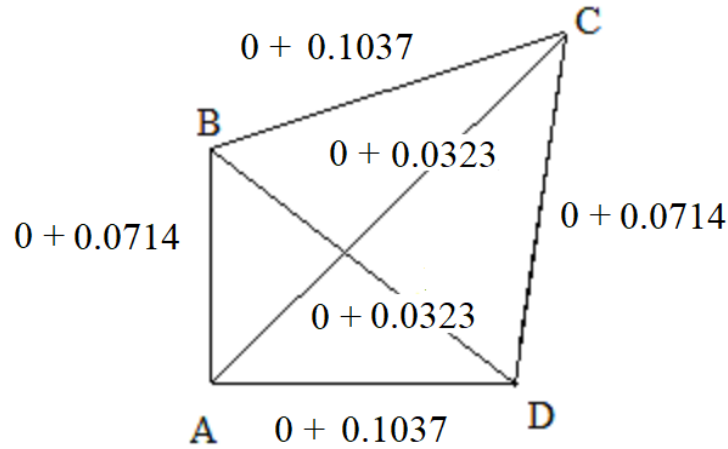
Algoritmada kırmızı ve yeşil renkte 2 adet karınca olduğunu, tercih ettiği yolların aynı renklerle işaretlendiklerini kabul edelim. Kırmızı karınca, ABCDA yolunu, yeşil karınca ACBDA yolunu izleyerek başlangıç noktasına dönmüşlerdir. Karıncaların tercih ettiği yolun toplam uzunlukları, aynı renkte şeklin sağ tarafında verilmiştir. Bu değerler maliyet matrisinden elde edilmiş olup, kırmızı karınca için $AB=1$, $BC=4$, $CD=5$, $DA=4$ olmak üzere tüm değerler toplanıp 14 sonucu elde edilmiştir. Benzer şekilde yeşil karınca için $AC=15$, $CB=4$, $BD=8$, $DA=4$ değerleri toplanıp 31 sonucu elde edilmiştir.



Şekil 12 Karıncaların Ziyaret Ettiği Yollara Bıraktığı Feromon Miktarları

$$\Delta t_{i,j}^k = \begin{cases} \frac{Q}{L_k}, \wedge k. \text{ karınca } i, j \text{ kenarını ziyaret ederse,} \\ 0, \wedge \text{ diğer durumda} \end{cases} \quad (1)$$

Denklem (1) de gösterildiği gibi, karıncaların renkleriyle aynı olacak şekilde yollar üzerine bıraktığı feromonlar yazılmıştır. Burada BC ve AD yolu ortak kullanılan yollar olduğundan değişim miktarları her iki karıncanın bıraktığı feromonların toplamı kadardır.

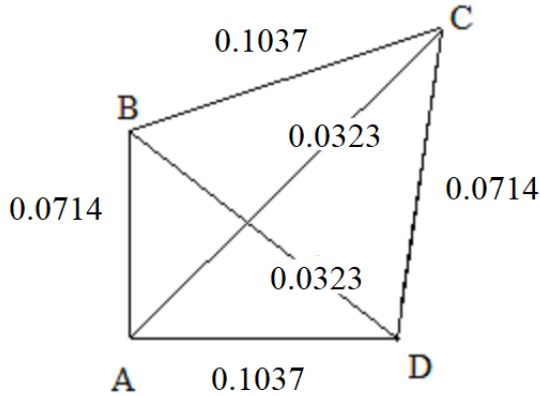


Şekil 13 Örnek Yollar Üzerindeki Feromon Güncellemesi

$$t_{i,j} = t_{i,j} + \sum_{k=1}^{n_{karınca}} \Delta t_{i,j}^k \quad (2)$$

Denklem (2) de gösterildiği gibi, yollar üzerinde oluşan feromon değişimi, önceki iterasyondaki feromon miktarıyla toplanır. (İlk iterasyon olduğundan tüm yollar 0 feromon değeri ile başlatılmıştı). Örneğin AB yolu için $0 + 0.0714$ değerinde, 0 değeri önceki iterasyondan gelen değeri, 0.0714 değeri bu iterasyondaki $\frac{1}{14}$ e karşılık gelen ondalıklı sayıyı

ifade eder. Benzer şekilde BC yolu için $0 + 0.1037$ değerinde, 0 değeri önceki iterasyondan gelen değeri, 0.1037 değeri $\frac{1}{14} + \frac{1}{31}$ toplamına karşılık gelen ondalıklı sayıyı ifade eder.



$$p_{a,c} = \frac{\tau_{a,c}}{\sum_{k=1}^{a'nın\ komşu\ sayısı} \tau_{a,k}} = \frac{0.0323}{0.0714+0.0323+0.1037} = \frac{0.0323}{0.2074} \approx 0.1557$$

$$p_{a,b} = \frac{0.0714}{0.0714+0.0323+0.1037} = \frac{0.0714}{0.2074} \approx 0.3443$$

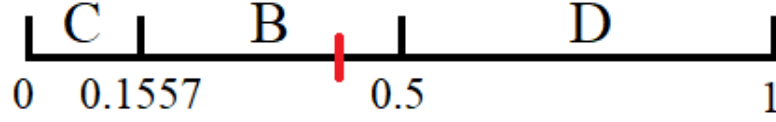
$$p_{a,d} = \frac{0.1037}{0.0714+0.0323+0.1037} = \frac{0.1037}{0.2074} \approx 0.5$$

Şekil 14 Örnek Yol Seçim Olasılığı Hesaplaması

$$p_{i,j} = \frac{t_{i,j}}{\sum_{k=1}^{i'nin\ komşu\ sayısı} t_{i,k}} \quad (3)$$

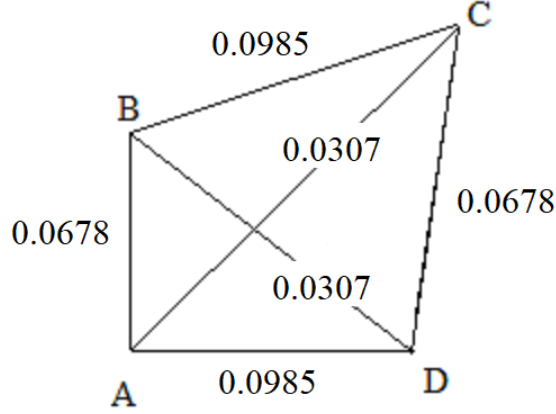
Denklem (3) de gösterildiği gibi, bir karıncanın i noktasından j noktasına giden bir yolu tercih etme olasılığı hesaplanır. Başlangıçta tüm yollar üzerindeki feromon miktarı 0 olduğundan bu adım detayları anlatılmak üzere atlanılmıştı. Ancak normalde algoritmada her iterasyon başlangıcında karıncaların yolu tercih etme olasılığı hesaplanarak karıncalar hareket ettirilir.

Örnek senaryo, A noktasında bir karınca olduğunu ve buranın bir kavşak olduğunu düşünelim. A' nın komşuları B,C ve D' dir, yani A kavşağından gidilebilecek yollardır. Feromon grafından A' nın komşularının feromon değerlerine bakılıp bu değerleri toplanır ve A' nın tüm komşularının feromon değerleri bu toplama bölünür. Böylelikle A kavşağından gidilebilecek yolların olasılıksal değerleri hesaplanmış olur. Karıncalar kararlarında bağımsızdır. Hesaplanan olasılık değerlerinin kümülatif toplamı 1' i verir. 0 ile 1 arasında rastgele bir sayı seçilip kümülatif toplam üzerinde hangi aralığa denk geldiği tespit edilir. Karınca tespit edilen yolu tercih etmiş olur.



Şekil 15 Kümülatif Toplam Barı

Örneğin, 0 ile 1 arasında seçilen rastgele sayının, kırmızı ile işaretlenen 0.40 gelmesi durumunda karınca A kavşağından B yoluna gitmeyi yani AB yolunu kullanmayı tercih edecektir.



$$t_{i,j} = (1 - e) t_{i,j} \quad (4)$$

Denklem (4) de gösterildiği gibi, iterasyon sonunda tüm yollardaki feromona buharlaştırma yapılarak son güncelleme yapılır. Böylelikle tercih edilmeyen yolların zaman içerisinde değeri azaltılır.

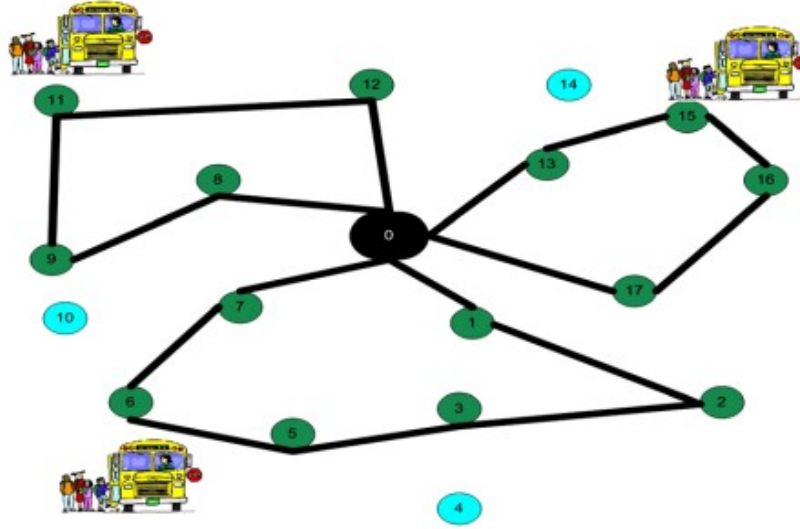
Örneğin, buharlaşma katsayısı e ' nin 0.05 olduğunu, AB yolunun buharlaşmadan önce 0.0714 feromon miktarına sahip olduğunu düşünelim. AB yolunun buharlaşma sonrası feromon miktarı $(1 - 0.05) * 0.0714 = 0.0678$ olarak hesaplanır. Benzer şekilde diğer yollardaki feromon miktarı güncellenir.

Algoritma istenen koşul sağlanıyaya kadar veya istenen iterasyon kadar yukarıdaki işlemler tekrarlanarak en iyi çözüme ulaşmak hedeflenir.

3.4 ACO Algoritması Kullanım Alanları

3.4.1 Araç Rotalarının İyileştirilmesi

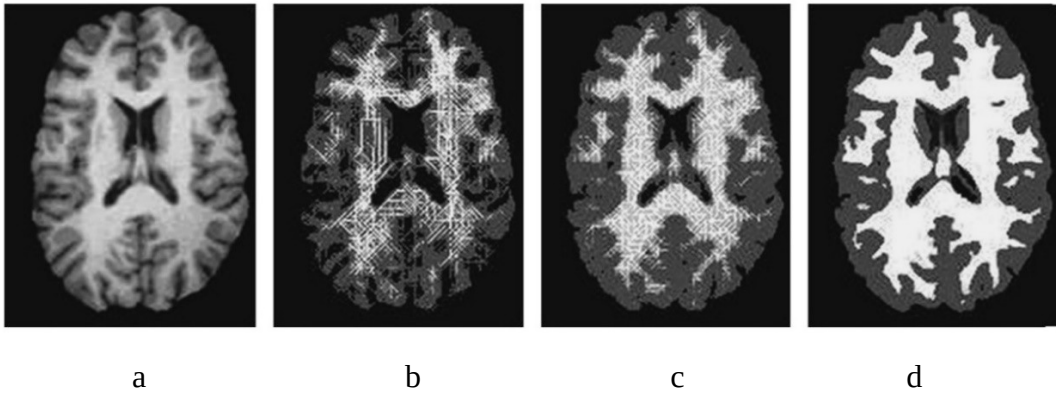
Euchi, J. ve Mrahi, R. (2012), Tunus'un toplu taşıma problemindeki öğrenci servislerinin rotalarının iyileştirilmesi ve bu servislerin minimize edilmesi için ACO ve Değişken Komşuluk algoritmasıyla kombinledikleri bir yaklaşım önermişlerdir. Yaklaşımlarında öğrenci servis güzergahlarını ACO ile belirleyerek bir güzergahtan alınan öğrencilerin maksimize edip, ancak aynı güzergaha minimum servis sağladıkları yaklaşımı ACO ile sağlamışlardır. Buradaki oluşturulan güzergahların amaç fonksiyonunu Değişken Komşuluk algoritmasıyla ölçmüşlerdir. Çalışma sonunda Tunus' ta bulunan bazı servislerin kaldırılması, bu sayede enerji, insan kaynağı maliyetlerin azalabileceği, trafik gibi problemlerin çözülebileceğini önermişlerdir. [6]



Şekil 16 Araç-Durak-Yolcu Grafi

3.4.2 Görüntü Bölütleme

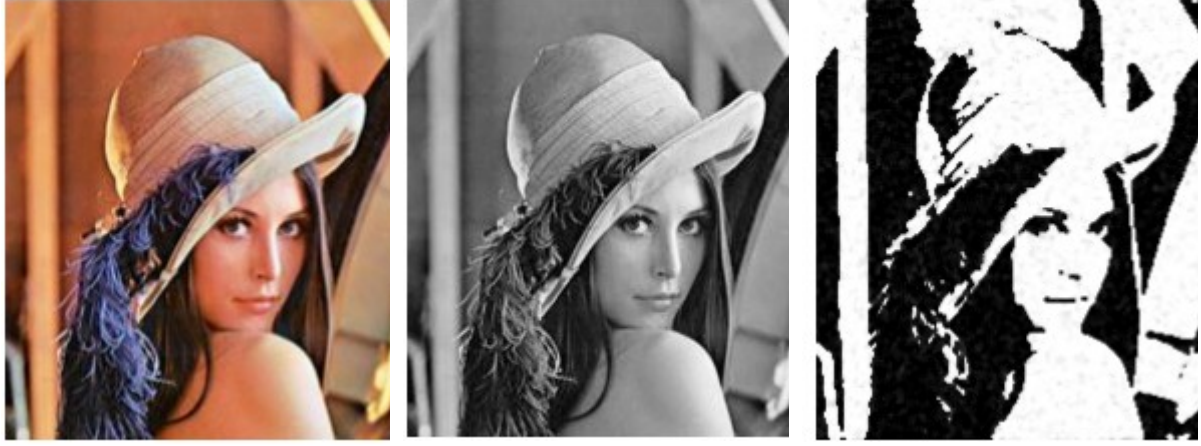
Huang, P. ve arkadaşları (2008)., tıbbi görüntüler (Beyin MR görüntüsü) üzerinde ACO tabanlı bir bölütleme modeli önermişlerdir. Bölütleme görevinde karınca sayısı görüntü piksel sayısının %30 olarak belirleyerek bir görüntü üzerindeki yiyecek tanımının gri seviyeli görüntüler üzerinde bulanık bağlantılılık ölçüsü şeklinde tanımlanmıştır. Bölütleme işlevinde yinelemeli olarak feromonların değerleri güncellemişler ve çok tercih edilen yolların, (feromon salgısının çok olduğu yerler) hedef nesne olduğunu önererek korumuşlardır. Bölütleme görevinde daha önceden kullanılan FCM (Fuzzy C Means)' den elde edilen sonuçlar ile önerdikleri modeli Jaccard benzerlik indeksine göre kıyaslayıp, önerdikleri modelin gürültüden bağımsız ve daha iyi başarı gösterdiğini belirtmişlerdir. [7]



Şekil 17 a-orijinal görüntü, b-d Farklı İterasyon Bölütleme Sonuçları

Zhang, X., ve Dahu, W. (2019), görüntü bölütleme görevi için ACO tabanlı bir algoritma önermişlerdir. Görüntü bölütleme, görüntünün arka planın yok edilmesiyle hedef nesnenin ön plana çıkarma işlevidir. Araştırmacılar ACO' da karıncaların feromon salgısının çok olduğu yeri tercih ederken, lokal minimuma takılma ihtimalinin olduğunu, karıncaların rastgele davranışlarının engellendiğini savunarak ACO' ya yol seçim algoritmasına Balık Sürülerinin Kalabalıklaşma etkisi işlevini tanıtmışlardır. Bu sayede kalabalık olan bir yolun feromon salgısının çok olduğu bilinse de karıncaların burayı tercih etmeden diğer yolları tercih ederek lokal minimum değerine takılmasını, mümkün ise global minimum değerine yakınlaşmalarını

sağlamışlardır. Algoritmayı TSP ve Gri seviyeli Lenna görüntüsü üzerinde uygulayarak, görüntü işleme alanında OTSU yöntemine ve Klasik ACO tabanlı görüntü bölütleme işlevine göre önerilen algoritmanın hem hızlı, hemde bölütleme performansının iyileştiğini belirtmişlerdir. [8]



Orijinal Lena Resmi

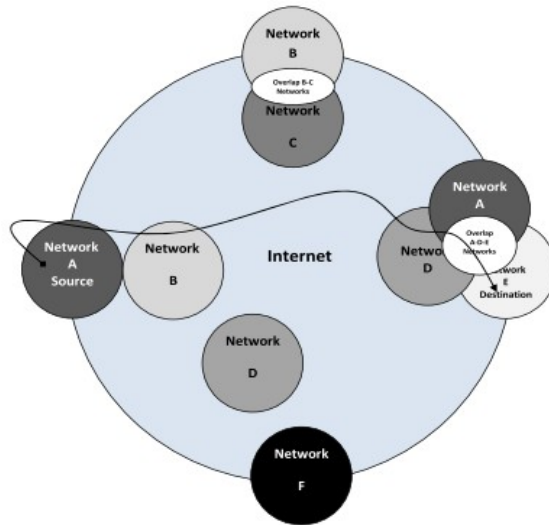
Gri Ölçekli Lena Görüntüsü

ACO ile Bölütlenmiş Görüntü

Şekil 18 Lena Görüntüsü Üzerinde Görüntü Bölütleme

3.4.3 Ağ Yönlendirme

Kefayat, M. ve arkadaşları (2015), nesnelerin interneti (IoT) sistemleri için ACO tabanlı çok ağ yapılı yönlendirme algoritması önerdiler. Ağ yapısını A,B,C,D,E,F olmak üzere 6 alt ağdan oluşturdular. Her alt ağında kendi ACO yönlendirme algoritması olduğunu, ağların kesişim noktaları baz alınarak aktarılmak istenen verinin alt ağlar içinde ACO ile en iyi giriş ve çıkış noktalarının bulunarak enerji tüketimi, bant genişliği tüketimi, veri kaybı, kontrol bitleri ek yükü gibi metriklerin değerlendirilmesi amacıyla Network Simülatör 2 simülasyon ortamında testlerini gerçekleştirdiler. Alternatif ağ yönlendirme algoritmaları olan REL, Markov-A, CASCR algoritmalarına göre daha başarılı olduğunu, önerilen sistemin daha sonradan eklenebilecek alt ağlar ile ölçekleme kolaylığı sağladığını belirtmişlerdir. [9]



Şekil 19 Ağ Rotalama

3.4.4 Yapay Sinir Ağına Veri Sunumu Aşamasında Özellik Seçimi

Beltramo, T. ve arkadaşları (2016), yenilenebilir enerji kaynaklarından biri olan biyogaz üretim süreci tahmini için önerilen Yapay Sinir Ağı (YSA) modelinin girdi parametrelerini ve yapısını optimize etmek için ACO kullandılar. Biyogaz üretim süreci olarak inek gübreleri ile otların fermantasyonunu simüle ettiler. Yaklaşık 100 gün süren süreçte 11 adet çeşitli kimyasal veriler topladılar. Verilerin ilk 70 gününü eğitim verisi olarak, geri kalan verileri test verisi olarak kullandılar. İlk önerilen YSA modeli girdi katmanı 11 nöron ve gizli katmanı 10 nöron olacak şekilde tasarlayıp, ortamdan elde edilen veriler ile modeli eğittiler. Veriler arasındaki korelasyonu ve önemli olan özelliği seçmek için ACO' nun feromon salgısının önemli özellik üzerinde çoğalacağı prensibiyle eşik değeri olarak 0.8' den yüksek 3 adet özelliği seçtiler. YSA yapısını girdi katmanı 3 nöron ve gizli katmanı 3 nörona sahip yeni bir ağ tasarladılar ve yaptıkları eğitim ve testlerde ikinci modelin ilk modele kıyasla daha başarılı olduğunu ve hesaplama karmaşıklığının daha az olduğunu belirtmişlerdir. [10]

4. Karınca Kolonisi Optimizasyon Algoritması Simülasyonu

Görüldüğü üzere, Karınca Kolonisi Optimizasyon Algoritması, karıncaların doğal süreçte yiyecek arama davranışlarının yapay olarak taklit edilmesi üzerine kuruludur. Algoritmanın can alıcı kısmı karıncaların birbirleri ile feromon ismi verilen kimyasal salgı aracılığıyla iletişim kurmasıdır.

Her ne kadar algoritmadaki hesaplar programlamaya doğru olarak aktarılabilir olsa da, bireysel karıncaların nasıl hareket ettiği, sürü zekasının nasıl ortaya çıktığı, yiyeceğin tükenmesi veya konumu değiştiğinde karıncaların adapte sürecinde nasıl davrandıkları doğrudan hesaplama programlamalar ile takip edilmesi ve anlaşılması güçtür. Bu sebeple karıncaların yiyecek arama davranışının simülasyonu tarafımdan oluşturularak <https://github.com/ArifAltioek26/AntColonyOptimizationSimulation> adresine yüklenmiştir.

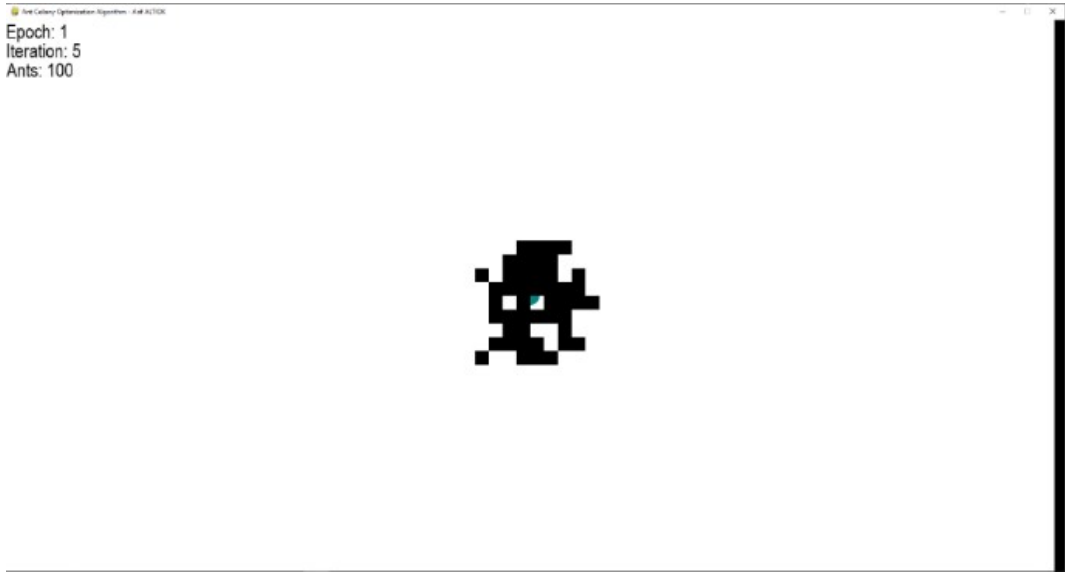
4.1 Simülasyonda Kullanılan Araçlar

- Python programlama dili,
- Numpy kütüphanesi,
- Pygame kütüphanesi,
- Windows 10 işletim sistemi

4.2 Simülasyon Detayları

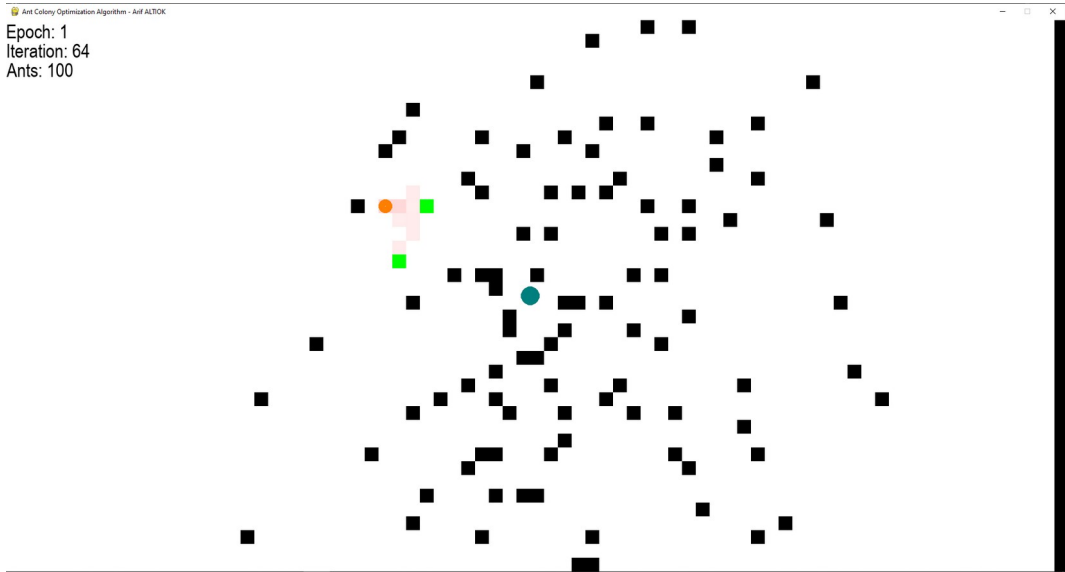
Simülasyon karıncaların doğal süreçte yiyecek arama davranışını simüle etmek amacıyla oluşturulmuştur. Karıncalar her iterasyon başında ekranın ortasında bulunan yuva konumundan harekete başlar. Karıncaların hareket kısıtı iterasyon başına 100 olarak belirlenmiştir. Karıncalar yuvadan hareket ettiğinde siyah renkli, yiyeceğe ulaşp yuvaya geri döndüğü esnada yeşil renkli, feromonlar pembe renkli kare olarak gözükür. Ortak kullanılan yollarda biriken feromonlar zaman içerisinde pembeden kırmızı renge, tercih edilmeyen yollar ise buharlaşma etkisi ile ilk hali olan beyaz renge döner. Yiyecekler farenin sol tuşu eklenebilir, sağ tuşu ile silinebilir. İşlem yapılacak yiyeceğin konumu farenin ekran üzerindeki konumudur. Her bir tıklama 100 birim yiyecek anlamına gelir. Yiyecekler turuncu çember şeklinde olup, miktarına göre çemberin kalınlığı değişir. Simülasyon daha detaylı incelemek adına boşluk (space) tuşu ile durdurulabilir. Yine aynı tuş ile devam ettirilir.

4.3 Simülasyon Ekran Çıktıları



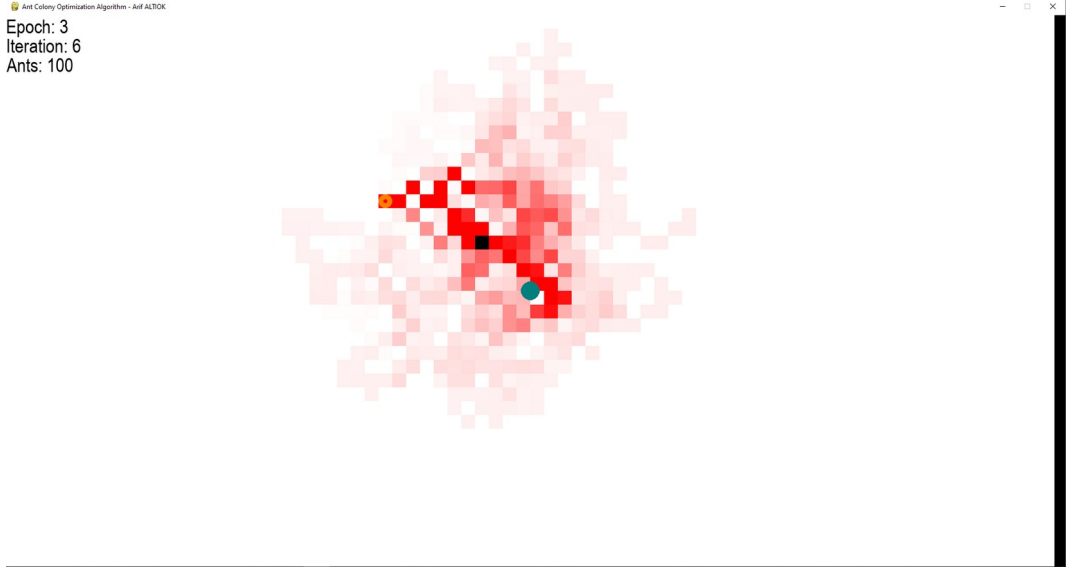
Şekil 20 Simülasyon Başlangıç Durumu

Başlangıçta karıncalar yuvadan harekete başlamıştır. Yapay dünyada herhangi bir konumda feromon olmadığından karıncalar şuan rastgele kararlar vermektedir. Ayrıca şuan herhangi bir konuma yiyecek bırakılmamıştır.



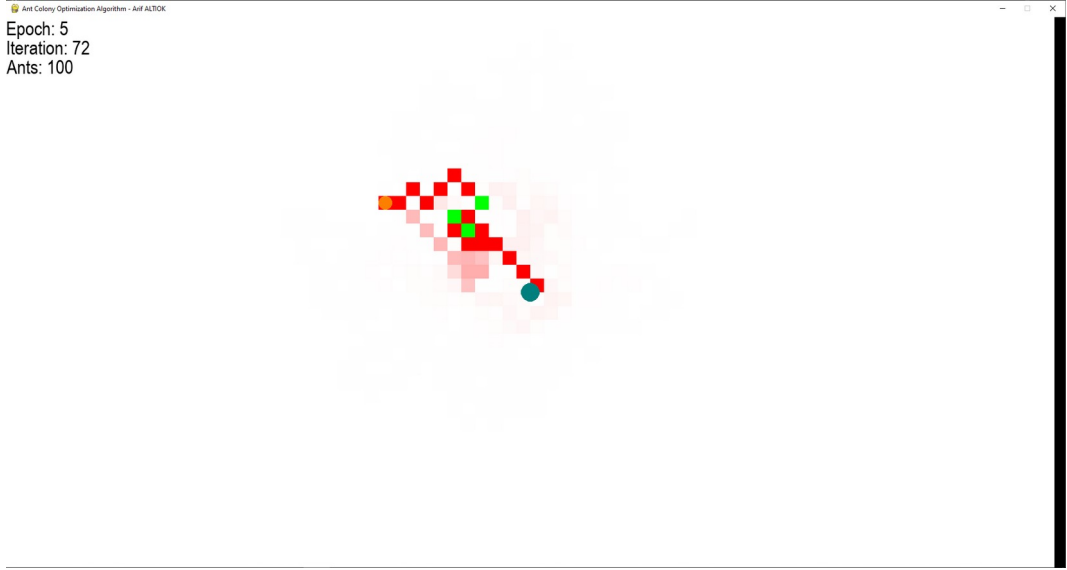
Şekil 21 Yiyecek Bulunma Durumu

Yuva konumuna göre sol üst konuma yiyecek bırakılmıştır. O konumda rastgele hareket eden birkaç karınca yiyeceğe ulaşmış ve yeşil renkte gösterilmiştir. Bu karıncalar geri izleme adımlarına geçmiş ve bıraktıkları feromonlar açık pembe tonları ile gösterilmiştir.



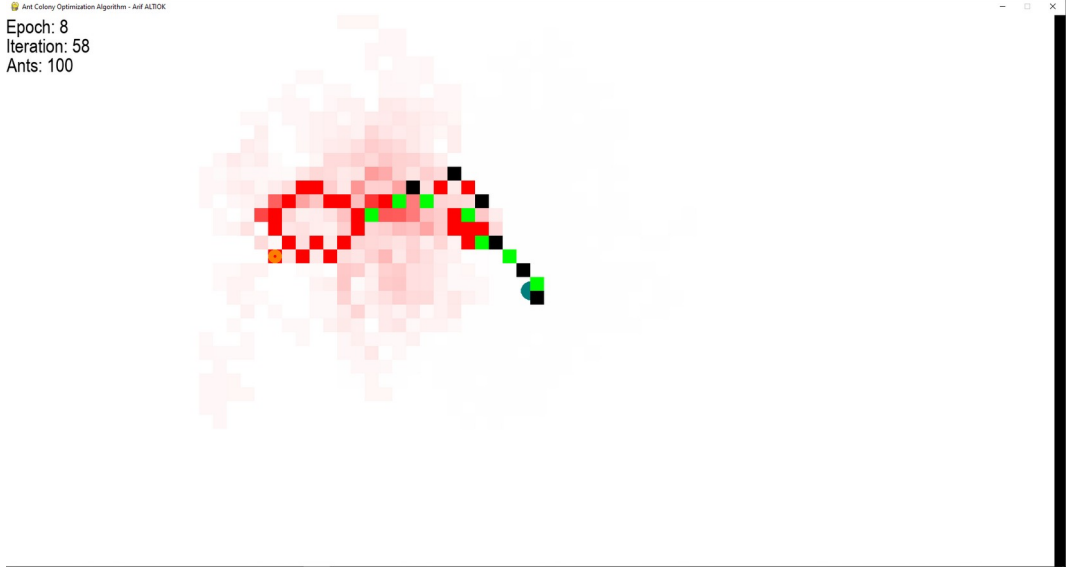
Şekil 22 Ortak Yolların Görülmesi

Devam eden iterasyonlar ile daha fazla karınca aynı konumda bulunan yiyeceğe ulaşmış ve ortak yollar çoğalmıştır. Ortak yollara bırakılan feromonlar daha yoğun olduğundan kırmızı renkli, az tercih edilen yollar pembenin tonları olarak gösterilmiştir.



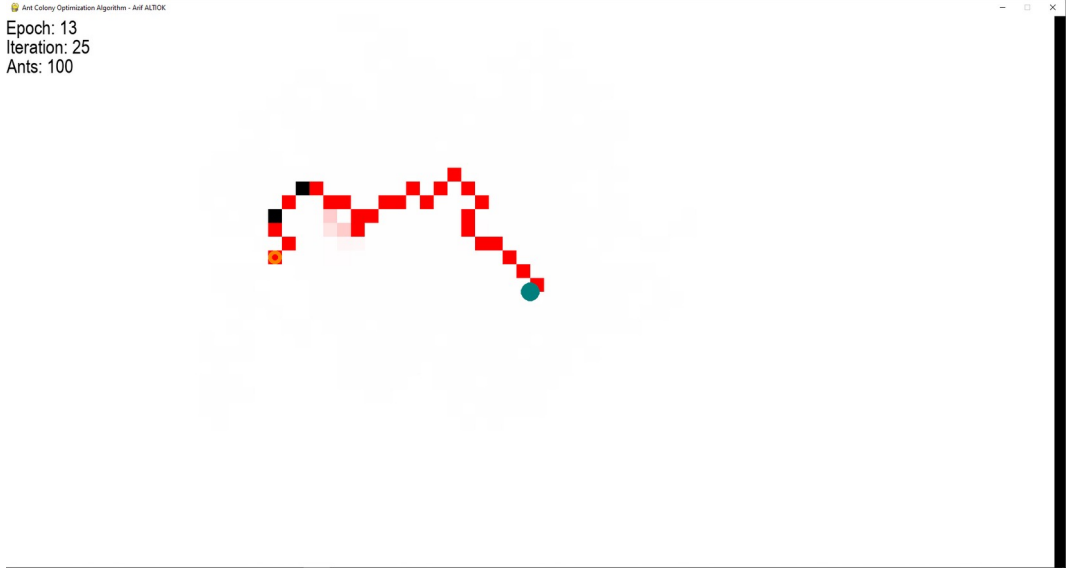
Şekil 23 Ortak Yolun Belirginleşmesi

Karıncalar yolları olasılıksal olarak feromon yoğunluğuna göre değerlendirdiğinden az feromon miktarı olan yolları tercih etmezler. Bu yollar buharlaşma etkisi ile zaman içerisinde beyaz renge döner.



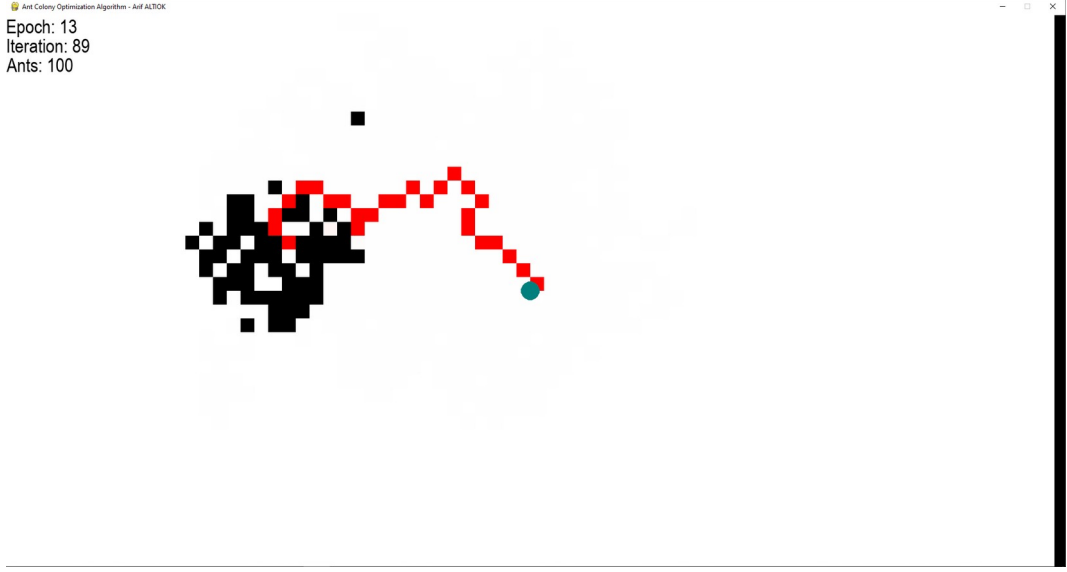
Şekil 24 Karıncaların Adaptasyon Süreci

Üst senaryolarda bahsedilen yiyecek konumunda yiyecek bitmiştir ve alt tarafına yeni bir yiyecek eklenmiştir. Karıncalar önceki yiyecek konumundan tekrar rastgele hareketler ile yeni yiyecek arayışına girmiş ve bu yiyeceği de keşfetmiştir. Bu durumun değişen ortam koşullarında karıncaların adapte olma sürecini gösterir.



Şekil 25 Adaptasyon Sürecinde Oluşan Yeni Ortak Yol

Yeni eklenen yiyecek için karıncalar rastgele aramalar yapmıştı. Üsteki görselde yiyecek konumuna yakın olmayan feromonlar görülmektedir. Ancak bu yollar zaman içerisinde tercih edilmediğinden buharlaşma neticesinde kayboldular ve koloni ortak kullandığı yolu tekrar ortaya çıkardı.



Şekil 26 Yiyeceğin Tükenmesi

Yiyeceğin tükenmesi ile karıncalar tekrar yeni yiyecek keşfetmek amacıyla arama davranışına geçer ve bu konumlarda feromon olmadığından rastgele olarak hareket eder.

5. Kaynaklar

- [1] <https://tr.wikipedia.org/wiki/Kar%C4%B1nca> erişim tarihi 22.12.2021
- [2] <https://www.teknikturk.com.tr/hasere-rehberi/karinca/8> [erişim tarihi 22.12.2021](#)
- [3] <https://www.youtube.com/watch?v=qYXeMFCf1Gk> erişim tarihi 16.12.2021
- [4] <https://towardsdatascience.com/the-inspiration-of-an-ant-colony-optimization-f377568ea03f> erişim tarihi 16.12.2021
- [5] Alaykiran, K., & Engin, O. (2005). Karınca Kolonileri Metasezgiseli ve Gezgin Satıcı Problemleri Üzerinde Bir Uygulaması. Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi, 20(1).
- [6] Euch, J., & Mrahi, R. (2012). The urban bus routing problem in the Tunisian case by the hybrid artificial ant colony algorithm. Swarm and Evolutionary Computation, 2, 15-24.
- [7] Huang, P., Cao, H., & Luo, S. (2008). An artificial ant colonies approach to medical image segmentation. Computer methods and programs in biomedicine, 92(3), 267-273.
- [8] Zhang, X., & Dahu, W. (2019). Application of artificial intelligence algorithms in image processing. Journal of Visual Communication and Image Representation, 61, 42-49.
- [9] Kefayat, M., Ara, A. L., & Niaki, S. N. (2015). A hybrid of ant colony optimization and artificial bee colony algorithm for probabilistic optimal placement and sizing of distributed energy resources. Energy Conversion and Management, 92, 149-161.

- [10] Beltramo, T., Ranzan, C., Hinrichs, J., & Hitzmann, B. (2016). Artificial neural network prediction of the biogas flow rate optimised with an ant colony algorithm. *Biosystems Engineering*, 143, 68-78.
- [11] <https://github.com/ArifAltioek26/AntColonyOptimizationSimulation>