

GSP ÇÖZÜMÜ İÇİN KARINCA KOLONİSİ OPTİMİZASYONU¹

Arş. Gör. Dr. Arzu EREN ŞENARAS²

Uludağ Üniversitesi
İktisadi ve İdari Bilimler Fakültesi,
Ekonometri Bölümü
arzueren@uludag.edu.tr

Öğr. Gör. Şahin İNANÇ

Uludağ Üniversitesi,
Keles Meslek Yüksekokulu
Bilgisayar Teknolojileri
sahininanc@uludag.edu.tr

ÖZET

Araç Rotalama Problemleri (ARP) birçok farklı yöntemle çözülebilmektedir. Bu çözüm yöntemleri sezgisel ve sezgisel olmayan yöntemler olarak iki ana gruba ayrılmaktadır. ARP de en kısa yol problem tiplerinin çözümünde düğüm sayısı fazla olan problemlerin sezgisel olmayan yöntemlerle çözümü imkansız olabilmektedir. Bu tür problemlerde sezgisel yöntemlerin kullanılması daha uygundur. Gezgin satıcı tarzı problemlerde veya en kısa yolun bulunması tarzındaki problemlerin çözümünde karınca kolonisi optimizasyonu en başarılı sezgisel yöntemlerden biridir. Bu çalışmada karınca kolonisi optimizasyonu örnek bir kargo şirketi için uygulanmıştır. Kargo şirketine ait bir aracın gezgin satıcı tarzı problemlerde olduğu gibi uğrayacağı şehirlere tekrar uğramamak koşuluyla her şehre uğrayıp başladığı yere dönecek şekilde 20 il (düğüm) için çözüm bulunmuştur. Ayrıca daha az düğüm ile karınca kolonisi optimizasyonu programının optimal çözüme yakın bir çözüm bulup bulunmadığını test etmek için de sezgisel olmayan algoritma ile çözüm bulunup sonuçlarla karşılaştırılmıştır.

Anahtar Kelimeler: Karınca Kolonisi Algoritması, Araç Rotalama Problemi, Gezgin Satıcı Problemleri, Sezgisel Algoritmalar.

ANT COLONY OPTIMIZATION FOR SOLVING VEHICLE ROUTING PROBLEMS

ABSTRACT

Vehicle Routing Problems (VRP) can be solved in many different ways. These solution methods are divided into two main groups as heuristic and non-heuristic methods. In VRP the solution of shortest path problem types is impossible to solve problems with large numbers of nodes by non-heuristic methods. It is more convenient to use heuristic methods in such problems. Ant colony optimization is one of the most successful heuristic methods in solving the problem of traveling sales man problems or finding the shortest path. In this study, ant colony optimization was applied to a sample cargo company. The vehicle of a cargo company, there are solutions for 20 provinces (nodes) so that they will stop at the cities where they will be visited as if they are traveling sales man problems. We also compared the results with non-heuristic algorithms to test whether the ant colony optimization program with the fewer nodules can find a solution that is close to optimal solution.

Keywords: Ant Colony Optimization, Vehicle Routing, Travelling Salesman Problems, Heuristic Algorithms.

¹Bu çalışma 20-21 Mayıs 2017 tarihinde İstanbul'da düzenlenen International Congress of Management Economy and Policy (ICOMEPEP)'te sunulan bildirinin genişletilmiş halidir.

²Sorumlu Yazar(Corresponding Author).

1. GİRİŞ

GSP'nin genel biçimi başta Karl Menger olmak üzere 1930'larda Viyana ve Harvard'daki matematikçiler tarafından çalışılmıştır. 1930'lardan günümüze kadar GSP'nin çözümü için birçok kesin ve sezgisel yöntem geliştirilmiştir (Mondal vd., 2013).

Gezgin Satıcı Problemi (GSP) herhangi bir şehirden başlayarak aralarındaki mesafeler belli olan şehirleri bir kez ziyaret ederek en kısa ve en az maliyetli yolun bulunarak tekrar başlangıç şehrine dönülmesini hedefleyen bir problemdir. GSP'nin formülasyonu kolay olmasına rağmen çözümü zordur. Şehir sayısının artmasına bağlı olarak problemin çözümünün karmaşıklığı da artmaktadır. Çözümünün zor ve uygulama alanının fazla olması nedeniyle çözüme yönelik birçok çalışma gerçekleştirilmiştir. Problemin çözümünde geleneksel yöntemlerin kullanılması değişken sayısının artmasıyla beraber yetersiz hale gelmiştir. Günümüzde, probleme çözüm üretmek ve optimizasyonunu sağlamak amacıyla yapay zekâ tekniklerinden yararlanılmaya başlanmıştır (Dikmen vd., 2014).

Literatürde karınca kolonisi algoritması ile GSP çözümlerine ilişkin yapılan çalışmalardan bazıları aşağıdaki gibidir.

Shang vd.(2007), seyahat eden satış elemanı problemi tarzı problemleri çözmek için karınca kolonisi optimizasyonunu ve birleştirme kuralını birleştirerek yeni bir algoritma önermişlerdir.

Lizárraga vd.(2013), karınca kolonisi optimizasyonu ile gezgin satıcı problemini farklı bir yolla çözmeyi denemişlerdir. Bu yöntemde karınca kolonisi algoritması üç ayrı alt algoritma şeklinde çalışmaktadır. Her bir algoritma kendine göre en optimal yolu bulmaya çalışmaktadır. Bu yöntem, karıncalar aynı yinelemede farklı şekillerde değerlendirildiği için, işin bölünmesinde uygulanacak olan "böl ve fethet" fikrine dayanmaktadır. Bu yöntem aynı zamanda, birkaç iterasyondan sonra düzgün çalışmıyorsa belirli bir iterasyon sonrasında duran bir durgunluk mekanizmasını da içerir.

Brezina ve Cicková (2011), yaptıkları çalışmada bazı kontrol parametrelerini değiştirerek karınca kolonisi algoritması ile gezgin satıcı problemlerinin optimal çözümünün nasıl etkilendiğini araştırmışlardır.

Mohsen(2016), çalışmasında karınca kolonisi algoritması, benzetim tavlama, mutasyon operatörünün avantajlarını ve yerel arama prosedürünü gezici satış elemanı problemini çözmek için birleştiren bir hibrid karınca kolonisi algoritması önermiştir.

Bajpai ve Yadav (2015), çalışmasında, seyirci problemi için bölme ve fethet stratejisine dayanan bölünme tekniğini tanıtmaktadır. Orijinal problem daha küçük alt problemlere bölünerek karınca kolonisi algoritmasına uygulanır ve sonra her küçük alt problemler için aday liste stratejisi kullanılır. Daha sonra da çözümü bulunan yerel alt problemler birleştirilerek karıncaların arama verimliliğini arttırarak orijinal sorunun iyi bir çözümünü bulunur.

Bu çalışmada karınca kolonisi optimizasyonu örnek bir kargo şirketi için uygulanmıştır. Kargo şirketine ait bir aracın gezgin satıcı tarzı problemlerde olduğu gibi uğrayacağı şehirlere tekrar uğramamak koşuluyla her şehre uğrayıp başladığı yere dönecek şekilde 20 il (düğüm) için çözüm bulunmuştur. Ayrıca daha az düğüm ile karınca kolonisi optimizasyonu programının optimal çözüme yakın bir çözüm bulup bulunamadığını test etmek için de sezgisel olmayan algoritma ile çözüm bulunup sonuçlarla karşılaştırılmıştır.

2. KARINCA KOLONİSİ OPTİMİZASYONU

Temel ilkeleri ilk kez Marco Dorigo tarafından ortaya atılmış olan karınca kolonisi algoritmaları, karınca kolonilerinin feromon salgılayarak yiyecek kaynakları ile yuvaları

arasındaki en kısa yolu bulma yöntemlerinden esinlenerek oluşturulmuş bir tekniktir (Demircioğlu, 2009).

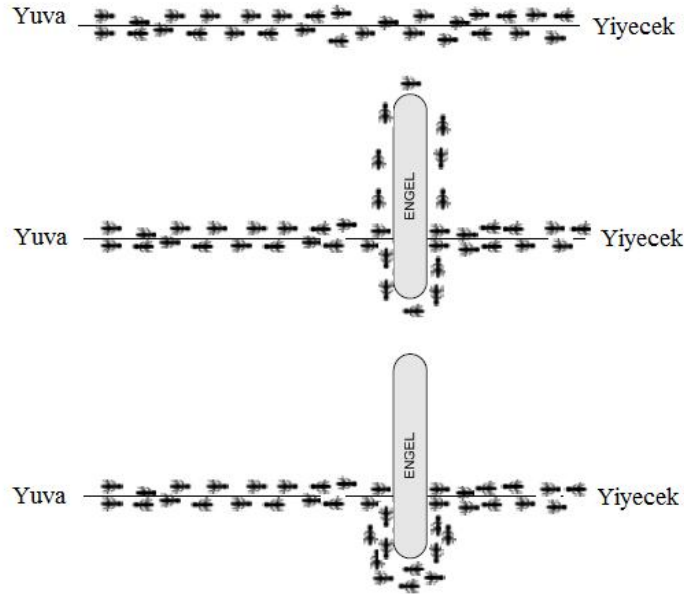
Karınca Kolonisi Optimizasyonu (ACO) NP-hard optimizasyon problemlerine başarılı bir şekilde uygulanabilen popülasyon temelli bir yaklaşımdır (Dorigo & Gambardella, 1997). İlk olarak Gezgin Satıcı Problemi (Travelling Salesman Problem) (TSP) üzerinde uygulanmıştır. Bir noktadan başlayarak ve tüm noktalara bir kez uğrayarak tekrar aynı noktaya en kısa yoldan dönmek diye tanımlanan TSP'nin çözümünde ACO'nun oldukça etkili olduğu görülmüştür (Dorigo & Gambardella, 1996).

Dorigo, karınca kolonilerinin davranışlarının matematiksel modelleri üzerine dayandığı karınca kolonisi algoritmalarını ilk kez gezgin satıcı problemi üzerinde kullanmış ve olumlu sonuçlar elde etmiştir. Bunun üzerine karınca kolonisi algoritmaları diğer araştırmacılar tarafından da kullanılmaya başlanmış ve günümüzde eniyileme problemlerinin çözümünde yaygın olarak kullanılan bir yapay zekâ tekniği haline gelmiştir (Dorigo vd., 1999).

ACO gerçek karınca kolonilerinin davranışlarından esinlenilerek geliştirilmiştir. Tekniğin en temel unsurlarından biri haberleşme aracı olarak kullanılan ve problemlerde çözümün kalitesini gösteren feromon kimyasalıdır. Feromon, gerçek karıncaların da bir haberleşme ve yön bulma aracı olarak kullandıkları, vücutlarından salgıladıkları kimyasaldır. Feromon izleri, karıncalar tarafından güncellenmekte ve bir bilgiyi temsil etmektedir. Bir yolda feromon izinin yoğun olması, yolun kalitesini gösterir ve tercih olasılığını artırır. ACO'da, yapay karıncalar, gerçek mesafeler dikkate alınarak yapılmış olan model üzerinde en kısa yolu araştırmaktadırlar. Yollardaki feromon izleri yine yapay olarak, karıncaların geçiş sıklığıyla orantılı bir şekilde güncellenmektedir (Keskintürk & Söyler, 2006).

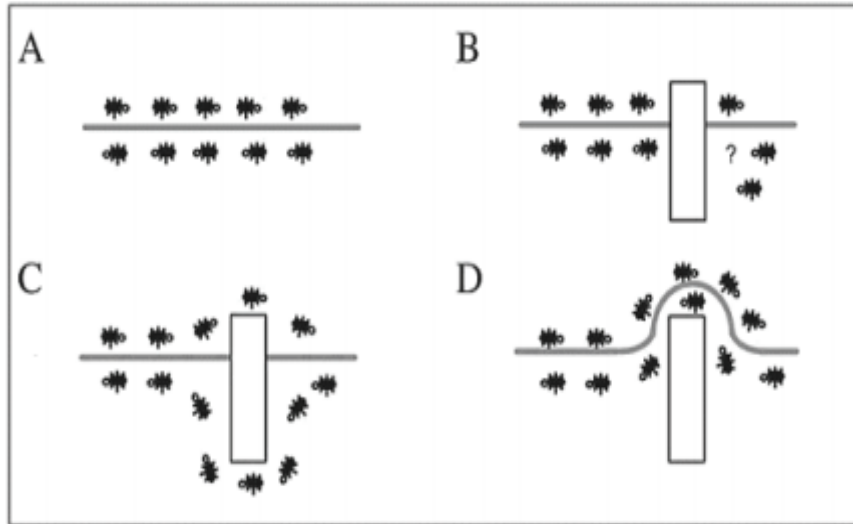
Karınca koloni optimizasyonlarının asıl kaynağı gerçek karıncaların yiyecek arama hareketidir. Karıncalar yiyecek ararken öncelikle kendi yuvalarına yakın çevreleri rastgele araştırırlar. Karıncalardan biri yiyecek kaynağı bulduğunda bu kaynağı kalite ve miktar açısından değerlendirir ve bir miktarını yuvasına taşır. Karınca yuvasına geri dönüş yolunda yol güzergâhına kimyasal feromon izleri adı verilen bir madde bırakır. Bırakılan feromon miktarı karıncanın bulunduğu yiyecek miktarı ve kalitesiyle ilişkilidir. Yola bırakılan bu feromon izleri diğer karıncaların bu yiyecek kaynağına ulaşabilmesi için yol gösterir. Bu feromon izleriyle karıncalar arasındaki dolaylı ilişki karıncaların yuvalarıyla yiyecek kaynağı arasındaki en kısa yolu bulmalarına yardımcı olur. Gerçek karıncalardaki bu karakteristik özellik tümleşik optimizasyon problemlerini çözmek için yapay karınca kolonilerinde kullanılmıştır (Serin, 2009).

Şekil 1: Karınca Kolonisi Algoritması



Algoritmanın ilk uygulamaları gezgin satıcı problemi ve diğer rotalama problemleri üzerine odaklanmasına rağmen günümüzde algoritma; en kısa müşterek üstdizi (shortest common super sequence), genelleştirilmiş atama, çoklu sırt çantası, kısıt sağlama problemlerinin de aralarında bulunduğu geniş bir kombinatoriyel optimizasyon (eniyeleme) problemleri yelpazesine uygulanmıştır (Cordonvd, 2002).

Şekil 2: (a) Karıncalar yiyecek kaynağı ve yuva arasında feromon izi bırakırlar (b) yerleştirilen bir engel izi böler (c) karıncalar engelin etrafından dolaşarak iki farklı yol bulurlar (d) daha kısa olan yol boyunca yeni bir feromon izi oluşur.



2.1. Karınca Turunun Oluşturulması

Karınca kolonisinde ilk olarak kaç tane karınca olacağı belirlenir. Daha sonra her bir karınca rastgele olarak bir düğüme yerleştirilir ve bütün düğümleri tek tek ziyaret ederek turunu tamamlar. Her bir karıncanın mevcut düğümden bir sonraki düğüme gidebilmesinin matematiksel formülü Denklem 1’de verilmiştir (Özdemir, 2008; Serin, 2009; Dikmen, 2014).

$$P_{i,j}^l = \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in N_i^l} [\tau_{il}]^\alpha [\eta_{il}]^\beta}, \text{ eğer } j \in N_i^l \quad (2.1)$$

Burada,

- $P_{i,j}^l$ l karıncasının i düğümünden j düğüme geçme olasılığı
 τ_{ij} i ve j düğümleri arasındaki feromon değeri
 η_{ij} i ve j düğümleri arasındaki sezgisel değeri
 α feromon katsayısı
 β sezgisel katsayısı
 N düğümler kümesi

2.2. Karınca Kolonisi Algoritması Parametreleri

Karınca Sayısı: Kolonide kaç tane karıncanın olacağını belirleyen parametredir(Dikmen, 2014).

İterasyon Sayısı: Arama işleminin kaç iterasyon (adım) gerçekleşeceğini belirleyen parametredir.

Feromon Kuvvetlendirme Oranı (α): Düğümler arasındaki feromon miktarlarının önem derecesini belirleyen parametredir.

Sezgisellik Kuvvetlendirme Oranı (β): Düğümler arasındaki mesafenin önem derecesini belirleyen parametredir.

Feromon Buharlaştırma Oranı (ρ): Her iterasyon sonunda düğümler arasındaki feromonların hangi oranda buharlaşacağını belirleyen parametredir.

3. C# İLE GELİŞTİRİLEN KARINCA KOLONİSİ ALGORİTMASI

Gezgin satıcı tarzı problemlerde veya en kısa yolun bulunması tarzındaki problemlerin çözümünde karınca kolonisi optimizasyonu en başarılı sezgisel yöntemlerden biridir.

Kuzu vd.(2014), çalışmalarında GSP’nin çözümünde kullanılan, tek noktadan arama yapan Tepe Tırmanma, İteratif Yerel Arama, Tavlama Benzetimi, Tabu Arama ve Kanguru Algoritmaları ile popülasyon temelli yöntemlerden Yapay Arı Kolonisi, Genetik Algoritma ve Karınca Kolonisi Algoritmasını incelemişlerdir. Genel olarak birden fazla noktada arama yapan meta sezgisellerin, tek nokta araması yapan algoritmalara göre daha iyi sonuç verdiğini belirtmişlerdir. Karınca kolonisi algoritmasının GSP problemlerine oldukça tatmin edici sonuçlar verdiğini göstermişlerdir. Bu bağlamda, bu çalışmada karınca kolonisi optimizasyonu bir kargo şirketi için uygulanmıştır. Kargo şirketi için gerçekleştirilen bu uygulama lojistik firmaları içinde örnek teşkil etmesi yönüyle literatüre katkı sağlamaktadır.

Kargo şirketine ait bir aracın gezgin satıcı tarzı problemlerde olduğu gibi uğrayacağı şehirlere tekrar uğramamak koşuluyla her şehre uğrayıp başladığı yere dönecek şekilde 20 il

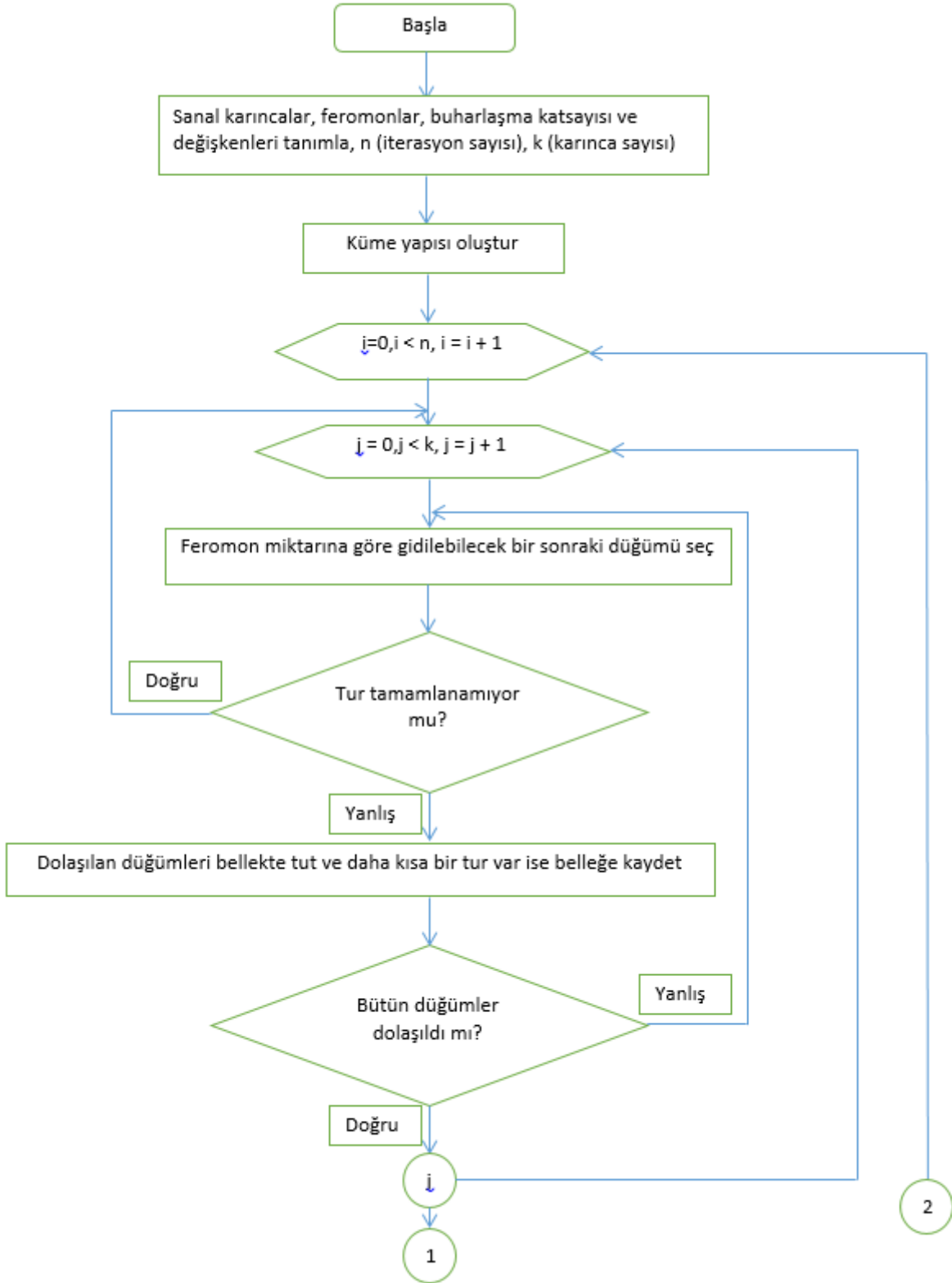
(düğüm) için çözüm bulunmuştur. C# programlama dilinde geliştirilen uygulamanın algoritması Tablo 1'deki gibidir.

Tablo 1: Algoritma Adımları

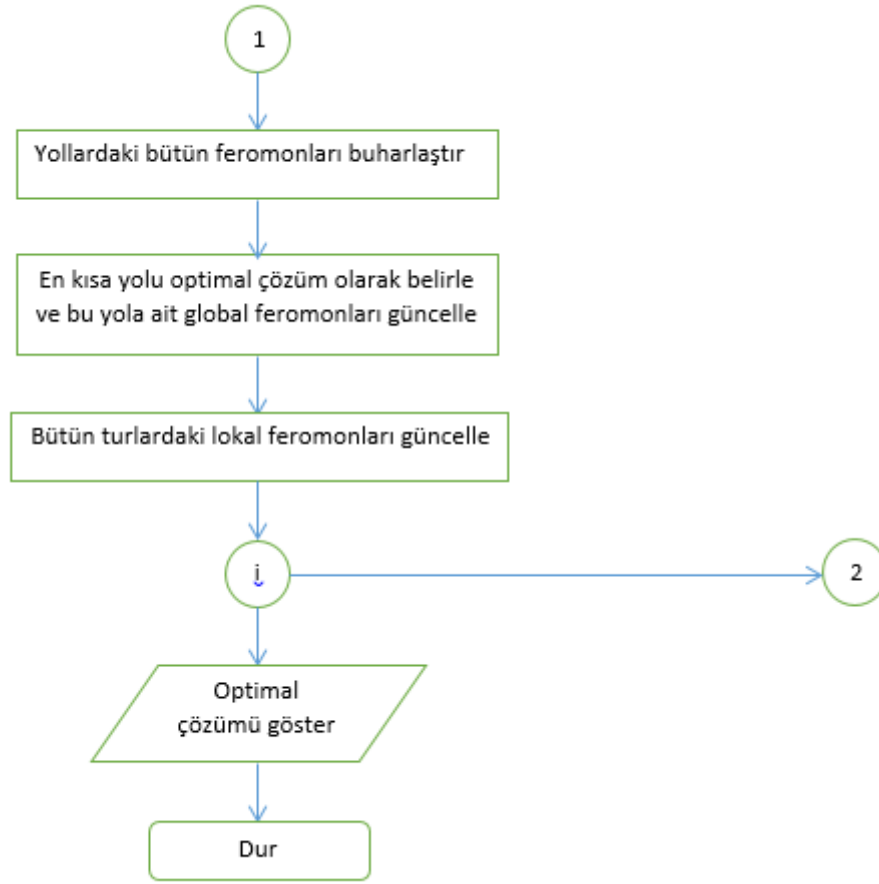
Adım	Algoritma
1	Başla
2	Sanal karıncalar, feromonlar, buharlaşma katsayısı gibi değişkenleri tanımla
3	Her bir düğümden diğer gidilebilecek düğümlere ilişkin bir küme yapısı oluştur.
4	Karıncaların ne kadar süre turlarına devam edeceklerine ilişkin bir döngü oluştur.
5	Her düğüme bir karınca denk gelecek şekilde sanal karıncalarla turlar oluştur.
6	Her turu düğümler bitene kadar ya da turun tamamlanmasını engelleyen çıkmaz vb. gibi durumlara karşılaşılan kadar yola devam et.
7	Yola devam ederken her düğümden feromon yola ait feromon değerlerini kontrol et ve belirli bir olasılık değerine göre (örneğin 0.9 gibi bir olasılıkla) fazla feromon değeri olan yolu seç.
8	$1-0.9 = 0.1$ olasılıkla da yollardaki feromon miktarına göre rasgele bir yol seç.
9	Her bir tur için yolların uzunluklarını toplam uzunluğunu ve yönünü hafızaya yaz
10	Tamamlanamayan turları yok et
11	Turlar tamamlandıktan sonra her yoldaki feromon değerlerini belirli bir oranda buharlaştır.
12	Her tur için yollardaki lokal feromon değerlerini artır.
13	En kısa yola sahip turun geçtiği yollardaki global feromon değerlerini artır ve bu yolu geçici optimal çözüm olarak belirle
14	Döngü bitmedi ise adım 4'e git
15	Geçici optimal çözümü optimal çözüm olarak seç
16	Dur

Programın akış çizgesi Şekil 3 ve Şekil 4'teki gibidir.

Şekil 3: Karınca Kolonisi Algoritmasının Akış Çizgesi-1



Şekil 4: Karınca Kolonisi Algoritmasının Akış Çizgesi-2



Yapılan bu çalışmada karınca kolonisi algoritmasına göre bütün düğümlere uğrayarak ve uğranan düğüme tekrar uğramamak koşuluyla ve en son olarak başlangıç noktasına dönerek kapalı bir tur oluşturacak şekilde en kısa yolu hesaplamak için kullanılmıştır. Bu tarz problemlere gezgin satıcı problemleri adı verilmektedir.

Bu çalışmada uygulama olarak başlangıçta 20 düğüm seçilmiştir. 20 düğüm ile bulunan çözüm düğüm sayısı fazla olmadığından sezgisel olmayan yöntemlerle de kıyaslanabilme imkânı vardır.

Her iki yöntemle de sonuçlar optimal ve aynı çıkmıştır. Daha fazla düğüm söz konusu olduğunda sonucun tam olarak optimal çıkmama durumu olmayabilir, fakat optimale yakın bir değer bulunabilir. Fazla düğüm için sezgisel olmayan yöntemlerle optimal çözüm bulunamayacağı için çözümün gerçekten optimal olup olmadığı tam olarak bilinmemektedir.

Uygulamada önce her düğümden hangi düğümlere gidilebileceği ile ilgili bir küme yapısı (dizi yapısı) oluşturulmuştur. Düğümler arası yollar feromon bilgilerini kaydetmek amacıyla dizi şeklinde tanımlanıp başlangıç feromonları ayarlanmıştır. Daha sonra başlangıç düğümünden başlanarak her düğüm geçişinde bir önceki düğüm tekrar aynı düğüme dönmek için işaretlenip sonraki geçilecek olan olası düğümlerin feromon değerlerine bakılarak bir sonraki düğüme geçilmiştir. Burada eğer bir sonraki düğüme geçiş yok ise ve bütün düğümlere uğranmamış ise o karınca için çözüm bulunamadığından tur (döngü) sonlandırılır.

Hangi düğüm tercih edileceğine karar verilirken ise yollardaki feromon miktarlarına bakılarak %90 olasılıkla en çok feromon bulunan yol seçilir. %10 olasılıkla da olası gidilebilecek yollardaki feromonlardan biri montecarlo yöntemi ile seçilerek karar verilir. Böylece karıncalar daima feromonun çok olduğu yolu değil de %10 olasılıkla olası diğer yolları da araştırır. Daha kısa yol bulunması durumunda iterasyon sonunda eğer bu tur diğer bütün turlardan daha iyi bir çözüme sahipse bu tura ait yolların cazip hale getirilmesi için global feromon güncellemesi tanımına göre feromon miktarı fazladan artırılır.

Turlarını tamamlayan bütün karıncaların geçtikleri yollar ise lokal feromon güncellemesi tanımına göre artırılır. Böylece karıncalar sonraki iterasyonlarda turların tamamlandığı yolları daha çok tercih edeceklerdir. Bir önceki çözümden bulunan en kısa tura ait yollar ise diğer yollardan daha cazip olacaktır.

Tekrarlı olarak her iterasyonda en kısa tur hep aynı tur çıkması durumunda ise belirli bir zaman (iterasyon) sonra karıncaların büyük bir kısmı bu yolu tercih etmeye başlayacaktır ve böylece en uygun muhtemel çözüm bulunmuş olacaktır.

Problemde 20 il için 50 iterasyon sonundaki deneme çözümünde en kısa yol 2529 km bulunmuştur. Denemede kullanılan iller şunlardır; Ankara, Düzce, Kocaeli, Uşak, Aydın, Edirne, Kütahya, Yalova, Bolu, Eskişehir, Manisa, Balıkesir, İstanbul, Muğla, Çanakkale, İzmir, Sakarya, Denizli, Kırklareli, Tekirdağ.

İlden ile geçişler ise Tablo 2'deki gibidir.

Tablo 2: İlden İle Geçişler

Yalova	İstanbul
İstanbul	Kırklareli
Kırklareli	Edirne
Edirne	Tekirdağ
Tekirdağ	Çanakkale
Çanakkale	Balıkesir
Balıkesir	Manisa
Manisa	İzmir
İzmir	Aydın
Aydın	Muğla
Muğla	Denizli
Denizli	Uşak
Uşak	Kütahya
Kütahya	Eskişehir
Eskişehir	Ankara
Ankara	Bolu
Bolu	Düzce
Düzce	Sakarya
Sakarya	Kocaeli
Kocaeli	Yalova

4. SONUÇ

Araç rotalama problemlerinde çok fazla düğümün olduğu (örneğin 600 düğüm gibi) durumlarda deneme çözümleri inanılmaz fazla olmaktadır. Bütün yolları deneyerek sezgisel olmayan bir yöntemle çözüm bulmak olası görünmemektedir. Optimal çözümü şu anki var olan teknolojik imkanlarla çözmek imkansız hale gelmektedir ya da daha az düğüm söz konusu bile olsa çözüm kısa sürede bulunamayacağı için yine pratik olmaktan çıkmaktadır. Bu sebeplerden dolayı sezgisel bir yöntemle bir çözüm bulmak çok daha kolaydır. Bu çözüm

en iyi çözüm olmasa bile çok muhtemelen en iyiye yakın bir çözüm olacaktır. Karınca kolonisi algoritması bu konuda en başarılı algoritmalarından biridir.

KAYNAKÇA

- Bajpai A., Yadav R., (2015). “Ant Colony Optimization (ACO) For The Traveling Salesman Problem (TSP) Using Partitioning”, *International Journal Of Scientific&Technology Research*, Volume 4, Issue 09, September.
- Brezina I., Jr.,Cicková Z., (2011). “Solving the Travelling Salesman Problem Using the Ant Colony Optimization”, *Management Information Systems*, Vol. 6, No. 4, pp. 010-014.
- Cordon, O.,Herrera, F. &Stützle, T. (2002). “A Review On The Ant Colony Optimization Metaheuristic: Basis, Models And Trends”, *Mathware&Soft Computing*, 9.
- Demircioğlu M., (2009). Araç Rotalama Probleminin Sezgisel Bir Yaklaşım İle Çözülmesi Üzerine Bir Uygulama. (Doktora Tezi). Çukurova Üniversitesi Sosyal Bilimler Enstitüsü, 163s.
- Dikmen H., Dikmen H., Elbir A., Ekşi Z. & Çelik F. (2014). “Gezgin Satıcı Probleminin Karınca Kolonisi Ve Genetik Algoritmalarla Eniyilemesi ve Karşılaştırılması”, *Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 18(1), 8-13.
- Dorigo M.,Dicaro G. &Gambardella L. M., (1999). *Ant Algorithms For Discrete Optimization*. Artificial Life 5, 137-172.
- Dorigo, M. &Gambardella L-M., (1996). “Ant Colonies For The Traveling Salesman Problem”, *Tr/Irídia/3*, Universite’ Libre De Bruxelles Belgium.
- Dorigo, M. &Gambardella, L.M., (1997). “Ant Colonies For The Travelling Salesman Problem”, *Biosystems*, 43 (2), 73–81.
- Keskintürk T. & Söyler H., (2006). Global Karınca Kolonisi Optimizasyonu. *Gazi Üniversitesi Mimarlık Ve Mühendislik Dergisi*, 21,689-698.
- Kuzu S., Önay O., Şen U., Tunçer M., Yıldırım B. F., & Keskintürk T.,(2014), “Gezgin Satıcı Problemlerinin Metasezgiseller ile Çözümü”, *Istanbul University Journal of the School of Business*, Vol 43, No 1, 1-27.
- Lizárraga E., Castillo O., & Soria J., (2013). “A Method to Solve the Traveling Salesman Problem Using Ant Colony Optimization Variants with Ant Set Partitioning”, *Part of the Studies in Computational Intelligence book series (SCI, volume 451)*.
- Mohsen A. M., (2016). “Annealing Ant Colony Optimization with Mutation Operator for Solving TSP”, *Computational Intelligence and Neu roscience* (2016).
- Mondal R., Hossain S. & Saha S.. (2013). “An Approach For Solving Travelling Salesman Problem”. *International Journal Of Applied Operational Research*, 3(22), 15-26, 2013.
- Özdemir Y.S. (2008). Karınca Kolonisi Algoritması İle Bilgisayar Ağlarının Topolojik En İyilenmesi. (Yüksek Lisans Tezi). Başkent Üniversitesi Fen Bilimleri Enstitüsü, 25.
- Shang G., Lei Z., Fengting Z., & Chunxian Z.(2007). “Solving Traveling Salesman Problem by Ant Colony Optimization Algorithm with Association Rule”, *Natural Computation*.
- Serin S. (2009). Karınca Kolonisi Yaklaşımıyla Karayolu Üstyapı Rutin Bakım Çalışmalarının Planlanması. (Yüksek Lisans Tezi). Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü, 99.