Süleyman Demirel Üniversitesi
Fen Bilimleri Enstitüsü Dergisi

Suleyman Demirel University
Journal of Natural andAppliedScience

18(1), 8-13, 2014

Gezgin Satıcı Probleminin Karınca Kolonisi ve Genetik Algoritmalarla Eniyilemesi ve Karşılaştırılması

Hasan DİKMEN[1], Hüseyin DİKMEN[1], Ahmet ELBİR[2], Ziya EKŞİ*[2], Fatih ÇELİK[3]
[1]Sakarya Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği ABD, 54187, Sakarya
[2]Yıldız Teknik Üniversitesi, Elektrik-Elektronik Fakültesi, Bilgisayar Mühendisliği Bölümü, 34220, İstanbul
[2]Sakarya Üniversitesi, Teknik Eğitim Fakültesi, Elektronik ve Bilgisayar Eğitimi Bölümü, 54187, Sakarya
[3]Sakarya Üniversitesi, Teknoloji Fakültesi, Bilgisayar Mühendisliği Bölümü, 54187, Sakarya

(Alınış Tarihi: 23.09.2013, Kabul Tarihi: 14.02.2014)

Anahtar Kelimeler
Gezgin Satıcı Problemi
Karınca Kolonisi Algoritması,
Genetik Algoritmalar,
Rota Planlama

Özet: Bu çalışmada, rota planlama problemlerinden olan gezgin satıcı probleminin (GSP) çözümünü gerçekleştirmek için yapay zekâ tekniklerinden olan karınca kolonisi ve genetik algoritmaların performansları karşılaştırılmıştır. Türkiye haritası üzerinde gerçekleştirilen çalışmada en iyi rotanın planlanması hedeflenmiştir. Her iki algoritmanın rota mesafesi yönünden başarım performansı ve bu rotayı hesaplama süresi incelenmiştir. Uygulamanın gerçekleştirilmesi ve deneysel sonuçların gözlemlenmesi için C# tabanlı bir arayüz tasarlanmıştır. Uygulama sonucunda karınca kolonisi algoritmasının hem rota mesafesi hem de başarım süresi yönünden genetik algoritmalara göre daha üstün olduğu gözlemlenmiştir.

Optimization and Comparison of Travelling Salesman Problem Using Ant Colony and Genetic Algorithms

Keywords
Travelling Salesman Problem
Ant Colony Algorithm,
Genetic Algorithm,
Route Planning

Abstract: Inthisstudy, performance of Ant Colony and Genetic Algorithms that is one of artificial intelligence techniques were compared in order to solve traveling salesman problem that is one of route-planning problem. The best route over the map of Turkey is aimed to find. Performance of the sealgorithms in terms of route distance and route prediction time were investigated. A C#-based interface was designed for implementation of the application and monitoring of the experimental results. As a result of the application, it is observed that Ant Colony algorithm has more high performance than genetic algorithm in terms of both route distance and performance time.

1. Giriş

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 (Cevre vd., 2007). GSP'nin formülasyonu kolay olmasına rağmen çözümü zordur. Şehir sayısının artmasına bağlı olarak problemin çözümünün karmaşıklığı da artmaktadır (Ateş, 2012). Çözümünün zor ve uygulama alanının fazla olması nedeniyle çözümüne 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. Bu çalışmalarda var olan verilere göre en iyi çözümün bulunması hedeflenmiştir. 2011 yılında Brezina ve Cickova (Brezina ve Cickova, 2011) tarafından karınca kolonisi optimizasyonu kullanılarak GSP'nin çözümüne yönelik bir çalışma gerçekleştirilmiştir. Cevre ve arkadaşları (Cevre vd., 2007) tarafından yapılan çalışmada ise genetik algoritmalar kullanılarak GSP optimize edilerek internet üzerinde görselleştirilmesi sağlanmıştır. Alaykıran ve Engin (Alaykıran ve Engin, 2005) ise karınca kolonileri metasezgiseli kullanılarak GSP üzerine bir uygulama gerçekleştirmişlerdir. Gerçekleştirilen çalışmada, Visual Basic programlama dili kullanılarak karınca kolonileri metasezgiseli için uygun parametre

* İlgili yazar: ziyae@sakarya.edu.tr

değerleri belirlenip GSP çözümü gerçekleştirilmiştir. Shang ve arkadaşları (Shang vd., 2007) tarafından yapılan çalışmada ise birliktelik kuralları tabanlı karınca koloni optimizasyonu algoritması geliştirilerek GSP'ye çözüm geliştirilmiştir. Geliştirilen algoritma standart algoritmalarla karşılaştırılarak başarım değeri hesaplanmıştır. Özkan ve arkadaşları (Özkan vd., 2008) tarafından gerçekleştirilen çalışmada ise GSP üzerinde genetik algoritmaları yerel arama sezgileri ile birleştiren melez bir çözüm geliştirilmiştir. Java ortamında geliştirilen uygulama da Türkiye'nin 81 ilini dolaşan en kısa güzergâh tespit edilmiştir. Çolak (Çolak, 2010) tarafından yapılan çalışmada ise genetik algoritmalardan faydalanılarak Adana ilinde gıda sektöründe faaliyet gösteren bir firma için rota planlamasına yönelik bir uygulama yazılımı gerçekleştirilmiştir.

Gerçekleştirilen bu çalışmada ise, C# tabanlı bir yazılım geliştirilmiştir. Gerçekleştirilen yazılımda optimum çözüm bulmak için genetik algoritmalar ve karınca kolonisi algoritmalarından faydalanılmıştır. Ayrıca bulunan en uygun güzergâh grafiklerden faydalanılarak gösterilmiştir. Ayrıca yararlanılan algoritmaların performans ve zaman kriterleri esas alınarak başarımları değerlendirilmiştir.

Makalenin diğer bölümleri şu şekilde organize edilmiştir. 2. bölümde makalede kullanılan algoritmalara ait teorik bilgilere yer verilmiştir. Gerçekleştirilen çalışmaya ait ayrıntılı bilgiler 3. bölümde sunulmuştur. 4. bölümde ise uygulamaya ait performans analizine ve çalışma sonuçlarına yer verilmiştir.

2. Temel Algoritmalar

Makalenin bu bölümünde gerçekleştirilen uygulamada kullanılan genetik algoritma ve karınca kolonisi algoritmasından bahsedilmiştir.

2.1. Genetik Algoritmalar (GA)

İlk olarak 1970'li yıllarda John Holland ve arkadaşlarının yaptığı çalışmalarda ortaya çıkan Genetik Algoritmalar (GA), Yapay Zekâ'nın hızlı gelişen alanlarındandır (Holland, 1975). Temel olarak, bir fonksiyonun global minimum yada maksimum değerlerine en yakın sonuçlar bulma yöntemidir. Evrimsel süreçlerin kullanılarak, rastgele seçilen ebeveyn bireylerden yeni nesiller meydana gelmesi ve bu yeni nesillerin en iyilerinin ebeveyn olarak kalıp en kötülerinin elenmesi esasına dayanır. Yeni nesil oluşturma sürecinin tekrarlanması, belirli bir eşik değeri miktarınca ya da bir koşula bağlı olarak gerçekleştirilir. (Cevrevd, 2007).

Genetik algoritmalar probleme tek bir çözüm aramak yerine bir çözüm kümesi üzerinde çalışırlar. Genetik algoritmalarda olası çözümlerin tamamı hemen oluşturulmaz. En iyi ya da en iyiye yakın çözüme, olası çözümlerin bir bölümü üzerinden gidilir. Genetik algoritmalar deterministik yöntemler değildir, olasılık üzerine kurulu (stokastik) yöntemlerdir. Genetik algoritmalar paralel çalışırlar yani aynı anda problem uzayının birden çok noktası üzerinde arama yaparlar.

2.1.1. Genetik Algoritma Bileşenleri

Gen: Genetik biliminde belirli bir özelliği şifreleyen DNA blokları anlamına gelen genler, genetik algoritmada ise olası çözümün temelini oluşturan bir veya birkaç bitlik bloklardır.
Birey: Genetik biliminde topluluğu oluşturan canlılardan her birine birey adı verilir. Bireyler genlerinde bulunan özellikleri taşırlar. Genetik algoritmada ise her birey problem için olası bir çözümü temsil eder. Genetik algoritmalarda bireyler kromozom olarak da adlandırılmaktadır.
Topluluk (Popülasyon): Topluluk genetik biliminde de genetik algoritmada da bireylerden oluşan kümeye verilen addır. Topluluk kavramı genetik biliminde canlılardan oluşan, genetik algoritmada ise olası çözümlerden oluşan bir topluluğu temsil etmektedir.
Uygunluk (Fitness): Genetik biliminde bireyin çevre şartlarına ne kadar dayanıklı olduğunu belirten uygunluk değeri genetik algoritmada ise çözümün kalitesini belirler ve uygunluk fonksiyonu kullanılarak hesaplanır. Örneğin GSP'de amaç en düşük maliyetli turu bulmak olduğu için uygunluk değeri bireyin içerdiği şehir sıralamasının toplam maliyetinin düşüklüğüyle doğru orantılı olmalıdır. Böylece en düşük maliyetli tur en kaliteli çözümü temsil etmiş olur.
Çaprazlama: Genetik biliminde üreme sırasında ebeveynlerin genlerinin belirli bölümlerinin birleşerek yeni bir kromozom oluşturması anlamına gelen çaprazlama genetik algoritmada da benzer biçimde olası çözümlerin farklı bölümlerini birleştirerek daha iyi çözümlere ulaşmak amacıyla gerçekleştirilen bir işlemdir.
Mutasyon: Genetik biliminde ebeveyn ile çocuk arasında yalnızca nükleotid değişimi olması anlamına gelen mutasyon, genetik algoritmada çeşitliliği sağlamak amacıyla gerçekleştirilen bir yöntemdir. Mutasyonlar gerçekleştirilerek meydana gelen yeni nesillerin yerel minimum ya da maksimum noktalarına takılması önlenir. Böylece bir sonraki nesiller global minimum yada maksimum noktasına yönlendirilebilir. Bu işlem kromozomun bir ya da daha çok bölgesinde yapılabilen $0 \rightarrow 1$, $1 \rightarrow 0$ gibi bir dönüşümdür.

2.1.2. Genetik Algoritma Parametreleri

Birey Sayısı: Oluşturulan popülasyonda kaç adet birey bulunacağını belirleyen parametredir. Birey sayısı problemin türü ve kapsamı göz önünde bulundurularak dikkatlice belirlenmelidir. Problemin

çözümünde birey sayısı belirlenirken belirli bir kriter bulunmamaktadır. Birey sayısının az olması yerel minimum ya da maksimuma takılmasına neden olurken, çok olması da en iyi çözüme ulaşma süresini artıracaktır. Genetik algoritmalar sezgisel yöntemler olduğu için birey sayısının belirlenmesi de sezgisel olarak yapılmaktadır.

Birey Seçim Yöntemi: Bireylerin çaprazlama için hangi yönteme göre seçileceğini belirleyen parametredir. Genetik algoritmalarda en çok kullanılan çaprazlama yöntemleri Rulet Seçilim Yöntemi, Sıralama Seçilim Yöntemi ve Turnuva Seçilim Yöntemi'dir.

Çaprazlama Olasılığı: Yeni nesil oluşturulurken seçilen bireylerin hangi sıklıkta çaprazlama işlemine tabi tutulacağını belirleyen olasılıktır. Bu olasılık doğrultusunda seçilen bireyler ya çaprazlanır ya da doğrudan bir sonraki nesle aktarılır. Çaprazlama olasılığının düşük olması yeni neslin mevcut nesilden yeterince farklılaşmamasına dolayısıyla da kuşak farkının azalmasına neden olur. Bu da çözüme daha uzun sürede ulaşılmasına yol açar. Literatürde ideal çaprazlama olasılığı 0.7 ya da 0.75 olarak belirtilir (Karaboğa, 2004; Özkan, 2008).

Mutasyon Olasılığı: Yeni nesil oluşturulurken mutasyon işleminin hangi sıklıkta yapılacağını belirleyen olasılıktır. Bu olasılık doğrultusunda seçilen bireyler ya mutasyona tabi tutulur ya da doğrudan bir sonraki nesle aktarılır. Yüksek mutasyon olasılığı çözüm adaylarında aşırı rastgeleliğe yol açacağından optimum çözümden uzaklaşmamıza neden olur. Çok düşük mutasyon oranı ise popülasyon içerisinde yer alan çözüm adaylarının çeşitliliğini azaltacağından çözüme ulaşmayı geciktirir. Literatürde ideal mutasyon oranı 0.001 olarak belirtilir (Karaboğa, 2004; Özkan, 2008).

2.1.3. Genetik Algoritma Sözde Kodu

Adım 1: Başlangıç popülasyonu oluşturulur.
Adım2: Popülasyondaki bireylerin uygunluk değerleri hesaplanır.
Adım3: Popülasyondaki bireyler çaprazlama için birey seçim yöntemine göre seçilir.
Adım 4: Seçilen bireyler çaprazlama olasılığına göre çaprazlanarak yeni bireyler oluşturulur.
Adım 5: Yeni bireylere mutasyon olasılığına göre mutasyon uygulanır.
Adım 6: Yeni bireyler popülasyonun mevcut bireyleri olarak belirlenir.
Adım 7: Belirtilen nesil sayısına ulaşılana kadar veya hedef değere ulaşılana kadar Adım 2'ye gidilir.

2.2. Karınca Kolonisi Algoritması(KKA)

Temel ilkeleri ilk kez MarcoDorigo 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). Dorigo, karınca kolonilerinin davranışlarının matematiksel modelleri üzerine dayandırdığı 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).

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 bulduğu 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).

2.2.1. 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).

$$P_{i,j}^{\ell} = \frac{\left[\tau_{ij}\right]^{\alpha}\left[\eta_{ij}\right]^{\beta}}{\sum_{l \in N_i^{\ell}}\left[\tau_{il}\right]^{\alpha}\left[\eta_{il}\right]^{\beta}} \quad \text{, eğer } j \in N_i^{\ell} \qquad (1)$$

Burada,

$\boldsymbol{P_{ij}^{\ell}}\ell$ karıncasının i düğümünden j düğümüne geçme olasılığı
$\boldsymbol{\tau_{ij}}$ i ve j düğümleri arasındaki feromon değeri
$\boldsymbol{\eta_{ij}}$ i ve j düğümleri arasındaki sezgisel değeri
$\boldsymbol{\alpha}$ feromon katsayısı
$\boldsymbol{\beta}$ sezgisel katsayısı
$\boldsymbol{N}$ düğümler kümesi

2.2.2.Karınca Kolonisi Algoritması Parametreleri

Karınca Sayısı: Kolonide kaç tane karıncanın olacağını belirleyen parametredir.

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

2.2.3.Karınca Kolonisi Algoritması Sözde Kodu

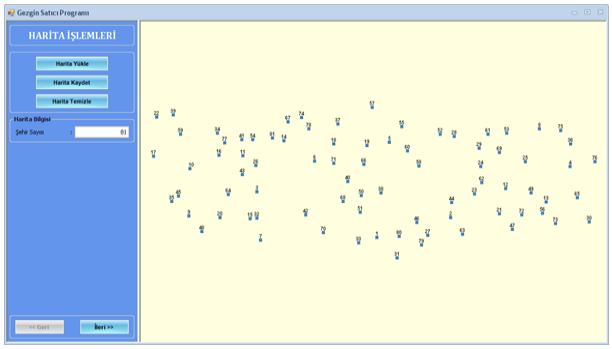
Adım 1: Karıncalar oluşturularak şehirlere rastgele olarak yerleştirilir.
Adım 2: Her bir karınca belirtilen alfa ve beta değerlerine göre şehirleri dolaşarak rotaları oluştururlar.
Adım 3: Her bir karıncanın rota mesafesi hesaplanır.
Adım 4: Karıncaların rota mesafelerine göre yollardaki feromon değerleri artırılır.
Adım 5: Feromon buharlaşma değerine göre bütün yollardaki feromon değeri azaltılır.
Adım 6: Belirtilen nesil sayısına ulaşılana kadar veya hedef değere ulaşılana kadar Adım 2'ye gidilir.

3. C# Tabanlı GSP Aracı

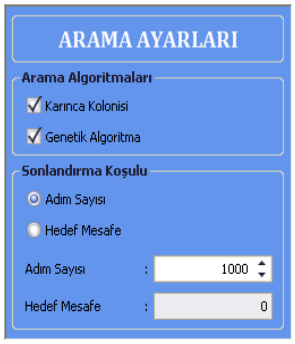
Makalenin bu bölümünde gerçekleştirilen uygulamaya dair bilgiler verilmektedir. Gerçekleştirilen uygulama Visual Studio 2010 platformunda C# dili kullanılarak gezgin satıcı probleminin görsel olarak gözlemlenmesi için bir arayüz tasarlanmıştır. Tasarlanan arayüz vasıtasıyla değişik haritalar kullanılarak karınca kolonisi ve genetik algoritma ile problemin çözümü gerçekleştirilebilmektedir. Uygulama kısmında Türkiye haritası üzerinde işlemler gerçekleştirilmiştir. Ayrıca bulunan sonuçlar daha önceki çalışmalarla karşılaştırılmıştır.

Şekil 1'de görülen arayüzde "Harita Yükle" butonu kullanılarak Türkiye haritası programa yüklenmektedir. Türkiye haritasında yer alan şehirler plaka numaraları ile gösterilmektedir. Ayrıca arama bilgileri ve adım değerlerinin girilmesi işlemi gerçekleştirilebilmektedir.

Şekil 2 ve 3'te kullanılan algoritmalara ait arama bilgilerinin ve adım değerlerinin girilmesi gibi seçenekler yer almaktadır. Opsiyonel olarak kullanıcı istediği algoritmaya göre arama yapabilmekte ve kullanılan algoritmaya ait sonuçları gözlemleyebilmektedir.



Şekil 1. Türkiye haritasının programa yüklenmesi



Şekil 2. Arama ayarlarının belirlenmesi



Şekil 3. Parametredeğerlerin in belirlenmesi

Gerekli parametre değerleri girildikten sonra ise arama işlemi başlatılmaktadır. Arama boyunca arama ile ilgili bilgiler ve bulunan rotalar anlık olarak gösterilmektedir. Arama sonunda elde edilen bu bilgiler rapor halinde kaydedilebilmektedir.

4. Deneysel Sonuçlar

Makalenin bu bölümünde gerçekleştirilen uygulamaya ait deneysel sonuçlar yer almaktadır. Karınca kolonisi ve genetik algoritmaların performans analizleri gerçekleştirilirken performans kriteri olarak "başarım süresi" ve "başarım performansı" kriterleri esas alınmıştır. Her iki algoritmada 1000 adımda çalıştırılmıştır. Türkiye haritası üzerinde bulunan en uygun rota mesafesi ve bu mesafeyi bulma süresi hesaplanmıştır.

Tablo 1 ve 2'de her iki algoritma içinde elde edilen sonuçlarda kullanılan parametre değerleri yer almaktadır.

Tablo 1. Karınca kolonisi parametreleri

Parametreler	Değerler
Karınca Sayısı	300
Alfa Değeri	1.0
Beta Değeri	3.5
Feromon Buharlaşma Değeri	0.8

Tablo 2. Genetik Algoritma Parametreleri

Parametreler	Değerler
Birey Sayısı	10000
Birey Seçim Yöntemi	Sıralama
Çaprazlama Olasılığı	0.9
Mutasyon Olasılığı	0.1

Tablo 3 ve 4'te ise kullanılan algoritmalar için arama sonuçları yer almaktadır.

Tablo 3. KKA için arama sonuçları

Arama Bilgileri	Sonuçlar
Adım Değeri	1000
Bulunan Rota Mesafesi	9966.00 km
Arama Süresi	00:15:10
Bulunan Rota	15-32-64-3-43-26-11-16-77-41-54-81-14-6-18-71-40-66-19-5-60-58-38-50-68-42-70-51-33-1-80-31-79-27-46-2-63-47-21-72-56-73-30-65-13-49-12-23-44-62-24-29-69-25-4-76-36-75-8-53-61-28-52-55-57-37-78-74-67-34-59-39-22-17-10-45-35-9-48-20-7-15

Tablo 4. GA için arama sonuçları

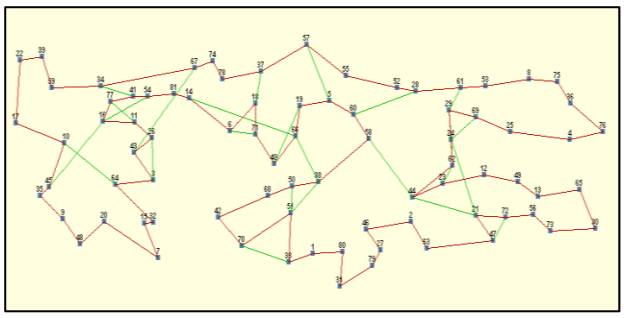
Arama Bilgileri	Sonuçlar
Adım Değeri	1000
Bulunan Rota Mesafesi	11739.00 km
Arama Süresi	00:15:10
Bulunan Rota	15-32-64-3-43-26-11-16-77-41-54-81-14-6-18-71-40-66-19-5-60-58-38-50-68-42-70-51-33-1-80-31-79-27-46-2-63-47-21-72-56-73-30-65-13-49-12-23-44-62-24-29-69-25-4-76-36-75-8-53-61-28-52-55-57-37-78-74-67-34-59-39-22-17-10-45-35-9-48-20-7-15

Tablo 3 ve 4'te yer alan sonuçlara göre karınca kolonisi algoritması ve genetik algoritma 1000 adım (iterasyon) çalıştırılmıştır. 1000 adım karınca kolonisi için 15 dakika 10 saniye, genetik algoritma için ise 23 dakika 38 saniye sürmüştür. Genetik algoritmanın 1000 iterasyonu daha uzun sürede gerçekleştirmesinin nedeni popülasyondaki birey sayısının çok fazla olmasıdır.

1000 adım sonunda karınca kolonisinin bulduğu en kısa rota mesafesi 9966 km, genetik algoritmanın ise 11739 km olduğu görülmektedir. Bu durumda karınca kolonisinin genetik algoritmaya göre daha başarılı olduğu söylenebilir. Türkiye haritası için bilinen en kısa rota mesafesi 9954 km olduğundan karınca kolonisinin minimum sonuca çok yaklaştığı söylenebilir (Söyler ve Keskintürk, 2007).

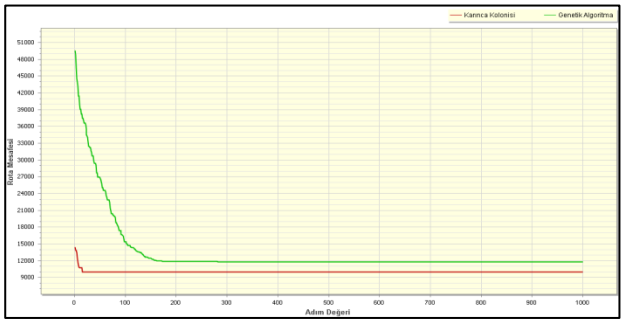
Şekil 4'de arama sonucunda elde edilen rotalar Türkiye haritası üzerinde gösterilmiştir. Kırmızı ile gösterilen rota karınca kolonisi algoritmasına, yeşil ile gösterilen rota ise genetik algoritmaya aittir.

Şehirler arasındaki mesafeler T.C. Karayolları Genel Müdürlüğü (KGM) internet sitesinden kilometre cinsinden alınmıştır (KGM, 2012). Bu mesafe değerleri bir matris formunda programa verilerek karınca kolonisi algoritmasının ve genetik algoritmanın bu mesafe tablosuna göre arama yapması sağlanmıştır. Dolayısıyla algoritmaların bulmuş olduğu mesafeler gerçek rota mesafeleridir. Ancak algoritmaların bulmuş olduğu rotalar programdaki harita üzerinde gösterilirken şehirler arasındaki bağlantılar kuş uçuşu yani direkt olarak gösterilmiştir.



Şekil 4. Arama sonucunda bulunan rotaların harita üzerinde gösterimi

Tasarlanan arayüz ile adım sayısına göre bulunan rota mesafesi rapor olarak gösterilebilmekte ve kaydedilebilmektedir. Şekil 5'de buna ait sonuç gösterilmiştir.

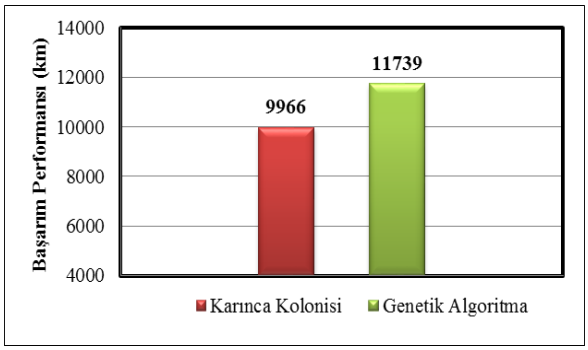


Şekil 5. Karınca kolonisi ve genetik algoritmanın arama grafiği

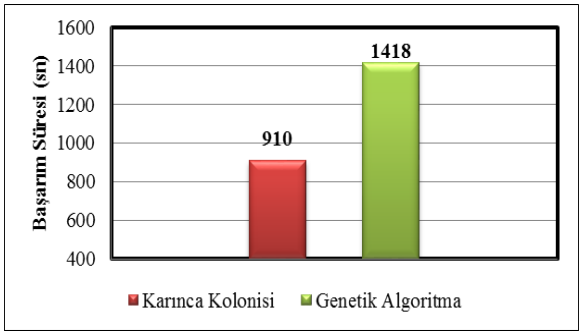
Şekil 5'de karınca kolonisi algoritmasının grafiği kırmızı renkle, genetik algoritmanın grafiği ise yeşil renkle gösterilmiştir. Grafikincelendiğinde karınca kolonisi algoritması, genetik algoritmaya göre daha kısa bir rota mesafesinde başlıyor. Bunun nedeni olarak başlangıçta karınca kolonisi algoritmasının kendisine en yakın şehirlere uğrayarak turunu tamamlaması, genetik algoritmanın ise başlangıçta tamamen rastsal olarak rotalar oluşturması gösterilebilir. Grafikten karınca kolonisi algoritmasının yaklaşık olarak 25. iterasyonda,

genetik algoritmanın ise yaklaşık olarak 375. iterasyonda yerel minimuma ulaştığı görülmektedir.

Şekil 6 ve 7'de ise karınca kolonisi algoritmasının ve genetik algoritmanın başarım performansı ve başarım süresi kıyaslanmaktadır.



Şekil 6. KKA ve GA'nın rota mesafesi yönünden başarım performansı



Şekil 7. KKA ve GA'nın arama süresi yönünden başarım performansı

Her iki algoritmanın 1000 adımda başarım süresi kriterine göre analizi gerçekleştirildiğinde karınca kolonisi algoritması genetik algoritmaya göre daha kısa sürede daha başarılı bir performans gerçekleştirmiştir.

Gerçekleştirilen bu uygulamada deneysel sonuçlara göre karınca kolonisi algoritmasının Türkiye haritası üzerinde en iyi rota mesafesini genetik algoritmaya göre daha kısa sürede bulduğu gözlemlenmiştir.

Kaynaklar

Alaykıran K., Engin O., 2005. Karınca Kolonileri Metasezgiseli ve Gezgin Satıcı Problemleri Üzerinde Bir Uygulaması. Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi, Cilt 20, No 1, 69-76.

Ateş E., 2012. Karınca Kolonisi Optimizasyonu Algoritmaları ile Gezgin Satıcı Probleminin Çözümü ve 3 Boyutlu Benzetimi. Ege Üniversitesi Bilgisayar Mühendisliği, Lisans Tezi, 49s, İzmir.

Brezina I.,Cickova Z., 2011. Solving the Travelling Salesman Problem Using the Ant Colony Optimization. Management Information Systems, Vol. 6, No 4, 010-014.

Cevre U., Özkan B., Uğur A. 2007. Gezgin Satıcı Probleminin Genetik Algoritmalarla Eniyilemesi ve Etkileşimli Olarak İnternet Üzerinde Görselleştirilmesi. XII. Türkiye'de İnternet Konferansı, Ankara.

Çolak S., 2010. Genetik Algoritmalar Yardımı ile Gezgin Satıcı Probleminin Çözümü Üzerine Bir Uygulama. Ç.Ü. Sosyal Bilimler Enstitüsü Dergisi, Cilt 19, Sayı 3, 423-438.

Demircioğlu M., 2009. Araç Rotalama Probleminin Sezgisel Bir Yaklaşım İle Çözümlenmesi Üzerine Bir Uygulama. Çukurova Üniversitesi Sosyal Bilimler Enstitüsü, Doktora Tezi, 163s.

Dorigo M., DiCaro G., Gambardella L. M., 1999. Ant Algorithms for Discrete Optimization. Artificial Life 5, 137-172.

Holland J., 1975. Adaptation In Natural and Artificial Systems. University of Michigan Press, AnnArbor.

Karaboğa, D., 2004. Yapay Zeka Optimizasyon Algoritmalari. AtlasYayınları, 199s.

Karayolları Genel Müdürlüğü, 2012. http://www.kgm.gov.tr/Sayfalar/KGM/SiteTr/Root/Uzakliklar.aspx (Erişim Tarihi: 23.04.2013).

Özdemir Y.S., 2008. Karınca Kolonisi Algoritması İle Bilgisayar Ağlarının Topolojik En İyilenmesi. Başkent Üniversitesi Fen Bilimleri Enstitüsü. Yüksek Lisans Tezi, 25s.

Özkan B., 2008. Dinamik Gezgin Satıcı Probleminin Çözümü İçin Bir Eniyileme Kütüphanesinin Tasarımı ve Görsel Yazılım Geliştirme Ortamı ile Birlikte Gerçekleştirimi. Ege Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği, Yüksek Lisans, 43s.

Özkan B., Cevre U., Uğur A., 2008. Melez Bir Eniyileme Yöntemi ile Rota Planlama. Akademik Bilişim, Çanakkale.

Serin S., 2009. Karınca Kolonisi Yaklaşımıyla Karayolu Üstyapı Rutin Bakım Çalışmalarının Planlanması. Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 99s.

Shang G.,Lei Z., Fengting Z., Chunxian Z., 2007. Solving Traveling Salesman Problem by Ant Colony Optimization Algorithm with Association Rule. Third International Conference on Natural Computation (ICNC 2007).

Söyler, H.,Keskintürk T., 2007. Karınca Kolonisi Algoritması ile Gezen Satıcı Probleminin Çözümü. 8. Türkiye Ekonometri ve İstatistik Kongresi, İnönü Üniversitesi, Malatya.