

Gezgin Satıcı Probleminin Genetik Algoritmalarla Eniyilemesi ve Etkileşimli Olarak İnternet Üzerinde Görselleştirilmesi

Utku Cevre¹, Barış Özkan¹, Aybars Uğur²

¹ Ege Üniversitesi, Bilgisayar Mühendisliği Bölümü, İzmir

² Ege Üniversitesi, Bilgisayar Mühendisliği Bölümü, İzmir

utkucevre@mail.ege.edu.tr, barisozkan@mail.ege.edu.tr, aybars.ugur@ege.edu.tr

Özet: Gezgin Satıcı Problemi (GSP), üzerinde yoğun olarak çalışılan kombinasyonel eniyileme problemlerinin başında gelmektedir. Çözümü için birçok sezgisel algoritma geliştirilmiştir. Çalışmamızda, iki boyutlu Gezgin Satıcı Problemini genetik algoritmalar kullanarak çözmek için, İnternet üzerinde etkileşimli olarak kullanılabilen Java tabanlı bir araç geliştirdik. Değişik parametre değerleri için elde edilen deneysel sonuçları sunduk ve karşılaştırdık. Simülasyon aracı olarak geliştirdiğimiz esnek ve etkileşimli programımızın, araştırmacılar ve diğer kullanıcılar açısından yararlarını kısaca belirttik.

Anahtar Sözcükler: Gezgin Satıcı Problemi, Genetik Algoritmalar, Eniyileme, Simülasyon Aracı, Web Tabanlı Yazılım Geliştirme, Nesneye Dayalı Programlama.

Abstract: Traveling Salesman Problem (TSP) is one of the extensively studied combinatorial optimization problems. A variety of heuristic algorithms are available for solving TSP. In this paper, we developed a Java-based software, which can be used interactively on the Web, to solve Planar TSP using genetic algorithms. We presented and compared the results, which are obtained for different parameter values. As a result, advantages of this flexible and interactive program as a simulation tool for researchers and other users are briefly discussed.

1. Giriş

Gezgin Satıcı Problemi (Traveling Salesman Problem) veya kısaca GSP (TSP), aralarındaki uzaklıklar bilinen N adet noktanın (şehir, parça veya düğüm gibi) her birisinden yalnız bir kez geçen en kısa veya en az maliyetli turun bulunmasını hedefleyen bir problemdir. Ayrık ve Kombinasyonel Eniyileme (Combinatorial Optimization) problemlerinin kapsamına girer. Kaba kuvvet arama yöntemi ile, doğrudan tüm permütasyonların toplam yol uzunluklarının hesaplanması ve en küçüğünün bulunması şeklinde çözülebilmekle birlikte, N 'in büyük değerleri için permütasyon sayısı $N!$ büyük değerlere ulaşacağından, bu işlem çok uzun zaman almaktadır. Bu nedenle hızlı ve etkin

çözüm için birçok yöntem geliştirilmiştir. Aşamalı Geliştirme (Iterative Improvement), Doğrusal Programlamaya dayalı Dallan ve Sınırla (Branch-and-Bound) ve bu yöntemin Düzlem Kesme yöntemleriyle melezleştirilmesinden oluşan Dallan ve Kes (Branch-and-Cut) kesin çözüm veren yöntemlerdendir. Sezgisel veya yaklaşıma dayalı çözümler veren yöntemler ise, makul bir sürede en iyi çözüme yakın sonuçlara ulaşılmasını sağlarlar. Sezgisel yöntemlerin en önemlileri, genetik algoritmalar, benzetimli tavlama (simulated annealing), yasak arama (Tabu search) ve karınca sistemi tabanlı algoritmalarıdır.

d-boyutlu uzayda verilen noktalar için en kısa turu bulmayı sağlayan GSP'ler içinde en yay-

gın olarak kullanılan, iki boyutlu olandır. Öklit GSP'nde düğümler R2 içindedir ve uzaklık ℓ_2 normundadır [1]. Bazı durumlarda da A şehrinde B şehrine gitmenin maliyeti ile, B şehrinde A şehrine gitmenin maliyeti farklıdır. Bazı şehirler arasında tek yön yolların olması, gidiş ve geliş yönlerindeki yollardaki trafik sıkışıklığının getireceği süre farklılıkları gibi durumlar dikkate alındığında Simetrik Olmayan GSP devreye girer.

1950'lerden günümüze kadar, büyük boyutlu GSP'ler için hızlı bir şekilde iyi çözümler veren birçok sezgisel yöntem üzerinde çalışılmıştır [3] [4] [5]. Genetik algoritmalar, yasadışı arama, benzetimli tavlama, yapay sinir ağları gibi birçok arama yöntemi üzerine bir derleme, [6]'da bulunabilir.

Bilgisayar mühendisliği müfredatında algoritmalar, veri yapıları, ayrık matematik, optimizasyon, yapay zeka gibi birçok derste bu probleme ve çözüm yöntemlerine değişik bakış açıları ile değinilmektedir. Çözümü ise, yol ve rota planlama (uçak, otobüs, dağıtım kamyonları, bilgisayar ağları, posta taşıyıcılar, vb.), iş planlama, baskı devre kartlarındaki delgi işlemlerinin belirlenmesi gibi birçok alanda kullanılmaktadır. Bu çalışmada Java platformu üzerinde, özellikle araştırmacıların web üzerinden kolaylıkla erişerek istenen sayıda nokta ve değişik nokta konumları için Gezgin Satıcı Problemi optimum turunu bulmalarını sağlayan etkileşimli ve grafik destekli esnek bir yazılım geliştirilmiştir. Altyapıda eniyileme yöntemi olarak Genetik Algoritmaların kullanılması tercih edilmiştir.

Makalenin ikinci bölümünde genetik algoritmalar incelenerek temel kavramlarla ilgili bilgi verilmiş, üçüncü bölümde gezgin satıcı probleminin genetik algoritmalarla eniyilenmesi konusunda daha önce yapılmış çalışmalar ele alınmıştır. Dördüncü bölümde çözüm için kullanılan yöntemler anlatılırken, beşinci bölümde geliştirilen uygulama tanıtılmıştır. Altıncı

bölümde deneysel sonuçlar değerlendirilmiş, son bölümde ise ileriki çalışmalar hakkında bilgi verilmiş ve önerilerde bulunulmuştur.

2. Genetik Algoritmalar

Evrimsel ve genetik algoritmalar, yapay zekanın hızlı gelişen alanlarından. Özellikle kombinasyonel eniyileme problemlerine yaklaşıp iyi sonuçlar bulmayı hedefleyen arama yöntemleridir. Problemin çözümünde kullanılacak rastgele seçilmiş bir çözüm kümesi oluşturabilmek için evrimsel mekanizmaların kullanıldığı bu yöntemlerin temel mantığı topluluğun nesilden nesle geçmesi sırasında kötü çözümlerin yok olmasına ve iyi çözümlerden daha iyi çözümlere ulaşılmasına dayanır.

Genetik algoritmalarda kromozomlar, problem için olası çözümleri temsil ederler. Topluluk (popülasyon) kromozomlardan oluşan kümedir. Uygunluk değeri ise, çözümün kalitesini belirler ve uygunluk fonksiyonu kullanılarak hesaplanır. Çaprazlama ve mutasyon işlemleri ise olası çözümleri temsil eden kromozomlara genetik bilimindekine benzer bir biçimde uygulanır. Yeni nesiller, seçilen bireylerin çaprazlama ve mutasyon gibi genetik operatörlerden geçirilmesi ile elde edilir. Bir problemin çözümünde genetik algoritmalar kullanıldığı takdirde izlenecek adımlar aşağıdaki gibi sıralanabilir:

1. [Başlat] N adet kromozom (probleme uygun çözümler) içeren topluluğu oluştur.
2. [Uygunluk] Her x kromozomu için $f(x)$ uygunluk değerini hesapla.
3. [Yeni Topluluk] Aşağıdaki adımları tekrarlayarak yeni popülasyonu oluştur.
 - a. [Seçilim] Topluluktan uygunluk değerlerini dikkate alarak (uygunluk değeri daha iyi olanların seçilme olasılığı yüksek olacak şekilde) iki kromozom seç.
 - b. [Çaprazlama] Belirli bir çaprazlama

olasılığıyla ebeveynlerden gelen kromozomları çaprazlayarak yeni birey oluştur. Çaprazlama yapılmazsa ebeveynlerden gelen kromozomları aynen bir sonraki nesle kopyala.

c. [Mutasyon] Yeni bireyi belirli bir olasılığa göre mutasyona uğrat.

d. [Ekleme] Oluşturulan bireyi yeni topluluğa ekle.

4. [Değiştir] Önceki topluluğu, yeni toplulukla değiştir.

5. [Test] Sonlandırma koşulu sağlandıysa mevcut topluluktaki en iyi çözümü döndür, sağlanmadıysa 2. adıma dön.

Kromozomların Kodlanması

Kromozomların kodlanması, genetik algoritmalar ile problem çözmenin ilk aşamasıdır. Kodlama yaklaşımı problem türüne göre farklılık gösterir. En sık kullanılan yöntemler ikili kodlama ve permutasyon kodlamadır. İkili kodlamada her kromozom 1 ve 0'lardan oluşan bir karakter dizisi şeklinde ifade edilir. Permutasyon kodlamada ise her kromozom, ilgili karakterin sıralamadaki pozisyonunu belirten sayılardan oluşan bir dizi ile ifade edilir. Permutasyon kodlama, genelde Gezgin Satıcı Problemi gibi sıralama problemlerinde kullanılır.

Seçilim

Yeni topluluğu oluşturmak için mevcut topluluktan çaprazlama ve mutasyon işlemine tabi tutulacak bireylerin seçilmesi gerekir. Teoriye göre iyi olan bireyler yaşamını sürdürmeli ve bu bireylerden yeni bireyler oluşturulmalıdır. Bu nedenle tüm seçim yöntemlerinde uygunluk değeri fazla olan bireylerin seçilme olasılığı daha yüksektir. En bilinen seçim yöntemleri Rulet Seçilimi, Turnuva Seçilimi ve Sıralı Seçilimdir.

Rulet Seçilimi: Topluluktaki tüm bireylerin uygunluk değerleri toplanır ve her bireyin seçilme olasılığı, uygunluk değerinin bu toplam

değere oranı kadardır. Uygunluk değerleri arasındaki farkların fazla olması durumunda rulet seçilimi hep aynı çözümler etrafında dolanma sorununa yol açabilir.

Sıralı Seçilim: En kötü uygunlukta olan kromozoma 1 değeri verilir, ondan daha iyi olana 2, daha iyisine 3 değeri verilerek devam edilir. Bu yöntemdeki amaç düşük uygunlukta kromozomlara da seçilme şansı tanımadır, ancak bu durum çözümün daha geç yakınsamasına neden olabilir.

Turnuva Seçilimi: Topluluk içerisinde rastgele k adet (3,5,7) birey alınır. Bu bireylerin içerisinde uygunluk değeri en iyi olan birey seçilir. Buradaki k değeri genellikle topluluğun büyüklüğüne göre değişir. Örneğin 10000 bireyden oluşan bir toplulukta k büyük bir değer olmalıdır.

Çaprazlama

Genetik Algoritmalarda çaprazlama işlemi, iyi çözümlerin farklı bölümlerini birleştirip daha iyi çözümler oluşturabilmek amacıyla kullanılır. Çaprazlamanın en kolay yolu rastgele bir çaprazlama noktası belirleyip, bu noktadan önceki bölümü ilk ebeveyninden, sonraki bölümü ise diğer ebeveyninden alarak yeni bir birey oluşturmaktır. Başlıca çaprazlama yöntemleri tek noktalı, iki noktalı ve aritmetik çaprazlamadır. Sıra mantığını koruyan bazı çaprazlama operatörleri ise Order Crossover [7], Modified Crossover [8], Partially Mapped Crossover [9], Cycle Crossover [10], 2-quick / 2-repair [11] şeklinde belirtilebilir.

Mutasyon

Bireyin bir sonraki nesle geçirilmesi sırasında kromozomu oluşturan karakter dizisinde yapılan rastgele değişikliğe mutasyon denir. Mutasyon, oluşan yeni çözümlerin önceki çözümü kopyalamasını önleyerek çeşitliliği sağlamak ve sonuçta daha hızlı ulaşmak amacıyla gerçekleştirilir. Mutasyon olasılığı çok

düşük (%0.01 gibi) tutulmalıdır. Yüksek mutasyon olasılığı uygun çözümleri de bozacak ve Genetik Algoritmanın, çalışması sırasında problemlerle karşılaşılmasına yol açacaktır. Displacement Mutation [12], Exchange Mutation [13], Insertion Mutation [14] [12], Simple Inversion Mutation [15] [16] ve Inversion Mutation [17] [18] [19] sıra tabanlı mutasyon operatörlerinden bazılarıdır.

Elitizm (Seçkincilik): Seçilim, çaprazlama ve mutasyon işlemleri sonrasında mevcut topluluktaki en iyi uygunluk değerine sahip bireyi yeni topluluğa aktarılabilir. Bunu önlemek için bu işlemlerden sonra, bir önceki topluluğun en iyi (elit) bir veya daha çok bireyi, yeni oluşturulan topluluğa doğrudan aktarılır. Buna seçkincilik adı verilir.

3. Önceki Çalışmalar

Gezgin Satıcı Problemi'nin (GSP) çözümü için, problemin ilk defa tartışılmaya başlandığı 1800'lerden beri, birçok çalışma yapılmıştır. Genetik algoritmalar ve yerel arama tekniklerini içeren yöntemler iyi sonuç vermektedirler. Sengoku ve Yoshihara'nın 1998'te yaptıkları çalışma [2] bu alandaki en önemli işlerden biridir, çünkü geliştirdikleri algoritma, mutasyon evresinde çok etkili olan 2opt [23] yöntemini kullanmaktadır. Bu çalışmada, çaprazlama işlemi için, olası en uzun ebeveyn dizisinin alt-turlarını elde eden ve Greedy Subtour Crossover (GXO - Açgözlü Alt-tur Çaprazlama) adını verdikleri bir yöntem önerdiler. Bu metot ile çözüm, yerel minimumlardan, benzetimli tavlama yöntemlerinin yapabileceğinden daha verimli bir şekilde kurtulabilmektedir. Ayrıca, doğal seçilim evresinde, en düşük uygunluk değerine sahip olan çözümleri elediler. Bu yaklaşım seçkinciliğe benzetilmektedir, ancak en yüksek uygunluk değerine sahip bireyin korunması yerine, en az uygun birey topluluktan çıkarılmaktadır.

2003'te Pullan tarafından yayınlanan bildiri [20] yerel eniyileme sezgileri ile fenotip genetik operatörlerin birleştirilerek GSP'ye uygulanmasını değerlendirmektedir. Yerel eniyileme sezgileri arama alanını küçültürken, fenotip genetik operatörler, geçersiz turların yaratılmasını engellemekte ve alt-optimal şemanın oluşturulmasına yardımcı olmaktadır.

2004'te Ray, Bandyopadhyay ve Pal [21] yeni bir bilgi temelli çoklu ters çevirme operatörü ve bir bilgi temelli komşuluk yer değiştirme operatörü önermişlerdir. Bilgi temelli çoklu ters çevirme operatörü doğal seçimden hemen önce, bilgi temelli komşuluk yer değiştirme operatörü ise çaprazlama ve mutasyon evrelerinin arasında kullanılmaktadır.

2005'te Takahashi [22] GSP'nin genetik algoritmalar yoluyla çözümü için Changing Crossover Operators (CXO - Değişken Çaprazlama Operatörleri) adını verdiği yeni bir çaprazlama yöntemini de içeren bir öneri yapmıştır. Bu yöntem sayesinde, halihazırdaki çaprazlama operatörü, herhangi bir anda uygun bir başka çaprazlama operatörü ile esnek olarak değiştirilebilmektedir.

4. Gezgin Satıcı Probleminin Genetik Algoritmalar ile Eniyilenmesi

Gezgin Satıcı Problemi yukarıda da bahsedildiği gibi sıra tabanlı bir eniyileme problemidir ve problemin çözümünde 1985 yılından bu yana genetik algoritmalarla yararlanılmaktadır. Biz de Gezgin Satıcı Problemini genetik algoritmalar ile eniyilerken aşağıdaki adımları izledik.

Kromozomların Kodlanması

Gezgin Satıcı Problemi, bir sıralama problemi olduğu için kromozomların kodlanmasında permutasyon kodlama yöntemi kullanılmıştır. Yukarıda da belirtildiği gibi permutasyon kodlamada her kromozom, ilgili karakterin

sıralamadaki pozisyonunu belirten sayılardan oluşan bir dizi ile ifade edilir. Örneğin Gezin Satıcı Problemi’nde bir kromozomu oluşturan karakter dizisi, şehirlerin hangi sıra ile dolaşılacağını belirler.

Topluluğun İlklenmesi

Topluluk, belirli sayıda (topluluk büyüklüğü kadar) kromozomun rastgele olarak oluşturulmasıyla ilklenmiştir. Topluluk büyüklüğü, kullanıcı tarafından girilebilir, girilmediği takdirde ise varsayılan topluluk büyüklüğü değeri kullanılır.

Doğal Seçilim

Problemin çözümünde bireyleri seçerken sıralı seçim yöntemini tercih ettik. Sıralı seçimde yukarıda da bahsettiğimiz gibi en kötü uygunlukta olan kromozoma 1, ondan daha iyi olana 2, daha iyisine 3, en iyi olana da n (topluluktaki kromozom sayısı) değeri verilir. Rulet tekerleği gibi diğer popüler yöntemleri tercih etmememizin nedeni, bu yöntemlerde uygunluk değerleri arasındaki farkların yüksek olması durumunda uygunluk değeri az olan bireylerin seçilme şanslarının çok düşük olmasıdır. Bu durum hep aynı çözümlerin seçilmesine dolayısıyla da çeşitliliğin azalmasına yol açar. Sıralı Seçilim ile Rulet Seçiliminin karşılaştırılması Tablo 1’de görülmektedir.

Uygunluk	Rulet Seçilimi	Sıralı Seçilim
16	16/20	3/6
3	3/20	2/6
1	1/20	1/6

Tablo 1. Rulet Seçilimi ve Sıralı Seçilimin Karşılaştırılması

Rulet tekerleği yöntemi, düşük uygunluk değerine sahip bireylere sadece 1/20 ve 3/20 gibi seçilme şansları tanırken, sıralı seçim yönteminde bu olasılıklar 1/6 ve 2/6’ya çıkmaktadır. Böylece her nesilde en iyi çözüm seçilerek çeşitliliğin azalması tehlikesinin önüne geçilmektedir.

Mutasyon

Mutasyon aşamasında, eniyileme benzeri bir yaklaşım kullandık. Bu yaklaşıma göre mutasyona uğrayan kromozomun uygunluk katsayısı artmadığı takdirde mutasyon işlemini gerçekleştirmedik. Böylece bir yandan çeşitlilik sağlanırken bir yandan da çözümlerin kalitesinin azalması engellenmiş oldu.

Çaprazlama

Çaprazlama aşamasında yerel minimumlardan kurtulma yeteneğinden dolayı Greedy Subtour Crossover (GXO) [2] yöntemini kullandık (şekil 1). Gezin Satıcı Probleminde turun belli kısımları için en iyi alt turu içeren çözümler yerel minimumlara neden olabilmektedir. Ancak GXO yöntemi, algoritmanın bu yerel minimumlara takılmasını engelleyebilmektedir.



Şekil 1. GXO Yönteminin Gerçekleştirimi

Yöntemi uygularken öncelikle kromozomun güzergah listesinden rastgele bir şehir seçtik. Seçilen şehrin şekil 1’deki gibi C şehri olduğunu varsayalım. Bu durumda C şehrinin konumu her iki ebeveynde de bulunur ve C şehri yeni bireye eklenir. İlk ebeveyn, seçilen şehrin bulunduğu konumdan sola doğru ikincisi ise sağa doğru dolaşılır. Bu dolaşma sırasında C şehrinde önceki (kromozomda solunda bulunan) şehirler yeni bireyde C şehrinin soluna, sonraki şehirler ise sağına eklenir. Dolaşma herhangi bir yönde yeni bireye eklenmiş şehre ulaştığımızda sonlandırılır. Dolaşma her iki yönde de durduysa ve hala yeni birey tamamen dolmadıysa, kalan şehirler yeni bireye karışık sırayla eklenir. Biz orijinal GXO’dan farklı olarak bu kalan şehirleri ilgili kromozomdaki

sıralarına göre ekledik. Böylece çaprazlama aşamasında rastgelelik oluşmasını önlemiş olduk. Ayrıca yöntemi her çaprazlamada iki birey üretecek şekilde geliştirdik. İkinci bireyi aynı ebeveynleri ters yönlerde (ilk ebeveyni seçilen şehirden sağa, ikincisini sola doğru) dolaşarak elde ettik.

Tüm bu işlemler sırasında mevcut topluluğun en iyi uygunluk değerine sahip bireyini kaybetmemek için seçkincilik (elitizm) yöntemini kullandık. Bir başka deyişle topluluğun en iyi bireyini herhangi bir işleme tabi tutmadan yeni topluluğa aktardık.

5. İnternet Üzerinde Java Tabanlı Etkileşimli GSP Aracı

Bildiri kapsamında geliştirdiğimiz Java tabanlı etkileşimli GSP (TSP) aracı, optimum çözümü genetik algoritmalar yardımı ile bulmakta, ve verilen tüm şehirlerin (parçaların, noktaların) dolaşılabilceği en etkin güzergahı grafiklerle göstermektedir. Uygulama, “<http://yzgrafik.ege.edu.tr/~aybars/GSP>” adresinde, kullanıcıların erişimine de açılmıştır.

Java, Web Tabanlı Programlama ve Görselleştirme Yazılımı, uygulamanın World Wide Web (WWW) üzerinde yayımlamaya izin veren, Java Applet olarak geliştirdik. Applet, Java programlama dilinde yazılmış bir programdır ve bir HTML sayfası içeriğine, tıpkı bir resim gibi eklenebilir. Applet içeren bir sayfayı görüntülemek için Java teknolojisini destekleyen bir tarayıcı kullanıldığında, Applet’in kaynak kodu kullanıcının sistemine aktarılır ve tarayıcının Java Sanal Makinesi (JVM - Java Virtual Machine) tarafından işletilir [24].

Geliştirdiğimiz uygulamanın kullanıcı arayüzünde Gezgin Satıcı Problemi (GSP) için bulunan çözümlerin gösterilebilmesi için Java 2D API’sini kullandık. Java 2D API’si ileri seviye iki boyutlu grafikler ve resimleme için

yazılmış bir sınıflar kümesidir. Doğru çizimi, metin ve resimleri ayrıntılı tek bir model ile çevrelemektedir. API, resim birleştirme ve alfa kanalı resimleri için kapsamlı destek, isabetli renk uzayı tanımı ve dönüşümü için sınıflar ve zengin bir görüntüye dayalı resimleme operatörü kümesi sağlamaktadır.

Java 2D, yüksek kalitede baskı yapabilmek, özel renkleri karıştırabilmek, eski yazı tiplerini dönüşüme uğratarak yeni yazı tipleri yaratma, çoklu yazı tipi ve çoklu renk içeren metin dizileri çizdirebilme, yazı tiplerinin ana hatlarını çizdirebilme ve yazı tiplerini resimler ya da gradyan geçişli renklerle doldurabilmek gibi özelliklere de sahiptir. API, düşük seviyeli görüntü işleme ve renk modeli değişimini, etkileyici metin işlemlerini ve çeşitli testleri desteklemektedir.

Aracın Kullanımı

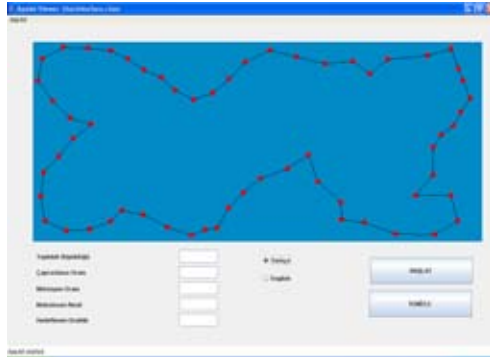
Java tabanlı etkileşimli GSP aracı, kullanım kolaylığına önem verilerek geliştirilmiştir. Kullanıcılar istedikleri noktalara fare ile tıklayarak şehirleri haritaya yerleştirebilirler. Tüm şehirleri yerleştirdikten sonra, topluluk büyüklüğü, çaprazlama oranı, mutasyon oranı, maksimum nesil sayısı ve hedeflenen uzaklık bilgilerini girebilirler. Bu parametreler isteğe bağlıdır. Eğer kullanıcılar tarafından bilgi girilmezse, uygulama varsayılan değerlerle işleyecektir. Genetik algoritmayı çalıştırmak için BAŞLAT düğmesine basılmalıdır.

Panelde bulunan şehirler (nokta) için optimum sonuç bulunduktan sonra, aşama aşama yeni nokta veya nokta kümeleri eklenerek güncellenmiş sonuçlar bulunabilir. Nokta yerleşimi tamamen değiştirilmek istendiğinde veya baştan yeni bir şehir kümesi eklenmek istendiğinde ise TEMİZLE düğmesine basılır.

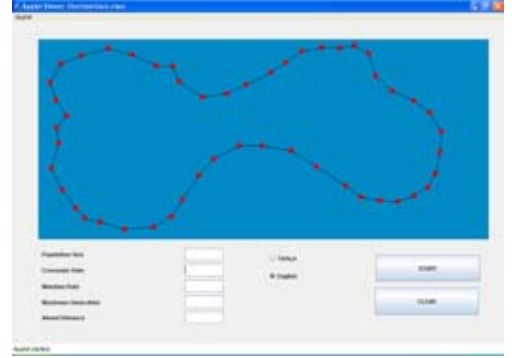
Topluluk büyüklüğü her yeni nesilde oluşturulacak olan çocuk sayısıdır. Bu parametrenin varsayılan değeri 200’dir. Toplam şehir sayısı

arttığında, çözüme erken nesillerde ulaşmak için topluluk büyüklüğünün yüksek tutulması gerekir. Çaprazlama oranı kromozom çiftlerinin çaprazlamaya girme ihtimalini belirtir. Çaprazlama oranının varsayılan değeri 0.75'tir. Buna göre her kromozom çifti $\frac{3}{4}$ ihtimalle çaprazlamaya girer. Mutasyon oranı kromozomların mutasyona uğraması ihtimalidir. Bu oranın varsayılan değeri 0.00001'dir. Buna göre her kromozom 1/100000 olasılıkla mutasyona uğrar. Mutasyon oranı, iyi çözümleri kaybetmeyi önlemek için düşük tutulmalıdır. Maksimum nesil sayısı, hedeflenen uzaklığa ulaşılmaması halinde genetik algoritmanın çalışmasının duracağı neslin numarasıdır. Bu parametrenin varsayılan değeri 120'dir. Hedeflenen uzaklık, tüm şehirlerin dolaşıldığı bir tur için, eşit veya altında olması arzu edilen toplam uzunluktur. Bu değere ulaşıldığında genetik algoritma yeni nesiller üretmeyi durdurur. Hedeflenen uzaklık için varsayılan değer 500'dür. Uygulama Türkçe (şekil 2) ve İngilizce (şekil 3) dil seçimine izin vermektedir. İngilizce tercih edilmesi halinde: topluluk büyüklüğü, population size; çaprazlama oranı, crossover rate; mutasyon oranı, mutation rate; maksimum nesil, maximum generation; hedeflenen uzaklık, aimed distance adını alacaktır. Bu durumda genetik algoritmayı çalıştırmak için START düğmesine basılmalıdır.

Ekran Görüntüleri:



Şekil 2. 60 şehir için bulunan çözümün



Şekil 3. İngilizce Menü Ekrana Görüntüsü

6. Deneysel Sonuçlar

Tablo 2, araç üzerinde gerçekleştirdiğimiz deneyler sonucu 30,40,50,60 ve 70 şehir için bulduğumuz ideal topluluk büyüklüğü ve ideal maksimum nesil değerlerini göstermektedir. Çözümün başarısında topluluk büyüklüğünün en önemli etken olduğu gözlenmektedir. Şehir sayısı ve buna bağlı olarak topluluk büyüklüğü belirli bir değer üstüne çıktıktan sonra, maksimum nesil parametresinin öneminin azaldığı ortaya çıkmaktadır. Bu nedenle maksimum nesil parametresinin belli bir noktadan sonra artırılmasına gerek kalmamaktadır.

Şehir Sayısı	İdeal Topluluk Büyüklüğü	İdeal Maksimum Nesil
30	75-100	60-80
40	100-150	80-100
50	250-350	100-120
60	450-550	100-120
70	1000+	120-150

Tablo 2. Deneysel Sonuçlar

Tablo, ayrıca Gezin Satıcı Problemi'nin karmaşıklığının şehir sayısına göre doğrusal biçimde değil, üstel olarak arttığını ortaya koymaktadır. Bu yüzden 30 şehirden 40 şehre geçişte parametreleri fazla arttırmaya ihtiyaç duyulmazken, 60 şehirden 70 şehre geçişte

parametrelerde büyük bir artışa gidilmesi gerekmektedir.

7. İleri Çalışmalar ve Öneriler

Geliştirdiğimiz web-tabanlı araç, kullanıcıların dolaşılacak tüm konumları etkileşimli olarak veya toplu halde bilgisayara girerek en uygun dolaşma sırasını ve toplam yol uzunluğunu (maliyeti) bulabilmesini sağlamaktadır. Değişik meslek gruplarından kişiler, ellerindeki parça toplama, parça yerleştirme ve dolaşmaya dayalı birçok problemi farklı nokta sayıları ve konumları için deneyerek en uygun sonucu alabilmektedirler. Bir fabrikadaki parçaların toplanmasından, bir kampüsteki posta dağıtımına, baskı devrelerdeki bileşenlerin yerleştirilmesinden, turlarda belirtilen tüm turistik merkezlere uğramaya kadar uyarlanabilmektedir.

Uygulamayı geliştirirken Java ortamını tercih etmemizin nedeni, Java'nın nesneye dayalı programlamayı desteklemesi ve platform bağımsızlığı sunarak, aracın çok sayıda insana ulaşmasını sağlamasıdır. Araç, değişik eniyileme yöntemi geliştiren araştırmacılar için, kendi sonuçlarını genetik algoritma sonuçları ile karşılaştırma olanağı da sunmaktadır.

Geliştirdiğimiz uygulamada Gezin Satıcı Problemi'nin çözümü için yalnızca genetik algoritmaları kullandık. Elde ettiğimiz deneysel sonuçları değerlendirdiğimizde, genetik algoritmaların, 2-opt gibi yerel arama sezgileriyle birlikte kullanıldığı melez yöntemlerin, daha etkin çözümlere ulaşmayı sağlayacağını tespit ettik. Aracın temel aldığı algoritmayı geliştirmek için yerel arama sezgileri dışında, Karınca Kolonisi Optimizasyonu (ACO - Ant Colony Optimization) gibi yeni eniyileme tekniklerinden de yararlanılabilir. Sadece genetik algoritmaların kullanıldığı durumlarda ise, problemin türüne göre, bazı genetik operatörlerde değişikliğe gidilebilir [22] ya da algoritmaya yeni genetik operatörler eklenebilir.

Bu tür web-tabanlı eniyileme yazılımlarının geliştirilmesi, araştırmacılara test olanağı sunması ve değişik meslek gruplarındaki kişilerin ellerindeki problemlere çözüm getirmesinin yanında, genetik algoritmalar gibi yapay zeka alanları ile daha çok kişinin de tanışmasını sağlamaktadır. Öğrenme sürecine yardım etmekte ve önemini anlamalarına katkıda bulunmaktadır. Çalışmamızın önemli katkılarından birisi de, endüstride değişik problemlerin çözümünde kullanılacak eğlenceli, kullanımı kolay, etkileşimli ve esnek yazılım araçlarının nesne yönelimli yaklaşımla geliştirilmesi için de bir örnek oluşturmasıdır.

Uygulamalarında genetik algoritma altyapısı kurmayı gerektiren durumlarda, hazır araçlardan yararlanmak isteyen geliştiriciler için, İnternet üzerinde pek çok hazır genetik algoritma kütüphanesi bulunmaktadır. Java GALib [25] bu hazır kütüphanelere önemli bir örnek olarak verilebilir.

Kaynaklar

- [1] Arora, S., 'Polynomial time approximation schemes for Euclidean TSP and other geometric problems', Proc. 37th Ann. Symp. Foundations of Computer Sci., IEEE Computer Soc., 1996, pp. 2-11.
- [2] Sengoku, H., Yoshihara, I., 'A Fast TSP Solution using Genetic Algorithm', 1998.
- [3] Gambardella, L.M., Dorigo, M., 'Solving Symmetric and Asymmetric TSPs by Ant Colonies', International Conference on Evolutionary Computation, 1996, pp 622-627.
- [4] Shinozawa, K., Uchiyama T., Shimohara, K., 'An approach for solving dynamic TSPs using neural networks', Neural Networks, 1991 IEEE International Joint Conference, vol.3, 1991, pp 2450 - 2454.

- [5] Tsujimura, Y., Gen, M., 'Entropy-based genetic algorithm for solving TSP', Knowledge-Based Intelligent Electronic Systems, Proceedings KES '98. 1998 Second International Conference on Volume 2, 1998, pp 285 - 290.
- [6] Johnson, D.S. and McGeoch, L.A., 'The traveling salesman problem: A case study in local optimization', In E.H.L. Aarts and J.K. Lenstra, editors, Local Search in Combinatorial Optimization, 1997, pp 215–310.
- [7] Syswerda, G., 'Schedule Optimization Using Genetic Algorithms', Handbook of Genetic Algorithms, New York NY, Van Nostrand Reinhold, 1991, pp 350-372.
- [8] Davis, L., 'Applying Adaptive Algorithms to Epistatic Domains', Proceedings of the International Joint Conference on Artificial Intelligence, 1985, pp 162-164.
- [9] Goldberg, D.E. and Lingle, J.R., 'Alleles, Loci and the TSP', Proceedings of the First International Conference on Genetic Algorithms and Their Applications, Lawrence Erlbaum, Hillsdale, New Jersey, 1985, pp 154-159.
- [10] Oliver, I.M., Smith, D.J. and Holland, J.R.C., 'A Study of Permutation Crossover Operators on the TSP', Genetic Algorithms and Their Applications, Proceedings of the Second International Conference, Lawrence Erlbaum, Hillsdale, New Jersey, 1987, pp 224-230.
- [11] Gorges-Schleuter, M., 'ASPARAGOS An Asynchronous Parallel Genetic Optimization Strategy', Proceedings of the Third International Conference on Genetic Algorithms, 1989, pp 422-427.
- [12] Michalewicz, Z., 'Genetic Algorithms + Data Structures = Evolution Programs', Berlin Germany, Springer Verlag, 1992.
- [13] Banzhaf, W., 'The Molecular Travelling Salesman', Biological Cybernetics, 64, 1990, pp 7-14.
- [14] Fogel, D.B., 'An Evolutionary Approach to the Travelling Salesman Problems', Biological Cybernetics, 60, 1988, pp 139-144.
- [15] Grefenstette, J., Gopal, R., Rosmaita, B. and Van Gucht, D., 'Genetic Algorithms for the TSP', in Grefenstette, J.J. (Ed.), Proceedings of the First International Conference on Genetic Algorithms and Their Applications, Lawrence Erlbaum, Hillsdale, New Jersey, 1985, pp 160-165.
- [16] Holland, J., 'Adaptation in Natural and Artificial Systems', Ann Arbor USA, University of Michigan, 1975.
- [17] Fogel, D.B., 'Empirical Estimation of the Computation Required to Discover Approximate Solutions to the Travelling Salesman Problem Using Evolutionary Programming', Proceedings of 2nd Annual Conference on Evolutionary Programming, 1993, pp 56-61.
- [18] Fogel, D.B., 'Applying Evolutionary Programming to Selected Travelling Salesman Problems', Cybernetics and Systems: An International Journal, 24, 1993, pp 27-36.
- [19] Larranaga, P., Kuijpers, C.M.H., Murga R.H., Inza, I. and Dizdarevic, S., 'Genetic algorithms for the travelling salesman problem: A review of representations and operators', Artificial Intelligence Review, 13, 1999, pp 129—170.
- [20] Pullan, W., 'Adapting the Genetic Algorithm to the Travelling Salesman Problem', Evolutionary Computation, CEC '03, The 2003 Congress on Volume 2, 2003.
- [21] Ray, S.S.; Bandyopadhyay, S.; Pal, S.K.,

'New Operators of Genetic Algorithms for Traveling Salesman Problem', Pattern Recognition, ICPR 2004, Proceedings of the 17th International Conference Volume 2, 2004.

[22] Takahashi, R., 'Solving the Traveling Salesman Problem through Genetic Algorithms with Changing Crossover Operators', Machine Learning and Applications, Proceedings, Fourth International Conference, 2005.

[23] Croes, G.A., 'A Method for Solving Traveling Salesman Problems', Operations Res. 6, 1958.

[24] Java Technology: <http://java.sun.com/>

[25] Java GALib, Genetic Algorithm Library: <http://sourceforge.net/projects/java-galib/>