

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

First Author¹, Second Author², Third Author³

¹Institution 1, İstanbul, Türkiye

²Institution 1, Kırşehir, Türkiye

³Institution 1, A City, A Country

¹mail@domain.com, ²mail2@domain.com, ³mail3@domain.com

Özet: Gezgin Satıcı Problemi (GSP), üzerinde yoğun olarak çalışılan kombinasyoneleniyileme problemlerinin başında gelmektedir. Çözümü için birçok sezgisel algoritma geliştirilmiştir. Çe- lışmamızda, iki boyutlu Gezgin Satıcı Problemini genetik algoritmalar kullanarak çözmek için, İnternet üzerinde etkileşimli olarak kullanılabilen Java tabanlı bir araç geliştirdik. Değişik pa- rametre 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 Introduction

Gezgin Satıcı Problemi (Traveling Salesman Problem) vya kısaca GSP (TSP), aralarındaki uzaklıklar bilinen N adet noktanın (şehir, par- ça 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 gi- rer. Kaba kuvvet arama yöntemi ile, doğrudan tüm permütasyonların toplam yoluzunlukları- nın 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. Aşa malı Geliştirme (Iterative Improvement), Doğ- rusal 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 yön- temlerin 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 yaygın olarak kullanılan, iki boyutlu olandır. Öklit GSP'nde düğümler R^2 içindedir ve uzaklık 12 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] [5] [8]. Genetik algoritmalar, yasak arama, benzetimli tavlama, yapay sinir ağları gibi birçok arama yöntemi üzerine bir derleme, [4] ile bulunabilir.

Bilgisayar mühendisliği müfredatında algorit- malar, veri yapıları, ayrık matematik, optimi- zasyon, yapay zeka gibi birçok derste bu probleme ve çözüm yöntemlerine değişik bakış açıları ile değinilmektedir. Çözümü ise, yol ve rota planlama (uçak, otobüs, dağıtım kamyon ları, bilgisayar ağları, posta taşıyıcılar, vb.), iş planlama, baskı devre kartlarındaki delgi işlemi sırasının belirlenmesi gibi birçok alan da kullanılmaktadır. Bu çalışmada Java plat formu 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 eniyi leme yöntemi olarak Genetik Algoritmaların kullanılması tercih edilmiştir. Makalenin ikinci bölümünde genetik algorit- malar 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 problem-

lerine yaklaşık 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. N adet kromozom (probleme uygun çözümler) içeren topluluğu oluştur.
2. Her x kromozomu için $f(x)$ uygunluk değerini hesapla.
3. Aşağıdaki adımları tekrarlayarak yeni popülasyonu oluştur.
 - (a) Topluluktan uygunluk değerlerini dikkate alarak (uygunluk değeri daha iyi olanların seçilme olasılığı yükselecek şekilde) iki kromozom seç.
 - (b) Belirli bir çaprazlama olasılığıyla ebeveynlerden gelen kromozomları çaprazlayarak yeni birey oluştur. Çaprazlama yapılmazsa ebeveynlerden gelen kromozomları aynen bir sonraki nesle kopyala.
 - (c) Yeni bireyi belirli bir olasılığa göre mutasyona uğrat.
 - (d) Oluşturulan bireyi yeni topluluğa ekle.
4. Önceki topluluğu, yeni toplulukla değiştir.
5. Sonlandırma koşulu sağlandıysa mevcut topluluktaki en iyi çözümü döndür, sağlanmadıysa 2. adıma dön

2.1 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.

2.2 Seçim

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.

2.2.1 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.

2.2.2 Sıralı Seçim

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.

2.2.3 Turnuva Seçilimi

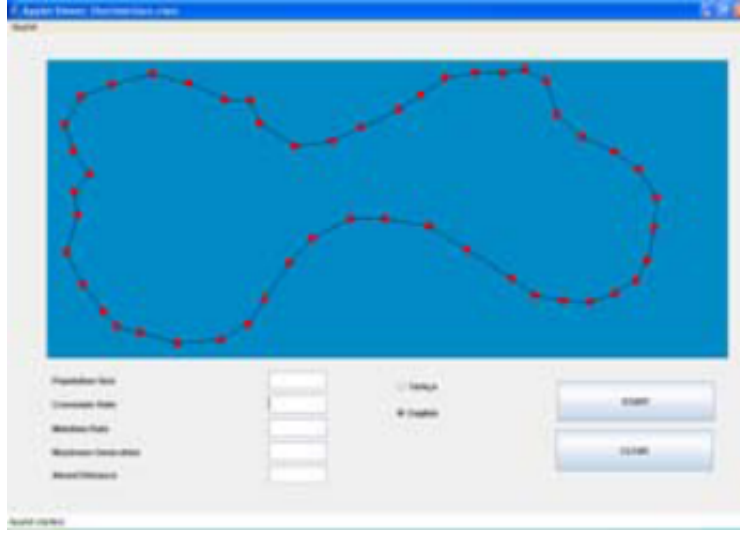
Topluluk içerisinde rastgele k adet (3,5,7) birey alınır. Bu bireylerin içerisinde uygunluk değeri en iyi olan birey seçilir. Buradaki k değeri genellikle topluluğun büyüklüğüne göre değişir. Örneğin 10000 bireyden oluşan bir toplulukta k büyük bir değer olmalıdır.

2.3 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 ebeveyniden, sonraki bölümü ise diğer ebeveyniden alarak yeni bir birey oluşturmaktır. Başlıca çaprazlama yöntemleri tek noktalı, iki noktalı ve aritmetik çaprazlamadır. Sıra mantığını koruyan bazı çaprazlama operatörleri ise Order Crossover [6], Modified Crossover [2], ... şeklinde belirtilebilir.

3 The Software

1 show a screen shot of the software used in the analysis.



Şekil 1: The software used in the analysis

If you want to use full-width images, first switch to one-column page and then load the figure like

4 Deneysel Sonuçlar

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 1: Table

Tablo 1, 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 değerin üstüne çıktıktan sonra, maksimum nesil parametresinin öneminin azaldığı ortaya çıkmaktadır. Bu nedenle maksimum nesil parametresinin belli bir noktadan sonra arttırılmasına gerek kalmamaktadır.

Tablo, ayrıca Gezgin 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.

5 İleriki 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 ala bilmektedirler. 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 Gezgin 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 [7] 'de vurgulandığı gibi ya da algoritmaya yeni genetik operatörler eklenebilir.

Kaynaklar

- [1] Sanjeev Arora. Polynomial time approximation schemes for euclidean tsp and other geometric problems. In *Proceedings of 37th Conference on Foundations of Computer Science*, pages 2–11. IEEE, 1996.
- [2] Lawrence Davis et al. Applying adaptive algorithms to epistatic domains. In *IJCAI*, volume 85, pages 162–164. Citeseer, 1985.
- [3] Luca Maria Gambardella and Marco Dorigo. Solving symmetric and asymmetric tsps by ant colonies. In *Proceedings of IEEE international conference on evolutionary computation*, pages 622–627. IEEE, 1996.
- [4] David S Johnson and Lyle A McGeoch. The traveling salesman problem: A case study in local optimization. *Local search in combinatorial optimization*, 1(1):215–310, 1997.
- [5] Kazuhiko Shinozawa, Tadasu Uchiyama, and Katsunori Shimohara. An approach for solving dynamic tsps using neural networks. In *[Proceedings] 1991 IEEE International Joint Conference on Neural Networks*, pages 2450–2454. IEEE, 1991.
- [6] Gilbert Syswerda. Schedule optimization using genetic algorithms. *Handbook of genetic algorithms*, 1991.
- [7] Ryouei Takahashi. Solving the traveling salesman problem through genetic algorithms with changing crossover operators. In *Fourth International Conference on Machine Learning and Applications (ICMLA'05)*, pages 6–pp. IEEE, 2005.
- [8] Yasuhiro Tsujimura and Mitsuo Gen. Entropy-based genetic algorithm for solving tsp. In *1998 Second International Conference. Knowledge-Based Intelligent Electronic Systems. Proceedings KES'98 (Cat. No. 98EX111)*, volume 2, pages 285–290. IEEE, 1998.