

GENETİK ALGORİTMALAR YARDIMI İLE GEZGİN SATICI PROBLEMİNİN ÇÖZÜMÜ ÜZERİNE BİR UYGULAMA

Yrd. Doç. Dr. Selçuk ÇOLAK
Çukurova Üniversitesi, İ.İ.B.F,
İşletme Bölümü, 01330, Balcalı/Adana
Tel: 322-3387254-60 (Dahili:294)
e-mail: scolak@cu.edu.tr

ÖZET

Gezgin (Seyyar) Satıcı Problemi'nde aralarındaki mesafeler bilinen n adet şehrin (nokta, düğüm, yerleşim yeri, müşteri, şube vb.) her birine yalnız bir kez uğranarak başlangıç noktasına geri dönülmesi esnasında kat edilen toplam yolun en kısa olduğu şehir sırasının bulunması hedeflenir. Dağıtım, planlama, lojistik gibi alanlar başta olmak üzere birçok sektörde geniş uygulama alanlarına sahip olan gezgin satıcı problemi, optimizasyon alanında araştırmacı ve akademisyenler tarafından üzerinde uzun yıllardır yoğun olarak çalışılan çözümü zor (NP-hard) bir problemdir. Bu çalışmada meta sezgisel bir yöntem olan genetik algoritmalar yardımı ile gezgin satıcı problemine çözüm aranmış ve geliştirilen algoritmanın uygulaması Adana ilinde gıda sektöründe faaliyet gösteren bir firma üzerinde gerçekleştirilmiştir. Firmanın güncel olarak kullandığı rotalar ile algoritma ile elde edilen rotalar karşılaştırılarak algoritmanın etkinliği gösterilmiştir.

Anahtar Kelimeler: *Gezgin satıcı problemi, genetik algoritmalar, meta sezgisel yöntemler.*

ABSTRACT

Traveling Salesman Problem aims to find the shortest route, among n cities (nodes, customers, branches etc.) with known distances between each city, where the salesman leaves a city, visits each of the cities exactly once and returns back to the starting point. The traveling salesman problem, one of the very important NP-hard problems in optimization, has wide application areas including distribution, planning, logistics, and it has been studied by researchers and academicians for decades. In this paper, a genetic algorithm has been applied to solve the traveling salesman problem and a real world application of the study has been implemented on a food delivery firm in Adana. The routes that the firm already uses and the routes that are found using genetic algorithms are compared to illustrate the efficiency of the algorithm.

Keywords: *Traveling salesman problem, genetic algorithms, metaheuristic methods.*

1. Giriş

Gezgin Satıcı Problemi (GSP) ilk olarak 1930'lu yılların başında Karl Menger (Menger, 1932) tarafından matematiksel olarak tanımlanmış olup araştırmacı ve akademisyenlerin üzerinde en çok çalıştığı kombinyonel optimizasyon problemleri (combinatorial optimization problems) arasında yer almaktadır. Bu problemin tanımlanması kolay olmasına rağmen çözümü çok zordur ve literatürde NP-zor (NP-hard) problemler arasında yer almaktadır (Garey ve Johnson, 1979). GSP dağıtım, planlama, lojistik, mikroçip imalatı gibi alanlar başta olmak üzere birçok sektörde geniş uygulamaya sahiptir. Günümüzde gelişen teknoloji ve küreselleşmenin getirmiş olduğu rekabetçi piyasalar, işletmeleri faaliyetlerini sürdürürken tecrübeye dayalı olmaktan çok bilimsel yöntemler kullanmaya zorlamaktadır. Bilimsel yaklaşımlarla elde edilen daha etkin yöntemler ile işletmeler maliyet tasarrufu ve rekabet avantajı sağlamaktadır.

Kombinyonel optimizasyon problemlerinin çözümünde sıklıkla kullanılan meta sezgisel yöntemlerden birisi de genetik algoritmalarıdır (GA). Genetik algoritmalar ilk kez Holland tarafından 1975 (Holland, 1975) yılında önerilmiş olup biyolojik sistemlerdeki evrim sürecine benzer bir şekilde çalışan bir arama ve optimizasyon yöntemidir. Birçok optimizasyon probleminin çözümünde kullanılmış olan genetik algoritmalar ile hızlı ve tatmin edici sonuçlar elde edilmiştir.

Bu çalışmada meta-sezgisel bir yöntem olan genetik algoritmalarla faydalanılarak gezgin satıcı problemine çözüm aranmıştır. Geliştirilen yöntem Visual Basic programlama dilinde kodlanmış ve bu yöntemin uygulaması Adana'da gıda sektöründe faaliyet gösteren bir firma üzerinde gerçekleştirilerek firmanın dağıtım alanında test edilmiştir. Elde edilen sonuçlar bu yöntemin kullanılması ile daha verimli rotaların belirlenerek zaman ve maliyet tasarrufu sağlanabileceğini göstermiştir.

Bu çalışmanın ikinci bölümünde gezgin satıcı probleminin matematiksel ifadesine yer verilecek ve problem ile ilgili bir literatür özeti yapılacaktır. Üçüncü bölümde problemin çözümünde kullanılan genetik algoritma ele alınacak ve adımları incelenecektir. Sonraki bölümde algoritmanın Adana'da bir dağıtım firması üzerinde uygulanmasına ilişkin detaylara yer verilecektir. Son bölümde ise elde edilen sonuçlar özetlenecek ve daha sonraki çalışmalar için önerilerde bulunulacaktır.

2. Gezgin Satıcı Problemi

Gezgin Satıcı Problemi'nde bir satış elemanının veya bir aracın belli bir nokta veya şehirden başlayıp sistemde yer alan diğer tüm nokta veya şehirlere yalnızca bir kez uğrayarak tekrar başlangıç noktası veya şehrine dönmesi sırasında yapmış olduğu toplam tur mesafesinin (veya maliyetinin) minimize edilmesi amaçlanır. Bu problemin

uygulanabilmesi için her bir noktadan diğer tüm noktalara olan uzaklıkların bilinmesi gerekmektedir. Toplam nokta sayısı n adet ise, birinci nokta için $(n - 1)$ adet, ikinci nokta için $(n - 2)$ adet, üçüncü nokta için $(n - 3)$ adet gidilebilecek nokta vardır. Burada n problemin boyutunu ifade etmektedir ve problem için olurlu tur sayısı $(n - 1)!$ adettir. Görüleceği üzere nokta sayısı az olduğu durumlarda kesin çözüme ulaşmak mümkün olabilirken, nokta sayısı arttıkça problem için alternatif çözüm sayısı hızla artmakta ve optimum çözüm bulunabilmesi için çok fazla zaman gerekmektedir. Bu sebeple sezgisel ve meta sezgisel yöntemler kullanılarak kısa sürede optimum veya optimuma yakın tatmin edici sonuçlar elde edilebilmektedir.

Gezgin satıcı problemi matematiksel olarak şu şekilde ifade edilebilir (Dantzig vd., 1954):

$$\text{Minimize: } z = \sum_{i=1}^n \sum_{j=1, i \neq j}^n x(i, j) d(i, j) \quad (1)$$

Kısıtlar:

$$\sum_{j=1, j \neq i}^n x(i, j) = 1, i = 1, 2, \dots, n \quad (2)$$

$$\sum_{i=1, i \neq j}^n x(i, j) = 1, j = 1, 2, \dots, n \quad (3)$$

$$\sum_{i, j \in S, i \neq j} x(i, j) \leq |S| - 1, \forall S \subset \{1, 2, \dots, n\} \quad (4)$$

$$x(i, j) = \begin{cases} 1, & i \text{ noktasından } j \text{ noktasına gidiliyor ise} \\ 0, & i \text{ noktasından } j \text{ noktasına gidilmiyor ise} \end{cases} \quad (5)$$

GSP'nin amaç fonksiyonu (1) numaralı eşitlikte verilmektedir. Burada $d(i, j)$ ifadesi i ve j noktaları arasındaki mesafeyi göstermektedir. $x(i, j)$ ise i noktasından j noktasına gidilip gidilmediğini ifade etmektedir. (2) ve (3) numaralı eşitlikler her bir noktaya yalnız bir kez uğranacağını garanti altına almaya yöneliktir. (2) numaralı eşitliğe göre her noktadan sadece bir kez çıkılacak, (3) numaralı eşitliğe göre her noktaya yalnızca bir kez gidilecektir. (4) numaralı eşitlik ise oluşabilecek alt turlardan kurtulmaya yönelik olan alt tur eleme kısıtıdır. (5) numaralı eşitlikte $x(i, j)$ 'nin 1 olması i noktasından j noktasına gidildiğini, 0 olması ise gidilmediğini göstermektedir.

Literatürde GSP için önerilen kesin çözüm algoritmalarının bazıları dal ve sınır yöntemi, dinamik programlama ve dal-kesme algoritmaları (Croes, 1958; Roberts ve Flores, 1966; Bellmore ve Nemhauser, 1968; Miliotis, 1976; Johnson ve McGeoch, 1995;

Arora, 1998; Applegate vd., 2007) vb. şeklindedir. Kesin çözüm algoritmaları belli bir boyuta kadar olan problemler için başarılı bir şekilde çalışıyor olsa da GSP’de nokta sayısı arttıkça problemin boyutu ve buna bağlı olarak çözüm için gerekli olan bilgisayar işlem zamanı artacağından optimum sonuçlara ulaşmak imkansız bir hal almaktadır. Bu sebeple araştırmacılar daha çok sezgisel ve meta sezgisel yöntemler kullanmaya yönelmişlerdir. Bu yöntemler arasında kurucu ve geliştirici sezgisel yöntemler, K-opt, V-opt, yasaklı arama, tavlama benzetimi, genetik algoritmalar, karınca kolonileri, yapay sinir ağları vb. yer almaktadır (Kirkpatrick vd., 1983; Malek vd., 1989; Laporte, 1992; Fiechter, 1994; Dorigo vd., 1996; Merz ve Freisleben, 2001; Cochrane ve Beasley, 2003; Gao vd., 2006; Tao vd. 2007; Applegate vd., 2007; Jozefowicz vd., 2008). Bu yöntemler optimum sonucu garanti etmemekle birlikte, kısa çözüm süresi ile tatmin edici sonuçlar vermektedir.

3. Genetik Algoritmalar ve GSP

3.1. Genetik Algoritmalar

Genetik algoritmalar klasik yöntemlerle çözümü zor olan problemlere çözüm bulmak amacıyla biyolojik evrim sürecinin bilgisayar ortamında kodlandığı ve bu evrim süreci kullanılarak optimum ya da optimuma yakın sonuçların araştırıldığı bir arama yöntemi olarak tanımlanabilir. Bu yaklaşımın temelini Holland (1975) ortaya atmıştır.

Genetik algoritmaların kısa sürede çözüme ulaşması ve global çözüm bulmada etkin olması gibi avantajlarının yanı sıra yerel çözüm bulmada çok başarılı olmaması ve çözüm kalitesindeki artışın nispeten yavaş olması şeklinde dezavantajları da bulunmaktadır. Genetik algoritmalar birçok optimizasyon probleminin çözümünde kullanılmış ve iyi sonuçlar elde edilmiştir. GSP’ye uygulanmış olan genetik algoritmalar ile ilgili çalışmalar arasında Goldberg ve Lingle (1985), Chatterjee vd. (1996), Schmitt ve Amini (1998), Larranaga vd. (1999), Tsai vd. (2004), Takahashi vd. (2005), Gao vd. (2006), Cevre vd. (2007), Applegate vd. (2007) yer almaktadır.

Genetik algoritmaların ilk aşamasında incelenilen problemin parametreleri uygun şekilde kodlanarak kromozomlar oluşturulur. Daha sonra bu kromozomlardan oluşan bir başlangıç popülasyonu meydana getirilir. Popülasyonda yer alan her bir kromozom problemin çözümü için bir adaydır. Ardından popülasyondan bazı kromozomlar seçilir ve doğadaki genetik süreçlere dayanan “çaprazlama” ve “mutasyon” genetik operatörleri uygulanarak daha iyi kromozomlar elde edilmeye çalışılır. Ardı ardına gerçekleştirilen iterasyonlar ile yeni nesiller oluşturulur ve bu yeni nesillerin verdiği uygunluk değerleri popülasyondakilerle karşılaştırılır. Yeni nesillerde yer alan kromozomlar içinde popülasyondakilere göre daha iyi uygunluk değerine sahip olan kromozomlar popülasyona dahil edilirler. Bu sayede popülasyonda yer alan kromozomlar bir evrim süreci içinde kendilerini daha iyi kromozomlarla yenilerler. Bu süreç yeni nesil sayısı belirli bir sayıya ulaşınca kadar veya popülasyondaki uygunluk

değerlerinde belirli bir miktarda iyileşme olmayıncaya kadar devam eder (Goldberg, 1989; Jang, 1997; Gonzalez ve Fernandez, 2000).

3.2. Gezgin Satıcı Problemi için Kromozom Yapısı

Genetik algoritmalarda en çok kullanılan kodlama yöntemleri ikili kodlama ve permütasyon kodlamadır. İkili kodlamada her kromozom 0 ve 1'lerden oluşan bir dizi şeklinde ifade edilirken, permütasyon kodlamada her kromozom kendisini oluşturan karakterlerin bir sırasını göstermektedir. Gezgin satıcı probleminde kromozomların kodlanması permütasyon yöntemine göre yapılmaktadır. Şekil 1'deki örnek kodlama herhangi bir satıcının başlangıç noktasından çıkıp, gitmesi gereken 10 adet noktaya uğradıktan sonra tekrar başlangıç noktasına dönmesine ilişkin kromozom yapısını göstermektedir.

7	5	8	1	4	10	6	2	9	3
---	---	---	---	---	----	---	---	---	---

Şekil 1. GSP için Genetik Algoritma Kromozom Yapısı

3.3. Başlangıç Popülasyonunun Oluşturulması

Genetik algoritmanın ilk aşamasında yeni nesillerin türetileceği bir başlangıç popülasyonunun oluşturulması gerekmektedir. Klasik genetik algoritmalarda başlangıç popülasyonu genellikle rastgele olarak oluşturulur. GSP'de kromozom yapısı farklı olduğundan başlangıç popülasyonu üretilirken [1,n] aralığındaki kesikli düzgün dağılımdan yerine koymadan sırayla n tane gen seçilir ve bu genler birleştirilerek kromozom oluşturulur. Bu çalışmada klasik genetik algoritmalardan farklı olarak başlangıç popülasyonunun tamamını rastgele oluşturmak yerine bazı sezgisel yöntemler (en yakın komşuluk veya tasarruf algoritması gibi) kullanılarak belirli sayıda çözüm elde edilmiş ve bu çözümler başlangıç popülasyonuna dahil edilmiştir. Bu kromozomlar ile elde edilen başlangıç popülasyonu sezgisel yöntemlerle elde edildiğinden, tamamı rastgele kromozomlardan oluşan başlangıç popülasyonuna göre daha iyi olması beklenmektedir.

3.4. GSP için Uygunluk Fonksiyonu

Başlangıç popülasyonu oluşturulduktan sonra yapılacak ilk işlem bu popülasyonda yer alan kromozomların uygunluk değerlerinin hesaplanmasıdır. Gezgin satıcı problemi için uygunluk değeri, ziyaret edilecek olan noktaların kromozomdaki sıraya uygun olarak mesafelerinin toplanması ile elde edilir. Burada en iyi kromozom en düşük toplam mesafeye sahip kromozomdur. Çaprazlama işlemi yapıp yeni kromozomlar elde edildiğinde, her bir kromozomun uygunluk değeri de benzer şekilde hesaplanabilir.

3.5. Seçim Süreci

Yeni kromozomların oluşturulabilmesi için başlangıç popülasyonundan iki ebeveynin seçilmesi ve bu ebeveynlerin çaprazlanarak yeni bireylerin (çocukların) elde edilmesi gerekmektedir. Popülasyonda yer alan kromozomlardan hangilerinin ebeveyn olarak seçileceği ile ilgili çeşitli yöntemler mevcuttur. Bu yöntemlerden en çok kullanılanları rulet tekeri, sıralama ve turnuva şeklindedir.

Rulet tekeri yönteminde popülasyondaki tüm bireylerin uygunluk değerleri toplanır. Burada her bir kromozomun seçilme şansı, kendi uygunluk değerinin tüm bireylerin uygunluk değerlerinin toplamına oranı kadardır.

Sıralama yönteminde popülasyondaki bireyler, uygunluk değerlerine göre küçükten büyüğe sıralanır. Bu sıraya göre her bir bireye sıra numarası verilir ve bu numaralara rulet tekeri yöntemi uygulanır. Bu sayede bütün kromozomlara seçilme şansı verilir. Bu yöntemde en iyi kromozomların seçilme şansı diğerlerinden çok farklı olmadığı için çözüme yakınsama daha yavaştır.

Turnuva yönteminde ise popülasyondaki kromozomlardan belirli bir sayıda birey rastgele olarak seçilir ve bunlardan uygunluk değeri en iyi olan kromozom ilk ebeveyn olarak alınır. Benzer şekilde bu işlem tekrar uygulanarak ikinci ebeveyn seçilir.

3.6. Çaprazlama

Çaprazlama genetik algoritmalarındaki en önemli parametrelerden bir tanesidir. Seçim süreci sonunda elde edilen ebeveynler çaprazlamaya tabi tutularak yeni bireyler oluşturulur. Burada amaç ebeveyn kromozom genlerinden uygunluk değeri daha yüksek olan çocuk kromozomlar meydana getirmeye çalışmaktır. Literatürde birçok çaprazlama yöntemi mevcuttur. Bunlar arasında en kolayı tek noktalı çaprazlamadır. Bu yöntemde seçilmiş olan iki ebeveyn için ortak bir çaprazlama noktası belirlenir. Birinci çocuk çaprazlama noktasına kadar olan genleri olduğu gibi birinci ebeveyninden alır. Diğer genler ise ikinci ebeveyndeki sıraya uygun olarak oluşturulur. Benzer şekilde ikinci çocuk çaprazlama noktasına kadar olan genleri ikinci ebeveyninden, diğer genleri ise birinci ebeveyninden uygun sıra ile alır. Şekil 2’de yer alan örnekte çaprazlama noktası olarak üçüncü gen belirlenmiş ve birinci çocuk ilk üç genini birinci ebeveyninden, geri kalan genlerini ikinci ebeveyninden almıştır.

E1 = **5 2 7 1 4 9 3 6 8** E2 = 6 2 8 4 1 5 9 3 7

Ç1 = **5 2 7 6 8 4 1 9 3** Ç2 = 6 2 8 **5 7 1 4 9 3**

Şekil 2. Tek Noktalı Çaprazlama Örneği

İki noktalı çaprazlama da tek noktalı çaprazlamaya benzemektedir. Bu yöntemde rastgele olarak iki çaprazlama noktası belirlenir. Birinci çocuk ilk çaprazlama noktasına kadar olan genleri birinci ebeveynden olduğu gibi alır. İkinci çaprazlama noktasına kadar olan genler ikinci ebeveyndeki sıraya uygun olarak alınır ve kalan genler tekrar birinci ebeveynden elde edilir.

Gezgin satıcı probleminin çözümünde kullanılan diğer çaprazlama yöntemleri Pozisyona Dayalı Çaprazlama (Murata vd. 1996), Kısmi Planlı Çaprazlama (Goldberg, 1989), Sıraya Dayalı Çaprazlama (Syswerda, 1991) vb. şeklindedir.

3.7. Mutasyon

Mutasyon popülasyonda genetik çeşitliliği sağlamak ve korumak için kullanılan bir operatördür. Mutasyon ile çaprazlama sonucu oluşan çocuğun genlerinden bir ya da bir kaç rastgele olarak değiştirilir ve yeni kromozomlar elde edilir. Gezgin satıcı probleminde mutasyon rastgele seçilen iki genin yer değiştirmesi şeklinde yapılır. Bu operatör tüm çocuklara uygulanmayıp, üretilen çocukların belirli bir yüzdesine uygulanmaktadır. Bu yüzdeye mutasyon oranı veya mutasyon olasılığı adı verilir. Mutasyon olasılığı genellikle (0.01 gibi) düşük tutulmaktadır. Bu nedenle mutasyon etkileri kromozomlarda az görülmektedir. Mutasyon sırasında kromozomdaki gen sayısı değişmez, sabit kalır. Çaprazlama sonucu oluşan bir çocuğa uygulanan mutasyon örneği Şekil 3'te görülmektedir. Bu örnekte birinci çocuğun üçüncü ve altıncı geninin rastgele seçilmiş iki gen olduğunu varsayalım. Bu şekilde çocuğun kromozomunda üçüncü ve altıncı genler yer değiştirilerek çocuk mutasyona uğratılmıştır.

Ç1 = 5 2 8 6 7 4 1 9 3 → Mutasyona uğramış Ç1 = 5 2 4 6 7 8 1 9 3

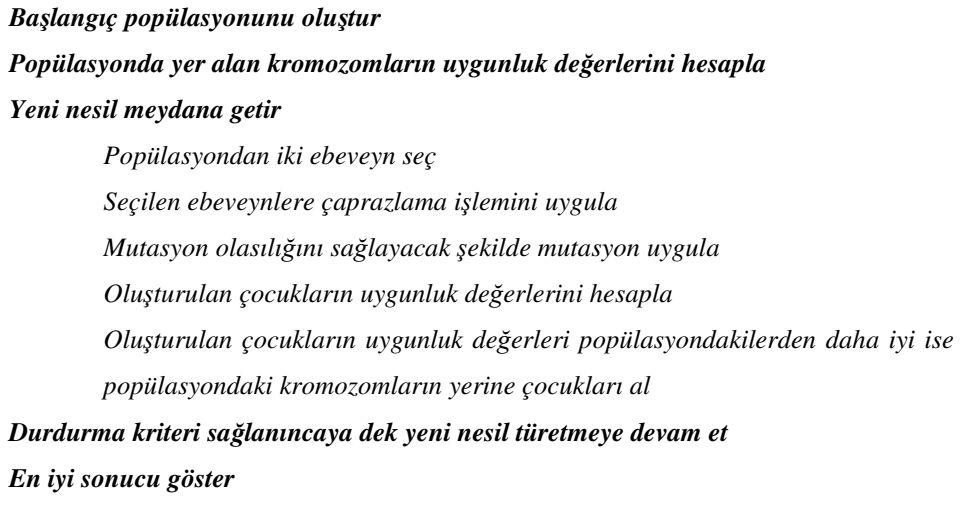
Şekil 3. Mutasyon Örneği

3.8. Yeni Neslin Oluşturulması

Genetik algoritmanın başlangıç popülasyonundan seçilen ebeveynlere çaprazlama ve mutasyon operatörleri uygulanarak yeni aday sonuçları olan çocuklar elde edilir. Öte yandan bu çocuklar popülasyonda yer alan kromozomlardan daha iyi uygunluk değerlerine sahip olabilirler. Bu aşamada elde edilen çocukların uygunluk değerleri ile popülasyondaki kromozomların uygunluk değerleri karşılaştırılır. Yeni çocukların uygunluk değeri popülasyondaki kromozomlarınkilerden daha iyi ise bu kromozomlar popülasyondan çıkarılır ve yerlerine yeni çocuklar alınır. Bu sayede yeni bir nesil oluşturulmuş olur. Bu işlem önceden belirlenen nesil sayısı kadar nesil elde edilinceye veya başka bir durdurma kriteri sağlanıncaya kadar tekrar edilir (Yeo ve Agyei, 1998).

3.9. Genetik Algoritma Yapay Kodu

Gezgin satıcı problemi için meydana getirilen genetik algoritmanın yapay kodu Şekil 4’de görülmektedir. Burada ilk olarak başlangıç popülasyonu oluşturulmaktadır. Ardından başlangıç popülasyonunda yer alan kromozomların uygunluk değerleri hesaplanmaktadır. Daha sonraki aşamada yeni nesillerin türetileceği iterasyonlara başlanmaktadır. İterasyonların her bir aşamasında popülasyondan iki ebeveyn seçilmekte ve seçilen bu ebeveynlere çaprazlama operatörlerinden birisi uygulanmaktadır. Çaprazlama sonucu elde edilen çocukların bazılarında mutasyon oranına bağlı olarak mutasyon operatörü uygulanır. Bu operatörler sonucunda meydana getirilen çocukların uygunluk değerleri hesaplanır. Hesaplanan bu değerler popülasyondakilerle karşılaştırılır ve elde edilen çocukların popülasyondaki kromozomlardan daha iyi bir birey olup olmadığı belirlenir. Eğer popülasyondaki kromozomlardan daha iyi çözüm veren bireyler elde edilmiş ise, popülasyondaki kötü bireyler çıkarılır ve yeni bireyler popülasyona eklenir. Bu işlem durdurma kriteri sağlanıncaya kadar tekrarlanır ve yeni nesiller türetilmiş olur. En son aşamada popülasyonda yer alan en iyi kromozom belirlenir ve problemin genetik algoritma çözümü olarak alınabilir.



Şekil 4. Genetik Algoritma Yapay Kodu

4. Yöntemin Bir Dağıtım Firmasında Uygulanması

Bu çalışmada gezgin satıcı probleminin çözümü için önerilen genetik algoritma ile ilgili kodlamalar Visual Basic 6.0 programlama dilinde tamamlanmış ve Adana ilinde gıda sektöründe faaliyet gösteren bir firmadan elde edilen gerçek veriler kullanılarak sonuçlar elde edilmiştir.

Uygulama yapılan firma Adana'da 1979 yılında kurulmuş ve çeşitli sektörlerde üretim ve dağıtım alanlarında faaliyetlerine devam etmektedir. Firma İstanbul Sanayi Odası raporuna göre Türkiye'deki ikinci 500 büyük sanayi kuruluşu içerisinde yer almaktadır. Bu çalışmada firmanın gıda alanında ticaretini yaptığı belirli bir ürün türü incelenmiştir. Firma Adana ilinde bulunan tesislerinde üretmiş olduğu bu ürünleri yine kendisine ait olan perakende satış mağazaları aracılığı ile müşterilerine sunmaktadır.

Firma Adana ilinde bölgesel olarak dağıtım yapmaktadır. Adana ili dağıtım amacıyla dört parçaya bölünmüştür. Bu parçaların her birinde tecrübeye dayalı olarak birer ziyaret sırası belirlenerek o bölgeye sevkiyat yapacak olan aracın rotası çıkarılmıştır. Firmadan bu bölgelere ait kullanılan rotalar, her bir rotadaki satış mağazaları ile ilgili enlem-boylam bilgileri ve her bir rotanın toplam tur mesafesi alınmıştır. Firmanın dağıtım yaptığı araçlar bölgelerde yer alan satış mağazalarına yapılacak olan toplam sevkiyatı karşılayabilecek niteliktedir. Firma dağıtım maliyetlerinin azaltılması amacıyla bölge bazında daha iyi rotaların belirlenmesini istemektedir. Firmanın bu problemi bu çalışmada incelenen gezgin satıcı problemi ile uyum göstermektedir. Bu sebeple bu çalışmada firmanın iki bölgesine ait veriler kullanılarak genetik algoritma yöntemi ile daha iyi rotaların elde edilmesi amaçlanmıştır.

Genetik algoritmanın bu problem üzerinde çalıştırılabilmesi için satış mağazaları arasındaki ikili uzaklıklara ihtiyaç vardır. Bu amaçla satış mağazaları enlem ve boylam bilgileri Google API'lerini kullanan bir Java Script kodu ile Google Maps (© 2010 Google) web sayfasında (<http://maps.google.com>) analiz edilmiştir. Bu şekilde mağazalar arası gerçek sürüş mesafeleri kilometre cinsinden elde edilmiştir.

Elde edilen veriler Visual Basic ortamında yazılan programa girdi olarak aktarılarak her iki bölge için çözümler bulunmuştur. Genetik algoritmanın uygulanmasında başlangıç popülasyonu, popülasyon büyüklüğü, ebeveyn seçim süreci, çaprazlama ve mutasyon ile ilgili önceki bölümlerde anlatılan alternatifler denenmiştir. Ayrıca farklı parametre değerleri için testler yapılmış ve bu testler sonucunda iyi çözümlerin popülasyon büyüklüğünün 40, başlangıç popülasyonunun sezgisel ve rastgele bir arada, ebeveyn seçim kuralının turnuva, çaprazlama operatörünün iki noktalı ve mutasyon olasılığının 0,015 olduğu durumda elde edildiği görülmüştür.

Firmanın Seyhan-1 ve Yüreğir-1 bölgeleri üzerinde yapılan çalışmalar ile ilgili sonuçlar sırası ile Tablo 1 ve Tablo 2'de verilmiştir. Tablo 1'in ilk sütunu Seyhan-1

bölgesi için firmanın mevcut durumda kullanmış olduğu rotayı göstermektedir. Firmanın merkezinden çıkan araç ilk olarak B-66 kodlu satış mağazasına uğrayacak, daha sonra B67'ye geçecektir. Bu şekilde tüm satış mağazalarına uğrayarak rotasını tamamlayacak olan araç tekrar merkeze dönecektir. Tablonun ikinci sütununda ise satış mağazaları arası uzaklıklar verilmiştir. Merkez ile B66 kodlu satış mağazası arası uzaklık 15,295 km, B66 ve B67 arası uzaklık 2,355 km şeklindedir. En alt satırda ise toplam tur uzunluğu 65,189 km olarak görülmektedir. Tablonun üçüncü sütunu genetik algoritma ile bulunmuş olan rotayı göstermekte, son sütun ise bu rotayı oluşturan noktalar arası mesafeleri ifade etmektedir. Genetik algoritma ile bulunan rotanın toplam uzunluğu tablodan da görüleceği üzere 54,323 km şeklindedir. Bu şekilde genetik algoritma ile bulunan rota ile firmanın mevcut durumda uygulamakta olduğu rotaya göre %20,003'lük bir iyileşme elde edilmiştir.

Tablo1. Seyhan-1 Bölgesi İçin Mevcut Rota Ve Genetik Algoritma Sonuçları

Firmanın Mevcut Rotası		Genetik Algoritma ile Bulunan Rota	
MERKEZ	Bir Sonraki Noktaya Uzaklık	MERKEZ	Bir Sonraki Noktaya Uzaklık
B66	15,295	B1	16,043
B67	2,355	B67	1,505
B1	1,898	B2	0,923
B8	1,842	B7	0,496
B7	1,021	B8	0,772
B2	0,594	B36	1,984
B28	3,447	B68	1,701
B36	3,988	B9	1,089
B68	1,701	B25	1,277
B26	2,575	B58	2,145
B69	1,106	B70	1,942
B9	2,281	B26	1,682
B25	1,277	B69	1,106
B70	1,071	B56	2,35
B58	2,493	B21	0,723
B56	4,718	B28	1,079
B-21	0,723	B66	2,864
MERKEZ	16,804	MERKEZ	14,642
Toplam Mesafe (km)	65,189	Toplam Mesafe (km)	54,323

Tablo 2'de Yüreğir-1 bölgesine ait bilgiler yer almaktadır. Bir önceki tabloya benzer şekilde ilk iki sütun firmanın halihazırda kullanmış olduğu rota ve mesafe bilgilerini içermektedir. İkinci sütunun son satırından görüleceği üzere bu durumda

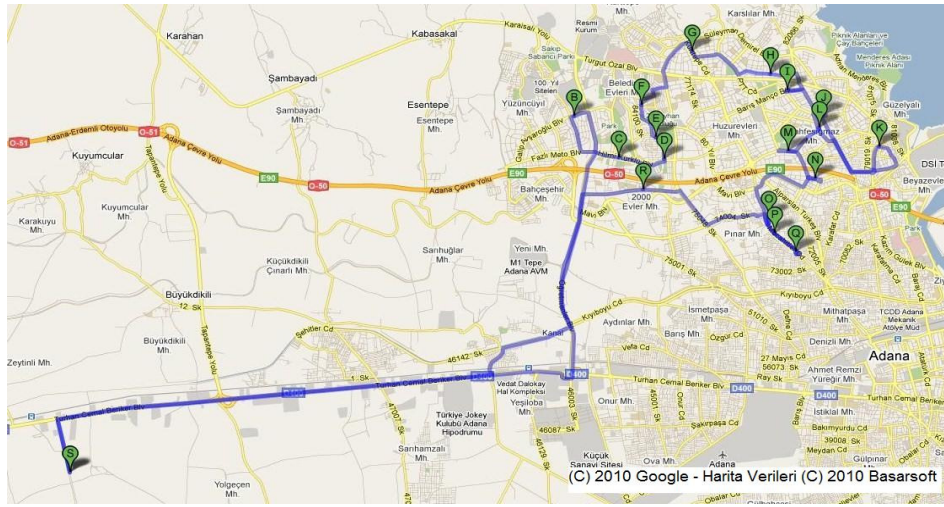
toplam kat edilen mesafe 119,809 km şeklindedir. Üçüncü ve dördüncü sütunlar genetik algoritma ile bulunan sonuçları göstermektedir. Genetik algoritma ile elde edilen rotanın toplam uzunluğu 99,725 km şeklindedir. Burada genetik algoritmanın uygulanması ile elde edilen rota firmanın mevcut rotasına göre 20,084 km daha kısadır ve bu rakam %20,139'luk bir iyileşmeye denk gelmektedir.

Şekil 5 ve Şekil 6'da yukarıdaki tablolarda belirtilen ve genetik algoritmalar kullanılarak elde edilen yeni rotaların Google Maps (© 2010 Google) web sayfasından (<http://maps.google.com>) elde edilen görselleri yer almaktadır. Şekil 5'de Seyhan Bölgesi için, Şekil 6'da ise Yüreğir Bölgesi için harita üzerinde satış mağazalarının gerçek noktalar bazında yerleri görülmektedir. Mersin yolu üzerinde yer alan merkez depo Şekil 5'de S, Şekil 6'da ise V noktası olarak işaretlenmiştir. Bu merkezden çıkan araçlar tüm satış noktalarına belirlenen sıra ile uğrayıp tekrar merkeze dönmektedirler.

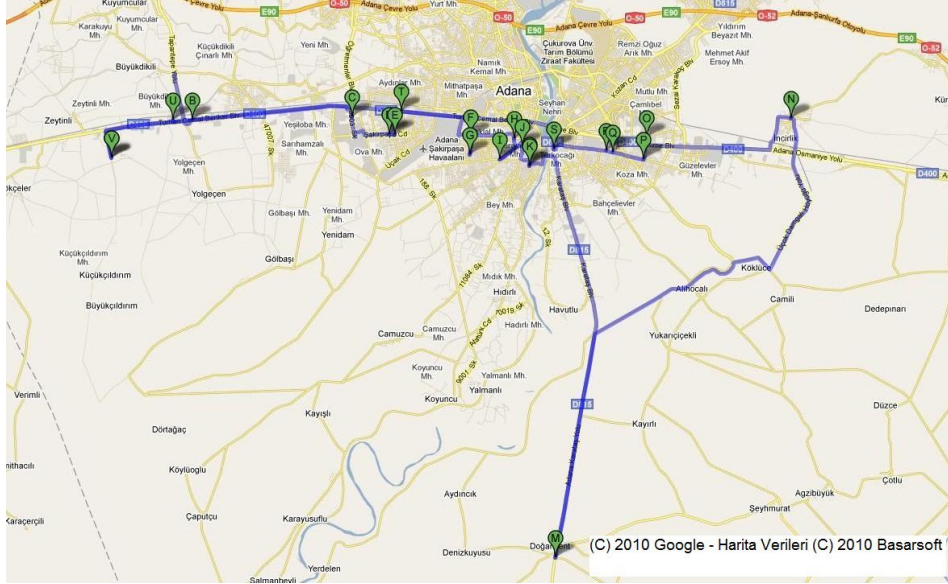
Tablo2. Yüreğir-1 Bölgesi İçin Mevcut Rota Ve Genetik Algoritma Sonuçları

Firmanın Mevcut Rotası		Genetik Algoritma ile Bulunan Rota	
MERKEZ	Bir Sonraki Noktaya Uzaklık	MERKEZ	Bir Sonraki Noktaya Uzaklık
B84	34,695	B91	8.695
B85	22,869	B93	6.417
B47	6,791	B89	2.298
B39	1,055	B53	0.205
B3	1,229	B88	4.996
B5	0,249	B87	0.696
B18	2,216	B46	2.523
B54	1,505	B86	1.095
B60	0,086	B50	1.281
B50	1,1	B60	1.1
B46	0,841	B54	0.085
B86	1,095	B84	16.75
B87	1,509	B85	22.869
B88	0,696	B47	6.791
B89	4,612	B39	1.055
B53	0,205	B3	1.229
B90	9,293	B5	0.249
B91	7,771	B18	2.216
B92	8,151	B92	6.762
B93	2,426	B90	8.691
MERKEZ	11,415	MERKEZ	3.722
Toplam Mesafe (km)	119.809	Toplam Mesafe (km)	99.725

Firmanın sadece iki bölgesinin incelenmesi ile elde edilen sonuçların incelenmesinde ortalama olarak %20 civarında bir kısalma olduğu görülmektedir. Günümüzde işletmelerin dağıtım maliyetlerinin önemi düşünüldüğünde günlük rotalarda yaklaşık olarak bu oranda yapılabilecek kısaltmalar işletmelere maliyet tasarrufu sağlayarak katkıda bulunacaktır.



Şekil 5. Genetik Algoritma Yöntemi İle Seyhan-1 Bölgesi İçin Elde Edilen Rota



Şekil 6. Genetik Algoritma Yöntemi İle Yüreğir-1 Bölgesi İçin Elde Edilen Rota

5. Sonuçlar

Bu çalışmada gezgin satıcı problemini çözmek amacı ile bir metasezgisel yaklaşım olan genetik algoritma önerilmiştir. Algoritmanın detayları ve adımları belirlenerek ilgili kodlamalar Visual Basic programlama dilinde tamamlanmıştır. Daha sonra çalışmanın bir uygulaması Adana'da faaliyet gösteren bir firma üzerinde gerçekleştirilmiştir.

Uygulama aşamasında firmadan elde edilen veriler kullanılarak geliştirilen algoritma ile yeni sonuçlar elde edilmiştir. Günümüz işletmelerinin birçoğunda dağıtım rotasının belirlenmesinde tecrübe ön planda tutulmakta ve tecrübeye dayalı olarak dağıtım sırası belirlenmektedir. Uygulama yapılan firmanın güncel olarak kullandığı rotaların da tecrübeye dayalı olarak belirlendiği gözlenmiştir. Genetik algoritmanın firmanın iki bölgesinin yeni rota belirlenmesi aşamasında kullanılması ile elde edilen sonuçlar bu algoritmanın etkinliğini göstermiştir. Bu sonuçlar incelendiğinde toplam kat edilen yolun her bir rotada firmanın güncel rotalarına oranla ortalama %20 civarı kısaldığı görülmüştür. Bu kısalma firmanın dağıtım maliyetlerinde azalmaya katkıda bulunacaktır.

Daha sonraki çalışmalarda benzer testler bölgesel dağıtım yapan farklı firmalarda yinelenerek firmaya özgü değişikliklerle daha iyi sonuçlar elde edilebilir ve firmalara avantaj sağlanabilir.

Kaynakça

Arora S., 1998. "Polynomial time approximation schemes for euclidian traveling salesman and other geometric problems", *Journal of the ACM*, 45(5): 753-782.

Applegate D.L., Bixby R.E., Chvatal V. ve Cook W.J., 2007. The traveling salesman problem: A computational study, Princeton University Press, Princeton, NJ,

Bellmore M. ve Nemhauser G. L., 1968. "The traveling salesman problem: A survey", *Operations Research*, 16:538-558.

Cevre U., Özkan B. ve Uğur A., 2007. "Gezgin satıcı probleminin genetik algoritmalarla eniyilemesi ve etkileşimli olarak internet üzerinde görselleştirilmesi", XII. "Türkiye'de İnternet" Konferansı 8-10 Kasım 2007, Ankara.

Chatterjee S., Carrera C. ve Lynch L., 1996. "Genetic algorithms and traveling salesman problems." *European Journal of Operational Research*, 93:490-510.

Cochrane E. M. ve Beasley J. E., 2003. "The co-adaptive neural network approach to the euclidean traveling salesman problem", *Neural Networks*, 16(10):1499-1525.

Croes G. A., 1958. "A method for solving traveling salesman problems", *Operations Research*, 6:791-812.

Dantzig G., Fulkerson R., ve Johnson S., 1954. "Solution of a large-scale traveling-salesman problem", *Operations Research*, 2: 393-410.

Dorigo M., Maniezzo V. ve Colomi A., 1996. "The ant system: optimization by a colony of cooperating agents", *IEEE Transactions on Systems, Man, and Cybernetics-Part B*, 26(1): 29-41.

Fiechter C.N., 1994. "A parallel tabu search algorithm for large traveling salesman problems", *Discrete Applied Mathematics*, 51: 243 -267.

Gao H.C., Feng B.Q., ve Zhu,L., 2006. "Reviews of the meta-heuristic algorithms for TSP", *Control ve Decision*, 3: 241-246.

Garey M.R. ve Johnson D.S., 1979. Computers and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman and co., New York.

- Goldberg, D. ve Lingle, R. (1985). "Alleles, loci, and the TSP". In *Proceedings of an International Conference on Genetic Algorithms and Their Applications*, 154-159.
- Goldberg D., 1989. *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, USA.
- Gonzales E.L. ve Fernandez M.A.R., 2000. "Genetic optimization of a fuzzy distribution model", *International Journal of Physical Distribution & Logistics Management*, 30(7/8): 681-696.
- Holland J.H., 1975. *Adaptation in Natural and Artificial Systems*, The University of Michigan Press, Ann Arbor, MI.
- Jang J.S.R., 1997. *Neuro-Fuzzy and Soft Computing: A Computational Approach To Learning and Machine Intelligence*, Prentice-Hall, USA, 173-196.
- Johnson D.S. ve McGeoch L.A., 1995. "The traveling salesman problem: a case study in local optimization", *Local Search in Combinatorial Optimization*, 215-310.
- Jozefowicz N., Glover F. ve Laguna M., 2008. "Multi-objective meta-heuristics for the traveling salesman problem with profits", *Journal of Mathematical Modelling and Algorithms*, 7(2): 177-195.
- Kirkpatrick S., Gelatt C.D. ve Vecchi M.P., 1983. "Optimization by simulated annealing", *Science*, 220: 671-680.
- Laporte G., 1992. "The traveling salesman problem: an overview of exact and approximate algorithms", *European Journal of Operational Research*, 59:231-247.
- Larranaga P., Kuijpers C.M.H., Murga R.H., Inza I., Dizdarevic S., 1999. "Genetic algorithms for the traveling salesman problem: A review of representations and operators", *Artificial Intelligence Review*, 13: 129 -70.
- Malek M., Guruswamy M., Pandya M. ve Owens H., 1989. "Serial and parallel simulated annealing and tabu search algorithms for the traveling salesman problem", *Annals of Operations Research*, 21(1): 59-84.
- Menger K., 1932. "Das botenproblem", in *Ergebnisse eines Mathematischen Kolloquiums 2* (K. Menger, editor), Teubner, Leipzig.
- Merz P. ve Freisleben B., 2001. "Memetic algorithms for the traveling salesman problem", *Complex Systems*, 4:297-345.

- Miliotis P., 1976. "Integer programming approaches to the traveling salesman problem", *Mathematical Programming* 10: 367-378.
- Murata, T., Ishibuchi, H. ve Tanaka, H. (1996). "Genetic algorithms for flow shop scheduling problems, *Computers and Industrial Engineering*, 30:4, 1061-1071.
- Roberts S. M. ve Flores B., 1966. "An engineering approach to the traveling salesman problem", *Management Science*, 13:269-288.
- Schmitt L ve Amini M., 1998. "Performance characteristics of alternative genetic algorithmic approaches to the traveling salesman problem using path representation: an empirical study." *European Journal of Operational Research*, 108:551-70.
- Syswerda, G., 1991. "Schedule optimization using genetic algorithms", *Handbook of Genetic Algorithms*, New York NY, 350-372.
- Takahashi, R., 2005. 'Solving the traveling salesman problem through genetic algorithms with changing crossover operators', *Machine Learning and Applications, Proceedings, Fourth International Conference*, Los Angeles, CA, 319-325.
- Tao Y. Q., Cui D. W., Miao X. L., ve Chen, H., 2007. "An improved swarm intelligence algorithm for solving TSP problem", *Lecture Notes in Computer Science*, 813-822.
- Tsai H.K., Yang J.M., Tsai Y.F. ve Kao C.Y., 2004. "Some issues of designing genetic algorithms for traveling salesman problems", *Soft Computing*, 8: 689-697.
- Yeo M.F., ve Agyei E.O., 1998. "Optimizing engineering problems using genetic algorithms", *Engineering Computations*, 15(2): 268-280.