Google Maps ve Genetik Algoritmalarla GSP Çözümü İçin Öneri

Onur Karasoy1, Serkan Ballı2

1. Muğla Sıtkı Koçman Üniversitesi, Bilgi İşlem Daire Başkanlığı, Muğla
2. Muğla Sıtkı Koçman Üniversitesi, Bilişim Sistemleri Mühendisliği Bölümü, Muğla

[okarasoy@mu.edu.tr](mailto:okarasoy@mu.edu.tr), [serkan@mu.edu.tr](mailto:serkan@mu.edu.tr)

Özet: Gezgin satıcı problemi optimizasyon (eniyileme) alanında uzun yıllardır üzerine çalışılan ve çözümü zor olarak kabul edilen (NP-Zor) bir problemdir. Özellikle rota belirlemenin gerekli olduğu iş alanlarında bu probleme sıklıkla rastlanmaktadır. Sadece matematiksel yöntemlerle çözüm bulmanın bazen imkânsız olduğu bu probleme sezgisel yöntemlerle çözüm aranmaktadır. Bu çalışmada meta sezgisel bir yöntem olan genetik algoritma ile Google Maps kullanarak harita üzerinde işaretlenmiş noktalar için en kısa (optimum) rota belirleyen bir uygulama gerçekleştirilmiştir.

Anahtar Sözcükler: Genetik Algoritmalar, Gezgin Satıcı Problemi, Google Maps.

A Web Application to Solve Traveling Salesman Problem by Using Genetic Algorithm and Google Maps

Abstract: Traveling Salesman Problem has been studied for a long time in the area of optimization and considered as difficult problem to solve (NP-hard). Especially in the working areas that need to be determining route, this problem is often be encountered. The solutions are sought with heuristic methods for this problem which is sometimes impossible to find a solution with only mathematical methods. In this study, an application is carried out for the shortest (optimum) route determination by using the points marked on the Google Maps with genetic algorithm that is a meta-heuristic method.

Keywords: Genetic Algorithm, Traveling Salesman Problem, Google Maps.

1. Giriş

GSP (Gezgin Satıcı Problemi) Np-Zor bir problem olarak tanımlanmaktadır. Bu problemde genel amaç, bir satıcının n adet şehri (node) en kısa yoldan şehirlerin her birine sadece bir defa uğrayarak başlangıç noktasına geri dönmesini sağlamaktır. Tanımlaması kolay fakat çözümü zor bir problemdir.[1]

Optimizasyon problemlerinin çözümünde kullanılan genetik algoritmalar ilk kez 1975 yılında Holland tarafından önerilmiştir. Doğal seçme ve genetik evrimden yola çıkarak oluşturulan bu algoritma biyolojik sistemdeki bireyin bulunduğu çevreye uyum sağlayıp uygun hale gelmesi örnek alınarak geliştirilmiştir.

Literatürdeki yayınlar incelendiğinde; Qu ve Sun [2], gezgin satıcı problemi çözümünde genetik algoritma kullanımını ele almış ve üstel bağıntı, entropi atlaması, asimilasyon gibi davranışların genetik algoritmada erken yakınsama ihtimaline etkisini gözlemişlerdir. Luis ve Li [3], öğrenebilen bir genetik algoritma tasarımı gerçekleştirmişlerdir. Çalışmada GSP için üretilmiş önceki çözümlerden yola çıkarak daha hızlı çözüm bulunmuştur. Katayama vd. [4] ve Moon vd. [5], GSP çözümünde genetik algoritma adımlarından çaprazlama işlemini geliştirmeye yönelik çalışmışlardır. Xing vd. [6], Asimetrik GSP çözümünde optimizasyon teknikleri ile geliştirilmiş bir genetik algoritma kullanılıp klasik genetik algoritmalar ile karşılaştırmışlardır. Zhu ve Li [7], GSP için karınca kolonisi algoritmasıyla çözüm üretmek için uygulama geliştirmişlerdir.

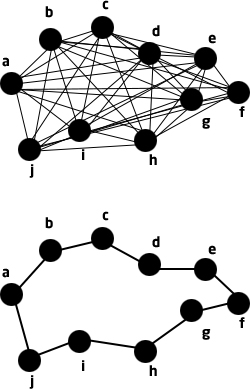
Bu çalışmada, Google Maps üzerinde işaretlenen konumlar kullanılarak genetik algoritma aracılığıyla GSP için çözüm üretilip görselleştirilmiştir.

2. Gezgin Satıcı Problemi

GSP’de amaç bir satıcının belli bir şehirden başlayıp her şehre sadece bir defa uğradıktan sonra başladığı şehre dönmek için en kısa mesafeyi hesaplamaktır. GSP’de yolların şehirlerarası olduğu ve uzunlukların bilindiği varsayımıyla hareket edilir. Anlaşıldığı üzere GSP kolay anlaşılabilen ama çözüm süreci nispeten zor bir problemdir.

GSP İngiliz matematikçi W.R.Hamilton ile Thomas Kirkman tarafından bir matematik problemi olarak ele alındı [8]. GSP’nin ilk uygulaması olarak kabul edilen Amerika kıtasında 48 şehirli bir GSP nin çözümü 1954 yılında Dentzig, Fulkerson ve Johnson tarafından, kesen düzlem yöntemiyle gerçekleştirilmiştir [9].

Durum uzayının büyük olmasından dolayı GSP’ye sezgisel yöntemlerle çözüm aranmaktadır. Problemdeki şehir sayısı az da olsa klasik matematiksel yöntemlerle çözülme işlemi hayli zaman almaktadır. Şekil 1’de 10 şehirli bir GSP için kullanılan ihtimaller ve çözüm gösterilmektedir[10].

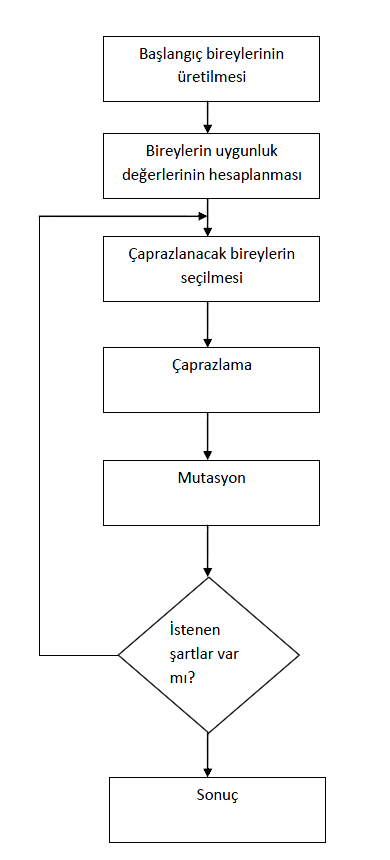
Şekil 1. Örnek GSP çözümü

3. Genetik Algoritmalar

Genetik algoritmalar, Darwin’in doğal seleksiyon ve evrim teorisi ilkelerine dayanarak oluşturulmuş bir arama optimizasyon yöntemidir. Bir probleme, olası pek çok çözüm üzerinden en uygununu bulmaya çalışan bir algoritmadır.

Genetik algoritmaların çıkış noktaları çözüm kümeleri olduğu gibi sonuç olarak da çözüm kümeleri üretir [12]. Popülasyon bir sonraki nesle aktarıldıkça iyi çözümler oluşturma eğilimindedir. İyinin ne olduğunu belirlemek için uygunluk fonksiyonundan faydalanılır. Yeni çözümler üretmek için yeniden kopyalama ve mutasyon gibi operatörler kullanılır.

Genetik algoritmaların işlem basamakları Şekil 2‘de gösterilmektedir [10]. Buna göre öncelikle olası çözümlerden rastgele bireyler



Şekil 2. GA Akış Diyagramı

oluşturulur. Oluşturulan her bireye kromozom ve topluluğa da popülasyon denmektedir. Oluşturulan bireylerin uygunluk değerleri hesaplanır. Uygunluk değeri probleme göre göre farklı uygunluk fonksiyonlarıyla belirlenir. Uygunluk fonksiyonuna göre çaprazlanacak bireyler seçilir. Çaprazlama işleminden sonra yerel minimuma takılmamak için önceden belirlenmiş oranda mutasyon işlemine tabi tutulur.

Yeni oluşturulan nesillerle birlikte uygunluk değerleri tekrardan hesaplanır. Eski kötü bireylerle yenileri yer değiştirerek popülasyonun belli sayıda kalması sağlanır.

Bu işlemler yine belli bir nesil sayısına kadar devam eder. Gerekli sonlandırma değerine gelindikten sonra en uygun kromozom çözüm olarak kabul edilir.

3.1 Genetik Algoritma ile Gezgin Satıcı Problemi Çözümü

Gezgin satıcı probleminde kromozomlar oluşturulurken dikkat edilmesi gereken noktaların en önemlisi satıcının aynı şehre birden fazla kez uğramaması ve başlangıç noktasına geri dönmüş olmasıdır.

Gezgin satıcı problemi için permütasyon yöntemiyle kromozomlar oluşturulmaktadır. Şekil 3’te örnek bir kromozom gösterilmiştir.



Şekil 3. GSP İçin Örnek Kromozom

3.2 Başlangıç Popülasyonu

Başlangıç popülasyonu olası çözümlerden oluşturulmuş bireyler topluluğudur. GSP için her birey şehirlerarası güzergâha karşılık gelir ve genlerin her biri bir şehri temsil eder. Bu koşullar doğrultusunda genler, aynı şehre ikince defa uğramama ilkesiyle sıralanır. Örnek popülasyon Şekil 4’deki gibi oluşturulabilir:

|  |
| --- |
| 3 5 9 0 8 2 7 1 6 4  1 4 6 2 3 9 5 8 0 7  5 1 9 2 4 7 3 6 0 8  7 8 3 9 6 5 0 2 1 4 |

Şekil 4. Örnek popülasyon

3.3 Uygunluk Değerinin hesaplanması

Popülasyon oluşturulduktan sonra tüm bireylerin uygunluk değerleri hesaplanır. Bu problemde uygunluk değeri şehirlerarası mesafelerin hesaplanmasıyla bulunur. En kısa mesafeye sahip olan kromozomlar istenen kromozomlardır. Bir çözümün uygunluk değeri ne kadar yüksekse, yaşama ve çoğalma şansı o kadar fazladır ve bir sonraki kuşakta temsil edilme oranı da o kadar yüksektir [12].

3.4 Yeniden Üretme

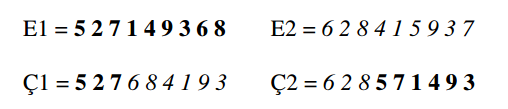
Çoğalma, bireyleri seçme, seçilmiş bireyleri bir eşleme havuzundan kopyalama ve havuzda bireyleri çiftler halinde gruplara ayırma işleminden oluşur [14].

Uygunluk değeri hesaplamalarının ardından sonraki kuşaklar için yeni bireyler oluşturulur. Uygunluk değerleri istenilene yakın olan bireylere çoğalma şansı tanınarak daha iyi bireyler üretilesi amaçlanır. Bir dizinin kopyalanma şansı uygunluk değerine bağlıdır [15].

3.5 Çaprazlama

Kromozomlar üzerinde yapılan çaprazlamada amaç daha iyi, daha dayanıklı birey oluşturup ait olduğu bireyler topluluğunun neslini, dışarıdan etkiye daha dayanıklı hale getirmektir [16]. Çaprazlama yapmak için farklı yöntemler bulunmakla birlikte en basit yöntem tek noktalı çaprazlamadır.

Tek noktalı çaprazlama işlemi için seçilen iki ebeveyn birey için ortak bir çaprazlama noktası belirlenir. Birinci çocuk birince ebeveynin çaprazlama noktasına kadar olan genlerini, ikinci ebeveynin çaprazlama noktasından sonraki genlerini alarak oluşur. İkinci çocukta ise bu işlemin tam tersi uygulanır. Şekil 5‘te yer olan örnekte çaprazlama noktası üçüncü gen olarak belirlenmiş yeni iki çocuk birey bu noktaya göre oluşturulmuştur [11].

Şekil 5. Tek Noktalı Çaprazlama

İki noktalı çaprazlamada ise kromozom iki noktadan kesilir ve karşılıklı olarak pozisyonlar yer değiştirilir [14]. Bu yöntemlerden farklı olarak pozisyona dayalı çaprazlama, kısmi planlı çaprazlama, sıraya dayalı çaprazlama vb. yöntemlerde bulunmaktadır.

3.6 Mutasyon

Popülasyon gerekli tüm kodlanmış bilgiyi içermezse çaprazlama tatmin edici bir çözüm üretemez. Bu nedenle mevcut kromozomlardan yeni kromozomlar üretmek için mutasyon operatörü kullanılmaktadır [17]. Mutasyon operatörü bir daha elde edilemeyecek iyi bir çözümün kaybına karşı koruma sağlamaktadır.[18]

Gezgin satıcı probleminde mutasyon rasgele seçilen iki genin yer değiştirmesi şeklinde yapılır. Şekil 6‘da ikinci ve yedinci genleri seçilerek yapılmış bir mutasyon örneği gösterilmektedir.

|  |
| --- |
| K= 6 2 8 1 4 9 7 5 3  Mutasyona uğradıktan sonra:  K= 6 7 8 1 4 9 2 5 3 |

Şekil 6. Mutasyon Örneği

3.7 Döngünün Sonlandırılması

Yeniden üreme, çaprazlama ve mutasyon işlemlerinden sonra yeni kromozomlar yani çocuklar oluşmaktadır. Bu yeni oluşan kromozomlar ebeveynlerine göre daha iyi uygunluk değerlerine sahip olabilir. Yapılan seçime göre uygunluk değeri daha iyi olan çocuklar ebeveynlerin yerine popülasyona alınır. Yani yeni nesil oluşturulmuş olur.

Bu işlemler önceden belirlenen nesil sayısına ulaşıncaya kadar veya belirlenen başka bir durdurma kriteri sağlanıncaya kadar devam eder.

4. Google Maps ve Javascript ile GSP için Genetik Algoritmaların Uygulanması

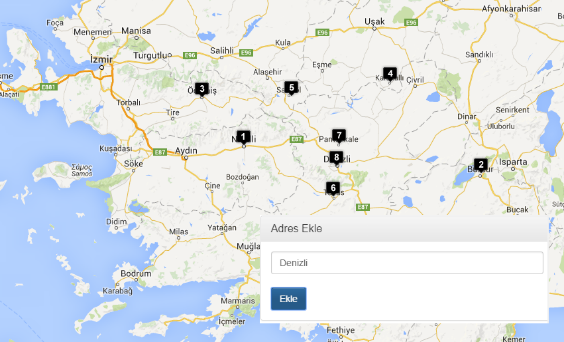
Bu çalışmada; GSP için önerilen Genetik algoritmalar Google Maps Api V3 ve Javascript kullanılarak hazırlanan web sayfasında görselleştirilerek Ege Bölgesine ait optimum yol bulma uygulaması hazırlanmıştır. Uygulamanın adımları aşağıdaki alt bölümlerde detaylı olarak anlatılacaktır.

4.1 Konumların Belirlenmesi

Gezgin satıcı probleminde tüm gidilmesi gereken konumların birbirlerine olan uzaklıkların bilinmesi gerekmektedir. Bu bilgiye ulaşmak için kullanılabilecek harita servislerinden en çok tercih edilenlerinden biri olan Google Maps‘tir. Konumlar belirlenirken Google Maps, konum bilgileri için doğru verilere ulaşmada verimli, kullanıcılar için kolay kullanılabilir bir araç olmaktadır.

Bu uygulamada konum seçimi için Google Maps görüntüsü kullanılmaktadır. Konum ekleme adımı harita üzerine tıklanarak yapılabileceği gibi isteye bağlı olarak *Adres Ekle* alanından adres yazılarak da tamamlanabilmektedir.

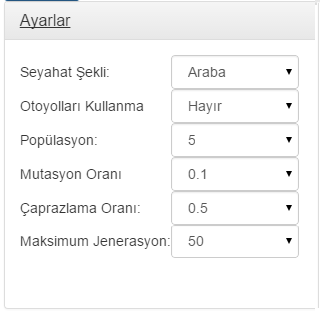
Örnek konumlar (Nazilli, Burdur, Ödemiş, Karahallı, Sarıgöl, Tavas, Pamukkale, Denizli) Şekil 7’de verilmiştir.



Şekil 7. Konum Seçimi Ekran Görüntüsü

4.2 Program Değişkenlerinin Tanımlanması

Genetik algoritmaya ve haritaya ait parametreler *Ayarlar* sekmesinden belirlenmektedir. Şekil 8’de gösterildiği gibi seyahat şekli, otoyolların kullanılıp kullanılmayacağı, popülasyon sayısı, mutasyon oranı, çaprazlama oranı ve maksimum jenerasyon sayısı alanları kullanılarak istenen değerler seçilmektedir.

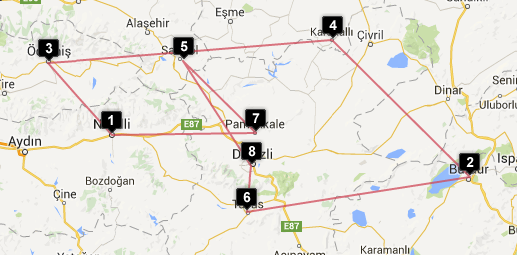


Şekil 8. Program parametreleri

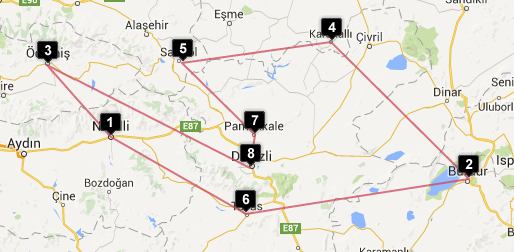
4.3 Programın çalıştırılması

Programa ait parametreler belirlendikten sonra program çalıştırılır. Program daha önce bahsedilen genetik algoritma adımlarını uyguladıktan sonra bulduğu en iyi sonucu istenilen rota olarak belirler ve kullanıcıya yol tarifi ile birlikte gösterir.

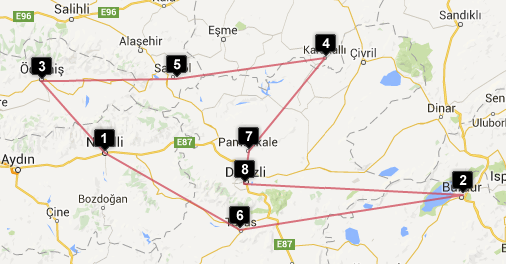
Örnek konumlarla işlem yapılacak olursa Şekil 9-15 ‘te görüldüğü üzere her yeni nesil oluştuğunda o adımdaki en iyi sonuç ekranda görüntülenmektedir.



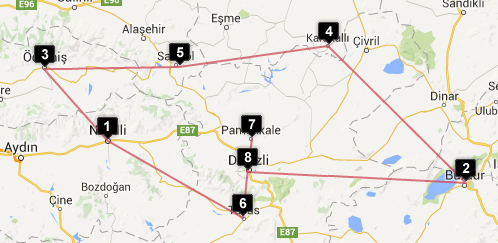
Şekil 9. 1.Nesilde bulunan en yakın yol (886.44 km)



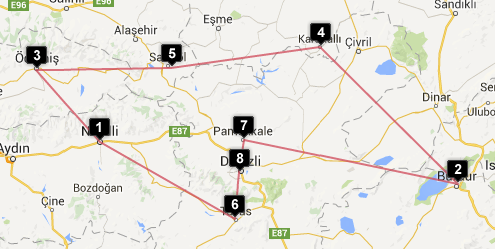
Şekil 10. 3.Nesilde bulunan en yakın yol (859.13 km)



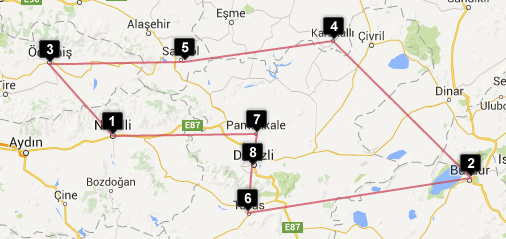
Şekil 11. 4.Nesilde bulunan en yakın yol (779.87 km)



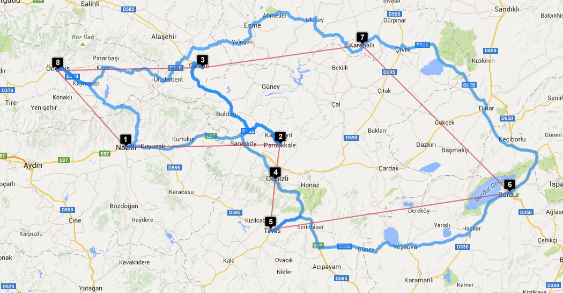
Şekil 12 .6.Nesilde bulunan en yakın yol (764.01 km)



Şekil 13. 10.Nesilde bulunan en yakın yol (752.01 km)



Şekil 14. 13.Nesilde bulunan en yakın yol (735.09 km)



Şekil 15. 20.Nesilde bulunan en yakın yol

ve yol tarifi (735.09 km)

Şekil 9-15’de gösterildiği gibi her adımdaki en iyi kromozom, toplam maliyeti düşürerek başlangıç pozisyonuna göre daha avantajlı çözüm üretilmiş olmaktadır. Şekil 9’daki 1. nesilde en kısa yol 859,13 km iken Şekil 15‘de 20.nesilde 735,09 km gibi bir sonuçla işlem tamamlanmaktadır.

5. Sonuç

Bu çalışmada gezgin satıcı problemi çözümü için Google Maps ile desteklenmiş Genetik Algoritmalar kullanılmıştır. Adımların detayları tartışılarak Javascript kodlarıyla web tabanlı bir uygulama geliştirilmiştir. Genetik algoritmanın çalışma prensibi gereği her zaman optimum sonuç alınamayabilir. Fakat kısa zamanda az işlem maliyetiyle optimuma en yakın değere ulaşılmaktadır. İlerleyen çalışmalarda harita büyütülerek daha çok şehir sayısıyla daha büyük problemlere çözüm aranacaktır.

6. Kaynaklar

[1] Potvin, J., Y., “Genetic algorithms for the travelling salesman problem”, **forthcoming in Annals of Operations Research**, 63: 339-370 (1996).

[2] Qu, L. ve Sun, R., “A synergetic approach to genetic algorithms for solving traveling salesman problem”, **Information Sciences** 117, 267-283 (1999).

[3] Louis, S. J. ve Li, G., "Case injected genetic algorithms for traveling salesman

problems", **Information Sciences**, 122, 201–225, (2000).

[4] Katayama, K., Sakamoto, H. ve Narihisa, H., “The efficiency of hybrid mutation genetic algorithm for the travelling salesman problem”, **Mathematical and Computer Modelling**, 31, 197–203, (2000).

[5] Moon, C., Kim, J., Choi, G. ve Seo, Y., "An efficient genetic algorithm for the traveling salesman problem with precedence constraints", **European Journal of Operational Research**, 140, 606–617, (2002).

[6] Xing L. N.,Chen Y., Yang K., Hou, F., Shen, X. ve Cai, H., "A hybrid approach combining an improved genetic algorithm and optimization strategies for the asymmetric traveling salesman problem", **Engineering Applications of Artificial Intelligence**, 21, 1370– 1380, (2008).

[7] Zhu J.ve Li Q., “Solving Travelling Salesman Problem by the Program of Ant Colony Algorithms”, **International Conference on Computational Intelligence and Software Engineering** **(CISE),** 1-3, (2009)

[8] Biggs N. L., LLOYD E. ve WILSON R.J., “Graph Theory” **Clarendon Press Oxford**, 1736-1936, (1976).

[9] Dantzig, G.B., Fulkerson, D.R. ve Johnson, S.M., “Solution of a large travelling salesman problem”, **Operations Research**, Vol 2:393-410, (1954).

[10] Özsağlam M.Y., “Parçacık Sürü Optimizasyonu algoritmasının gezgin satıcı problemine uygulanması ve performansının incelenmesi”*,* **Yüksek Lisans, Selçuk Üniversitesi, Konya**, 120s. (2009).

[11] Çolak, S., “Genetik Algoritmalar Yardımı İle Gezgin Satıcı Probleminin Çözümü Üzerine Bir Uygulama” **Ç.Ü. Sosyal Bilimler Enstitüsü Dergisi**, 19,3:423-438s, (2010).

[12] Albayrak, M., “Baskı Devre Kartı Delik Delme Makineleri için Genetik Algoritmalar Yardımı ile Güzergah Belirleme”, **Yüksek Lisans, Selçuk Üniversitesi, Konya** 180s, (2008).

[13] Yeniay, Ö., “An Overview of Genetic Algorithms”, **Anadolu Üniversitesi Bilim ve Teknoloji Dergisi**, 2, 1: 37-49s, (2001).

[14] Fığlalı, A. ve Engin O., “Genetik Algoritmalarla Akış Tipi Çizelgelemede Üreme Yöntemi Optimizasyonu”, **İTÜ Dergisi**, 1-6s, (2002).

[15] Jang, J. S. R., “Neuro-Fuzzy and Soft Computing: A Computational Approach To Learning and Machine Intelligence”, Chapter 7: **Derivative-Free Optimization, Prentice-Hall**, USA, 173-196s, (1997).

[16] Kaya, M.,”Genetik Algroitma ve Gezgin Satıcı Problemi”, **Yüksek Lisans, Fırat Üniversitesi, Elazığ**, 85s, (1999).

[17]Emel, G.G. ve Taşkın Ç., Genetik algroitmalar ve uygulama alanları, **Uludağ Üniversitesi İktisadi İdari Bilimler Fakültesi Dergisi**, 21,1:129-152s, (2002).

[18] Goldberg D.E., Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, USA, (1989).

[19] Yeo M.F., ve Agyei E.O., “Optimizing engineering problems using genetic algorithms”, **Engineering Computations**, 15(2): 268-280, (1998).