

Genetik Algoritma Yaklaşımı ve Yöneylem Araştırmasında Bir Uygulama

Dr. Öznur İŞÇİ

Celal Bayar Üniversitesi, Bilgisayar Araştırma ve Uygulama Merkezi, MANİSA

Prof. Dr. Serdar KORUKOĞLU

Ege Üniversitesi, Bilgisayar Mühendisliği Bölümü, İZMİR

ÖZET

Endüstri alanlarında klasik programlama ve yöneylem araştırması teknikleri ile geliştirilen programlar yerini artık yapay zeka teknikleri kullanılarak geliştirilen çalışmalara bırakmaktadır. Böylece planlanan üretimi artırmak ve kârı maksimize etmek için; sezgisel parametreleri kullanma, doğru analiz yapabilme ve anında karar verme gibi insana özgü olan yetileri kullanarak karar veren veya tavsiyelerde bulunan sistemlerin geliştirilmesi ile daha hızlı ve gerçekçi çözümler elde edilecektir.

Halen araştırılmakta olan genetik algoritmaların optimizasyon problemlerinin çözümü için kullanılması oldukça yenidir. Bu çalışmada genetik algoritmanın nasıl çalıştığı ve yöneylem araştırması problemleri arasında yer alan gezgin satıcı probleminin genetik algoritma ile çözümü üzerinde durulmuş ve bunun için geliştirilen bir java programı ile çözümü tanıtılmıştır. Ayrıca genetik algoritma çözümü ve klasik yöntemlerle çözümleri karşılaştırılmaktadır.

Anahtar Kelimeler: Genetik Algoritma, Yöneylem Araştırması, Optimizasyon, Gezgin Satıcı Problemi

Genetic Algorithm Approach and an Application in Operational Research

ABSTRACT

In industrial area, programs that are developed using artificial intelligence have been replacing the programs that are developed using classical programming and operational research techniques. Thus to increase the planned production and to maximize the profit; faster and realistic solutions will be obtained by developing systems that make recommendations and decide using facilities that are peculiar to human such as using intuitional parameters, making right analysis and decisions.

It is very new to use genetic algorithms, which are still being researched, in solving optimization problems. In this study how genetic algorithm works and the solution of traveling salesman problem, which is among the operational research problems, using genetic algorithm are explained and a solution by a Java program is shown. Additionally solutions obtained using genetic algorithm and classical methods are compared.

Key Words: Genetic Algorithm, Operation Research, Optimization, Traveling Salesman Problem.

1. GİRİŞ

Son yıllarda optimizasyonun önemi, bir çok büyük ölçekli kombinatorol optimizasyon(combinatorial optimization) problemlerinin ve yüksek kısıtlı mühendislik problemlerinin günümüz bilgisayarları ile yaklaşık olarak çözülebildiğinden daha da artmıştır. Genetik algoritmaların(GA) amacı böyle kompleks problemlerdir. Bu problemler, olasılıklı algoritmalar sınıfına ait olmakla birlikte rasgele algoritmalarından çok farklıdır.

GA doğadaki evrim yöntemlerini kullanan bir arama yöntemidir. Genetik algoritma tekniği, Michigan Üniversitesinde yer alan John Holland ve arkadaşlarının liderliğinde ki çalışmalar sonucu 1970’li yıllarda ortaya çıkmış ve 1975’de Holland “Doğal ve Yapay Sistemlerin Uygulanması” adlı kitabını yayınlamıştır. Mekanik öğrenme konusunda çalışan Holland, Darwin’in evrim kavramından etkilenerek canlılarda yaşanan genetik süreci bilgisayar ortamında gerçekleştirmeyi düşündü. 1985’te Holland’ın öğrencisi olarak doktorasını veren David E. Goldberg adlı inşaat mühendisi 1989’da konusunda bir klasik sayılan kitabını yayınladığı dek genetik algoritmaların pek yararı olmayan bir araştırma konusu olduğu düşünülüyordu.

Bugün bilgisayar yöntemleri biyolojik değerlendirmeden esinlenerek evrimsel hesaplama olarak adlandırılan bir şemsiye altında gruplandırılmıştır. Evrimsel hesaplamaların ana elemanları aşağıda tanımlanmaktadır (Karr ve Freeman, 1999:5):

- (1) Değerlendirme stratejileri
- (2) Evrimsel programlama
- (3) Genetik algoritmalar

Bu üç tekniğin her biri doğal değerlendirmedeki gözlemlenen süreci taklit eder ve verilen problem için aday çözümlerin değerlendirme popülasyonlarıyla etkili arama motorları sağlar. GA’lar genel olarak evrimsel hesaplama alanındaki en göze çarpan teknik olarak düşünülebilir.

GA evrimsel programlamanın en yaygın ve en çok kullanılan dalıdır. Türkiye dahil dünyada pek çok araştırmacı bu konuda çalışmaktadır. Son yıllarda genetik algoritmalara ilgi büyüyerek artmaktadır. Bu konu üzerine bir çok kitap basılmış ve buna ek olarak yılda iki kez genetik algoritmalar üzerine konferanslar yapılmaktadır.

GA hem problem çözmek hem de modelleme için kullanılmaktadır. Günümüzde genetik algoritmaların uygulama alanları genişlemektedir. Bunlardan bazıları: Atölye Çizelgeleme, Yapay Sinir Ağları Tasarımı, Görüntü Kontrolü, Elektronik Devre Tasarımı, Optimizasyon, Uzman Sistemler, Paketleme Problemleri, Makine ve Robot Öğrenmesi, Gezgin Satıcı Problemi, Ekonomik Model Çıkarma v.b sayılabilir(Mitchell ve Forest,1994:3).

Canlıların yapılarında var olan bir takım özellikler sanal ortamlarda taklit edilerek modeller geliştirilmeye ve bu modellerle de karşılaşılan problemlere çözümler bulunmaya çalışılmaktadır. Bu modellerin birisi olan genetik algoritmalar canlıların çevreye uyum ve genetik özelliklerinin araştırılmasıyla geliştirilmiştir(Kömür ve Altan,2001:194).

GA geleneksel sezgisel yöntemlerinden daha etkili ve çözüm yaklaşımında yapılacak küçük değişikliklerle halledilebildiklerinden dolayı da daha esneklerdir.

Bu sebeple, GA araştırmacıların ilgisini çekmektedir. Bilindiği üzere optimizasyondaki temel amaç optimal bir noktaya ulaşabilmek, daha doğrusu

mümkün oldukça yaklaşmaktır. Bunu gerçekleştirmek için bilinen pek çok klasik yöntem vardır. Bu yöntemlerin başarısı optimal noktaya ulaşıp ulaşmadıkları veya ne kadar ulaşabildikleri ile ölçülür. Genetik algoritmalar, klasik optimizasyon algoritmalarından dört temel noktada ayrılır(Oğuz ve Akbaş,1997:8):

- (i) **GA parametrelerinin kendileri ile değil, parametre takımının kodlanmış bir haliyle uğraşılır.**
- (ii) **GA aramaya tek bir noktada değil, bir nokta ailesinden başlarlar. Dolayısıyla yerel bir optimuma takılmadan çalışabilirler.**
- (iii) **GA amaç fonksiyonunun (objective function) türevlerini ve bir takım ek bilgileri değil, doğrudan amaç fonksiyonunun kendisini kullanırlar.**
- (iv) **GA'da deterministik değil rastlantısal geçiş kuralları kullanılır.**

GA'ların parametreleri; çaprazlama oranı, mutasyon oranı, populasyon büyüklüğü, seçim, kodlama(encoding), çaprazlama ve mutasyon tipi gibi genel parametrelerdir. Çaprazlama oranı yüksek olmalıdır. Buna karşılık mutasyon oranı çok düşük olmalıdır. Şaşırtıcı olan çok büyük populasyon büyüklüğü genellikle GA değerini arttırmaz (Çözüm bulma hızı anlamında). İyi populasyon büyüklüğü yaklaşık olarak 20-30 olmalıdır, bununla birlikte bazen 50-100 daha iyi sonuç verebilir. Seçim için genellikle rulet tekerleği kullanılır, bunun yanısıra rank seçimi, kararlı durum(steady state) ve elitizm gibi seçim yöntemleri de kullanılmaktadır.

GA yöntemi, yıllar boyu süregelen genetik mühendisliği ve biyoloji alanında yapılan çalışmalar sonucu ortaya atılmış bir tekniktir ve her sisteme uygulanmaya bilir(Portman,1996:2). Standart bir GA yöntemi aşağıdaki gibi verilebilir:

- (i) **Başlangıç populasyonunu rastlantısal olarak üret.**
- (ii) **Populasyon içindeki tüm kromozomların amaç fonksiyonu değerlerini hesapla.**
- (iii) **Tekrar üreme, çaprazlama ve mutasyon operatörlerini uygula.**
- (iv) **Oluşturulan her yeni kromozomun amaç fonksiyonu değerlerini bul.**
- (v) **Amaç fonksiyonu değerleri kötü olan kromozomları populyasyondan çıkar.**
- (vi) **3-5 arasındaki adımları tekrar et.**

GA çalışmasını açıklamaya yönelik pek çok çalışma yapılmıştır. Bu çalışmalardan en çok bilineni John Holland tarafından ortaya atılan şema (schema ya da schemata) kuramıdır. Şema ikili dizileri göstermek için kullanılan bir gösterimdir. Holland bu kuramı ikili diziler üzerinde temel bir genetik algoritma için tanımlamıştır. Bu yöntemle göre genetik algoritma iyi yapıları ortaya çıkarır, çoğaltır ve birleştirir. Holland'ın şema teoremi hala genetik algoritmaların

başarısını açıklamak için en başarılı teorem olarak kullanılır. Genel olarak bir genetik algoritma ikili kodlama (binary strings) sistemi üzerinde çalışır. Bu kodlamalar doğal sistemlerdeki kodlamalar ile ilgilidir. Genetik algoritmalar doğal genetiklerden kelimeleri ödünç alır.

Genetik algoritmalarda oluşan başarılı bireyler incelenirse, bu bireyler arasındaki benzerlikler bulunabilir. Bu benzerliklerden yola çıkarak şemalar oluşturulabilir. İkili dizi kodlaması için aşağıdaki yöntem önerilebilir: 0,1 ve * (“*” o konumda 0 veya 1 olmasının önemsiz olduğunu gösterir). Örnek olarak ikinci ve dördüncü bitleri 1, altıncı biti 0 olan çözümlerin başarılı olduğu bir toplumda şu şema oluşturulabilir :

* 1 * 1 * 0

Bu şemaya uygun aşağıdaki ikili diziler yazılabilir :

010100, 010110, 011100, 011110, 110100, 110110, 111100, 111110

Görüldüğü gibi şemaların katılması ikili dizilerle gösterilen arama aralığını büyütülmektedir. Arama aralığının büyümesinin sonucun bulunmasını zorlaştırması beklenir ancak durum böyle değildir.

2. GENETİK ALGORİTMA NASIL ÇALIŞIR?

Genetik algoritmanın çalışmasını aşağıdaki adımlarla özetleyebiliriz (Oğuz ve Akbaş,1997:5):

1. Adım : Olası çözümlerin kodlandığı bir çözüm grubu oluşturulur. Çözüm grubuna biyolojideki benzerliği nedeniyle populasyon, çözümlerin kodları da kromozom olarak adlandırılır. Bu adıma populasyonda bulunan birey sayısını belirleyerek başlanır. Bu sayı için bir standart yoktur. Genel olarak önerilen 100-300 aralığında bir büyüklüktür. Büyüklük seçiminde yapılan işlemlerin karmaşıklığı ve aramanın derinliği önemlidir. Populasyon bu işlemde sonra rasgele oluşturulur.

2. Adım : Her kromozomun ne kadar iyi olduğu bulunur. Kromozomların ne kadar iyi olduğunu bulan fonksiyona uygunluk fonksiyonu denir. Bu fonksiyon işletilerek kromozomların uygunluklarının bulunması ise hesaplama (evaluation) adı verilir. Bu fonksiyon genetik algoritmanın beynini oluşturmaktadır. GA’da probleme özel çalışan tek kısım bu fonksiyondur. Çoğu zaman GA’nın başarısı bu fonksiyonun verimli ve hassas olmasına bağlı olmaktadır.

3. Adım : Bu kromozomları eşleyerek yeniden kopyalama ve değiştirme operatörleri uygulanır. Bu sayede yeni bir populasyon oluşturulur. Kromozomların eşlenmesi kromozomların uygunluk değerlerine göre yapılır. Bu seçimi yapmak için rulet tekerleği seçimi, turnuva seçimi gibi seçme yöntemleri vardır.

4. Adım : Yeni kromozomlara yer açmak için eski kromozomlar ortadan kaldırılır. Eski kromozomlar çıkartılarak sabit büyüklükte bir populasyon sağlanır.

5. Adım : Tüm kromozomların uygunlukları tekrar hesaplanır. Tüm kromozomlar yeniden hesaplanarak yeni populasyonun başarısı bulunur.

6. Adım : GA defalarca çalıştırılarak çok sayıda populasyon oluşturulup hesaplanır. Eğer zaman dolmamışsa 3. adıma gidilir.

7. Adım : O ana kadar bulunan en iyi kromozom sonuçtur. Çünkü populasyonların hesaplanmasında en iyi bireyler saklanmıştır.

3. GENETİK OPERATÖRLER

Kullanılan genetik operatörler, varolan populasyon üzerine uygulanan işlemlerdir. Bu işlemlerin amacı daha iyi özelliğe sahip yeni nesiller üretmek ve arama algoritmasının alanını genişletmektir. Farklı uygulamalarda farklı operatörler kullanılmakla birlikte genetik algoritmada 3 standart operatör kullanılır. Bu operatörler:

- Yeniden Üretim (Reproduction)
- Çaprazlama (Crossover)
- Mutasyon (Mutation)

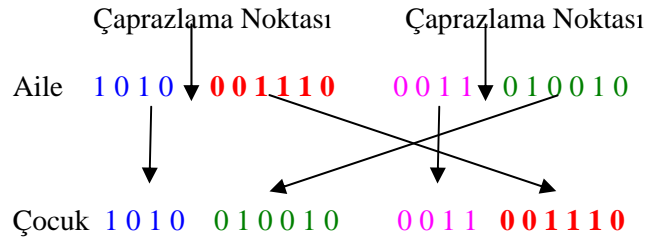
(i) Yeniden Üretim (Reproduction)

Nesil Üretimi (Generational Reproduction): Varolan nesilin yerine yeniden nesil oluşturmak. Daha sonra açıklanacak olan gezgin satıcı probleminde bu üretim kuralı kullanılmıştır.

Kararlı Durum Üretimi(Steady-State Reproduction): Sadece nesildeki birkaç birey yeni bireylerle yer değiştirir. Yeniden üretim sonucunda elde edilen ara nesil, çaprazlama ve mutasyon genetik operatörleri kullanılarak yeni nesil elde etmede kullanılır.

(ii) Çaprazlama (Crossover)

Çaprazlama operatörü GA'lardaki en önemli operatördür. İki çözümün yapıları kullanılarak yeni bir çözüm oluşturulması esasına dayanır. Çaprazlama işlemi genel olarak ikili dizilerin parçalarının değiş tokuşu şeklinde gerçekleştirilir. Farklı uygulamalarda farklı kodlama yöntemleri kullanıldığı için farklı çaprazlama yöntemleri kullanılır, tek noktalı çaprazlama, iki noktalı çaprazlama, ve üniform çaprazlama gibi.

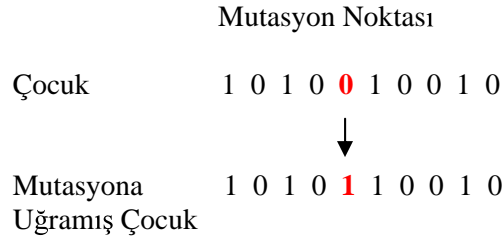


Şekil 1 : Tek Noktalı Çaprazlama

Amaç, Aile(parent) kromozom genlerinin yerini değiştirerek Çocuk(child) kromozomlar üretmek ve böylece varolan uygunluk değeri daha yüksek olan kromozomlar elde etmektedir.

(iii) Mutasyon (Mutation)

Mutasyon GA'lardaki operasyonda karar verici olarak ikinci derecede rol oynar. Amaç, varolan bir kromozomun genlerinin bir ya da bir kaçının yerlerini değiştirerek yeni kromozom oluşturmaktır. Yeniden ve sürekli yeni nesil üretimi sonucunda belirli bir süre sonra nesildeki kromozomlar birbirini tekrarlama konumuna gelebilir ve bunun sonucunda farklı kromozom üretimi durabilir veya çok azalabilir. İşte bu sebeple nesildeki kromozomların çeşitliliğini arttırmak için kromozomlardan bazıları mutasyona uğrattılır.



Şekil 2 : Tek Bir Mutasyon

4. GENETİK ALGORİTMA İLE GEZGİN SATICI PROBLEMİNİN ÇÖZÜLMESİ

GA kullanılarak çözülen problemlerden en günceli gezgin satıcı problemi (travelling salesman problem) dir. Her ne kadar bu problem ve benzerleri için çok sayıda çözüm yöntemi önerilse de bu tür problemleri – büyüklükleri ne olursa olsun – çözebilecek kesin bir yöntem bulunamadı. Bu nedenle araştırmacılar bilgisayarları daha etkin kullanmak amacıyla bazı sezgisel yöntemler geliştirdiler. Evrimsel programlama yöntemlerinden GA'larda bu yöntemlerden birisidir. Problemin çözümünde genel olarak genetik algoritma esas alınsa da uygulamada farklılıklar olduğundan dikkat çekilmesi gerekir. Çözümün esası genetik operatörlerin kullanılmasıdır fakat, bu operatörler özüne bağlı kalınarak değişik şekilde kullanılmıştır.

Gezgin satıcı problemi, başladığı noktaya tekrar geri dönmek koşulu ile n-1 sayıda başka yerleşim yerlerine uğrayan satıcı ile ilgilidir. Amaç, sayısı n olan yerlerden bir satıcının n-1 sayıdaki kente en kısa sürede uğrayarak başladığı kente dönmesini sağlayacak bir gezi planını hazırlamaktır. Tüm şehirler birbirine bağlı ve satıcı istediği şehirden başlayabilir. Bu durumda problemin çözümü için n! kadar diziliş olması gerekir. Örneğin 5 şehir varsa tek yapılması gereken $5! = 120$ tane diziliş çıkarmak ve uzunlukları tek tek hesaplamaktır (Lalena,1998:1). Genetik algoritma daha kısa zamanda çözüm bulmak için kullanılabilir. Muhtemelen en iyi çözümü bulmamasına rağmen bir dakikadan daha kısa

zamanda mükemmele yakın bir sonuç bulabilir. Sunulan çözümde her harf bir çözümü göstermek üzere (Doyranlı,1999:7):

A : Adana
B : Bursa
D : Diyarbakır
I : İstanbul

.

Üretilen kromozomların yapısı örneği şu şekilde olsun: STYPKMNVDZES

Problemin tanımı şu şekildedir: Her şehre sadece bir kere uğranmalı ve başlangıç şehrine geri dönüldüğünde kullanılacak yol en kısa mesafe olmalıdır.

(i) Yeniden Üretim

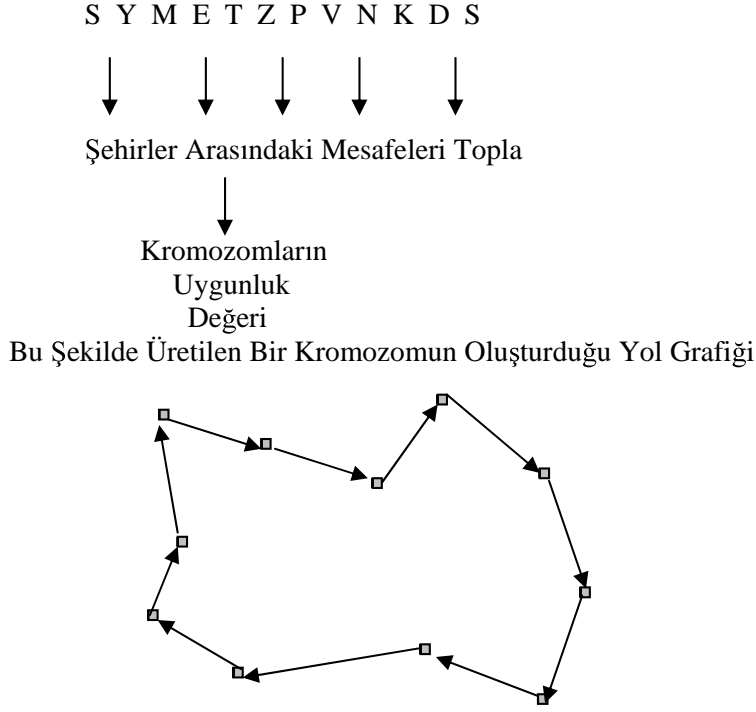
Problemin çözümünde ilk olarak başlangıç popülasyonu oluşturulmalıdır. Başlangıç popülasyonunu oluşturmada kullanılan yöntem aşağıda açıklanmıştır. Başlangıç şehri belirlendikten sonra, kromozomların yapısı şu şekilde olacaktır. S başlangıç şehri olmak üzere kromozomun ilk ve son genine bu şehir yerleştirilir ve aradaki şehirler de rasgele ve sadece kromozomda bir kez yer alacak şekilde istenen popülasyon sayısı kadar rasgele oluşturulur.

STYPKMNVDZES
SZPMVDYNETKS
SYMETZPVNKDS

.

SMYEZTNVDKPS

Bilindiği gibi kromozomlar yeniden oluşturulurken bir uygunluk fonksiyonuna göre seçilir ve yeni nesile aktarılır burada kullanılacak olan uygunluk: En küçük değere sahip olan kromozomdur. Kromozomun uygunluğu tur mesafesi yani şehirler arasındaki uzaklıkların toplamıdır (Doyranlı,1999:5).



Şekil 3: En Kısa Yol Problemi

Bir sonraki popülasyonu elde etmek için kullanılan yöntem: Eğer, en iyi uygunluk değerine sahip olan ve yeni nesli de kullanılmak istenen kromozom sayısı 0 dan farklı olarak belirtildiyse, yeni popülasyona ilk olarak istenen sayıdaki kromozom hiçbir genetik operatöre tabi tutulmada aktarılır. Birkaç üretimden sonra uygunluk değeri yüksek popülasyon üyeleri elde edilmek isteniyorsa bu değer sıfırdan farklı girilmelidir. Fakat bunun bir sakıncası da belli bir süre sonra popülasyon üyelerinin birbirine benzeme olasılığının artmasıdır ki bu durum optimum çözümü bulmayı zorlaştıracaktır. Böyle bir durumdan kurtulmak için yapılması gereken çaprazlama ve mutasyona uğrama yüzdesini büyük tutmaktır.

En iyi kromozomlar seçildikten sonra bir önceki popülasyondaki diğer kromozomlar verilen çaprazlama ve mutasyona uğrama yüzdelere bağlı olarak bu iki işlem ile yeniden yeni bir popülasyon üretilir. Unutulmaması gereken bir önemli nokta da bir kromozomun ya da kromozom çiftinin verilen olasılıklar dahilinde herhangi bir genetik operatöre uğramadan yeni popülasyona aktarılacağıdır.

Yeniden üretimde, yeni popülasyon oluşturmak için kullanılan yöntem “yüzde alanlı tekerlek çevirme” yöntemidir. Eski popülasyondan seçilecek olan kromozom bu yöntem ile belirlenerek ve çaprazlama ve mutasyon yüzdeleri göz önüne alınarak yeni popülasyona eklenir.

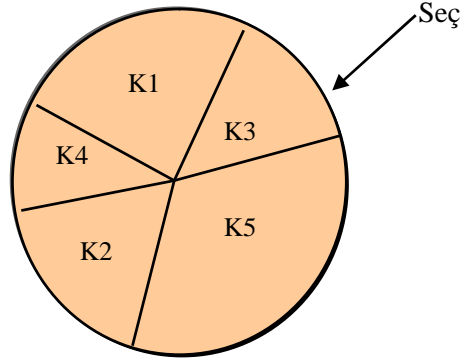
Populasyon sayısı 5 için aşağıda uygunluk değerleri verilmektedir:

<u>Kromozomlar</u>		<u>Uygunluk Değeri</u>	<u>Seçim</u>
STYPKMNVDZES	K1	3500 km	%13,75
SZPMVDYNETKS	K2	4759 km	%18,60
SYMETZPVNKDS	K3	5970 km	%23,45
SMYEZTNVDKDS	K4	8637 km	%33,93
SZPEVTNMDKYS	K5	2591 km	%10,18

$$\text{Toplam Uygunluk Değeri} = K1+K2+K3+K4+K5 = 25457$$

$$\text{Uygunluk Yüzdesi} = \text{Uygunluk Değeri} / \text{Toplam Uygunluk Değeri}$$

olduğuna göre her bir kromozomun tekerlekte yer alma yüzdeleri en küçük uygunluk değeri en yüksek yüzdeye sahip olacak şekilde tekerleğe yerleştirilir.



Şekil 4: Rulet Tekerleği Örneği

Rulet Tekerleği seçimi aşağıda açıklanmıştır:

- Tüm bireylerin uygunluk değerleri bir tabloya yazılır,
- Bu değerler toplanır,
- Tüm bireylerin uygunluk değerleri toplama bölünerek [0,1] aralığında sayılar elde edilir. Bu sayılar bireylerin seçilme olasılıklarıdır.

-Seçilme olasılıklarını tuttuğumuz tablodaki sayılar birbirine eklenerek rasgele bir sayıya kadar ilerlenir. Bu sayıya ulaşıldığında ya da geçildiğinde son eklenen sayının ait olduğu çözüm seçilmiş olur.

Yeni popülasyona kromozom seçimi bu tekerlek döndürülerek yapılır. Seç oku hangi kromozom yüzde alanına geldi ise o kromozom seçilerek ve genetik operatörler kullanılarak yeni popülasyona eklenir. Daha önceden de bahsedildiği gibi çaprazlama genelde gezgin satıcı probleminin çözümünde uygulanmayan bir yöntemdir. Fakat burada farklı bir genetik operatöre aykırı olmayan çaprazlama kullanılmış ve etkili bir yöntem olduğu belirlenmiştir.

(ii) Çaprazlama(Crossover)

Yüzde tekerlek yöntemi ile seçilen iki kromozom eğer çaprazlama yüzdesi nedeni ile çaprazlamaya uğrayacağına karar verildiyse çaprazlamaya girecektir.

İki kromozomun çaprazlanması ve bundan yeni bir çocuk kromozomun elde edilmesi şu şekilde olmaktadır .

- Kromozom boyu ile aynı uzunlukta sıfır ve birlerden oluşan bir bit katarı rasgele (ilk ve son gen 1 olmak şartı ile) üret,
- Üretilen bu bit katarı ile ana(aile1) kromozomu eşle. Ana kromozomda 1'lere karşılık gelen genleri aynı pozisyonda olacak şekilde çocuk kromozoma kopyala, 0'lara karşılık gelenleri aynı pozisyonda olacak şekilde çocuk kromozoma kopyala, 0'lara karşılık gelenleri çocuk kromozomda aynı sırada boş bırak,
- İlk ana kromozomdaki 0 lara denk gelen genlerden bir sıralı dizi oluştur,
- Bu dizide yer alan genlerin sırasını ikinci ana kromozomda (aile2) görüldüğü şekilde yer değiştir,
- Yeni sırada oluşan listeyi aynı sıradan çocuk kromozomda boş bırakılan genlere yerleştir.

Çaprazlama için Aile 1 ve Aile 2 aşağıdaki şekilde verilmiş olsun,

Aile 1 : S E G T H L P O B M S

Aile 2 : S B T P E M G L H O S

Üretilen : 1 0 1 0 1 1 0 0 1 0 1

İkinci adımdan sonra :

Çocuk : S-G-HL- -B-S

Dizi : E T P O M Aile 2'ye göre yer değiştirir → T P E M O ve yer değiştirilen bu diziyi çocuk kromozoma aynı sırada yerleştir. Bu durumda çaprazlama sonucu oluşan yeni kromozom ;

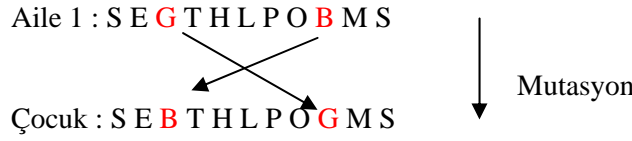
Çocuk : S T G P H L E M B O S

şeklinde olacaktır.

Kullanılan bu çaprazlama yöntemi ile oluşan yeni kromozomda aynı şehire sadece bir kez uğrama şartı da sağlanmıştır. Çaprazlamada ilk ve son genler kullanılması bunu sağlamıştır.

(iii) Mutasyon(Mutation)

Bir genin mutasyona uğraması herhangi iki genin yer değiştirmesidir. Algoritmada kullanılan yöntemde, bir kromozomun birden fazla geni mutasyona uğrayabildiği gibi mutasyon yüzdesinden dolayı bir kromozom mutasyona uğramadan yeni popülasyona eklenebilir.



Daha öncede söz edildiği gibi mutasyonun amacı optimum yolu bulmaya çalışırken popülasyondaki kromozom çeşitliliğini sağlayarak arama uzayını yaymak, genişletmektir.

Genetik algoritma yöntemi ile çözüm için TSP Solver programı kullanılmaktadır.

Ekte verilen ilk örnek dikkate alınırsa,

$$X : 27 \ 49 \ 30 \ 23 \ 59$$

$$Y : 35 \ 41 \ 68 \ 85 \ 73$$

Generation: 2600 Value: 155 olarak bulunmuştur. Tablo 1'de verilen uzaklıklar matrisi aşağıdaki gibi hesaplanmıştır.

$$x_{13} = (27 - 30)^2 + (35 - 68)^2 = 9 + 1089 = \sqrt{1098} = 33,14$$

$$x_{12} = (27 - 49)^2 + (35 - 41)^2 = 484 + 36 = \sqrt{520} = 22,80$$

·
·
·

$$x_{45} = (23 - 59)^2 + (85 - 73)^2 = 144 + 1296 = \sqrt{1440} = 37,95$$

Tablo 1: Uzaklıklar Matrisi

-	22,80	33,14	50,16	49,68
22,80	-	33,01	51,12	33,53
33,14	33,01	-	18,38	29,43
50,16	51,12	18,38	-	37,95
49,68	33,53	29,43	37,95	-

Gezgin satıcı problemi için optimale yakın sonuç veren yöntemlerden biri en yakın yaklaşım yöntemidir(Öztürk,2001:244). Tablo 1’de verilen uzaklıklar matrisi kullanılarak bulunan en yakın yaklaşım yöntemi ile çözümü aşağıda verilmiştir. 1. Adımda en yakın yaklaşım yönteminde maliyet matrisindeki boş gözelerle M gibi çok büyük bir değer atanır. Maliyet matrisindeki en küçük değerli eleman Tablo 2’de 18,38 değeri gezi planının halkasına eklenir bunun satırında ve sütündeki diğer değerlerin yerine M yerleştirilerek yeni maliyet matrisi oluşturulur. 2. Adım olarak Tablo 3’de en küçük değer 22.80 değeri gezi planına eklenerek yine bunun satır ve sütununa karşılık gelen değerlere M atanır. Benzer işlemler yapıldıktan sonra elde edilen optimal çözüm Tablo 4’de verilmektedir.

Tablo 2: En Yakın Yaklaşım Yöntemi 1.Adım

-	22,80	33,14	M	49,68
22,80	-	33,01	M	33,53
M	M	-	18,38	M
50,16	51,12	18,38	-	37,95
49,68	33,53	29,43	M	-

Tablo 3: En Yakın Yaklaşım Yöntemi 2.Adım

-	22,80	M	M	M
22,80	-	33,01	M	33,53
M	M	-	18,38	M
50,16	M	18,38	-	37,95
49,68	M	29,43	M	-

Tablo 4: En Yakın Yaklaşım Yöntemi Optimal Çözüm

-	22,80	M	M	M
M	-	M	M	33,53
M	M	-	18,38	M
50,16	M	M	-	M
M	M	29,43	M	-

En yakın yaklaşım yöntemi ile $1 \rightarrow 2 \rightarrow 5 \rightarrow 3 \rightarrow 4 \rightarrow 1$
($22,80+33,53+18,38+50,16+29,43=154,3$) sonucu elde edilmiştir.

Gezgin satıcı probleminin WinQsb paket programı çözümleri;

En yakın komşu heuristik çözüm, en ucuz ekleme heuristik çözüm, iki-yollu değişim geliştirme heuristik çözüm, dal ve sınır yöntemidir.

Tablo 5: En Yakın Komşu Heuristik Çözüm (Nearest Neighbor Heuristic)

01-21-2004	From Node	Connect To	Distance/Cost		From Node	Connect To	Distance/Cost
1	Node1	Node2	22,8	4	Node4	Node5	37,95
2	Node2	Node3	33,01	5	Node5	Node1	49,68
3	Node3	Node4	18,38				
	Total	Minimal	Traveling	Distance	or Cost	=	161,82
	(Result	from	Nearest	Neighbor	Heuristic)		

Tablo 5’de WinQsb paket programı ile en yakın komşu heuristik çözüm 161,82 olarak bulunmuştur.

Tablo 6: En Ucuz Ekleme Heuristik Çözüm (Cheapest Insertion Heuristic):

01-21-2004	From Node	Connect To	Distance/Cost		From Node	Connect To	Distance/Cost
1	Node3	Node1	33,14	4	Node5	Node4	37,95
2	Node1	Node2	22,8	5	Node4	Node3	18,38
3	Node2	Node5	33,53				
	Total	Minimal	Traveling	Distance	or Cost	=	145,80
	(Result	from	Cheapest	Insertion	Heuristic)		

Tablo 6’da WinQsb paket programı ile en ucuz ekleme heuristik çözüm 145,80 elde edilmektedir.

Tablo 7: İki-Yollu Değişim Geliştirme Heuristik Çözüm (Two-way Exchange Improvement Heuristic)

01-21-2004	From Node	Connect To	Distance/Cost		From Node	Connect To	Distance/Cost
1	Node3	Node1	33,14	4	Node5	Node4	37,95
2	Node1	Node2	22,8	5	Node4	Node3	18,38
3	Node2	Node5	33,53				
	Total	Minimal	Traveling	Distance	or Cost	=	145,80
	(Result	from	Two-way	Exchange	Improvement	Heuristic)	

WinQsb paket programı ile iki-yollu değişim geliştirme heuristik çözüm sonucu, en ucuz ekleme heuristik çözümde olduğu gibi 145,80 olarak bulunmuştur.

Tablo 8: Dal ve Sınır Yöntemi (Branch and Bound Method)

01-21-2004	From Node	Connect To	Distance/Cost		From Node	Connect To	Distance/Cost
1	Node1	Node2	22,8	4	Node4	Node3	18,38
2	Node2	Node5	33,53	5	Node3	Node1	33,14
3	Node5	Node4	37,95				
	Total	Minimal	Traveling	Distance	or Cost	=	145,80
	(Result	from	Branch	and	Bound	Method)	

Tablo 8’de WinQsb paket programı ile dal ve sınır çözüm yöntemi ile yine 145,80 sonucu elde edilmektedir.

GA çözüm için Tablo 1’de verilen uzaklıklar matrisi kullanılmaktadır. Ek de 5 nokta için şekil olarak da görülmektedir. Problemi açıklamak için sadece 5 nokta seçilmiştir. Seçilen 5 nokta için klasik yöntemler ve genetik algoritma çözümü birbirine yakın sonuçlar vermektedir. GA özellikle nokta sayısı arttıkça daha kısa sürede optimale yakın bir çözüm bulmak amacıyla kullanılabilir. Ekte 12, 30, 50 ve 75 nokta için bulunan çözümler verilmektedir. Jenerasyon sayısı arttıkça daha iyi sonuçlar da bulunabilmektedir fakat, belli bir süre sonra daha iyi sonuç bulamayacağı için jenerasyon sayısının artması sonucu değiştirmeyecektir.

5. SONUÇ

GA’lar sezgisel bir teknik olarak, yöneylem araştırması problemlerinde kullanımı giderek artmaktadır. GA’lar yaklaşımlarının kullanılmasıyla yöneylem araştırması alanında ortaya çıkan en iyileme problemlerinin çözümüne ilişkin alternatif yöntemler ortaya atılmış ve bu yeni yöntemlerin bazı problemlerin çözümünde kullanılması, uygulamaya açık olan diğer problemlere de farklı bir boyut kazandırmıştır.

Hasşerbetçi(1997) yöneylem araştırmasında yer alan klasik problemlerin çözümleri için kullanılan GA’lar içeren bir çok çalışma olmasına karşın, pratikte karşılaşılan yöneylem araştırması problemlerinin çözümünde GA tekniğinin performansının hala yeterli olmadığını ifade etmiştir. Birbirleri ile ilişkilendirilmiş küçük çaplı problemlerde, deneysel araştırmaya dönük çalışmalar GA tekniği baz alınarak uygulanmaya başlamıştır.

Yukarıdaki örnekte de görüldüğü gibi genetik algoritma çözümü ile optimuma yakın sonuç bulunabilmektedir. Özellikle nokta sayısı arttıkça kısa sürede çözüm bulabilmek için bu yöntem daha etkili sonuçlar verebilmektedir. GA’lar problemin kendisi hakkında az bilgiye ihtiyaç duyduğundan caziptirler, bu algoritmalara dayanan çözümler karmaşık dinamik yapıların optimizasyonu için daha uygundur.

KAYNAKLAR

- CHAMBERS, L.D. (1999), *Practical Handbook of Genetic Algorithms :Complex Coding Systems* Volume III, CRC Press, 572 p.
- DOYRANLI, Ş.(1999), *Genetik Algoritmalar ile Seyahat Eden Satıcı Probleminin Çözülmesi*, <<http://www.doruk~.net.tr/~Arifprogramlar/delphiT>>
- GOLDBERG, D. E. (1989), *Genetic Algorithms in Search, Optimization, and Machine Learning* Reading, MA : Addison-Wisley, 412 p.
- HASŞERBETÇİ, H., K. (1997), *Genetik Algoritmaların Yöneylem Araştırmasında Kullanılması*, Yüksek Lisans Tezi, İ.Ü İşletme Bölümü, 53 s.
- KAYA, M.(1999), *Genetik Algoritma ve Gezgin Satıcı Probleminin Çözümü*, Fırat Üniversitesi FBE Yüksek Lisans Tezi.
- KARR, L. C., FREEMAN L. M. (1999), *Industrial Applications of Genetic Algorithms*, CRC Prees. 350 p.
- KÖMÜR, M. ve ALTAN, M.(2001), *Betonarme Bir Kiriş ve Kolonun Genetik Algoritma ile Optimum Tasarımı*, Mühendislikte Modern Yöntemler Sempozyum Kitabı .
- LALENA, M.(1998), *Travelling Salesman Problem Using Genetic Algorithms* <<http://www.lalena.com/ai/tsp/>>
- MİCHALEWICZ, Z. (1992), *Genetic Algorithms + Data Structures = Evolution Programs*, Springer-Verlag, 250 p.
- MİTCHELL, M. VE FOREST S.(1994), *Genetic Algorithms and Artificial Life*. Vol. 1, No. 3, pp. 267-289. Reprinted in C. G. Langton (Ed.) *Artificial Life: an Overview*, MIT Press, Cambridge, MA (1995).
- OBİTKO, M., (1998), *Introduction to Genetic Algorithms*, <<http://cs.felk.cvut.cz/~xobitko/ga/>>
- OĞUZ, M. ve AKBAŞ, S.(1997), *Genetik Algoritmalar*, YTÜ Endüstri Mühendisliği, Bitirme Tezi. <<http://artemis.efes.net/moguz/>>
- ÖZTÜRK, A.(2001), *Yöneylem Araştırması*, Genişletilmiş 7. Basım, Ekin Kitabevi Yayınları, Bursa.
- PORTMANN M.C.(1996), *Genetic Algorithms and Scheduling*, A State of Art and Some Propositions, Workshop on Production Planning and Control, Mons, Belçika.

EKLER

