

# STEINER MİNİMUM YAYILAN AĞAÇ PROBLEMİNİN ÇÖZÜMÜNDE GENETİK ALGORİTMANIN KULLANILMASI

Öznur İŞÇİ\*, Serdar KORUKOĞLU\*\*

\*Muğla Üniversitesi, Fen Edebiyat Fakültesi, İstatistik Bölümü, 48170/Muğla

\*\*Ege Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, 35000/İzmir

Geliş Tarihi : 06.12.2005

## ÖZET

Genetik algoritmalar (GA) stokastik bir arama yöntemidir ve bu nedenle optimum ya da optimuma yakın çözümler verebilmektedir. GA son yıllarda gezgin satıcı problemi, karesel atama, yerleşim, atölye çizelgeleme, ders/sınav programının hazırlanması, iletişim şebekelerinin tasarımı, montaj hattı dengeleme, en küçük yayılan ağaç problemi ve benzeri bir çok kombinatoriyal optimizasyon problemlerine başarılı bir şekilde uygulanabilmektedir. Bu çalışmada Steiner minimum yayılan ağaç probleminin genetik algoritma ile çözümü için java programı geliştirilerek performansı incelenmiştir. Literatürde daha önce çözülmüş test problemleri üzerinde yapılan denemeler sonucunda bu çalışmada önerilen GA yaklaşımı ile optimuma yakın sonuçlar elde edilmektedir. Çalışmada belirlenen noktalar için Steiner minimum yayılan ağaç problemi ve minimum yayılan ağaç problemi için uzunluklar ve kazanç değerleri hesaplatılmıştır.

**Anahtar Kelimeler :** Genetik algoritma, Minimum yayılan ağaç problemi, Steiner minimum yayılan ağaç problemi

## USING GENETIC ALGORITHM TO SOLVE STEINER MINIMUM SPANNING TREE PROBLEM

### ABSTRACT

Genetic algorithms (GA) are a stochastic search methods, and they produce solutions that are close to optimum or near optimum. In addition to GA's successful application to traveling salesman problem, square designation, allocation, workshop table, preparation of lesson/examination schedules, planning of communication networks, assembling line balanced, minimum spanning tree type many combinatorial optimization problems it would be applicable to make the best comparison in optimization. In this study a Java program is developed to solve Steiner minimum spanning tree problem by genetic algorithm and its performance is examined. According to the tests carried out on the problems that were given before in the literature, results that are close to optimum are obtained in by GA approach that is recommended in this study. For the predetermined points in the study, length and gain are calculated for Steiner minimum spanning tree problem and minimum spanning tree problem.

**Key Words :** Genetic algorithm, Minimum spanning tree problem, Steiner minimum spanning tree problem

## 1. GİRİŞ

Genetik algoritma doğadaki evrim yöntemlerini kullanan bir arama yöntemidir. Başka bir ifade ile

doğadaki evrim mekanizmasının bilgisayar ortamındaki benzeşimidir. Darwin'in "en iyi olan yaşar" prensibine dayalı olarak biyolojik sistemlerin gelişim sürecini taklit eden genetik algoritmalar ilk defa Holland (1975) tarafından önerilmiştir.

Özellikle son yıllarda diğer bazı sezgisellerle oluşturulan melez yaklaşımları ise GA'nın uygulamalardaki başarısını artırarak uygulama alanını genişletmektedir (Altıparmak ve Dengiz, 1998).

Geleneksel optimizasyon yöntemlerine göre farklılıkları olan genetik algoritmalar, parametre kümesini değil kodlanmış biçimlerini kullanırlar. Olasılık kurallarına göre çalışan genetik algoritmalar, yalnızca amaç fonksiyonuna gereksinim duyar. Çözüm uzayının tamamını değil belirli bir kısmını tararlar. Böylece, etkin arama yaparak çözüme ulaşırlar (Goldberg, 1989). Diğer bir önemli üstünlükleri ise çözümlerden oluşan yığını eş zamanlı incelemeleri ve böylelikle yerel en iyi çözümlere takılmamalarıdır. GA'nın her problem için performansı farklıdır. Bir problemde GA ile en iyi ya da en iyiye yakın sonuç elde edilebilirken diğer bir problemde başka bir tümel arama tekniği ile en iyi ya da en iyiye yakın sonuç bulunabilir.

GA doğada geçerli olan en iyinin yaşaması kuralına dayanarak sürekli iyileşen çözümler üretir. Bunun için "iyi"nin ne olduğunu belirleyen bir uygunluk fonksiyonu ve yeni çözümler üretmek için yeniden üretim, çaprazlama ve mutasyon gibi operatörleri kullanırlar. Genetik algoritmaların bir diğer özelliği de bir grup çözümle uğraşmasıdır. Bu sayede çok sayıda çözümün içinden iyileri seçilip kötülerini de elenebilir (Oğuz ve Akbaş, 1999).

## 2. MİNİMUM YAYILAN AĞAÇ PROBLEMİ

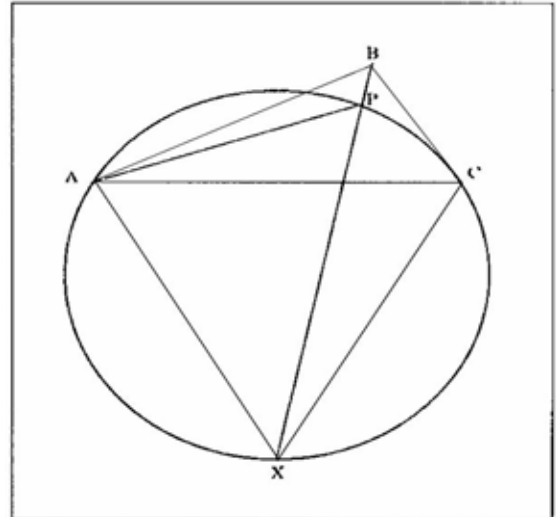
Minimum yayılan ağaç problemi (Minimum Spanning Tree Problem-MST) en iyi bilinen şebeke optimizasyon problemlerinden biridir. Bu problem, birkaç coğrafik bölgeye dağılmış şehirler arasında kurulacak iletişim şebekesinin topoloji tasarımı veya doğalgaz boru hattı, su şebekesi gibi düşünülebilir. Her bir şehir arasında yerleştirilen kabloların maliyeti önceden belirlenmelidir. Problemin amacı fiber-optik kablo şebekesini oluşturmak ve en az maliyetle tüm şehirler arasındaki bağlantı planının yapılmasıdır. Bu örnek için grafik teorisinde şehirler düğümler (nodes) ve kablolarla hatlar (edge) olarak adlandırılır (Dong, 2002).

Problemin formülasyonu:  $G=(V,E)$  şeklindedir. Burada  $V=\{v_1, v_2, \dots, v_m\}$  düğümlerini (şehirlerin) listesini ve  $E=\{e_1, e_2, \dots, e_m\}$  hatları göstermektedir. Her bir hat negatif olmayan bir ağırlıkla birleştirilir ve  $W=\{w_1, w_2, \dots, w_m\}$  şeklinde gösterilir. Minimum yayılan ağaç problemi polinom zamanda çözülebilen

bir problemidir. Ancak çeşitli kısıtlar (kapasite, erteleme, tepe kısıtları, v.b) olduğunda NP-hard bir problem haline dönüşmektedir. Bunlardan her bir problem üzerinde değişik çalışmalar vardır. Genellikle yapılan çalışmalar alt ve üst sınır geliştirme şeklindedir. Minimum yayılan ağaç probleminin çözümü için üç algoritma kullanılmaktadır. Bunlar, Kruskal Algoritması (1956), Prim Algoritması (1957) ve Boruvka Algoritması (1926) dir (Zhou and Gen, 1999).

## 3. STEİNER MİNİMUM YAYILAN AĞAÇ PROBLEMİ

Şebeke uzunluğunu minimize etmek matematikteki en eski optimizasyon problemlerinden biridir (Jones and Harris, 1996). 17. yüzyılın ortalarında minimum yayılan ağaç problemi basit bir problem olarak görülmüştür. Problemin çözümü için, düzlemde verilen 3 noktanın her biri için P'den uzaklıkların toplamını minimize eden P noktası bulunmaktadır. 3 nokta problemi için matematikçilerin teklif ettiği yöntem Şekil 1'de verilmektedir. Bu yöntemde verilen A, B ve C noktaları Steiner noktalarını hesaplamak için oluşturulmaktadır. (AC) iki nokta arasında (B) gibi üçgenin dışında üçüncü bir kenar arasında en uzun kenar kullanılarak (ACX) eşkenar üçgene yerleştirilir. Üçgen etrafında bir çember oluşturulur ve çizgi üçüncü nokta (B)'den üçgendeki en uzak kenar (X) geometrik olarak çizilir. Steiner noktasının yeri (P) çember ile (BX) bu çizginin kesiştiği yerdedir (Harris, 1995).



Şekil 1.  $AP+CP = PX$

19. yüzyılda Berlin üniversitesinde bir matematikçi olan Jacob Steiner minimum yayılan ağaç problemi üzerinde çalışarak, yüzeydeki noktalar setinin keyfi

büyüklerdeki bir setini içine alarak genelleştirmiştir. Bu genelleştirme, küme probleminin iki boyutlu merkezi için geometrik bir yaklaşımdır. Steiner geometrik şekiller ile onların izdüşümleri arasındaki ilişkiler ve en küçük ağ üzerinde çalışarak bu gün Steiner yayılan ağaç problemi olarak bildiğimiz problemi bulmasıyla tanınmaktadır. Bu problemde "Steiner noktaları" olarak adlandırılan tüm noktalar, minimum bir yol oluşturacak şekilde çok sayıda noktalar seti birbirine bağlanarak oluşturulur.

Steiner minimum yayılan ağaç (SMT) probleminin, çok sayıda şehir arasındaki telefon kablolarını çekmek, noktalar arasında boruların yerleşimi, gerekli kablo miktarını en az kullanarak çok sayıda bilgisayar arasında bağlantı kurmak v.b birçok uygulama alanı vardır.

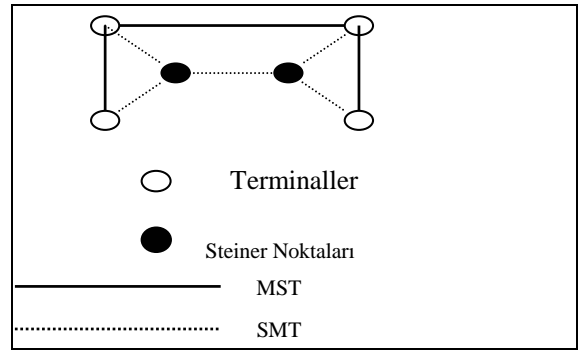
Steiner minimum yayılan ağaç problemi bir karayolu sistemi geliştirmek için uygulanabilir (Clarke, 2002). Bu nedenle alınan üç şehir bakım ve inşaa maliyetleri minimum olacak şekilde birleştirilmek istenirse, eğer dağlar, nehirler ya da diğer engeller yol üzerinde yoksa iki nokta arasındaki en kısa yol bir hat olduğundan yollar düz olacaktır. Steiner, bunun için verilen bir orijinal nokta setlerini birbirine bağlayacak ilave noktalar (Steiner noktaları) ekleyerek toplam minimum yayılan ağacı bulmaktadır. Steiner noktalarını seçmek için Brute-Force yaklaşımı ve rasgele yaklaşım olmak üzere iki yaklaşım kullanılmaktadır (Senior, 2002).

Brute - Force Yaklaşımı: Orijinal noktaların minimum, maksimum x ve y koordinatları bulunur. Bunlar potansiyel Steiner noktalarını araştırmak için kullanılır. Sonra orijinal noktaların MST ağırlığı bulunur. Steiner noktalarının ortasına eklenir. Belirli bir zamanda potansiyel noktalar minimum, maksimum x ve y değerlerini oluşturmak için bir iskelet oluşturur, daha sonra sürekli bir nokta almır ve ağacın toplam uzunluğunu minimize etmek için başka bir nokta bulunur. En düşük ağırlıklar ve koşullar (Steiner noktalarının yeri ve sayısı) dikkatle izlenerek minimum ağırlıklar hesaplanır. n-2 noktanın her biri için tüm potansiyel noktalar çalışılır, minimum toplam uzunluğunu hesaplayan koşullar yeniden oluşturulur.

Rasgele Yaklaşım : Optimal Steiner noktalarının yerini tahmin etmek için kullanılır. Bu algoritma ilkinden tasarlanır. Bu nedenle Brute-Force yaklaşımına çok benzerdir. Önce potansiyel Steiner noktaları gibi her bir noktayı düşünmek yerine, tabakalı rasgele örnekleme kullanılarak potansiyel Steiner noktaları seçilir. Belirli bir nokta seçmek için rasgele bir x koordinatı seçerek, minimum ve maksimum değerlerini iskelet içindeki y koordinatların sırası belirlenir.

### 3. 1. Steiner Minimum Yayılan Ağaç Probleminin Genetik Algoritmayla Çözümü

Steiner minimum yayılan ağaç problemi klasik kombinatoriyal optimizasyon problemidir. Bu nedenle polinom zamanda çözülemeyen tam (NP-complete) problemler arasında gösterilmektedir. Steiner minimum yayılan ağaç probleminin polinomiyal zamanda kesin çözümü yoktur. Öklit düzleminde  $|V| = n$  olduğunda V terminalerin setini vermektedir. Bu terminaller arasındaki koordinatların aralığı hem x hem de y'de  $[0,1]$  arasındadır (Starsky, 2001). Şekil 2'de minimum yayılan ağaç ve Steiner minimum yayılan ağaç gösterilmektedir.



Şekil 2. MST ve SMT

#### 3. 1. 1. Kromozom Kodlama

Steiner minimum yayılan ağaç probleminde Steiner noktalarının tam sayısının bilinmesi gerekli değildir. Çünkü en fazla n-2 sayıda Steiner noktasının olması gerektiği bilinmektedir (Starsky, 2001). Şekil 3'de kromozom kodlaması görülmektedir.  $x_{c,i}$  c kromozomundaki i.nci x koordinatını ifade etmektedir.

$\langle x_{c,1}, y_{c,1} \rangle$	$\langle x_{c,1}, y_{c,1} \rangle$	.....	$\langle x_{c,k}, y_{c,k} \rangle$
$k \leq n$			

Şekil 3. Kromozom kodlama

#### 3. 1. 2. Uygunluk Fonksiyonu

i. nesildeki bir c kromozomunun uygunluk fonksiyonu  $f_{c,i}$  olmak üzere,

$$f_{c,i} = smtr_{c,i} + dr_{c,i} \quad (1)$$

$smtr_{c,i}$  : i nesildeki c kromozomun toplam Steiner ağaç uzunluğunun sıralamasıdır. Bu sıralama, p her bir nesildeki yığın sayısı iken 1'den p'ye yapılır. Sıralama 1 en kısa toplam uzunluğu verir. Sıralama p, en uzun toplam uzunluk olacaktır (Starsky, 2001).

$$dr_{c,i} = \sum_{i=1}^p \sum_{j=1}^k \sum_{l=1}^k |X_{c,l} - X_{i,j}| + |Y_{c,l} - Y_{i,j}| \quad (2)$$

$dr_{c,i}$  : Benzer olmayanların sıralaması.

Yığın sayısı kadar 1'den p'ye kromozomlar sıralanır. Sıralama 1 ile  $dr_{c,i}$ 'nin en yüksek değeri ve sıralama p ile  $dr_{c,i}$ 'nin en düşük değeri olacak şekilde sıralanacaktır.  $dr_{c,i}$ 'nin amacı kromozomlar içindeki Steiner noktalarını farklı tutmaktır. Eğer kromozomlar içindeki noktalar çok benzer, yakın olursa doğru uzay üzerinde çalışılmamış olunur.  $smtr_{c,i}$ 'nin amacı ise terminallere en kısa mesafedeki kromozomları bulmaktır. Bu nedenle  $smtr$  kromozomlardaki Steiner noktalarını terminallere en yakın mesafede tutar, böylece yakın olmayan kromozomlar atılır. Bu da kromozomları en iyiden en kötüye sıralayıp en tepedeki %70'i olmaktadır.  $dr$  ise kromozomları kendi içinde karşılaştırır ve birbirlerinden farklı olmalarını sağlar. Böylece birbirinden en farklı olan en tepe de olacaktır. Steiner minimum yayılan ağaç probleminin çözümünde kullanılan genetik operatörler aşağıda verilmektedir.

### 3. 1. 3. Genetik Operatörler

Kullanılan genetik operatörler, başlangıçta bilinmeyen bir arama uzayından topladığı bilgileri yığıp, daha sonraki aramaları alt arama uzaylarına yönlendirmek için kullanılır. Çaprazlama ve mutasyon genetik algoritmanın en önemli parçasıdır. Genetik algoritmanın performansı büyük ölçüde bu iki operatöre bağlıdır.

Yeniden Üretim : Başlangıç yığını oluşturulduktan sonra algoritmanın her çevriminde, yeni yığının dizileri bir olasılıklı seçim süreci ile mevcut yığının dizileri arasından seçilir. Yüksek uygunluk değerine sahip diziler, yeni dizilerin elde edilmesinde yüksek olasılığa sahiptir (Austin, 1990). Bu operatör doğal seçimi yapay olarak gerçekleştirmektedir. Doğal yığınların uygunluğu, bireyin büyümesi ve çoğalmasında engellere karşı koyma yeteneği ile belirlenir. Amaç fonksiyonunun (burada uygunluk değeri) bir dizinin yaşaması ya da elenmesinde son karar verici olarak kullanımı ile "Doğal Seçim" yapay olarak gerçekleştirilir.

Goldberg and Deb (1991) literatürde mevcut ve çok kullanılan seçim mekanizmalarını; orantılı yeniden üretim, sıralı, turnuva, denge durumu, olmak üzere 4 ana sınıfa ayırmaktadır (Dengiz ve Altıparmak, 1997). Ayrıca bunların ilk üçü ile birlikte elitist stratejisi olarak kullanılan en iyi bir veya birkaç dizinin bir sonraki yığına taşınması ile en iyi uygunluk değerine sahip dizinin örnekleme hatası ya

da genetik operatörler kullanımı sonucunda kaybolması önlenmektedir.

Çaprazlama : Çaprazlama operatörü genetik algoritmalarındaki en önemli operatördür. İki çözümün yapıları kullanılarak yeni bir çözüm oluşturulması esasına dayanır. Farklı uygulamalarda farklı kodlama yöntemleri kullanıldığı için farklı çaprazlama yöntemleri kullanılır. Bu çalışmada c kromozomu için çaprazlama operatörü şu şekilde ifade edilir. nc rasgele bir kromozom  $c \neq nc$  iken nesil havuzundan seçilir. Sonra  $\langle x,y \rangle$  değerinin bir çifti c ve nc arasında yer değiştirir.

Mutasyon : Mutasyon genetik algoritmalarındaki operasyonda karar verici olarak ikinci derecede rol oynar. Mutasyon bir çözümün çoğunlukla rasgele olarak değiştirilmesidir. Bu işlem çok değişik şekillerde kullanılır. Problemin yapısı bu aşamada çok önemlidir. Mutasyon GA'nın özünü oluşturur. Starsky (2001), 3 mutasyon operatörü tanımlamaktadır. Mutasyon 1- Bu mutasyonda eğer kromozom uzunluğu n-2'ye (maksimum Steiner sayısı) eşit değilse bir nokta bu kromozoma eklenecektir. Mutasyon 2- Eğer kromozom uzunluğu sifira (minimum Steiner sayısı) eşit değilse, bir nokta bu kromozomdan silinecektir. Silinen nokta bu kromozomdan rasgele seçilir. Mutasyon 3- Bu kromozomdaki tüm noktalar  $r_x$  ve  $r_y$  [-0.5,0.5] arasında yer alan rasgele sayı iken x ekseninde  $r_x$  ve y ekseninde  $r_y$  değişir.

### 3. 1. 4. Genetik Algoritma Parametreleri

Parametreler, genetik algoritma performansı üzerinde önemli etkiye sahiptir. Optimal kontrol parametreleri bulmak için birçok çalışma yapılmıştır. Fakat tüm problemler için genel olarak kullanılacak parametreler bulunamamıştır (Altıparmak et al., 2000).

Mutasyon Oranı (mutation rate): Her bir nesildeki GA mutasyon oranını tanımlar. Bu program çalışırken değişebilir.

Çaprazlama Oranı (crossover rate): Her bir nesildeki GA çaprazlama oranını tanımlar. Program çalışırken değişebilir.

Yığın (population): Her bir nesildeki yığın büyüklüğünü tanımlar. Bu da program çalışırken değişebilir.

Noktaların Sayıları (point number): Grafikteki noktaların (terminals) sayısını belirler.

Yeni set (New Set): Terminallerin yeni bir seti oluşturulur.

MST uzunluğu (MST length): Minimum yayılan ağaç probleminin uzunluğu.

SMT uzunluğu (SMT length): Potansiyel Steiner minimum yayılan ağaç uzunluğu.

Nesil (generation): Mevcut ağaç için nesil sayısı.

Kazanç (Gain): Yüzde olarak SMT probleminin MST'den ne kadar kısa olduğunu hesaplar.  $Kazanç = (MST - SMT)/MST * 100$  ile hesaplanmaktadır.

Start/End Button: Start programı başlatır, stop programı durdurur.

Erteleme (Delay): Programın çok hızlı çalıştığı düşünülüyorsa her bir nesil arası ertelenerek programın daha yavaş çalışması sağlanarak MST hesaplanır.

Square: Appletta kare oluşturacak şekilde seçilen koordinatlar.

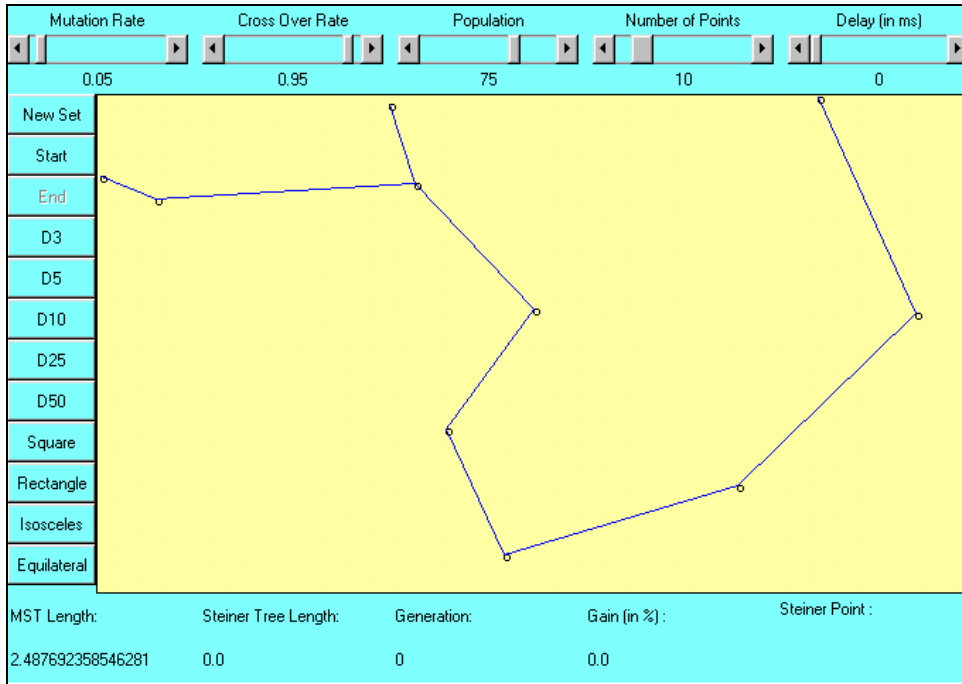
Rectangle: Appletta dikdörtgen oluşturacak şekilde seçilen koordinatlar.

Isosceles: Appletta ikizkenar üçgen oluşturacak şekilde seçilen koordinatlar.

Equilateral: Appletta eşkenar üçgen oluşturacak şekilde seçilen koordinatlar.

Steiner Point: x ve y koordinatlarında Steiner noktasının yeri.

Şekil 4'de minimum yayılan ağaç probleminin genetik algoritma ile çözümü için geliştirilen JAVA programı görülmektedir.



Şekil 4. Programın görünümü

Çalışmada 3, 5, 10, 25, 50 nokta için ve özel şekiller olarak; kare, dikdörtgen, ikizkenar üçgen, eşkenar üçgen oluşturacak şekilde koordinatlar belirlenmiştir. Örneğin D3 butonu ile belirlediğimiz rasgele 3 nokta için değişik mutasyon ve çaprazlama oranları için 10-50-75-100 yığındaki 5000 nesile kadar kazanç değerleri hesaplanabilmektedir. Diğerleri de benzer şekilde çalışmaktadır. Bu çalışmada Steiner noktası sadece 3 nokta için Applet üzerinde yazdırılmaktadır. Herhangi bir nokta seçip start tuşuna bastığımızda program çalışmakta ve 5000 nesile gelince durmaktadır. Program çalıştırıldığında ekranda iki pencere açılmaktadır. İlk

pencerede nokta sayısı, yığın sayısı, nesil, mutasyon ve çaprazlama için bulunan MST ve SMT uzunluğu ve kazanç(gain) değerleri görülmektedir. İkinci pencere de ise yine seçilen noktalar için koordinat değerleri yazdırılmaktadır. Steiner noktasının belirlenmesi için mutasyon oranı=0.05, çaprazlama oranı=0.95 ve yığın=5 alınarak aşağıdaki 3 noktadaki koordinat değerleri için Tablo 1'de bir örnek verilmektedir (İşçi, 2002).

x[0] 0.222810314683889	y[0] 0.8797481800735926
x[1] 0.0626845237969796	y[1] 0.11186210270219199
x[2] 0.759689660986021	y[2] 0.42459092145346666

Tablo 1. Başlangıç Koordinat Değerleri

Kromozomlar	Smt-Uzunluğu	Steiner Koordinatları	
0. kromozom	1.3379863162468921	x[3]= 0.343046195256698	y[3]= 0.342956613484493
1. kromozom	1.3404199378314465	x[3]=0.3388358668642255	y[3]=0.33838795800320065
2. kromozom	1.3396025410615255	x[3]= 0.3407170840805299	y[3]= 0.3399108481306852
3. kromozom	1.3416912524487001	x[3]= 0.33650675568805744	y[3]= 0.3360588468270326
4. kromozom	1.3408570752092468	x[3]= 0.33767131872672207	y[3]= 0.3375817369545171

Tablo 2: Çaprazlama Yapıldıktan Sonra Bulunan SMT Uzunlukları ve Steiner Koordinat Değerleri

Kromozomlar	Smt-Uzunluğu	Steiner Koordinatları	
0. kromozom	1.3379863162468921	x[3]=0.343046195256698	y[3]= 0.342956613484493
1. kromozom	1.3404199378314465	x[3]=0.3388358668642255	y[3]=0.33838795800320065
2. kromozom	1.3396025410615255	x[3]= 0.3407170840805299	y[3]= 0.3399108481306852
3. kromozom	1.3416912524487001	x[3]=0.33650675568805744	y[3]= 0.3360588468270326
4. kromozom	1.3408570752092468	x[3]=0.33767131872672207	y[3]= 0.3375817369545171
5. kromozom	1.3379863162468921	x[3]= 0.343046195256698	y[3]= 0.342956613484493
6. kromozom	1.3416912524487001	x[3]=0.33650675568805744	y[3]= 0.3360588468270326
7. kromozom	1.3408570752092468	x[3]=0.33767131872672207	y[3]= 0.3375817369545171
8. kromozom	1.3396025410615255	x[3]= 0.3407170840805299	y[3]= 0.3399108481306852
9. kromozom	1.3408570752092468	x[3]=0.33767131872672207	y[3]= 0.3375817369545171

Tablo 2'de çaprazlama yapıldıktan sonra aynı kopyalanıyormuş görünmesinin nedeni Steiner nokta sayısının 3 nokta için bir tane olmasından dolayıdır.

Steiner nokta sayısı birden fazla olsaydı o zaman farklı kromozomlar elde edilirdi.

Tablo 3. Mutasyon Yapıldıktan Sonra Bulunan SMT Uzunlukları ve Steiner Koordinat Değerleri

Kromozomlar	Smt-Uzunluğu	Steiner Koordinatları	
0. kromozom	1.3379863162468921	x[3]=0.343046195256698	y[3]= 0.342956613484493
1. kromozom	1.3404199378314465	x[3]= 0.3388358668642255	y[3]=0.33838795800320065
2. kromozom	1.3396025410615255	x[3]= 0.3407170840805299	y[3]= 0.3399108481306852
3. kromozom	1.3416912524487001	x[3]=0.33650675568805744	y[3]= 0.3360588468270326
4. kromozom	1.3408570752092468	x[3]=0.33767131872672207	y[3]= 0.3375817369545171
5. kromozom	1.3379863162468921	x[3]= 0.343046195256698	y[3]= 0.342956613484493
6. kromozom	1.3416912524487001	x[3]=0.33650675568805744	y[3]= 0.3360588468270326
7. kromozom	1.3408570752092468	x[3]=0.33767131872672207	y[3]= 0.3375817369545171
8. kromozom	1.3396025410615255	x[3]= 0.3407170840805299	y[3]= 0.3399108481306852
9. kromozom	1.3408570752092468	x[3]=0.33767131872672207	y[3]= 0.3375817369545171

Mutasyon yapıldıktan sonra elde edilen değerler Tablo 3'de verilmektedir. Kromozom çeşitliliğini

arttırmak için 5 tane rasgele kromozom ekledikten sonra elde edilen değerler Tablo 4'de verilmektedir.

Tablo 4. 5 Adet Rasgele Kromozom Ekledikten Sonra Elde Edilen Değerler

Kromozomlar	Smt-Uzunluğu	Steiner Koordinatları	
0. kromozom	1.3379863162468921	x[3]=0.343046195256698	y[3]=0.342956613484493
1. kromozom	1.3404199378314465	x[3]= 0.3388358668642255	y[3]=0.33838795800320065
2. kromozom	1.3396025410615255	x[3]= 0.3407170840805299	y[3]= 0.3399108481306852
3. kromozom	1.3416912524487001	x[3]=0.33650675568805744	y[3]= 0.3360588468270326
4. kromozom	1.3408570752092468	x[3]=0.33767131872672207	y[3]= 0.3375817369545171
5. kromozom	1.3379863162468921	x[3]= 0.343046195256698	y[3]= 0.342956613484493
6. kromozom	1.3416912524487001	x[3]=0.33650675568805744	y[3]= 0.3360588468270326
7. kromozom	1.3408570752092468	x[3]=0.33767131872672207	y[3]= 0.3375817369545171
8. kromozom	1.3396025410615255	x[3]= 0.3407170840805299	y[3]= 0.3399108481306852
9. kromozom	1.3408570752092468	x[3]=0.33767131872672207	y[3]= 0.3375817369545171
10. kromozom	1.3362801847634442	x[3]=0.34671903301594065	y[3]= 0.3462711241549158
11. kromozom	1.3277041212415468	x[3]=0.36499362513878775	y[3]= 0.3645457162777629
12. kromozom	1.3272463169727269	x[3]= 0.3660685915040862	y[3]=0.36562069754422255
13. kromozom	1.3278994163352205	x[3]=0.36141036915175007	y[3]= 0.3638290621001231
14. kromozom	1.3300727697987798	x[3]= 0.3596187486088118	y[3]=0.35917082484662577

Son olarak elde edilen değerler küçükten büyüğe doğru sıralanır. En kısa SMT uzunluğu en tepede yer almaktadır. Bu durumda kromozom havuzu

şağıdaki gibi olur. Sıralama yaptıktan sonra elde edilen değerler aşağıda Tablo 5’de görüldüğü gibidir;

Tablo 5. Sıralama Yaptıktan Sonra Elde Edilen Değerler

Kromozomlar	Smt-Uzunluğu	Steiner Koordinatları	
0. kromozom	1.3272463169727269	x[3]= 0.3660685915040862	y[3]=0.3656206975442255
1. kromozom	1.3277041212415468	x[3]= 0.36499362513878775	y[3]= 0.3645457162777629
2. kromozom	1.3278994163352205	x[3]= 0.36141036915175007	y[3]= 0.3638290621001231
3. kromozom	1.3300727697987798	x[3]= 0.3596187486088118	y[3]= 0.35917082484662577
4. kromozom	1.3362801847634442	x[3]= 0.34671903301594065	y[3]= 0.3462711241549158
5. kromozom	1.3379863162468921	x[3]= 0.343046195256698	y[3]= 0.342956613484493
6. kromozom	1.3379863162468921	x[3]= 0.343046195256698	y[3]= 0.342956613484493
7. kromozom	1.3396025410615255	x[3]= 0.3407170840805299	y[3]= 0.3399108481306852
8. kromozom	1.3396025410615255	x[3]= 0.3407170840805299	y[3]= 0.3399108481306852
9. kromozom	1.3404199378314465	x[3]= 0.3388358668642255	y[3]= 0.33838795800320065
10. kromozom	1.3408570752092468	x[3]= 0.33767131872672207	y[3]= 0.3375817369545171
11. kromozom	1.3408570752092468	x[3]= 0.33767131872672207	y[3]= 0.3375817369545171
12. kromozom	1.3408570752092468	x[3]= 0.33767131872672207	y[3]= 0.3375817369545171
13. kromozom	1.3416912524487001	x[3]= 0.33650675568805744	y[3]= 0.3360588468270326
14. kromozom	1.3416912524487001	x[3]= 0.33650675568805744	y[3]= 0.3360588468270326

Nesil bitimi Tablo 6’da verilmektedir. Burada tamsayı(0,7\*5)=3 tane dizin alınır ve daha sonra 2

tane kromozom rasgele üretilerek vektörün sonuna eklenir.

Tablo 6. Nesil Sonunda Elde Edilen Değerler

Kromozomlar	Smt-Uzunluğu	Steiner Koordinatları	
0. kromozom	1.3272463169727269	x[3]= 0.3660685915040862	y[3]= 0.3656206975442255
1. kromozom	1.3277041212415468	x[3]= 0.36499362513878775	y[3]= 0.3645457162777629
2. kromozom	1.3278994163352205	x[3]= 0.36141036915175007	y[3]= 0.3638290621001231
3. kromozom	1.3295885179182638	x[3]= 0.3606937149741103	y[3]= 0.3602458210142466
4. kromozom	1.3317242450268547	x[3]= 0.35603549262177414	y[3]= 0.3555875837607493

Minimum yayılan ağaç probleminin genetik algoritma çözümü iki durumda ele alınmaktadır: I. durumda Applet’te de D3, D5, D10, D25, D50 olarak görülen 3, 5, 10, 25 ve 50 noktaları için hesaplamalar yapılmıştır.

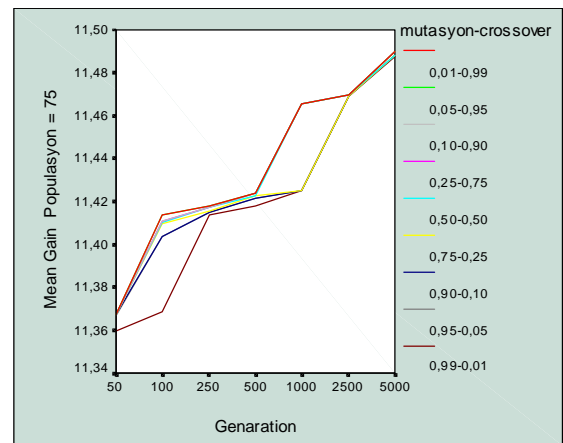
Bunun için seçilen yığın (10-25-50-75-100), nesil (50-100-250-500-1000-2500-5000), mutasyon ve çaprazlama oranları(0.01-0.05-0.10-0.25-0.50-0.75-0.90-0.95-0.99) olarak alınmıştır.

Böylece 9 farklı mutasyon ve çaprazlama oranı için MST uzunluğu, SMT uzunluğu ve kazanç değerleri hesaplanmıştır.

Şekil 5’de I. durum için nokta sayısı 3 alındığında ve yığın 50 olduğunda kazanç değerleri mutasyon ve çaprazlama oranlarına göre gösterilmektedir. Şekilde görüldüğü gibi yığın ve nesil arttıkça kazanç değerleri de artmaktadır. 5000 nesilde program durdurulduğunda elde edilen kazanç değeri yaklaşık 11.49 olarak bulunmuştur.

Burada seçilen çaprazlama ve mutasyon oranları için elde edilen kazanç değerleri aynı olduğunda çizgiler

üst üste gelmiştir. Yığın ve nesil ne kadar arttırılırsa arttırılsın mutasyon ve çaprazlama oranları için kazanç değerlerinin değişmediği görülmüştür.



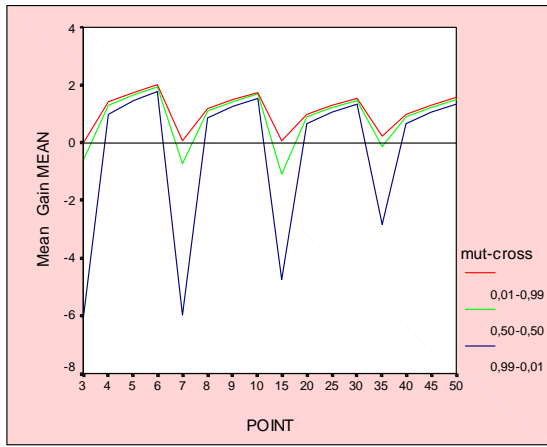
Şekil 5. Nokta Sayısı = 3, Yığın = 50

Örneğin yığın 10 olarak alındığında negatif kazanç değerleri elde edilirken yığın 100 olduğunda pozitif kazanç değerleri elde edilebilmektedir. II. durumda 3, 4, 5, 6, 7, 8, 9, 10, 15, 20, 25, 30, 35, 40, 45, 50

nokta sayısı için 1000 nesil, yığın (10-25-50-75-100), mutasyon ve çaprazlama oranları(0.01-0.05-0.10-0.25-0.50-0.75-0.90-0.95-0.99) olarak alınarak 9 farklı mutasyon ve çaprazlama oranı için MST uzunluğu, SMT uzunluğu ve kazanç değerleri hesaplanmıştır.

Ayrıca II. durumda belirlenen bu 16 nokta için 100 tane farklı yeni set üreterek çaprazlama ve mutasyon oranlarının yığın sayısına etkisi de araştırılmıştır.

Şekil 6'da II. durum için yığın büyüklüğü 75 olarak sabit tutulduğunda 0.01-0.99; 0.50-0.50; 0.99-0.01 mutasyon ve çaprazlama oranları için elde edilen kazanç değerleri grafiksel olarak gösterilmiştir.



Şekil 6. Yığın = 75 olduğunda ortalama kazanç

#### 4. SONUÇ VE TARTIŞMA

Bu çalışmada elde edilen değerler istatistiksel olarak değerlendirildiğinde Steiner minimum yayılan ağaç probleminin genetik algoritma ile çözülmesi için seçilecek mutasyon oranı küçük, çaprazlama oranı ise yüksek olmalıdır. Özellikle nesil sayısı arttıkça elde edilen kazanç değerleri daha yüksek çıkmakta fakat belirli bir süre sonra kazanç değeri değişmemektedir. Program başlatıldığında başlangıçta negatif kazanç değeri de bulabilmektedir. Fakat birkaç saniye içinde oldukça küçük negatif değeri bulmakta daha sonra ise pozitif değerler bulabilmektedir. Bunun nedeni ise başlangıçta rasgele belirlemesidir. Nokta sayısı az olduğunda program çok hızlı çalıştığı için Applet üzerinden de Steiner noktaları, MST ve SMT uzunluklarının değişimini görebilmek için erteleme (delay) tuşu ile geciktirmeler yapılabilmektedir.

Çalışma kapsamında aldığımız noktalar açısından değerlendirilirse özellikle az sayıda nokta için çok kısa sürede iyi sonuçlar bulabilmektedir. Nokta

sayısı arttıkça işlem yükü arttığı için sonuçların elde edilmesi daha uzun zaman almaktadır. Örneğin, Steiner minimum yayılan ağaç problemi için yaptığımız çalışma 3 ile 10 arasında seçilen noktalar için 1 dk.'dan daha kısa bir zamanda sonuç bulurken nokta sayısı 50 ile 75 için 30 dk.'daha fazla zaman almıştır.

Genetik algoritma ile problem çözümü için çaprazlama ve mutasyon oranlarının etkili olduğu ve yığın sayısının pozitif kazanç değerlerinin elde edilmesinde rol oynadığı görülmüştür. Koordinatlarının yerinin kazanç değerlerini etkilediği Applet üzerinde de açıkça görülmektedir. Bu koordinat değerleri özel durumlar dışında rasgele belirlenmiştir. Fakat kendimizde bu noktaların koordinatlarını vererek MST uzunluğu, SMT uzunluğu ve kazanç değerlerini hesaplayabiliriz. Bu araştırma kapsamında elde edilen sonuçlar göstermiştir ki; bundan sonraki yapılacak çalışmalarda özellikle seçilecek çaprazlama ve mutasyon oranı elde edilecek kazanç değerlerini etkilemektedir ve yığının çok olması daha iyi değerleri elde edilmesini sağlamaktadır. Yine, nesil sayısı optimal sonuca yaklaşmak için önemlidir. Nesil sayısı arttıkça daha iyi kazanç değerleri elde edilmektedir.

#### 5. KAYNAKLAR

Altıparmak, F. ve Deniz, B. 1998. Genetik Algoritmalar, Gazi Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 11 (3) : 523-542.

Altıparmak, F., Dengiz B. ve Smith, A. E. 2000. An Evolutionary Approach For Reliability Optimization in Fixed Topology Computer Networks, Transactions On Operational Research, 2 (1-2) : 57-75.

Austin, S. 1990. "An Introduction to Genetic Algorithms," AI Expert, 5 (3) : 48-53.

Boruvka, O. 1926. O Jistem Problemu Minimaln İm, Prace Mor. Prrodove Spol. v Brne (Acta Societ. Scient. Natur. Moravicae, 3 : 37-58.

Clarke, L. 2002. The Steiner Problem <http://www.css.tayloru.edu/~lclarke/srproj/Documentation2.html> (12.08.2002)

Dengiz, B., Altıparmak, F. 1997. Kombinatorial Optimizasyon ve Genetik Algoritmalar, Makina Mühendisleri Odası Endüstri Mühendisliği Dergisi, 9 (3) : 3-14.



Dong, S. Z. 2002. Genetic Algorithm Applications, Mech 580, Quantitative Analysis, Reasoning and Optimization Methods in CAD/CAM and Concurrent Engineering, [http://www.me.uvic.ca/~zdong/courses/mech620/GA\\_App.PDF](http://www.me.uvic.ca/~zdong/courses/mech620/GA_App.PDF) (27.09.2002).

Goldberg, D. E. 1989. Genetic Algorithms In Search, Optimization And Machine Learning”, Addison-Wesley, New York, 1-7.

Goldberg, D. E. and Deb, K. 1991. A Comparison Of Selection Schemes Used In Genetic Algorithms”, Foundations Of Genetic Algorithms, 69-93.

Holland, J. R. 1975. Adaptation in Natural and Artificial Systems, University of Michigan Press, 2-3.

Harris, C. F. 1995. An Introduction To Steiner Minimal Trees On Grids, 111 : 3-17. <http://www.cs.unr.edu/~fredh/papers/journal/Journal.html>

İşçi, Ö. 2002. Genetik Algoritmaların Yöneylem Araştırmasındaki Uygulamaları, Doktora Tezi, Ege Üniversitesi, Fen Bilimleri Enstitüsü, İzmir.

Jones, J., Harris, C. F.A. 1996. Genetic Algorithm for the Steiner Minimal Tree Problem, Proc. ISCA's

Int. Conf. on Intelligent Systems (IS '96), Reno, NV, 19-21.

Kruskal, J. 1956. On The Shortest Spanning Subtree Of A Graph And The Traveling Salesman Problem, Proceedings Of The American Mathematical Society, 7:48-50.

Oğuz, M., Akbaş S. 1999. Genetik Algoritmalar, Bitirme Tezi, YTÜ Endüstri Mühendisliği, <http://artemis.efes.net/moguz/>.

Prim, R. C. 1957. Shortest Connection Networks And Some Generalizations, Bell System Technology Journal, 36:1389-1401.

Starsky, W. 2001. Using Algorithm for Solving the Steiner Minimal Tree Problem <http://appsrv.cse.cuhk.edu.hk/~csc6200/y99/applet/Steiner/>.

Senior, C. 2002. Steiner Trees by: Jennifer Dorenbos, [http://www.css.tayloru.edu/~jdorenbos/senior\\_project/](http://www.css.tayloru.edu/~jdorenbos/senior_project/) (08.02.2002).

Zhou, G., Gen, M. 1999. Genetic Algorithm Approach on Multi- Criteria Minimum Spanning Tree Problem” European Journal of Operational Research, 114:141-151.