*Öztürk ve Çelik, Erciyes Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 28(4): 322-328*

Erciyes University Journal of the Institute of Science and Technology
Erciyes Üniversitesi Fen Bilimleri Enstitüsü Dergisi
ISSN 1012-2354
Cilt (Volume): 28, Sayı (Issue): 4, Temmuz/July-2012
http://fbe.erciyes.edu.tr/

# Polinom olmayan denklemlerin genetik algoritma tabanlı çözümü

**Nihat ÖZTÜRK*, Emre ÇELİK**
*Gazi Üniversitesi Teknoloji Fakültesi Elektrik Elektronik Mühendisliği Bölümü, ANKARA*

**ÖZET**

Bu çalışmada, son yıllarda optimizasyon alanında sıkça kullanılan genetik algoritmalar (GA) polinom olmayan denklem çözümlerine uygulanmıştır. Genetik algoritmalar problemlerin çözümü için evrimsel süreci bilgisayar ortamında taklit ederler ve tepe tırmanma algoritması, tavlama benzetimi, v.s. gibi diğer optimizasyon yöntemlerinde olduğu gibi çözüm için tek bir yapının geliştirilmesi yerine, bir çok potansiyel çözümden meydana gelen bir popülasyon oluştururlar. Karınca kolonisi algoritmasından farklı olarak optimizasyon işlemi bittiğinde popülasyonda bulunan en iyi birey probleme çözüm oluşturmaktadır. Önerilen yöntemin uygunluğunu test etmek için dört farklı doğrusal olmayan fonksiyondan yararlanılmış ve test sonuçları her bir fonksiyonun gerçek çözümleri ile karşılaştırılmıştır. MATLAB® 6.5 ortamında gerçekleştirilen bu çalışmadan elde edilen sonuçlar ortaya konulan yöntemin test fonksiyonlarının yakınsanan kök değerleri ve iterasyon sayısı bakımından iyi bir performansa sahip olduğunu göstermiştir.

**Anahtar Kelimeler:** Genetik algoritma, denklem çözümü, optimizasyon

# Solution of non-polynomial equations based on genetic algorithm

**ABSTRACT**

In this study, genetic algorithms (GAs), which have been used heavily in the field of optimization recently, have been applied to non-polynomial equation solutions. GAs simulate the evolutionary process in computer environment for the solutions of problems and instead of improving a single solution as in other optimization methods, such as hill-climbing algorithm, simulated annealing, etc., a GA forms a population composed of many potential solutions. Unlike ant colony algorithm, when the optimization process ends up, the best individual in the population forms the solution to the problem. To test the suitability of the proposed method, four different non-linear functions are utilized and the test results are compared to the real solutions of each functions. The obtained results from this study, realized in MATLAB® 6.5 environment, reveal that the suggested method has a good performance in terms of the converged root values of test functions and number of iterations.

**Key Words:** Genetic algorithm, equation solution, optimization

*Sorumlu Yazar (Corresponding author) e-posta: ozturk@gazi.edu.tr

## 1. Giriş

Sayısal analiz (nümerik analiz veya sayısal çözümleme) matematik problemlerinin bilgisayar destekli çözümleme tekniğidir. Sayısal analizde temel amaç çözümün elle yapılmasının pratik olmadığı karmaşık problemlerin yaklaşık sayısal çözümlerinin elde edilmesidir ve bu anlamıyla da özellikle mühendislik ve uygulamalı matematikte önemlidir. Sayısal çözümün vazgeçilmez parçalarından biri de elektronik araçlardır. Bilgisayar teknolojisi ile sayısal analiz metotları birbirine paralel olarak gelişmiştir. Bunun en güzel örneği günümüzün en popüler nümerik analiz metotlarından biri olan sonlu elemanlar metodunun teorisi 1930'larda olmasına rağmen; yöntem el ile işlem yapmaya uygun olmadığından dolayı gerekli ilgiyi o yıllarda görmemiş ve gelişen bilgisayar teknolojisiyle birlikte kullanım alanı bulmuştur. Bunun yanında analitik işlemlerin bilgisayar ortamında yapılabilmesi için yine sayısal analiz metotlarının kullanılma zorunluluğu vardır. Tüm bu ihtiyaçlar sayısal analiz metotlarının gelişmesine neden olmuştur [1].

Denklem çözümüne yönelik olarak birçok sayısal yöntem ve teknik geliştirilmiştir. Bunların başında Yarılama yöntemi, Kiriş yöntemi, Sabit nokta yinelemesi, Newton Raphson ve Regula-False yöntemleri gelmektedir [1]. Bütün bu sayısal analiz yöntemlerinde az sayıda iterasyon ve en az hatayla doğru sonuca ulaşmak temel hedeftir.

Evrim algoritmaları (EA), doğal evrim mantık ve prensiplerine göre geliştirilen popülasyon tabanlı stokastik araştırma algoritmalarıdır. EA'lar özellikle birçok yerel optimumun üretildiği büyük karmaşık problemlerin çözümünde faydalıdır ve klasik gradient tabanlı araştırma algoritmalarına göre yerel minimuma takılma olasılıkları daha azdır. EA'lar gradient bilgisine bağlı değildirler ve bu nedenle bu tip bilginin elde edilmesinin çok zor olduğu problemlerin çözülmesi için oldukça elverişlidirler [2].

Bu çalışmada, son yıllarda optimizasyon alanında sıkça kullanılan genetik algoritmalar, polinom olmayan denklem çözümlerine uyarlanmıştır. Belirlenen kriterler doğrultusunda genetik algoritmalar en iyi çözüme ulaşmaktadırlar.

## 2. Sayısal Analiz Teknikleri

Denklem köklerinin hesap edilmesi için kullanılan birçok yöntem vardır. Bunlardan bazıları Yarılama, Kiriş, Sabit nokta yinelemesi, Newton Raphson ve Regula-False yöntemleridir.

a) *Yarılama yöntemi*: Yarılama yöntemi *f(x)=0* fonksiyonunun kökün bulunacağı tahmin edilen veya verilen aralığı ikiye bölerek kök arama işlemidir [3]. Fonksiyonun kökü bir [a,b] aralığında aranıyorsa ve *f(a).f(b)*<0 ise bu aralıkta en az bir kök vardır.

b) *Kiriş (Secand) yöntemi*: Bu metot yarılama metoduna benzemekle birlikte ondan daha fazla sayıda yineleme yapılarak sonuca varması bir dezavantaj olarak kabul edilse de, diğer taraftan bu metodun yarılama metodunda olduğu gibi aranan kökün verilen [a,b] arasında gerekli olmaması bir avantajdır.

c) *Sabit nokta yinelemesi*: Bu yöntemde verilen *f(x)* fonksiyonunun içinde bulunan bilinmeyen *x* yalnız bırakılarak *x=g(x)* şekline getirilir. Elbette lineer olmayan

bir fonksiyonu birçok şekilde x=g(x) haline dönüştürülebilir. Daha sonra x'e değerler verilerek buna karşılık gelen g(x) değerleri bulunur. Verilen değer ile bulunan değerin aynı olması istendiğinden dolayı bu sonuç bulunana kadar yinelemeye devam edilir. Birinci adımda değerine karşı bulunan değeri ikinci adımda değeri olarak adlandırılır. Bu oluncaya kadar devam eder.

d) *Newton-Raphson yöntemi:* Bu yöntem kök bulmak için kullanılan en popüler yöntemlerden biridir [1]. Daha önceden gösterilen yöntemlerden daha hızlı yakınsadığı için diğerlerine göre daha avantajlıdır. Kiriş metodunda verilen fonksiyonun lineer olarak modellemesi yapılırken Newton metodunda doğrusal modelleme yerine fonksiyonun türevi alınarak işlem yapılır. Verilen fonksiyonun herhangi bir noktaya yakın olan kökünü bulmak için; bu noktada fonksiyonun türevi alınarak, o noktadaki fonksiyonun teğeti bulunur. Bu teğetin x eksenini kestiği yer Denklem (1) yardımı ile bulunabilir. Burada fonksiyonun birinci türevini gösterir. Bu işlemler belli sayıda veya belli yakınsaklık değerine ulaşıncaya kadar devam eder.

$$x_{i+1} = x_i - \frac{f(x)}{f'(x)} \tag{1}$$

e) *Regula-False yöntemi*: [a,b] aralığında $y = f(x)$ fonksiyonunun kökünü hesaplamak için her iterasyonda,

$$x_0 = \frac{a.f(b) - b.f(a)}{f(b) - f(a)} \tag{2}$$

eşitliğinden kesim noktası bulunarak fonksiyonda yerine yazılır. Eğer belirlenen hassasiyet/hata değerine ulaşılamadıysa bir sonraki iterasyonda; kökün bulunduğu yeni aralık,

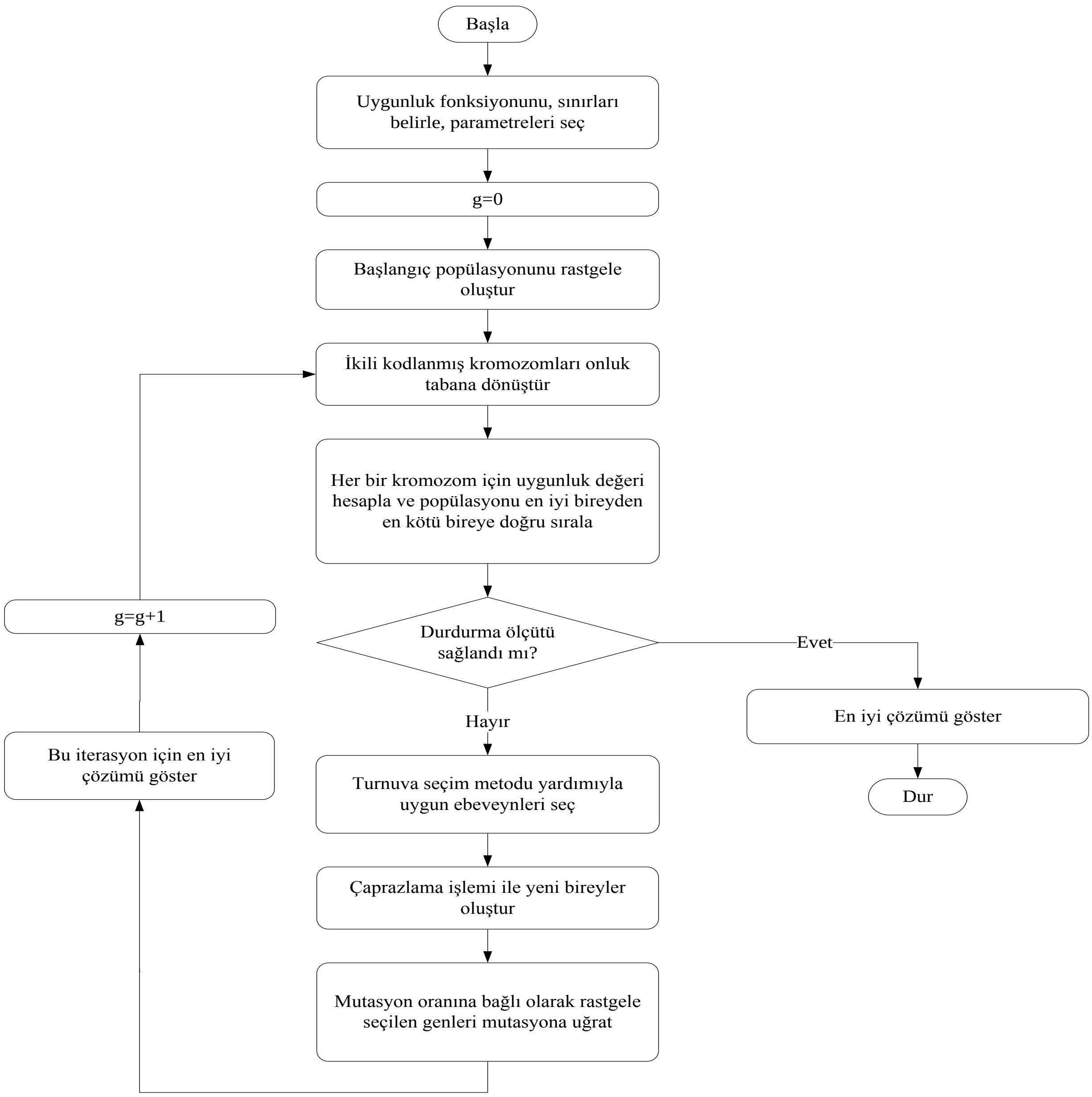
$$f(a).f(x_0)<0 \qquad \Rightarrow b = x_0 \tag{3}$$

$$f(a).f(x_0)>0 \qquad \Rightarrow a = x_0 \tag{4}$$

koşuluna göre belirlenerek devam edilir [4].
Özet olarak, klasik birçok yöntem en uygun çözüme ulaşmak için uygun başlangıç şartlarının belirlenmesi, çok sayıda iterasyon ve her iterasyonda türev hesaplamaları gibi karmaşık matematiksel işlemler gerektirmektedir. Çözülmesi istenen fonksiyonun analitik türevinin alınamadığı ve sayısal türevinin de ağır hesaplama gerektirdiği durumlarda ise işlemler daha da zorlaşmaktadır.

## 3. Genetik Algoritmalar

Genetik algoritmalar (GA), biyolojik süreci modelleyerek fonksiyonları optimize eden evrim algoritmalarıdır [5]. GA parametreleri, biyolojideki genleri temsil ederken, parametrelerin toplu kümesi de kromozomu oluşturmaktadır. GA'lar, her bir ferdi kromozomlar (bireyler) şeklinde temsil edilen popülasyonlardan oluşur. Her bir kromozomun çözüm uzayında bir uygunluğu olup, ikili veya onlu sayı dizileri ile kodlanır. Popülasyonun uygunluğu, belirli kurallar dahilinde maksimize veya minimize edilir. Her yeni nesil, rastgele bilgi değişimi ile oluşturulan diziler içinde hayatta kalanların birleştirilmesi ile elde edilmektedir [6]. GA'nın yürütülmesinde, ilk olarak popülasyonda bulunan kromozomların gözlenen performansına göre kopyalama işlemi yapılır. Böylece daha iyi kromozomların kopya üretme şansı daha fazla olup, bu kromozomların yeni popülasyona katkı sağlama ihtimalleri daha yüksektir [7]. Daha sonra her kuşakta GA, çaprazlama ve mutasyon gibi genetik operatörleri kullanarak yeni bir popülasyon oluşturur. Birkaç kuşak sonunda, popülasyon daha iyi uygunluk değerine sahip üyeleri içerir. Bu, Darwin'in rastsal mutasyona ve doğal seçime dayanan evrim modellerine benzemektedir. GA'lar, çözümlerin kodlanmasını, uygunlukların hesaplanmasını, çoğalma, çaprazlama ve mutasyon operatörlerinin uygulanmasını içerir [8].GA'lar optimizasyon işlemine, probleme muhtemel çözüm olabilecek kromozomları içeren rastgele oluşturulan bir popülasyon ile başlar [9]. Diğer optimizasyon metotlarında olduğu gibi GA'da da amaç fonksiyonu, parametreler ve sınırlar tanımlanır. Aynı şekilde yakınsama kontrol edilerek algoritma son bulur [6]. Bu çalışmada durdurma ölçütü olarak maksimum iterasyon sayısının aşılması ya da fonksiyon değerinin mutlak değerinin $\varepsilon = 10^{-5}$, den küçük olduğu durumlar olarak belirlenmiştir. Geliştirilen ikili kodlu GA'nın arama prosedürü Şekil 1'de verilmektedir.




Şekil 1. İkili genetik algoritmaların arama prosedürü

Bu çalışmada kullanılan GA aşağıdaki program kod ile Popülasyondan atılmış olan kromozomlar yerine yeni bireyler özetlenebilir.

**1.** $g=0$ ata.

**2.** $C_g$ ilk neslini rastgele oluştur.

**3.** While (yakınsama olmazken ya da maximum iterasyon sayısı aşılmamışsa)

- **a.** Her bireyin uygunluğunu değerlendir.
- **b.** Popülasyonu uygunluğu en yüksek olan bireyden uygunluğu en düşük olan bireye doğru sırala.
- **c.** Popülasyondan atılma oranına göre uygunluğu kötü olan bireyleri popülasyondan at.
- **d.** Üreme için popülasyonda atılmadan kalan iyi bireyler arasında turnuva seçim mekanizmasını kullanarak ailelerin seçimini gerçekleştir.
- **e.** Çaprazlama ve mutasyon işlemleri ile yeni bireyleri üret.
- **f.** Üretilen yeni bireylerle popülasyondan atılan eski jenerasyondaki bireyleri yer değiştir.
- **g.** $g=g+1$

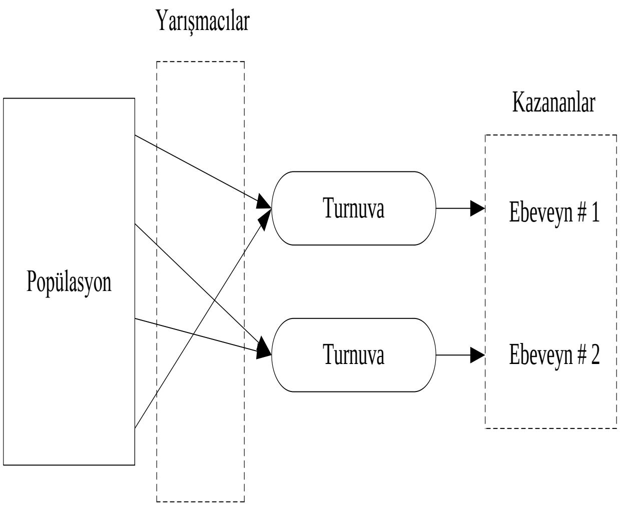
## 4. Simülasyon Sonuçları

Simülasyon çalışmalarında başlangıç popülasyonu şu koşullar altında oluşturulmuştur; optimize edilecek bir tane bağımsız değişken olup bu değişken yirmi beş bitlik ikili sayı sisteminde kodlanmıştır. Başlangıç popülasyonu rastgele üretilen 8 kromozomdan meydana gelmiştir. Her bir kromozomun uygunluğu değerlendirilip popülasyon, uygunluğu en yüksek bireyden uygunluğu en düşük bireye doğru sıralanmıştır. %50 atılma oranını kullanarak uygunluğu en düşük dört kromozom popülasyondan atılmıştır. Tablo 1'de GA parametreleri verilmiştir.

Tablo 1. Genetik algoritma parametreleri

| *GA parametresi* | *Değer / Metod* |
|---|---|
| Popülasyon büyüklügü | 8 |
| Maksimum iterasyon sayısı | 1000 |
| Kromozom uzunluğu | 25 |
| Seçim metodu | Turnuva Seçimi |
| Çaprazlama metodu | Rastgele |
| Popülasyondan dışlanma oranı | % 50 |
| Mutasyon olasılığı | 0.10 |
| Uygunluk fonksiyonu | Fonksiyonun mutlak değeri |

meydana getirmek için turnuva seçim (tournament selection) mekanizması, popülasyonda atılmadan kalan bireyler arasında uygulanmıştır. Bu seçime göre; iki grup oluşturulmuş ve her gruptaki iki kromozom popülasyondan rastgele seçilmiştir. Gruplara rastgele seçilen kromozomlardan en iyisi bu turnuvayı kazanmış ve ebeveyn olarak seçilmiştir. Böylece popülasyon büyüklüğü sabit tutulmuştur. Her bir turnuvaya iki kromozomun seçildiği işlem Şekil 2'de görülmektedir. Çaprazlama ve mutasyon işlemleri her bir seçme işleminin sonunda gerçekleştirilmiştir. Aynı zamanda bu çalışmada, elitizm olarak bilinen popülasyonun en iyi çözümü, bir sonraki nesile hiçbir değişikliğe uğramadan aktarılmıştır [10].




Şekil 2. Turnuva Seçimi

Tasarlanan genetik tabanlı denklem çözüm metodunun test edilmesi için dört ayrı doğrusal olmayan fonksiyondan yararlanılmıştır. Test aşamasında kullanılan doğrusal olmayan test fonksiyonları Şekil 3'de verilmiştir ve aşağıda olduğu gibi tanımlanmıştır.

$$f^{(1)}(x)=e^{x}\cdot\sin(x),\ -2\le x\le 3$$

$$f^{(2)}(x)=150.e^{-x/2}.(\cos x-\sin x),\ 1\le x\le 6$$

Hiperbolik tanjant:

$$f^{(3)}(x)=\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}},\ -10\le x\le 10$$

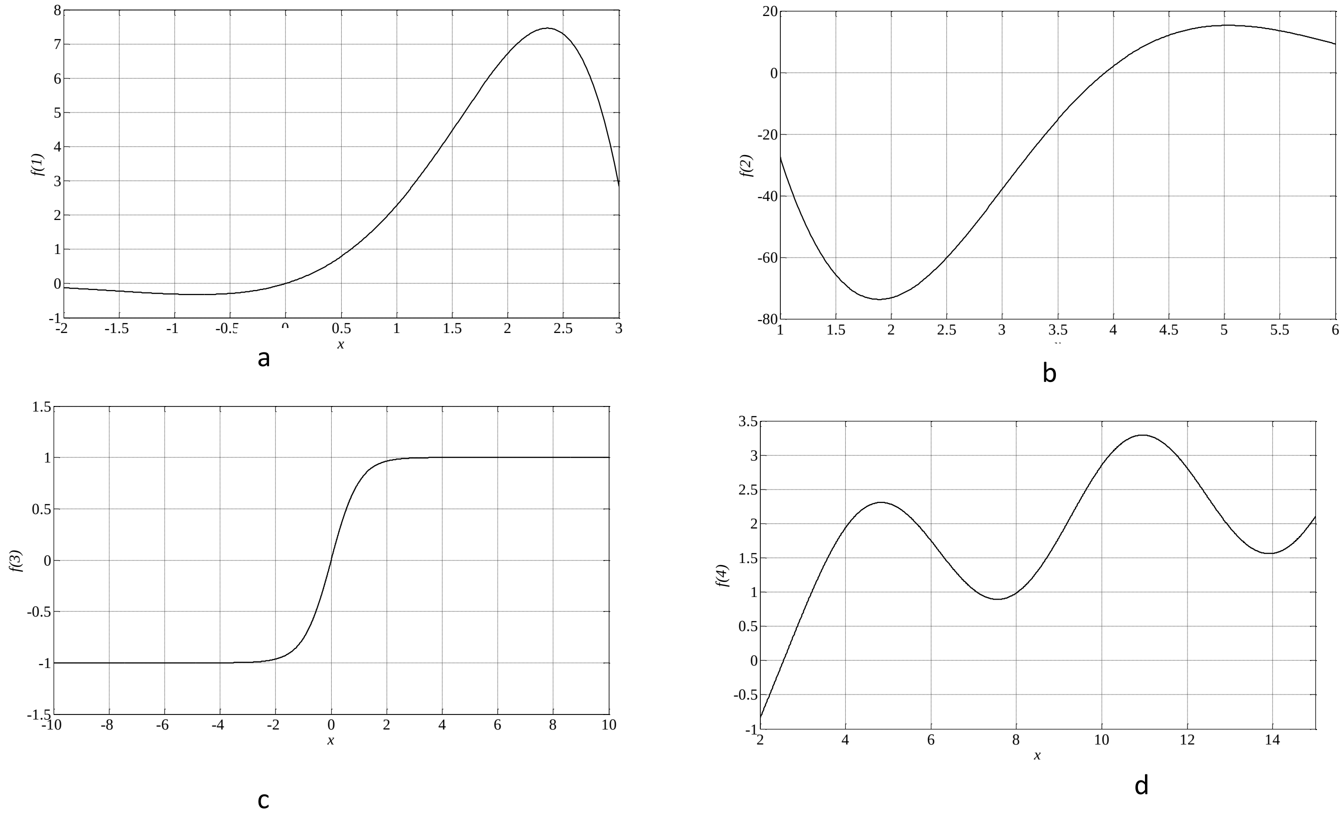
ve

$$f^{(4)}(x)=\ln(x-1)+\sin(x-3),\ 2\le x\le 15$$

Bu tanımlamalardan sonra uygunluk fonksiyonu *F*, Denklem (5)'de olduğu gibi tanımlanabilir.

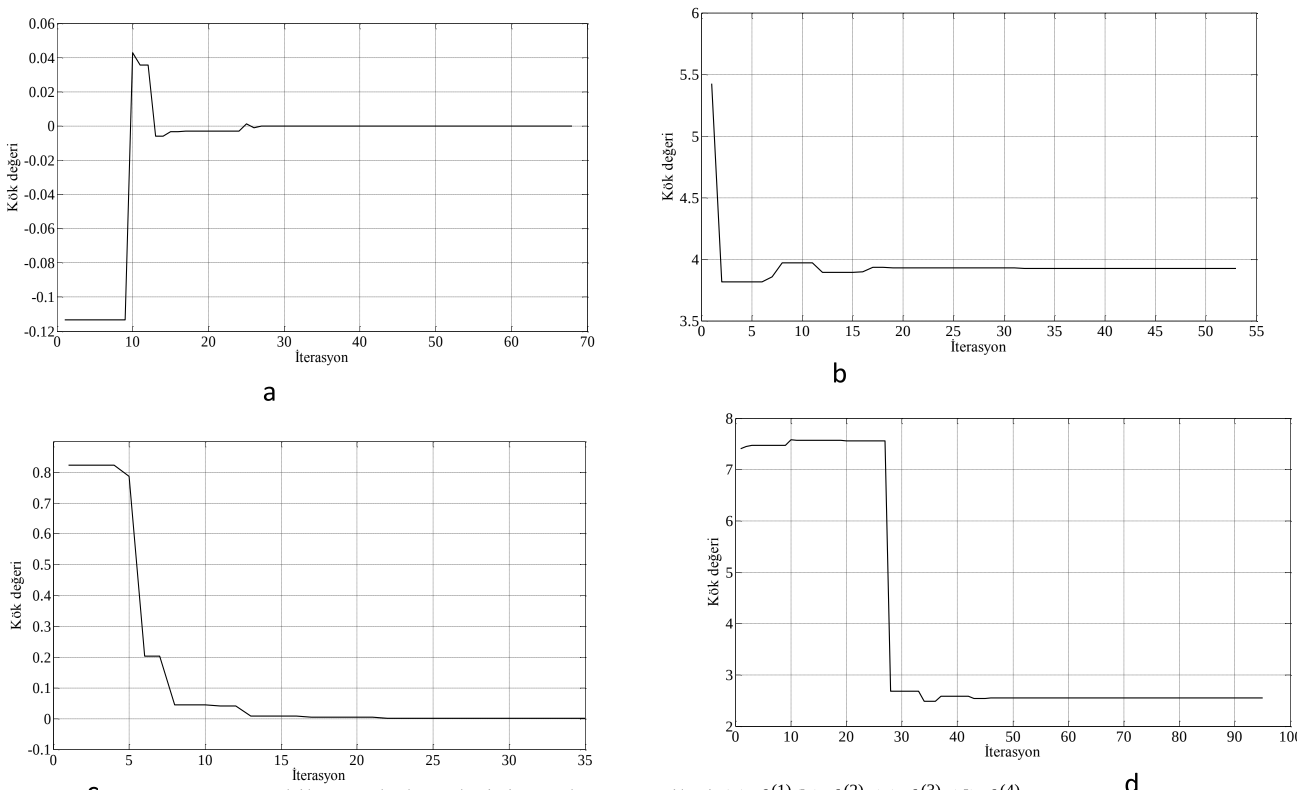
$$F=\left|f^{(i)}(x)\right| \tag{5}$$

Burada, $i=1,2,3,4$ şeklindedir.



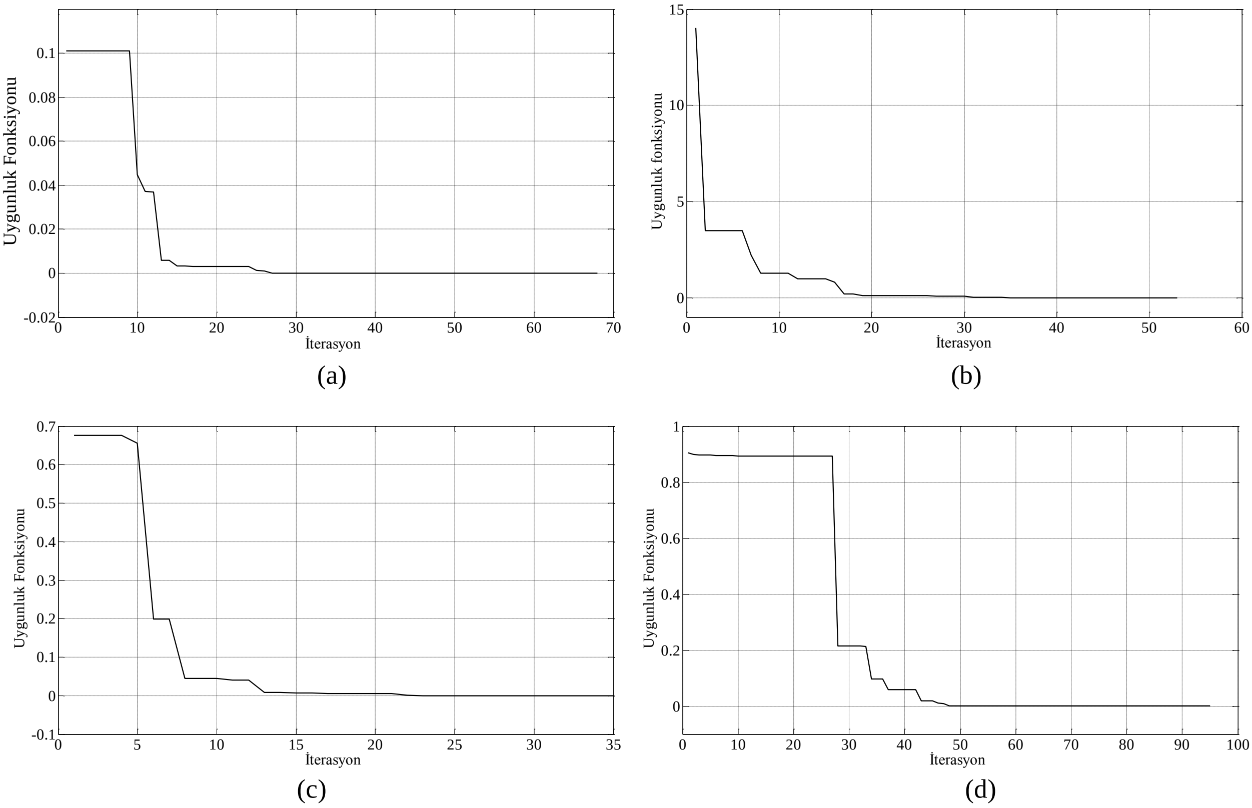

Şekil 3. Test fonksiyonlarının grafikleri (a) $f^{(1)}$(b) $f^{(2)}$ (c) $f^{(3)}$ (d) $f^{(4)}$ .

Tasarlanan genetik algoritmalarla yukarıda verilen *x* aralıkları kullanılarak simülasyon çalışmaları gerçekleştirilmiştir. Elde edilen fonksiyon değeri $\varepsilon = 10^{-5}$ değerinden küçük olduğunda optimizasyon işlemi sona ermektedir. Aksi takdirde, arama işlemi 1000 iterasyon boyunca devam etmektedir. 4 test fonksiyonu için kök çözüm değerlerinin iterasyona bağlı değişim eğrileri Şekil 4'de verilmektedir. Bu eğrilere göre, $f^{(1)}$, $f^{(2)}$, $f^{(3)}$ ve $f^{(4)}$ fonksiyonları için sırasıyla 68, 53, 35 ve 95 nesil sonra önceden belirlenmiş durdurma ölçütü sağlanmış ve kök arama işlemi sonlandırılmıştır.




Şekil 4. Kök değerlerinin yerleşme eğrileri (a) $f^{(1)}$(b) $f^{(2)}$ (c) $f^{(3)}$ (d) $f^{(4)}$ .

Şekil 5 test fonksiyonlarına ait genetik algoritmanın yakınsamasını göstermektedir. Çözülmeye çalışılan problem bir minimizasyon problemi olduğu için popülasyonda bulunan kromozomların, uygunluk fonksiyonunda aldığı değer ne kadar küçük ise değerliliği de o derece artmaktadır. Bu grafikler incelendiğinde tüm fonksiyonlar için en iyi çözümün uygunluğunu gösteren eğri bir önceki iterasyona göre ya düşmekte ya da daha iyi çözümler bulununcaya kadar aynı seviyede kalmaktadır. Bunun nedeni algoritmada kullanılan elitist yaklaşımdır.




Şekil 5. Eğitme aşamasında genetik algoritmanın yakınsaması (a) $f^{(1)}$(b) $f^{(2)}$ (c) $f^{(3)}$ (d) $f^{(4)}$ .

Dört test fonksiyonuna göre genetik algoritmanın önerdiği sonuçlar Tablo 2'de verilmiştir. Burada $x_0$ değeri belirlenen aralıkta fonksiyonun gerçek bir köküdür. $x$ değeri ise aynı aralıkta önerilen kök değeridir.

Tablo 2. Önerilen genetik tabanlı denklem çözümü için test sonuçları

| Test fonksiyonu | İterasyon sayısı | $y_0=f(x_0)$ | Önerilen $x$ değeri | $y=f(x)$ |
|---|---|---|---|---|
| $f^{(1)}$ | 68 | 0 | $-6{,}0.10^{-6}$ | $-6{,}0.10^{-6}$ |
| $f^{(2)}$ | 53 | 0 | 3.926991 | $5{,}4.10^{-6}$ |
| $f^{(3)}$ | 35 | 0 | $3{,}8.10^{-6}$ | $3{,}8.10^{-6}$ |
| $f^{(4)}$ | 95 | 0 | 2.547874 | $3{,}4.10^{-6}$ |

Bu tabloya göre örneğin; $f^{(4)}$ test fonksiyonu için genetik algoritma 95 iterasyon sonunda arama işlemini tamamlamıştır ve çözüm olarak $x = 2.547874$ kök değerini bulmuştur. Bu kök değeri fonksiyonda yerine koyulduğunda sonuç $y = f^{(4)}(x) = 3{,}4.10^{-6}$ olarak hesaplanmıştır. Böylece hesaplanan $y$ değerleri ve gerçek $y_0$ değerleri karşılaştırıldığında ortaya koyulan yaklaşımla fonksiyonun köküne oldukça yüksek bir doğrulukta yaklaşıldığını ispatlamaktadır. Diğer üç fonksiyon için de benzer sonuçların olduğu görülmektedir.

GA'lar arama uzayında belirli kurallar dahilinde rastgele olarak gezindiğinden dolayı programın yeniden icra edilmesi her defasında farklı sonuçlar doğurmaktadır. Bu yüzden, önerilen yöntemin gerçek çözümleri ile karşılaştırılabilmesi için GA, her fonksiyon için art arda 25 kez çalıştırılmış ve sonuçlar en iyi çözüm, en kötü çözüm, çözüm ortalaması ve standart sapması ve ayrıca iterasyon ortalaması ve standart sapması olarak Tablo 3'de verilmiştir.

Tablo 3. 25 kez çalıştırma için test fonksiyonlarına ait hesaplanan sonuçlar

| Test fonksiyonları / İstatistikler | $f^{(1)}$ | $y(f^{(1)})$ | $f^{(2)}$ | $y(f^{(2)})$ | $f^{(3)}$ | $y(f^{(3)})$ | $f^{(4)}$ | $y(f^{(4)})$ |
|---|---|---|---|---|---|---|---|---|
| En iyi çözüm | $8{,}9.10^{-8}$ | $8{,}9.10^{-8}$ | 3,926991 | $5{,}4.10^{-6}$ | $-2{,}9.10^{-7}$ | $-2{,}9.10^{-7}$ | 2.547872 | $3{,}8.10^{-7}$ |
| En kötü çözüm | $9{,}4.10^{-6}$ | $9{,}4.10^{-6}$ | 3,927246 | $7{,}6.10^{-3}$ | $-9{,}8.10^{-6}$ | $-9{,}8.10^{-6}$ | 2.547877 | $8{,}1.10^{-6}$ |
| Çözüm ortalaması | $4{,}7.10^{-6}$ | | 3,927003 | | $-1{,}3.10^{-6}$ | | 2.547871 | |
| Çözüm standart sapması | $4{,}1.10^{-6}$ | | $5{,}7.10^{-5}$ | | $5{,}8.10^{-6}$ | | $4{,}8.10^{-6}$ | |
| Ortalama iterasyon | 133 | | 300 | | 53 | | 227 | |
| İterasyon standart sapması | 80 | | 245 | | 20 | | 271 | |

Bu tabloda $f^{(i)}$ sütunu fonksiyonlara ilişkin elde edilen kök çözüm değerlerini gösterirken $y\left(f^{(i)}\right)$ sütunu da elde edilen çözüm değerlerinin fonksiyon değerlerini göstermektedir. Tablo 3' den elde edilen en iyi çözüm ile en kötü çözüm arasında çok büyük farkların olmadığı görülmektedir. Ayrıca, çözüm ortalamasının da en iyi çözüme yakın olması, her çalıştırmada en iyi çözüm ya da en iyi çözüme yakın değerler elde edilmesi anlamına gelmektedir. Popülasyon büyüklüğü ve mutasyon oranı gibi GA parametrelerini değiştirmek de farklı sonuçlar üretecektir. Aynı zamanda GA'ların rastgelelik doğası çıktısının tahmini hakkında bilgi sahibi olmamıza engel olmaktadır. Sonuç olarak, GA'lar iyi sonuçlar vermelerine rağmen karmaşık bir fonksiyonun optimizasyonunda ürettiği çözümler en iyi çözümler olmayabilir [11].

## 5. Sonuçlar

Bu çalışmada denklem çözüm tekniklerine alternatif olarak genetik algoritma yaklaşımı önerilmiştir. Bu amaçla, test fonksiyonları kullanılarak testler gerçekleştirilmiştir. Elde edilen sonuçlar ortaya konulan yöntemle fonksiyonların köküne oldukça yüksek bir hassasiyetle yaklaşıldığını göstermiştir. Ayrıca, geliştirilen genetik algoritmalar yardımıyla klasik yöntemlerdeki türev, integral ve karmaşık işlemlere gerek duyulmamıştır. Bu yüzden, kök hesaplama işlemi, bu çalışmada faydalanılan test fonksiyonlarından başka analitik olarak türevi olmayan ve sayısal yöntemlerle türev işlemi ağır hesaplama yükü gerektiren birçok karmaşık fonksiyon için de gerçekleştirilebilir.

## Kaynaklar

1. Sönmez, M., Sayısal Analiz Notları, s. 5-18, Aksaray Üniversitesi, 2008.
2. Üstün, O., Yıldız, İ., Geri-Yayılmalı Öğrenme Algoritmasındaki Öğrenme Parametrelerinin Genetik Algoritma İle Belirlenmesi, SDU International Technologic Science, Vol. 1, No. 2, 61-73, October, 2009.
3. Bayram, M., Nümerik Analiz,s. 48, Birsen Yayınevi, İstanbul, 2009.
4. Vatansever, F., Batık, Z., Genetik Algoritma Tabanlı Denklem Çözümleri, 5. Uluslararası İleri Teknolojiler Sempozyumu (İATS'09), 13-15 Mayıs, Karabük, Türkiye, 2009.
4. Angeline, P.J., An introduction to the special track on genetic and evolutionary programming, IEEE Expert Intelligent Systems and their Applications 10, 6-10, June, 1995.
5. Goldberg, D.E., Genetic Algorithms in Search, Optimization and Machine Learning, A.B.D., Addison Wesley Publishing Company Inc., 1989.
6. Holland, J.H., Miller, J.H., Artificial Adaptive Agents in Economic Theory, The American Economic Review, Vol. 81, No. 2, May, 1991.
7. Haupt, R.L., Haupt, S.E., Practical Genetic Algorithms, Wiley-Interscience Publication, 2nd edition, 2004.
8. Öztürk, N., Çelik, E., Application of Genetic Algorithms to Core Loss Coefficient Extraction, Progress in Electromagnetics Research M, Vol. 19, 133-146, 2011.
9. Çetin, E.,Yiğit, T., Genetic Algorithm Based On-line Tuning of a PI Controller for a Switched Reluctance Motor Drive, Electric Power Components and Systems, 35:6, 675-691, 2007.
10. Haupt, R.L., Werner, D.H., Genetic Algorithms in Electromagnetics, A Wiley-Interscience Puplication, 53-54, 2007.