

İleri Teknoloji Bilimleri Dergisi
Journal of Advanced Technology Sciences
ISSN:2147-3455

PARÇACIK SÜRÜ OPTİMİZASYONU İLE DOĞRUSAL OLMAYAN DENKLEM KÖKLERİNİN BULUNMASI VE GENETİK ALGORİTMA İLE MUKAYESESİ

Pakize Erdoğan

Bilgisayar Mühendisliği Bölümü, Mühendislik Fakültesi, Düzce Üniversitesi, Düzce, TÜRKİYE

Özet- Bu çalışmada son yıllarda oldukça çok çalışılan sezgisel algoritmalarından biri olan Parçacık Sürü Optimizasyonu(PSO), doğrusal olmayan denklemlerin köklerinin bulunmasında kullanılmıştır. Öncelikle kök bulma problemi bir optimizasyon problemine dönüştürülmüştür. Çalışmada tek değişkenli doğrusal olmayan denklemler ele alınmıştır. PSO'nun doğrusal olmayan denklemleri çözme başarısı, literatürde ki bir çalışmada kullanılan dört adet doğrusal olmayan denkleme ek olarak farklı bir denklem ile de test edilmiş ve sonuçlar, çözümün optimalitesi, iterasyon sayısı ve çözüm süresi bakımından karşılaştırılarak sunulmuştur. PSO kök bulmada oldukça başarılı sonuçlar vermiştir. Genetik Algoritma(GA)'ya göre daha hızlı yakınsadığı için, kökleri daha az sayıda iterasyon ile aynı doğrulukta bulmaktadır. Sonuç olarak, kök bulma yöntemlerinin dezavantajlarına sahip olmayan ve hızlı bir sezgisel optimizasyon algoritması olan PSO'nun Nümerik Analiz'de kök bulma yöntemi olarak kullanılabileceği gösterilmiştir.

Anahtar Kelimeler: Parçacık Sürü Optimizasyonu, Sezgisel Araştırma, Doğrusal Olmayan Denklemler

FINDING THE ROOTS OF NON-LINEAR EQUATIONS WITH PARTICLE SWARM OPTIMIZATION AND COMPARISON TO GENETIC ALGORITHM

Abstract- In this study, Particle Swarm Optimization (PSO), one of the most studied heuristic algorithms, is used finding the roots of non-linear equations. Firstly root finding problem was converted to an optimization problem. In this study, non-linear equations which have one independent variable have been considered. The performance of the non-linear equation solution of PSO is tested with four non-linear equations in the literature and an extra equation. The performances of algorithms have been tested and results have been submitted in view of

optimality, the number of iterations and solution time. PSO has been quite successful for finding roots. Since PSO converges more fast than Genetic Algorithm(GA), it has find the roots less iteration with the same accuracy. As a result, since it hasn't the disadvantage of the root finding methods and it is a fast heuristic algorithm, PSO can be used as a root finding method in Numerical Analysis.

Keywords: Particle Swarm Optimization, Heuristic Search, Non-Linear Equations

1. GİRİŞ (INTRODUCTION)

Optimizasyon, çeşitli değişkenlere bağlı olarak değişen bir amaç fonksiyonunu verilen kısıtlayıcı şartlar altında en optimum(en küçük veya en büyük) yapan değerlerin bulunması olarak tanımlanır. Özellikle mühendislik problemleri başta olmak üzere, işletme, ekonomi ve tasarım problemlerinde optimizasyona ihtiyaç duyulur.

Problemin boyutu arttıkça, çözüm süresinin de üstel olarak artması, bazı problemlerin klasik yöntemlerle çözümünün çok zor olması, klasik yöntemlerin türev bilgisi gerektirmesi ve bu yöntemlerle sadece kısıtlı sayıda gerçek hayat probleminin çözülebilmesi sebebi ile 1970'li yıllarda yeni çözüm yöntemleri arayışına gidilmiş ve sezgisel yöntemler çalışılmaya başlanmıştır. Genetik Algoritma ile başlayan, Benzetim Tavlama ve Karınca Kolonisi Optimizasyonu ile devam eden sezgisel yöntemler, 2000'li yıllarda çok artmıştır. Özellikle son yıllarda kompleks ve zor optimizasyon problemlerinin çözümü için bir çok sezgisel algoritma geliştirilmiştir. Sezgisel algoritmalar günümüzde temel olarak tabiatın, hayvan davranışlarından ve fiziksel süreçlerden esinlenilerek geliştirilmektedir.

Sezgisel araştırma yöntemlerinin birçoğu doğadan esinlenilerek geliştirilmiştir. Bu yöntemlerin bazıları doğadaki fiziksel yasaları veya süreçleri temel alırken, büyük bir kısmı canlı varlıkların sosyal etkileşimlerini örnek almıştır. Genetik Algoritma(Genetic Algorithm)(GA)[1] canlı varlıkların genlerinin nesilden nesile aktarımından, Benzetim Tavlama(Simulated Annealing)(SA)[2], metallerin tavlama sürecinden, Yerçekimi Araştırma Algoritması(Gravity Search Algorithm)(GSA)[3] ise Newton'un yerçekimi yasasından esinlenilerek geliştirilmiştir.

Canlıların davranışlarının benzetimi ile geliştirilmiş algoritmalarından ilki 1991 yılında Dorigo tarafından geliştirilen, Karınca Kolonisi Optimizasyonu(Ant Colony Optimization)(ACO)dur. Dorigo, karıncaların sürü zekası ile yiyeceğe giden en kısa yolu bulmalarını optimizasyon problemlerinin çözümüne uyarlamıştır[4]. Başta Gezin Satıcı olmak üzere bir çok ayrık optimizasyon problemlerinin çözümünde başarılı uygulamaları olan ACO algoritması adapte edilerek sürekli optimizasyon problemlerinin çözümünde de kullanılmış ve başarılı çözümler ürettiği görülmüştür.

Eberhart ve Kenedy tarafından 1995 yılında sürekli optimizasyon problemleri için geliştirilen Parçacık Sürü Optimizasyonu(Particle Swarm Optimization)[5] kuş ve balık sürülerinin yiyecek arama davranışını simüle eden bir algoritmadır.

Karaboğa tarafından 2004 yılında geliştirilen Yapay Arı Kolonisi(Artificial Bee Colony)(ABC), arı kolonilerinin kovana en yakın nektarı bulma davranışlarını simüle etmiş ve kablosuz ağ

optimizasyonu, sayısal filtre optimizasyonu, elektrik güç sistemleri optimizasyonu ve daha birçok mühendislik problemine çözüm getirmiştir[6].

Hayvan davranışlarını simüle ederek son yıllarda geliştirilen algoritmalarından bazıları da Guguk kuşlarını temel almıştır [7].

Sayısal analiz yöntemleri, çözülmesi zor veya imkânsız olan matematiksel problemleri, dört işlem ve algoritma yolu ile çözmeyi amaçlar. Özellikle bilgisayarların gelişimine paralel olarak, mühendislik problemlerinin çözümünde Sayısal Analiz yöntemleri kullanılır. Elektrik Güç sistemleri[8], Tıp Mühendisliği[9], Makine Mühendisliği[10] gibi birçok alanda karşılaşılan problemler sayısal analiz yöntemleri ile çözülür.

Kök bulma, bir fonksiyonu sıfır yapan bağımsız değişken değerinin bulunması işlemi olup, literatürde birçok kök bulma yöntemi geliştirilmiştir[11]. Bunun yanında Sayısal analiz yöntemlerinin dezavantajları sebebi ile bazı yapay zeka yöntemleri Sayısal Analiz problemlerinin çözümünde kullanılmaktadır. Genetik Algoritma ile doğrusal olmayan denklem köklerinin bulunması[12], yine Genetik Algoritma ile doğrusal olmayan denklem sistemlerinin köklerinin bulunması[13] ve yapay zeka ile 3X3'lük doğrusal olmayan denklem sistemlerinin çözümü[14] bu çalışmalardan bazılarıdır.

Bu çalışmada da, Doğrusal olmayan denklem köklerinin, zeki arama yöntemlerinden PSO ile bulunması amaçlanmıştır. Kök bulma yöntemlerinden literatürde de en çok kullanılan Aralık Yarılama(Bisection) yöntemi, Kiriş(Secant) yöntemi ve Newton Raphson yöntemi kullanılmıştır. Çalışmanın 2. Bölümünde PSO ve 3. Bölümünde adı geçen Sayısal Analiz yöntemleri ele alınmıştır. 4. Bölümde ise literatürde yer alan benzer bir çalışmada[12] kullanılan dört adet problem ve bir adet özel problem Sayısal Analiz yöntemleri ve PSO ile çözülmüştür. Sonuç kısmında ise PSO'nun kök bulma başarısı Sayısal analiz yöntemleri ile karşılaştırılarak incelenmiştir.

2. PARÇACIK SÜRÜ OPTİMİZASYONU

Sezgisel optimizasyon alanında bireylerin, bireysel ve ortak davranışlarının incelenerek modellenmesine dayalı bir çok algoritma geliştirilmiştir. Hayvanların bazı bireysel hareketleri ve kitlesel hareketleri ile yiyecek bulma ve yön bulma gibi problem çözme yeteneklerine sürü zekâsı denilmektedir. ACO, karıncaların, PSO kuş ve balık sürülerinin, ABC bal arılarının yiyecek aramada kullandıkları sürü zekâsını modeller.

PSO, kuş ve balık sürülerinin davranışlarından esinlenilerek, sürekli optimizasyon problemleri için geliştirilmiş bir metoddur. Balık sürüleri, kuş sürüleri ve diğer sosyal hayvanlar incelendiğinde bu hayvanların yiyecek ararken etkileşim içerisinde oldukları ve birinin yiyecek bulması ile diğerlerinin de sürüden kopmadan konumlarını yiyeceğin olduğu yöne çevirdikleri ve hızlarını da buna göre güncelledikleri görülmüştür. Bu sosyal etkileşim PSO ile modellenmiştir[15].

PSO, popülasyon temelli bir algoritma olup, birden fazla çözümle çalışır. PSO'da popülasyona sürü(swarm) adı verilir. PSO'yu diğer popülasyon temelli yöntemlere göre öne çıkaran özelliği basitliği ve hızlı yakınsamasıdır. Sürüyü oluşturan her bir kuş bir parçacık (particle) olarak adlandırılır. PSO'da her bir parçacık bir aday çözümdür. Bu parçacıklar uzayda hareket ederek en uygun çözümü bulmayı hedefler. Parçacıklar araştırma uzayında iki önemli parametreye göre hareket ederler. Her iterasyonda önce her bireyin kendi en iyisi *pbest* ve tüm bireylerin en iyisi *gbest* elde

edilir. n adet parçacık ile çalışan bir PSO'da her iterasyonda n adet *pbest* var iken sadece bir *gbest* mevcuttur. Algoritmanın başında her parçacık için başlangıç değeri aynı zamanda *pbest* değeridir. Sonraki iterasyonlarda diğer parçacıkların durumuna göre konumu güncellenir ve o anki konum önceki *pbest* ile karşılaştırılır. Eğer *pbest*'ten daha iyi bir çözüm ise artık yeni çözüm *pbest* olarak atanır.

PSO'da sürü bir başlangıç çözümü ile başlar. Başlangıçta her bir parçacığın hız ve konum değeri vardır. Parçacık konumları tanımlı aralık içindeki değişken değerleridir. Parçacıklar hızlarını hem kendilerine hem de sürüye göre ayarlayarak hareket ederler. Parçacıklar her bir iterasyonda optimum çözüme biraz daha yaklaşır. *i*. Parçacığa ait hız ve konum formülleri denklem (1) ve (2)'de verildiği gibidir[16].

$$v_i^{k+1} = K(v_i^k + \varphi_1 \text{rand}() (p_{best_i}^k - x_i^k) + \varphi_2 \text{rand}() (g_{best} - x_i^k)) \quad (1)$$

Bu denklemde *k* iterasyon sayısıdır. v_i^k *i*. parçacığın *k*. iterasyondaki hızıdır. v_i^{k+1} ise bir sonraki iterasyon için hızıdır. *K*, φ_1 ve φ_2 birbirine bağlı katsayılarıdır. *K* kısıtlama faktörü parçacıkların hareketlerindeki osilasyonu sönmüleme etkisine sahiptir. φ_1 and φ_2 sosyal(social) ve bilişsel(cognitive) parametrelerdir. rand düzgün rastgele dağılımdan rastgele çekilen (0-1) arası bir sayıdır.

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (2)$$

K, φ_1 and φ_2 'ye bağlı bir sabittir. Denklem (3)'te *K* sabitinin formülü verilmiştir.

$$K = \frac{2}{\left| 2 - \varphi - \sqrt{\varphi^2 - 4\varphi} \right|} \quad \varphi = \varphi_1 + \varphi_2 \quad \varphi > 4 \quad (3)$$

φ_1 , parçacığın kendi tecrübelerine göre hareket etmesini(bilişsel), φ_2 ise sürüdeki diğer parçacıkların tecrübelerine göre(sosyal) hareket etmesini sağlar. Şekil 1'de PSO'nun sözde kodu verilmiştir.

```

Başlangıç P parçacığı oluştur.
P parçacığın kendi en iyisini ata. Pbesti
P parçacığın global en iyisini ata. Gbest
Tekrar et
    i=1'den P parçacık sayısına kadar tekrar et
        Eğer i. Parçacığın uygunluğu Gbest'ten daha iyi ise Gbest olarak ata.
        Eğer i. Parçacığın uygunluğu daha önceki değerlerinin en iyisi ise Pbesti
        olarak ata.
        Hız formülünü güncelle
        Konum formülünü güncelle
    dön
Durma koşulu gerçekleşti ise dur

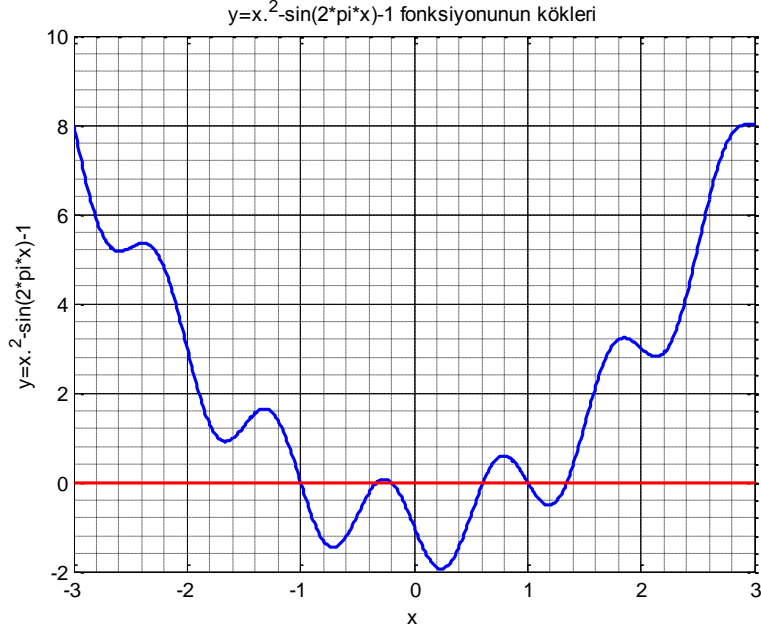
```

Şekil 1. PSO'nun sözde kodu

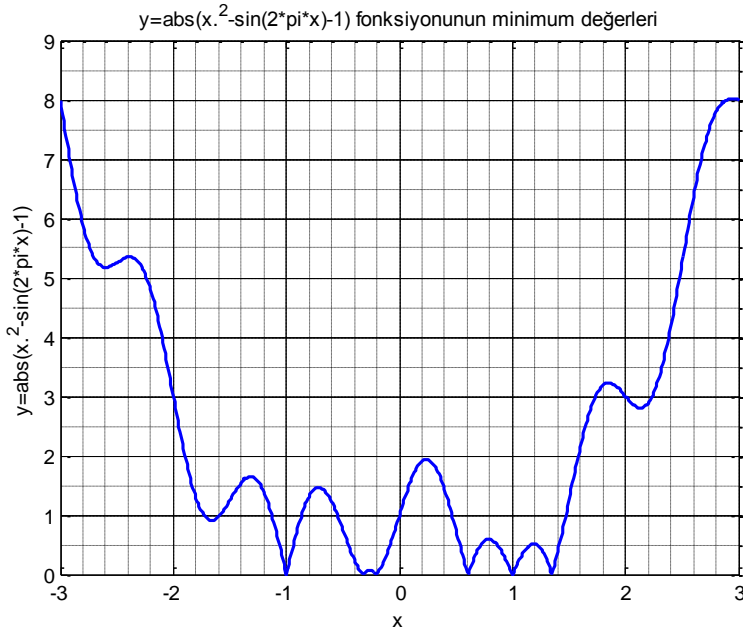
3. KÖK BULMA YÖNTEMLERİ

Kök bulma yöntemleri bir fonksiyonun değerini sıfır yapan bağımsız değişken değeri veya değerlerini bulmayı amaçlar. Aslında optimizasyon ile kök bulma yöntemleri iç içe geçmiş iki konudur denilebilir. Bir fonksiyonun optimum noktası, o fonksiyonun türevinin köküdür. Tepe veya dip noktalarda fonksiyon değişimi(türev) sıfırdır. Ters durumda bir fonksiyonun kökleri, mutlak olarak

fonksiyonun alacağı en minimum değerdir. Daha açık bir ifade ile, “Bir fonksiyonun kökleri, fonksiyon değerini mutlak olarak minimum yapan değer veya değerlerdir.” Yukarıdaki ifadelerden, “Optimizasyon problemlerinin türevi alınarak kök bulma yöntemleri ile çözülebileceği gibi, optimizasyon yöntemleri ile de kök bulmak mümkündür.” Aşağıda Şekil 2’de bir fonksiyonun kökleri Şekil 3’de ise aynı fonksiyonun mutlak değeri görülmektedir.



Şekil 2. $y = x^2 - \sin(2\pi x) - 1$ fonksiyonunun kökleri



Şekil 3. $y = |x^2 - \sin(2\pi x) - 1|$ fonksiyonunun minimum değerleri

Bu çalışmada da üç temel kök bulma yöntemine ek olarak hızlı yakınsayan bir algoritma olarak PSO, alternatif kök bulma yöntemi olarak kullanılmıştır. Aşağıda kısaca Aralık Yarılama, Kiriş ve Newton-Rapson yöntemleri ele alınmıştır.

3.1. Aralık yarılama

Kök bulmada kullanılan kapalı yöntemlerdendir. Bu yöntemin kökü bulabilmesi için bir tek kökü içeren bir aralık değeri ile çözüme başlanır. Köke yakın olan aralık değeri dikkate alınmadan, kökün bulunduğu aralık, sürekli küçültülerek köke yaklaşılır. Bu sebeple yakınsaması yavaş bir kök bulma yöntemidir. Bir fonksiyon değerinin sıfır olabilmesi için negatif bir değerden, pozitif veya tersi bir geçiş olması gerekir ilkesini kullanarak, kökün olduğu aralık tespit edilir. Köke belirlenen hata değeri kadar yakınsanana kadar algoritma devam eder[17]. Eğer kök x eksenini kesmeyip, dokunuyor ise (Negatiften pozitive veya tersi bir geçiş olmadan) bu yöntem ile kök bulunamaz. Aralık yarılama yönteminin algoritması Şekil 4.'de verilmiştir.

1. Kökü içeren $[x_a, x_u]$ aralıkları ve ε hata oranı belirle.
2. $x_y = (x_a + x_u) / 2$
3. $f(x_y) * f(x_a) < 0$ ise $x_u = x_y$ 5. adıma git.
4. Eğer $f(x_y) * f(x_a) > 0$ ise $x_a = x_y$ 5. adıma git.
5. $f(x_a) * f(x_y) = 0$ veya $abs(f(x_a) - f(x_y)) < \varepsilon$ ise kökü yaz ve dur. Yoksa 2.adıma git

Şekil 4. Aralık Yarılama Algoritması

x_a alt değer, x_u üst değer, x_y yeni ara değer, ε yaklaşım hatasıdır.

3.2. Kiriş Yöntemi

Kiriş Yöntemi de çözüm için iki değer gerektirdiği için kapalı bir yöntem olarak değerlendirilse de açık yöntem olarak da kabul edilebilir. Newton-Rapson Yönteminin sonlu farklarla uygulandığı bir yöntemdir. Fonksiyonun kökünün olduğu aralık $[x_0, x_1]$ olmak üzere Taylor serisi yardımı ile denklem (4) ile yeni x,y değerleri elde edilir. Belirlenen hata oranına ulaşılan kadar çözüme devam edilir[18].

$$x = x_0 - \frac{(x_1 - x_0)}{(y_1 - y_0)} y_0 \quad (4)$$

Kiriş yönteminin algoritması Şekil 5'te verilmiştir.

1. Kökü içeren $[x_0, x_1]$ aralıkları ve ε hata oranı belirle.
2. Denklem (4)'den x değerini hesapla.
3. Köke ε hatası oranında yaklaşıldı ise kökü yaz ve dur. Yoksa $x_0 = x_1$, $x_1 = x$ olarak 2.adıma git

Şekil 5. Kiriş yöntemi algoritması

x_0 , x_1 başlangıç değerler, ε yaklaşım hatasıdır.

3.3. Newton-Rapson Yöntemi

Newton-Rapson algoritması türevelere dayalı ve hızlı yakınsayan açık yöntemlerdendir. Bu yöntemde $f(x)$ fonksiyonunun kökünü bulmak için Taylor serisine açılırsa;

$$f(x_i + h) = f(x_i) + h_i f'(x_i) = 0$$

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} \quad (5)$$

bulunur. Newton-Rapson Yönteminin algoritması Şekil 6. de verilmiştir[19].

1. Bir başlangıç değeri x_i , ϵ hata oranı belirle.
2. Denklem (5) 'den x_{i+1} değeri hesapla.
3. Köke ϵ hatası oranında yaklaşıldı ise kökü yaz ve dur.
Yoksa $x_{i+1}=x_i$ olarak 2.adıma git

Şekil 6. Newton-Rapson Yöntemi Algoritması

x_i , x_{i+1} ardışık x değerleri , ϵ yaklaşım hatasıdır

4. PSO VE KÖK BULMA YÖNTEMLERİ İLE BULUNAN SONUÇLAR

Literatürde bir çalışmada Genetik Algoritma ile çözümü gerçekleştirilen problemler ve bir özel problem ele alınarak, PSO ve sayısal yöntemlerle çözülmüştür. Tablo 1'de ele alınan fonksiyonlar verilmiştir. PSO'da ve sayısal yöntemlerde fonksiyonun mutlak hata ile köke $e=1e-5$ kadar yaklaşması durdurma kriteri olarak belirlenmiştir. Kapalı yöntemlerde Tablo 1'deki aralık değerleri kullanılırken, Açık yöntemlerde ki başlangıç değeri ise verilen aralık değerinde her bir çalıştırmada rastgele bir başlangıç değeri ile çözüm yapılmıştır.

Tablo 1. Kökleri bulunacak fonksiyonlar ve aralık değerleri [12]

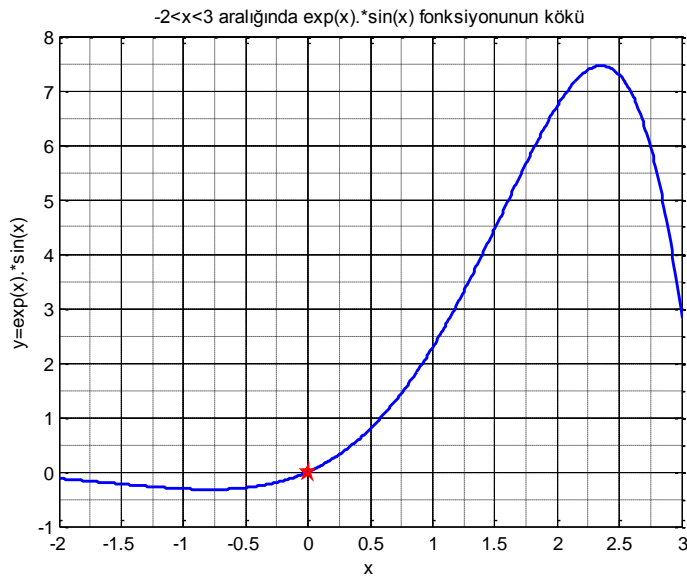
Problem no	Fonksiyon	Aralık değeri	Aralıktaki Kök Değeri
1	$f(x) = e^x \sin(x)$	$-2 \leq x < 3$	0
2	$f(x) = 150e^{-x/2} (\cos(x) - \sin(x))$	$1 \leq x < 6$	3.9270
3	$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	$-10 \leq x < 10$	0
4	$f(x) = \ln(x-1) + \sin(x-3)$	$2 \leq x < 15$	2.547874
5	$f(x) = -x^2(x^2 - 4)(x - 5)$	$-1.5 \leq x < 1$	0

PSO ve GA algoritmasında kullanılan parametreler aşağıda Tablo 2'de verilmiştir.

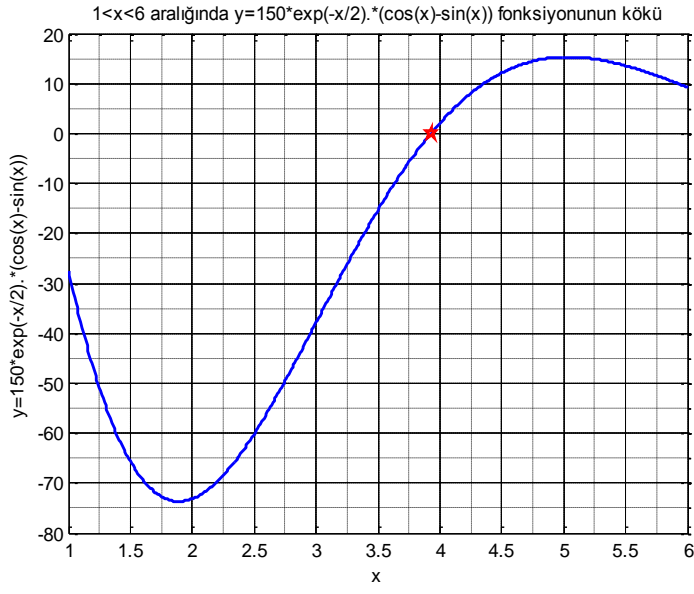
Tablo 2. PSO ve GA algoritmalarında kullanılan parametreler

	GA	PSO
Populasyon büyüklüğü	8	20
Maksimum iterasyon	1000	1000
Mutasyon oranı	0.1	-
Kromozom uzunluğu	25	-
Seçim metodu	Turnuva	-
ϕ_1 (Bilişsel faktör)		2
ϕ_2 (Sosyal faktör)		2.1
K(Kısıtlama faktörü)	-	0.72984378

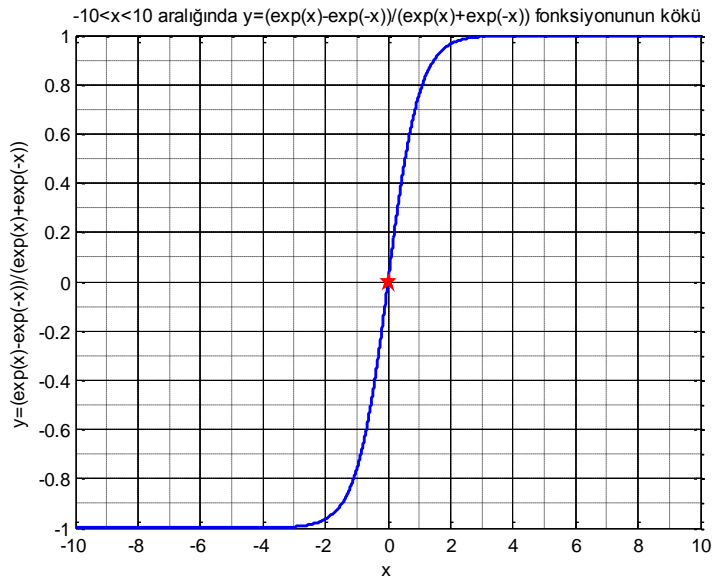
Tablo 1’de verilen kökleri bulunacak fonksiyonlara ait grafikler Şekil 7,8, 9,10 ve 11’de verilmiştir.



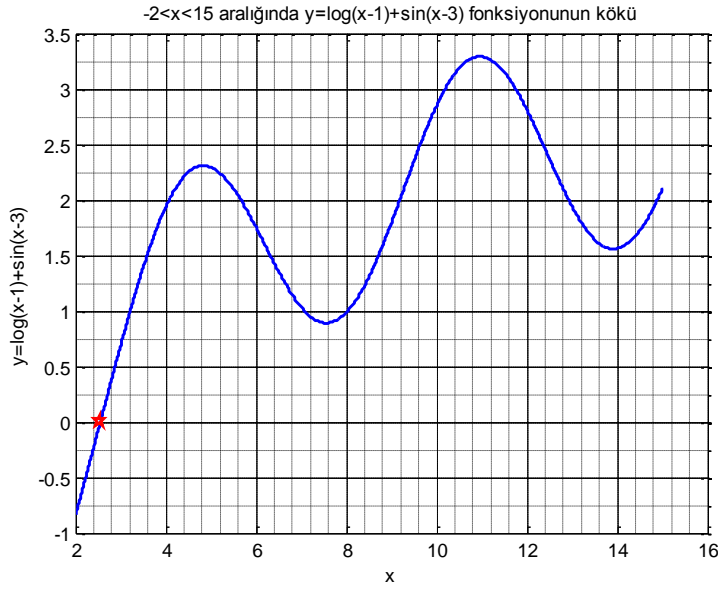
Şekil 7. 1. Nolu fonksiyonun grafiği



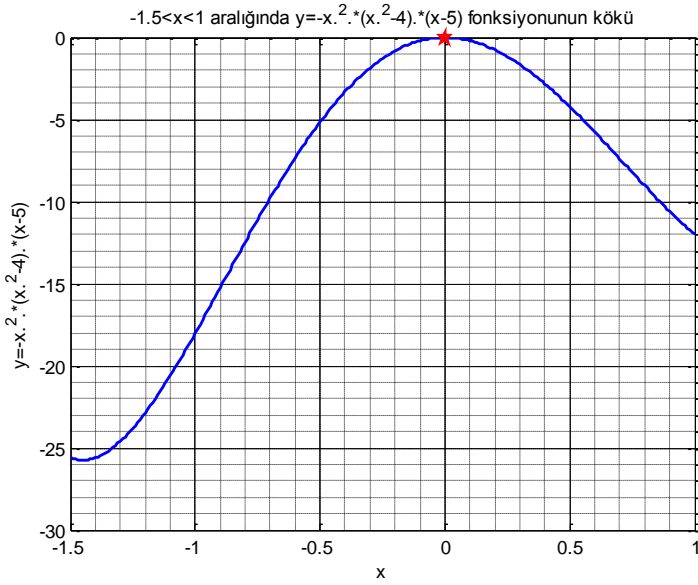
Şekil 8. 2. Nolu fonksiyonun grafiği



Şekil 9. 3. Nolu fonksiyonun grafiği



Şekil 10. 4. Nolu fonksiyonun grafiği



Şekil 11. 5. Nolu fonksiyonun grafiği

Tüm simülasyonlar Intel i7-3612 QM 2.1GHz işlemci ve 4GB RAM'e sahip bir bilgisayarda gerçekleştirilmiştir. Programlama ortamı olarak Matlab®7.12 kullanılmıştır. Kodlar 30 kez çalıştırılarak her bir çalışmadaki çalışma süresi ve sonuçlar Excel® dosyasına kaydedilmiştir. Aşağıda sırası ile Tablo 3, Tablo 4, Tablo 5, Tablo 6 ve Tablo 7'de problemlerin PSO, GA(İlgili çalışmadaki değerler alınmıştır.), Aralık yarılama, Kiriş ve Newton-Rapson ile çözümlerinin, çözüm süresi, iterasyon sayısı ve bulunan kök değerleri sunulmuştur.

Tablo 3. 1 nolu Problemin PSO, GA ve Kök Bulma Yöntemleri ile Çözümü

Yöntem	Bulunan Kök			İterasyon Sayısı			Çözüm Süresi(sn)		
	En iyi	En kötü	Ort.	En iyi	En kötü	Ort.	En iyi	En kötü	Ort.
PSO	3,55E-06	-9,16E-06	9,37E-06	38	89	58	0,008472	0,044365	0,02095
GA	-6E-6	-	-	68	-	-	-	-	-
Aralık Yar.	-1,90E-6	-1,90E-06	-1,90E-06	19	19	19	0,001542	0,010361	0,00192
Kiriş*	-3,14159	-3,14159	-3,14159	9	9	9	0,001206	0,046004	0,002294
Newton-Rapson**	5,47E-10	9,14E-06	-	2	5	4,50	0,001906	0,010962	0,002456

*Kiriş metodu hiçbir çözümde kök olarak sıfır değerini bulamamıştır.

**Newton-Rapson 30 çözümün 16'sında kök olarak sıfır değerini bulmuştur. Diğer çözümlerde pi ve katlarındaki çözümleri bulmuştur.

Tablo 4. 2 nolu Problemin PSO ve Kök Bulma Yöntemleri ile Çözümü

Yöntem	Bulunan Kök			İterasyon Sayısı			Çözüm Süresi(sn)		
	En iyi	En kötü	Ort.	En iyi	En kötü	Ort.	En iyi	En kötü	Ort.
PSO	3,926991	3,926991	3,926991	39	102	80,36	0,017238	0,017238	0,017238
GA	3,926991	-	-	53	-	-	-	-	-
Aralık Yar.	3,926991	3,926991	3,926991	24	24	24	0,002451	0,012649	0,002979
Kiriş*	7,068583	7,068583	7,068583	10	10	10	0,001927	0,012259	0,002561
Newton-Rapson**	3,926991	3,926991	3,926991	2	5	3,56	0,003162	0,014907	0,004455

*Kiriş metodu hiçbir çözümde kök olarak aralıktaki kök değerini bulamamıştır.

**Newton-Rapson 30 çözümün 16'sında aralıktaki kök değerini bulmuştur. Farklı başlangıç değerleri için sekiz farklı kök bulmuştur.

Tablo 5. 3 nolu Problemin PSO ve Kök Bulma Yöntemleri ile Çözümü

Yöntemler	Bulunan Kök			İterasyon Sayısı			Çözüm Süresi(sn)		
	En iyi	En kötü	Ort.	En iyi	En kötü	Ort.	En iyi	En kötü	Ort.
PSO*	-9,68E-06	9,10E-06	5,46E-07	48	88	66,89	0,017944	0,0581180	0,0259425
GA	3,80E-6	-	-	35	-	-	-	-	-
Aralık Yar.	0	0	0	1	1	1	0,00104	0,00988	0,001526
Kiriş	0	0	0	1	1	1	0,001031	0,00917	0,001491
Newton-Rapson	-6,73E-06	-1,72E-10	-1,24E-06	4	502	63,82	0,002023	0,015152	0,004004

Tablo 6. 4 nolu Problemin PSO ve Kök Bulma Yöntemleri ile Çözümü

Yöntemler	Bulunan Kök			İterasyon Sayısı			Çözüm Süresi(sn)		
	En iyi	En kötü	Ort.	En iyi	En kötü	Ort.	En iyi	En kötü	Ort.
PSO	2,547865	2,547876	2,547871	28	74	55	0,012137	0,083517	0,024498
GA	2,547874	-	-	95	-	-	-	-	-
Aralık Yar.	2,54787	2,54787	2,54787	20	20	20	0,002426	0,005104	0,002867
Kiriş*	-124.815	-	-	19	19	19	0,002178	0,055277	0,004575
Newton-Rapson**	2,547872	2,547875	2,547872	2	21	6,90	0,002607	0,012523	0,00386

Kiriş* Verilen başlangıç değerleri ile 30 simülasyon sonucunda da bu aralıktaki kökü bulamamıştır. Newto-Rapson**30 simülasyondan 10'unda aralıktaki kök değeri bulunmuş. Diğer simülasyonlarda fonksiyonun diğer kökleri bulunmuş veya algoritma köke yakınsamamıştır.

Tablo 7. 5 nolu Problemin PSO ve Kök Bulma Yöntemleri ile Çözümü

Yöntemler	Bulunan Kök			İterasyon Sayısı			Çözüm Süresi		
	En iyi	En kötü	Ort.	En iyi	En kötü	Ort.	En iyi	En kötü	Ort.
PSO	-0,00057	0,000694	-6,55E-05	1	34	17,55	0,001634	0,108469	0,010497
GA*	0.005	-	-	51	-	-	-	-	-
Aralık Yar.**	-	-	-	-	-	-	-	-	-
Kiriş***	2	2	2	14	14	14	0,001119	0,008723	0,00169
Newton-Rapson****	-0,00069	0,000682	0,000197	7	10	9,39	0,001867	0,016486	0,003078

GA* Matlab GA komutu ile bulunmuştur.

Aralık Yar.** Maksimum itersyonda kok bulunamamıştır.

Kiriş*** Tüm simülasyonlarda aralıktaki kök yerine aralık dışındaki kök bulunmuştur.

Newton**** Bir simülasyonda aralık dışında bir kök bulunmuştur.

5. SONUÇ VE TARTIŞMA (CONCLUSION AND DISCUSSION)

Bu çalışmada, genelde tüm sezgisel araştırma yöntemlerinin birer kök bulma yöntemi olarak kullanılabilceği, PSO algoritması ile gösterilmiştir. “Bir fonksiyonun kökleri onu sıfır yapan değerlerdir ifadesi”, “Bir fonksiyonun kökleri mutlak olarak o fonksiyonu minimum yapan değerlerdir” ifadesine dönüştürüldüğünde tüm optimizasyon yöntemlerinin aslında birer kök bulma

yöntemi olarak ele alınabileceklerini gösterir. Bu çalışmada PSO algoritması, diğer sezgisel araştırma yöntemlerine göre daha az parametre gerektirdiği ve daha hızlı yakınsadığı için seçilmiştir.

Tablo'daki sonuçlardan görüleceği üzere Nümerik Kök Bulma yöntemlerinden açık yöntemler başlangıç değer olarak verilen aralıkta bir değer seçilse de, kök olarak bu aralık dışındaki farklı kök değerlerini bulabilmektedir. Buda açık yöntemlerin dezavantajlarından birtanesidir. Bu çalışmada da açık yöntemlerden olan Newton-Rapson ve Kiriş yöntemleri, verilen aralıktaki kök değerlerinden başka kökler de bulmuştur. Bu sebeple açık yöntemlerden Newton-Rapson ve Kiriş için sonuçlar verilmiş olmasına rağmen, PSO'nun performansı tam anlamı ile sadece Aralık Yarılama metodu ile karşılaştırılabilir.

Sonuç olarak, bu çalışmada ele alınan PSO algoritması verilen aralıkta, multak minimum değeri arayarak, kök bulmaktadır ve kapalı yöntemler arasında değerlendirilebilir. Kapalı yöntemlerle kıyaslandığında ise PSO kapalı yöntemlerin sahip olduğu dezavantajlara sahip değildir. Aralık yarılama yönteminde aralık değerlerinden birinin köke yakın olması durumunda yakınsama çok yavaş olur. Aralık yarılama algoritmasının sahip olduğu dezavantajlardan birisi x eksenini kesmeden, sadece x eksenine dokunan kökleri bulamamasıdır. Bu sebepten Aralık yarılama algoritması 5 nolu probleme çözüm bulamamıştır. PSO ise, bu dezavantajlara sahip olmayan, bir kök bulma yöntemi olarak kullanılabilir. Çözüm süreleri karşılaştırıldığında ise ele alınan fonksiyona göre değişmekle birlikte kök buma yöntemlerinin PSO'ya göre daha hızlı oldukları görülmüştür. Bunun sebebi ise PSO'nun aynı anda n adet parçacık ile çalışmasıdır. Çalışmada problem çözme başarısına odaklanıldığı için parçacık sayısı ile ilgili bir çalışma yapılmamıştır. Çözümler sabit parçacık sayısı ile gerçekleştirilmiştir.

6. KAYNAKLAR (REFERENCES)

- [1] Lim, T. Y.,(2014). Structured population genetic algorithms: a literature survey, *Artif Intell Rev* 41, 385–399.
- [2] Kirkpatrick,S., Gelatt,C., Vecchi, M., (1983). Optimization by simulated annealing, *Science* 220, 671–680.
- [3] Rashedi, E., Nezamabadi-pour, H., Saryazdi, S., (2009). GSA: a gravitational search algorithm. *Inf Sci*;179:2232–48.
- [4] Blum, C., M. Dorigo, Stützle, T., (2004). *Ant Colony Optimization*,. MIT Press, Cambridge, MA, *Artificial Intelligence*,165(2), 261-264.
- [5] Kennedy, J., Eberhart, R., (1995). Particle swarm optimization, *Neural Networks, Proceedings. IEEE International Conference on* , 4(),1942-1948.
- [6] Karaboga, D., Gorkemli, B., (2014). A quick artificial bee colony (qABC) algorithm and its performance on optimization problems, *Applied Soft Computing*, 23, Pages 227-238.
- [7] Rajabioun, R.,(2011). Cuckoo Optimization Algorithm, *Applied Soft Computing*, 11(8), 5508-5518.

- [8] Kilyeni, S., Barbulescu, C., Simo,A.,(2015). POWERNUM. Power Engineering Numerical Methods Educational Software Library, Procedia - Social and Behavioral Sciences, 191(2), 1783-1789.
- [9] Almeida, H. A., Bártolo, P. J., (2014). Design of tissue engineering scaffolds based on hyperbolic surfaces: Structural numerical evaluation, Medical Engineering & Physics, Volume 36, Issue 8,1033-1040, ISSN 1350-4533
- [10] Santhosh, B., Padmanabhan, C., Narayanan,S., (2014).Numeric-analytic solutions of the smooth and discontinuous oscillator, International Journal of Mechanical Sciences, 84,102-119.
- [11] Proinov, P. D.,(2016). Relationships between different types of initial conditions for simultaneous root finding methods, Applied Mathematics Letters, 52() 102-111.
- [12] Ozturk, N., Çelik, E.,(2012). Polinom olmayan denklemlerin genetik algoritma tabanlı çözümü, Erciyes Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 28(4), 322-328.
- [13] El-Emary, İ.M.M, Abd El-Kareem, M.M.,(2008), Towards Using Genetic Algorithm for Solving Nonlinear Equation Systems, World Applied Sciences Journal 5 (3): 282-289.
- [14] Goulianas, K., Margaris, A., Adamopoulos,M.,(2013). Finding all real roots of 3×3 nonlinear algebraic systems using neural networks, Applied Mathematics and Computation, 219(9),4444-4464.
- [15]Clerc, M.(1999). The swarm and the queen: towards a deterministic and adaptive particle swarm optimization. Evolutionary Computation, CEC 99. Proceedings of the 1999 Congress on ,3(),1951-1957.
- [16] Parrott, D., Xiaodong Li(2006). Locating and tracking multiple dynamic optima by a particle swarm model using speciation, Evolutionary Computation, IEEE Transactions on , 10(4),440,458.
- [17] Mansouri,P., Asady,B., Gupta,N.,(2015). The Bisection–Artificial Bee Colony algorithm to solve Fixed point problems, Applied Soft Computing,26, 143-148.
- [18]Magreñán, A.A., Argyros, I.K., (2016). New improved convergence analysis for the secant method, Mathematics and Computers in Simulation, 119, Pages 161-170.
- [19] Krismadinata, N.A.R., Ping,H.W., Selvaraj,J.,(2013). Elimination of Harmonics in Photovoltaic Seven-level Inverter with Newton-raphson Optimization, Procedia Environmental Sciences, 17, 519-528.