



A solution of some commonly used optimization functions by a hybrid BFGS-PSO algorithm

Mehmet Cengiz^{1*}, Emre Çomak²

¹Computer Engineering Department, Aydın Adnan Menderes University, Aydın, 09100, Turkey

²Computer Engineering Department, Burdur Mehmet Akif Ersoy University, Burdur, 15030, Turkey

Highlights:

- Designing a new hybrid optimization algorithm
- Providing a new perspective to achieve optimized results
- Designing an optimization algorithm for faster reach optimized solutions

Keywords:

- Numerical Optimization
- Heuristic Optimization
- PSO
- BFGS
- Hybrid Optimization

Article Info:

Research Article
Received: 20.09.2019
Accepted: 15.11.2020

DOI:

10.17341/gazimmfd.622588

Correspondence:

Author: Mehmet Cengiz
e-mail:
mehmet.cengiz@adu.edu.tr
phone: +90 554 827 8361

Graphical/Tabular Abstract

In this study, the development of a hybrid optimization method has been emphasized in order to eliminate the deficiencies in the structure of heuristic and numerical methods. In our applications, Particle Swarm Optimization (PSO) method and Broydon-Fletcher-Goldfarb-Shanno (BFGS) method are aimed to reach the optimum solution more accurately and faster. In the algorithm of the developed optimization method, the solution is firstly sought with BFGS. Optimum result points are screened using PSO. Until the final result is reached, the intermediate solution points are continuously transferred between the BFGS and the PSO. Using the original BFGS and the original PSO, two different methods are proposed: BFGS first, then PSO hybrid method, and vice versa hybrid method (first PSO then BFGS).

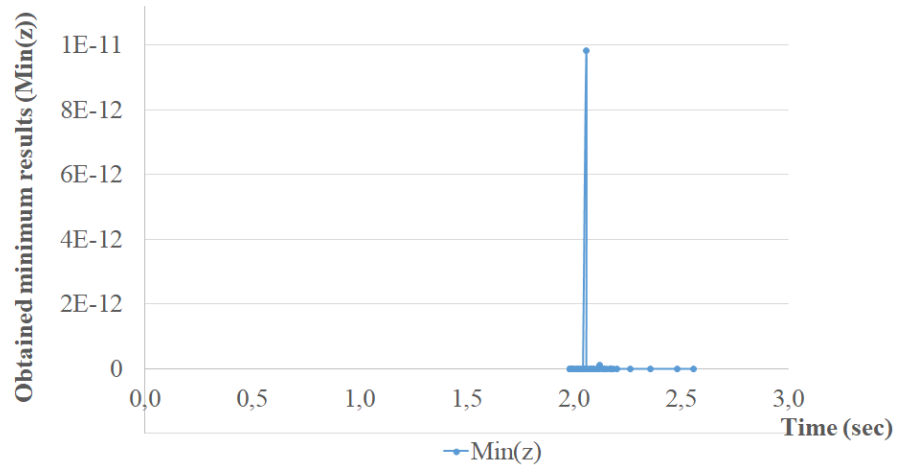


Figure A. Recycled aggregate production

Purpose: Although numerical optimization algorithms guarantee the exact solution, they remain relatively slow. On the other hand, heuristic optimization algorithms are fast but they cannot guarantee the exact solution. From this point of view, it is aimed to design an algorithm that achieves both fast and exact solutions.

Theory and Methods: In this study, it is proposed that numerical and heuristic methods can be hybridized and exact solutions can be reached faster. In the literature, hybrid optimization algorithms apply heuristic and then numerical approaches. In the developed method, the opposite way was followed.

Results: The developed method was tested on the functions used in the literature for testing purposes. Accordingly, as the test function becomes more complex, the speed of the solution is slow in milliseconds according to the comparison algorithm.

Conclusion: Consequently, the combination of numerical methods such as BFGS and heuristic methods such as PSO may be an appropriate approach for solution accuracy. As intended in this study, heuristics can be hybridized with other numerical methods. On the other hand, the literature research and the proposed algorithm proved that hybrid algorithms obtained better results than standard heuristics and numerical methods.



Yaygın kullanılan bazı optimizasyon fonksiyonlarının melez bir BFGS-PSO algoritması ile çözümü

Mehmet Cengiz^{1*}, Emre Çomak²

¹Aydın Adnan Menderes Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, 09100 Efeler Aydın, Türkiye

²Burdur Mehmet Akif Ersoy Üniversitesi, Mühendislik-Mimarlık Fakültesi, Bilgisayar Mühendisliği Bölümü, 15030, Burdur, Türkiye

Ö N E Ç İ K A N L A R

- Yeni bir melez optimizasyon tasarımı
- Optimize edilmiş sonuçlar elde etmek için yeni bir bakış açısı sağlamak
- Optimize edilmiş çözümlere daha hızlı ulaşan bir optimizasyon algoritması tasarlamak

Makale Bilgileri

Araştırma Makalesi
Geliş: 20.09.2019
Kabul: 15.11.2020

DOI:

10.17341/gazimmfd.622588

Anahtar Kelimeler:

Nümerik optimizasyon,
sezgisel optimizasyon,
PSO,
BFGS,
melez optimizasyon

ÖZET

Bu çalışmada, sezgisel ve sayısal yöntemlerin yapısında var olan eksiklikleri gidermek amacıyla melez bir optimizasyon yönteminin geliştirilmesi üzerinde durulmuştur. Sezgisel yöntemler kesin çözümü garanti edemezler. Ancak, sayısal yöntemlerden göreceli olarak daha hızlı çalıştırılabilirler. Öte yandan, sayısal yöntemler ise güçlü matematiksel çözümler içerdikleri için kesin çözüme ulaşabilmektedirler. Uygulamalarımızda, sezgisel optimizasyon yöntemlerinden Parçacık Sürü Optimizasyonu (PSO) yöntemi ile sayısal optimizasyon yöntemlerinden Broydon-Fletcher-Goldfarb-Shanno (BFGS) yöntemi birleştirilerek optimum çözüme daha kesin ve daha hızlı bir şekilde ulaşılması amaçlanmıştır. Geliştirilen optimizasyon yönteminin algoritmasında, çözüm öncelikle BFGS ile aranmaktadır. Böylece, amaç fonksiyonu için en küçük veya en büyük noktalar belirlenmektedir. Daha sonra, PSO ile bu noktalar arasında eleme işlemi gerçekleştirilmektedir. Nihai sonuca ulaşılan kadar, ara çözüm noktaları BFGS ile PSO arasında sürekli aktarılmaktadır. Özgün BFGS ve özgün PSO kullanılarak önerilen yöntem (önce BFGS sonra PSO) ve tersi melez yöntem (önce PSO sonra BFGS) olmak üzere iki farklı yöntem; araştırmacılar arasında sıklıkla kullanılan test fonksiyonları üzerinde çalıştırılarak uygulanmıştır. Uygulama sonuçları, hem önerilen melez algoritmanın hem de tersi melez algoritmanın küresel en iyi değere kısa süreler içinde ulaştığını göstermiştir.

A solution of some commonly used optimization functions by a hybrid BFGS-PSO algorithm

H I G H L I G H T S

- Designing a new hybrid optimization algorithm
- Providing a new perspective to achieve optimized results
- Designing an optimization algorithm for faster reach optimized solutions

Article Info

Research Article
Received: 20.09.2019
Accepted: 15.11.2020

DOI:

10.17341/gazimmfd.622588

Keywords:

Numerical optimization,
heuristic optimization,
PSO,
BFGS,
hybrid optimization

ABSTRACT

In this study, the development of a hybrid optimization method has been emphasized in order to eliminate the deficiencies in the structure of heuristic and numerical methods. Heuristic methods cannot guarantee the exact solution. However, they can run relatively faster than numerical methods. On the other hand, numerical methods contain strong mathematical solutions and can reach a definite solution. In our applications, one of the heuristic optimization methods, Particle Swarm Optimization (PSO) method and Broydon-Fletcher-Goldfarb-Shanno (BFGS) method, which is one of the numerical optimization methods, are aimed to reach the optimum solution more accurately and faster. In the algorithm of the developed optimization method, the solution is firstly sought with BFGS. Thus, the smallest or maximum points for the objective function are determined. Then these points are screened using PSO. Until the final result is reached, the intermediate solution points are continuously transferred between the BFGS and the PSO. Using the original BFGS and the original PSO, two different methods are proposed: BFGS first, then PSO hybrid method, and vice versa hybrid method (first PSO then BFGS); It has been applied by working on the frequently used test functions among the researchers. Experimental results have indicated that both proposed hybrid algorithm and another hybrid algorithm implemented in reverse way can reach the global optimal results in short times.

1. GİRİŞ (INTRODUCTION)

Mühendislik ve diğer alanlardaki herhangi bir sorunun çözümü için genellikle birden çok çözüm yolu bulunmasına rağmen bunların tümü hedeflenen sonuç değildir. Bu sebeple *optimizasyon* yani *en uygun duruma getirme* kavramı ortaya çıkmıştır. Belirli sınırlamaları sağlayacak şekilde, bilinmeyen parametre değerlerinin en uygun şekilde bulunmasını içeren herhangi bir sorun, optimizasyon sorunu olarak adlandırılabilir [1]. Matematiksel olarak konuşursak, optimizasyon; bir işlevin (fonksiyon) kendi değişkenlerine bağımlı olarak en küçük veya en büyük noktasının bulunması işlemidir [2].

Bu çalışmada çok boyutlu, doğrusal olmayan, sayısal optimizasyon yöntemlerinden Variable Metric tipi yöntemlere (VMM) dahil olan Broydon-Fletcher-Goldfarb-Shanno (BFGS) yöntemi ile sezgisel optimizasyon yöntemlerinden Parçacık Sürü Optimizasyonu (PSO) geliştirilerek melez bir yöntem önerilmiştir. BFGS için gerekli olan kısıtlama ise tek boyutlu doğrusal olmayan sayısal çözüm yöntemlerinden Altın Bölme Yöntemi (ABY) ile sağlanmıştır. Aşağıdaki iki paragrafta anlatılan özelliklerinden dolayı ve araştırmacılar tarafından sıklıkla kullanıldığı için, çalışmamızda BFGS ve PSO yöntemleri tercih edilmiştir.

BFGS yöntemi kısıtlamasız, doğrusal olmayan optimizasyon sorunlarını çözmek için geliştirilmiş, tekrarlamalı bir algoritmadır. Bu yöntemde temel olarak incelenecek işlev üzerinde türevler alınır ve türevin sifıra eşit olduğu nokta aranır [3].

PSO yöntemi Kennedy ve Eberhart [4] tarafından kuş ve balık sürülerinin yem bulma davranışları örneklenerek, *doğrusal* olmayan, sınırlı ve/veya sınırsız, çok modellenmiş işlevlerin çözümü için geliştirilmiş bir yöntemdir. PSO yöntemi diğer sezgisel yöntemlere göre daha az değişken içerir. Her parçacığın hız ve konum bilgilerinin tutulması yeterlidir.

“2. Literatür Taraması” bölümünde belirtildiği gibi, bazı çalışmalarda sezgisel ve sayısal yöntemler bir arada kullanılmaktadır. Bu çalışmalarda genellikle sayısal yöntem sezgisel algoritmanın yalnız bir adımını oluşturmaktadır ve sonraki adımlarda sezgisel algoritma özünde olduğu gibi çalışmaya devam etmektedir. Böylece; sezgisel algoritmanın yerel arama kabiliyeti geliştirilip, yerel minimuma takılma ve erken yakınsama gibi problemleri azaltılmaya çalışılmaktadır. Bu çalışmada ise, sezgisel ve sayısal yöntemler bağımsız olarak birbirlerini takiben (sırayla önce BFGS sonra PSO) çalışacak şekilde melez bir yapıda tasarlanmıştır. Sayısal yöntemin çalışması sezgisel yöntemin çalışma şeklini ve benzer şekilde sezgisel yöntemin çalışması da sayısal yöntemin çalışma şeklini değiştirmemektedir. Ayrıca önerilen yapının karşılaştırma algoritması (sırayla önce PSO sonra BFGS) da tasarlanmış ve önerilen algoritma ile sonuçları karşılaştırılmıştır. Bu

sayede bir sezgisel yöntemin içerisine ilave bir hesaplama yükü getirilmeden sezgisel ve sayısal yöntemlerin güçlü yanlarından yararlanılmıştır. “4. Yöntem” bölümünde algoritmamızın sözde kodu ve çalışma örneği detaylı olarak verilmiştir.

Yayının ikinci bölümünde üzerinde çalıştığımız konular hakkında literatürde daha önceden yapılmış çalışmalara genel olarak değinilmiştir. Üçüncü bölümde ise kullanılan tekniklerin üzerinde durulmuştur. Dördüncü bölümde geliştirilen yöntem anlatılmıştır. Beşinci bölümde ise önerilen yöntem literatürde sınamak için üretilmiş işlevlerle denenmiş, başka yöntemlerle karşılaştırılmış ve sonuçları verilmiştir. Altıncı ve son bölümde de sonuçlar tartışılmıştır.

2. LİTERATÜR TARAMASI (LITERATURE REVIEW)

Çalışılan konu üzerine ilk çalışmalar 1970'lere kadar dayanmaktadır. Fletcher [5] VMM'ler için yeni bir yaklaşım geliştirmiştir. Lowrie ve Wah [6] sayısal optimizasyon yöntemlerini geliştirmek amacıyla sezgisel bir öğrenme yöntemi önermiştir. VMM davranışları üzerine Davidon [7] da bir çalışma yapmıştır. Semeter ve Mendillo [8] atmosferik salınım tomografisi değerleri BFGS kullanılarak çıkarılmıştır.

2000lerin başında Li ve Fukushima [9] kısıtlamasız optimizasyon sorunları için BFGS yöntemini değiştirerek küresel yakınsama sağlamıştır. Morales [10] kısıtlı bellekli BFGS (L-BFGS) ile sayısal bir çalışma ortaya koymuştur. Bu çalışmada SNOPT yazılımı [11] kullanılarak sonuçlar elde edilmiş ve başarılı bulunmuştur.

Optimizasyon yöntemlerini denemek için literatürde birçok işlev bulunmaktadır. Bu işlevlerden 20 kadarı Molga ve Smutnicki [12] tarafından yapılan bir çalışmada, dönüm noktaları ve kısıtlı olduğu noktalar belirtilerek, sergilenmiştir. Aynı yıl Victoire ve Jeyakumar [13] PSO üzerinde geliştirme yaparak “Dynamic Economic Dispatch” sorununu çözmek için “Deterministically Guided PSO (DGPSO)” yöntemini ortaya atmıştır. Yine aynı yılda Xia vd. [14] tarafından BFGS üzerinde geliştirme yapılmıştır ve sonuçları “Optimal Power Flow” sorunu ile denenmiştir.

Nawi vd [15] Geri Beslemeli Yapay Sinir Ağları için BFGS temelli yeni bir öğrenme algoritması geliştirmişlerdir ve verimli sonuçlar aldıklarını iddia etmişlerdir. Andrei [16] eşlenik gradyan yöntemi çerçevesinde ön şart tekniği ile belleksiz ölçekli BFGS yöntemini birleştirmiştir. Sezgisel yöntemlerde ise Perez ve Behdinan [17] yapısal tasarım optimizasyonları için PSO kullanmıştır.

Xiao vd. [18] büyük ölçekli sorunları çözmek için L-BFGS üzerinde değişiklikler yapmıştır. Aynı yıl Yuan ve Lu [19] tarafından doğrusal olmayan bakışlı (simetrik) başka bir tane BFGS yöntemi daha ortaya atılmıştır. Zhao vd. [20] PSO yöntemini birçok sürüyü ortaklaşa çalıştıracak şekilde tekrar tasarlamışlardır. Özsağlam ve Çunkaş [21] tarafından

ve yine aynı yılda PSO yönteminin Genetik Algoritmalar ve Diferansiyel Gelişim Algoritması gibi diğer sezgisel yöntemlerle karşılaştırılması yapılmış ve PSO yönteminin daha iyi sonuçlar verdiği gözlemlenmiştir.

Thomas ve Reed [22] yarı Newton bir yöntemle PSO yöntemini birleştirerek L-PSO yöntemini ortaya atmıştır. Li ve Tan [23] PSO temelli kapsamlı bir öğrenme algoritması CLPSO ile BFGS yöntemlerini birleştirerek Support Vector Machine parametrelerini ayarlamışlardır.

Li vd. [24] PSO ve BFGS yöntemlerini birleştirerek çok modelli küresel optimizasyon sorunlarını çözmeye çalışmışlardır. Öte yandan Nezhad vd. [25] yine PSO ve değiştirilmiş BFGS yöntemlerini birleştirerek doğrusal olmayan sorunları çözmüşlerdir ve yöntemi sınamak için de geliştirilmiş Lagrange sorununu kullanmışlardır. Yine aynı yıl Shi vd. [26] işlev çözümlerine yönelik Hücrel Parçacık Sürü Optimizasyonu isimli yeni bir yöntem önererek Hücrel Otomata ile PSO yöntemlerini birleştirmişlerdir.

Uriarte vd. [27] Parçacık Sürü Optimizasyonunda bulunan yerel minimum ve erken yakınsama problemlerini çözmek istedikler. Bu amaçla geriye yayılma yaklaşımını PSO'daki iyileşmeyi ölçen bir fonksiyon olarak PSO'nun içine yerleştirdiler. Sonuçların geleneksel PSO'ya kıyasla daha iyi olduğunu iddia ettiler.

Shatnawi vd. [28] kutupsal koordinatlarda işlem yapan bir PSO tasarladılar. Ayrıca, bu yeni PSO'nun eksikliğini gidermek için yeni bir başlatma yöntemi geliştirdiler. Yöntem, yaygın kullanılan işlevlerin bazılarında küçük başarımlarını gösterdi. Ozcan [29] sekiz adet yaygın çalışılan işlev üzerinde Parçacık Sürü Optimizasyonu ve Diferansiyel Evrim algoritmasının ortalama ve standart sapma değerleri ile karşılaştırma yapmıştır. Parçacık Sürü Optimizasyonun daha hızlı çalıştığını ve algoritmaların sonuçlarının birbirine yakın çıktığını göstermiştir.

Mehmood vd. [30] ikili Parçacık Sürü Optimizasyonunun hızlanma katsayılarını uygunluk değerine göre bireysel olarak değiştiren yeni bir algoritma önerdiler. Önerilen algoritma yaygın olarak kullanılan dört işlev üzerinde daha iyi yakınsama özelliği gösterdi. Cavalca ve Fernandes [31] türev temelli bir sayısal yöntemi Parçacık Sürü Optimizasyonuna dâhil etmişlerdir. Önerilen yöntemin sonuçlarını üç farklı algoritmanın sonuçları ile sekiz yaygın işlev üzerinde karşılaştırdılar. Sonuçların diğerlerinden üstün olduğunu gösterdiler. Badem vd. [32] ise Yapay Arı Kolonisi ile sınırlı bellekli BFGS algoritmalarını birleştirerek melez bir algoritma önerdiler. Uygulama sonuçlarının istatistiksel analizi ile önerilen algoritmanın sık kullanılan işlevler üzerindeki etkinliğini ortaya koydular. Yine aynı yıl Çınaroğlu ve Bulut [33] standart K-means ve PSO kümeleme algoritmalarının başlangıç merkezlerini daha iyi seçebilmek amacıyla yeni bir yöntem geliştirmişlerdir. Çeşitli değerlendirme ölçütlerine göre, yeni yöntem standart yöntemlerden daha üstün performans ortaya koymuştur. Geraldine ve Dinakaran [34] da kuantum fiziğinde yer alan

tünelleme etkisini PSO'ya uyarladılar. Rastrigin, Rosenbrock ve Griewangk işlevlerinin de bulunduğu bazı işlevlerde PSO'nun başarımının arttığını ortaya koydular.

Çimen ve Boz [35] ölü zamanlı ve geri tepmeli bir ikinci dereceden sistemin (gri kutu) parametrelerini Cuckoo Arama, Parçacık Sürü Optimizasyonu (PSO) ve FireFly algoritmaları ile belirlemiştir. Yazarlar elde edilen sonuçların literatürde bulunan Genetik Algoritma ile hesaplanan sonuçlara göre gerçeğe daha yakın olduğunu göstermiştir. Jarrah vd. [36] Yapay Arı Kolonisi (YAK) algoritmasının yerel arama kabiliyetini güçlendirmek için türev temelli çalışan beta tepe tırmanma algoritmasını YAK'a dahil ettiler. Yaygın kullanılan işlevler üzerinde sekiz farklı algoritmadan daha iyi sonuçlar elde ettiler.

Yukarıda açıklanan çalışmalarda görüldüğü üzere, PSO ve BFGS yöntemleri hem geleneksel algoritma düzenlerinde hem de bu geleneksel algoritmaları iyileştirilerek birçok alanda kullanılmışlardır. Bu alanlar doğrudan bir problemin çözümüne yönelik olanları ve bunların yanı sıra diğer makine öğrenmesi yöntemlerinde ihtiyaç duyulan çözümleri de içerebilmektedir. Algoritmalarda gerçekleştirilen iyileştirmeler ise bu uygulamaların yapısına uygun olacak şekilde tasarlanmıştır. Ayrıca, PSO ve BFGS yöntemleri ve bu yöntemlerin alternatifleri aynı problemler üzerinde çalıştırılarak sonuçları karşılaştırılmıştır. Bütün bu literatür çalışmalarının hedefi, uygulamaya özel problemin çözümünü en iyi verecek yöntem veya melez yöntemin belirlenmesidir.

3. DENEYSEL YÖNTEMLER (EXPERIMENTAL METHODS)

Daha önceden de bahsedildiği gibi geliştirilen yöntemde sayısal yöntemlerden BFGS ile sezgisel yöntemlerden PSO birleştirilmiştir. Ayrıca, BFGS yöntemini sınırlamak için de, ABY kullanılmıştır.

3.1. Altın Bölme Yöntemi (Golden Partition Method)

Altın oran; matematik ve sanatta, bir bütünün parçaları arasında gözlemlenen, uyum açısından en yetkin boyutları verdiği sanılan geometrik ve sayısal bir oran bağıntısıdır [37]. Oranın bağıntısı Eş. 1'de verilmiştir. Bu değer yaklaşık olarak 1,61803'e denk gelmektedir.

$$\text{Altın Oran} = (1 + \sqrt{5})/2 \quad (1)$$

Altın Bölme Yöntemi (ABY) aralık daraltma ve kısıtlama yöntemleri içerisinde en çok kullanılanlardan birisidir. Bu yöntem, aralığı uçlarından aynı oranda daraltır. Optimize edilecek işlevin şekil ve özelliklerinden bağımsız olarak çalışır. En önemlisi de, çözüme belli bir toleransla ulaşmak için gerekli döngü sayısı önceden tahmin edilebilir. Tablo 1'de ABY algoritması görülmektedir [38].

Tablo 1'de x_{alt} ve x_{ust} değerleri hedef işlevi incelemeye başlamak için seçilecek sınırları göstermektedir. τ değeri,

Altın Oran'dan üretilen bir sabittir. ε değeri, x_{alt} ve x_{ust} değerlerine bağlı bir değişken olup döngünün tur sayısı olan N 'yi belirlemede kullanılmaktadır.

Tablo 1. Altın Bölme Yöntemi Algoritması
(The Golden Section Search Algorithm)

Adım 1: x_{alt} ve x_{ust} belirle
 $\tau = 0.38197$; (Altın Oran'dan)
 $\varepsilon = \text{tolerans} = (\Delta x)_{\text{son}} / (x_{alt} - x_{ust})$;
 $N = \text{dönüş sayısı} = -2.078 * \ln \varepsilon$;
 $k = 1$;
Adım 2: $x_1 = (1 - \tau) x_{alt} + \tau x_{ust}$;
 $f_1 = f(x_1)$;
 $x_2 = \tau x_{alt} + (1 - \tau) x_{ust}$;
 $f_2 = f(x_2)$;
Adım 3: Eğer $k < N$ ise
Eğer $f_1 > f_2$ ise
 $x_{alt} = x_1$;
 $x_1 = x_2$;
 $f_1 = f_2$;
 $x_2 = \tau x_{alt} + (1 - \tau) x_{ust}$;
 $f_2 = f(x_2)$;
Eğer $f_2 > f_1$ ise
 $x_{ust} = x_2$;
 $x_2 = x_1$;
 $f_2 = f_1$;
 $x_1 = (1 - \tau) x_{alt} + \tau x_{ust}$;
 $f_1 = f(x_1)$;
 $k = k + 1$;
Adım 3'e git.

3.2. Broydon-Fletcher-Goldfarb-Shanno Yöntemi (Broydon-Fletcher-Goldfarb-Shanno Method)

Bu yöntem VMM ailesinin bir üyesi olup, en çok tercih edilenlerindedir. Braydon-Fletcher-Goldfarb-Shanno (BFGS) yönteminde başlangıç olarak bakışimli ve artı yönde tanımlı bir düzey (matris) atanır ve genellikle birim düzey seçilir. Çözümüne ulaşıldığında başlangıç düzeyi Hessian düzeyine eşit olur. Tablo 2'de algoritma görülmektedir [38].

Tablo 2'de *Adım 2* içerisinde görülen $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{s}_k \mathbf{p}_k$ işlemi çok boyutlu sayısal optimizasyon sorunlarında genel güncelleme kuralı olup buradaki \mathbf{p} arama yönünü ve \mathbf{s} ise adım aralığını belirtmektedir. Bu aşamada bir boyutlu optimizasyon tekniği olarak ABY kullanılmıştır.

3.3. Parçacık Sürü Optimizasyonu Yöntemi (Particle Swarm Optimization Method)

Parçacık Sürü Optimizasyonu (PSO) yönteminin çalışma ilkesine göre, çözülecek işlevde birden fazla başlangıç noktasından çözüm aranmaya başlanır. Bu her başlangıç noktası bir parçacıktır ve parçacıklardan çözüme yakın olma derecesine göre sürünün yani tüm parçacıkların hızı ve yönü sürekli olarak güncellenir. Bu sayede yerel çözümlere takılma olasılığı da azalır. Her ne kadar PSO diğer yöntemlere göre daha az değişken içerse de, Tablo 3'te bu değişkenler açıklanmıştır.

Tablo 2. Broydon-Fletcher-Goldfarb-Shanno Algoritması
(The Broydon-Fletcher-Goldfarb-Shanno Algorithm)

Adım 1: Başlangıç noktası = \mathbf{x}_1 , Başlangıç düzeyi = \mathbf{A} ,
Döngü tur sayısı = N belirle.
Sonlandırma koşulları $\varepsilon_1, \varepsilon_2, \varepsilon_3$ değerlerini belirle.
 $k=1$ ile döngüyü başlat.
Adım 2: \mathbf{x}_k noktasındaki gradyant vektör = $\nabla f(\mathbf{x}_k)$ hesapla.
 $\mathbf{A}_k \mathbf{p}_k = -\nabla f(\mathbf{x}_k)$
Bir boyutlu optimizasyon ile $f(\mathbf{x}_k + \mathbf{s}_k \mathbf{p}_k)$ 'yi minimum yapan \mathbf{s}_k 'yi bul.
 $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{s}_k \mathbf{p}_k$
Adım 3: $\Delta f = f(\mathbf{x}_{k+1}) - f(\mathbf{x}_k)$; $\Delta \mathbf{x} = \mathbf{x}_{k+1} - \mathbf{x}_k$
Eğer $|\Delta f| \leq \varepsilon_1$ ise fonksiyon değişmediği için dur.
Eğer $\|\Delta \mathbf{x}\| \leq \varepsilon_2$ ise değişkenler değişmediği için dur.
Eğer $\|\nabla f(\mathbf{x}_{k+1})\| \leq \varepsilon_3$ ise sonuca yakınsadığı için dur.
Eğer $k+1 = N$ ise döngü bittiği için dur.
Adım 4: Eğer algoritma sonlandırılmamışsa
 $\mathbf{y} = \nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k)$
 $\Delta \mathbf{x} = \mathbf{s}_k \mathbf{p}_k$
 $\mathbf{B} = (\mathbf{y} \mathbf{y}^T) / (\mathbf{y}^T / \Delta \mathbf{x})$
 $\mathbf{C} = (\nabla f(\mathbf{x}_k) \nabla f(\mathbf{x}_k)^T) / (\nabla f(\mathbf{x}_k)^T \mathbf{p}_k)$
 $\mathbf{A}_{k+1} = \mathbf{A}_k + \mathbf{B} + \mathbf{C}$
 $k = k + 1$
Adım 2'ye git.

Tablo 3. Parçacık Sürü Optimizasyonu değişkenleri
(The Variables of Particle Swarm Optimization)

c_1 : Parçacık hızlandırma etkeni
 c_2 : Sürü hızlandırma etkeni
 r_1, r_2 : $[0, 1]$ Aralığında rastgele üretilen değerler
 w : Atalet ağırlığı
 $P_{eniyi(i)}$: i parçacığının bireysel en iyi değeri
 G_{eniyi} : Sürünün en iyi değeri
 $X_i(k)$: i parçacığının k anındaki konum bilgisi
 $V_i(k)$: i parçacığının k anındaki hız bilgisi
 PS : Parçacık sayısı
 N : Döngü (tur) sayısı

Tablo 3'te bulunan atalet ağırlığı (w) bir önceki hız bileşeninin şimdiki hız bileşeni üzerine etkisini hesaplamak için kullanılır. Atalet ağırlığının en uygun değerinin hesaplanması için birden fazla yöntem bulunmaktadır. Bu çalışmada, rastgele olasılıkların da göz önünde bulunması için, aşağıda gösterilen Eş. 2 kullanılmıştır.

$$w = 0,5 + \frac{\text{rastgele}(\cdot)}{2} \quad (2)$$

Eş. 2'de gösterilen $\text{rastgele}(\cdot)$ ifadesi, $[0, 1]$ arasında rastgele bir sayı üretilmesini belirtmektedir. Eş. 3 vasıtasıyla bu atalet ağırlığı değeri, Tablo 3'te anlatılan değişkenler ve parçacığın o anki konumu kullanılarak parçacıkların hızları güncellenmektedir. Eş. 4'te ise güncel hıza bağlı olarak güncel konumu bulunmuştur.

$$V_i(k+1) = w * V_i(k) + c_1 * r_1(k) * (P_{eniyi(i)}(k) - X_i(k)) + c_2 * r_2(k) * (G_{eniyi}(k) - X_i(k)) \quad (3)$$

$$X_i(k+1) = X_i(k) + V_i(k+1) \quad (4)$$

Eş. 3 ve Eş. 4'teki güncelleme işlemlerinin de yer aldığı PSO algoritması, Tablo 4'te gösterilmektedir.

Tablo 4. Parçacık Sürü Optimizasyonu algoritması
(The Particle Swarm Optimization Algorithm)

Adım 1: Parçacık sayısını (PS) belirle.
Her parçacık için ilk konumları (X) ata.
Parçacıklara ilk hız (V) değerlerini ata.
Algoritma parametreleri olan c1, c2, r1, r2, w ve N için değerler ata.
Her parçacık için P_{eniye} değerini ilk konumlar olarak ata.
En iyi P_{eniye} değerini G_{eniye} olarak ata.
Algoritmayı sonlandırmak için istenilen başka ölçüt varsa ata.
 $k = 1$ ile başla.

Adım 2: $k \leq N$ olana veya sonlandırma şartı sağlanıncaya kadar devam et.

Adım 3: Her parçacık için uygunluk değerlerini hesapla.
Her parçacık için yeni çıkan değerler eski değerlerden iyiyse bu değerleri P_{eniye} yap.
Tüm P_{eniye} değerleri arasındaki en iyi konumu G_{eniye} olarak belirle.

Adım 4: Her parçacık için Eş. 3'ü kullanarak yeni hızları belirle.
Her parçacık için Eş. 4'ü kullanarak yeni konumları belirle.
 $k = k + 1$
Adım 2'ye git.

4. YÖNTEM (METHOD)

Temel olarak, bahsedilen algoritmaya göre Altın Bölme Yöntemi ile hedef işlev kısıtlanır. Ardından Broydon-Fletcher-Goldfarb-Shanno yöntemi sayesinde hedef işlevin aralığı türevler ile kesilerek iyice daraltılır. Daha da derin inceleme olması amacıyla hedef işlevin kalan aralığı Parçacık Sürü Optimizasyonu yöntemi ile incelenir. Önerilen algoritma en uygun noktayı bulunca veya daha önceden belirlenmiş bir yakınsama oranına ulaşıncaya durdurulur. Anlatılan bu durum Tablo 5'teki algoritmada gösterilmiştir.

Tablo 5. Geliştirilen Algoritma (The Developed Algorithm)

Adım 1: Bir başlangıç değeri x_0 , durdurma ölçütü ϵ , döngü tur sayısı N belirle.
Optimize edilecek amaç fonksiyonunu seç.
 $x_k = \text{BFGS}(x_0)$
 $k = 1$

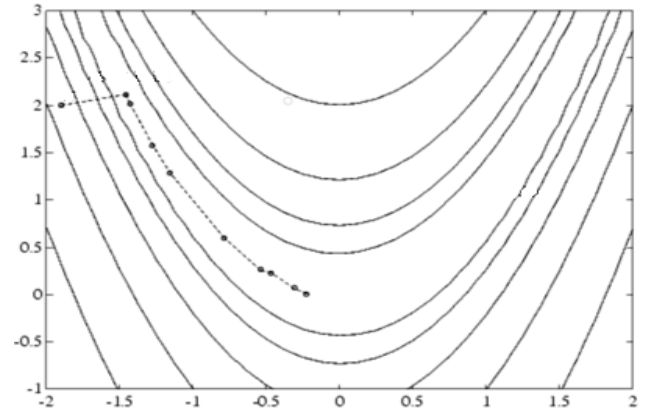
Adım 2: $\|\Delta x\| = \text{norm}(x_k - x_{k-1})$
Eğer $\|\Delta x\| < \epsilon$ ise, değişkenler değişmediği için dur.
Eğer $k + 1 = N$ ise, döngü tur sayısı bittiği için dur.

Adım 3: $x_k = \text{PSO}(x_k)$
 $x_k = \text{BFGS}(x_k)$
 $k = k + 1$
Adım 2'ye git.

Burada amaç işlevi olarak Bölüm 5.1'deki deneme işlevleri kullanılmıştır. x_0 değeri başlangıç noktası seçilmiş ve

deneme işlevinin araştırma bölgesi için üretilmiş rastgele bir noktadır. ϵ durdurma değeri olarak 10^{-3} tercih edilmiştir. Literatürde iki veya üç yineleme ile çözüm aranmakta iken çalışmamızda sonuç değişimlerini incelemek için N değerinin 20 olması tercih edilmiştir.

Örnek vermek gerekirse, amaç işlevi olarak küresel en küçük noktası 0 olan Rosenbrock işlevini ele alalım. Başlangıç noktası olarak (-1,9, 2) noktası seçelim. Tablo 5'teki Adım 1 gereği başlangıç noktamız öncelikle BFGS ile değerlendirilmelidir. Yaptığımız denemeler sonucunda BFGS uygun bir türevlenebilir noktaya doğru hareket etmektedir. İlk değerler yaklaşık (-1,48, 2,12) noktasından başlayıp azalan Şekil 1'deki gibi bir eğim göstermektedir. Sonuçlarımız küresel en küçük noktaya yaklaştıkça algoritmamız gereği hassas incelemeyi PSO devralmaktadır. Bu aşamada PSO bulunan en küçük değer küresel olup olmadığını incelemektedir. Eğer bulunan en küçük nokta küresel en küçük nokta değilse görevi tekrar BFGS'ye devretmekte ve amaç işlevinin başka bir salınımı içerisinde hareket etmektedir. Sonuç olarak önerilen yöntem yineleme değerini aştığı için (0,998, 0,999) noktasında durmuştur ve küresel en küçük noktayı yaklaşık olarak $1,9E-06$ değeri olarak bulmuştur.



Şekil 1. BFGS'nin Rosenbrock işlevi üzerindeki hareketi
(The movement of BFGS on the Rosenbrock Function)

Karşılaştırma yapabilmek amacıyla kullanılan yöntem, geliştirilen yöntemin tersi bir yaklaşım sergilemektedir. Karşılaştırma algoritması öncelikle PSO ile başlar ardından BFGS yöntemi çalıştırılır. Tablo 6'da karşılaştırma yöntemi görülmektedir.

Tablo 6. Karşılaştırma Algoritması (The Comparison Algorithm)

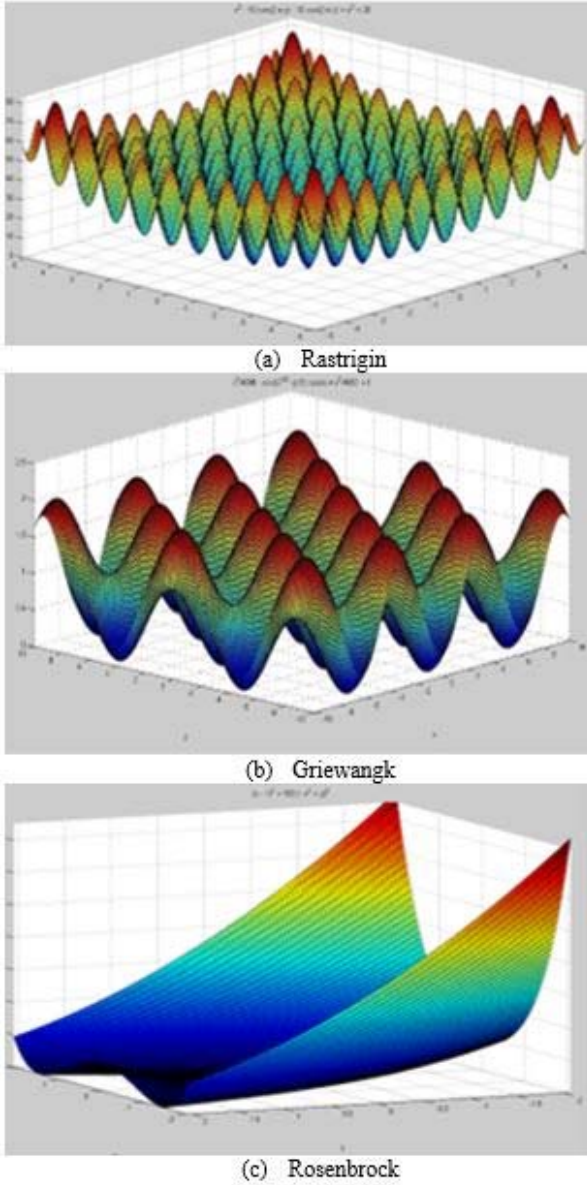
Adım 1: Bir başlangıç değeri x_0 , durdurma ölçütü ϵ , döngü tur sayısı N belirle.
Optimize edilecek amaç fonksiyonunu seç.
 $k = 1$

Adım 2: $k < N$ ve ϵ şartı sağlanmamış ise
 $x_k = \text{PSO}(x_k)$
 $x_k = \text{BFGS}(x_k)$
 $k = k + 1$
Adım 2'ye git.

Geliştirilen algoritma ve karşılaştırma algoritması arasındaki başarımları kıyaslamak için, ileride değinileceği gibi, literatürde sıklıkla tercih edilen sınaama işlevleri kullanılmıştır.

5. DENEY VE SONUÇLAR (EXPERIMENT AND RESULTS)

Daha önceden de değinildiği gibi geliştirilen yöntemin başarımlarını ölçmek için literatürde bulunan üç sınaama işlevi kullanılmıştır. Bunlar; Rastrigin, Griewangk ve Rosenbrock, işlevleri olup 3 bilinmeyenli durumlar üzerinde çalışılmıştır. Şekil 2’de kullanılan sınaama işlevlerinin 3 değişkenli gösterimleri verilmiştir. Her işlev kendi tanımlı olduğu sınırlarda rastgele başlangıç değerleri kullanılarak ellışer defa çalıştırılmıştır. Hassasiyet 10^{-15} değerine kadar küçültülmüş olup, bu değerden daha küçük değerler elde edildiğinde sonuç 0 olarak atanmıştır.



Şekil 2. Kullanılan deneme işlevleri (The test functions used)

5.1. Deneme İşlevleri (Test Functions)

Rastrigin işlevi (Şekil 2a) genellikle -5,12 ile +5,12 arasında incelenir ve küresel en küçük noktası sıfırdır. Griewangk işlevi (Şekil 2b) genellikle -600 ile +600 arasında incelenir ve küresel en küçük noktası sıfırdır. Şekil 2c’de görülebileceği gibi Rosenbrock işlevi, Rosenbrock vadisi olarak da geçer, -2,048 ile +2,048 arasında incelenmesi sonucunda küresel en küçük noktası sıfır olarak bulunur. Sırasıyla Eş 5, 6 ve 7 Rastrigin, Griewangk ve Rosenbrock fonksiyonlarının matematiksel ifadelerini vermektedir.

$$f(x) = 10n + \sum_{i=1}^n (x_i - \cos(2\pi x_i)) \quad (5)$$

$$f(x) = \frac{1}{4000} + \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos \frac{x_i}{\sqrt{i}} + 1 \quad (6)$$

$$f(x) = \sum_{i=1}^{n-1} (100(x_{i+1} + x_i^2)^2 + (1 - x_i)^2) \quad (7)$$

5.2. Deneysel Sonuçlar (Experimental Results)

Yöntemimizi denemek için kullanılan işlevlerin üç tanesinin başlangıç değerleri rastgele olarak alınmış, bu değerlere göre elde edilen sonuçlar tek tek hesaplanmış ve bunlar karşılaştırma için kullanılan algoritma ile karşılaştırılmıştır.

Tablo 7’de algoritmamıza girilmiş rastgele değerlere göre elde ettiğimiz sonuçların karşılaştırmaları görülmektedir. Tablo 7’de ÖA kısaltması “Önerilen Algoritma”, KA kısaltması ise “Karşılaştırma Algoritması” için kullanılmıştır. Aynı şekilde EKNO kısaltması “En Küçük Noktaların Ortalaması”, ÇS kısaltması “Çözüm Süresi”, SS “Standart Sapma”, EKN “En Küçük Nokta” ve EBN “En Büyük Nokta” demektir. Ayrıca tablodaki bütün değerler ellışer deneme sonucunda elde edilen değerlerin ortalamasıdır. Tabloda bulunan E ifadesi 10^7 ’un katı anlamına gelmektedir. Bu durumda E-11 değeri 10^{-11} olarak düşünülmelidir.

Rastrigin işlevi için sonuçlar değerlendirildiğinde, önerilen algoritmanın saniyenin binde biri farkla daha hızlı olduğu görülecektir. Karşılaştırma algoritması ortalama 2,09 saniyede beklenen sonuca yaklaşırken önerilen algoritma 2,08 saniyede bir ya da iki istisna hariç tam sonuca ulaşmaktadır. Bu sonuçların detayları Tablo 8 ve Tablo 9’da gösterilmektedir.

Rosenbrock işlevi için ise, önerilen algoritmanın ortalama değer olarak daha yavaş sonuç verdiği gözlemlenmiştir. Tablo 10 ve Tablo 11’den de görülebileceği gibi sadece iki durumda istenilen sonuçtan çok uzak noktalara ulaşmıştır. Dolayısıyla sonuçların standart sapmasında çok büyük bir fark yaratmaktadır. Eğer bahsedilen uç değerler değerlendirme dışı bırakılırsa Rosenbrock işlevi için EKNO değeri yaklaşık olarak $4,52E-06$ olmaktadır ve karşılaştırma algoritmasının sonuçlarından çok daha iyi sonuçlar elde etmiştir. Öte yandan iki algoritmanın sonuçları beklenen küresel en küçük nokta olan 0 değerine ulaşamamış ama önerilen algoritma kimi durumlarda yaklaşık on kat daha çok yaklaşmıştır. Ek olarak, yine Tablo 10 ve Tablo 11’den de

Tablo 7. Önerilen Algoritmanın Sonuç Karşılaştırmaları (The Results Comparison of Proposed Algorithm)

	Rosenbrock		Griewangk		Rastrigin	
	ÖA	KA	ÖA	KA	ÖA	KA
EKNO	1,87E+07	1,37E-01	1,83E-03	1,38E-03	2,0E-13	3,82E-10
ÇS (sn)	1,91	1,37	2,17	2,12	2,08	2,09
SS	9,17E+07	9,61E-01	3,60E-03	2,14E-03	1,38E-12	1,7E-09
EKN	8,93E-09	5,41E-08	4,07E-11	3,01E-05	0	2,77E-13
EBN	4,68E+08	6,86E+00	1,65E-02	1,33E-02	9,84E-12	9,14E-09

Tablo 8. Rastrigin İçin Üretilen Rastgele Başlangıç Değerleri (Random Initial Values Generated for Rastrigin)

#	x	y	#	x	y	#	x	y
1	0,414422	3,459031	18	4,946481	3,774889	35	-4,52091	-2,16013
2	5,025061	1,82763	19	-3,88392	-3,70267	36	-2,20695	1,444159
3	2,575229	2,250852	20	4,185291	-4,15246	37	-2,36549	4,405628
4	-3,07967	-3,32389	21	1,239004	-1,96155	38	-1,15323	-4,19119
5	-0,82314	-1,67503	22	3,610822	-3,22837	39	-2,32972	-4,91543
6	0,872693	-3,226	23	-2,23367	0,351977	40	-3,85543	-2,52805
7	3,415469	4,422788	24	0,419666	-1,63698	41	3,736424	1,302909
8	-2,53843	-0,22805	25	1,894312	2,555619	42	3,310572	0,613304
9	1,058851	-3,66965	26	2,750299	2,468887	43	-4,46854	-3,06095
10	-2,76263	-0,85854	27	-4,16062	0,486652	44	4,361015	2,087583
11	1,832154	-1,99289	28	3,026772	-2,7656	45	-0,76147	-0,49614
12	4,432218	3,462926	29	-1,50362	1,745546	46	-1,16034	-2,32348
13	-4,46576	0,191333	30	-2,25468	-2,48896	47	0,388895	-4,53985
14	-2,72854	-4,41434	31	5,099247	4,84939	48	1,713755	-0,52427
15	0,119462	1,434441	32	-1,33986	1,66405	49	0,293461	0,8076
16	-0,55698	1,190547	33	-1,35833	-0,51418	50	-4,49731	-3,974
17	-3,23252	0,961818	34	-4,38828	-4,44747			

görülebileceği gibi ortalama çözüm süreleri arasında saniyenin binde altısı kadar fark bulunmaktadır.

Tablo 12 ve Tablo 13'ten de görülebileceği gibi Griewangk işlevinde iki algoritma arasında binde beşlik bir çözüm hızı ortaya çıkmıştır ve karşılaştırma algoritması ortalama olarak daha hızlı çıkmıştır. Karşılaştırma algoritmasının ortalama çözüm süresi 2,12 saniye olurken, önerilen algoritma 2,17 saniyede çözüme ulaşmıştır. Ancak, önerilen algoritma yine sonuç için beklenen sıfır değerine daha yakın sonuçlar üretse de ortalama değerlere bakıldığında yaklaşık yüzde beşlik bir hata payı bulunmaktadır. Uygulama sonuçlarına göre,

Rosenbrock işlevi için bulunan sonuçlar Griewangk ve Rastrigin işlevlerine kıyasla daha kötüdür. Çünkü, önerilen algoritma ve karşılaştırma algoritması temelinde PSO ve BFGS yöntemlerini kullanmaktadır. PSO yönteminin yapısında var olan erken yakınsama ile yerel minimuma takılma problemleri ve BFGS yönteminin yapısında var olan yerel minimuma takılma problemi önerilen algoritmanın sonuçlarına da yansımaktadır. Bu önerilen algoritmanın zayıf yönü olarak görülebilir. Diğer taraftan ise, önerilen algoritmadaki melez yapıda bu problemlerin tek başlarına PSO ve BFGS yöntemleri ile kıyaslandığında azaldığı görülmektedir.

Tablo 9. Rastrigin İşlevi Çözümleri ve Çözüm Süreleri (Rastrigin Function Solutions and Resolution Times)

#	Önerilen Algoritma				Karşılaştırma Algoritması			
	\min_x	\min_y	\min_z	süre(sn)	\min_x	\min_y	\min_z	süre(sn)
1	1,82E-08	3,72E-08	0	2,04361	-2,14E-06	-2,46E-06	1,13E-11	2,10601
2	4,76E-08	-3,18E-08	3,55E-15	1,98121	8,58E-07	3,15E-06	1,13E-11	2,01241
3	5,02E-09	-4,33E-08	0	2,04361	3,36E-07	1,74E-06	3,32E-12	1,98121
4	3,02E-08	2,06E-08	0	1,99681	-4,30E-07	2,06E-06	4,70E-12	1,98121
5	-2,00E-08	-1,83E-09	0	1,98121	3,01E-06	2,48E-06	1,61E-11	1,99681
6	2,90E-08	-2,87E-08	0	1,98121	1,80E-06	2,14E-06	8,29E-12	1,98121
7	-2,35E-08	1,66E-08	0	1,98121	-1,83E-06	1,85E-06	7,21E-12	1,99681
8	-2,92E-08	3,60E-08	0	1,98121	-2,30E-06	-2,10E-06	1,03E-11	1,99681
9	-5,81E-10	-2,94E-09	0	1,99681	-3,71E-06	6,67E-07	1,50E-11	2,01241
10	4,25E-08	-2,57E-08	0	2,01241	-1,67E-06	-2,98E-06	1,23E-11	2,07481
11	2,94E-08	-5,15E-08	0	2,04361	-2,25E-06	2,40E-06	1,15E-11	2,05921
12	-1,79E-08	1,52E-08	0	2,02801	1,36E-06	1,24E-06	3,59E-12	2,07481
13	1,69E-06	2,53E-06	9,84E-12	2,05921	-2,79E-05	-8,10E-05	7,78E-09	2,04361
14	-3,20E-08	1,87E-09	0	2,05921	-2,31E-06	3,22E-07	5,78E-12	2,07481
15	-1,32E-08	2,72E-08	0	2,05921	1,01E-08	9,37E-07	9,31E-13	2,01241
16	-8,94E-09	3,85E-08	0	2,01241	-4,04E-06	3,46E-06	3,00E-11	1,95001
17	1,67E-08	7,77E-09	0	2,04361	1,68E-06	1,71E-06	6,10E-12	2,07481
18	-2,40E-08	-2,36E-08	0	2,04361	4,68E-07	-2,04E-07	2,77E-13	2,05921
19	-3,78E-09	-4,41E-10	0	2,04361	1,32E-06	-1,16E-06	3,27E-12	2,04361
20	1,91E-08	3,89E-08	0	2,05921	-7,76E-07	-3,24E-06	1,18E-11	2,02801
21	2,53E-08	4,05E-08	0	2,13721	6,29E-07	4,84E-06	2,53E-11	2,05921
22	3,50E-08	2,01E-08	0	2,04361	3,68E-06	-3,02E-06	2,41E-11	2,05921
23	-6,32E-09	4,30E-08	0	2,19961	3,11E-06	-1,46E-06	1,25E-11	2,10601
24	-3,94E-08	9,40E-09	0	2,04361	4,89E-07	1,93E-07	2,91E-13	2,05921
25	-2,56E-08	3,47E-08	0	2,26201	-2,04E-06	-7,91E-07	5,07E-12	2,09041
26	-1,98E-09	-3,94E-08	0	2,18401	1,93E-06	-1,83E-06	7,49E-12	2,30882
27	8,12E-09	1,41E-08	0	2,55842	-5,09E-06	-3,02E-06	3,72E-11	2,62082
28	-2,24E-08	5,08E-08	0	2,48042	1,49E-06	-6,08E-07	2,74E-12	2,26201
29	-2,83E-09	3,43E-08	0	2,12161	1,94E-06	5,95E-06	4,16E-11	2,13721
30	-6,89E-09	-4,11E-10	0	2,02801	-2,53E-06	-2,28E-06	1,23E-11	2,01241
31	-2,94E-08	1,51E-09	0	2,05921	-6,95E-06	2,07E-06	5,57E-11	2,04361
32	-9,52E-09	-3,51E-08	0	2,07481	3,67E-07	1,54E-06	2,66E-12	2,12161
33	1,59E-08	-8,15E-09	0	2,07481	2,79E-06	-1,03E-06	9,36E-12	2,12161
34	-1,95E-08	3,60E-08	0	2,15281	-8,81E-07	1,48E-05	2,32E-10	2,16841
35	-3,10E-08	3,00E-08	3,55E-15	2,13721	3,12E-07	-1,90E-06	3,94E-12	2,38682
36	-4,43E-08	2,19E-08	0	2,09041	3,22E-05	8,71E-05	9,14E-09	2,18401
37	3,36E-08	2,35E-08	0	2,16841	-1,99E-06	-1,97E-06	8,33E-12	2,24641
38	1,49E-08	-5,78E-09	0	2,16841	1,10E-06	-9,19E-07	2,18E-12	2,12161
39	1,38E-09	-4,95E-08	0	2,16841	1,58E-06	1,66E-06	5,54E-12	2,01241
40	-1,97E-09	1,53E-08	0	1,99681	2,62E-06	-2,07E-06	1,18E-11	2,01241
41	-2,91E-08	1,68E-09	0	1,99681	-1,36E-07	8,26E-07	7,43E-13	1,99681
42	-6,83E-09	-2,68E-08	0	1,98121	-1,06E-05	-3,38E-05	1,33E-09	2,02801
43	2,46E-08	1,57E-08	0	2,09041	1,03E-05	2,70E-06	1,20E-10	2,29322
44	-1,68E-08	-1,18E-08	0	2,35562	-3,25E-06	3,07E-06	2,12E-11	2,19961
45	-4,60E-08	3,07E-08	3,55E-15	2,10601	1,06E-06	-1,08E-06	2,44E-12	2,13721
46	-2,40E-08	-3,75E-08	0	1,99681	1,11E-06	-3,61E-06	1,51E-11	1,99681
47	-1,04E-08	3,40E-09	0	2,10601	1,32E-06	-1,21E-06	3,40E-12	2,01241
48	-1,74E-08	1,35E-09	0	2,04361	1,09E-06	3,19E-08	1,25E-12	2,09041
49	-1,96E-07	-2,62E-07	1,14E-13	2,12161	-3,13E-06	-6,56E-08	1,04E-11	2,02801
50	3,15E-10	-5,06E-08	0	2,09041	8,27E-07	1,07E-06	1,93E-12	2,04361

Tablo 10. Rosenbrock İçin Üretilen Rastgele Başlangıç Değerleri (Random Initial Values Generated for Rosenbrock)

#	x	y	#	x	y	#	x	y
1	-0,00179	0,085297	18	-0,60733	-0,04904	35	-0,43155	0,593216
2	0,59581	-0,99015	19	-1,14735	0,599034	36	-1,61385	-1,13585
3	1,417894	0,026898	20	-0,70345	-1,03533	37	-0,54721	-0,14763
4	-0,16514	-0,12944	21	0,058591	-0,81843	38	-1,27852	0,725169
5	1,854763	2,042388	22	0,937408	1,564599	39	-1,96313	0,717305
6	-1,98948	-1,44306	23	0,435206	-0,68226	40	0,989002	0,242149
7	-1,66467	-0,83965	24	0,088862	-0,08563	41	-1,23835	0,67414
8	-0,53662	-0,08858	25	1,959471	-1,68983	42	1,84865	-0,28472
9	-0,17289	-0,06279	26	-1,92387	1,914235	43	0,466336	0,529937
10	-0,99483	1,869926	27	-1,4455	1,21504	44	-0,43868	1,654327
11	1,063118	0,469915	28	1,780458	0,04984	45	-0,20364	1,289845
12	0,430013	0,44172	29	1,74325	1,5212	46	2,047172	-0,52786
13	1,930589	-1,18366	30	-1,91945	-0,95283	47	-1,11476	-2,00817
14	0,794797	1,840222	31	0,647534	0,494838	48	-1,89881	1,96506
15	1,573514	1,225775	32	-0,66762	1,811903	49	0,534151	0,051882
16	1,71697	-1,54948	33	1,348782	-1,52509	50	1,779922	-0,38935
17	1,111165	-1,14657	34	-0,59766	-1,11633			

Tablo 11. Rosenbrock İşlevi Çözümleri ve Çözüm Süreleri (Rosenbrock Function Solutions and Resolution Times)

Önerilen Algoritma				Karşılaştırma Algoritması				
#	min _x	min _y	min _z	süre(sn)	min _x	min _y	min _z	süre(sn)
1	0,998577	0,997385	7,26E-06	1,4976	0,998096	0,99619	3,63E-06	1,54441
2	0,999384	0,998772	3,80E-07	1,3884	1,00372	1,00742	1,38E-05	1,48201
3	-21631,8	4,68E+08	4,68E+08	2,8080	0,99949	0,998953	3,09E-07	1,40401
4	0,998522	0,997215	5,09E-06	1,3572	0,998983	0,997687	8,85E-06	1,40401
5	0,999978	0,999965	8,93E-09	1,3884	0,999013	0,99794	1,74E-06	1,37281
6	-2,16318	4,68E+08	4,68E+08	2,8080	1,00143	1,00277	2,79E-06	1,43521
7	0,999873	0,999772	8,63E-08	2,7300	0,999704	0,999329	7,04E-07	1,37281
8	0,998338	0,996557	4,25E-06	1,3884	0,998388	0,996762	2,62E-06	1,37281
9	1,000169	1,00031	1,12E-07	1,3884	1,00071	1,00129	1,96E-06	1,37281
10	0,99972	0,999462	1,24E-07	1,4040	0,99961	0,999174	3,71E-07	1,38841
11	0,999742	0,999521	2,11E-07	1,3884	0,997308	0,994693	7,73E-06	1,35721
12	1,000115	1,00033	9,80E-07	2,7144	0,998574	0,997033	3,41E-06	1,32601
13	0,999608	0,998981	5,63E-06	2,7456	1,00112	1,00213	2,44E-06	1,37281
14	1,001065	1,00217	1,26E-06	1,3416	0,99995	0,999859	1,73E-07	1,35721
15	0,997213	0,994803	2,13E-05	1,3884	0,998439	0,996717	5,11E-06	1,35721
16	1,003144	1,00640	1,09E-05	2,7144	0,999062	0,998161	1,01E-06	1,34161
17	1,000065	1,00042	8,30E-06	2,6832	0,99923	0,998267	4,29E-06	1,34161
18	1,001124	1,00235	2,37E-06	1,3728	1,00225	1,004289E	1,00E-05	1,51321
19	1,000087	1,00015	8,02E-08	1,3572	1,00047	1,00077	2,85E-06	1,40401
20	0,999457	0,999057	2,36E-06	2,7456	0,997744	0,995284	9,44E-06	1,35721
21	0,999472	0,999021	8,63E-07	1,3728	0,999632	0,999221	3,15E-07	1,37281
22	1,000829	1,00165	7,01E-07	1,3572	1,00479	1,00953	2,34E-05	1,35721
23	0,999584	0,999041	1,79E-06	1,3416	0,999849	0,999671	9,90E-08	1,32601
24	0,999755	0,999495	8,07E-08	1,3260	0,999863	0,999696	1,06E-07	1,34161
25	1,000594	1,00122	4,23E-07	2,6676	0,993881	0,987573	4,26E-05	1,32601
26	0,999658	0,999451	1,93E-06	2,6676	0,999355	0,99853	3,68E-06	1,34161
27	0,999892	0,998055	5,68E-06	1,3416	0,998537	0,99687	6,42E-06	1,34161
28	0,999127	0,998194	1,13E-06	2,6832	0,999733	0,999371	9,54E-07	1,35721
29	0,999319	0,998594	6,52E-07	1,3416	1,00092	1,00169	2,81E-06	1,34161
30	0,999788	0,999592	7,47E-08	2,6832	0,99635	0,99286	1,55E-05	1,40401
31	0,999411	0,998786	4,81E-07	1,4040	0,997907	0,995881	4,79E-06	1,35721
32	0,999638	0,999222	4,28E-07	1,3728	1,00292	1,00564	1,30E-05	1,35721

33	0,998661	0,997294	1,89E-06	2,7300	0,998693	0,99731	2,30E-06	1,35721
34	1,000,821	1,00216	2,71E-05	1,3416	0,996998	0,993562	2,86E-05	1,34161
35	0,999656	0,998929	1,48E-05	1,3572	1,00027	1,00049	2,85E-07	1,34161
36	0,99969	0,999546	2,84E-06	2,6520	0,999887	0,999699	5,79E-07	1,32601
37	1,001422	1,00327	2,00E-05	1,3260	0,999546	0,999028	6,06E-07	1,32601
38	1,000533	1,00044	3,94E-05	1,3416	0,998733	0,997324	3,64E-06	1,32601
39	1,000765	1,00143	1,57E-06	2,6364	0,997469	0,995015	6,90E-06	1,32601
40	0,999826	0,99967	6,51E-08	2,7300	0,998788	0,997445	3,20E-06	1,38841
41	1,001594	1,00303	5,29E-06	1,3572	0,999786	0,999562	5,41E-08	1,34161
42	0,999643	0,999286	1,27E-07	2,7300	1,00088	1,00161	2,81E-06	1,35721
43	1,000874	1,00157	4,07E-06	1,3572	1,00061	1,00121	3,77E-07	1,35721
44	0,999164	0,99826	1,16E-06	1,4352	0,999384	0,998747	4,30E-07	1,34161
45	0,99958	0,999243	8,71E-07	1,3884	0,998076	0,995965	7,36E-06	1,32601
46	0,998786	0,997748	4,48E-06	2,6676	0,999537	0,999029	4,28E-07	1,35721
47	1,000176	1,00035	3,55E-08	2,7456	0,999149	0,998208	1,54E-06	1,34161
48	1,000944	1,00173	3,38E-06	1,3728	-1,61940	2,62236	6,86125	1,35721
49	1,001194	1,00237	1,45E-06	1,3572	0,999784	0,999502	4,79E-07	1,35721
50	0,999357	0,998895	3,66E-06	2,6988	0,997397	0,994538	1,37E-05	1,35721

Tablo 12. Griewangk İçin Üretilen Rastgele Başlangıç Değerleri (Random Initial Values Generated for Griewangk)

#	x	y	#	x	y	#	x	y
1	449,2999	-465,85	18	133,4814	135,3945	35	153,6292	1,166554
2	-273,202	-262,665	19	345,1364	465,2657	36	175,7778	-20,4157
3	-161,556	244,1957	20	-560,065	-299,098	37	-39,4779	-520,74
4	487,4203	-254,169	21	323,9972	-194,642	38	84,53378	-238,934
5	-565,14	-523,865	22	-227,233	-421,423	39	88,39209	-454,628
6	-408,908	-236,474	23	-369,114	423,9601	40	-476,886	181,1761
7	-442,783	-275,128	24	-582,325	-201,258	41	-467,524	285,4149
8	-431,493	570,2428	25	-171,314	-412,015	42	266,6015	-562,617
9	-539,855	326,0999	26	-40,0249	356,8216	43	-214,422	438,758
10	-567,339	-364,259	27	-192,878	-469,793	44	557,0893	471,879
11	104,0779	401,6937	28	556,0343	-423,938	45	-42,4605	-226,624
12	25,95572	344,1872	29	-541,839	195,9196	46	15,38425	-207,258
13	-570,18	226,5861	30	-33,4538	-36,7465	47	240,9133	248,3451
14	-484,669	-234,193	31	487,2795	386,47	48	-576,991	230,4726
15	-486,559	337,1844	32	190,4178	-214,363	49	61,80231	317,1223
16	-545,726	581,9787	33	-498,97	94,3159	50	-71,3362	-182,867
17	196,0687	-85,0692	34	-279,462	301,3264			

Tablo 13. Griewangk İşlevi Çözümleri ve Çözüm Süreleri (Griewangk Function Solutions and Resolution Times)

#	Önerilen Algoritma				Karşılaştırma Algoritması			
	minx	miny	minz	süre(sn)	minx	miny	minz	süre(sn)
1	-0,73663	-1,56795	0,00102	2,152814	0,660088	-0,99927	0,000501	2,074813
2	0,019827	0,653612	0,000139	2,121614	-0,15687	1,111157	0,000413	2,074813
3	0,053626	-1,01807	0,000339	2,106013	1,91021	2,507635	0,003519	2,074813
4	0,049614	0,709857	0,000165	2,106014	-0,97064	-1,380690	0,001001	2,074813
5	0,003173	-0,00524	1,30E-08	2,121614	0,030076	0,527276	9,10E-05	2,059213
6	-0,00859	0,05986	1,20E-06	2,106014	0,428412	0,087749	7,63E-05	2,090413
7	1,13397	-0,01691	0,000517	2,184014	-0,11256	1,001839	0,000332	2,464816
8	0,003173	-0,00524	1,30E-08	2,308815	-0,45285	-0,16014	9,09E-05	2,184014
9	-0,02556	0,411365	5,55E-05	2,246414	-0,07695	-0,82829	0,000226	2,106013
10	0,003173	-0,00524	1,30E-08	2,558416	0,893924	-1,507090	0,001062	2,215214
11	-4,79264	-4,41849	0,015601	2,168414	-0,09496	-1,189550	0,000465	2,152814
12	1,91451	2,13259	0,002958	2,355615	-1,07650	-0,59132	0,00058	2,184014
13	0,688786	1,42115	0,00085	2,215214	-1,64508	-3,517110	0,005123	2,168414

14	-2,11925	-2,08955	0,003231	2,277615	-1,32639	-1,233080	0,001204	2,246414
15	1,13589	-2,57100	0,002179	2,293215	0,635479	2,498883	0,002199	2,152814
16	0,003173	-0,00524	1,30E-08	2,230814	0,23518	1,135600	0,000443	2,137214
17	-0,70831	0,064461	0,000203	2,168414	-0,03287	-0,44682	6,55E-05	2,184014
18	1,78706	2,00258	0,002593	2,168414	-0,76382	-0,79409	0,00044	2,152814
19	2,09206	-2,40250	0,003643	2,152814	0,545872	-0,23123	0,000137	2,074813
20	4,09071	4,35560	0,012915	2,137214	-0,26842	-0,05741	3,01E-05	2,059213
21	0,69377	-1,51146	0,000939	2,106013	0,026263	0,729092	0,000174	2,059213
22	0,565886	1,07913	0,000509	2,106013	0,017285	0,310215	3,15E-05	2,074813
23	0,616766	1,78310	0,00119	2,106014	0,11883	-1,588050	0,000828	2,059213
24	0,01158	0,01082	9,21E-08	2,106014	-0,35555	-3,874200	0,004946	2,059213
25	-0,04218	-1,38196	0,000624	2,090413	0,046345	-0,56573	0,000105	2,106014
26	1,50162	-2,08425	0,002324	2,152814	0,361705	1,627714	0,000917	2,121614
27	-3,55948	-2,63746	0,007364	2,152814	1,04935	3,404283	0,004223	2,121614
28	0,003173	-0,00524	1,30E-08	2,137214	-0,84039	-2,251950	0,001938	2,106014
29	-0,28672	2,55476	0,002162	2,137214	-0,3086	-1,351610	0,000634	2,074813
30	0,000318	-9,53E-06	4,07E-11	2,106014	-0,44433	-2,364570	0,001903	2,074813
31	-4,52181	-5,02798	0,016463	2,168414	0,171286	-1,985220	0,001297	2,121614
32	-0,04365	-0,45503	6,83E-05	2,137214	0,666881	-0,19356	0,000191	2,106014
33	1,16889	-3,29292	0,004086	2,121614	-0,03301	-0,48687	7,78E-05	2,152814
34	0,013783	-0,26893	2,37E-05	2,137214	-0,9265	-3,625260	0,004632	2,137214
35	-0,17694	-0,83322	0,000239	2,215214	-2,69681	-5,632150	0,013268	2,137214
36	-0,66568	-2,76219	0,002667	2,168414	-2,00879	-0,60817	0,001744	2,121614
37	0,003743	-0,2103	1,44E-05	2,152814	-0,90405	2,030905	0,001674	2,074813
38	0,612748	-0,53891	0,000246	2,106014	-0,83028	-3,186210	0,003588	2,090413
39	0,050518	-0,08949	3,64E-06	2,215214	-0,64548	-1,967350	0,00143	2,090413
40	-0,01007	-0,2849	2,65E-05	2,152814	0,073727	0,855332	0,000241	2,059213
41	-0,87424	0,957788	0,000607	2,168414	0,021076	-0,53789	9,45E-05	2,090413
42	0,003173	-0,00524	1,30E-08	2,137214	-0,09182	-1,352370	0,0006	2,106014
43	0,081991	1,31373	0,000566	2,168414	-0,71814	1,200092	0,000677	2,106013
44	0,003173	-0,00524	1,30E-08	2,152814	0,15086	-1,074660	0,000386	2,090413
45	-0,23425	-1,29571	0,00057	2,168414	0,08589	-1,241480	0,000506	2,137214
46	-0,29087	0,630026	0,000163	2,121614	0,57668	2,158156	0,001653	2,106014
47	-0,00322	-0,02439	1,98E-07	2,137214	0,431864	-1,359020	0,000677	2,090413
48	-2,35504	-0,02308	0,002231	2,121614	2,15365	0,660307	0,002008	2,090413
49	0,522032	0,454551	0,000177	2,090413	0,327741	0,06803	4,47E-05	2,090413
50	0,392902	-2,28113	0,001759	2,090413	-0,13172	-0,92741	0,000287	2,074813

6. SONUÇLAR (CONCLUSIONS)

Başarım (performans) karşılaştırmalarından elde edilen verilere göre, karşılaştırma algoritması milisaniye düzeyinde daha hızlı olmaktadır. Hız farkları işlevin karmaşıklığına bağlı olarak değişmektedir. Örneğin; Rastrigin gibi daha basit yapıları test işlevlerinde milisaniye düzeyinde önerilen algoritma daha hızlıyken, Griewangk gibi nispeten daha zorlu işlevlerde karşılaştırma algoritmasının milisaniye düzeyinde daha hızlı olduğu tespit edilmiştir.

Önerilen algoritma başarım açısından eşdeğerlerine yakın sonuçlar vermiş olup, çözüme ulaşma konusunda daha başarılı olduğunu ispatlamıştır. Öte yandan amaç işlevi karmaşıklıkça önerilen algoritmanın yavaşladığı da gözlemlenmiştir. Ancak, eşdeğer yöntemlerle karşılaştırıldığında nispeten daha hızlı olduğu görülmektedir. Çalışmamızda BFGS gibi türev bilgisinden yararlanan bir yöntem kullanıldığı için, önerilen melez

algoritmanın çözeceği problemler doğrudan veya dolaylı olarak oluşturulabilen bir amaç işlevine sahip olmalıdır. Bu yüzden Gezgin Satıcı ve Araç Rotalama gibi problemlerin çözümüne de uygulanabilir. Herhangi bir sezgisel yöntem kullanılarak bütün en iyileme problemlerinde en başarılı sonuçları bulmak mümkün değildir. Öte yandan yapılan literatür araştırması ve önerilen algoritma melez algoritmaların standart sezgisellere ve numerik yöntemlere göre daha iyi sonuçlar elde ettiğini ispatlamıştır [32, 36]. Bu yüzden gelecek çalışmalarda, farklı sayısal ve sezgisel yöntemlerin kombinasyonları farklı işlev çözümlerinde çalıştırılmalıdır. Böylece hangi melez algoritmanın hangi işlevlerde daha başarılı sonuçlar vereceği tespit edilebilir. Bu tür gelecek çalışmalar yapıldığında melez algoritmaların eksiklikleri de daha net görülecektir. Çünkü sayısal yöntemlerde yerel arama kabiliyeti iyi iken küresel arama kabiliyeti ise zayıftır. Sezgisel yöntemlerde ise rasgele değerlerle çalışmanın getirdiği bazı sorunlar ile yerel en iyi değere takılma gibi sorunlar bulunmaktadır.

KAYNAKLAR (REFERENCES)

1. Karaboğa D., Yapay Zeka Optimizasyon Algoritmaları, Nobel Yayın Dağıtım, Ankara, Türkiye, 2011.
2. J. Nocedal and S. Wright, Numerical Optimization, Springer-Verlag, New York, A. B. D., 1999.
3. Wikipedia, Broyden–Fletcher–Goldfarb–Shanno algorithm. https://en.wikipedia.org/wiki/Broyden–Fletcher–Goldfarb–Shanno_algorithm. Yayın tarihi Ekim 3, 2016. Erişim tarihi Ağustos 25, 2020.
4. Kennedy J. ve Eberhart R., Particle swarm optimization, International Conference on Neural Networks (ICNN'95), Perth-Avustralya, 1942–1948, 27 Kasım - 1 Aralık, 1995.
5. Fletcher R., A new approach to variable metric algorithms, Oxford Journals, 13 (3), 317–322, 1970.
6. Lowrie M. ve Wah B., Learning Heuristic Function for Numeric Optimization Problems, COMPSAC 88: The Twelfth Annual International Computer Software & Applications Conference, Chicago, A. B. D., 443–450, 1988.
7. Davidon W. C., Variable metric method for minimization, SIAM J. Optimization, 1 (1), 1–17, 1991.
8. Semeter J. ve Mendillo M., Nonlinear optimization technique for ground-based atmospheric emission tomography, IEEE Transactions on Geoscience and Remote Sensing, 35 (5), 1105–1116, 1997.
9. Li D. H. ve Fukushima M., A modified BFGS method and its global convergence in nonconvex minimization, Journal of Computational and Applied Mathematics, 129, 15–35, 2001.
10. Morales J. L., A numerical study of limited memory BFGS methods, Applied Mathematics Letters, 15 (4), 481–487, 2002.
11. Business Standford Software Inc., SNOPT. http://www.sbsi-sol-optimize.com/asp/sol_product_snopt.htm. Yayın tarihi Eylül 5, 2016. Erişim tarihi Ağustos 25, 2020.
12. Molga M. ve Smutnicki C., Test functions for optimization needs, 1–43, 2005.
13. Victoire T. A. A ve Jeyakumar A. E., Deterministically guided PSO for dynamic dispatch considering valve-point effect, Electric Power Systems Research, 73, (3), 313–322, 2005.
14. Yan X., Ka Wing C. ve Mingbo L., Improved BFGS method for optimal power flow calculation with transient stability constraints, 2019 IEEE Power & Energy Society General Meeting, Atlanta, A. B. D., 434–439, 4-8 Ağustos, 2005.
15. Nawi N. M., Ransing M. R. ve Ransing R. S., An improved learning algorithm based on the Broyden-Fletcher-GoldfarbShanno (BFGS) method for back propagation neural networks, ISDA '06: Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications, 1, 152–157, 2006.
16. Andrei N., A scaled BFGS preconditioned conjugate gradient algorithm for unconstrained optimization, Applied Mathematics Letters, 20 (6), 645–650, 2007.
17. Perez R. E. ve Behdinan K., Particle swarm approach for structural design optimization, Computers & Structures, 85 (19–20), 1579–1588, 2007.
18. Xiao Y., Wei Z. ve Wang Z., A limited memory BFGS-type method for large-scale unconstrained optimization, Computers & Mathematics with Applications, 56 (4), 1001–1009, 2008.
19. Yuan G. ve Lu X., A new backtracking inexact BFGS method for symmetric nonlinear equations, Computers & Mathematics with Applications, 55 (1), 116–129, 2008.
20. Zhao S., Liang J. J., Suganthan P. N. ve Tasgetiren M. F., Dynamic multi-swarm particle swarm optimizer with local search for Large Scale Global Optimization, 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence), Hong Kong, Çin, 3845–3852, 1-6 Temmuz, 2008.
21. Özsağlam Y. ve Çunkaş M., Optimizasyon Problemlerinin Çözümü için Parçacık Sürü Optimizasyonu Algoritması, Politeknik Dergisi, 11 (4), 299–305, 2008.
22. Thomas N. ve Reed M., A hybrid algorithm for continuous optimisation, 2009 IEEE Congress on Evolutionary Computation (IEEE CEC 2009), Trondheim, Norveç, 1–6, 18-21 Mayıs, 2009.
23. Li S. ve Tan M., Tuning SVM parameters by using a hybrid CLPSO-BFGS algorithm, Neurocomputing, 73 (10–12), 2089–2096, 2010.
24. Li S., Tan M., Tsang I. W. ve Kwok J. T. Y., A hybrid PSO-BFGS strategy for global optimization of multimodal functions, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 41 (4), 1003–1014, 2011.
25. Nezhad A. M., Shandiz R. A. ve Jahromi A. E., A particle swarm-BFGS algorithm for nonlinear programming problems, Computers & Operations Research, 40 (4), 963–972, 2013.
26. Shi Y., Liu H., Gao L. ve Zhang G., Cellular particle swarm optimization, Information Sciences, 181 (20), 4460–4493, 2011.
27. Uriarte A., Melin P. ve Valdez F., A New Hybrid PSO Method Applied to Benchmark Functions, Nature-Inspired Design of Hybrid Intelligent Systems, Melin P., Castillo O., Kacprzyk J., Springer, Berlin-Almanya, 423–430, 2017.
28. Shatnawi M., Nasrudin M. F. ve Sahran S., A new initialization technique in polar coordinates for particle swarm optimization and polar PSO, International Journal on Advanced Science, Engineering and Information Technology, 7 (1), 242–249, 2017.
29. Özcan H., Comparison Of Particle Swarm And Differential Evolution Optimization Algorithms Considering Various Benchmark Functions, Politeknik Dergisi, 20 (4), 899–905, 2017.
30. Mehmood Y., Sadiq M., Shahzad W. ve Amin F., Fitness-based acceleration coefficients to enhance the convergence speed of novel binary particle swarm optimization, 2018 International Conference on

- Frontiers of Information Technology (FIT), Islamabad, Pakistan, 355–360, 17-19 Aralık, 2018.
31. Cavalca D. L. ve Fernandes R. A. S., Gradient-based Mechanism for PSO Algorithm: A Comparative Study on Numerical Benchmarks, 2018 IEEE Congress on Evolutionary Computation (CEC), Rio de Janeiro, Brezilya, 1-7, 8-13 Temmuz, 2018.
 32. Badem H., Basturk A., Caliskan A. and Yuksel M. E., A new hybrid optimization method combining artificial bee colony and limited-memory BFGS algorithms for efficient numerical optimization, Applied Soft Computing, 70, 826–844, 2018.
 33. Çınaroğlu S. ve Bulut H., New initialization approaches for the k-means and particle swarm optimization based clustering algorithms, Journal of the Faculty of Engineering and Architecture of Gazi University, 33 (2), 425–440, 2018.
 34. Geraldine Bessie Amali D. ve Dinakaran M., A new Quantum Tunneling Particle Swarm Optimization algorithm for training feedforward neural networks, International Journal of Intelligent Systems and Applications 10 (11), 64-75, 2018.
 35. Çimen M. E. ve Boz A. F., Parameter identification of a non-minimum phase second order system with time delay using relay test and PSO, CS, FA algorithms, Journal of the Faculty of Engineering and Architecture of Gazi University, 34 (1), 461–477, 2019.
 36. Jarrah M., Mohamad Jaya A., Alqattan Z., Azam M. A., Abdullah R., Jarrah H. ve Abu-Khadrah A., A novel explanatory hybrid artificial bee colony algorithm for numerical function optimization, The Journal of Supercomputing, 2020.
 37. Wikipedia, Altın Oran. https://tr.wikipedia.org/wiki/Altın_oran. Yayın tarihi Eylül 9, 2016. Erişim tarihi Ağustos 25, 2020.
 38. İplikçi S., “Lineer Olmayan Programlama,” 2010.