



Düzce Üniversitesi Bilim ve Teknoloji Dergisi

Araştırma Makalesi

Hibrit Çok-Amaçlı Rüzgar Güdümlü Optimizasyon Algoritması

 Fethiye Sultan ÖZPEHLİVAN AY^{a,*},  Pakize ERDOĞMUŞ^b

^a Bilgisayar Mühendisliği A.B.D., Fen Bilimleri Enstitüsü, Düzce Üniversitesi, Düzce, TÜRKİYE

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

* Sorumlu yazarın e-posta adresi: ozpehlivanfs@gmail.com

DOI: 10.29130/dubited.765012

ÖZET

Temel anlamda optimizasyon, bir veya birden fazla problemin belirli koşullar altındaki en iyi çözümlerini bulma işlemidir. Günümüzde bu problemlerin çözümü için klasik yöntemler ve sezgisel yöntemler kullanılmaktadır. Sezgisel yöntemlerden biri olan Rüzgar Güdümlü Optimizasyon algoritması, rüzgarın atmosfer içerisindeki hareketini temel alarak atmosferik dinamik eşitlikten yararlanan tek amaçlı optimizasyon problemlerine çözüm arayan bir algoritmadır.

Bu çalışmada çok-amaçlı optimizasyon problemlerinin çözümü için Rüzgar Güdümlü Optimizasyon algoritması yeniden düzenlenmiştir. Çok-amaçlı optimizasyon problemlerinde elde edilen sonuçların gerçek sonuçlara ne kadar yakınsadığı ve bu sonuçların ne kadar çeşitli olduğu kullanılan yöntemlerin performansı hakkında bilgi vermektedir. Baskın olmayan sıralama, ağırlıklı toplam, normal sınır kesişimi gibi metotlar çok-amaçlı optimizasyon problemlerinde sıklıkla kullanılan yaklaşımlardır. Bu yaklaşımlardan bazıları çeşitlilik açısından ön plana çıkarken bazılarının ise en iyi sonuca daha iyi yakınsadığı gözlenmiştir. Bu çalışmanın temel amacı elde edilen çözümleri hem çeşitlilik hem de yakınsama açısından en iyi hale getirmektir.

Bu amaç kapsamında baskın olmayan sıralama ve adaptif ızgara yaklaşımları bir arada kullanılarak yeni bir hibrit yaklaşım geliştirilmiştir. Daha iyi bir yakınsama için baskın olmayan sıralama, çeşitlilik için adaptif ızgara yaklaşımı bir arada kullanılmıştır. Geliştirilen bu hibrit yaklaşım test problemleri ve doğrusal olmayan denklem sistemlerinde test edilerek sonuçları literatürde iyi bilinen Baskın Olmayan Sıralamalı Genetik Algoritma (NSGA-II) ve Çok-Amaçlı Parçacık Sürü Optimizasyonu (MOPSO) algoritmaları ile karşılaştırılmıştır. Deneysel sonuçlar incelendiğinde çeşitlilik ve yakınsama performansı açısından geliştirilen hibrit yaklaşımın kabul edilebilir olduğu gözlenmiştir.

Anahtar Kelimeler: Çok-amaçlı Optimizasyon, Optimizasyon, Rüzgar Güdümlü Optimizasyon Algoritması

A Hybrid Multi-Objective Wind Driven Optimization Algorithm

ABSTRACT

Basically, optimization is the process of finding the best solutions for one or more problems under certain conditions. Today, classical methods and heuristic methods are used to solve these problems. Wind Driven Optimization algorithm, which is one of the heuristic methods, is an algorithm that seeks solutions for single-objective optimization problems that benefit from atmospheric dynamic equality based on the movement of the wind in the atmosphere.

In this study, Wind Driven Optimization algorithm was rearranged to solve multi-objective optimization problems. In multi-objective optimization problems, the performance of the used method depends on how closely the results obtained converge to the actual results and how diverse these results are. Methods such as non-dominant sorting,

weighted sum, normal boundary intersection are frequently used approaches in multi-objective optimization problems. While some of these approaches have more diverse to results, others have been observed to better converge. The main purpose of this study is to optimize the solutions obtained in terms of both diversity and convergence.

Within this scope, a new hybrid approach has been developed by using non-dominant sorting and adaptive grid approaches. Non-dominant sorting used for better convergence and adaptive grid approach is used for diversity. The developed hybrid approach has been tested in test problems and nonlinear equation systems. Results have been compared with Non-Dominated Sorting Genetic Algorithm (NSGA-II) and Multi-Objective Particle Swarm Optimization (MOPSO). When the experimental results were examined, it was observed that the hybrid approach developed in terms of diversity and convergence performance was acceptable.

Keywords: Multi-objective Optimization, Optimization, Wind Driven Optimization Algorithm

I. GİRİŞ

Teknolojinin gelişmesi ve hesaplama maliyetinin kolaylaşması ile birlikte optimizasyon problemlerinin çözümünde kullanılan klasik yöntemler yerini sezgisel araştırma yöntemlerine bırakmıştır. Ekosistemde yer alan canlıların davranışlarında optimum çözüme ulaşması bilim adamlarının araştırmalarına konu olmakta ve doğadan esinlenen algoritmaların artışı ile sonuçlanmaktadır [1]. Bu nedenle problemlerin çözümünde doğadan esinlenmek geçerlilik kazanmıştır. 1977 yılında optimizasyon alanında Holland tarafından devrim niteliğinde bir fikir ortaya atılmış ve doğadaki evrimselliğin bilgisayarlar aracılığıyla benzetimi yapılarak optimizasyon problemlerinin çözümü sağlanmıştır. Sezgisel algoritmaların en bilineni olan bu yöntem Genetik Algoritma (GA) olarak adlandırılmış ve karmaşık problemlerin çözümüne yeni bir bakış açısı getiren bir çalışma alanı haline gelmiştir [2].

Doğadan esinlenme ve evrimsellik kavramlarıyla birlikte GA'nın yanında birçok optimizasyon algoritması geliştirilmiştir. Parçacık Sürü Optimizasyonu (PSO) [3], Karınca Kolonisi Optimizasyonu (ACO) [4] gibi birçok algoritma tek-amaçlı optimizasyon problemlerini başarı ile çözebilmektedir. Bu algoritmaların düzenlenmiş versiyonları çok-amaçlı optimizasyon problemlerinin çözümünde kullanılmaktadır.

Çok-amaçlı optimizasyon birden fazla amacın en iyilenmesini amaçlar. Ancak bu amaçlar çoğu zaman birbirleri ile çelişirler. Bir fabrikada hem karın maksimize edilmesi, hem de maliyetin minimize edilmesi bir noktada denge durumuna ulaşır. Çok amaçlı optimizasyon problemlerinin çözümünde çeşitlilik ve yakınsama önemli iki kavramdır [5]. Bu amaçla çok amaçlı optimizasyonu için yeniden düzenlenmiş algoritmaların, problem çözümünde elde ettikleri pareto noktaların optimum çözüme mümkün olduğu kadar yakınsamış ve çeşitlilik göstermiş olması gerekmektedir. Yakınsama ve çeşitliliği sağlamak için baskın olmayan sıralama (non-dominated sorting), adaptif ızgara gibi birçok metot geliştirilmiştir [6] [7].

Genel olarak çok-amaçlı optimizasyon algoritmalarında uygunluk değerleri belirlenirken kümeleme ve pareto baskınlığı [8] tabanlı iki yaklaşım kullanılmaktadır [9]. Goldenberg'in baskınlık kavramını ortaya atmasından sonra birçok araştırmacı aynı yöntemi kullanarak çok-amaçlı optimizasyon algoritması geliştirmiştir. Fonseca ve Fleming, Çok-Amaçlı Genetik Algoritmalar (MOGAs) için sıralama tabanlı uygunluk atama metodu geliştirmişlerdir [10]. Horn ve ark. çok-amaçlı problemlerin optimize edilmesinde pareto baskınlığından yararlanmış ve GA'yı yeniden düzenleyerek Yerleştirilmiş Pareto Genetik Algoritma (NPGA) adlı yeni bir yöntem sunmuştur [11]. Srinvas ve Deb., daha önce bahsedilen yaklaşımlardan farklı olarak pareto baskınlığı ve sıralama tabanlı yaklaşımları bir arada kullanarak Baskın Olmayan Sıralamalı Genetik Algoritma (NSGA) adlı bir yöntem geliştirmiştir [6]. Geliştirilen bu yöntem daha çok Goldenberg'in yaklaşımına benzerlik göstermektedir. İlerleyen yıllarda Deb, baskınlık tabanlı yaklaşımlarda çok tartışılan hesaplama karmaşıklığını azaltarak NSGA II algoritmasını öne sürmüştür [12]. Mirjalili ve ark. doğadaki gri kurtların liderlik ve avlanma mekanizmasını taklit eden sezgisel bir yöntem olan Gri Kurt Algoritmasını [13] öne sürmüşler ve bu yöntemi geliştirerek çok-amaçlı problemlerin çözümüne uygun hale getirmişlerdir [14]. Aynı şekilde Saremi ve ark. çekirge

sürülerinin doğadaki davranışlarını taklit ederek çok değişkenli optimizasyon problemlerini çözen popülasyon tabanlı Çekirge Optimizasyon Algoritmasını [15] geliştirilmiştir. 2018 yılında Mirjalili ve ark. [16] bu algoritmayı yeniden düzenleyerek çok-amaçlı problemleri çözebilir hale getirmiştir. Zeng ve ark. [17], Reddy ve Kumar [18], Al Jadaan ve ark. [19], Ghiasi ve ark. [20], Nobahari ve ark. [21], Costa ve ark. [9], Salcedo-Sanz ve ark. [22] tarafından önerilen çok-amaçlı optimizasyon yaklaşımlarının hepsi aynı yöntemi kullanarak pareto cephesini daha yakınsamış ve çeşitli hale getirmeye çalışmışlardır.

Yapılan bu çalışmada popülasyon tabanlı Rüzgar GÜdümlü Optimizasyon (RGO) algoritmasının çok-amaçlı optimizasyon problemlerinin çözümünde kullanılabilmesini amaçlayan bir uyarlama içermektedir. Literatürde yer alan çok-amaçlı RGO, pareto-optimum noktaların elde edilmesinde hızlı bastırılmayan sıralama yöntemini kullanmaktadır [23]. Bu çalışmada ise Bayraktar ve arkadaşlarının geliştirmiş olduğu çok-amaçlı RGO'ya alternatif olarak, düzenlenmiş hızlı bastırılmayan sıralama ve adaptif ızgara yaklaşımı bir arada kullanılarak çözüme daha hızlı ve başarılı yakınsayacak hale getirilmiştir. Geliştirilen Hibrit Çok-Amaçlı Rüzgar GÜdümlü Optimizasyon Algoritması (HÇA-RGO) ile pareto-mesafe tabanlı bir yaklaşım geliştirilerek daha çeşitli çözümler elde edilmiştir.

Makalenin ikinci bölümünde tek amaçlı RGO tanıtılmaktadır. Üçüncü bölümde HÇA-RGO algoritması detaylı bir şekilde anlatılmaktadır. Son bölümde, elde edilen bulgular neticesinde diğer algoritmalar ile karşılaştırmalar yapılmıştır. Elde edilen sonuçlar geliştirilen yöntemin çok-amaçlı optimizasyon problemlerinde çeşitlilik ve yakınsama açısından başarılı olduğunu göstermektedir.

II. RÜZGAR GÜDÜMLÜ OPTİMİZASYON ALGORİTMASI

Bayraktar ve ark. tarafından 2010 yılında geliştirilen RGO, doğadan esinlenen popülasyon tabanlı bir global optimizasyon algoritmasıdır [24]. RGO, hidrostatik denge içerisinde yer alan, atmosferik dinamik eşitlikten türetilmiş ve rüzgarın atmosfer içerisindeki hareketinden etkilenmiştir. Rüzgarı oluşturan en küçük parçacığa hava parseli denilmektedir. Bu hava parselinin anlık konum ve hız bilgisi Newton'un ikinci hareket kanunu temel alınarak güncellenmektedir. Bir hava parseline uygulanan kuvvetlerin toplamı hava parselinin o anki hızı ve bir sonraki konumunu ifade etmektedir. Konum ve hız güncelleme eşitliği şu şekildedir:

$$u_n = (1 - \alpha)u_c - gx_c + \left| \frac{i-1}{i} \right| RT(x_{max} - x_c) + \frac{cu_c^{otherd}}{i} \quad (1)$$

Yukarıdaki eşitlikte u_n , bir sonraki iterasyona aktarılacak hız bilgisi, u_c ise a parselin anlık hızını ifade etmektedir. Hava parselinin anlık konum bilgisi x_c , o ana kadar bulunduğu konumlardan en iyisi x_{max} olarak gösterilmektedir. i değişkeni hava parselinin popülasyondaki sırasını ifade etmektedir. Hava parselleri kendilerine uygulanan basınç değerlerine göre sıralanmaktadır. Minimizasyon işlemi alçak basınç iyi, yüksek basınç kötü sonuç olarak kabul edilmektedir. u_c^{otherd} anlık hız değerini etkileyen bir başka boyuttaki hız değeridir. Klasik RGO kullanıcı tanımlamalı birkaç katsayı ile ilişkilidir. Yine Bayraktar ve ark. tarafından 2015 yılında geliştirilen Uyarlanı Rüzgar GÜdümlü Optimizasyon (AWDO) algoritması kullanılan bu katsayıları kullanıcılardan bağımsız dinamik bir şekilde güncelleyebilmektedir [25]. Eş. 1'de sürtünme katsayısı α , yerçekimi sabiti g , korolis kuvveti c , evrensel gaz sabiti R ve sıcaklık T ile gösterilmiştir. Algoritma içerisinde evrensel gaz sabiti ve sıcaklık tek bir katsayıda RT birleştirilerek kullanılmaktadır. Eşitlik 1'de yer alan katsayılar ve değişkenler ile elde edilen yeni hız değeri hava parselinin bir sonraki pozisyonunu belirlemekte kullanılır. Hava parselinin bir sonraki pozisyonu Eş. 2'de gösterilmiştir.

$$x_n = x_c + u_n \Delta t \quad (2)$$

Eş. 2’de parselin yeni konumu x_n , mevcut konumu x_c , güncellenmiş hız vektörü u_n ve zamana bağlı değişim katsayısı Δt ile ifade edilmektedir. Her iterasyonda hız ve konum bilgileri güncellenerek hava parselleri mevcut olan en iyi konuma, alçak basınca doğru hareket eder.

RGO uygulanış biçimi olarak PSO’ya benzerlik göstermektedir. PSO’da olduğu gibi her iterasyonda parsellerin konum bilgisi güncellenir ve en iyi konuma doğru bir yakınsama söz konusudur. RGO parçacık tabanlı diğer algoritmalarla karşılaştırıldığında hızlı yakınsaması ile ön plana çıkmaktadır.

Yapılan bu çalışma hızlı yakınsaması ile bilinen RGO algoritmasının çok-amaçlı problemlerin çözümünde etkin bir şekilde kullanımını hedeflemektedir. Bastırılmayan hızlı sıralama ve adaptif ızgara yaklaşımlarının birlikte kullanılmasıyla elde edilen yöntem diğer algoritmaların çok-amaçlı problemler için uyarlanabilmesini kolaylaştırmaktadır.

III. HİBRİT ÇOK AMAÇLI RÜZGAR GÜDÜMLÜ OPTİMİZASYON ALGORİTMASI

Tek-amaçlı problemlerin çözümünde başarılı sonuçlar veren RGO, bu bölümde çok-amaçlı problemleri optimize edebilmesi için uyarlanmıştır. Çok-amaçlı sezgisel algoritmaların birçoğu pareto baskınlığı tabanlı yöntemler kullanmaktadır. Bayraktar ve arkadaşları tarafından geliştirilen Çok-amaçlı RGO algoritmasında ise pareto cephesinin elde edilmesinde hızlı bastırılmayan sıralama kullanılmaktadır. Bu çalışmada ise [19] nolu çalışmaya ek olarak çeşitlilik ve yakınsamayı daha fazla arttırabilmek için hızlı bastırılmayan sıralama ile adaptif ızgara yaklaşımı bir arada kullanılmaktadır. Aynı zamanda çözüm uzayının pareto cephesine yakınsarken belli noktalara takılıp kalmaması için yakın komşulukları ortadan kaldıran ve yakınsama yapılan çözümü derecelendiren bir algoritma geliştirilmiştir. Bu algoritmaya göre daha önceden belirlenmiş bir mesafe içerisinde yer alan çözümler çözüm uzayından çıkarılır. Böylelikle yakınsama gerçekleşirken belli bölgede yoğunlaşma ortadan kaldırılmış olacaktır. Yakınsanan çözümler her yakınsamada derecelendirilip en düşük dereceli çözümler arasından rasgele seçim yapılarak çeşitlilik sağlanmış olur.

A. PAREMETRELERİN TANIMLANMASI

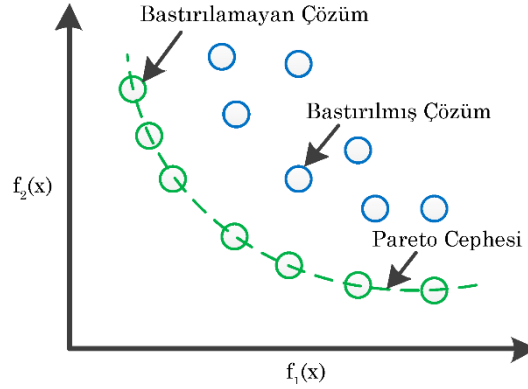
HÇA-RGO algoritması, popülasyon boyutu, karar değişkenleri sayısı, maksimum iterasyon sayısı, amaç sayısı, depo kapasitesi ve alt-üst limitler gibi parametreler ile optimizasyon işlemini gerçekleştirmektedir. Bunların dışında tek-amaçlı RGO algoritmasında yer alan temel parametreler de kullanılmaktadır. Popülasyon ve karar değişkenlerinin sayısı, çalışma performansı ve sonuçları etkileyen en önemli parametrelerdir. Bunların dışında depo kapasitesi ve maksimum bastırılmayan pareto-optimal nokta sayısı belirlenmektedir.

B. BAŞLANGIÇ KOŞULLARININ OLUŞTURULMASI

HÇA-RGO algoritması başlangıç aşamasında diğer popülasyon tabanlı algoritmalara benzer şekilde alt ve üst limitler içerisinde rasgele hız ve konum değerleri ile optimizasyon sürecini başlatmaktadır. Konum bilgisi optimize edilecek amaç fonksiyonlarını oluşturan karar değişkenleridir. Popülasyon sayısı kadar oluşturulan konumsal bilgiler amaç fonksiyonuna aktarılarak her bir konuma karşılık gelen değerler hesaplanır. Algoritmanın ilerleyen aşamalarında konum bilgisi, kurulum aşamasında belirlenmiş olan parametreler ve her bir konuma karşılık gelen değerlerden elde edilen hız bilgisi ile güncellenerek amaç fonksiyonu optimize edilir.

Çok-amaçlı optimizasyon algoritmalarında pareto verimliliği, pareto optimumu veya optimum pareto gibi kavramlar ön plana çıkmaktadır. Arama uzayında yer alan bir çözümün diğer çözümler tarafından bastırılmıyor olması o çözümün optimum çözüm olduğunu ifade eder. Bu çözüme pareto-optimal çözüm, tüm pareto-optimal çözümlerin oluşturduğu kümeye pareto-optimal küme adı verilir [26].

Çok-amaçlı optimizasyon algoritmalarında amaç, pareto-optimal kümeyi oluşturan çözüm noktalarını elde edebilmektir. Bu küme aynı zamanda bastırılmayan çözüm kümesi olarak adlandırılır [27]. Şekil 1'de çözüm kümesinde yer alan bastırılmış ve bastırılmayan çözümler bir arada gösterilmiştir.



Şekil 1. Pareto Cephesi, Bastırılan ve Bastırılmayan Çözümler

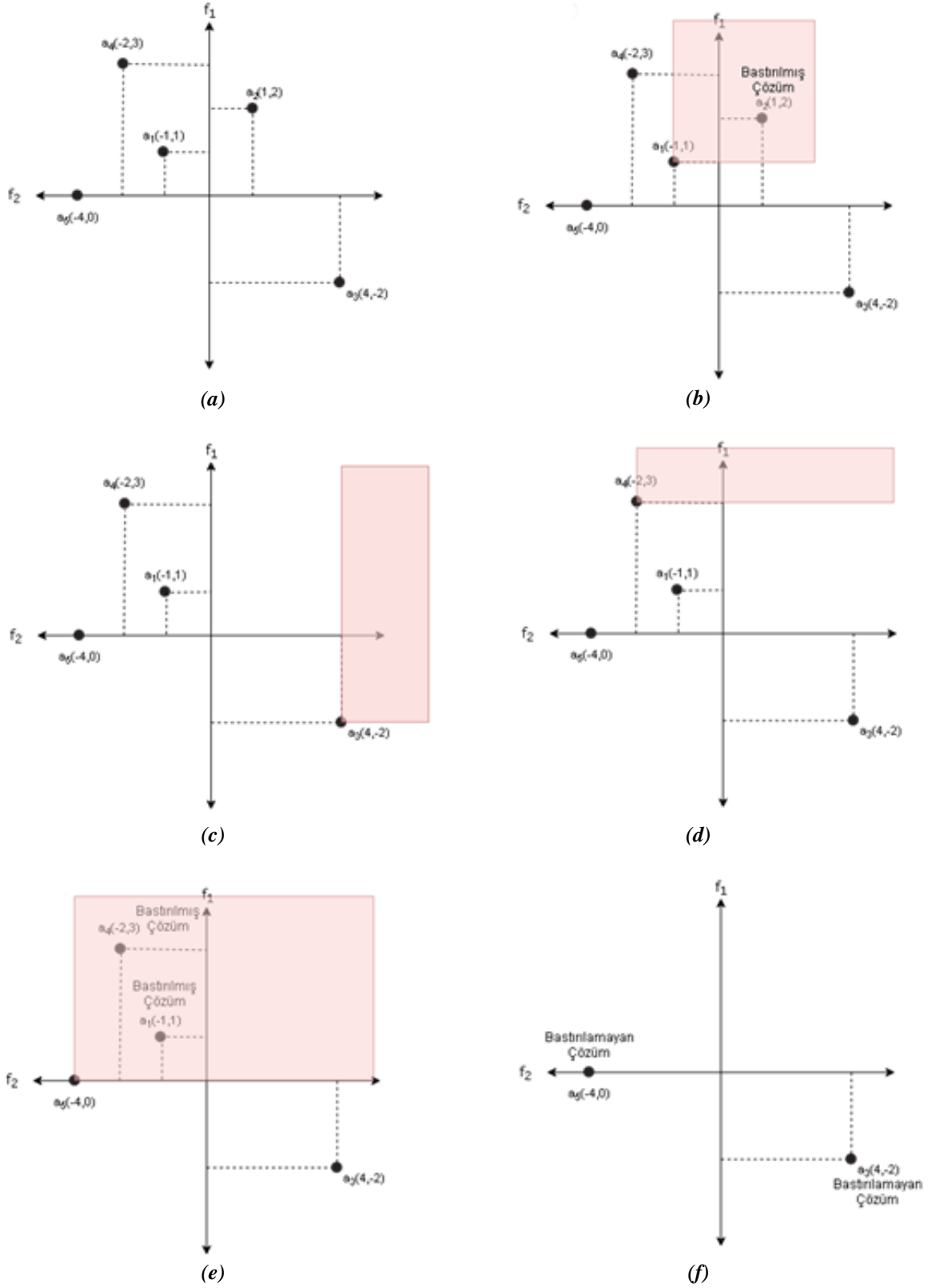
C. BASTIRILAMAYAN ÇÖZÜMLERİN BELİRLENMESİ

Tek-amaçlı optimizasyon problemlerinde genel olarak elde edilen çözüm kümesinden bir veya birkaç çözüm seçilerek optimum çözüme yakınsama işlemi yapılır. Seçilen çözümler problemin amacına göre minimum veya maksimum değerlikte çözümler arasından yapılır. Çok-amaçlı optimizasyon problemlerinde tek bir amaç olmadığından seçim işlemi pareto-optimum çözüm kümesinde yer alan çözümler arasından yapılır.

Çözüm kümesinde bastırılmayan çözümlerin elde edilmesi iki aşamada gerçekleşir. Birinci aşamada popülasyonda yer alan bir çözümün diğer çözümlere baskın olup olmadığı kontrol edilir. İki-amaçlı bir minimizasyon probleminde bir çözüm, çözüm uzayında bir $A(f_2, f_1)$ noktası şeklinde ifade edilir. f_1 ve f_2 belirli koşullar altında problemin çözüm noktasını ifade etmektedir. Eğer diğer tüm çözüm noktaları f_1 , f_2 noktasından büyük bir değere sahipse bu çözüm bastırılmayan çözüm olarak ifade edilir.

Bastırılmayan çözümlerin elde edilmesini bir örnek ile ifade etmek gerekirse; $a_1(-1,1)$, $a_2(1,2)$, $a_3(4,-2)$, $a_4(-2,3)$ ve $a_5(-4,0)$ noktaları önceden belirlenmiş konum ve hız değerlerine ait çözüm kümesini oluşturan noktalar olsun. Bu çözümlerden hangisi veya hangilerinin pareto-optimal çözüm olduğunu belirlemek için sırasıyla her bir noktanın baskın olduğu bölge belirlenir ve o bölgede yer alan noktalar çözüm kümesinden çıkartılır. a_1 noktası ele alındığında $f_2 > -1$ ve $f_1 > 1$ bölgesi a_1 noktasına ait bastırılmış bölge olarak ifade edilir ve bu bölge içerisinde yer alan noktalar çözüm kümesinden çıkartılır. a_1 noktasına ait baskın bölgede a_2 noktası yer alındığından bu nokta çözüm kümesinden çıkartılacaktır. Bir sonraki adımda a_2 noktası çözüm kümesinden çıkartıldığından pareto-optimal çözüm arayışına a_3 noktası ile devam edilir. Geriye kalan tüm noktalara ait baskın bölgeler incelendiğinde başta baskın olan a_1 noktası da dahil olmak üzere a_4 noktası çözüm kümesinden çıkartılarak pareto-optimal çözümler elde edilmiş olur. Pareto-optimal çözümlerin belirlenme aşaması Şekil 2'de gösterilmiştir.

Elde edilen pareto-optimal çözüm kümesi diğer çözümlerden bağımsız tutularak algoritmanın başlangıç aşaması tamamlanmış olur. Bir sonraki aşamada konum ve hız bilgileri iteratif bir şekilde güncellenerek yeni pareto-optimal çözümler elde edilir.



Şekil 2. (a) Çözüm kümesi (b) a_1 noktasına ait baskın bölge (c) a_3 noktasına ait baskın bölge (d) a_4 noktasına ait baskın bölge (e) a_5 noktasına ait baskın bölge (f) Pareto-optimal çözüm kümesi

D. LİDER SEÇİMİ

Baskın olmayan sıralama yaklaşımında çözüm kümesinde yer alan her bir çözüm için baskınlık değerine göre bir derece verilir. En düşük dereceli çözüm baskın çözüm olarak ifade edilir. Yapılan bu çalışmada çözümlerin baskınlıklarını derecelendirmek yerine her iterasyonda en baskın çözümler depoya kaydedilir. Daha sonra depodan lider seçimi yapılarak çözüm kümesinde yer alan çözümlerin konum ve hız bilgileri güncellenir.

Lider seçiminde depoda yer alan her çözüm lider olarak kabul edilir. Adaptif ızgara yaklaşımına göre çözüm uzayı eşit-aralıklı ızgaralara bölünür. Bu ızgaralar arasında yer alan her bir çözüm yine o ızgarada yer alan baskın çözümü lider kabul eder. Eğer ızgara içerisinde birden fazla baskın çözüm mevcut ise liderler arasında rasgele seçim yapılır ve lider sayacı adı verilen bir değişken o çözümün kaç kere lider seçildiğini kaydeder. Bir sonraki aşamada aynı ızgarada yer alan baskın çözümler arasında lider seçimi yapılırken lider sayacının değerine göre rasgele seçim yapılır. Lider sayacı yüksek olan bir baskın çözümün seçilme ihtimali lider sayacı ile ters orantılı olacak şekilde düşük olacaktır.

Başlangıçta geniş bir alana sahip olan ızgara, iterasyon sayısına bağlı olarak küçülerek arama uzayı daraltılır. Bir sonraki aşamadan lider çözümler kullanılarak elde edilen tüm çözümlerin konum ve hız bilgisi güncellenir.

E. KONUM VE HIZ GÜNCELLEME

RGO algoritması, hareketin dinamiği kanununu temel alarak rüzgârı oluşturan en küçük hava parselini konumlandırmaktadır. Algoritmada konumsal değişiklik diğer parçacık tabanlı algoritmalarda olduğu gibi en iyi veya en iyiye yakın çözümlere doğru olmaktadır. En iyi çözümlere doğru gerçekleşen hareket yeni en iyi çözümler elde edilmesine olanak sağlamaktadır. Bu aşamada konumsal değişiklik lider seçilmiş çözümlere doğru olmaktadır.

Konum ve hız bilgisi güncellenirken eşit 1’de yer alan formül kullanılmaktadır. Eşitlik kullanılarak elde edilen hız bilgisi hava parselinin o anki konumuna eklenerek yeni konum elde edilir.

Bir sonraki aşamada konum bilgisi güncellenen hava parseline ait yeni uygunluk değerleri hesaplanır ve çözüm kümesinde yer alan yeni baskın çözümler depoya eklenir.

F. YAKIN KOMŞULUKLARIN SİLİNMESİ

Pareto-optimal çözümlerin daha çeşitli ve gerçek çözüme yakınsamış olabilmesi için elde edilen pareto-optimal çözüm kümesinde belli bir mesafenin altında bulunan komşuluklarda temizleme işlemi yapılır. Çözüm kümesinde yer alan her bir çözümün diğer çözümler ile olan öklid uzaklığı hesaplanır. A ve B noktaları pareto-optimal çözüm kümesinde yer alan iki farklı nokta olsun. Bu iki noktanın arasındaki mesafe eşitlik 3’de yer alan formül ile hesaplanır.

$$d(A, B) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (3)$$

Birbirine oldukça yakın olan komşuluklar çıkarılarak çözüm kümesinin daha çeşitli ve yakınsamış olması sağlanır.

G. PARETO-OPTİMAL ÇÖZÜM KÜMESİNİN ELDE EDİLMESİ

Tüm bu işlemler önceden belirlenmiş olan iterasyon sayısına kadar devam eder. Her iterasyonda pareto-optimal çözüm kümesi içerisinde liderler seçimi yapılır. Konum ve hız bilgileri bu liderler temel alınarak güncellenir. Yeni çözümler elde edildikten sonra baskın olmayan çözümler çıkartılır, baskınlar ise bir sonraki iterasyonda kullanılmak üzere depo adı verilen kümeye eklenir. İterasyon işlemi sınıra

ulaştığında depoda yer alan çözümler çok amaçlı problemin pareto-optimal çözümleri olacaktır. Algoritmaya ait işlem adımları aşağıda yer alan sözde-kod'da detaylı olarak gösterilmiştir.

Sözde Kod	
Başlangıç	
1.	Parametreleri Belirle
2.	Parçacıklara Ait Konum ve Hız Bilgisi Oluştur
3.	Konum ve Hız Bilgilerine Göre Uygunluk Değerlerini Bul
4.	Baskınları Bul Depola
Ana Döngü	
While $i < MaxIt$	
While $j < PopSize$	
1.	Depodan Lider Seç
2.	Izgara Oluştur
3.	Liderlere Göre Konum ve Hız Güncelle
4.	Konum ve Hız Bilgilerine Göre Uygunluk Değerini Bul
End	
1.	Baskınları Bul Depola
2.	Yakın Komşulukları Sil
End	
5.	Baskınları Görüntüle

IV. DENEYSEL SONUÇLAR

A. TEST PROBLEMLERİNİN ÇOK-AMAÇLI RGO İLE ÇÖZÜMLERİ

HÇA-RGO algoritmasını performans açısından değerlendirmek için literatürde yer alan 7 farklı kısıtsız çok amaçlı test problemi kullanılmıştır. Kullanılan test fonksiyonları Tablo 1'de gösterilmiştir. Bununla birlikte HÇA-RGO algoritmasına ait sonuçlar, NSGA II ve Çok-Amaçlı Parçacık Sürü Optimizasyonu (MOPSO) gibi iyi bilinen algoritmalar kullanılarak elde edilen sonuçlar ile karşılaştırılmıştır [12] [28]. Her bir fonksiyon için kullanılan parametreler Tablo 2'de gösterilmiştir. Deneysel sonuçların elde edilmesinde kullanılan sistemin temel özellikleri şu şekildedir:

- İşlemci : Intel(R) Core(TM) i7-7700 CPU @ 3.60 GHz
- RAM : 16.0 GB
- İşletim Sistemi : Windows 10 Enterprise N, x64
- Harddisk : 256 GB SSD

Aynı zamanda hata payını minimize etmek için her bir optimizasyon işlemi 30 tekrarlı test şeklinde uygulanmıştır.

Tablo 1. Çok Amaçlı test problemleri

Problem	Formül	Arama Uzayı
F1 Fonseca- Fleming [29]	$f_1(x) = 1 - e^{-\sum_{i=1}^n (x_i - \frac{1}{\sqrt{n}})^2}$ $f_2(x) = 1 - e^{-\sum_{i=1}^n (x_i + \frac{1}{\sqrt{n}})^2}$	$-4 \leq x_i \leq 4$ $1 \leq i \leq n$
F2 Kursawe [30]	$f_1(x) = \sum_{i=1}^2 -10e^{-0.2\sqrt{x_i^2 + x_{i+1}^2}}$ $f_2(x) = \sum_{i=1}^3 x_i ^{0.8} + 5\sin(x_i^3)$	$-5 \leq x_i \leq 5$ $1 \leq i \leq 3$
F3 Poloni [31]	$f_1(x, y) = 1 + (A_1 - B_1(x, y))^2 + (A_2 - B_2(x, y))^2$ $f_2(x, y) = (x + 3)^2 + (y + 1)^2$ $A_1 = 0.5 \sin(1) - 2 \cos(1) + \sin(2) - 1.5 \cos(2)$ $A_2 = 1.5 \sin(1) - \cos(1) + 2 \sin(2) - 0.5 \cos(2)$ $B_1(x, y) = 0.5 \sin(x) - 2 \cos(x) + \sin(y) - 1.5 \cos(y)$ $B_2(x, y) = 1.5 \sin(x) - \cos(x) + 2 \sin(y) - 0.5 \cos(y)$	$-\pi \leq x, y \leq \pi$

Tablo 1 (devam). Çok Amaçlı test problemleri

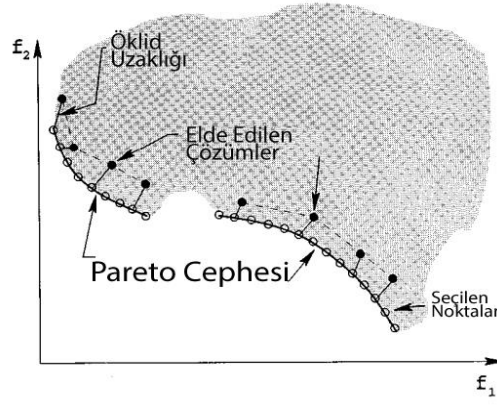
F4	Schaffer N.2 [32]	$f_1(x) = \begin{cases} -x, & x \leq 1 \\ x-2, & 1 < x \leq 3 \\ 4-x, & 3 < x \leq 4 \\ x-4, & x > 4 \end{cases}$ $f_2(x) = (x-5)^2$	$-5 \leq x \leq 10$
F5	ZDT 1 [33]	$f_1(x) = x_1$ $f_2(x) = g(x)h(f_1(x), g(x))$ $g(x) = 1 + \frac{9}{29} \sum_{i=2}^{30} x_i$ $h(f_1(x), g(x)) = 1 - \sqrt{\frac{f_1(x)}{g(x)}}$	$0 \leq x_i \leq 1$ $1 \leq i \leq 30$
F6	ZDT 2 [33]	$f_1(x) = x_1$ $f_2(x) = g(x)h(f_1(x), g(x))$ $g(x) = 1 + \frac{9}{29} \sum_{i=2}^{30} x_i$ $h(f_1(x), g(x)) = 1 - \left(\sqrt{\frac{f_1(x)}{g(x)}} \right)^2$	$0 \leq x_i \leq 1$ $1 \leq i \leq 30$
F7	ZDT 3 [33]	$f_1(x) = x_1$ $f_2(x) = g(x)h(f_1(x), g(x))$ $g(x) = 1 + \frac{9}{29} \sum_{i=2}^{30} x_i$ $h(f_1(x), g(x)) = 1 - \sqrt{\frac{f_1(x)}{g(x)}} - \frac{f_1(x)}{g(x)} \sin(10\pi f_1(x))$	$0 \leq x_i \leq 1$ $1 \leq i \leq 30$

Tablo 2. Kullanılan Parametreler

Parametreler	F1	F2	F3	F4	F5	F6	F7
Popülasyon	100	100	100	100	100	100	100
Maks. İter.	300	300	300	300	500	500	500
MaxV.	0.1	0.3	0.3	0.1	1	1	1
MinV.	0.05	0.01	0.05	0.05	0.01	0.01	0.01
MinDist.	0.01	0.1	0.1	0.1	0.01	0.01	0.01

Kullanılan parametreler her test fonksiyonu için benzerlik göstermektedir. Her fonksiyon için popülasyon sayısı 100 olarak tanımlanmıştır. Maksimum iterasyon sayısı 300 olarak tanımlandığında pareto-optimal çözümler doyuma ulaşmaktadır. Karar verme değişkenlerinin fazla olduğu test fonksiyonlarında çözüm kümesini daha çeşitli hale getirmek için iterasyon sayısı artırılmıştır. Diğer çok amaçlı algoritmalar incelendiğinde maksimum iterasyon sayısının 500 olması kabul edilebilir. Maksimum ve minimum hız değerleri algoritmanın alt ve üst limitlerine bağlı olarak değişmektedir. Minimum mesafe pareto-optimal çözümlerin birbirine ne kadar yakın olması gerektiğini göstermektedir. Yakınsama ve çeşitlilik için bu değer oldukça önemlidir. Algoritmanın belli noktalara takılıp kalmaması için minimum mesafe tanımlaması uygun şekilde yapılmalıdır.

Elde edilen çözümlerin kalitesini ölçmek için mesafe metriği Υ kullanılmıştır [12]. İlk aşamada kullanılan test fonksiyonuna ait gerçek pareto-optimal çözümlerden eşit-aralıklı 100 çözüm seçilir. Sonrasında bu çözümler ile NSGA II, MOPSO ve HÇA-RGO algoritmalarına ait pareto-optimal çözümler arasındaki öklid uzaklığı hesaplanır. Hesaplanan uzaklıklar arasında en düşük değerlikte mesafe kaydedilir. Her bir eşit-aralıklı gerçek pareto-optimal çözüm için kaydedilen minimum mesafelerin ortalaması ve standart sapması hesaplanır. Elde edilen değerler ne kadar küçük olursa algoritma o kadar iyi yakınsamış ve çeşitlilik göstermiş olacaktır. Mesafe metriği Υ Şekil'3 de gösterilmiştir [12].

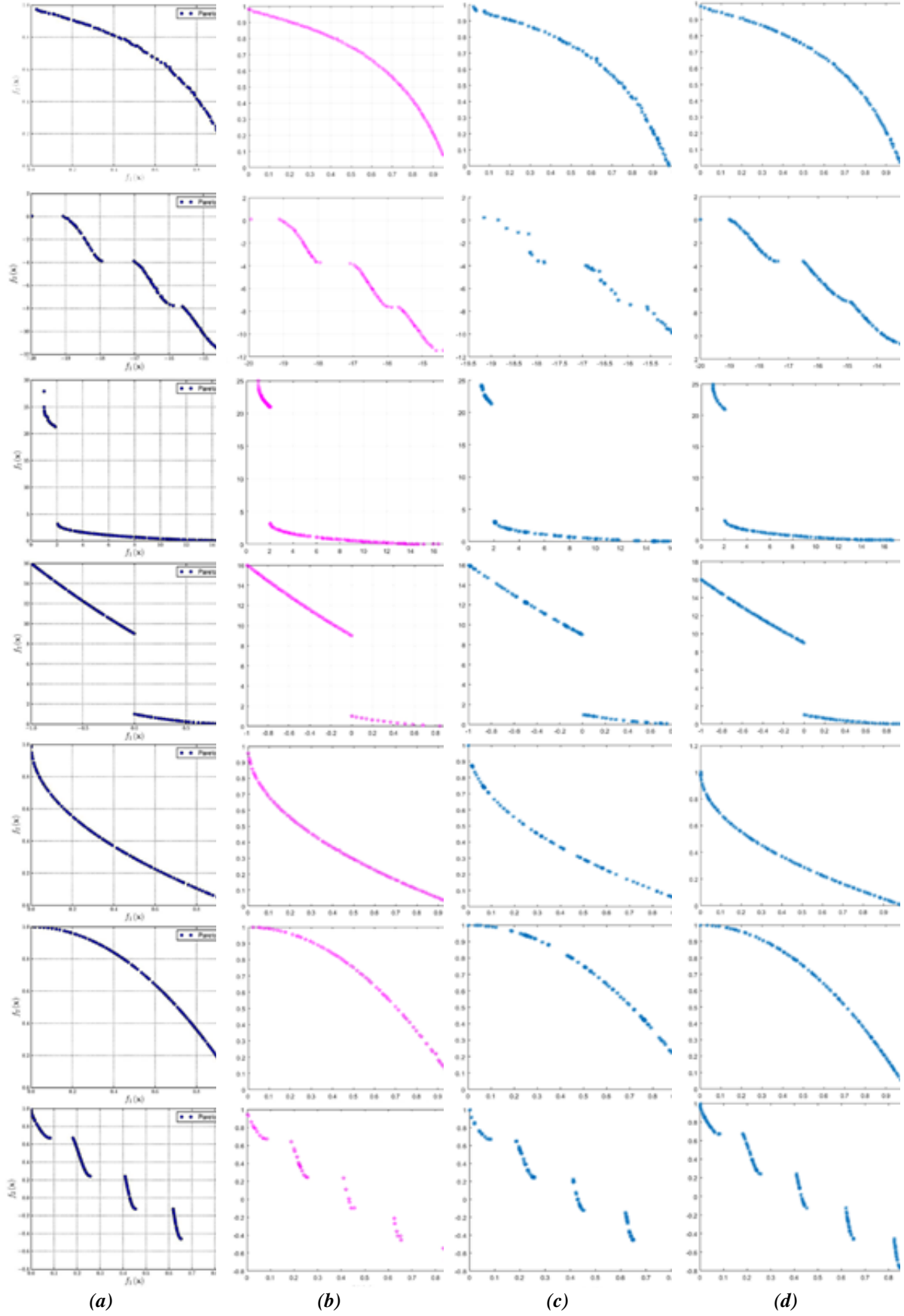


Şekil 3. Mesafe metriği Y

Her bir test fonksiyonu Tablo 2’de yer alan parametreler kullanılarak NSGA II, MOPSO ve HÇA-RGO algoritmalarına uygulanmış ve elde edilen pareto cepheleri Şekil 4’de gösterilmiştir. Simülasyon sonuçlarına ait veriler Tablo 3’de gösterilmiştir. Ortalama ve standart sapma değerleri 30 tekrarlı test sonucunda elde edilmiştir. Ortalama değer her bir çözümün gerçek pareto-optimal çözüme ne kadar yakınsadığını ifade eder. Her bir algoritmaya ait ortalama minimum mesafe değerleri incelendiğinde 4 adet test fonksiyonunda HÇA-RGO diğer algoritmalarından daha iyi sonuçlar vermiştir. Karar değişkeni sayısı fazla olan F6 ve F7 fonksiyonlarında NSGA II algoritması ön plana çıkarken HÇA-RGO ile kıyaslandığında sonuçlar oldukça yakındır. Standart sapma değeri pareto-optimal çözümlerin ne kadar çeşitli olduğunu ifade eder. Ortalama standart sapma değerleri incelendiğinde F1, F2 ve F4 fonksiyonlarında en iyi, diğer fonksiyonlarda ise en iyi çözüme yakın değerler elde edildiği gözlenmiştir.

Tablo 3. Mesafe metriğine ait deneysel sonuçlar; Ortalama (\bar{x}) ve Standart Sapma (σ)

Alg.		F1	F2	F3	F4	F5	F6	F7
HÇA-RGO	\bar{x}	0,004082481	0,0390115	0,050263877	0,028397946	0,004625615	0,006653173	0,017426835
	σ	0,002380803	0,019369698	0,060936042	0,019493732	0,005031273	0,004855082	0,015186426
NSGA II	\bar{x}	0,005153612	0,850896001	0,062463868	0,023169799	0,004756987	0,004728225	0,005777209
	σ	0,003454363	0,335321862	0,041788833	0,021920297	0,003577742	0,003737699	0,005372786
MOPSO	\bar{x}	0,007312853	0,16818375	0,12612483	0,041097475	0,009171657	0,008377408	0,010020761
	σ	0,004455553	0,153264052	0,13389233	0,053504649	0,009676832	0,00906622	0,010876826



Şekil 4. (a) Gerçek Pareto Cephesi, (b) HÇA-RGO, (c) MOPSO, (d) NSGA II

B. DOĞRUSAL OLMAYAN DENKLEM SİSTEMLERİNİN ÇOK-AMAÇLI RGO İLE ÇÖZÜMLERİ

Doğrusal olmayan denklem sistemleri; n adet bağımsız değişkeni ve n adet doğrusal olmayan denklemi olan sistemlerdir.

$$\begin{aligned}
f_1(x_1, x_2, \dots, x_n) &= 0, \\
f_2(x_1, x_2, \dots, x_n) &= 0, \\
\dots\dots\dots \\
f_n(x_1, x_2, \dots, x_n) &= 0
\end{aligned} \tag{4}$$

Eşitlik değerleri sol tarafa atılacak olursa (denklemler sifıra eşit olacak), doğrusal olmayan denklem sisteminin çözümü; n adet denklemi birlikte çözecek n adet kökün bulunması demektir. Bir denklemin kökü onu sıfır yapan veya mutlak minimum yapan değerdir [34].

Doğrusal olmayan denklem sistemlerinin çözümünde çok çeşitli yöntemler kullanılmaktadır. Bazı çalışmalarda doğrusal olmayan denklem sistemleri kısıtsız tek amaçlı optimizasyon problem olarak ele alınır. Bu tür problemlerde minimize edilecek amaç fonksiyonu, eşit. 5’deki gibi doğrusal olmayan denklemlerin kareleri toplamı olarak kabul edilir.

$$fmin = \sum_{i=1}^n f_i(x_1, x_2, \dots, x_n)^2 \tag{5}$$

Ancak bu yaklaşım tüm denklemlerin kökünü bulmayı değil, toplam hatayı minimize etmeyi garanti eder.

Birçok çalışmada ise doğrusal olmayan denklem sistemleri çok amaçlı optimizasyon problem olarak ele alınır [35] [36]. Her bir denklem mutlak olarak minimize edilmesi gereken bir amaç fonksiyonudur. Doğrusal olmayan denklem sistemlerini çok amaçlı optimizasyon problem olarak çözümlenmenin avantajı, denklem sisteminin tek bir çözümü yerine alternatif çözümlerinin de optimizasyon algoritmaları ile bulunabilmesidir.

Doğrusal olmayan denklem sistemlerinin çözümünde sıklıkla kullanılan yöntemler [37] [38] [39] [40] [41] [42] [43] de verilmiştir. Bu yöntemlerden en çok kullanılanlar aşağıda sıralanmıştır;

- Trust-Region [44]
- Broyden [45]
- Secant [41]
- Halley [43]

HÇA-RGO algoritmasını doğrusal olmayan denklem sistemlerinin çözümündeki performansını değerlendirmek için literatürde yer alan 4 farklı denklem sistemi kullanılmıştır. Kullanılan denklem sistemleri Tablo 4’de gösterilmiştir.

Tablo 4. Doğrusal olmayan denklem sistemleri

Problem	Formül	Arama Uzayı
D1 [36]	$ \begin{aligned} x_1^2 + x_2^2 - 1 &= 0 \\ x_1 - x_2 &= 0 \end{aligned} $	$ \begin{aligned} -1 \leq x_i \leq 1 \\ 1 \leq i \leq 2 \end{aligned} $
D2 [46]	$ \begin{aligned} x_1^2 + 2x_2^2 + \cos(x_3) - x_4^2 &= 0 \\ 3x_1^2 + x_2^2 + \sin^2(x_3) - x_4^2 &= 0 \\ -2x_1^2 - x_2^2 - \cos(x_3) + x_4^2 &= 0 \\ -x_1^2 - x_2^2 - \cos^2(x_3) + x_4^2 &= 0 \end{aligned} $	$ \begin{aligned} -2 \leq x_i \leq 2 \\ 1 \leq i \leq 4 \end{aligned} $

Tablo 4 (devam). Doğrusal olmayan denklem sistemleri

D3	Interval Arithmetic Benchmark [46]	$x_1 - 0.25428722 - 0.18324757x_4x_3x_9 = 0$	$-2 \leq x_i \leq 2$ $1 \leq i \leq 10$
		$x_2 - 0.37842197 - 0.16275449x_1x_{10}x_6 = 0$	
		$x_3 - 0.27162577 - 0.16955071x_1x_2x_{10} = 0$	
		$x_4 - 0.19807914 - 0.15585316x_7x_1x_6 = 0$	
		$x_5 - 0.44166728 - 0.19950920x_7x_6x_3 = 0$	
		$x_6 - 0.14654113 - 0.18922793x_8x_5x_{10} = 0$	
		$x_7 - 0.42937161 - 0.21180486x_2x_5x_8 = 0$	
		$x_8 - 0.07056438 - 0.17081208x_1x_7x_6 = 0$	
		$x_9 - 0.34504906 - 0.19612740x_{10}x_6x_8 = 0$	
		$x_{10} - 0.42651102 - 0.21466544x_4x_8x_1 = 0$	
D4	Neuro- physiology Application [46]	$x_1^2 + x_3^2 = 1$	$-1 \leq x_i \leq 1$ $1 \leq i \leq 6$ $1 \leq c \leq 4$ $c_i = 0$
		$x_2^2 + x_4^2 = 1$	
		$x_5x_3^3 + x_6x_4^3 = c_1$	
		$x_5x_1^3 + x_6x_2^3 = c_2$	
		$x_5x_1x_3^2 + x_6x_4^2x_2 = c_3$	
		$x_5x_1^2x_3 + x_6x_2^2x_4 = c_4$	

Denklemlerin klasik bir yöntem olan Trust-Region metodu kullanılarak elde edilen çözümleri Tablo 5’de gösterilmiştir [47].

Tablo 5. Trust-Region ile doğrusal olmayan denklem sistemlerinin çözümleri

Problem	Karar Değişkeni	Çözüm
D1	2	$x = (0.707107, 0.707107)$ $eq_1 = 0$ $eq_2 = 2.01E - 7$
D2	4	$x = (-9.6E - 08, -2.8E - 09, 1.394741, 0.5349)$ $eq_1 = -0.11097$ $eq_2 = 0.683206$ $eq_3 = -0.46126$ $eq_4 = 0.110971$
D3	10	$x = (0.257833, 0.381097, 0.278745, 0.200669, 0.445251, 0.149184, 0.43201, 0.073403, 0.345967, 0.427326)$ $eq_1 = 1.88E - 11$ $eq_2 = -3E - 14$ $eq_3 = -3.7E - 14$ $eq_4 = -2E - 13$ $eq_5 = 1.09E - 11$ $eq_6 = 2.06E - 12$ $eq_7 = 4.7E - 13$ $eq_8 = -2.2E - 13$ $eq_9 = -1.3E - 12$ $eq_{10} = -2.3E - 12$
D4	6	$x = (0.754488, 0.954881, 0.656314, 0.296989, -1.4E11, 4E - 07)$ $eq_1 = 6.88E - 15$ $eq_2 = 3E - 09$ $eq_3 = 1.05E - 08$ $eq_4 = 3,48E - 07$ $eq_5 = 3,37E - 08$ $eq_6 = 2.06E - 12$

Bununla birlikte HÇA-RGO algoritmasına ait sonuçlar, test problemlerinde olduğu gibi NSGA II ve Çok-Amaçlı Parçacık Sürü Optimizasyonu (MOPSO) kullanılarak elde edilen sonuçlar ile karşılaştırılarak Tablo 6’da gösterilmiştir. Deneysel sonuçlar incelendiğinde test problemlerinin dışında doğrusal olmayan denklem sistemlerinin çözülmesinde başarılı sonuçlar elde ettiği ve algoritmanın kullanılabilir olduğu gözlenmiştir.

Tablo 6. Doğrusal olmayan denklem sistemleri karşılaştırmalı sonuçlar; Ortalama (\bar{x}), Standart Sapma (σ) ve Toplam (Σ)

Denklemler	HÇA- RGO		NSGA-II		MOPSO		
	\bar{x}	Σ	\bar{x}	σ	\bar{x}	σ	
D1	eq_1	0.000220	0.000422	3.654401e-10	2.001598e-09	4.171823e-05	9.180501e-05
	eq_2	0.000892	0.001860	4.229778e-10	2.316744e-09	2.081175e-05	4.296842e-05
	Σ	0.001113	0.001839	7.884180e-10	4.318343e-09	6.252999e-05	0.000105

Tablo 6 (devam). Doğrusal olmayan denklem sistemleri karşılaştırmalı sonuçlar; Ortalama (\bar{x}), Standart Sapma (σ) ve Toplam (Σ)

D2	eq_1	0.045320	0.051632	0.211931	0.166559	0.194218	0.152565
	eq_2	0.389621	0.068497	0.586158	0.366147	0.435695	0.122648
	eq_3	0.352602	0.088929	0.089188	0.144438	0.174631	0.147708
	eq_4	0.046171	0.052305	0.272755	0.202010	0.316751	0.150192
	Σ	0.833715	0.105726	1.160033	0.243731	1.121297	0.178830
D3	eq_1	0.117913	0.171911	1.549991	1.381743	0.624084	0.734583
	eq_2	0.114158	0.174627	1.093895	1.288475	0.626450	0.823823
	eq_3	0.093223	0.156028	0.846264	1.041580	0.675210	0.718247
	eq_4	0.097002	0.113068	1.084293	0.973015	0.455211	0.565928
	eq_5	0.080523	0.096309	0.949058	1.135044	0.827122	0.704204
	eq_6	0.114201	0.207663	0.889087	1.176545	0.748125	0.765581
	eq_7	0.090627	0.119581	1.211447	1.303337	0.764480	0.912182
	eq_8	0.107067	0.204127	0.928754	1.094620	0.534452	0.523205
	eq_9	0.123530	0.205318	1.027816	1.160161	0.621151	0.574184
	eq_{10}	0.150019	0.183516	0.730268	0.920892	0.387760	0.666612
	Σ	1.088267	0.370239	10.31087	3.192165	6.264050	2.184322
D4	eq_1	0.035533	0.042967	0.131760	0.168764	0.012220	0.016821
	eq_2	0.026702	0.025672	0.124733	0.182261	0.010031	0.016915
	eq_3	0.020143	0.030866	0.113107	0.103906	0.004673	0.006881
	eq_4	0.020556	0.030583	0.117179	0.112201	0.010950	0.014421
	eq_5	0.013941	0.021771	0.167275	0.151491	0.012111	0.021989
	eq_6	0.012505	0.015368	0.178582	0.169187	0.007654	0.011480
	Σ	0.129382	0.065639	0.832637	0.361976	0.057642	0.040228

IV. SONUÇ

Bu çalışmada, çok-amaçlı optimizasyon problemlerini çözmek için RGO algoritması kullanılarak hibrit bir yaklaşım geliştirilmiştir. Geliştirilen bu hibrit yaklaşımda yakınsama için baskın olmayan sıralama algoritması, çeşitlik için ise adaptif ızgara yaklaşımı kullanılmıştır. Aynı zamanda yakın komşuluk temizleme ve yakınsama derecelendirme ile algoritmanın tek bir çözüme takılmayıp çözüm uzayını çeşitlendirmesi sağlanmıştır. HÇA-RGO, literatürde iyi bilinen NSGA-II ve MOPSO algoritmaları ile karşılaştırılmıştır. Çok-amaçlı test problemlerine ait deneysel sonuçlar incelendiğinde hibrit yaklaşımın 7 test probleminin 4'ünde en iyi sonucu verdiği, doğrusal olmayan denklem sistemlerinin çözümlerine ait deneysel sonuçlar incelendiğinde ise hibrit yaklaşımın 4 problemin 2'sinde en iyi, diğer 2 problemde ise en iyiye oldukça yakın sonuçlar verdiği gözlenmiştir.

Geliştirilen hibrit yaklaşımın elde edilen bulgular neticesinde yakınsama ve çeşitlilik kriterleri temel alındığında literatürde kabul edilebilir olduğunu gözlenmiştir.

V. KAYNAKLAR

- [1] P. Erdoğan, "Doğadan esinlenen optimizasyon algoritmaları ve optimizasyon algoritmalarının optimizasyonu," *Düzce Üniversitesi Bilim and Teknoloji Dergisi*, ss. 293-304, 2016.
- [2] J. H. Holland, "Control and artificial intelligence," in *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology*, Cambridge: MIT Press, 1992.

- [3] J. Kennedy and R. Eberhar, "Particle swarm optimization," presented at Neural Networks Proceedings, IEEE International Conference, Australia, 1995.
- [4] M. Dorigo, M. Birattari and T. Stutzle, "Ant colony optimization," *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28-39, 2006.
- [5] M. Laumanns, L. Thiele, K. Deb and E. Zitzler, "Combining convergence and diversity in evolutionary multiobjective optimization," *Evolutionary Computation*, vol. 10, no. 3, pp. 263-282, 2002.
- [6] N. Srinivas and K. Deb, "Multiobjective optimization using nondominated sorting in genetic algorithms," *Evolutionary Computation*, vol. 2, no. 3, pp. 221-248, 1994.
- [7] S. Rostami and A. Shenfield, "A multi-tier adaptive grid algorithm for the evolutionary multi-objective optimisation of complex problems," *Soft Computing*, vol. 21, no. 17, pp. 4963-4979, 2017.
- [8] D. E. Goldberg, "Genetic algorithms and classifier systems," in *Genetic Algorithms in Search, Optimization and Machine Learning*, USA: Addison-Wesley Publishing Company, 1989, ch. 3, pp. 60-61.
- [9] M. F. P. Costa, A. M. A. Rocha and E. M. Fernandes, "Combining nondominance, objective-order and spread metric to extend Firefly Algorithm to multi-objective optimization," presented at International Conference On Evolutionary Multi-Criterion Optimization, Guimares, Portugal, 2015.
- [10] C. M. Fonseca and P. J. Fleming, "Genetic Algorithms for multiobjective optimization: Formulation discussion and generalization," *ICGA*, vol. 93, 1993.
- [11] J. Horn, N. Nafpliotis and D. Goldberg, "A niched pareto genetic algorithm for multiobjective optimization," presented at IEEE World Congress on Computational Intelligence, Orlando, USA, 1994.
- [12] K. Deb, S. Agrawal, A. Pratap and T. Meyarivan et al. "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II," presented at International Conference On Parallel Problem Solving From Nature, Berlin, Germany, 2000.
- [13] S. Mirjalili, S. M. Mirjalili and A. Lewis, "Grey Wolf Optimizer," *Advances in Engineering Software*, vol. 69, pp. 46-61, 2014.
- [14] S. Mirjalili, S. Saremi, S. M. Mirjalili and L. D. S. Coelho, "Multi-Objective Grey Wolf Optimizer: A novel algorithm for multi-criterion optimization," *Expert Systems with Applications*, vol. 47, pp. 106-119, 2016.
- [15] S. Saremi, S. M. Mirjalili and A. Lewis, "Grasshopper Optimization Algorithm: Theory and application," *Advances in Engineering Software*, vol. 105, pp. 30-47, 2017.
- [16] S. Z. Mirjalili, S. Mirjalili, S. Saremi, H. Faris and I. Aljarah, "Grasshopper Optimization Algorithm for multi-objective optimization problems," *Applied Intelligence*, vol. 48, pp. 805-820, 2018.
- [17] S. Y. Zeng, L. S. Kang and L. X. Ding, "An orthogonal multi-objective evolutionary algorithm for multi-objective optimization problems with constraints," *Evolutionary Computation*, vol. 12, no. 1, pp. 77-98, 2004.
- [18] M. J. Reddy and D. Kumar, "An efficient multi-objective optimization algorithm based on swarm intelligence for engineering design," *Engineering Optimization*, vol. 39, no. 1, pp. 49-68, 2007.

- [19] O. Jadaan and C. R. Rao, "Non-dominated ranked genetic algorithm for solving multi-objective optimization problems: NREGA," *Journal of Theoretical and Applied Information Technology*, pp. 60-67, 2008.
- [20] H. Ghiasi, D. Pasini and L. Lessard, "A non-dominated sorting hybrid algorithm for multi-objective optimization of engineering problems," *Engineering Optimization*, vol. 43, pp. 39-59, 2011.
- [21] H. Nobahari, M. Nikusokhan and P. Siarry, "A multi-objective Gravitational Search Algorithm based on non-dominated sorting," *International Journal of Swarm Intelligence Research*, vol. 3, no. 3, pp. 32-49, 2012.
- [22] S. Sanz, A. Figueras, Á. Sánchez and L. Prieto, "Effective multi-objective optimization with the Coral Reefs Optimization Algorithm," *Engineering Optimization*, vol. 48, pp. 966-984, 2015.
- [23] Z. Bayraktar and M. Kömürçü, "Multiobjective Adaptive Wind Driven Optimization," presented at The 8th International Conference On Computational Intelligence (IJCCI 2016), Porto, Portugal, 2016.
- [24] Z. Bayraktar, M. Komurcu and D. H. Werner, "Wind Driven Optimization (WDO): A novel nature-inspired optimization algorithm and its application to electromagnetics," *Antennas and Propagation Society International Symposium (APSURSI)*, Toronto, Canada, 2010.
- [25] Z. Bayraktar and M. Komurcu, "Adaptive Wind Driven Optimization," presented at BICT'15 Proceedings of The 9th EAI International Conference On Bio-Inspired Information and Communications Technologies, New York City, USA, 2015.
- [26] A. Dias and J. D. Vasconcelos, "Multiobjective Genetic Algorithms applied to solve optimization problems," *IEEE Transactions On Magnetics*, vol. 38, no. 2, pp. 1133-1136, 2002.
- [27] K. Mahesh, P. Nallagownden and I. Elamvazuthi, "Advanced Pareto Front Non-Dominated Sorting Multi-Objective Particle Swarm Optimization for optimal placement and sizing of distributed generation," *Energies*, vol. 9, no. 12, 2016.
- [28] K. E. Parsopoulos and M. N. Vrahatis, "Particle Swarm Optimization method in multiobjective problems," presented at SAC '02 Proceedings of The 2002 ACM Symposium On Applied Computing, Madrid, Spain, 2002.
- [29] C. M. Fonseca and P. J. Fleming, "Multiobjective optimization and multiple constraint handling with evolutionary algorithms-part II: Application example," *IEEE Trans. Syst., Man, Cybern. A*, vol. 28, pp. 38-47, 1998.
- [30] F. Kursawe, "A variant of evolution strategies for vector optimization," presented at PPSN I Proceedings of The 1st Workshop On Parallel Problem Solving From Nature, Berlin, Germany, 1990.
- [31] C. Poloni, "Hybrid GA for Multiobjective Aerodynamic Shape Optimization," presented at Genetic Algorithms in Engineering and Computer Science, Japan, 1997.
- [32] J. D. Schaffer, "Multiple objective optimization with Vector Evaluated Genetic Algorithms," in *Proceedings of The First International Conference On Genetic Algorithms*, USA, 1987, pp. 93-100.
- [33] E. Zitzler, K. Deb and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evolutionary Computation*, vol. 8, no. 2, 2000.
- [34] P. Erdoğan, "Parçacık Sürü Optimizasyonu ile doğrusal olmayan denklem köklerinin bulunması ve Genetik Algoritma ile mukayesesi," *İleri Teknoloji Bilimleri Dergisi*, c. 5, s. 1, 2016.

- [35] S. Qin, no. Zeng, W. Dong and X. Li, "nonlinear equation system solved by many-objective hype," presented at IEEE Congress On Evolutionary Computation (CEC), Sendai, Japan, 2015.
- [36] W. Song, Y. Wang, X. Li and Z. Cai, "Locating multiple optimal solutions of nonlinear equation systems based on multobjective optimization," *IEEE Transactions On Evolutionary Computation*, vol. 19, no. 3, pp. 414-431, 2015.
- [37] C. Brezinski, "Projection methods for systems of equations," *North-Holland*, vol. 7, 1997.
- [38] J. Denis, "On newton's method and nonlinear simultaneous replacements," *SIAM J. Numer. Anal.*, vol. 4, pp. 103-108, 1967.
- [39] J. Denis, "On newton like methods," *Numer. Math.*, vol. 11, pp. 324-330, 1968.
- [40] J. Denis, "On the convergence of broyden's method for nonlinear systems of equations," *Math. Comput.*, vol. 25, pp. 559-567, 1971.
- [41] J. Denis and H. Wolkowicz, "Least change secant methods sizing and shifting," *SIAM J. Numer. Anal.*, vol. 30, pp. 1291-1314, 1993.
- [42] J. Denis, M. El Alem and K. Williamson, "A trust-region algorithm for least-squares solutions of nonlinear systems of equalities and inequalities," *SIAM J. Opt.*, vol. 9, pp. 291-315, 1999.
- [43] J. Ortega and W. Rheinboldt, "On the convergence of Halley's method," in *Iterative Solution of Nonlinear Equations in Several Variables*, New York, USA: Academic Press, 1970, ch. 2, pp. 179-179.
- [44] A. Conn, N. Gould and P. Toint, "Trust-Region methods," presented at SIAM, Philadelphia, USA, 2000.
- [45] C. Broyden, "A class of methods for solving nonlinear simultaneous equations," *Math. Comput.*, vol. 19, pp. 577-593, 1965.
- [46] C. Grosan and A. Abraham, "A new approach for solving nonlinear equations systems," in *IEEE Transactions On Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 38, pp. 698-714, 2008.
- [47] P. Erdoğan, "A new solution approach for non-linear equation systems with Grey Wolf Optimizer," *Sakarya University Journal of Computer and Information Sciences*, pp. 1-11, 2018.