Erciyes Üniversitesi
Fen Bilimleri Enstitüsü Dergisi
Cilt 36, Sayı 3 , 2020

Erciyes University
Journal of Institue Of Science and Technology
Volume 36, Issue 3, 2020

Sınırlama Ele Alış Metotlarının Yapay Arı Kolonisi Algoritması Üzerinde Etkisinin İncelenmesi

Demet ALICI KARACA[1]*, Bahriye AKAY[2],

*[1] Erzincan Binali Yıldırım Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği, ERZİNCAN
[2] Erciyes Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği, KAYSERİ

(Alınış / Received: 02.03.2020, Kabul / Accepted: 19.11.2020, Online Yayınlanma / Published Online: 25.12.2020)

Anahtar Kelimeler
Yapay Arı Kolonisi algoritması,
Sınırlamalı optimizasyon,
Sınırlama ele alış metotları,
PEN-ABC,
SR-ABC,
DEB-SR-ABC

Özet: Sürü zekası algoritmaları arasında performansıyla öne çıkan algoritmalardan biri olan Yapay Arı Kolonisi algoritmasının, tasarım parametrelerinin bazı koşullarla kısıtlandığı ve optimum değerin kabul edilebilir bölge içinde olması gerektiği sınırlamalı optimizasyon problemlerini çözmek için farklı versiyonları geliştirilmiştir. Kısıtları dikkate alarak problemleri çözmek amacıyla ceza terimine dayalı metotlar, çözümleri kabul edilebilir bölgede tutan metotlar, kabul edilebilir ve kabul edilebilir olmayan çözümler arasında ayrım yapan metotlar ve karma metotlar bulunmaktadır. Bu çalışmada sınırlamalı optimizasyon problemlerini çözmek amacıyla temel Yapay Arı Kolonisi algoritmasına ceza fonksiyonları, rasgele sıralama ve stokastik Deb kuralları entegre edilerek yeni yöntemler önerilmiştir. Önerilen yöntemler literatürde sıklıkla kullanılan sınırlamalı test problemleri üzerinde test edilmiş ve literatürdeki algoritmalar ile karşılaştırılmıştır. Yapılan analizler sonucunda çalışma kapsamında önerilen yöntemler sınırlamalı optimizasyon problemlerinin çözümünde diğer algoritmalarla benzer ya da daha iyi sonuçlar üretmiştir.

Analyzing the Effect of Constraint-handling Methods on Artificial Bee Colony Algorithm

Keywords
Artificial Bee Colony algorithm,
Constrained optimization,
constraint-handling methods,
PEN-ABC,
SR-ABC,
DEB-SR-ABC

Abstract: Different modifications of the Artificial Bee Colony algorithm, which stands out with its performance among swarm intelligence algorithms, have been developed to solve the constrained optimization problems in which the design parameters are restricted by certain conditions and the optimum value should be located within the feasible region. In order to solve the problems by considering the constraints, there are methods based on penalty functions, methods based on preserving feasibility of solutions, methods which distinguish between feasible and infeasible solutions and hybrid methods. In this study, in order to solve the constrained optimization problems, new methods have been proposed by integrating penalty functions, stochastic ranking and stochastic Deb's rules into the basic Artificial Bee Colony algorithm. The proposed methods have been tested on well-known constrained test problems in the literature, and the results have been compared with other state-of-the-art algorithms. The results indicate that proposed methods showed similar or better performance to solve constrained optimization problems when compared to the other algorithms considered in this study.

Sorumlu Yazar: demet.karaca@erciyes.edu.tr

1. Giriş

Optimizasyon, kısıtlar altında bir problemin minimum ya da maksimum değerini veren parametre değerlerini bulma işlemidir. Sınırlamalı optimizasyon problemleri birçok bilim ve mühendislik alanında kaynaklarla ilgili kısıtların bulunduğu yapısal tasarım, ekonomi, yerleşim ve mühendislik problemleri gibi gerçel problemlerde karşımıza çıkmaktadır [1]. Matematiksel olarak doğrusal olmayan (nonlineer) sınırlamalı optimizasyon problemi, eşitsizlik ve/veya eşitlik sınırlamalarına tabi olarak $f(\vec{x})$ amaç fonksiyonunu minimize eden $\vec{x}$

parametre vektörünü bulma işlemi olarak tanımlanır. Sürekli uzayda Eşitlik 1'de gösterildiği gibi formülize edilebilir.

$$\begin{aligned} &Minimize\ f(\vec{x}) \\ &g_j(\vec{x}) \leq 0,\ \ j = 1, \dots, m \\ &h_k(\vec{x}) = 0,\ \ k = 1, \dots, n \\ &x_i^l \ \leq \ x_i \leq \ x_i^u,\ \ i = 1, \dots, D \end{aligned} \tag{1}$$

$\vec{x} = (x_1, x_2, x_3, \dots, x_D)$ D-boyutlu tasarım parametreleri vektörü, $f(\vec{x})$ en iyilenecek amaç fonksiyonu, araştırma uzayında kabul edilebilir bölgeyi tanımlayan $g_j(\vec{x})$ ve $h_k(\vec{x})$ sırasıyla eşitsizlik ve eşitlik sınırlama fonksiyonlarıdır [2]. m eşitsizlik sınırlamaları sayısı ve n eşitlik sınırlamaları sayısıdır. x_i^l ve x_i^u, x_i parametresinin sırasıyla alt ve üst sınırlarıdır ve optimizasyon probleminin araştırma uzayını tanımlar. Sınırlamalar araştırma uzayını daralttığından dolayı kabul edilebilir çözümleri bulmak zorlaşmaktadır. Çünkü bir çözüm eğer tüm sınırlamaları sağlıyorsa kabul edilebilir çözüm (feasible), tüm sınırlamalar içinde bir tanesini bile sağlamıyorsa kabul edilebilir olmayan (infeasible) çözümdür. Sınırlamaların etkisiyle zorlaşan problemlerin çözümünde, klasik optimizasyon metotları, amaç fonksiyonunun türevlenebilirliği ve sürekliliği konusunda güçlü varsayımlara dayandığından dolayı yetersiz kalmıştır [3]. Modern optimizasyon yöntemleri genellikle ilk olarak sınırlamasız optimizasyon problemlerinin çözümü için geliştirilmiştir. Sınırlama ele alış (constraint-handling) metotları ile sınırlamalı problemleri çözebilir hale gelmekte ve etkin sonuçlar üretebilmektedir. Sınırlama ele alış metotları, Michalewicz ve Schoenauer [4] tarafından dört kategoriye ayrılmıştır: (i) çözümleri kabul edilebilir bölgede tutan metotlar, (ii) ceza terimine dayalı metotlar, (iii) kabul edilebilir ve kabul edilebilir olmayan çözümler arasında ayrım yapan metotlar ve (iv) karma metotlar.

Runarsson ve Yao, amaç ve ceza fonksiyonların baskınlığına göre ceza fonksiyonları için rasgele sıralamaya (stochastic ranking, SR) dayanan bir yaklaşım önermişlerdir [5]. Runarsson ve Yao başka bir çalışmasında ise kabul edilebilir çözümleri amaç fonksiyonu değerlerine göre, kabul edilebilir olmayan çözümleri ise ceza fonksiyonu değerlerine göre sıralayan aşırı ceza yaklaşımını (over penalty approach, OPA) geliştirmiştir. Ayrıca koordinat eksenine bağlı olarak araştırmanın istenen yönde olmaması sorununun çözümü için ise geliştirilmiş stokastik sıralama (improved stochastic ranking, ISR) yöntemini önermişlerdir [6].

Mezura-Montes ve Coello [7] çalışmasında basit çok üyeli evrimsel stratejiyi (simple multimembered evolution strategy-SMES) önermiştir. Bu yaklaşım ceza fonksiyonlarının kullanımını gerektirmeden kabul edilebilir olmayan çözümlerin popülasyonda kalmasına izin veren basit bir çeşitlilik (diversity) mekanizması kullanır. Basit bir kabul edilebilirlik temelli karşılaştırma mekanizması ile araştırmayı kabul edilebilir bölgeye doğru yönlendirir. Ayrıca daha iyi bir arama yapabilmek için standart evrimsel stratejideki başlangıç adım büyüklüğü azaltılmış ve panmiktik rekombinasyon tekniği kullanılmıştır.

Munoz [8] sınırlamalı optimizasyon problemlerini çözmek için Deb'in kurallarını [9] Parçacık Sürü Optimizasyonu (Particle Swarm Optimization, PSO) algoritmasına entegre etmiştir. Karaboğa ve Akay [10] ise Deb'in kurallarını Diferansiyel Gelişim (Differential Evolution, DE) ve Yapay Arı Kolonisi (Artificial Bee Colony, ABC) algoritmalarına entegre ederek sınırlamalı optimizasyon problemine çözüm önermiştir.

Bu çalışmada Karaboğa [11] tarafından önerilen arıların yiyecek arama davranışını simüle eden ABC algoritmasına farklı sınırlama ele alış metotları entegre edilerek sınırlamalı optimizasyon problemlerini çözmek için yöntemler geliştirilmiştir. İlk olarak sınırlamaların aşım miktarına göre bir ceza uygulayan statik ceza fonksiyonları ABC algoritmasına entegre edilmiştir [12]. İkinci yöntem olarak Runarsson ve Yao [5] tarafından geliştirilen SR yaklaşımı yine standart ABC algoritmasına entegre edilerek sınırlamalı optimizasyon problemlerinin çözümü gerçekleştirilmiştir. Bu yaklaşım ile kabul edilebilir olmayan çözümler de popülasyona dahil edilerek global optimuma ulaşılmaya çalışılmıştır. Son olarak ise ABC algoritmasının seleksiyon mekanizmasında Deb'in kuralları [9] ve SR yaklaşımı birleştirilerek problemlerin çözümünde uygulanmıştır. Bu yöntemle popülasyona hem iyi çözümlerin hem de kabul edilebilir olmayan çözümlerin de dahil edilmesiyle performans ve çeşitlilik katılması amaçlanmıştır.

Çalışmanın ikinci bölümünde sınırlamasız optimizasyon problemlerinin çözümü için geliştirilen ABC algoritması ve bu çalışmada kullanılan sınırlama ele alış metotlarına (ceza fonksiyonları, SR yaklaşımı ve Deb'in kuralları) değinilmiştir. Ardından bu bölümde anlatılan sınırlama ele alış metotlarının ABC algoritmasına nasıl entegre edildiği anlatılmıştır. Bulgular bölümünde önerilen yeni yöntemler sınırlamalı test problemleri üzerinde test edilmiş ve literatürdeki diğer algoritmalarla karşılaştırılmış, elde edilen bulgular analiz edilmiştir. Tartışma ve sonuç bölümünde ise analizlerden varılan genel değerlendirmeler yer almaktadır.

2. Materyal ve Metot

2.1. Yapay Ari Kolonisi (ABC) Algoritması

Sınırlamasız optimizasyon problemlerinin çözümü için Karaboğa tarafından önerilen ABC algoritmasında [11] arıların yiyecek arama davranışından esinlenilmiştir. Bu algoritmada üç tür arı bulunur: işçi arı, gözcü arı ve kaşif arı. Bunlar algoritmanın arama ve seleksiyon yapan birimlerine karşılık gelmektedir. Yiyecek kaynaklarının konumu problemin olası çözümleridir. Yiyecek kaynağında bulunan nektar miktarı çözümlerin kalitesini (uygunluk değeri) ifade eder. ABC algoritması, araştırma uzayındaki çözümlerden en fazla nektara sahip kaynağın yerini bulmaya çalışarak problemi en iyileyecek çözümü bulmaya çalışır. Algoritmanın temel adımları Algoritma 1'de verilmiştir [13]:

Algoritma 1. ABC algoritmasının temel adımları

1. Başlangıç yiyecek kaynağı bölgelerinin üretilmesi
2. **repeat**
3. Görevli arıların yiyecek kaynağı bölgelerine gönderilmesi
4. Yiyecek Kaynaklarının Seçilebilme Olasılıklarının Hesaplanması
5. Gözcü Arıların Yiyecek Kaynağı Bölgesi Seçmeleri
6. Kaşif arı üretimi
7. **until** $cevrim\ sayisi\ =\ maksimum\ cevrim\ sayisi$

1. Başlangıç Yiyecek Kaynağı Bölgelerinin Üretilmesi: Algoritma kovan çevresini araştırma uzayı varsayarak rasgele yiyecek kaynağı yerleri oluşturur. Bu yerleri oluştururken her bir parametrenin alt ve üst sınırları dikkate alınarak rasgele değerler üretilir. Yiyecek kaynağı bölgelerinin üretilmesi Eşitlik 2 ile gerçekleştirilir:

$$x_{ij} = x_j^{min} + rand(0,1)\left(x_j^{max} - x_j^{min}\right) \tag{2}$$

Bu eşitlikte $i = 1, \dots, SN$, $j = 1, \dots, D$ ve SN yiyecek kaynağı sayısı ve D optimize edilecek parametre sayısıdır. x_j^{min}, j. parametrenin alt sınırı, x_j^{max} ise j. parametrenin üst sınırıdır. Aynı zamanda başlangıç aşamasında her kaynağın geliştirilememe sayısını ifade eden $sayac_i$ (i. kaynağın geliştirilememe sayısı) sayaçları da sıfırlanmaktadır. Başlangıç yiyecek kaynağı bölgeleri üretildikten sonra görevli arı, gözcü arı ve kaşif arı adımları ile daha iyi kaynak bölgeleri bulunmaya çalışılır. ABC algoritmasında durdurma kriteri olarak kabul edilebilir hata değeri (err), maksimum çevrim sayısı (MCN) veya standart durdurma kriteri uygulanabilir.

2. Görevli Arıların Yiyecek Kaynağı Bölgelerine Gönderilmesi: Görevli arı biriminde her kaynağın civarında yeni bir yiyecek kaynağı Eşitlik 3 ile belirlenir:

$$v_{ij} = x_{ij} + \phi_{ij}\left(x_{ij} - x_{kj}\right) \tag{3}$$

$\vec{x}_i$ ile gösterilen her bir kaynak için, bu kaynağın yani çözümün tek bir parametresi (rastgele seçilen parametresi, j) değiştirilerek $\vec{x}_i$ komşuluğunda $\vec{v}_i$ kaynağı bulunur. Eşitlik 3'te j, $[1, D]$ aralığında rastgele üretilen tamsayıdır. Rastgele seçilen j parametresi değiştirilirken, yine rastgele seçilen $\vec{x}_k$ komşu çözümünün ($k \in \{1,2, \dots, SN\}$) j. parametresi ile mevcut kaynağın j. parametresinin farkları alınıp $[-1,1]$ arasında değer alan ϕ_{ij} sayısı ile ağırlıklandırıldıktan sonra mevcut kaynağın j. parametresine eklenmektedir.
Eşitlik sonucu yeni üretilen $\vec{v}_{ij}$ daha önceden belli olan parametre sınırlarını aşarsa Eşitlik 4 ile j. parametreye ait olan alt veya üst sınır değerlerine çekilir.

$$v_{ij} = \begin{cases} x_j^{min} & v_{ij} < x_j^{min} \\ x_j^{max} & v_{ij} > x_j^{max} \end{cases} \tag{4}$$

Sınırlar dahilinde üretilen v_i parametre vektörü yeni bir kaynağı temsil etmekte ve bunun kalitesi Eşitlik 5 ile hesaplanarak bir uygunluk değeri atanmaktadır.

$$fitness_i = \begin{cases} 1/(1+f_i) & f_i \geq 0 \\ 1 + abs(f_i) & f_i < 0 \end{cases} \tag{5}$$

Burada f_i, $\vec{v}_i$ kaynağının yani çözümünün maliyet değeridir. $\vec{x}_i$ ile $\vec{v}_i$ arasında nektar miktarlarına yani uygunluk değerlerine göre bir aç gözlü (greedy) seçim işlemi uygulanır. Yeni bulunan $\vec{v}_i$ çözümü daha iyi ise görevli arı hafızasından eski kaynağın yerini silerek $\vec{v}_i$ kaynağının yerini hafızaya alır. Aksi taktirde görevli arı $\vec{x}_i$ kaynağına gitmeye devam eder ve $\vec{x}_i$ çözümü geliştiremediği için $\vec{x}_i$ kaynağı ile ilgili geliştirememe sayacı ($sayac_i$) bir artar, yeni çözümün kabul edildiği durumda ise bu sayaç sıfırlanır.

3. Yiyecek Kaynaklarının Seçilebilme Olasılıklarının Hesaplanması: Görevli arı biriminde bütün kaynakların komşuluğunda araştırma tamamlandıktan sonra bulunan kaynakların nektar miktarlarıyla ilgili bilgiler gözcü arı birimi ile paylaşılır. Gözcü arılar yiyecek kaynaklarının nektar miktarıyla orantılı bir olasılık kullanarak bir kaynak seçer. Bu seçim sıralamaya dayalı, stokastik, rulet tekerleği, turnuva yöntemi ya da başka bir seçim yöntemlerinden istenilen ile yapılabilir. Temel ABC algoritması rulet tekerleği kullanarak seçim işlemini yapar. Tekerlekteki her bir dilimin açısı uygunluk değeriyle orantılıdır. Bir kaynağın uygunluk değerinin, tüm kaynakların uygunluk değeri toplamına oranı Eşitlik 6 ile bulunur ve bu oran nispi seçilme olasılığını verir.

$$p_i = \frac{fitness_i}{\sum_{i=1}^{SN} fitness_i} \tag{6}$$

Burada $fitness_i$, i. kaynağın kalitesini, SN yiyecek kaynağı sayısını göstermektedir. Hesaplanan bu olasılık değerinden bir kaynağın uygunluk miktarı arttıkça yani nektar miktarı arttıkça, bu kaynağı seçecek gözcü arı sayısı da dolayısıyla artacaktır.

4. Gözcü Arıların Yiyecek Kaynağı Bölgesi Seçmeleri: Olasılık değerleri hesaplandıktan sonra her bir kaynak için [0,1] aralığında rastgele sayı üretilir. Rulet tekerleğine göre seçim işleminde bu üretilen sayı, p_i değerinden küçükse gözcü arılar tıpkı görevli arıların yaptığı gibi Eşitlik 3'ü kullanarak bu kaynak bölgesinde yeni bir çözüm üretir. Yeni çözüm değerlendirilir ve kalitesi hesaplanır. Yeni çözüm ve eski çözüm uygunluklarına göre karşılaştırılır. Aç gözlü seleksiyon işlemi kullanılarak iyi olan çözüm seçilir. Yeni çözüm seçilirse çözüm geliştirememe sayacı ($sayac_i$) sıfırlanır. Eski çözümün daha iyi olduğuna karar verilirse mevcut çözüm korunur ve geliştirememe sayacı ($sayac_i$) bir artırılır. Bu aşama tüm gözcü arılar yiyecek kaynaklarına yönlendirilene kadar sürer.

5. Kâşif Arı Üretimi: Tüm görevli ve gözcü arı aşamaları bir çevrimi bitirdikten sonra çözüm geliştirememe sayaçlarına ($sayac_i$) bakılır. Bir kaynağın nektarının tükenip tükenmediği bu sayaçlardan anlaşılır. Bir kaynak için çözüm geliştirememe sayacı, kontrol parametresi olan *limit* değerini aştıysa, bu kaynak tükenmiş demektir ve bu kaynağın görevli arısı kendine başka bir kaynak bulmalıdır. Yani görevli arının o kaynakla ilgili görevi biter ve kâşif arı olarak kendine rasgele yeni bir kaynak aramaya başlar (Eşitlik 2). Temel ABC algoritmasında her çevrimde sadece bir kâşif arının çıkmasına izin verilir.

2.2. Ceza Fonksiyonu Kullanan ABC Algoritması

Bu çalışmada ceza fonksiyonu türlerinden statik ceza yapısı ABC algoritmasına entegre edilmiştir ve bu yönteme PEN-ABC adı verilmiştir. Statik ceza yapısında her problem için verilen sabit bir ceza faktörü bulunmaktadır. Temel ABC algoritmasından farklı olarak her çözüm için hesaplanan ihlal değeri yani sınırlama aşım değeri belli bir eşitlikle amaç fonksiyonuna eklenir. Eşitlik 7 ile yeni amaç fonksiyonu oluşturulur. Bu eşitlikte r_g ceza faktörü ve g_i sınırlama aşım miktarıdır [12]. r_g parametresinin farklı değerlerine göre PEN-ABC algoritmasının performansı ölçülmüştür.

$$f'(x) = f(x) + \sum_{i=1}^{m} \left(r_g \times max[0, g_i(x)]^2\right) \tag{7}$$

Temel ABC algoritmasından farklı olarak Karaboğa ve Akay [14] sınırlamalı optimizasyon problemlerinin çözümünde yakınsama hızını artırmak için komşu çözüm oluştururken değişim oranı (MR) adı verilen bir parametre ve kaşif arı aşamasında ise kaşif üretim periyodu (SPP) adı verilen başka bir parametre eklemiştir. Ayrıca sınırlamalı ABC algoritmasında popülasyonda kabul edilebilir bölge dışında çözümler olmasına izin verildiği için gözcü arı aşamasında bu çözümler için olasılık hesabında sınırlama aşım değerlerine göre olasılık atanmaktadır.

PEN-ABC algoritmasında yeni bir çözüm üretirken Eşitlik 8 kullanılır [14]. R_j, [0,1] aralığında uniform dağılımlı rasgele sayı, MR, [0,1] aralığında değer alan kontrol parametresidir. ϕ_{ij}, $[-1,1]$ aralığında uniform dağılımlı rasgele bir sayı ve $k \in \{1,2,\dots,SN\}$, i'den farklı rasgele bir indistir.

$$v_{ij} = \begin{cases} x_{ij} + \phi_{ij}(x_{ij} - x_{kj}), & R_j < MR\ ise \\ x_{ij}, & \text{aksi durumda} \end{cases} \quad (8)$$

PEN-ABC algoritmasında gözcü arıların kaynak seçerken kullandıkları olasılık hesabı Eşitlik 9'da verilmiştir [14]. $\vec{x}_i$ çözümünün, ihlal değeri $\vec{g}_i$ ve uygunluk miktarı $fitness_i$'dir.

$$p_i = \begin{cases} 0{,}5 + \left(\frac{fitness_i}{\sum_{i=1}^{SN} fitness_i}\right) * 0{,}5 & kabul\ edilebilir\ ise \\ \left(1 - \frac{g_i}{\sum_{i=1}^{SN} g_i}\right) * 0{,}5 & kabul\ edilebilir\ değilse \end{cases} \quad (9)$$

2.3. Stokastik Beslemeli ABC Algoritması

Runarsson ve Yao [5], ceza fonksiyonlarında kullanılan r_g ceza faktörünün belirlenme zorluğunu ortadan kaldırmak ve amaç fonksiyonu ile ceza fonksiyonu arasında dengeyi sağlamak için stokastik sıralama mantığını önermiştir. Sıralamayı yaparken ise kabarcık sıralama prosedürünü kullanmışlardır. Bu prosedür ABC algoritmasının çalışma mantığına göre uyarlanarak ABC algoritmasına entegre edilmiştir ve stokastik beslemeli ABC algoritması SR-ABC olarak isimlendirilmiştir.

SR-ABC algoritmasında işçi ve gözcü arı aşamalarında komşu çözümler temel ABC algoritmasındaki gibi oluşturulur. Burada seleksiyon işlemine geçildiğinde ise aç gözlü seleksiyon işlemi uygulanmaz. Algoritma 2 ile verilen stokastik sıralama prosedürü devreye girer. Öncelikle SN tane mevcut çözüm ve SN tane yeni oluşturulan komşu çözüm birleştirilir. Yani elde edilen $2 * SN$ çözüm stokastik sıralama prosedürüne göre amaç fonksiyonu ve ihlal değerine göre sıralanır. Bu sıralanan çözümler stokastik sıralama algoritmasındaki gibi doğrudan sıraya göre alınmaz. İçlerinden bazı çözümler sıralı olarak bazıları ise çeşitlilik olması açısından rasgele seçilir. Böylece bu seçilen SN tane çözümün içinde tamamen kabul edilebilir çözümler bulunmamış olur ve popülasyondaki farklılık (diversity) sağlanır. Seçilen SN çözüm için uygunluk değerleri hesaplanır ve ABC algoritmasının diğer adımları temel ABC algoritmasındaki ile aynı devam eder.

Hem işçi arı fazında hem de gözcü arı fazında stokastik yapı kullanılarak oluşturulan stokastik beslemeli ABC algoritmasında yakınsama hızını artırmak için komşu çözüm üretme aşamasında tek komşu yerine birkaç komşu bilgisinin kullanıldığı bir komşuluk mekanizması kullanılmıştır. Sadece görevli arı aşamasında, sadece gözcü arı aşamasında ya da hem görevli arı hem de gözcü arı aşamalarında Eşitlik 10 kullanılması durumları ayrı ayrı incelenmiştir. Bu eşitlikte $r_1 \neq r_2 \neq i$ olarak varsayılmaktadır.

$$v_{ij} = \begin{cases} x_{r_1 j} + F * (x_{ij} - x_{r_2 j}) & R_j < MR \\ x_{ij} & aksi\ durumda \end{cases} \quad (10)$$

2.4. Stokastik Deb Kuralları Kullanan ABC Algoritması

İki çözüm arasında seleksiyon işleminde Deb [9] aşağıdaki kuralları önermiştir:

- Kabul edilebilir bölgedeki çözüm kabul edilebilir bölgede olmayan çözüme tercih edilir.
- İki kabul edilebilir çözüm arasında amaç fonksiyonu değeri daha iyi olan çözüm tercih edilir.
- İki kabul edilebilir olmayan çözüm arasında sınırlama aşım değeri daha küçük olan çözüm tercih edilir.

Sınırlamalı optimizasyon problemlerinin çözümü için Karaboğa ve Akay [10] tarafından ABC algoritmasının seleksiyon aşamasında Deb'in kuralları kullanılmıştır. Seleksiyon aşamasında Deb'in kuralları kullanıldığında popülasyona büyük oranda kabul edilebilir çözümler dahil olmuştur. Popülasyona farklılık (diversity) katılması amacıyla Deb'in kurallarına stokastiklik eklenmiştir. Dolayısıyla stokastiklik ve Deb'in kurallarının bir arada kullanıldığı bir yöntem geliştirilmiştir. Bu yöntem DEB-SR-ABC olarak adlandırılmıştır. DEB-SR-ABC algoritmasının seleksiyon aşamasında Algoritma 3 kullanılır. Bu algoritma ile popülasyona çeşitlilik katılması amaçlanmıştır. Popülasyona sadece kabul edilebilir çözümlerin yerine kabul edilebilir olmayan çözümler P_f katsayısının etkisiyle dahil olur.

Algoritma 2. Stokastik besleme prosedürü

Mevcut çözümler dizisi ile komşu çözümler dizisini birleştir
for $i = 1$ to $SN * 2$ **do**
  for $j = 1$ to $SN*2$ **do**
    if $\phi_j = \phi_{j+1} = 0$ **then**
      if $f_j > f_{j+1}$ **then**
        Yer değiştir $(i, i+1)$
      end if
    end if
    if $\phi_j > \phi_{j+1}$ **then**
      Yer değiştir $(i, i+1)$
    end if
  end for
end for
SN tane çözümün bazılarını sıralı bazılarını rasgele seç

DEB-SR-ABC algoritmasının seleksiyon aşamasında kullanılan Algoritma 3 ile aşağıdaki kurallar önerilmektedir:

- Mevcut çözüm ve komşu çözümün her ikisi de kabul edilebilir bölgede ise veya rasgele üretilen random sayı P_f'den küçükse amaç fonksiyonu değerine göre seçim yapılır.
- Mevcut çözüm kabul edilebilir bölgede değil ancak komşu çözümün kabul edilebilir bölgede olduğu durumda, random sayının P_f'den küçük olması şartıyla komşu çözümün amaç fonksiyonu değeri daha iyiyse komşu çözüm seçilir. Komşu çözümün amaç fonksiyonu değeri daha iyi olmadığı halde sınırlama aşım değeri daha küçükse yine komşu çözüm seçilir. Aksi durumda mevcut çözüm korunur.
- İki çözüm de kabul edilebilir bölgede değilse sınırlama aşım değeri daha küçük olan seçilir.

Algoritma 3. DEB-SR-ABC algoritmasının seleksiyon aşaması

if $(violation_i \leq and\ violation_j \leq 0)\ or\ random < P_f$ **then**
  if $fitness_j > fitness_i$ **then**
    Komşu çözümü seç
  else
    Mevcut çözümü tut.
  end if
else
  if $(violation_i > 0\ and\ violation_j \leq 0)$ **then**
    if $random < P_f$ **then**
      if $fitness_j > fitness_i$ **then**
        Komşu çözümü seç
      end if
    else
      if $violation_j < violation_i$ **then**
        Komşu çözümü seç
      else
        Mevcut çözümü tut.
      end if
    end if
  end if
else
  if $(violation_i > 0\ and\ violation_j > 0)$ **then**
    if $violation_j < violation_i$ **then**
      Komşu çözümü seç
    else
      Mevcut çözümü tut.
    end if
  end if
end if

3. Bulgular

Bu bölümde bu çalışmada önerilen yeni yöntemlerle ilgili deneysel çalışmalara yer verilmiştir. Öncelikle PEN-ABC, SR-ABC ve DEB-SR-ABC algoritmaları test problemleri aracılığıyla değerlendirilmiştir. Geliştirilen üç yöntem sonuçları literatürdeki stokastik sıralama (SR) metodu [5], geliştirilmiş stokastik sıralama (ISR) metodu [6], aşırı ceza yaklaşımı (OPA) [6], genetik algoritma (GA) [7], basit çok üyeli evrimsel strateji (SMES) [7], diferansiyel gelişim (DE) [10], parçacık sürü optimizasyonu (PSO) algoritması [8] ve ABC algoritması [10] ile karşılaştırılmıştır.

Algoritma, Visual Studio 2017'de C#.NET programlama dili ile kodlanmıştır. Deneyler Intel (R) Core (TM) i5-3210 M 2.5 GHz işlemci, 6 GB RAM ve Windows 10 işletim sistemi bulunan bilgisayar ile yapılmıştır.

3.1. Test Problemleri

Geliştirilen yöntemleri test etmek amacıyla Liang ve diğ. [15] tarafından önerilen literatürde sıklıkla kullanılan 13 sınırlamalı optimizasyon problemi kullanılmıştır. Her test probleminin temel özellikleri Tablo 1'de özetlenmiştir. Tablo 1'de D problemdeki değişken sayısı, "NI" nonlineer eşitsizlik sayısı, "LI" lineer eşitsizlik sayısı, "NE" nonlineer eşitlik sayısı ve "LE" ise lineer eşitlik sayısıdır. "α" aktif sınırlamaların sayısı ve "ρ" kabul edilebilir bölgenin yaklaşık alanıdır. $\rho = |F|/|S|$ şeklinde hesaplanır ve burada $|S|$ araştırma uzayı ve $|F|$ ise kabul edilebilir bölgedir. g05, g07 ve g13 problemlerinin kabul edilebilir bölge oranlar açısından çok dar bir alana sahip olduğu görülmektedir.

Tablo 1. Çalışmada kullanılan test problemlerinin karakteristikleri

Problem	Min./Max.	D	Problemin Türü	ρ	LI	NI	LE	NE	α
g01	Min.	13	Kuadratik	%0.0003	9	0	0	0	6
g02	Max.	20	Nonlineer	%99.9973	0	2	0	0	1
g03	Max.	10	Polinomial	%0.0026	0	0	0	1	1
g04	Min	5	Kuadratik	%27.0079	0	6	0	0	2
g05	Min	4	Kübik	%0.0000	2	0	0	3	3
g06	Min	2	Kübik	%0.0057	0	2	0	0	2
g07	Min	10	Kuadratik	%0.0000	3	5	0	0	6
g08	Max.	2	Nonlineer	%0.8581	0	2	0	0	0
g09	Min	7	Polinomial	%0.5199	0	4	0	0	2
g10	Min	8	Lineer	%0.0020	3	3	0	0	6
g11	Min	2	Kuadratik	%0.0973	0	0	0	1	1
g12	Max.	3	Kuadratik	%4.7697	0	1	0	0	0
g13	Min	5	Nonlineer	%0.0000	0	0	0	3	3

3.2. Kontrol Parametreleri ve Değerleri

Karaboğa ve Akay [10], deneysel çalışmalarında sabit parametre değerleri kullanmışlardır. Buna göre koloni büyüklüğü $SN = 40$, maksimum çevrim sayısı $MCN = 6000$, değişim oranı $MR = 0.8$, D probleme ait parametre sayısı olmak üzere $limit = 0.5 * SN * D$ ve $SPP = 0.5 * SN * D$'dir. Eşitlik sınırlamalarını eşitsizlik sınırlamalarına çevirirken kullanılan $\varepsilon = 0.001$ olarak işleme katılmıştır. Her bir deney rasgele popülasyonlarla 30 kez koşulmuştur.

PEN-ABC algoritmasını denerken r_g parametresinin beş farklı değeri için performans değerlendirmesi yapılmıştır. $r_g = \{1, 5, 10, 20, 50\}$ değerleri verilmiştir. SR-ABC algoritmasının ilk olarak performansı incelenmiştir. Sonrasında ise sınırlamaları aşan parametreler için ceza fonksiyonlarında olduğu gibi ihlal değerleri amaç fonksiyonuna eklenerek performansına bakılmıştır. Burada $r_g = 1$ alınmıştır. DE komşuluk mekanizması kullanılarak yapılan performans incelemesinde is F ölçekleme faktörü 0.8 olarak kabul edilmiştir. Yani toplamda üç farklı şekilde SR-ABC algoritması değerlendirilmiştir. DEB-SR-ABC algoritmasında ise temel ABC algoritmasındaki parametrelere ek olarak kullanılan P_f parametresi Runarsson ve Yao'nun [5] önerdiği şekilde $P_f = 0.45$ olarak verilmiştir.

SR [5], gelişimsel stratejide stokastik sıralama mantığına dayanan bir sınırlama ele alış metodu kullanmıştır. $(30, 200) - ES$ kullanılmıştır ve tüm koşmalar 1750 iterasyondan sonra durdurulmuştur. Böylelikle 200x1750=350000 amaç fonksiyonu değerlendirmesi yapılmış olur. Tüm eşitlik sınırlamalarını eşitsizlik sınırlamalarına çevirirken $|h_j| \in \varepsilon$ olmak üzere $\varepsilon = 0.0001$ seçilmiştir.

ISR [6], 875 iterasyon boyunca $(60, 400) - ES$ kullanmıştır. Dolayısıyla $400x875 = 350000$ amaç fonksiyonu değerlendirilmiş olur. Burada SR'den farklı olarak adım büyüklüğünü ölçeklemek için γ kontrol parametresi kullanılmıştır. Adım büyüklüğündeki azalma yaklaşık 0.85'tir.

OPA [6], ceza yaklaşımına dayanan sınırlama ele alış metoduyla $(60, 400) - ES$ kullanmıştır. g12 hariç bütün koşmalar 875 iterasyon sonra durdurulmuştur. g12 ise 87 iterasyonda durdurulmuştur. Fonksiyonun değerlendirme sayısı toplamda $400x875 = 350000$'dir.

GA [7], sınırlama ele alış metotlarından Deb'in kurallarını kullanmıştır. Eşitlik sınırlamalarını eşitsizlik sınırlamalarına çevirirken ε değeri dinamik olarak belirlenir. Koloni büyüklüğü 200, maksimum iterasyon sayısı 1200, çaprazlama oranı 0.8 ve değişim oranı 0.6'dır. Toplamda $200x1200 = 240000$ amaç fonksiyonu değerlendirmesi yapılmıştır.

SMES [7], $(100, 300) - ES$ ve sınırlamaları ele almak için Deb'in kurallarını kullanır. İterasyon sayısı 800'dür. Toplam amaç fonksiyonu değerlendirme sayısı $300x800 = 240000$'dir.

DE algoritması [10], sınırlamaları ele almak için Deb'in kurallarını kullanmıştır. Koloni büyüklüğü 40, çaprazlama oranı (CR) 0.9, ölçekleme faktörü (F) 0.5 ve maksimum iterasyon sayısı 6000'dir. Dolayısıyla amaç fonksiyonu değerlendirme sayısı $40x6000 = 240000$'dir.

PSO algoritması [8], sınırlamaları ele alış metotlarından Deb'in kurallarını kullanır. Sürü (popülasyon) büyüklüğü 50 ve iterasyon sayısı 7000'dir. Toplamda amaç fonksiyonu değerlendirme sayısı $50x7000 = 350000$'dir. Sosyal ve bilişsel bileşenlerin her ikisi de 1 olarak alınmıştır. Eylemsizlik ağırlığı (inertia weight) $[0.5 - 1]$ aralığında uniform dağılmış rasgele bir reel sayıdır. Eşitlik sınırlamalarını eşitsizlik sınırlamalarına çevirirken $|h_j| \in \varepsilon$ 'de kullanılan $\varepsilon = 0.001$'dir.

ABC algoritması [10], sınırlamaları ele alırken Deb'in kurallarını kullanmıştır. Koloni büyüklüğü (SN) 40 değişim oranı (MR) 0.8, maksimum çevrim sayısı (MCN) 6000'dir. D probleme ait parametre sayısı olmak üzere $limit = 0.5 * SN * D$ ve $SPP = 0.5 * SN * D$'dir. Amaç fonksiyonu değerlendirme sayısı $40x6000 = 240000$'dir. Deneyler rasgele popülasyonlarla 30 kez tekrarlanmıştır.

3.3. İstatistiksel Sonuçlar

Sınırlamalı optimizasyon problemlerinin çözümü için geliştirilen yöntemler literatürdeki diğer algoritmalarla karşılaştırılmıştır. Tablo 2 ve Tablo 3'te problemler için elde edilen en iyi değerler ve ortalama değerler karşılaştırmalı olarak analiz edilmiştir. Her bir problem için bulunan en iyi sonuç koyu olarak yazılmıştır. Tablolarda yer alan NA kabul edilebilir çözüm bulunamayan değerleri göstermektedir.

PEN-ABC algoritmasının r_g parametresinin farklı değerleriyle 30 koşma sonucunda bulunan sonuçları tablolarda sunulmuştur. Stokastik beslemeli ABC algoritması yedi farklı şekilde incelenmiştir. Sadece işçi arı fazına stokastik yapı entegre edilerek (Employed-SR), sadece gözcü arı fazına stokastik yapı entegre edilerek (Onlooker-SR), hem işçi arı hem de gözcü arı fazına stokastik yapı entegre edilerek (Employed-Onlooker-SR) ve hem işçi arı hem de gözcü arı fazına stokastik yapı entegre edilmesine ek olarak sınırlama aşım değerlerinin de amaç fonksiyonuna eklenme işlemi uygulanarak (Employed-Onlooker-SR/rg) analizler yapılmıştır. Ayrıca ABC algoritmasının işçi ve gözcü arı fazlarında stokastik yapı kullanılmasına ek olarak komşu çözüm üretme aşamasında DE algoritmasının komşu üretme mekanizması kullanılarak üç yöntem daha denenmiştir. Burada DE komşu üretme mekanizması sadece işçi arı fazında kullanıldığında (Employed-DE), sadece gözcü arı fazında kullanıldığında (Onlooker-DE) ve hem işçi arı hem de gözcü arı fazında kullanıldığında (Employed-onlooker-DE) elde edilen sonuçlar incelenmiştir. DEB-SR-ABC algoritması ile 30 koşma sonucu elde edilen sonuçlar da tablonun sonuna eklenmiştir.

Tablo 2. SR, ISR, OPA, GA, SMES, PSO, DE, ABC, PEN-ABC. SR-ABC ve DEB-SR-ABC algoritmalarının 13 test fonksiyonu için elde edilen sonuçların en iyi değerleri. (-) kabul edilebilir çözümün bulunamadığı anlamına gelir.

Yöntem	g01	g02	g03	g04	g05	g06	g07	g08	g09	g10	g11	g12	g13
Optimal	**-15.000**	**0.803619**	**1.000**	**-30665.539**	**5126.498**	**-6961.814**	**24.306**	**0.095825**	**680.630**	**7049.25**	**0.75**	**1.000**	**0.053950**
SR	**-15.000**	0.803515	**1.000**	**-30665.539**	**5126.497**	**-6961.814**	24.307	**0.095825**	**680.630**	7054.316	**0.750**	**1.000**	**0.053957**
ISR	**-15.000**	**0.803619**	1.001	**-30665.539**	**5126.497**	**-6961.814**	**24.306**	**0.095825**	**680.630**	**7049.248**	**0.750**	**1.000**	0.053942
OPA	**-15.000**	**0.803619**	0.747	**-30665.539**	**5126.497**	**-6961.814**	**24.306**	**0.095825**	**680.630**	**7049.248**	**0.750**	**1.000**	0.447118
GA	-14.440	0.796231	0.990	-30626.053	-	-6952.472	31.097	**0.095825**	685.994	9079.770	**0.75**	**1.000**	0.134057
SMES	**-15.000**	0.803601	**1.000**	**-30665.539**	5126.599	**-6961.814**	24.327	**0.095825**	680.632	7051.903	**0.75**	**1.000**	0.053986
PSO	**-15.000**	0.669158	0.993930	**-30665.539**	5126.484	**-6961.814**	24.370153	**0.095825**	**680.630**	7049.381	0.749	**1.000**	0.085655
DE	**-15.000**	0.472	**1.000**	**-30665.539**	5126.484	-6954.434	**24.306**	**0.095825**	**680.630**	**7049.248**	0.752	**1.000**	0.385
ABC	**-15.000**	0.803598	**1.000**	**-30665.539**	5126.484	**-6961.814**	24.330	**0.095825**	680.634	7053.904	**0.750**	**1.000**	0.760
rg=1 (PEN-ABC)	-14.414	0.802638	0.8830	-30135.576	5121.079	-7950.263	25.4577	0.095823	680.807	2101.46	0.739	**1.000**	0.05611
rg=5 (PEN-ABC)	-14.915	0.803602	0.3604	-30127.391	5129.407	-7781.586	24.7538	**0.095825**	680.653	2107.29	0.747	**1.000**	0.09534
rg=10 (PEN-ABC)	-14.920	0.803607	0.7294	-30241.728	5126.855	-7848.131	24.5334	**0.095825**	680.640	2114.81	0.748	**1.000**	0.14488
rg=20 (PEN-ABC)	-14.969	0.803591	0.8244	-30466.900	5126.534	-7464.222	24.4382	**0.095825**	680.633	2128.76	0.749	**1.000**	0.40804
rg=50 (PEN-ABC)	-14.986	0.803599	0.9752	-30459.850	5126.528	-6827.402	24.3659	**0.095825**	680.633	2171.19	0.749	**1.000**	0.56397
Employed-SR	**-15.000**	0.781265	0.9960	**-30665.539**	5133.786	**-6961.814**	24.6819	**0.095825**	680.655	7146.448	**0.750**	**1.000**	0.27329
Onlooker-SR	-14.999	0.794766	0.9611	**-30665.539**	5127.795	**-6961.814**	25.1593	**0.095825**	680.677	7931.900	**0.750**	**1.000**	0.06732
Emp-Onl-SR	-14.999	0.788303	1.0032	**-30665.539**	5128.492	-7950.961	24.6205	**0.095825**	680.681	7113.398	**0.750**	**1.000**	0.38624
Emp-Onl-SR/rg	-14.999	0.791442	0.9998	**-30665.539**	5127.042	-7950.205	24.7066	**0.095825**	680.656	7062.924	**0.750**	**1.000**	0.56700
Emp-DE (SR)	-14.988	0.599782	0.5547	-30665.291	5129.668	-7950.962	319.284	**0.095825**	682.563	7794.773	**0.750**	0.999	0.50870
Onl-DE (SR)	-14.999	0.691965	0.6394	-30665.480	5139.132	-7973	300.470	**0.095825**	681.994	8046.979	**0.750**	0.999	0.46625
Emp-Onl-DE (SR)	-14.971	0.617807	0.3250	-30665.006	5167.000	-7950.963	410.585	**0.095825**	688.752	8269.454	**0.750**	0.999	0.26848
DEB-SR-ABC	**-15.000**	0.803591	1.004	**-30665.539**	4986.987	-7661.009	243.438	**0.095825**	680.633	2100	0.749	**1.000**	0.057301

Tablo 3. SR, ISR, OPA, GA, SMES, PSO, DE, ABC, PEN-ABC, SR-ABC ve DEB-SR-ABC algoritmalarının 13 test fonksiyonu için elde edilen sonuçların ortalama değerleri. (-) kabul edilebilir çözümün bulunamadığı anlamına gelir.

Yöntem	g01	g02	g03	g04	g05	g06	g07	g08	g09	g10	g11	g12	g13
Optimal	**-15.000**	**0.803619**	**1.000**	**-30665.539**	**5126.498**	**-6961.814**	**24.306**	**0.095825**	**680.630**	**7049.25**	**0.75**	**1.000**	**0.053950**
SR	**-15.000**	0.781975	**1.000**	**-30665.539**	5128.881	-6875.940	24.374	**0.095825**	680.656	7559.192	**0.750**	**1.000**	0.057006
ISR	**-15.000**	0.782725	1.001	**-30665.539**	5126.497	**-6961.814**	**24.306**	**0.095825**	**680.630**	**7049.250**	**0.750**	**1.000**	0.066770
OPA	**-15.000**	0.776283	0.257	**-30665.539**	5268.610	**-6961.814**	24.307	**0.095825**	**680.630**	7049.248	0.755	0.999889	0.964323
GA	-14.236	0.788588	0.976	-30590.455	-	-6872.204	34.980	0.095799	692.064	10003.225	**0.75**	**1.000**	-
SMES	**-15.000**	0.785238	**1.000**	**-30665.539**	5174.492	-6961.284	24.475	**0.095825**	680.643	7253.047	**0.75**	**1.000**	0.166385
PSO	-14.710	0.419960	0.764813	**-30665.539**	5135.973	**-6961.814**	32.407	**0.095825**	**680.630**	7205.5	0.749	0.998875	0.569358
DE	-14.555	0.665	**1.000**	**-30665.539**	5264.270	-	24.310	**0.095825**	**680.630**	7147.334	0.901	**1.000**	0.872
ABC	**-15.000**	0.792412	**1.000**	**-30665.539**	5185.714	-6961.813	24.473	**0.095825**	680.640	7224.407	**0.750**	**1.000**	0.968
rg=1 (PEN-ABC	-12.120	0.787720	0.0082	-29267.342	5189.360	-5840.023	31.673	0.094926	680.915	11098.665	0.927	**1.000**	0.45962
rg=5 (PEN-ABC)	-14.604	0.789204	0.0240	-29210.471	5305.625	-4984.099	25.546	0.095629	680.693	8166.607	0.852	**1.000**	0.79161
rg=10 (PEN-ABC)	-14.794	0.791964	0.1141	-29388.250	5312.793	-5515.717	25.007	0.095783	680.655	9386.364	0.783	**1.000**	0.75960
rg=20 (PEN-ABC)	-14.858	0.786921	0.2189	-29385.895	5201.197	-5149.604	24.767	0.095799	680.648	8098.925	0.749	**1.000**	0.83561
rg=50 (PEN-ABC)	-14.948	0.793056	0.6737	-29337.438	5316.717	-5523.459	24.591	0.095629	680.642	9044.092	0.749	**1.000**	0.88431
Employed-SR	-13.766	0.721954	0.7576	-30635.714	5495.967	**-6961.814**	29.3491	**0.095825**	680.755	7820.773	**0.75**	**1.000**	1.24207
Onlooker-SR	-13.099	0.717791	0.6143	**-30665.539**	5913.843	**-6961.814**	27.2965	**0.095825**	680.958	10464.995	0.753	**1.000**	0.69821
Emp-Onl-SR	-12.679	0.623135	0.4954	-30653.502	5451.812	-6992.820	57.0992	**0.095825**	681.416	10395.296	0.832	0.999	0.92376
Emp-Onl-SR/rg	-11.916	0.621560	0.6898	-30662.664	5616.404	-7027.668	98.4280	**0.095825**	681.630	10620.716	0.776	0.999	1.04602
Emp-DE (SR)	-10.399	0.466272	0.0525	-30632.953	5580.917	-6891.767	315.393	0.095824	1017.019	12152.608	0.840	0.999	1.50786
Onl-DE (SR)	-11.631	0.487656	0.0734	-30588.714	5656.963	-6980.580	483.881	**0.095825**	1014.704	11304.605	0.824	0.999	1.73945
Emp-Onl-DE (SR)	-10.719	0.432700	0.0108	-30661.255	5701.441	-6440.205	376.612	0.095817	1033.509	14177.442	0.832	0.999	1.21924
DEB-SR-ABC	**-15.000**	0.791671	1.002	**-30665.539**	5559.641	-6344.100	24.513	**0.095825**	680.641	11091.682	**0.75**	**1.000**	0.59645

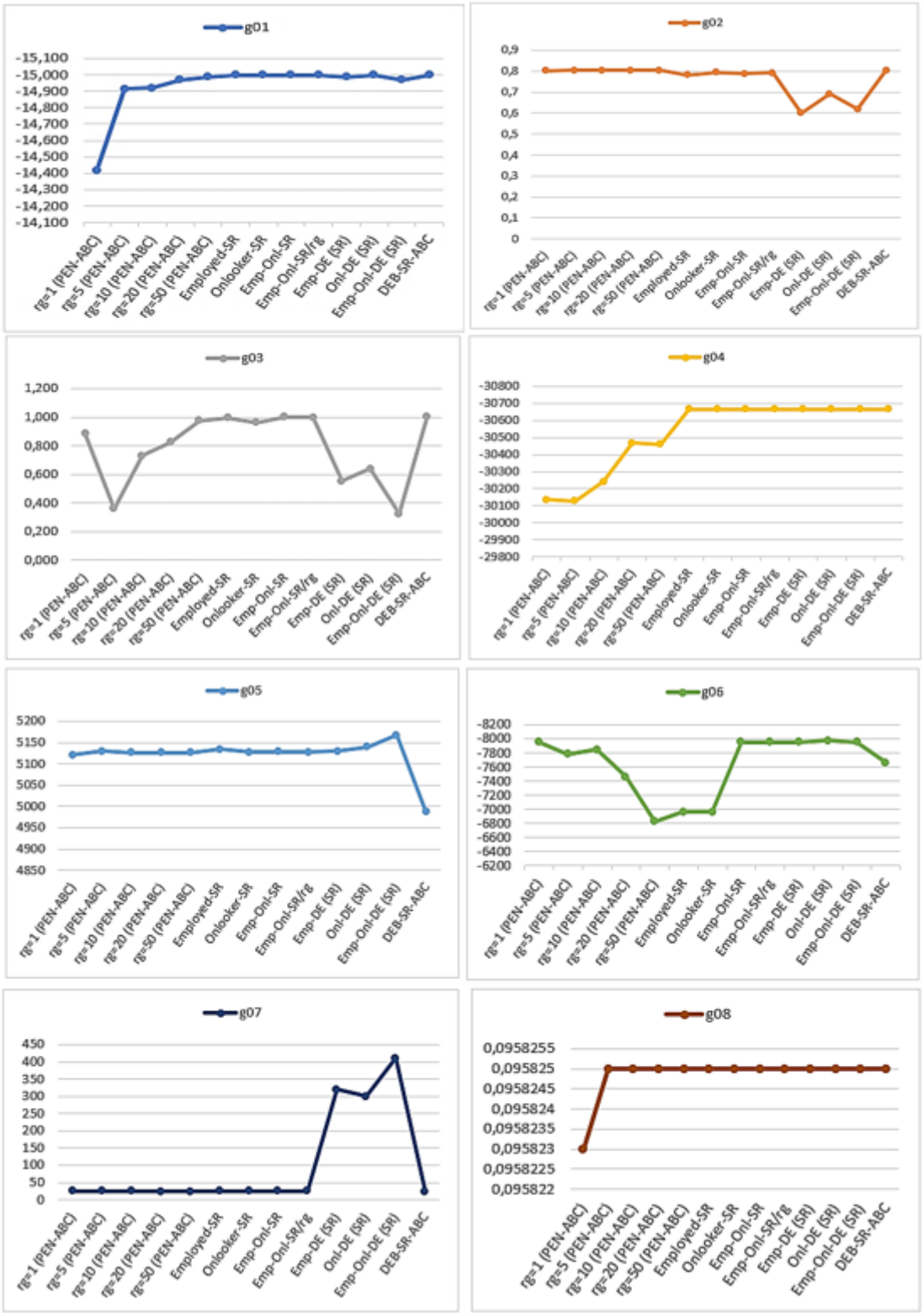
PEN-ABC algoritmasının Tablo 2 ve Tablo 3'te sunulan sonuçlarına göre her bir problem üzerinde karşılaştırma yapılmıştır. g01 probleminde optimal değer bulunamasa da ortalama değerler incelendiğinde GA, PSO ve DE algoritmalarından daha iyi olduğu söylenebilir. g02 probleminde optimal değere ulaşılamasa da r_g'nin tüm değerlerinde GA, PSO ve DE algoritmalarından daha iyi sonuç alınmıştır. $r_g = \{5, 10\}$ iken SR, SMES ve ABC algoritmalarından daha iyidir. Ortalama değerlere bakıldığında r_g parametresi arttıkça ortalama iyileşmiş ve diğer algoritmalardan daha iyi bir değere ulaşmıştır. g03 probleminde $r_g = 50$ iken diğer r_g değerlerinden daha iyi sonuç almış ancak optimale ulaşamamıştır. Hem en iyi sonucuyla hem de ortalama sonucuyla sadece OPA'dan daha iyidir. PEN-ABC algoritması g04 probleminde performans diğer algoritmaların gerisinde kalmıştır. g05 probleminde SMES, PSO, DE ve ABC algoritmalarıyla en iyi sonuçlara göre benzer sonuçlar alınmıştır. Ortalama sonuçlara göre OPA ve DE algoritmalarından daha iyidir. GA'dan ise daha iyi bir performansa sahiptir. g06 probleminde diğer algoritmalardan düşük performansa sahiptir. $r_g = 50$ parametresiyle PEN-ABC algoritması g07 probleminde en iyi sonucunu elde etmiş ve GA ve PSO algoritmalarından daha iyi sonuç almıştır ve ortalama sonuçlar da daha başarılıdır. g08 probleminde diğer algoritmalar gibi optimal değere ulaşmıştır. g09 probleminde $r_g = 20$ ve $r_g = 50$ değerleriyle optimale en yakın sonuç elde edilmiştir ve bu sonuçlarla GA'dan daha iyi, SMES ve ABC algoritması ile yakın performans göstermiştir. g10 probleminde diğer algoritmalardan daha kötü sonuçlar alınmıştır. g11 probleminde $r_g = \{20, 50\}$ iken optimal değer bulunmuş ve diğer algoritmalarla (DE algoritması hariç) benzer performans göstermiştir. Ortalama değerler incelendiğinde DE algoritmasından daha iyidir. g12 probleminde optimal değer bulunmuş ve diğer algoritmalarla benzer performans göstermiştir. g13 probleminde $r_g = 1$ parametresiyle en iyi sonucunu bulmuştur ve bu sonuçla OPA, GA, PSO, DE ve ABC algoritmalarından daha iyidir.

SR-ABC algoritmasının Tablo 2 ve Tablo 3'te verilen sonuçlarına göre her bir problem için yöntemlerin performansına göre karşılaştırma yapılmıştır. g01 probleminde Employed-SR optimal değeri bulurken diğerleri optimale yakın değer bulmuştur. Ancak ortalama değerler incelendiğinde SR-ABC algoritmasının diğer algoritmalara göre daha düşük bir performansa sahip olduğu görülmektedir. g02 probleminde en iyi sonuçlar Onlooker-SR ve Employed-Onlooker-SR/rg yöntemleriyle alınmıştır. Bu sonuçlar ile sadece PSO ve DE algoritmalarından performans olarak üsttedir. g03 probleminde Employed-SR, Employed-Onlooker-SR ve Employed-Onlooker-SR/rg yöntemleri optimale yakın sonuçlar bularak OPA, GA ve PSO algoritmalarından daha iyi performans göstermiştir. Bu problemde ortalama ve standart sapma değerlerine bakıldığında kararsız yapıda olduğu söylenebilir. g04 probleminde Employed-SR, Onlooker-SR, Employed-Onlooker-SR ve Employed-Onlooker-SR/rg yöntemleriyle optimal değer yakalanmıştır. Diğer algoritmalarla (GA hariç) benzer performans gösterildiği söylenebilir. g05 probleminde Onlooker-SR, Employed-Onlooker-SR ve Employed-Onlooker-SR/rg yöntemleriyle optimale yakın sonuçlar alınmıştır. Ancak diğer algoritmalara nazaran (GA hariç) daha düşük performansa sahiptirler. g06 probleminde Employed-SR ve Onlooker-SR yöntemleri optimal değere ulaşmıştır. GA ve DE algoritmaları hariç diğer algoritmalarla benzer performans elde edilmiştir. g07 probleminde Employed-SR, Employed-Onlooker-SR ve Employed-Onlooker-SR/rg yöntemleri optimale en yakın sonuçlar elde ederek sadece GA'dan iyi performans göstermiştir. g08 probleminde tüm yöntemlerle optimale ulaşarak diğer algoritmalarla benzer performans elde edilmiştir. g09 probleminde Employed-SR, Onlooker-SR, Employed-Onlooker-SR ve Employed-Onlooker-SR/rg yöntemleri optimale yakın değerler bulmuştur. Ancak diğer algoritmalar (GA haricinde) kadar başarılı değildir. g10 probleminde Employed-Onl-SR/rg yöntemi optimale en yakın değer bulmuştur. Sadece GA'dan daha iyi sonuç alınmıştır. g11 ve g12 problemlerinde optimal değere ulaşılmıştır ve tüm yöntemlerin diğer algoritmalarla benzer performansa sahip olduğu söylenebilir. g13 probleminde ise Onlooker-SR yöntemi ile optimale daha yakın sonuç alınmıştır ve bu sonuçla OPA, GA, PSO ve DE algoritmalarından daha iyidir. Genel itibariyle SR-ABC algoritmasında uygulanan yöntemlerden Employed-SR, Onlooker-SR, Employed-Onlooker-SR ve Employed-Onlooker-SR/rg optimal/optimale yakın değere ulaşmış ve literatürdeki diğer algoritmalara yakın sonuçlar elde etmiştir.

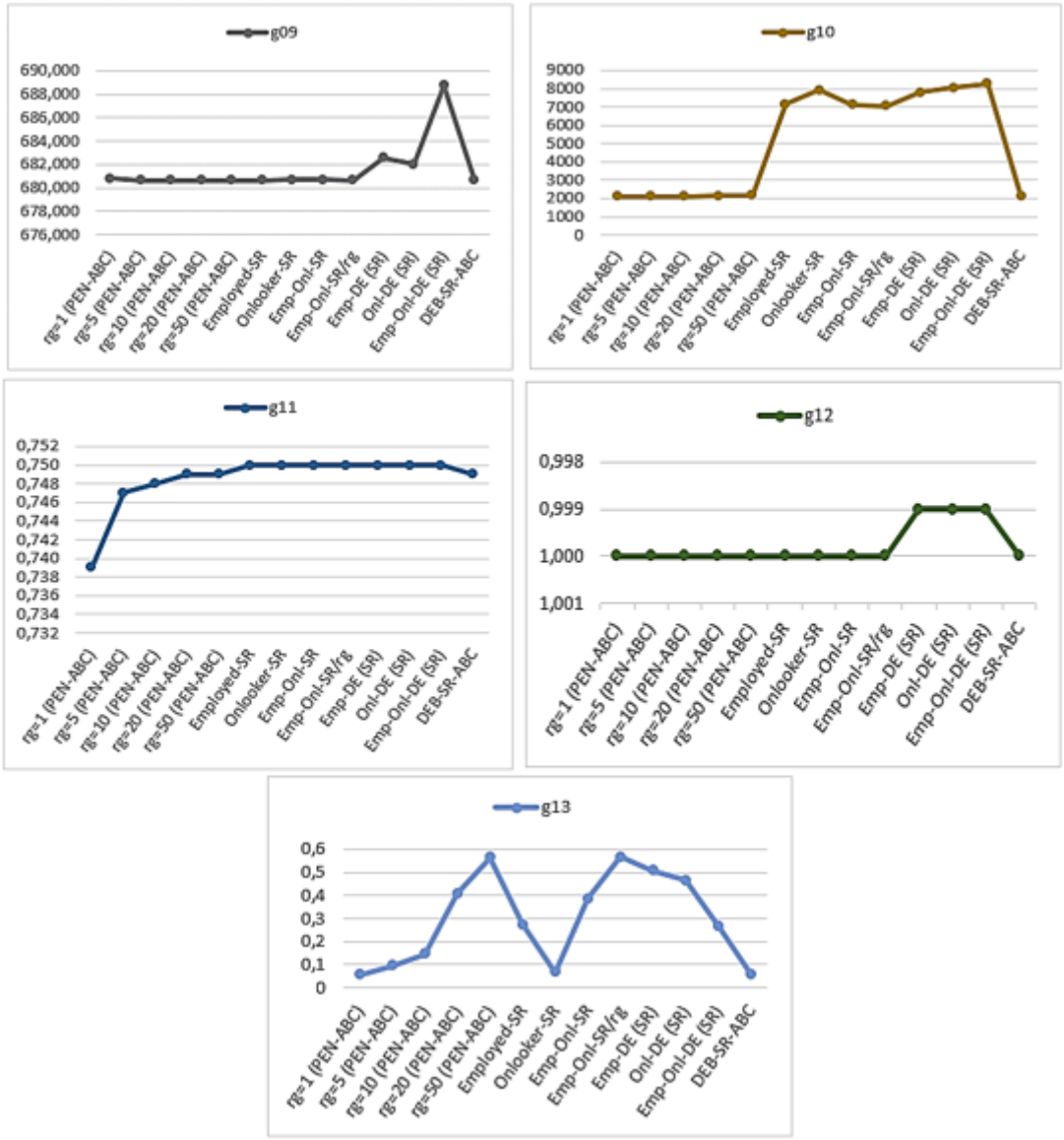
Sunulan sonuçlara göre DEB-SR-ABC algoritmasının optimale daha yakın sonuçlar elde ettiği görülmektedir. DEB-SR-ABC algoritması g01 probleminde optimali bulup diğer algoritmalarla benzer performans göstermiştir. En iyi sonuca göre sadece GA'dan daha iyiyken ortalama sonuçlara bakıldığında GA, PSO ve DE algoritmalarından daha iyidir. g02 probleminde optimale çok yakın sonuç bularak DE ve PSO algoritmalarından daha iyi sonuç alırken diğer algoritmalarla benzerdir. g03 probleminde optimal değer bulunup yalnız OPA'dan üst bir performans göstermiştir, diğerleriyle ise benzerdir. g04 probleminde optimal değer bulunarak diğer algoritmalarla (GA hariç) benzer performans elde edilmiştir. g05 ve g06 problemlerinde diğer algoritmalardan düşük performans göstermiştir. g07 probleminde optimale yakın sonuç alınmıştır. GA'dan iyi, SMES, PSO ve ABC ile benzerdir. g08 probleminde diğer algoritmalarda olduğu gibi optimal değer bulunmuştur. g09 probleminde optimale çok yakın sonuç bulunmuştur ve diğer algoritmalarla (GA hariç) performans olarak yakındır. DEB-SR-ABC algoritması g10 probleminde optimalden uzak sonuç alarak diğer algoritmalardan düşük performans göstermiştir. g11 ve g12 problemlerinde optimal değer bulunmuş ve diğer algoritmalarla benzer performans

göstermiştir. g13 probleminde DEB-SR-ABC algoritması optimal değeri bulamasa da OPA, GA, PSO, DE ve ABC algoritmalarından iyi performansa sahiptir.

Şekil 1 ve Şekil 2'de 13 test problemi için bulunan en iyi sonuçların grafikleri sunulmuştur. Grafiklerde x-ekseni denenen yöntemi, y-ekseni ise elde edilen sonucu göstermektedir.



Şekil 1. g01-g08 problemleri için bulunan en iyi sonuçların grafikleri



Şekil 2. g09-g13 problemleri için bulunan en iyi sonuçların grafikleri

4. Tartışma ve Sonuç

Tablo 4'te literatürde sıklıkla kullanılan 13 test probleminin çözümünde üç yöntem (PEN-ABC, SR-ABC ve DEB-SR-ABC) arasından en iyi sonuca ulaşan yöntemler listelenmiştir. Bu çizelgede PEN-ABC algoritması için hangi parametre (r_g) değeriyle en iyi sonuca ulaştığı, SR-ABC algoritması için kullanılan hangi yöntemle en iyi sonuca ulaştığı ve DEB-SR-ABC algoritmasında ise ulaşılan sonuç bilgisi yer almaktadır. g01 probleminin çözümü için Employed-SR ve DEB-SR-ABC yöntemleri kullanılabilir. g02 probleminin çözümü için PEN-ABC ($r_g = \{5{,}10{,}50\}$) ve DEB-SR-ABC yöntemi kullanılabilir. g03 problemi Employed-Onlooker-SR, Employed-Onlooker-SR/rg ve DEB-SR-ABC yöntemleri ile çözülebilir. g04 problemi Employed-SR, Onlooker-SR, Employed-Onlooker-SR, Employed-Onlooker-SR/rg ve DEB-SR-ABC yöntemleri ile çözülebilir. g05 probleminde PEN-ABC ($r_g = \{10, 20, 50\}$) yöntemiyle optimale yakın sonuç alınabilir. g06 probleminde optimale ulaşan yöntemler Employed-SR ve Onlooker-SR'dir. g07 problemi optimale en yakın değer bulan PEN-ABC ($r_g = 50$) ve DEB-SR-ABC yöntemleri ile çözülebilir. g08 probleminde üç algoritma (PEN-ABC, SR-ABC ve DEB-SR-ABC) ile çözüme ulaşılabilir. g09 probleminin çözümünde PEN-ABC ($r_g = \{20, 50\}$) ve DEB-SR-ABC yöntemleri uygulanabilir. g10 probleminde yalnızca Employed-Onlooker-SR/rg yöntemiyle optimale yakın sonuç alınabilir. g11 ve g12 problemlerinde üç algoritma ile optimal/optimale yakın değere ulaşılmıştır. g13 probleminde PEN-ABC ($r_g = 1$) ve DEB-SR-ABC yöntemleriyle optimale yakın sonuçlar elde edilmiştir.

Tablo 4. Problemlerin çözümünde en iyi sonuca ulaşan yöntemler

Fonksiyon / Optimal	PEN-ABC		SR-ABC		DEB-SR-ABC	
	En iyi sonuç (yöntem)	Ortalama ± std. sapma	En iyi sonuç (yöntem)	Ortalama ± std. sapma	En iyi sonuç	Ortalama± std. sapma
g01 -15.000	-14.986 (rg=50)	-14.948 ± 0.023	-15.000 (Employed-SR)	-13.766 ± 1.546	-15.000	-15.000 ± 0
g02 0.803619	0.803607 (rg=10)	0.791964 ± 0.014	0.794766 (Onlooker-SR)	0.717791 ± 0.098	0.8035917	0.7916718 ± 0.010
g03 1	0.9752 (rg=50)	0.6737 ± 0.236	0.9998 (Employed-Onlooker-SR/rg)	0.6898 ± 0.461	1.004	1.002 ± 0.003
g04 -30665.539	-30459.850 (rg=50)	-29337.438 ± 536.312	-30665.539 (Onlooker-SR)	-30665.539 ± 0.0007	-30665.539	-30665.539 ± 0
g05 5126.4981	5126.5284 (rg=50)	5316.7171 ± 247.537	5127.0429 (Employed-Onlooker-SR/rg)	5616.4044 ± 938.696	4986.8712	5559.6410 ± 494.511
g06 -6961.81388	-6827.40219 (rg=50)	-5523.45991 ± 1223.779	-6961.81388 (Employed-SR/Onlooker-SR)	-6961.81388 ± 0	-7661.00942	-6344.10005 ± 1063.805
g07 24.3062091	24.3659 (rg=50)	24.5919 ± 0.194	24.620 (Employed-Onlooker-SR)	24.620 ± 92.996	24.343841	24.513237 ± 0.190
g08 0.095825	0.095825 (rg=20)	0.095799 ± 0	0.095825 (Tüm yöntemler)	0.095825 ± 0	0.095825	0.095825 ± 0
g09 680.6300573	680.633721 (rg=20)	680.648 ± 0.009	680.655 (Employed-SR/rg)	680.755 ± 0.064	680.633	680.641 ± 0.004
g10 7049.25	2128.763 (rg=20)	8098.9257 ± 7409.291	7062.924 (Employed-Onlooker-SR/rg)	10620.716 ± 3942.388	2100	11091.6829 ± 6517.06
g11 0.75	0.749 (rg=20.50)	0.749 ± 0.0004	0.750 (Employed-SR)	0.750 ± 0	0.749	0.750 ± 0.006
g12 1	1.000 (Tüm yöntemler)	1.000 ± 0	1.000 (Tüm yöntemler)	1.000 ± 0	1.000	1.000 ± 0
g13 0.0539498	0.05611 (rg=1)	0.45962 ± 0.386	0.06732 (Onlooker -SR)	0.69821 ± 0.271	0.05730	0.59645 ± 0.365

Bu çalışmada sınırlamalı optimizasyon problemlerini çözmek amacıyla üç yöntem (PEN-ABC, SR-ABC ve DEB-SR-ABC) önerilmiştir. Standart ABC algoritmasına sınırlama ele alış metotlarının entegre edilmesiyle geliştirilen bu yöntemlerin 13 test problemi üzerindeki performansı incelenmiştir. Problem bazında en iyi sonucu bulan yöntem/yöntemler belirlenmiştir. PEN-ABC algoritmasında farklı r_g parametrelerinin sonuca etkisi incelenmiş ve r_g parametresinin artmasıyla genel olarak sonucun daha iyileştiği görülmüştür. Stokastik beslemeli ABC algoritmasında (SR-ABC) stokastik yapı ABC algoritmasının farklı fazlarında denenmiştir ve sadece işçi arı fazına stokastik yapı entegre edilerek (Employed-SR), sadece gözcü arı fazına stokastik yapı entegre edilerek (Onlokker-SR), hem işçi arı hem de gözcü arı fazlarına stokastik yapı entegre edildiğinde (Employed-onlooker-SR) ve ihlal değerleri amaç fonksiyonuna dahil edildiğinde (Employed-Onlooker-SR/rg) daha iyi sonuç alındığı görülmüştür. DEB-SR-ABC algoritmasında, Deb'in kurallarını kullanan ABC algoritmasına ek olarak stokastik yapı entegre edilerek popülasyona çeşitlilik katılması amaçlanmış ancak stokastik yapının entegre edilmesiyle bazı problemlerde daha iyi sonuç alınsa da genel itibariyle Deb'in kurallarını kullanan ABC kadar başarılı olamamıştır. Bu da kaşif arı biriminin yeterince farklılık kattığı sonucuna götürmüştür. Önerilen yöntemler literatürdeki diğer optimizasyon algoritmalarıyla, literatürde sıklıkla kullanılan 13 test problemi üzerinden karşılaştırılmıştır. Yapılan istatistiksel sonuçlar incelendiğinde diğer algoritmalar ile benzer performans gösterdiği söylenebilir.

Kaynakça

[1] Parsopoulos, K. E., Vrahatis, M. N., 2002, "Particle swarm optimization method for constrained optimization problems", *Intelligent Technologies–Theory and Application: New Trends in Intelligent Technologies*, Vol. 76, No. 1, pp. 214-220.

[2] Huang, Z., Wang, C. and Tian, H., 2009, "A genetic algorithm with constrained sorting method for constrained optimization problems", 2009 IEEE International Conference on Intelligent Computing and Intelligent System, Vol. 1, pp. 806-811, IEEE.

[3] Lu, H. and Chen, W., 2008, "Self-adaptive velocity particle swarm optimization for solving constrained optimization problems", Journal of Global Optimization, Vol. 41, No. 3, pp.427-445.
[4] Michalewicz, Z. and Schoenauer, M., 1996, "Evolutionary algorithms for constrained parameter optimization problems", Evolutionary computation, Vol. 4, No. 1, pp.1-32
[5] Runarsson, T.P. and Yao, X., 2000, "Stochastic ranking for constrained evolutionary optimization", IEEE Transactions on evolutionary computation, Vol. 4, No. 3, pp.284-294.
[6] Runarsson, T.P. and Yao, X., 2005, "Search biases in constrained evolutionary optimization", IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), Vol. 35, No. 2, pp.233-243.
[7] Mezura-Montes, E. and Coello, C.A.C., 2005, "A simple multimembered evolution strategy to solve constrained optimization problems", IEEE Transactions on Evolutionary computation, Vol. 9, No. 1, pp.1-17.
[8] Muñoz Zavala, A.E., Aguirre, A.H. and Villa Diharce, E.R., 2005, June, "Constrained optimization via particle evolutionary swarm optimization algorithm (PESO)", In Proceedings of the 7th annual conference on Genetic and evolutionary computation, pp. 209-216, ACM.
[9] Deb, K., 2000, "An efficient constraint handling method for genetic algorithms", Computer methods in applied mechanics and engineering, Vol. 186, No. 2-4, pp.311-338.
[10] Karaboga, D. and Akay, B., 2011, "A modified artificial bee colony (ABC) algorithm for constrained optimization problems", Applied soft computing, Vol. 11, No. 3, pp.3021-3031.
[11] Karaboga, D., 2005, "An idea based on honey bee swarm for numerical optimization", Vol. 200, Technical report-tr06, Erciyes university, engineering faculty, computer engineering department.
[12] Coello, C.A.C., 2002, "Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art", Computer methods in applied mechanics and engineering, Vol. 191, No. 11-12, pp.1245-1287.
[13] Akay, B., 2009, Nümerik optimizasyon problemlerinde yapay arı kolonisi (artificial bee colony) algoritmasının performans analizi, Doktora Tezi, Erciyes Üniversitesi, Fen Bilimleri Enstitüsü, Kayseri.
[14] Karaboga, D. and Basturk, B., 2007, June, "Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems", In International fuzzy systems association world congress, pp. 789-798, Springer, Berlin, Heidelberg.
[15] Liang, J.J., Runarsson, T.P., Mezura-Montes, E., Clerc, M., Suganthan, P.N., Coello, C.C. and Deb, K., 2006, "Problem definitions and evaluation criteria for the CEC 2006 special session on constrained real-parameter optimization", Journal of Applied Mechanics, Vol. 41, No. 8, pp.8-31.