Avrupa Bilim ve Teknoloji Dergisi
Özel Sayı, S. 463-477, Ekim 2019
© Telif hakkı EJOSAT'a aittir
Derleme Makale



www.ejosat.com ISSN:2148-2683

European Journal of Science and Technology
Special Issue, pp. 463-477, October 2019
Copyright © 2019 EJOSAT
Review Article

Son Üç Yılda Geliştirilen Metasezgisel Algoritmalar Hakkında Kısa Bir İnceleme

Yüksel Çelik[1], İlker Yildiz[2], Alper Talha Karadeniz[3*]

[1]Bilgisayar Mühendisliği Bölümü / Mühendislik Fakültesi, Karabük Üniversitesi, Karabük,Türkiye (ORCID: 0000-0002-7117-9736)
[2]Bilgisayar Teknolojileri Bölümü / Bolu Teknik Bilimler Meslek Yüksekokulu, Bolu Abant İzzet Baysal Üniversitesi, Bolu,Türkiye (ORCID: 0000-0002-1575-2673)
[3]Bilgisayar Teknolojileri Bölümü /Bolu Teknik Bilimler Meslek Yüksekokulu, Bolu Abant İzzet Baysal Üniversitesi, Bolu, Türkiye (ORCID: 0000-0003-4165-3932)

(Bu yayın International Congress on Human-Computer Interaction,Optimization and Robotic Application kongresinde sözlü olarak sunulmuştur.)

(İlk Geliş Tarihi 1 Ağustos 2019 ve Kabul Tarihi 25 Ekim 2019)

(**DOI:** 10.31590/ejosat.638431)

ATIF/REFERENCE: Çelik, Y. Yıldız,İ. Karadeniz, A.T. (2019). A Brief Review of Metaheuristic Algorithms Improved in the Last Three Years. *Avrupa Bilim ve Teknoloji Dergisi*, (Özel Sayı), 463-477.

Öz

Optimizasyon bir problemin olabilecek farklı çözümleri arasından en iyi sonucu verenin bulunmasıdır. Optimizasyon problemlerinin çözümünde birçok algoritma kullanılmaktadır. Optimizasyon algoritmaları genel olarak sezgisel optimizasyon algoritmaları ve matematiksel optimizasyon algoritmaları olarak ikiye ayrılmaktadır. Matematiksel optimizasyon algoritmaları, tüm çözüm kümesini tarayarak çözüme ulaşmayı amaçlarken, sezgisel optimizasyon algoritmaları ise, çözüm kümesine sezgisel olarak yaklaşmakta ve en iyi çözüme yada en iyiye yakın bir çözüme ulaşmayı amaçlamaktadır. Çözüm kümesi geniş olan problemlerde matematiksel optimizasyon algoritmalarının kullanımı maliyetlidir. Bu tip problemlerin çözümünde, sezgisel optimizasyon algoritmaları daha avantajlı olup daha fazla tercih edilmektedir. Bir optimizasyon algoritmasının her tür problem veya test fonksiyonu üzerinde başarılı olması beklenemez. Bu nedenle hangi tür problemin hangi algoritma ile daha iyi çözüldüğünün belirlenmesi gerekmektedir. Günümüzde temel sezgisel metotların birleşerek etkili kullanımı sonucunda Metasezgisel isimli algoritmalar geliştirilmiştir. Bu algoritmalar, yüksek seviyeli çalışma ortamında, verimli arama işlemleri kullanarak çözüm uzayındaki optimum çözüme daha hızlı şekilde ulaşmaktadır. Metasezgisel optimizasyon tekniklerinin kullanımının yaygın olması nedeniyle, günümüzde birçok yeni metasezgisel optimizasyon algoritmaları önerilmektedir. Önerilen bu algoritmalar üzerinde geliştirmeler ve hibrit çalışmalar da yapılmaktadır. Bu çalışmada, literatürde son üç yılda (2017-2019) önerilmiş olan, Harris Hawks Optimizasyon Algoritması (HHO), Satin Bowerbird Optimizasyon Algoritması (SBO), Optimal Foraging Algoritması (OFA), Butterfly Optimizasyon Algoritması (BOA), Pity Beetle Algoritması (PBA) ve Collective Decision Optimizasyon Algoritması (CDOA) ele alınmıştır. Geliştirilen bu yeni optimizasyon algoritmalarının esinlendikleri alan, çalışma mantıkları ve arama stratejileri incelenerek sunulmuştur. Gerçekleştirilen bu derlemenin metasezgizel optimizasyon problemleri alanında yapılacak olan çalışmalara ışık tutacağı düşünülmektedir.

Anahtar Kelimeler: Optimizasyon, Metasezgisel, Arama algoritmaları.

A Brief Review of Metaheuristic Algorithms Improved in the Last Three Years

Abstract

Optimization is to find the best solution among the different solutions of a problem. Many algorithms are used to solve optimization problems. Optimization algorithms are generally divided into heuristic optimization algorithms and mathematical optimization algorithms. While mathematical optimization algorithms aim to reach the solution by scanning the whole solution set, heuristic optimization algorithms approach the solution set intuitively and aim to reach the best solution or the best solution. Mathematical optimization algorithms are costly to solve problems with a large set of solutions. In solving such problems, heuristic optimization algorithms are more advantageous and more preferred. An optimization algorithm cannot be expected to succeed on any problem or test function. Therefore, it is necessary to determine which kind of problem is best solved by which algorithm. Nowadays, as a result of the effective use of basic heuristic methods combined, Metaheuristic algorithms have been developed. These algorithms reach the optimum solution in the solution space faster by using efficient search operations in a high level working environment. Due to the

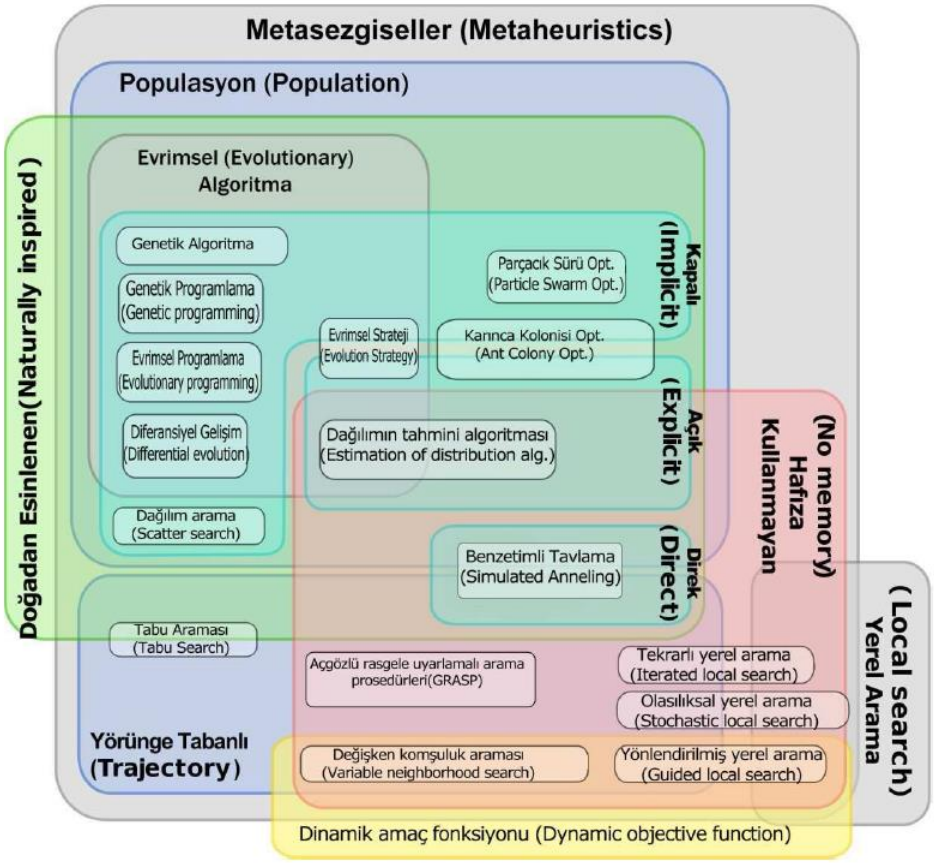
[*]Alper Talha KARADENİZ: Bolu Abant İzzet Baysal Üniversitesi, Bolu Teknik Bilimler Meslek Yüksekokulu, Bilgisayar Teknolojileri Bölümü, Bolu, Türkiye, ORCID: 0000-0003-4165-3932, alperkaradeniz@ibu.edu.tr

widespread use of metaheuristic optimization techniques, many new metaheuristic optimization algorithms have been proposed today. Improved and hybrid studies are also performed on these proposed algorithms. In this study, Harris Hawks Optimization Algorithm (HHO), Satin Bowerbird Optimization Algorithm (SBO), Optimal Foraging Algorithm (OFA), Butterfly Optimization Algorithm (BOA) and Pity Beetle Algorithm (PBA) and Collective Decision Optimization Algorithm (CDOA) have been proposed in the literature in the last three years (2017-2019). The field of inspiration of these new optimization algorithms, study logic and search strategies are presented. It is thought that this review will shed light on the studies to be conducted in the field of metaheuristic optimization problems.

Keywords: Optimization, Metaheuristic, Search algorithm.

1. Giriş

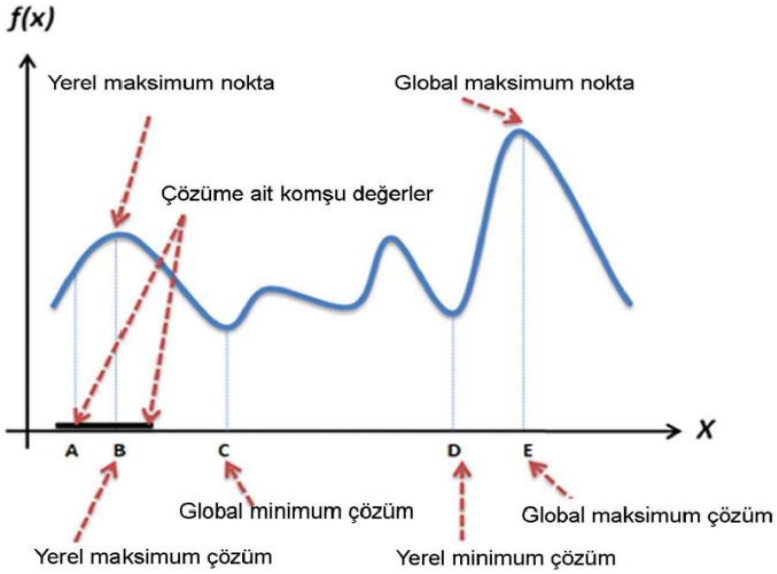
Optimizasyon, bir problemin çözüm uzayındaki en iyi sonucu veren çözümün bulunmasıdır. Bir başka değişle verilen şartlar altında en iyi sonucun elde edilmesinin sağlanmasıdır. Günümüzde birçok problemin çözümünde farklı optimizasyon algoritmaları kullanılmaktadır. Optimizasyon problemlerinin çözümünde matematiksel ve sezgisel teknikler kullanılmaktadır. Çözüm uzayı geniş olan problemlerde, matematiksel algoritmalar tüm çözüm uzayını taradığı için daha maliyetlidir. Bu tip problemlerde çözüm uzayını sezgisel olarak tarayarak daha kısa sürede çözüme ulaşan sezgisel algoritmaların kullanılması daha avantajlıdır (Çelik, 2013). Günümüzde temel sezgisel metotların birleşerek etkili kullanımı sonucunda Metasezgisel isimli algoritmalar geliştirilmiştir. Bu algoritmalar, yüksek seviyeli çalışma ortamında, verimli arama işlemleri kullanarak çözüm uzayındaki optimum çözüme daha hızlı şekilde ulaşmaktadır (Çelik, 2013). Metasezgisel optimizayon algoritmalarının en önemli avantajlarından biri de, yerel optimum noktalara takılmadan global sonuca ulaşabilme yeteneğidir (Laporte, 2006). Literatürde sıklıkla kullanılan ve kabul görmüş birçok metasezgisel algoritma vardır. Bu algoritmalardan başlıcaları; Genetik Algoritma (Genetic Algorithm)(GA),Karınca Kolonisi Optimizasyonu (Ant Colony Optimization)(ACO), Parçacık Sürü Optimizasyonu (Particle Swarm Optimization)(PSO),Yapay Arı Kolonisi (Artificial Bee Colony)(ABC) ,Diferansiyel Gelişim Algoritması (Differential Evolution Algorithm) (DEA), Benzetim Tavlama (Simulated Annealing)(SA), Yerçekimi Arama Algoritması (Gravity Search Algorithm)(GSA), Gaz Brownian Hareketi Optimizasyonu(Gases Brownian Motion Optimization) (GBMO), Isı Transferi Arama (Heat transfer search)(HTS), Elektromanyetik Alan Optimizasyonu (Electromagnetic Field Optimization) (EFO), Optikten Esinlenen Optimizasyon (Optic Inspired Optimization)(OIO), Ağırlıklı Süperpozisyon Çekimi (Weighted Superposition Attraction (WSA), Orman Optimizasyonu Algoritması (Forest Optimization Algorithm)(FOA) olarak verilebilir. Bu algoritmaların yanında her yıl birçok yeni optimizasyon algoritması önerilmekte ve önerilen algoritmalar üzerinde geliştirilmeler de yapılmaktadır. Metasezgisel algoritmalar ve sınıflandırılması şekil 1 de gösterilmiştir (Çelik, 2013).



Şekil 1. Metasezgisel Algoritmaların Sınıflandırılması (Çelik, 2013)

Optimizasyon problemlerinin çözümünde Global arama ve Yerel arama teknikleri uygulanmaktadır. Yerel arama çözüm uzayının belirli bir alanındaki en iyi çözümleri ararken, global arama genel en iyi çözümü bulmayı amaçlamaktadır. Yerel aramada bulunan en

iyi nokta, çözüm uzayındaki en iyi nokta olduğunu garanti etmemektedir (Hart, 1994). Optimizasyon algoritmalarındaki kavramlar şekil 2' de görülmektedir (Çelik, 2013).



Şekil 2. Optimizasyon algoritmalarındaki kavramlar.

Bu çalışmanın ikinci bölümünde literatürde son üç yılda önerilmiş olan bazı Metasezgisel optimizasyon algoritmaları incelenmiştir. Bu algoritmalar Harris Hawks optimization (HHO), Satin Bowerbird Optimizer (SBO), Optimal Foraging Algorithm (OFA), Butterfly Optimization Algorithm (BOA), Collective Decision Optimization Algorithm (CDOA), Pity Beetle Algorithm (PBA) algoritmalarıdır. Üçüncü bölümde, incelenen metasezgisel algoritmaların dahil olduğu sınıflar ve literatürde bu algoritmalar ile yapılan çalışmalar yer almaktadır. Dördüncü bölümde ise yapılan çalışma değerlendirilmiştir.

2. Son 3 yılda Geliştirilen Bazı Metasezgisel Algortimalar

Yapılan literatür incelemesinde, son 3 yılda geliştirilen bazı metasezgisel algortimaların arama stratejileri ve çalışma mantıkları aşağıda ayrıntılı olarak verilmiştir.

2.1. Harris Hawks Optimization Algoritması

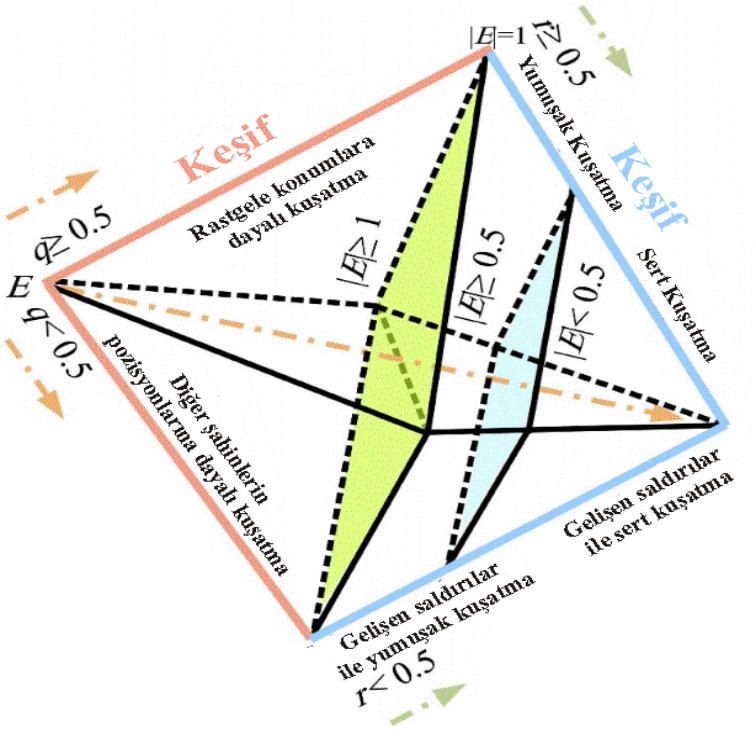
Harris hawks optimizasyon algoritması (HHO), 2019 yılında Ali Asghar Heidari ve arkadaşları tarafından geliştirilmiş popülasyon tabanlı bir algoritmadır. HHO, Harris şahinlerinin doğada sürpriz saldırı olarak adlandırılan, takım davranışlarından ve avcılık modelinden esinlenilen yeni bir sürü zekası optimizasyon paradigmasıdır. (Asghar Heidari, 2019).

Harris şahinleri, doğada bilinen en zeki avcı kuşlardan birisidir (L. Lefebvre, 1997). Bir grup şahin toplanıp ava başladığında, bazı şahinler birbiri ardına kısa turlar yapar ve sonra oldukça yüksek tuniklere inmektedirler. Şahinler taradıkları alanda "sıçrayarak ilerleme" adı verilen bir hareket ile avı tespit etmeye çalışmaktadırlar. Harris şahinlerinin avlanmadaki ana taktiği "sürpriz saldırı" olarak adlandırılmaktadır. Bu akıllı stratejide, birkaç şahin işbirliği içinde, sığınağın dışına kaçtığı tespit edilen ava farklı yönlerden saldırmaya ve eşzamanlı olarak yaklaşmaya çalışmaktadırlar. Saldırı, şahinlerin avlarını birkaç saniye içinde yakalaması ile hızlı bir şekilde tamamlanabilmektedir (Asghar Heidari, 2019).

Bazen avın kaçma yetenekleri ve davranışlarıyla ilgili olarak, sürpriz saldırı hamlesi birkaç dakika boyunca avın yakınında çok sayıda, kısa uzunlukta, hızlı dalışlar içerebilmektedir. Harris şahinleri, koşulların dinamik yapısına ve avın kaçış kalıplarına bağlı olarak çeşitli kovalama tarzları gösterebilmektedir. Bu işbirlikçi taktiklerin temel avantajı ise, Harris şahinlerinin tespit edilen avın tükenmişliğini ve bunun getirdiği savunmasızlığını tespit ederek izleyebilmesidir. Dahası, kaçan avı şaşırtmaktadır ve av savunma yeteneklerini geri kazanamamaktadır. Nihayet bir sürü kuşatması ile karşı karşıya kalan av tükenmiştir ve kaçamayacak duruma gelmiştir. Genellikle en güçlü ve tecrübeli olan şahinlerden birisi, zahmetsizce yorgun düşen avı yakalar ve diğer sürü üyeleriyle paylaşmaktadır (Asghar Heidari, 2019).

2.1.1. Algoritmanın Çalışma Mantığı

HHO' nın arama ve sömürü aşamasındaki tüm fazları şekil 3' de gösterilmektedir (Heming Jia, 2019). Bu fazlar bu bölümde detaylı olarak açıklanmaktadır.



Şekil 3. HHO' nun tüm fazları.

HHO' nun av arama işleminde 2 farklı yöntem kullanılmaktadır. Hangi yöntemin kullanılacağına [0-1] arasında rastgele olarak belirlenen q değerine göre karar verilmektedir. Arama yöntemi denklem 1 ve denklem 2'de gösterilmektedir (Asghar Heidari, 2019).

$$q \geq 0.5 \text{ ise;} \quad X(t+1) = X_rand\ (t) - r_1\ |X_rand\ (t) - 2r_2\ X(t)| \tag{1}$$

$$q < 0.5 \text{ ise;} \quad X(t+1) = (X_rabbit\ (t) - X_m\ (t)) - r_3\ (LB + r_4\ (UB - LB)) \tag{2}$$

Verilen denklemde; X_rabbit hedefin o anki pozisyonu, X_m popülasyon matrisinin ortalama değerini, UB ve LB üst ve alt sınırları, r_1, r_2, r_3, r_4 ise rastgele sayıları ifade etmektedir (Asghar Heidari, 2019).

Keşif işlemi gerçekleştikten sonra, sömürü aşamasına geçilmektedir. İlk etapta avın kaçış enerjisi hesaplanmaktadır. Kaçış enerjisi denklem 3'de gösterilmektedir (Asghar Heidari, 2019).

$$E = 2E_0\ (1 - t / T) \tag{3}$$

Verilen denklemde E_0 [0-1] aralığında rastgele tanımlanan avın başlangıç değeridir. T maksimum iterasyon sayısı, t ise mevcut iterasyon sayısıdır (Asghar Heidari, 2019).

Bu aşamadan sonra Harris şahinleri çok sayıda farklı saldırı yöntemiyle sömürü işlemine başlamaktadır (Bednarz, 1988). Öncelikle [0-1] arasında rastgele bir r sayısı atanmaktadır. Bu r değerine ve E kaçış enerjisine göre farklı stratejilerde sömürü yapılmaktadır. HHO' nun sömürüsü için kullanılacak stratejiler aşağıdaki sözde kodda verilmiştir (Asghar Heidari, 2019).

---

if (|E|≥ 1) **then** ▷ Exploration phase

1. Eşitlik kullanılarak konum vektörü güncelleniyor.

Else if (|E|< 1) **then** ▷ Exploitation phase

if (r ≥0.5 and |E|≥ 0.5) **then** ▷ Soft besiege;

3. Eşitlik kullanılarak konum vektörü güncelleniyor.

else if (r ≥0.5 and |E|< 0.5) **then** ▷ Hard besiege;

4. Eşitlik kullanılarak konum vektörü güncelleniyor.

else if (r < 0.5 and |E|≥ 0.5) **then** ▷ Soft besiege with progressive rapid dives;

5. Eşitlik kullanılarak konum vektörü güncelleniyor.

else if (r < 0.5 and |E|< 0.5) **then** ▷ Hard besiege with progressive rapid dives;

6. Eşitlik kullanılarak konum vektörü güncelleniyor.

Soft besiege (yumuşak kuşatma) işleminin denklemi aşağıda verilmiştir. J ($2(1 - r_5)$) avın kaçma prosedürüdür. Her iterasyonda rastgele olarak değişmektedir.

$$X(t + 1) = \Delta X(t) - E\,|J\,Xrabbit(t) - X(t)| \quad (4)$$

$$\Delta X(t) = Xrabbit(t) - X(t) \quad (5)$$

Hard besiege (sert kuşatma) işleminin denklemi aşağıda verilmiştir.

$$X(t + 1) = Xrabbit(t) - E\,|\Delta X(t)| \quad (6)$$

Soft besiege with progressive rapid dives (gelişen saldırılar ile yumuşak kuşatma) işleminin denklemi aşağıda verilmiştir. D problemin boyutunu, S rastgele vektörü, LF ise levy uçuşu fonksiyonunu ifade etmektedir (Yang, 2010).

$$X(t+1)= \{ \quad (Y \text{ eğer } F(Y) < F(X(t)) \text{ ve } Z \text{ eğer } F(Z)<F(X(t) \quad \} \quad (7)$$

$$Y = Xrabbit(t) - E\,|J\,Xrabbit(t) - X(t)| \quad (8)$$

$$Z = Y + S \times LF(D) \quad (9)$$

Hard besiege with progressive rapid dives (gelişen saldırılar ile sert kuşatma) işleminin denklemi aşağıda verilmiştir.

$$X(t+1) = \{ \quad (Y \text{ eğer } F(Y)<F(X(t)) \text{ ve } Z \text{ eğer } F(Z)<F(X(t) \quad \} \quad (10)$$

$$Y = Xrabbit(t) - E\,|J\,Xrabbit(t) - X_m(t)| \quad (11)$$

$$Z = Y + S \times LF(D) \quad (12)$$

2.1.2. Algoritmanın Arama Stratejisi

- HHO algoritması, popülasyon tabanlı, sürü zekasına dayalı bir optimizasyon algoritmasıdır.
- HHO keşif aşamasında diğer popülasyon tabanlı algoritmaların yanı sıra kullanılan ortalama konum sayesinde keşif işlemini geliştirmiştir.
- Sömürü aşamasında kullanılan E parametresi HHO' nun sömürü yeteneğini, dinamik bir yapı sunarak arttırmaktadır.
- E ve r parametreleri sayesinde birçok farklı strateji seçeneği sunarak, en iyi hareket basamağının seçilmesi sağlanmaktadır.
- Levy uçuşu algoritmasını sömürü aşamasına ekleyerek, yerel optimum noktalarına takılmadan yerel arama işlemine devam edilmesini sağlamaktadır.
- Kullanılan E, r parametreleri ve LF sayesinde HHO keşif ve sömürü arasında oldukça başarılı bir denge elde etmiştir.
- HHO algoritması literatürde yer alan benchmarck (J Digalakis, 2000) test fonksiyonları ile test edilmiş ve literatürde iyi bilinen algoritmalar ile karşılaştırılmıştır (Asghar Heidari, 2019).

2.2. Satin Bowerbird Optimization Algoritması (SBO)

SBO Algoritması 2017 yılında Seyyed Hamid Samareh Moosavi ve Vahid Khatibi Bardsiri tarafından geliştirilmiştir. SBO Algoritması Saten Çardak Kuşlarının çiftleşmesinden esinlenerek oluşturulmuş, evrimsel bir metasezgisel algoritmadır. Doğada, erkek çardak kuşları çeşitli materyaller ile bir çardak inşa ederek bu çardakları farklı materyaller ile süslemektedirler (Coleman SW, 2004). Bunun yanı sıra erkek çardak kuşları yakındaki çardaklardan çeşitli materyalleri kendi çardaklarının seçilme olasılığını arttırmak amacı ile kullanabilmektedir. Dişi kuşlar ise var olan çardakları dolaşarak en dikkat çekici çardağı belirleyerek oraya yerleşmektedirler. SBO algoritmasında, çardak kuşlarının biyolojik yaşayışlarında olduğu gibi altı farklı aşama gerçekleştirilmektedir (Seyyed Moosavi, 2017).

2.2.1. Algoritmanın Çalışma Mantığı

Başlangıç popülasyonu, diğer metasezgisel algoritmalarda olduğu gibi verilen alt ve üst sınırlar arasında rastgele olarak atanmaktadır. Başlangıç popülasyonu atandıktan sonra denklem 13 ve denklem 14'de gösterildiği şekilde ilk çekicilikler hesaplanmaktadır. Fitness değeri denklem 8' de gösterilmiştir. Verilen denklemde fit algoritmadaki fitness değeri, NB ise çardak numarası, $f(X_i)$ o anki pozisyonun maliyet değeridir. (Seyyed Moosavi, 2017).

$$Pi = \frac{fit(i)}{\sum_{n=1}^{NB} fit(n)} \quad (13)$$

$$fit(i) = \{ \quad 1/(1+f(Xi)), f(Xi)\geq 0 \;\; ve \;\; 1+|f(Xi)|, \;\; f(Xi)<0 \quad (14)$$

Yukarıda ki işlemden sonra elitizm aşamasına geçilmektedir. Elitizm, evrimsel algoritmaların önemli özelliklerinden birisidir. En iyi çözüm ya da çözümlerin optimizasyon sürecinin her aşamasında korunmasını sağlamaktadır. Bu amaçla en iyi olasılığa sahip çardak hafızaya alınmaktadır (Seyyed Moosavi, 2017).

Algoritmanın her iterasyonunda konum güncellemesinin yapılması denklem 15' de verilmiştir (Seyyed Moosavi, 2017).

$$x_{ik}^{new} = x_{ik}^{old} + \lambda\left(\left(\frac{x_{jk}+x_{elite,k}}{2}\right) - x_{ik}^{old}\right) \quad (15)$$

Denklemdeki X_i çözüm vektörünü, X_{ik} çözüm vektörünün k. elemanını, X_j mevcut çözümler arasındaki hedef çözümü, Xelite, algoritmanın her iterasyonundaki en iyi değerin konumunu, λ ise her değişken için hesaplanan adım miktarını belirlemektedir. λ parametresinin hesaplanması denklem 16'da verilmiştir. J değeri konumlardan türetilen olasılık değerine göre hesaplanmakta, bu hesaplama için Roulette Wheel (Rulet Tekerleği) seçme yöntemi kullanılmaktadır (Seyyed Moosavi, 2017).

$$\lambda_k = \alpha / (1+Pj) \quad (16)$$

Burada α en büyük adım aralığını belirleyen sabit, Pj ise denklem 16' da hesaplanan çekiciliği ifade etmektedir.

Bir sonraki adımda mutasyon işlemi uygulanmaktadır. Çardağı yapan erkekler dişi tarafından göz ardı edilebileceği gibi, güçlü erkekler zayıf erkeklerden materyal çalabilmekte ve çardaklarını yok edebilmektedir. Bu nedenle, algoritmanın her döngüsünün sonunda, rasgele değişiklikler belirli bir olasılıkla uygulanmaktadır. Rasgele değişiklikler belirli bir olasılıkla X_{ik}'e uygulanmaktadır. Mutasyon işlemi, denklem 17-19' da gösterilmektedir (Seyyed Moosavi, 2017).

$$x_{ik}^{new} = x_{ik}^{old} + \lambda\left(\left(\frac{x_{jk}+x_{elite,k}}{2}\right) - x_{ik}^{old}\right) \quad (17)$$

$$x_{ik}^{new} = x_{ik}^{old} + (\sigma * randn) \quad (18)$$

$$\sigma = z * (UB - LB) \quad (19)$$

Burada UB üst sınır, LB alt sınır, z parametresi üst ve alt sınır arasındaki farkın yüzdesini ifade etmektedir. Ayrıca kovaryans hesabı yapılarak adım değerinin oranı hesaplanmaktadır.

Her iterasyon sonunda eski popülasyon ile mutasyon sonucunda elde edilen popülasyon birleştirilerek sıralanmakta ve yeni popülasyon elde edilmektedir.

SBO algoritmasının sözde kodu şekil 4'te gösterilmiştir (Jagadeeswar Chintam, 2017).

```
While the end criterion is not satisfied
    Calculate the probability of bowers using Equations (11) and (12)
    For every bower
        For every element of bower
            Select a bower using roulette wheel
            Calculate λk using Equation (14)
            Update the position of bower using Equations (13) and (16)
        End for
    End for
    Calculate the cost of all bowers
    Update elite if a bower becomes fitter than the elite
End while
```

Şekil 4. SBO algoritmasının sözde kodu

2.2.2. Algoritmanın Arama Stratejisi

- SBO algoritması doğadan ilham alan evrimsel tabanlı bir metasezgisel algoritmadır.
- Diğer evrimsel tabanlı algoritmaların aksine gerçek dünyadaki veri setleri ile çalışmaktadır.
- SBO optimum çözümü ararken küçük değişiklikler yaparak yakınsama kabiliyetini arttırmaktadır.
- SBO hem en iyi çözüme ulaşmaya çalışırken bir yandan da nüfus çeşitliliğini sağlamaktadır.
- SBO arama alanında sürekli olarak yeni çözümler aramaktadır (Jagadeeswar Chintam, 2017).
- Algoritma iyi bilinen tek modlu ve çok modlu 13 matematiksel fonksiyon ile test edilmiştir. Performansının değerlendirilmesi için PSO, GA, ABC, ALQ ve FA algoritmaları ile karşılaştırılmıştır (Seyyed Moosavi, 2017).

2.3. Optimal Froaging Optimizasyon Algoritması (OFA)

Algoritma Guang-Yu Zhu, Wei-Bo Zhang tarafından 2017 yılında geliştirilmiştir. Optimal Foraging Algoritması (OFA), Hayvan Davranışsal Ekoloji Teorisi olan Optimal yem arama teorisinden ilham alarak geliştirilmiş bir optimizasyon algoritmasıdır. Yeni bir stokastik (olasılıksal) arama algoritması olan OFA, hayvan arama davranışını izleyerek global optimizasyon problemlerini çözmek için kullanılır. Yiyecek arayan hayvan üç soruya cevap bulmalıdır. Bunlar hayvanlar nerede yiyecek arar, hayvan ne zaman yiyecek

arar ve hayvan ne tür yiyecekleri seçer (G Pyke, 1977). Yiyecek arama sırasında hayvanlar, bol miktarda yem barındıran en iyi alanı nasıl bulacağını bilir. OFA' nın kurulduğu temel operatör bu yiyecek arama stratejisidir. Global optimizasyon problemini çözme süreci, hayvan yemlenmesine benzer. Global optimal çözümün komşusu, yani kısıtlama alanındaki f (X) fonksiyonunun lokal optimal çözümü, hayvan yemleme ortamındaki çeşitli alanlar olarak görülebilir. Optimizasyon işlemi, Optimal Foraging Theory (Hayvan Besleme Teorisi) ' nin ardından, net enerji alım oranının maksimize edilebileceği optimum alanı bulmak için farklı alanlardaki yemleri görebilir. Optimum alan bulunduktan sonra, hayvan en iyi yem modeline göre alan içindeki en uygun pozisyonu arayacaktır. Son optimizasyon pozisyonu bulunduğunda, global optimizasyon probleminin optimal çözümü elde edilir (Guang-Yu Zhu, 2017).

2.3.1. Algoritmanın Çalışma Mantığı

Algoritmanın başlangıç popülasyonu, diğer optimizasyon problemlerinin çoğunda olduğu gibi rastgele olarak verilen sınırlar içinde atanmaktadır. Oluşturulan başlangıç popülasyonu maliyet değerlerine göre hesaplanarak büyükten küçüğe doğru sıralanmaktadır. Yem arama işlemi fitness değerine göre iki farklı şekilde denklem 20 ve denklem 21'de gösterilmiştir (Guang-Yu Zhu, 2017).

$$\text{if fitness==1} \qquad r_{1ji} = rand(0,1), r_{2ji} = rand(0,1);$$

$$x_{ji}^{t+1} = x_{ji}^{t} - k \times r_{1ji} \times \left(x_{Ni}^{t} - x_{ji}^{t}\right) + k \times r_{2ji} \times \left(x_{Ni}^{t} - x_{ji}^{t}\right) \tag{20}$$

$$\text{else} \qquad r_{1ji} = rand(0,1), r_{2ji} = rand(0,1);$$

$$x_{ji}^{t+1} = x_{ji}^{t} - k \times r_{1ji} \times \left(x_{bi}^{t} - x_{ji}^{t}\right) + k \times r_{2ji} \times \left(x_{bi}^{t} - x_{ji}^{t}\right) \tag{21}$$

Yem arama işlemi tamamlandıktan sonra yeni pozisyonlar verilen sınırlar içinde denklem 22 ve denklem 23'de gösterildiği şekilde belirlenmektedir.

$$\boldsymbol{if}\left(x_{ji}^{t+1} > x_i^U\right) \quad x_{ji}^{t+1} = 2 \times x_i^U - x_{ji}^{t+1}; \tag{22}$$

$$\boldsymbol{if}\left(x_{ji}^{t+1} < x_i^U\right) \quad x_{ji}^{t+1} = 2 \times x_i^L - x_{ji}^{t+1}; \tag{23}$$

Yeni oluşturulan popülasyonun amaç fonksiyonu değerleri denklem 24'de gösterildiği gibi hesaplanmaktadır.

$$\boldsymbol{for}(j = 1; j < N; j++)$$

$$\{ \quad \lambda_j^{t+1} = rand(0,1);$$

$$\boldsymbol{if}\left(\frac{\lambda_j^{t+1} F_j^{t+1}}{1 + \lambda_j^{t+1}(t+1)} < \frac{F_j^t}{t}\right)$$

$$\{X_j^{t+1} = X_j^{t+1}; \quad F_j^{t+1} = XF_j^{t+1};\}$$

$$\boldsymbol{else}$$

$$\{X_j^{t+1} = X_j^t; \quad F_j^{t+1} = XF_j^t;\} \tag{24}$$

Hesaplama işlemi sonucunda fitness ve tüm popülasyon değerleri en iyiden en kötüye doğru denklem 25'da gösterildiği şekilde sıralanmaktadır.

$$\boldsymbol{if}(F_1^{t+1} < F_{best})$$

$$\{F_{best} = F_1^{t+1}; X_{best} = X_1^{t+1};\}$$

$$t = t + 1; \tag{25}$$

OFA optimizasyon algoritmasının sözde kodu şekil 5'de görülmektedir (Guang-Yu Zhu, 2017).

/* Optimal Foraging Algorithm */

/* Initialization. Generate uniformly distributed random group P^1 composed N individuals*/

1. **for** (j=1; j<=*N*; j++)
2. {
3. **for** (i=1; i <= *d*; i++)
4. $\{x_{ji}^1 = x_i^L + rand(0,1) \times (x_i^U - x_i^L);$ }
5. }
6. Compute the objective function value F_j^1 of each individual X_j^1 in the original group P^1;
7. Order F_j^1 from the best to the worse to obtain F_j^1 $(j = 1,2,\cdots,N)$, and the corresponding sequence of $\boldsymbol{X}_j^1$;
8. $t = 1, F_{best} = F_1^t = \min(F_1^t,\cdots,F_N^t);\ F_N^t = \max(F_1^t,\cdots,F_N^t)$
9. **while** (*t* <= *Max_t*)
10. {
11. **for** (j=1; j<=*N*; j++)
12. { **if** $(F_j^t = F_{best})$
13. **for** (i=1; i <= *d*; i++)
14. { $r_{1ji} = rand(0,1),\ r_{2ji} = rand(0,1);$
15. $x_{ji}^{t+1} = x_{ji}^t - k \times r_{1ji} \times (x_{Ni}^t - x_{ji}^t) + k \times r_{2ji} \times (x_{Ni}^t - x_{ji}^t)$
16. **if** $(x_{ji}^{t+1} > x_i^U)$ $x_{ji}^{t+1} = 2 \times x_i^U - x_{ji}^{t+1};$
17. **if** $(x_{ji}^{t+1} < x_i^U)$ $x_{ji}^{t+1} = 2 \times x_i^L - x_{ji}^{t+1};$
18. }
19. **else**
20. { According to the ordered F_j^t, choose a value from $1,2,\cdots,j-1$ randomly as variable b, then get X_b^t and $F_b^t < F_j^t$.
21. **for** (i=1; i <= *d*; i++)
22. { $r_{1ji} = rand(0,1),\ r_{2ji} = rand(0,1);$
23. $x_{ji}^{t+1} = x_{ji}^t - k \times r_{1ji} \times (x_{bi}^t - x_{ji}^t) + k \times r_{2ji} \times (x_{bi}^t - x_{ji}^t)$
24. **if** $(x_{ji}^{t+1} > x_i^U)$ $x_{ji}^{t+1} = 2 \times x_i^U - x_{ji}^{t+1};$
25. **if** $(x_{ji}^{t+1} < x_i^U)$ $x_{ji}^{t+1} = 2 \times x_i^L - x_{ji}^{t+1};$
26. }
27. }
28. Compute the objective function value F_j^{t+1} of $\boldsymbol{X}_j^{t+1}$;
29. }
30. **for** (j=1; j<=*N*; j++)
31. { $\lambda_j^{t+1} = rand(0,1)$;
32. **if** $(\frac{\lambda_j^{t+1} F_j^{t+1}}{1+\lambda_j^{t+1}(t+1)} < \frac{F_j^t}{t})$
33. $\{X_j^{t+1} = X_j^{t+1};\ F_j^{t+1} = F_j^{t+1};\}$
34. **else**
35. $\{ X_j^{t+1} = X_j^t;\ F_j^{t+1} = F_j^t;\}$
36. }
37. Order F_j^{t+1} from the best to the worse to obtain F_j^{t+1} (*j*=1,2,...,*N*) and the corresponding sequence of $\boldsymbol{X}_j^{t+1}$.
38. **if** $(F_1^{t+1} < F_{best})$
39. $\{F_{best} = F_1^{t+1};\ X_{best} = X_1^{t+1};\}$
40. *t* = *t*+1;
41. }
42. Output X_{best}, F_{best};

Şekil 5. OFA' nın sözde kodu

2.3.2. Algoritmanın Arama Stratejisi

- Literatürde toplama davranışını temel alan birçok optimizasyon algoritması bulunmaktadır. Bu algoritmalardan başlıcaları Karınca Koloni Algoritması (ACO) (M Dorigo, 2019), Bakteriyel Yem Arama Algoritması (BFO) (P Gupta, 2019), Yapay Arı Kolonisi Algoritması (ABC) (D Karaboga, 2019)'dır.
- OFA en uygun yem alanı bulduktan sonra, alan içerisinde ki en iyi pozisyonu bulmayı amaçlamaktadır.
- En son bulunan pozisyon genel en iyi çözüm olarak belirlenmektedir.
- OFA arama işlemine, mevcut pozisyona en yakın çözümden başlanmaktadır.
- Algoritmanın başarısının değerlendirilmesi için OFA temel benchmark test fonksiyonları ile test edilmiş ve literatürde sıklıkla kullanılan PSO, BA, BFO, ABC, DE gibi algoritmalar ile karşılaştırılmıştır.

2.4. Butterfly Optimization Algoritması (BOA)

Sankalap Arora ve Satvir Singh tarafından geliştirilmiş, doğadan esinlenilen metasezgisel bir algoritmadır (Sankalap Arora, 2019). Kelebeklerin yiyecek toplama ve çiftleşme davranışını taklit eden yeni bir optimizasyon tekniğidir. Kelebeklerin yiyecek bulmak ve çiftleşecek eşini bulmak için kullandıkları en önemli duyu kokudur (Blair RB, 1997). Kelebekler birbirlerini çekmek için koku yayarlar. Her kelebek daha fazla koku yayan kelebeğe doğru hareket etmektedir. Daha fazla koku yayan kelebek yoksa rastgele hareket yapar. Amaç fonksiyonuna göre, kelebeğin uyarılma yoğunlukları hesaplanmaktadır.

2.4.1. Algoritmanın Çalışma Mantığı

İlk olarak, algoritmada amaç fonksiyonu, çözüm uzayı, başlangıç popülasyonu tanımlanmakta ve BOA' da kullanılan parametrelerin değerleri atanmaktadır. Kelebeklerin konumları, arama alanı içinde rastgele oluşturulduktan sonra, koku ve zindelik değerleri hesaplanmakta ve saklanmaktadır. Başlangıç değerleri atandıktan sonra, her yinelemede, çözüm uzayındaki tüm kelebekler yeni konumlara taşınır ve ardından uygunluk değerleri hesaplanarak global ve yerel aramalar gerçekleştirilir. Durdurma kriteri sağlanan kadar bu işleme devam edilir. BOA' nın sözde kodu şekil 6'de gösterilmiştir (Sankalap Arora, 2019).

Algorithm 1 Butterfly optimization algorithm

1: Objective function $f(\mathbf{x})$, $\mathbf{x}=(x_1, x_2, \ldots, x_{dim})$, dim = no. of dimensions
2: Generate initial population of n Butterflies $\mathbf{x}_i = (i=1, 2, \ldots, n)$
3: Stimulus Intensity I_i at $\mathbf{x}_i$ is determined by $f(\mathbf{x}_i)$
4: Define sensor modality c, power exponent a and switch probability p
5: **while** stopping criteria not met **do**
6: **for each** butterfly *bf* in population **do**
7: Calculate fragrance for *bf* using Eq. (1)
8: **end for**
9: Find the best *bf*
10: **for each** butterfly *bf* in population **do**
11: Generate a random number r from [0, 1]
12: **if** $r < p$ **then**
13: Move towards best butterfly/solution using Eq. (2)
14: **else**
15: Move randomly using Eq. (3)
16: **end if**
17: **end for**
18: Update the value of a
19: **end while**
20: Output the best solution found.

Şekil 6. BOA' nın sözde kodu

Her bir kelebek için algılanan kokunun hesaplanması denklem 26'da gösterilmiştir (Blair RB, 1997).

$$f = cI^a \tag{26}$$

Burada a ve c [0,1] arasında rastgele bir sayı, f kokunun algılanan büyüklüğü, I ise uyarıcı yoğunluğunu ifade eder.

Her kelebeğin konumu güncellenerek optimum noktaya ulaşmaya çalışılmaktadır. Konum güncellemesi denklem 27' de verilen formüle göre global şekilde, denklem 28'de lokal şekilde gerçekleştirilerek, her bir kelebeğin yeni pozisyonu belirlenmektedir (Blair RB, 1997).

$$x_i^{t+1} = x_i^t + (r^2 \times g^* - x_i^t) \times f_i \tag{27}$$

$$x_i^{t+1} = x_i^t + \left(r^2 \times x_j^t - x_k^t\right) \times f_i \tag{28}$$

Burada x_i^t mevcut konumu, r [0,1] arasında rastgele bir sayıyı, g* genel en iyi değeri, x_k^t sürüde ki rastgele bir kelebeğin konumunu, f_i ise kelebeğin hissedilen kokusunu temsil etmektedir.

2.4.2. Algoritmanın Arama Stratejisi

- BOA birçok metasezgisel algoritmaya benzerdir ancak bazı farklılıkları mevcuttur.
- Tüm kelebekler birbirlerini çekmek için bir koku yaymaktadır.
- Her kelebek en fazla koku yayan kelebeğe doğru ya da rastgele olarak hareket etmektedir.
- BOA' da kelebeklerin sayısı yani popülasyon sayısı değişmez.
- PSO gibi sürü zekâsına dayalı bir algoritmadır ancak temelde farklı biyolojik yapılardan esinlenmektedir. PSO da tüm parçalar birbirleriyle haberli ve koordineli bir şekilde çalışmaktadır ve bilgi kaybı yoktur. Ancak BOA' genel en iyi kelebeğin bilgisi haricinde kayıp vardır. Bu kaybın algoritmayı nasıl etkileyeceği üzerinde çalışmalar yapılmaktadır.
- BOA, Ateş Böceği (FA) algoritmasına çok benzemektedir. İki algoritma arasında ki en önemli fark arama düzenidir. FA da bireyler daha iyi olan bireye doğru hareket ederken, BOA da tüm bireyler en iyiye doğru hareket etmektedir.
- BOA ve ABC arasındaki temel fark, BOA da hiçbir çözüm atılmaz ve her çözüme iyileşmesi için eşit fırsat verilmektedir.
- BOA kavramsal olarak kolay anlaşılabilir ve kolay kullanılabilir bir yapıya sahiptir.
- BOA' nın performans değerlendirilmesi için benchmarck fonksiyonları ve temel mühendislik problemleri ile test edilmiş ve literatürde sıkça kullanılan metasezgisel algoritmalar ile karşılaştırılmıştır.

2.5. Collective Decision Optimization Algoritması (CDOA)

Collective decision optimization algorithm (CDOA), yapay sinir ağlarının eğitimi için doğadan ilham alan yeni bir metasezgisel algoritmadır. İnsanın karar verme davranışı, önerilen tekniğin temel ilham kaynağıdır. Bunu bir toplantıda herkes fikrini söyler ve sonuç olarak en doğru karar alınır şeklinde düşünebiliriz. Toplantıda bulunan her kişiye karar verici denmektedir. Her karar vericinin kendi fikri vardır, aralarında tartışmalar yaparak en doğru fikre karar vermektedirler (Qingyang Zhang, 2017).

İyi bir plan geliştirmek basit bir iş değildir, birçok faktör gerekmektedir. Bu etki faktörleri arasında deneyim, lider, diğer karar vericilerin bakış açısı ve yenilik sayılabilmektedir (Nahavandi, 2006). Bu faktörlerin herhangi birisine bağlı olarak daha iyi bir plan oluşturulabilmektedir. CDOA algoritmasında bu faktörler ayrı ayrı incelenmektedir.

2.5.1. Algoritmanın Çalışma Mantığı

CDOA algoritmasında, metasezgisel algoritmaların çoğunda olduğu gibi, başlangıç popülasyonu rastgele olarak atanmaktadır. CDOA algoritmasının çalışmasında önerilen birçok yöntem bulunmaktadır (Qingyang Zhang, 2017).

- **Deneyim Tabanlı Aşama**

Kişisel tecrübenin aktarılması olarak tanımlanmaktadır. Hesaplanması denklem 29'da gösterilmiştir (Qingyang Zhang, 2017).

$$\text{newX}_{i0} = X_i(t) + \vec{\tau} \times step_{size}(t) \times d_0 d_0 = \varphi p - X_i(t) \tag{29}$$

Burada, r [0,1] arasında rastgele bir sayı, $step_{size}(t)$ mevcut iterasyon, $d_0$iterasyon yönü, φp bireyin en iyi pozisyonudur.

- **Diğerlerine Dayalı Aşama**

Deneyime dayalı aşamada, karar vericilerin kendi düşünceleri veya planları oluşturulmuştur. Toplantıdaki diğer üyelerle rastgele etkileşimde bulunmaktadırlar. Tartışmanın ve iletişimin yardımı ile farklı bir kararcının kendisinden daha iyi düşünmesi durumunda, karar vericinin fikrini değiştirebilmektedir. Bu işlem denklem 30'da gösterilmiştir.

$$\text{newX}_{i1} = newX_{i0} + \vec{\tau} \times step_{size}(t) \times d_1 d_1 = beta_1 \times d_0 + beta_{11} \times \left(X_j(t) - X_i(t)\right) \tag{30}$$

Burada, J [1, N] arasındaki rastgele bir sayı, $beta_1$ve $beta_{11}$ (-1,1) ve (0,2) arasında rastgele birer sayıdır.

- **Grup Düşüncesine Dayalı Aşama**

Tüm bireylerin ortalama değerini temsil etmektedir. Grup düşüncesine dayalı aşama denklem 31 ve denklem 32'de gösterilmektedir.

$$Q_G = 7\frac{1}{N}\left(x_1(t), x_2(t), \dots\dots, x_N(t)\right) = \left\{\frac{1}{N}\sum_{i=1}^{N} \text{x}_i^1(t), \frac{1}{N}\sum_{i=1}^{N} x_i^2(t), \dots\dots, \frac{1}{N}\sum_{i=1}^{N} x_i^D(t)\right\} \tag{31}$$

$$\text{newX}_{i2} = newX_{i1} + \vec{\tau} \times step_{size}(t) \times d_2 d_2 = beta_2 \times d_1 + beta_{22} \times \left(Q_G - X_i(t)\right) \tag{32}$$

- **Lider Tabanlı Aşama**

Lider popülasyondaki en iyi birey olarak kabul edilmektedir. Hesaplanması denklem 33'de gösterilmektedir.

$$\text{newX}_{i3} = newX_{i2} + \vec{\tau} \times step_{size}(t) \times d_3 d_3 = beta_3 \times d_2 + beta_{33} \times \left(Q_L - X_i(t)\right) \tag{33}$$

Liderin etrafında yerel arama yapılması avantaj sağlayabilmektedir. Yerel arama işlemi denklem 34'de gösterilmektedir.

$$newX_q = Q_L + \overrightarrow{W_q} \ (q = 1,2,3,4,5) \tag{34}$$

Burada W_q rastgele bir sayıyı temsil etmektedir.

- **Yenilik Temelli Aşama**

Aynı fikirde sabit kalmamak için küçük bir mutasyon işlemi kullanılmıştır. Bu işlem denklem 35'de gösterildiği gibi açıklanmaktadır.

$$\mathrm{r_1} \leq \mathrm{MFnewX_{i4}} = \mathrm{newX_{i3}newX_{i4}^{p}} = \mathrm{LB(p)} + \mathrm{r_2} \times \big(\mathrm{UB(p)} - \mathrm{LB(p)}\big) \tag{35}$$

2.4.2. Algoritmanın Arama Stratejisi

- Yerel en iyi durum, diğer üyelerin durumu, üyelerin ortalaması, genel en iyi durum ve mutasyon işlemlerinin tamamı uygulandığı için başarılı sonuçlar verebilen bir algoritmadır.
- Bulunan en iyi çözümün bir sonraki aşamada kullanılması evrimsel algoritmalara benzemektedir.
- Yeni aday çözümler hem nüfus çeşitliliğini sağlar hem de optimum değeri bulurken kötü bireylerin dışarda kalmasını sağlamaktadır.
- Erken yakınsamayı önlemek için, farklı birçok yeni çözüm önermektedir.
- Aramanın farklı konumlara hareket etmesi, arama alanının daha iyi keşfedilmesini sağlamaktadır. Bu sayede, en uygun çözümü bulma olasılığı artmaktadır.
- Mutasyon işlemi ile çeşitliliğin artması sonucunda, yerel optimum noktalardan kaçınılabilmektedir.
- Adım büyüklüğü parametresi, algoritmanın keşif ve sömürü aşamasını dengelemek için kullanılmıştır.
- CDOA algoritması PSO' ya benzemektedir. Ancak yeni adayların üretimi PSO' dan tamamen farklıdır. PSO' da her aşamada bir tane yeni birey oluşturulabilirken, CDOA çok sayıda yeni birey oluşturulabilmektedir.

2.6. Pity Beetle Optimizasyon Algoritması (PBA)

Pity Böceği algoritması (PBA) Nikos Ath. Kallioras, Nikos D. Lagaros ve Dimitrios N. Avtzis tarafından 2018 yılında geliştirilmiş doğadan esinlenilen metasezgisel bir algoritmadır. Altı dişli ladin kabuğu böceği olarak da bilinen Pityogenes chalcographus adlı böceğin yuva ve yiyecek araştırma davranışından ilham alınmıştır (Spyros Chandrinos, 2018). Bu böcek, zayıflamış ağaçların kabuğundaki bir ormana yerleşme ve hasat etme kabiliyetine sahipken, popülasyon belirli bir eşiği aştığında sağlıklı ve sağlam ağaçlara da zarar verebilir. Başlangıçta, öncü erkek böcekler, zayıflamış ağaçların yaydığı kimyasal özelliklerden yararlanarak uygun bir konak bulur. Bu böcekler konakta beslenmeye başladığında, yerel olarak popülasyonu artırarak erkekleri ve dişileri çeken bir feromon üretir. Belirli bir popülasyon sayısına ulaşıldığında, konağın savunma mekanizmaları artık bu kitlesel istilayı savunamaz ve böcekler hem sağlam hem de zayıflamış ağaçlara saldırılabilirler. Bununla birlikte, kalabalık bir konak ağaç, istila üzerinde olumsuz bir etkiye sahip olabileceğinden, istila yoğunluğu çok yükseldiğinde, burada beslenen böcekler daha fazla böcekleri diğer ağaçlara yönlendiren bir feromon yayar (Nikos Kallioras, 2018).

2.6.1. Algoritmanın Çalışma Mantığı

PBA başlangıç, yeni hiper hacim seçme modeli ve popülasyon pozisyonun güncellenmesi olmak üzere 3 adımdan oluşur. Algoritmanın ilk basamağında, ilk popülasyon konumu, arama alanına rasgele olarak üretilmektedir (birinci nesil). İkinci aşamada, ilk popülasyonun parçacıkları, yeni popülasyonlar oluşturmak için farklı hiper hacimler içindeki diğer pozisyonlara hareket etmektedir. (ikinci nesil). Her nesilde yeni popülasyonlar üretilmekte ve üçüncü adımda yeni popülasyonlar öncekilerin yerine geçmektedir. PBA' nın sözde kodu Şekil 7'de sunulmuştur (Nikos Kallioras, 2018).

```
1.  Begin
2.     g:=0
3.     Initialize() Eq. (6)
4.     Repeat
5.      For k:=1 To N_broods Do Begin
6.           For j:=1 To N_pop Do Begin
7.             NewHypervolumeSelectionPattern()
8.             CalculateF()
9.             FE:=FE+1
10.          End
11.          UpdatePopulationLocation()
12.       End
13.    Until TerminationCriterion(FE>FE_total)
14. End
```

Şekil 7. PBA' nın sözde kodu

Yeni hiper hacim seçme modeli denklem 36-38'e göre hesaplanmaktadır (Nikos Kallioras, 2018).

$$x_j^{(g)} = RST(l^{(g)}, u^{(g)}, D, N_{pop}) \tag{36}$$

$$[l^{(g)}, u^{(g)}] \in [x_{birth,i}^{(g)}(1 - f_{pat}), x_{birth,i}^{(g)}(1 + f_{pat})] \tag{37}$$

Burada i j nci bireysel çözüm vektörünün (x_j) elemman bileşenini, g nesili, $x_{birth,i}^{(g)}$ i nci birth pozisyon/çözüm vektörünü, $l^{(g)}$ ve $u^{(g)}$ g nci takip aşaması için i nci boyutun alt üst sınırlarını ifade etmektedir.

$$x_{birth}^{(g+1)} = \begin{cases} x_{birth}^{(g)} \; if \; F\left(x_{birth}^{(g)}\right) < F\left(x_{j,i}^{(g)}\right), \forall \, j = 1,2,\dots,N_{pop}, k = 1,2..,N_{broods} \\ x_{j,k}^{(g)} \; otherwise \end{cases} \tag{38}$$

Burada g nesil, $x_{birth}^{(g)}$ birth pozisyon vektörü, $x_{j,k}^{(g)}$ k popülasyonunda bulunan yeni pozisyon vektörü, N_{broods} sonlandırma kriteri olarak kullanılan maksimum yavru sayısıdır.

Popülasyon yerlerinin güncellenmesi Şekil 8'de verilen formüllerle hesaplanmaktadır.

$$x_{j,k}^{(g)} = \begin{cases} RST([x_{birth,i}^{(g)} \cdot (1 - f_{nb}), x_{birth,i}^{(g)} \cdot (1 + f_{nb})], D, N_{pop}), \\ if \; k = 1 \\ else \\ \begin{cases} RST(\boldsymbol{L}, \boldsymbol{U}, D, N_{pop}), \quad if \; FE > FE_{un} \\ else \\ \begin{cases} RST([x_{birth,i}^{(g)} \cdot (1 - f_{ms}), x_{birth,i}^{(g)} \cdot (1 + f_{ms})], D, N_{pop}) \\ \quad , \; \exists j f(x_{j,k-1}^{(g)}) < f(x_{birth}^{(g)}) \\ else \\ \begin{cases} RST([x_{birth,i}^{(g)} \cdot (1 - f_{ls}), x_{birth,i}^{(g)} \cdot (1 + f_{ls})], D, N_{pop}), \text{if } r < pr \\ \boldsymbol{MEM}, otherwise \end{cases} \end{cases} \end{cases} \\ i = 1, 2, \dots, D, j = 1, 2, \dots, N_{pop} \end{cases}$$

Şekil 8. Popülasyon yerlerinin güncellenmesi

2.6.2. Algoritmanın Arama Stratejisi

- PSO' nun aksine PBA' da ki küme popülasyon olarak adlandırılmaktadır.
- Tüm parçalar, kendi popülasyonlarını oluşturmak için çözüm uzayı içinde daha iyi bir çözüm aramaktadır.
- PBA' da 5 tür seçim modeli uygulanmaktadır.

  (i) komşu arama hacmi

  (ii) orta ölçekli arama hacmi

  (iii) büyük ölçekli arama hacmi

  (iv) global ölçekli arama hacmi

  (v) bellek değerlendirmesi

- Oluşturulan popülasyonun uygunluk değeri verilen eşik değeri aşarsa, popülasyon tekrar rastgele olarak üretilir.
- Algoritmanın optimum çözüme ulaşmasında iyi olan bireylerin tutulması önemlidir. PBA' da iyi olan bireyler hafıza (MEM) da tutulmaktadır.
- Algoritma literatürde iyi bilinen meta-sezgisel algoritmalarla, birçok test fonksiyonu sonucuna göre test edilmiştir.

3. Tartışma

Metasezgisel algoritmaları sınıflandırmanın birçok yolu bulunmaktadır. Metasezgisel algoritmalar, literatürde en çok sürü zekasına dayalı, doğadan ilham alan, evrimsel, popülasyon tabanlı ve olasılıksal olarak sınıflandırılmaktadır. Sürü zekâsına dayalı algoritmalar, sürülerin, kuşların, böceklerin ve hayvan gruplarının, sosyal davranışlarını taklit ederek, genellikle yiyecek bulma davranışından esinlenmektedir (Fatma Hashim, 2019). Doğadan ilham alan algoritmalar, doğada var olan olayları veya canlıları referans alarak, aralarındaki sistematik ilişkiden esinlenilmiş algoritmalardır (Çelik, 2013). Evrimsel algoritmalar, biyolojideki temel evrim kurallarından esinlenmektedir (Holland, 1992). Popülasyon tabanlı algoritmalar, çözüm kümesinde birden fazla çözüm bulunduran algoritmalardır (Zahra Beheshti, 2013). Olasılıksal algoritmalar ise, sistemin sürekli olarak rastgele çalıştığı algoritmalardır. Takip edilen yol her iterasyonda farklılık göstermektedir (Çelik, 2013).

Çalışmada incelenen metasezgisel algoritmaların dahil olduğu algoritma sınıfları Tablo 1' de verilmiştir.

Tablo 1. İncelenen metasezgisel algoritmaların sınıflandırılması

	Sürü Zekâsına Dayalı	Doğadan İlham Alan	Evrimsel	Popülasyon Tabanlı	Olasılıksal
Harris Hawks Optimizasyon Algoritması (HHO)	+	+	-	+	-
Satin Bowerbird Optimizasyon Algoritması (SBO)	-	+	+	-	-
Optimal Froaging Optimizasyon Algoritması (OFA)	-	+	-	+	+
Butterfly Optimizasyon Algoritması (BOA)	+	+	-	+	-
Pity Beetle Optimizasyon Algoritması (PBA)	+	+	-	+	-
Collective Decision Optimizasyon Algoritması (CDO	+	+	+	+	+

Tablo 1'de gösterildiği gibi bir metasezgisel algoritma birden fazla sınıfa dahil olabilmektedir. SBO ve OFA haricindeki algoritmaların hepsi sürü zekâsına dayalıdır. OFA ve CDO haricinde ki algoritmalar olasılıksal değildir. İncelenen algoritmaların çoğunun popülasyon tabanlı ve sürü zekasına dayalı olduğu görülmektedir. Bu derlemede incelenen son üç yılda (2017, 2018, 2019) geliştirilmiş bazı algoritmalar ile birçok çalışma yapılmıştır. Literatürde yer alan bu çalışmalar, Tablo 2'de verilmiştir.

Tablo 2. . Literatürde yer alan bu çalışmalar

Algoritmalar	Yapılan Çalışmalar
HHO	Uydu görüntüsünde gürültü arındırmada kullanılmıştır (NA Golilarz, 2019). Hibrid yapıda çok düzeyli eşik renkli görüntü segmentasyonunda kullanılmıştır (X Bao, 2019). Günlük hava kirliliği tahminde Multiobjective Harris Hawks Optimization Algorithm (MOHHO) adında geliştirmiştir (P Du, 2019).
SBO	Kullanım faktörüne dayalı bir optimizasyon konsepti olarak kullanılmıştır (K Senthilkumar, 2019). Rüzgaz hızı tahmininde Multi-objective Satin Bowerbird Optimizer (MOSBO) adında geliştirilmiştir (C Tian, 2018). Yelken balığı (Sail Fish) Optimizasyonun (SFO)' nun performansının doğrulanmasında kullanılmıştır (S Shadravan, 2019). Radyal dağıtım şebekelerinde güç kaybını azaltmak için yenilenebilir radyal dağılım sistemleri ve rüzgar tribününün optimum yerleşme ve boyutlandırma işleminde kullanılmıştır (M Tolba, 2018). Jeneratörlerin tıkanıklık yönetiminde gerçek gücün yeniden planlamasında kullanılmıştır (J Chintam, 2018).
OFA	Delim güzergahı optimizasyonunda kullanılmıştır (WB Zhang, 2017). Destek vektör makinesinin parametrelerinin optimizasyonunda OFA modifiye edilerek kullanılmıştır (GI Sayed, 2018).
BOA	Numerik optimizayon için BOA algoritması Artificial Bee Colony (ABC) algoritmasıyla birleştirerek kullanılmıştır (S Arora, 2017). Mekanik tasarım optimizasyon problemleri için BOA algoritması Modified Butterfly Optimization Algorithm (MBOA) adında modifiye edilmiştir (S Arora S. S., 2018). Kablosuz Sensör Ağlarda sensör düğümlerinin lokalizasyonunda kullanılmıştır (Arora S, 2018) . BOA nın binary tiplerini özellik seçiminde kullanılmıştır (S Arora P. A., 2019).
PBA	Finansal piyasalarda uygulanabilecek karlı bir portföyün oluşturulmasında ampirik teknik ticaret stratejisi geliştirilmesinde kullanılmıştır (SK Chandrinos, 2018).

CDO	Ekonomik Enenerji dağıtım Probleminde CDOA geliştirilerek a Multiobjective Collective Decision Optimization Algorithm (MOCDOA) adında uygulanmıştır (X Xu, 2018). Dinamik termal hat dereceli yeniden yapılandırılabilir mikro şebekelerin etkin planlanmasında CDOA ya dayanan optimizasyon çalışması yapılmıştır (M Dabbaghjamanesh, 2018). Sınırlı geçici kararlı optimal güç akışı probleminde kullanılmıştır (A Saha, 2018). İnsansız hava araçlarının güzergah planlama problemini çözmek için CDOA Extension Collective Decision Optimization Algorithm (ECDOA) adında genişletilerek uygulanmıştır (Q Zhang, 2018).

Verilen bilgiler ışığında, metasezgisel algoritmalar literatüre sunulduktan hemen sonra, araştırmacılar birbirinden farklı çalışmalar yaparak bu algoritmaları test etmeye başlamışlarıdır. Bu durum metasezgisel algoritmaların popülerliğini göstermektedir.

4. Sonuç

Sonuç olarak, metasezgisel optimizasyon algoritmaları, optimizasyon ve hesaplama problemlerinde yeni bir çığır açmıştır. Zaman maliyetinin düşük olması ve kullanımının yaygın olması sonucunda, literatüre her yıl birçok yeni metasezgisel algoritma girmektedir. Her optimizasyon algoritmasının, her türden problemi iyi çözmesi beklenmemelidir. Amaç probleme uygun algoritmanın kullanılmasıdır. Problemin türüne göre metasezgisel algoritma sınıflarından uygun olanı seçilerek, bu sınıfta yer alan algoritmalar kıyaslanmalı ve problem için iyi olan algoritma seçilmelidir. Bu çalışmada, literatüre son 3 yılda giren bazı metasezgisel algoritmalar incelenmiştir. Son yıllarda geliştirilen yeni metasezgisel optimizasyon algoritmalarının bir çok çalışmada etkili bir biçimde kullanılabileceği açıktır. Yapılan çalışmanın ileride metasezgisel optimizasyon algoritmalarıyla yapılacak olan çalışmalara ışık tutabileceği düşünülmektedir.

Kaynakça

A Saha, A. B. (2018). CDO - A New Metaheuristic Algorithm Towards the Solution of Transient Stability Constrained Optimal Power Flow. *International IEEE*.

Arora S, S. S. (2018). Node Localization in Wireless Sensor Networks Using Butterfly Optimization Algorithm. *COMPUTER ENGINEERING AND COMPUTER SCIENCE*.

Asghar Heidari, S. M. (2019). Harris hawks optimization: Algorithm and applications. *Future Generation Computer Systems*, 849–872.

Bednarz, J. (1988). Cooperative Hunting Harris' Hawks. *American Association for the Advancement of Science.*, 1525-1527.

Blair RB, L. A. (1997). Butterfly Diversity And Human Land Use:Species Assemblages Along An Urban Grandient. *Biol Conserv*, 113–125.

C Tian, Y. H. (2018). A Novel Wind Speed Forecasting System Based On Hybrid Data Preprocessing And Multi-Objective Optimization. *Applied energy*.

Coleman SW, P. G. (2004). Variable Female Preferences Drive Complex Male Displays. *Research Gate*.

Çelik, Y. (2013). Optimizasyon Problemlerinde Bal Arılarının Evlilik Optimizasyonu Algoritmasının Performansının Geliştirilmesi. *Doktora Tezi*. Konya, Türkiye: Selçuk Üniversitesi Fen Bilimleri Enstitüsü.

D Karaboga, B. G. (2019). Solving Traveling Salesman Problem by Using Combinatorial Artificial Bee Colony Algorithms. *International Journal on Artificial Intelligence Tools*.

Fatma Hashim, E. H. (2019). Henry Gas Solubility Optimization: A Novel Physics-Based Algorithm. *Future Generation Computer Systems*, 646–667.

G Pyke, H. P. (1977). A Selective Review Of Theory And Tests. *Quarterly Review of Biology* , 137-154.

GI Sayed, M. S. (2018). Modified Optimal Foraging Algorithm For Parameters Optimization Of Support Vector Machine. *International Conference on Advanced* .

Guang-Yu Zhu, W.-B. Z. (2017). Optimal Foraging Algorithm For Global Optimization. *Applied Soft Computing*, 294–313.

Hart, W. E. (1994). Adaptive Global Optimization With Local Search. *A dissertation submitted in partial satisfaction of the requirements for the degree Doctor of Philosophy in Computer Science & Engineering*. San Diego: University of California, San Diego.

Heming Jia, C. L. (2019). Dynamic Harris Hawks Optimization with Mutation Mechanism for Satellite Image Segmentation. *Remote sensing* .

Holland, J. (1992). Genetic Algorithms. *Scientific American*, 66-72.

J Chintam, M. D. (2018). Real-Power Rescheduling of Generators for Congestion Management Using a Novel Satin Bowerbird Optimization Algorithm. *Energies*.

J Digalakis, K. M. (2000). On Benchmarking Functions For Genetic Algorithms. *International Journal of Computer Mathematics*, 1-27.

Jagadeeswar Chintam, M. D. (2017). Real-Power Rescheduling of Generators for Congestion Management Using a Novel Satin Bowerbird Optimization Algorithm. *Energies*.

K Senthilkumar, R. R. (2019). Optimized Scheduling Of Multicore ECU Architecture With Bio-Security CAN network using AUTOSAR. *Future Generation Computer Systems*.

L. Lefebvre, P. W. (1997). Feeding İnnovations And Forebrain Size İn Birds. *Animal Behaviour*, 549-560.

Laporte, G. (2006). Classical And Modern Heuristics For The Vehicle Routing Problem. *International transactions in operation research* , 285-300.

M Dabbaghjamanesh, A. K.-F. (2018). Effective Scheduling of Reconfigurable Microgrids With Dynamic Thermal Line Rating. *IEEE Transactions*.

M Dorigo, T. S. (2019). Ant Colony Optimization: Overview And Recent Advances. *Springer*.

M Tolba, H. R. (2018). Impact of Optimal Allocation of Renewable Distributed Generation in Radial Distribution Systems Based on Different Optimization Algorithms. *Energies*.

NA Golilarz, H. G. (2019). Satellite Image De-Noising With Harris Hawks Meta Heuristic Optimization Algorithm and Improved Adaptive Generalized Gaussian Distribution Threshold Function. *IEEE Access*.

Nahavandi, A. (2006). *The Art and Science of Leadership.* Prentice Hall.

Nikos Kallioras, N. L. (2018). A New Metaheuristic İnspired By Bark Beetles For Solving Engineering Problems. *Advances in Engineering Software* .

P Du, J. W. (2019). A Novel Hybrid Model Based On Multi-Objective Harris Hawks Optimization Algorithm For Daily PM2. 5 And PM10 Forecasting. *arXiv.org*.

P Gupta, A. S. (2019). Clustering-Based Optimized HEED Protocols For Wsns Using Bacterial Foraging Optimization And Fuzzy Logic System. *Springer*.

Q Zhang, R. W. (2018). Modified Collective Decision Optimization Algorithm With Application İn Trajectory Planning Of UAV. *Applied Intelligence*.

Qingyang Zhang, R. W. (2017). COLLECTİVE DECİSİON OPTİMİZATİON ALGORİTHM: A NEW HEURİSTİC OPTİMİZATİON METHOD. *Neurocomputing*, 123–137.

S Arora, P. A. (2019). Binary Butterfly Optimization Approaches For Feature Selection. *Expert Systems with Applications*.

S Arora, S. S. (2017). An Effective Hybrid Butterfly Optimization Algorithm with Artificial Bee Colony for Numerical Optimization. *International Journal of Interactive Multimedia*.

S Arora, S. S. (2018). A Modified Butterfly Optimization Algorithm For Mechanical Design Optimization Problems. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*.

S Shadravan, H. N. (2019). The Sailfish Optimizer: A novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems. *Engineering Applications of Artificial*.

Sankalap Arora, S. S. (2019). Butterfly Optimization Algorithm: A Novel Approach For Global Optimization. *Soft Computing*, 715–734.

Seyyed Moosavi, V. B. (2017). Satin Bowerbird Optimizer: A New Optimization Algorithm To Optimize ANFIS For Software Development Effort Estimation. *Engineering Applications of Artificial Intelligence*, 1-15.

SK Chandrinos, N. L. (2018). Construction Of Currency Portfolios By Means Of An Optimized İnvestment Strategy. *Operations Research Perspectives*.

Spyros Chandrinos, N. L. (2018). Construction of Currency Portfolios by means of an Optimized Investment Strategy. *Operations Research Perspectives*, 32–44.

WB Zhang, G. Z. (2017). Drilling Path Optimization By Optimal Foraging Algorithm. *IEEE Transactions on Industrial*.

wikipedia. (2019, Temmuz). *wikipedia*. June 20, 2019 tarihinde wikipedia: http://en.wikipedia.org adresinden alındı

X Bao, H. J. (2019). A Novel Hybrid Harris Hawks Optimization for Color Image Multilevel Thresholding Segmentation. *IEEE Access*.

X Xu, Z. H. (2018). Multiobjective Collective Decision Optimization Algorithm for Economic Emission Dispatch Problem. *Complexity*.

Yang, X.-S. (2010). *Nature-inspired Metaheuristic Algorithms.* United Kindom: Luniver Press.

Zahra Beheshti, S. M. (2013). A Review of Population-based Meta-Heuristic Algorithm. *International journal of advances in soft computing and its applications*.