Nature-Inspired Engineering Optimization

Kawal Jeet
Department of Computer Science
D.A.V. College, Jalandhar, Jalandhar, India
kawaljeet80@gmail.com

***Abstract*—Nature has always been a source of inspiration for human beings. It is quite apparent in recent engineering and optimization problems that have found their solutions in nature-inspired algorithms. The basic steps followed by most of these algorithms are same. Difference lies in the way these basic steps are implemented. In this paper, implementation details of six recent and popular nature-inspired algorithms namely, Artificial Bee Colony Algorithm, Bat Algorithm, Black Hole Algorithm, Cuckoo Search Algorithm, Flower Pollination Algorithm and Grey Wolf Optimization Algorithm have been discussed. They are further compared on the basis of attributes such as their source of inspiration, the individuals in the population, way of selecting current best solution, ways to identify new solutions, ways to search better solutions and ways to abandon bad solutions.**

Keywords-Nature-Inspired Algorithm, Black Hole Algorithm, Artificial Bee Colony Algorithm, Bat Algorithm, Cuckoo Search Algorithm, Flower Pollination Algorithm, Grey Wolf Optimization Algorithm.

I. INTRODUCTION

Optimization is a mathematical problem that has been encountered ever since in all engineering disciplines. Literally it is described as the process of finding the best possible solution. Wide variety of important optimization problems are faced by researchers and engineers and so it has observed inclination as an active research topic [1]. Real-world engineering optimization problems are difficult to be solved, and most of them are NP-hard problems. Optimization tools are used to find optimal solutions for such problems. Actually, for NP-hard problems, there happens to be no single efficient solution. They are then resolved by hit and trial or brute force approach. In addition, new algorithms have been developed to cope up with these challenging optimization problems. Working on the same context meta-heuristic algorithms such as flower pollination, black hole, grey wolf etc. has gained popularity due to their high efficiency.

Optimization algorithms are broadly classified (on the basis of method of operation) as deterministic or probabilistic/randomized in nature. Deterministic optimization algorithms require large computational efforts and so these algorithms tend to fail as the size of the problem increases. Probabilistic or randomization optimization algorithms are computationally inexpensive alternatives to this deterministic approach. These algorithms include at least one instruction that is depending on a random numbers [2].

Heuristics are the functions which are when used for global optimization helps to select a solution from a set of possible solutions that are to be examined next. Deterministic optimization algorithms usually employ heuristics for the cause of identifying solution to the problem under consideration. Randomized optimization algorithms may use heuristic for selecting the elements of the search space for further computations. Meta-heuristics are algorithms ‘beyond’ or at ‘higher level’ than heuristics [3]. It is conventional these days that randomized optimization algorithms are considered meta-heuristic.

Meta-heuristics are becoming successful and popular due to their simplicity, ease for implementation as well as solution diversity. Most of these algorithms can be easily implemented and their code is generally of less than a hundred lines[4]. Along with this, these simple algorithms after being properly implemented can be subsequently used for handling a wide variety of optimization problems without much reprogramming. The main factor in the success of any meta-heuristic is to find the appropriate balance between the diversity of the solution and the computational efforts required to find the solution. Ideally, a meta-heuristic is desirable to obtain the global best solution with the maximum speed. Nature-inspired algorithms are most popular meta-heuristic algorithms observed these days. In this paper, some of the most commonly used and popular nature-inspired algorithms have been discussed and compared.

Rest of the paper is organized as follows. Section II disuses various classifications of nature-inspired computing. Section III is dedicated to discussion and comparison of six nature-inspired algorithms namely, Artificial Bee Colony Algorithm, Bat, Black Hole Algorithm, Cuckoo Search Algorithm, Flower Pollination Algorithm and Grey Wolf Optimization Algorithm. Section IV concludes the paper.

II. NATURE-INSPIRED COMPUTING

Nature has always been a source of inspiration for many researchers. Nowadays, solution to most of the problems is nature-inspired [3]. It is quite evident in recent optimization and clustering algorithms that are inspired by nature. The real beauty of such algorithms lies in the fact that these algorithms inculcate their sole inspiration from nature. These algorithms have the capacity to find solutions even when the user is having little or no knowledge about the search space. Even when the general source of inspiration of these algorithms is nature, they could be classified on the basis of their specific source such as biology, physics or chemistry [5]. It is a relatively new area with a brief history at this early stage of development. Even then as compared to the traditional and well-established technique these have still proved their great potential, flexibility and efficiency as well as ever-increasing and wide variety of applications. The theme behind the execution of these algorithms is simple even then the results are amazing and motivating. Nature is our best teacher forever and its capabilities are enormous and mysterious. Nature-Inspired algorithms have an enormous computational intelligence and capabilities and are observing diverse applications.

A. Classification of Nature-Inspired Algorithms

Nature-Inspired algorithms could be classified on the basis of inspiration as:

- Bio-inspired
- Physics/Chemistry Inspired
- Other Inspirations

A special subset of bio-inspired algorithms is swarm intelligence (SI) algorithms and most of the bio-inspired optimization algorithms belong to this category [6]. Further, bio-inspired algorithms are a subset of nature-inspired algorithms as shown in Figure 1. In other words, it could be said that not all nature-inspired algorithms are bio-inspired.

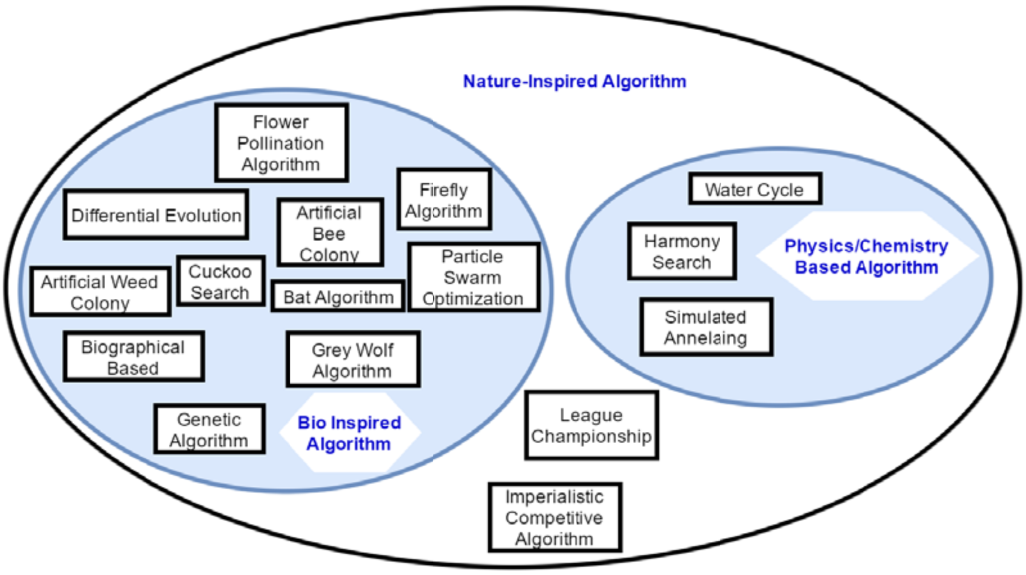

Figure 1. Classification of nature-inspired algorithms from set theory point of view

Some of the bio-inspired algorithms are not directly inspired by the swarming behaviour. So they are nature-inspired but not SI. As an instance, GA is bio-inspired, but not SI. Some of the nature-inspired algorithms have got inspiration from nature but not directly biology. They are found to be inspired from physics and chemistry. These algorithms have been developed by taking inspiration from certain physical and/or chemical laws such as electrical charges, gravity, river systems, etc. Further, there are certain algorithms such as Imperialistic Competitive algorithm which are nature-inspired but do not fit into any of the above mentioned categories [7].

Although nature-inspired algorithms are extensively used these days, but bio-inspired algorithms have been found to be more popular and vastly used. These are further classified as shown in Figure 2 [8]:

- Evolution based
- Swarm based
- Ecology based

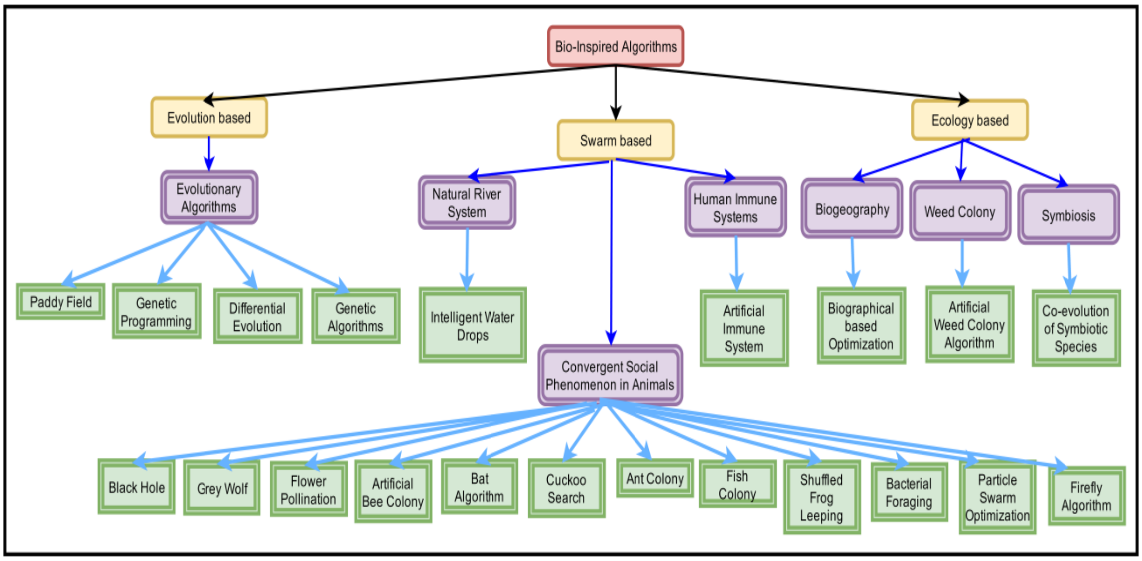

Figure 2. Taxonomy of Nature-inspired Algorithms grouped by the area of inspiration

One of the most populated classes of bio-inspired algorithms is Swarm Intelligence algorithms. Black Hole [9], Grey wolf [10], Flower Pollination [11], Cuckoo Search [12], Bat [13], Artificial Bee Colony [14], Firefly [15], Particle Swarm Optimization, Bacterial Foraging [16], are the names of few of these algorithms.

Swarm is a term that is used to represent aggregation of animals (fish schools, bird's flocks and insect colonies, bee colonies) that are performing collective behaviour. Each individual agent in the swarm behaves in a stochastic manner, without any supervision and with the perception in the neighbourhood. They follow some local rules, interact in a self organised manner and leads to a global pattern showing collective intelligence called swarm intelligence. This collective intelligence provides them the ability to intelligently and efficiently use their environment and resources. The key feature that leads to the concept of swarm intelligence is self-organisation. It leads to global response by performing local interactions. They have the ability to learn during iteration and have the ability of parallelism due to the use of multiple agents. This parallelism of agents helps in easy implementation of large scale optimization.

General steps to execute any nature-inspired algorithm are described in block diagram shown in Figure 3.

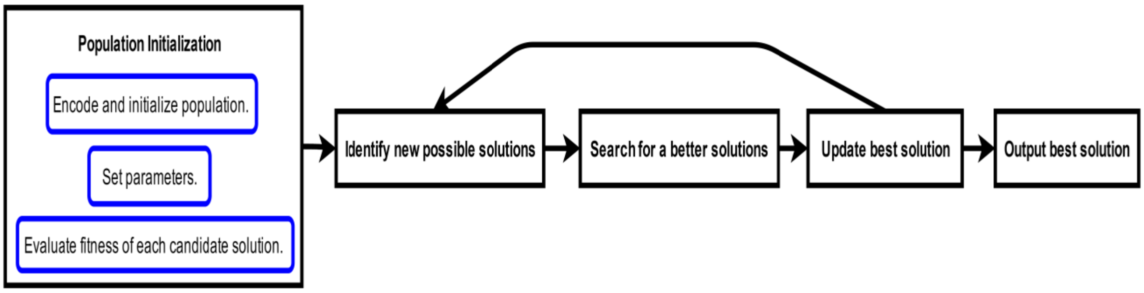

Figure 3. Block Diagram of Nature-Inspired Algorithms

B. *Global Vs local optimization*

Two major components required for the successful implementation of the met-heuristic are:

- Diversification (also called exploration or global optimization)
- Intensification (also called exploitation or local optimization).

Diversification means to explore the search space globally and produce diverse solutions whereas intensification means to focus the search to local space and exploit the local information to find current local best solution. A good optimization algorithm should have both diversification and intensification which leads to better exploration of the search space and should converge quickly. Intensification assures the convergence by selection best possible solution and diversification adds randomization in the algorithm and avoids getting trap at local optima.

C. Single-Objective Vs Multi-Objective optimization

Optimization means to identify best possible solution to a given problem. Another classification of optimization algorithm is on the basis of number of objectives to be optimized. The algorithm could be

- Single objective optimization algorithm
- Multi-objective optimization algorithm

In case of Single objective optimization, the main goal of optimization is to find the best solution having best value of a single objective function. The best value could be minimum or maximum depending on the problem under consideration

III. DESCRIPTION OF NATURE-INSPIRED ALGORITHMS

In this paper, the main inclination is towards swarm-based algorithms namely Artificial Bee Colony (ABC) algorithm, Bat algorithm, Black Hole (BH) algorithm, Flower Pollination algorithm (FPA), Cuckoo Search (CS), Grey Wolf Optimization algorithm (GWO).

A. Artificial Bee Colony

Artificial Bee Colony (ABC) [17-18] algorithm inspired by the intelligent foraging behaviour of honey bee swarms. Each individual in the population is known as a *bee*. The population of random candidate solutions is initialized. According to this algorithm, the colony of honey bees contains three groups of bees: employee, onlooker and scout bees.

Employee bee is associated with a particular food source and carries information about it. *Onlooker bee* waits in the dance area for making a decision to choose a food source on the basis of information shared by employee bees. *Scout bee* goes on a random search to discover new sources. If a candidate solution can't be improved further during a pre-specified number of cycles called limit, then that candidate is replaced by a new food source. This newly created solution is compared to existing solutions and best solution achieved so far is memorized.

Next iteration is started (cycle = cycle + 1) until the stopping criteria is met. The first half of the population acts as employee bees and other as onlooker bees. The algorithm improves the solutions under consideration by using neighbour search mechanism. Poor solutions are abandoned. The quality of a candidate solution is compared to nectar amount of flowers discovered by bees. So, more the nectar of the flower better is the attraction of bees towards it. Food sources that can't be improved are considered scout and are abandoned. Fresh food source and hence candidate solutions are generated in place of them to maintain population size. The flowchart for ABC algorithm is shown in Figure 4.

The main features of this algorithm include:

- The position of a food source represents a candidate solution to the problem.
- The amount of nectar associated with a food source represents the quality (fitness) of the underlying solution.
- The number of employee bees is equal to the number of food sources.
- Each employed bee is associated with one and only one food source.

The major applications of ABC [19]includes lot streaming flow shop scheduling problem [20], dynamic deployment of wireless sensor networks [21] etc.

B. BAT Algorithm

BAT algorithm [13, 22-23] is a meta-heuristic global optimization nature-inspired algorithm that has been inspired by a bat's capability of echolocation to detect prey and avoid obstacles in the dark. Each individual in the population is called a *bat*. On the basis of search space (having dimensions equal to the number of decision variables to be optimized), velocity of all the bats are initialized. Pulse rate (r) and loudness (A) (required to build three-dimensional scenario of the surrounding) of all the bats are initialized to random numbers between 0 and 1. The flowchart for BAT algorithm is shown in Figure 5.

Loudness decreases once a bat has found its prey whereas pulse rate emission increases. Each bat depicts a possible candidate solution. It is assumed that each bat (taken as a possible solution) flies in a random direction with a velocity v_i at position x_i (possible solution) with a varying frequency or wavelength and particular loudness A_i. Search is based on local random walk.

It has observed wide variety of application including image matching [24], optimization of power system stabilizers [25] etc.

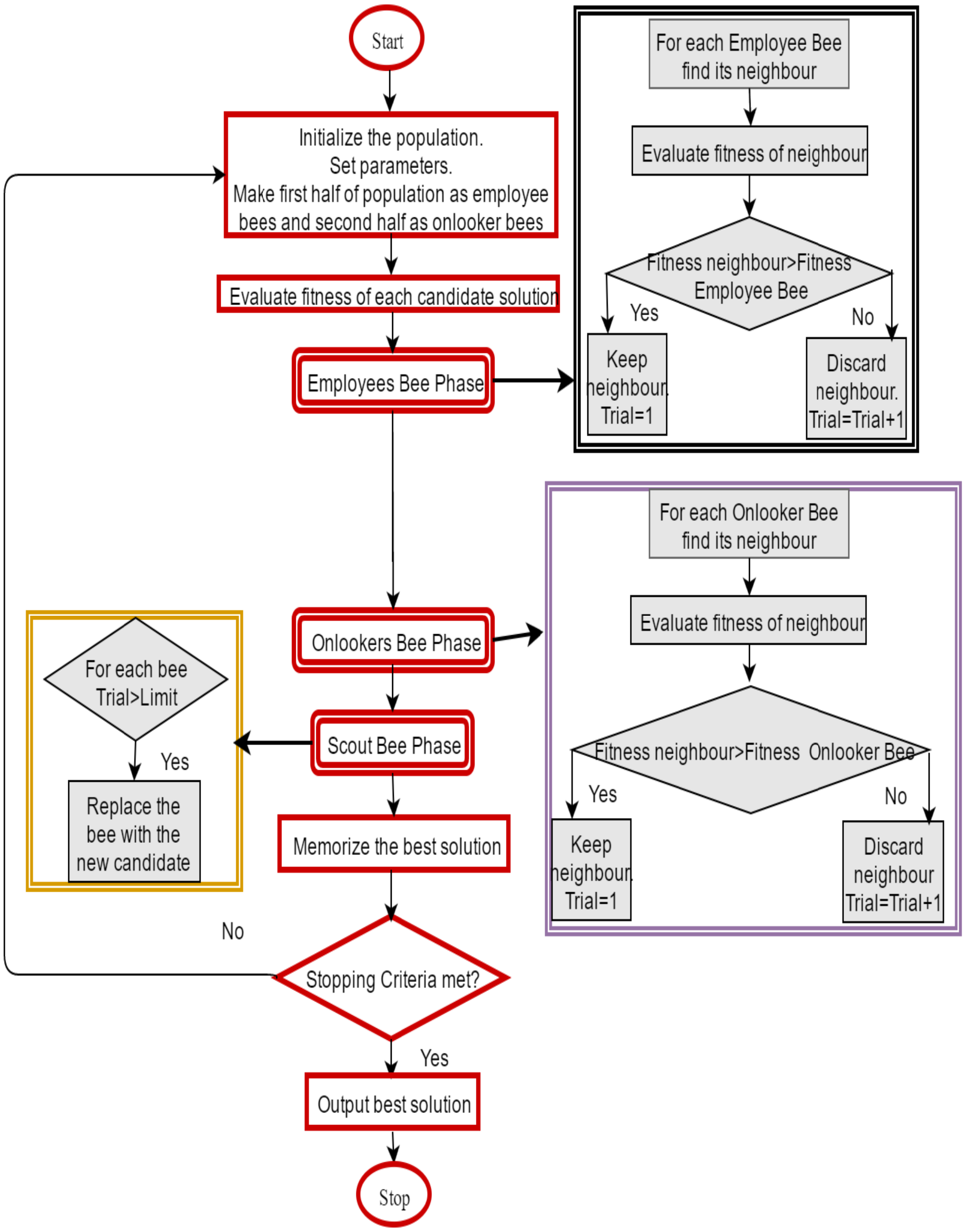

Figure 4. Flowchart for Artificial Bee Colony algorithm

C. *Black Hole Algorithm*

This algorithm is inspired by the black hole theory of universe [26]. Black hole is a region of space that has so much mass concentrated in it that there is no way for a nearby object to escape its gravitational pull. If something falls into a black hole, it is believed to be vanished from the universe [9]. The algorithm is discussed below. It is a population based algorithm which starts with initial population of candidate solutions. Fitness of each is calculated and the star (candidate solution) with least value of objective function is considered as Black Hole (x_{BH}) [27].

The algorithm repeats until the stopping criteria is met. New location ($x_i(t+1)$) of each star ($x_i(t)$) is identified as shown below

$$x_i(t+1)=x_i(t)) + rand*(x_{BH} - x_i(t))$$

If new candidate solution is better than the current candidate solution, then replace the current solution with this new solution else ignore it. This step is required to locally search for a better sequence. It moves the current candidate randomly in search for a better solution. If the new solution is better than the current Black Hole (x_{BH}), then designate this new solution as new Black Hole (x_{BH}). Calculate the radius of the event of horizon (R) of the Black Hole.

$$R = F_{BH} / \sum_{i=1}^{Pop} F_i$$

Where F_{BH} is the fitness for Black Hole and F_i is the fitness of i^{th} candidate solution. *Pop* is the size of the population under consideration. If a star enters this event horizon, it is absorbed by the Black Hole and a new candidate is generated to compensate the total population. It has been applied for scheduling of thermal power systems [28], optimal power flow [29] etc. The flowchart for Black Hole (BH) algorithm is shown in Figure 6.

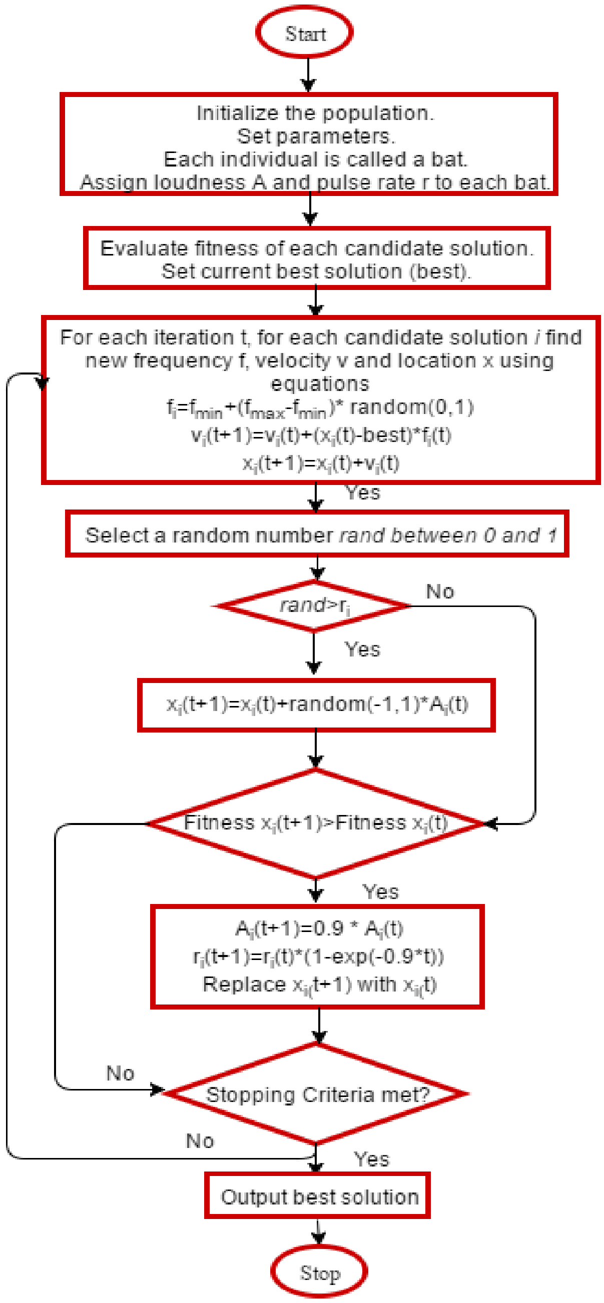

Figure 5. Flowchart for BAT algorithm

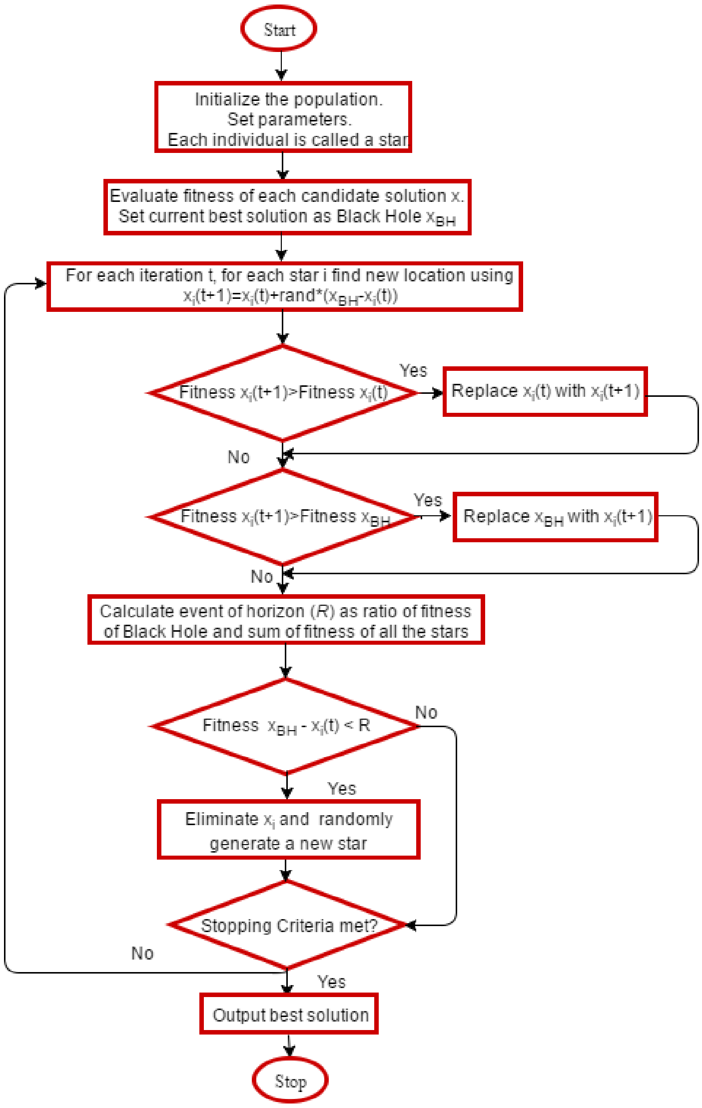

Figure 6. Flowchart for Black Hole algorithm

D. *Cuckoo Search Algorithm*

Cuckoo Search (CS) algorithm [12, 30] is inspired by the brood parasitism of some cuckoo species. Cuckoo is one of the most interesting birds due to their aggressive reproduction strategy [31]. Some of its species performs the obligate brood parasitism which means the bird lays their eggs in the nests of other host birds which might be of some other species. In some cases, the host birds have direct conflict with the intruding cuckoos. In case the host bird discovers the intruder eggs, they either throw these eggs or simply abandon its nest.

Some of the assumptions of Cuckoo Search (CS) algorithms are discussed below [23, 32].

- Single egg is being laid by the cuckoo at a time and that too is placed in randomly chosen nest.
- The egg with highest quality is carried over to next generation.
- The number of available host nests is fixed.
- The host bird discovers the egg of intruder bird with probability pa ∈ [0, 1].

The algorithm developed on the basis of the theory of cuckoo birds is discussed below.

1. Encode and initialize the population of possible candidate solutions. Each individual is called a *nest*.
2. Define abandon fraction (*pa*) and set parameters. Calculate fitness of each candidate and select the current best solution.
3. *S*elect a candidate solution *i* by using step vector drawn from Levy distribution (Yang and Deb 2009).
4. Randomly select another candidate solution *j*.
5. If candidate solution *i*, x_i is better than x_j, replace x_j with x_i.

6. Abandon fraction of solutions (*pa*) and generate new to balance the population. Keep the best clustering.

Cuckoo search has proved to be one of the widely used solution for optimization problems such as scheduling optimization of flexible manufacturing systems [33], for economic dispatch [34]. The flowchart is shown in Figure 7.

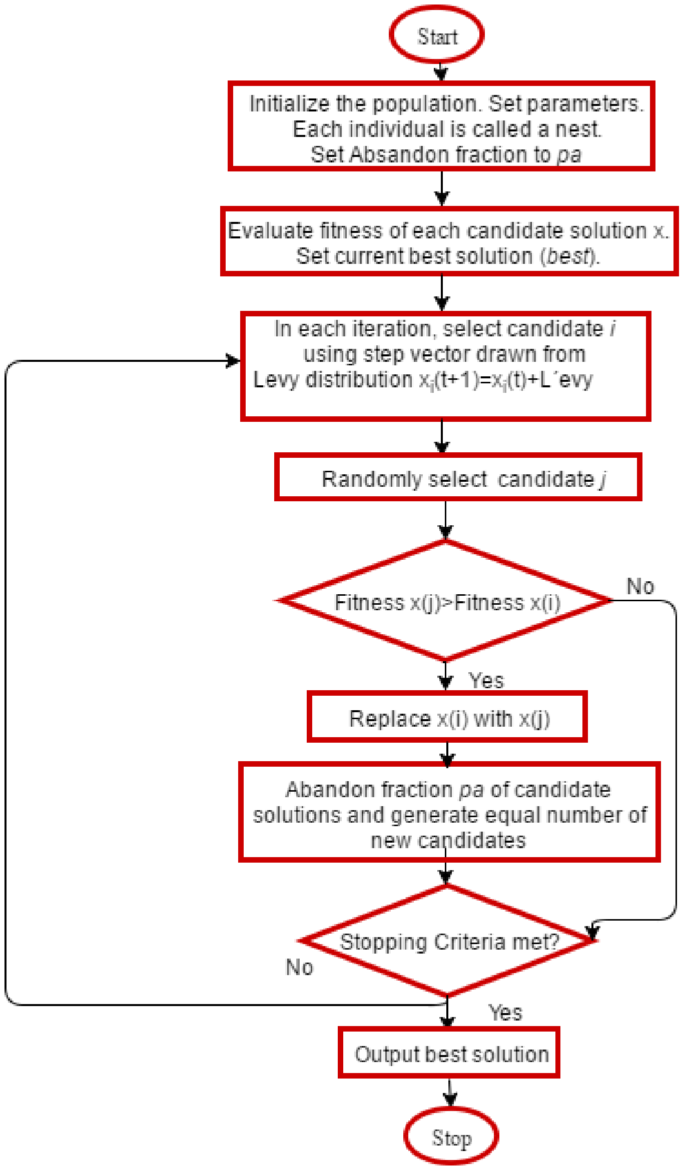

Figure 7. Flowchart for Cuckoo Search algorithm

E. *Flower Pollination Algorithm*

Flower Pollination algorithm (FPA) is based on the process of transfer of flower pollens. Flowering plants are evolving from past million years. The two main forms of pollinations are Abiotic and biotic pollination. It is said that about 90% of flowering plants use biotic pollination in which pollen is communicated by an agent such as insects and animals and about 10% of agents takes abiotic form in which no agent is required and use wind and diffusion performs pollination [35]. Pollination can be achieved by self-pollination or crosspollination ot local pollination. Cross-pollination, also called allogamy is a kind of pollination in which pollination can happen from pollen of a flower of a different plant. In self-pollination, pollination takes place from pollen of the same flower or different flowers of the same plant. Crosspollination can take place at a long distance by means of bees, bats, birds and flies as they can fly a long distance and help in pollination. It is also called global pollination. Bees and birds can follow L´evy flight behaviour [36], with fly distance steps obeying a L´evy distribution [11, 37]. The use of abiotic and self-pollination is called local pollination. The process of flower pollination can occur at global as well as local level. The neighbouring flowers could be pollinated by local pollen with higher probability than those far away. The flowchart is shown in Figure 8.

The probability of switching of local and global pollination has been controlled by switch probability $p \in [0,1]$. The algorithm works as follows.

1. Encode and initialize the population of possible candidate solutions (*x*). Each individual clustering is called *flower*. Define switching probability (*p*) and set other parameters.
2. Select current best candidate (best)
3. Randomly perform search. Generate a random number *r*.
4. If $r > p$

   //Perform global search

   $x_i(t+1)=x_i(t) + L\,(best - x_i(t))$ where *L* is step vector following Levy distribution, *t* is the current iteration.

   Else

   //Perform local search

   $x_i(t+1)=x_i(t) + \varepsilon\,(x_a(t) - x_b(t))$, where $x_a(t)$ and $x_b(t)$ are randomly selected candidate solutions, ε is drawn from uniform distribution between 0 and 1.
5. If new solution $x_i(t+1)$ is better than that of $x_i(t)$, then replace $x_i(t)$ with $x_i(t+1)$.

As discussed above, FPA and CS algorithms are based on Lévy flight which is considered powerful than a random walk. In case of Lévy flights both global and local search capabilities can be carried out simultaneously. Lévy flights occasionally perform Lévy steps [38] that help the algorithm to get rid of local valleys.

Lévy step is depicted as shown in below.

$$L \approx 1/s^{(1+\beta)}$$

where β refers to the Lévy exponent. Parameter *s*, *u* and *v* are described in Equations 1.5 and 1.6.

$$s = u/|v|^{(1+\beta)}$$

$$u \approx N(0,\sigma^2), v \approx N(0,1)$$

where σ is a function of β..

F. Grey Wolf Optimizer Algorithm

Grey Wolf Optimizer (GWO) algorithm is inspired from grey wolves those are predators at the top of the food chain. Grey wolf are considered as apex predators and are observed to belong to Canidae family. They have the tendency to live in a pack of 5 to 12 wolves. The flowchart is shown in Figure 9. The Grey wolf community follows following hierarchy.

The *alphas* (x_α) (male and females) are the leaders. They take decisions regarding hunting, sleeping place etc. The whole pack acknowledges the decisions made by alpha by holding their tails down [39]. Only they are allowed to mate in the pack.

The *betas* (x_β) (male or female) are at second level in the hierarchy of grey wolves. They have the duty to help the alpha in decision-making or other pack activities. They happen to be the best candidate to be the alpha in case one of the alpha could not perform its duties. They command the other lower-level wolves but obey the alpha. They act as a bridge between alpha and rest of the pack especially in conveying alphas decisions. The omega (x_γ) (scapegoat) falls at lowest rank in the hierarchy of grey wolf. They have to obey all other wolves and are allowed to eat in the end. Sometimes they act as babysitters in the pack.

Delta wolves have to obey to alphas and betas and are dominated by the omega. Scouts, elders, hunters, caretakers has been allocated to this category. Scouts have the duty to warn the pack in case of any danger. Sentinels are responsible for the protection of the pack. Alphas and betas that become old falls in the category called Elders. Hunters have the duty to help the alphas and betas while hunting. The caretaker wolves have the responsibility to take care of the weak, ill, and wounded wolves in the pack. The main phases of their hunting mechanism are:

- Tracking, chasing, and approaching the prey
- Pursuing, encircling, and harassing the prey until it stops moving
- Attack towards the prey.

Some of the important things considered while implementing GWO algorithm are as follows:

- The hierarchy of wolves is exploited to keep the best solutions obtained so far over each execution. The encircling mechanism involved in implementation defines a circle-shaped neighbourhood around the solutions also called hyper-sphere. *A* and *C* are randomly selected to maintain hyper-spheres with different random radii.

- The theme of any search based optimization algorithm is exploration and exploitation. The adaptive values of *a* and *A* helps in implementing and transition between exploration and exploitation.

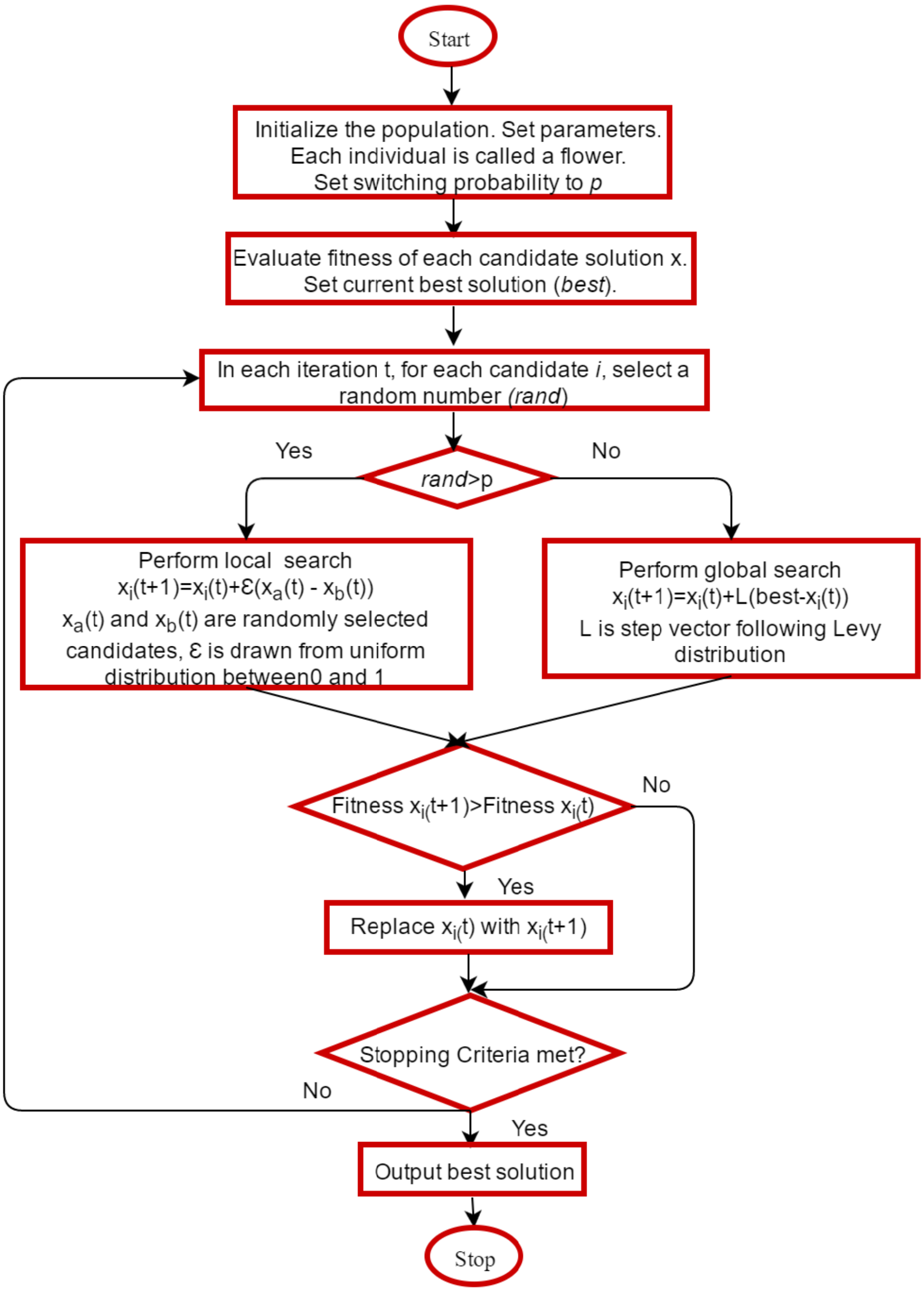

Figure 8. Flowchart for Flower Pollination algorithm

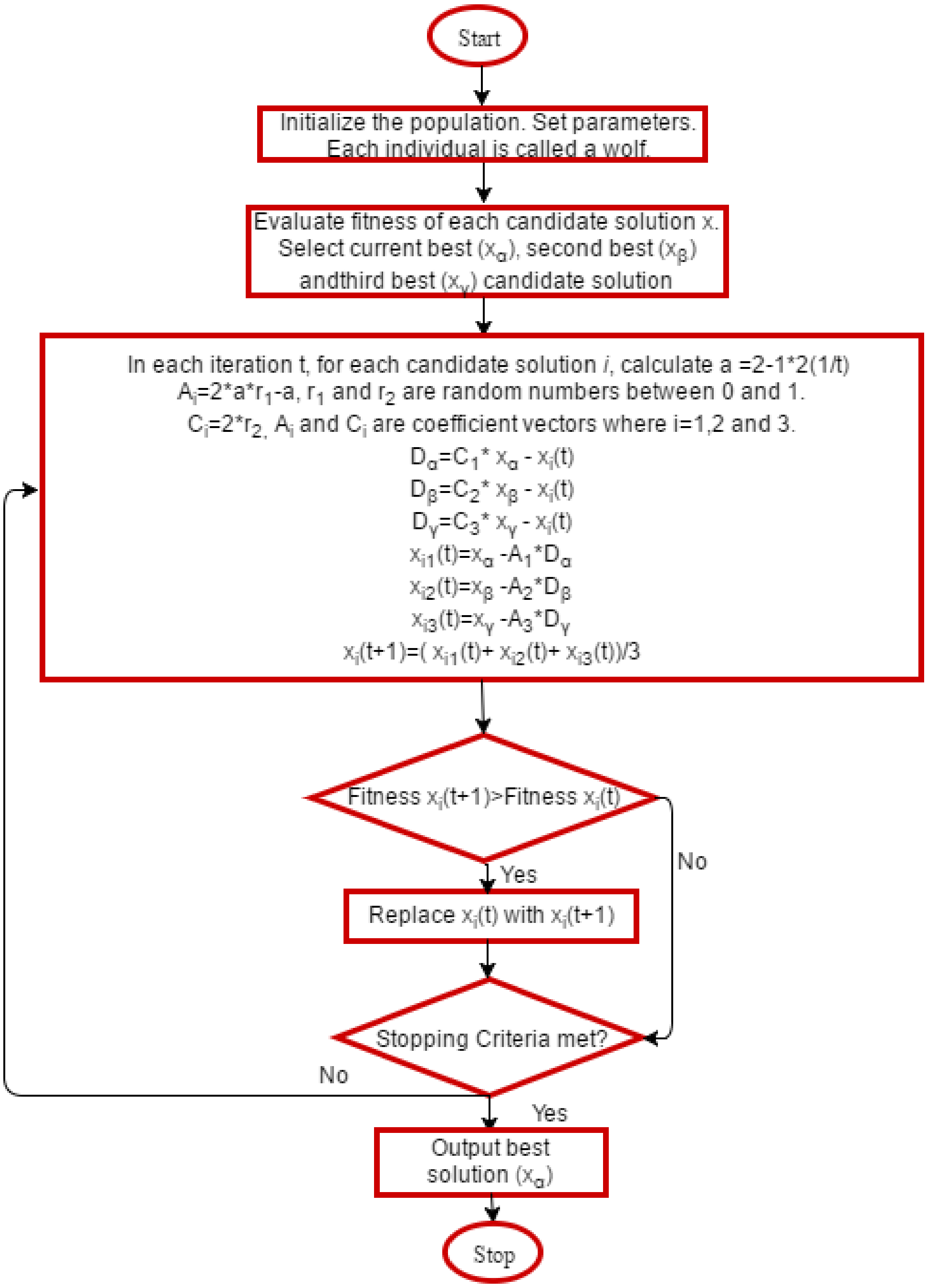

Figure 9. Flowchart for Grey Wolf Optimizer algorithm

The major steps involved in implementing GWO are discussed below.

1. Encode and initialize the population of possible candidate solutions. Each individual is called a *grey wolf*.
2. Select current best (x_α), second best (x_β) and third best (x_γ) solutions on the basis of their fitness.
3. For each candidate solution x_i exploration and exploitation is implemented as

   a =2-1*2(1/t) t is the current iteration.

   A_i=2*a*r_1-a r_1 and r_2 are random numbers between 0 and 1.

   C_i=2*r_2 A_i and C_i are coefficient vectors where i=1,2 and 3.

   $D_\alpha = C_1 * x_\alpha - x_i(t)$

   $D_\beta = C_2 * x_\beta - x_i(t)$

   $D_\gamma = C_3 * x_\gamma - x_i(t)$

   $x_{i1}(t) = x_\alpha - A_1 * D_\alpha$

$x_{i2}(t)=x_\beta - A_2 * D_\beta$
$x_{i3}(t)= x_\gamma - A_3 * D_\gamma$
$x_i(t+1)=(x_{i1}(t)+ x_{i2}(t)+ x_{i3}(t))/3$

4. If fitness $x_i(t+1)$ is better than $x_i(t)$ then replace $x_i(t)$ with $x_i(t+1)$.
5. Update x_α, x_β and x_γ.

Grey wolf optimizer has been used for wide variety of problems such as parameter estimation in surface waves [40], two stage assembly flow shop scheduling [41] etc.

Table I compares the nature-inspired algorithms under consideration. The algorithms are compared on the basis of their source of inspiration, the name of each individual in the population, way of selecting current best solution, ways to identify new solutions, ways to search better solutions and ways to abandon solutions.

TABLE I. COMPARISON OF NATURE-INSPIRED ALGORITHMS

Criteria	ABC	BAT	BH	CS	FPA	GWO
Inspiration	Foraging behaviour of honey bees	Echolocation used by bats to search their prey	Black hole theory of the universe	Parasitic behaviour of some cuckoo species	Pollination of flowering plants	Hunting behaviour of grey wolves
Individual	Bee	Bat	Star	Nest	Flower	Wolf
Current Best solution	No such mechanism	Fittest bat	Black Hole	Fittest nest	Fittest flower	Alpha x_α, Beta x_β, Delta x_γ
Identify new solution	Identify new food source in the neighbour	Update frequency, velocity and location of each bat	Black hole attracts all other stars and changes their position	Select nest i by using step vector drawn from Lévy distribution	If a randomly generated number r is greater than switching probability p perform global search else perform local search	Population of wolves update their positions on the basis of the position of x_α, x_β and x_γ.
Search for a better solution	If new food source is better; then replace it	If a randomly generated number is less than Loudness of the bat and new location is better, then keep this new location	If the new solution is better than x_{BH}, then designate this new solution as new Black Hole	Randomly select a solution j. If j is better than i then exchange them	If new flower is better than replace it.	From the updated population of grey wolves, update x_α, x_β and x_γ.
Abandon solution	Abandon solutions if can't be improved	No such mechanism	Abandon star if it enters event of horizon of x_{BH}	Abandon fraction of candidates (pa)	No such mechanism	No such mechanism

IV. CONLCUSION

Optimization problems are widely found in all engineering disciplines. It is described as the process of finding the best possible solution. Real-world engineering optimization problems are difficult to be solved, and most of them are NP-hard problems. Heuristics are the methods which are used for global optimization and help to select a solution from a set of possible solutions that are to be examined next. Meta-heuristics are algorithms 'beyond' or at 'higher level' than heuristics. They are becoming successful and popular due to their simplicity, ease for implementation as well as solution diversity. Further, nature-inspired algorithms are meta- heuristic algorithms that are widely used for engineering optimization and are developed by taking inspiration from nature. In this paper, various kinds of classifications of nature-inspired algorithms have been discussed. Six popular nature-inspired algorithms are discussed along with their flowcharts and compared on the basis of attributes such as source of inspiration, individuals in the population, selection of current best solution, identification of new solutions, search for better solutions and abandonment of bad solutions.

REFERENCES

[1] I. Fister Jr, et al., "A brief review of nature-inspired algorithms for optimization," arXiv preprint arXiv:1307.4186, 2013.
[2] T. Weise, "Global optimization algorithms-theory and application," Self-Published, 2009.
[3] X.-S. Yang, Nature-inspired metaheuristic algorithms: Luniver press, 2010.
[4] I. H. Osman and J. P. Kelly, Meta-heuristics: theory and applications: Springer Science & Business Media, 2012.
[5] S. J. Nanda and G. Panda, "A survey on nature inspired metaheuristic algorithms for partitional clustering," Swarm and Evolutionary computation, vol. 16, pp. 1-18, 2014.
[6] A. E. Hassanien and E. Alamry, Swarm Intelligence: Principles, Advances, and Applications: CRC Press, 2015.
[7] X.-S. Yang, et al., Swarm intelligence and bio-inspired computation: theory and applications: Newnes, 2013.
[8] H. M. Dubey, et al., "Bio-inspired optimisation for economic load dispatch: a review," International Journal of Bio-Inspired Computation, vol. 6, pp. 7-21, 2014.
[9] A. Hatamlou, "Black hole: A new heuristic optimization approach for data clustering," Information Sciences, vol. 222, pp. 175-184, 2013.
[10] S. Mirjalili, et al., "Grey wolf optimizer," Advances in Engineering Software, vol. 69, pp. 46-61, 2014.
[11] X.-S. Yang, "Flower pollination algorithm for global optimization," in Unconventional computation and natural computation, ed: Springer, 2012, pp. 240-249.
[12] X.-S. Yang and S. Deb, "Cuckoo search via Lévy flights," in World Congress on Nature & Biologically Inspired Computing, NaBIC 2009 2009, pp. 210-214.
[13] X.-S. Yang, "A new metaheuristic bat-inspired algorithm," in Nature inspired cooperative strategies for optimization (NICSO 2010), ed: Springer, 2010, pp. 65-74.
[14] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," Journal of global optimization, vol. 39, pp. 459-471, 2007.
[15] X.-S. Yang, "Firefly algorithm, stochastic test functions and design optimisation," International Journal of Bio-Inspired Computation, vol. 2, pp. 78-84, 2010.
[16] K. M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control," Control Systems, IEEE, vol. 22, pp. 52-67, 2002.
[17] R. Akbari, et al., "A multi-objective artificial bee colony algorithm," Swarm and Evolutionary Computation, vol. 2, pp. 39-52, 2012.
[18] X. Yan, et al., "A new approach for data clustering using hybrid artificial bee colony algorithm," Neurocomputing, vol. 97, pp. 241-250, 2012.
[19] D. Karaboga, et al., "A comprehensive survey: artificial bee colony (ABC) algorithm and applications," Artificial Intelligence Review, vol. 42, pp. 21-57, 2014.
[20] Q.-K. Pan, et al., "A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem," Information sciences, vol. 181, pp. 2455-2468, 2011.
[21] C. Öztürk, et al., "Artificial bee colony algorithm for dynamic deployment of wireless sensor networks," Turkish Journal of Electrical Engineering & Computer Sciences, vol. 20, pp. 255-262, 2012.
[22] X.-S. Yang, "Bat algorithm for multi-objective optimisation," International Journal of Bio-Inspired Computation, vol. 3, pp. 267-274, 2011.
[23] X.-S. Yang and X. He, "Bat algorithm: literature review and applications," International Journal of Bio-Inspired Computation, vol. 5, pp. 141-149, 2013.
[24] J. W. Zhang and G. G. Wang, "Image matching using a bat algorithm with mutation," in International Conference on Applied Mechanics and Materials, 2012, pp. 88-93.
[25] E. Ali, "Optimization of power system stabilizers using BAT search algorithm," International Journal of Electrical Power & Energy Systems, vol. 61, pp. 683-690, 2014.
[26] I. Fister, et al., "Swarm and evolutionary computation," Swarm Evolutionary Comput, vol. 10, 2013.
[27] M. Nemati, et al., "Black Holes Algorithm: A Swarm Algorithm inspired of Black Holes for Optimization Problems," IAES International Journal of Artificial Intelligence, vol. 2, p. 143, 2013.
[28] R. Azizipanah-Abarghooee, et al., "Short-term scheduling of thermal power systems using hybrid gradient based modified teaching–learning optimizer with black hole algorithm," Electric Power Systems Research, vol. 108, pp. 16-34, 2014.
[29] H. Bouchekara, "Optimal power flow using black-hole-based optimization approach," Applied Soft Computing, vol. 24, pp. 879-888, 2014.
[30] X.-S. Yang and S. Deb, "Multiobjective cuckoo search for design optimization," Computers & Operations Research, vol. 40, pp. 1616-1624, 2013.
[31] R. Payne, et al., "The cuckoos, bird families of the world," ed: Oxford University Press, New York, 2005.
[32] X.-S. Yang and S. Deb, "Cuckoo search via Lévy flights," in Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on, 2009, pp. 210-214.
[33] S. Burnwal and S. Deb, "Scheduling optimization of flexible manufacturing system using cuckoo search-based approach," The International Journal of Advanced Manufacturing Technology, vol. 64, pp. 951-959, 2013.
[34] M. Basu and A. Chowdhury, "Cuckoo search algorithm for economic dispatch," Energy, vol. 60, pp. 99-108, 2013.
[35] B. J. Glover, Understanding flowers and flowering: an integrated approach vol. 277: Oxford University Press Oxford, UK, 2007.
[36] I. Pavlyukevich, "Lévy flights, non-local search and simulated annealing," Journal of Computational Physics, vol. 226, pp. 1830-1844, 2007.
[37] X.-S. Yang, et al., "Flower pollination algorithm: a novel approach for multiobjective optimization," Engineering Optimization, vol. 46, pp. 1222-1237, 2014.
[38] X.-S. Yang and S. Deb, "Cuckoo search: recent advances and applications," Neural Computing and Applications, vol. 24, pp. 169-174, 2014.
[39] L. D. Mech, "Alpha status, dominance, and division of labor in wolf packs," Canadian Journal of Zoology, vol. 77, pp. 1196-1203, 1999.
[40] X. Song, et al., "Grey Wolf Optimizer for parameter estimation in surface waves," Soil Dynamics and Earthquake Engineering, vol. 75, pp. 147-157, 2015.
[41] G. M. Komaki and V. Kayvanfar, "Grey Wolf Optimizer algorithm for the two-stage assembly flow shop scheduling problem with release time," Journal of Computational Science, vol. 8, pp. 109-120, 2015.