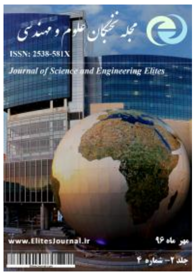
www.ElitesJournal.ir

مجله نخبگان علوم و مهندسی

Journal of Science and Engineering Elites

ISSN 2538-581X

جلد ۲- شماره ۴- سال ۱۳۹۶




# An overview of the Newest Intelligent Optimization Methods in Electrical Engineering

**Behrouz Najafi [1*]**

**1. Department of Electrical Engineering,, Islamic Azad University , Gonabad branch, and Young Researchers Club, Kermanshah**

* behroz1.najafi1@gmail.com

Received: September 2017 Accepted: October 2017

**Abstract**

In this paper we give an overview of newest intelligent optimization methods in electrical engineering., In an optimization problem, the types of mathematical relationships between the objective function and constraints and the decision variables determine how hard it is to solve, the solution methods or algorithms that can be used for optimization, and the confidence we can have that the solution is truly optimal. In this work, most of the algorithms and applications about Intelligent optimization have been collected and described. It is explained how to solve each optimization problem, step by step with the simplest possible form.

**Keywords:** Optimization, Optimization Methods, global optimum, decision variable, best solution

## 1. Introduction

Optimization is the process of making something better. In other words, optimization is the process of adjusting the inputs to or characteristics of a device, mathematical process, or experiment to find the minimum or maximum output or result. The input consists of variables: the process or function is known as the cost function, objective function, or fitness function; and the output is the cost or fitness . There are different methods for solving an optimization problem. Some of these methods are inspired from natural processes. These methods usually start with an initial set of variables and then evolve to obtain the global minimum or maximum of the objective function [1].

Classical optimization algorithms are insufficient in large scale combinatorial problems and in nonlinear problems. Hence, metaheuristic optimization algorithms have been proposed. General purpose metaheuristic methods are evaluated in nine different groups: biology-based, physics-based, social-based, music-based, chemical-based, sport-based, mathematics-based, swarm-based, and hybrid methods which are combinations of these. Idea of evolutionary computing was introduced in the 1960s by I. Rechenberg in his work "Evolution strategies",His idea was then developed by other researchers.

There are some important developments in recent years, and this special issue aims to provide a timely review of such developments, including ant colony optimization, genetic algorithms, cuckoo search, particle swarm optimization, , neural networks, and others. The main advantages of evolutionary algorithms are [2]:

Being robust to dynamic changes: Traditional methods of optimization are not robust to dynamic changes in the environment and they require a complete restart for providing a solution. In contrary, evolutionary computation can be used to adapt solutions to changing circumstances.
Broad applicability: Evolutionary algorithms can be applied to any problems that can be formulated as function optimization problems.
Hybridization with other methods: Evolutionary algorithms can be combined with more traditional optimization techniques.
Solves problems that have no solutions: The advantage of evolutionary algorithms includes the ability to address problems for which there is no human expertise. Even though human expertise should be used when it is needed and available; it often proves less adequate for automated problem-solving routines.
In mathematics and computer science, an optimization problem is the problem of finding the best solution from all feasible solutions. Optimization problems can be divided into two categories depending on whether the variables are continuous or discrete. An optimization problem with discrete variables is known as a combinatorial optimization problem. In a combinatorial optimization problem, we are looking for an object such as an integer, permutation or graph from a finite (or possibly countable infinite) set. Problems with continuous variables include constrained problems and multimodal problems[1].
In [21],Has been studied Unconstrained Optimization and constrained Optimization. Optimization Concepts and Applications in Engineering Has been studied In [22]. Genetic Algorithms and Engineering Optimization was discussed in [23]. also In [24], Practical Methods of Optimization Volumes is explained. Linear and Nonlinear Programming, Which is widely used in engineering branches, is presented in [25]. Optimization Methods in [26] - describes how optimization problems can be solved and which different types of optimization methods exist for discrete optimization problems. in [27] Has been studied Survey of Multi-Objective Optimization Methods for Engineering. Also In this paper a survey of current continuous nonlinear multi-objective optimization (MOO) concepts and methods is presented. That consolidates and relates seemingly different terminology and methods. In this paper, The methods are divided into three major categories: methods with a priori articulation of preferences, methods with a posteriori articulation of preferences, and methods with no articulation of preferences. Genetic algorithms are surveyed as well. Commentary is provided on three fronts, concerning the advantages and pitfalls of individual methods, the different classes of methods, and the field of MOO as a whole.
In [3] we have investigated the following optimization methods:
Genetic Algorithm (GA)-Particle Swarm Optimization (PSO)- Simulated Annealing (SA)- Ant Colony Optimization (ACO)- Artificial Bee Colony (ABC)- Firefly Algorithm (FA)- Tabu Search (TS)- Cuckoo Search algorithm (CS)- Artificial Neural Network (ANN).
In this article we will study other optimization methods, That They are newer and more interesting. These methods are investigated in the second part.

## 2. Intelligent (Metaheuristics) optimization algorithms

### 2.1. Imperialist Competitive Algorithm (ICA):

ICA as an evolutionary optimization method is inspired by imperialistic competition and is based on the behavior of imperialists in their attempt to overcome colonies. Similar to other evolutionary algorithms,ICA begins with an initial population and is separated into two types: the colonies and the imperialists (the ones with the best objective function values). All of these colonies, based on their power, are shared among these imperialists. The power of each country is proportionate to its cost in reverse, i.e. the more powerful an imperialist is, the more colonies it will dominate [4-5].
during imperialistic competition, the most authoritative empires tend to enhance their power, whereas the weaker ones tend to fall in. These two mechanisms result the algorithm to progressively converge into a condition in which there exists only one empire and colonies have the same objective function [5].
The iterative approach of ICA can be described by the following steps [5]:

**Step 1:** Select some random points on the function and initialize the empires.
**Step 2:** Move the colonies toward their relevant imperialist (Assimilation).
**Step 3:** Randomly change the position of some colonies (Revolution).
**Step 4:** If there is a colony in an empire which has lower cost than the imperialist, exchange the positions of that colony and the imperialist.
**Step 5:** Unite the similar empires.
**Step 6:** Compute the total cost of all empires.
**Step 7:** Pick the weakest colony (colonies) from the weakest empires and give it (them) to one of the empires (imperialistic competition).

**Step 8:** Eliminate the powerless empires.
**Step 9:** If stop conditions are satisfied, stop, if not go to 2.

Based on what was said,The flowchart of the ICA is shown in Figure 1.

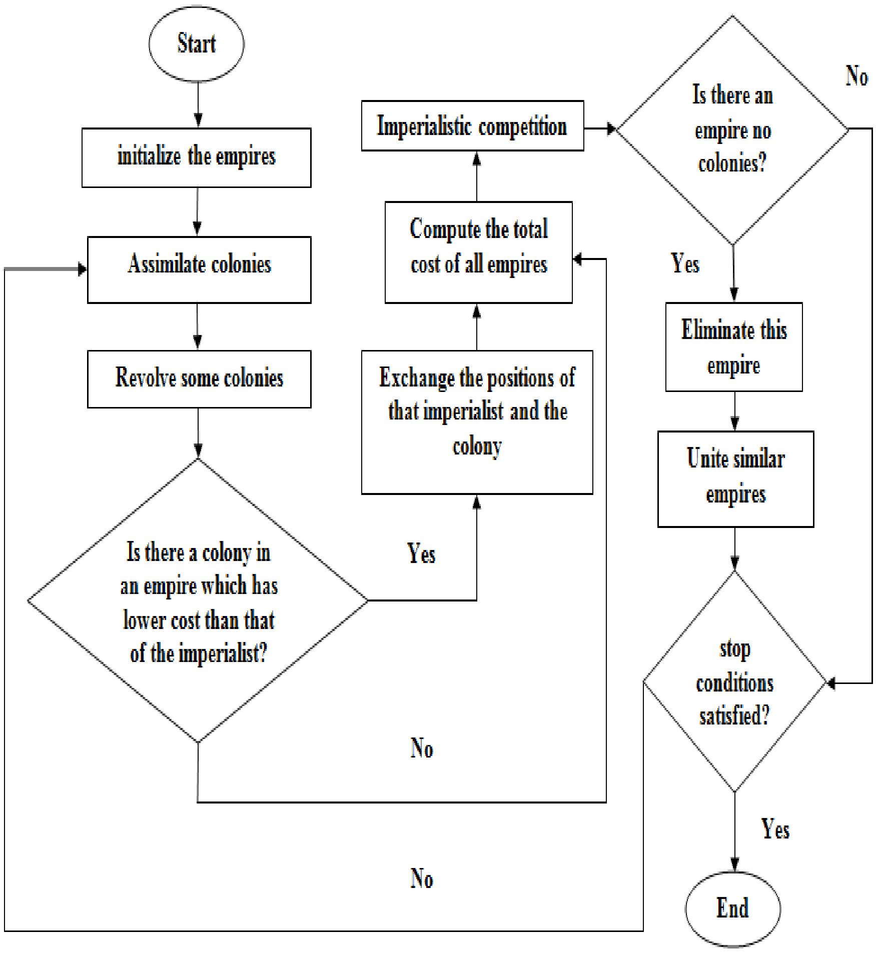



**Figure 1- Flowchart of the Imperialist Competitive Algorithm (ICA)**

### 2.2. Harmony Search Algorithm (HS):

Harmony search is inspired by the improvisation process of jazz musicians. In the HS algorithm, each musician (decision variable) plays (generates) a note (a value) for finding a best harmony (global optimum) all together[6-7].
Based on [6-7],The iterative approach of HS can be described by the following steps:

**Step 1:** Initialize the HS memory (HM). The initial HM consists of a certain number of randomly generated solutions to the optimization problems under consideration. For an *n*-dimension problem, an HM with the size of *N* can be represented as follows:

$$HM = \begin{bmatrix} x_1^1, x_2^1, \ldots & x_n^1 \\ x_1^2, x_2^2, \ldots & x_n^2 \\ . & \\ . & \\ . & \\ x_1^{HMS}\, x_2^{HMS}, \ldots & x_n^{HMS} \end{bmatrix} \quad (1)$$

where $[x_1^{\,i}, x_2^{\,i}, \ldots, x_n^{\,i}]$ ($i = 1,2,\ldots,$ HMS) is a solution candidate. HMS is typically set to be between 50 and 100.

**Step 2 :**Improvise a new solution $[x_1', x_2', \ldots, x_n']$ from the HM. Each component of this solution, $x_j'$, is obtained based on the Harmony Memory Considering Rate (HMCR). The HMCR is defined as the probability of selecting a component from the HM members, and 1-HMCR is, therefore, the probability of generating it randomly. If $x_j'$ comes from the HM, it is chosen from the jth dimension of a random HM member and is further mutated according to the Pitching Adjust Rate (PAR). The PAR determines the probability of a candidate from the HM to be mutated. As we can see, the improvisation of $[x_1', x_2', \ldots, x_n']$ is rather similar to the production of offspring in the Genetic Algorithms (GAs)with the mutation and crossover operations. However, the GA creates new chromosomes using only one (mutation) or two (simple crossover) existing ones, while the generation of new solutions in the HS method makes full use of all the HM members.

**Step 3 :**Update the HM. The new solution from Step 2 is evaluated. If it yields a better fitness than that of the worst member in the HM, it will replace that one. Otherwise, it is eliminated.

Step 4 .Repeat Step 2 to Step 3 until a preset termination criterion, for example, the maximal number of iterations, is met. The flowchart of the HS is shown in Figure 2.

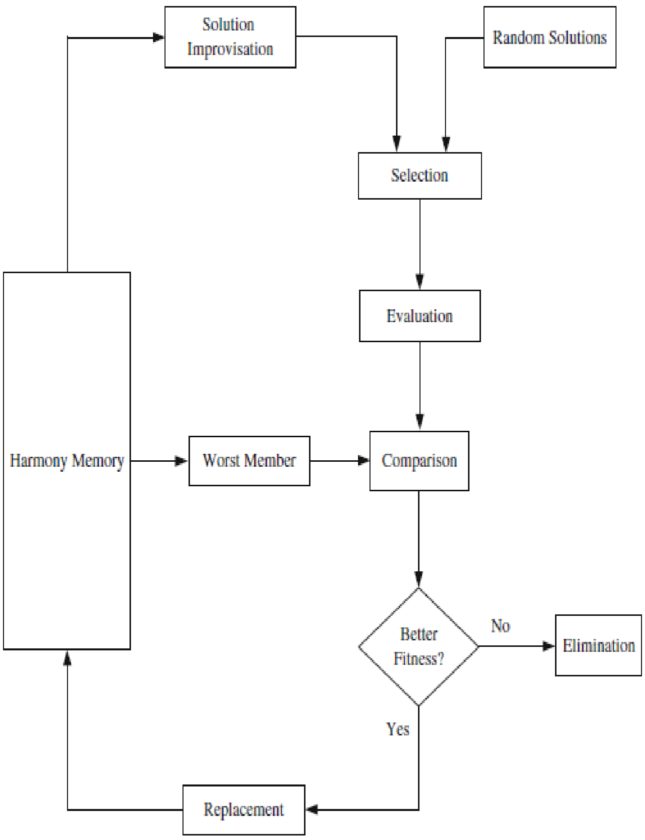



**Figure 2- Flowchart of the Harmony Search Algorithm (HS)**

### 2.3. Artificial Immune System (AIS):

Artificial immune system algorithm is derived from the mechanism where biological immune system fights against foreign pathogens. By using the adaptive immune response, this algorithm can be used to search for the best solution of an optimization problem. It can also be used to cluster or classify data for a problem [8]. Based on [8],The iterative approach of Artificial Immune System (AIS) can be described by the following steps:

**Step1:** Generate initial antibodies. These antibodies can be seen as the initial solution for the evolutionary algorithm.
**Step2:** Identify antigens. This step is used to simulate the situation when a body is invaded by foreign pathogens. In AIS algorithm, we just read in unidentified data to complete this step.
**Step3:** Calculate affinity. When the antibodies are trying to identify the antigens, the binding between the antibody and antigen has to be determined.
**Step4:** Adjust antibodies. Select good antibodies for adjustment.
**Step5:** Generate new antibodies. Copy the adjusted antibodies to create new antibodies and add them into the antibodies' pool.
**Step6:** Check termination condition. If the stop condition is met, go to Step7. Otherwise, go back to Step2.
**Step7:** Finish the algorithm.

The flowchart of the AIS is shown in Figure 3.

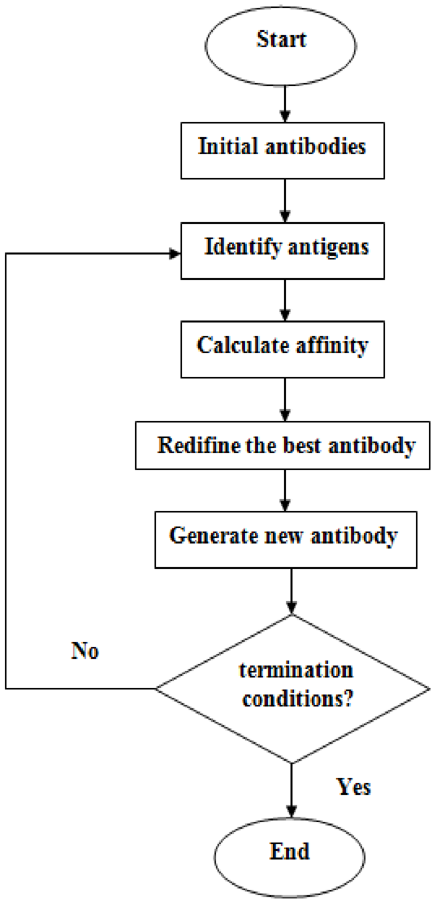



**Figure 3- Flowchart of the Artificial Immune System (AIS)**

### 2.4. Differential Evolution (DE):

Differential evolution(DE) is a population-based stochastic optimization algorithm for real-valued optimization problems. In DE each design variable is represented in the chromosome by a real number. The DE algorithm is simple and requires
only three control parameters: weight factor(F), crossover rates(CR), and population size(NP). The initial population is randomly generated by uniformly distributed random numbers using the upper and lower limitation of each design variable.
Then the objective function values of all the individuals of population are calculated to find out the best individual xbest,G of current generation, where G is the index of generation. DE operates through similar computational steps as employed by a standard evolutionary algorithm (EA) [9-10].
Based on[9-10],The iterative approach of Differential Evolution (DE) can be described by the following steps:

**Step1:** Initialization population and  to make the "Generated initial solution vector randomly": Initialization. Include the total number of solution vector
**Step2:**Evaluate the fitness value of  the solution vector.( of all Population)
**Step3:** Differential Mutation: a parent vector from the current generation is called target vector, a mutant vector obtained through the differential mutation operation is known as donor vector and finally an offspring formed by recombining the donor with the target vector is called trial vector.
**Step4:** Crossover operation or Recombination: Recombination incorporates successful solutions from the previous generation
**Step5:** Selection operation: Compare target vector and trial vector to determine the one can be reserved.
**Step6:** Check termination condition. If the stop condition is met, go to Step7. Otherwise, go back to Step2.
**Step7:** Finish the algorithm.

The flowchart of the DE is shown in Figure 4.

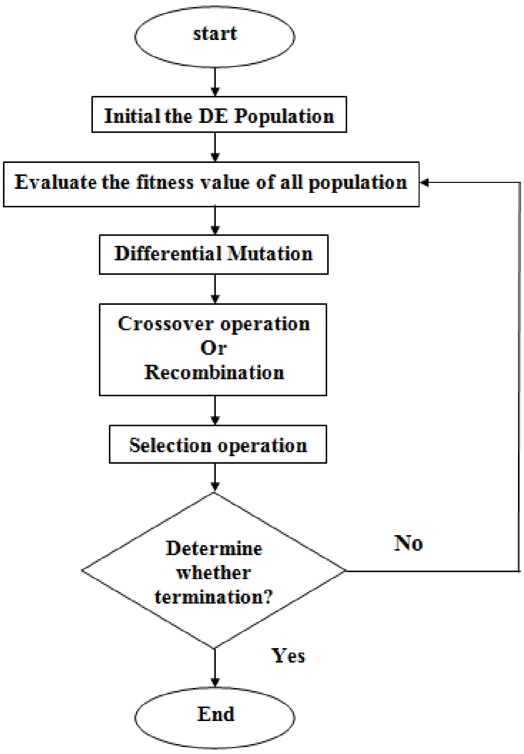



**Figure 4- Flowchart of the Differential Evolution (DE)**

### 2.5. Shuffled Frog Leaping Algorithm (SFLA):

SFLA is a novel meta-heuristic algorithm for combinatorial optimization problems. The basic idea of SFLA is inspired by the frog foraging behavior. Like other meta-heuristic algorithms, SFLA is also a population-based iterative algorithm. The goal of the frogs in population is to get the maksimum food by using minimum step. All the frogs in social group are the population of the algorithm and each frog represents a solution for the problem. Each frog in population has a fitness value which represents closeness degree to the food.

A lot of frogs live in the wetlands where many stones are discretely placed. The frogs try to find a place with more food by jumping to different stones. Every frog has its own culture, and the frogs in the same population can exchange their food information through communication. The frogs in the wetlands are divided into several sub-populations according to a certain strategy and each sub-population has its own culture. The sub-population conducts a local search. When the sub-populations' local search is satisfied, the information exchange between different sub-populations will begin to complete the global search. The local search and the global search will be conducted alternately until a frog finds the food or the alternating times reach the maximum[11-12].

Based on [11-12],The iterative approach of SFLA can be described by the following steps:

**Step1:** Initialize parameters:
-Population size
-number of memeplexes
-number of iterations within each memeplex
**Step2:** Generation of initial population(P) randomly and evaluating the fitness of each frog
**Step3:** Sorting population in descending order in term of fitness value
**Step4:** Distribution of frogs into (m) memeplexes
**Step5:** Local search Iterative updating the worst frog of each memeplex
**Step6:** Shuffle the memeplexes
**Step7:** Check termination condition. If the stop condition is met, go to Step8. Otherwise, go back to Step3.
**Step8:** Determine the best solution. The flowchart of the SFLA is shown in Figure 5.

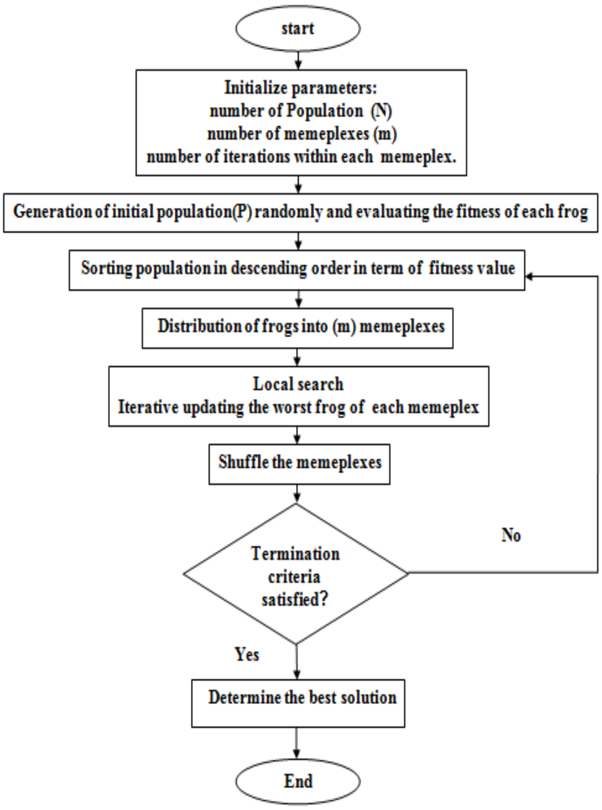



**Figure 5-Flowchart of the Shuffled Frog Leaping Algorithm (SFLA)**

**2.6. Invasive Weed Optimization(IWO):**

A weed is any plant growing where it is not wanted. Weeds have shown very robust and adaptive nature which turns them to undesirable plants in agriculture.Weeds invade a cropping system (field) by means of dispersal and occupy opportunity spaces between the crops. Each invading weed takes the unused resources in the field and grows to a flowering weed and produces new weeds, independently. The number of new weeds produced by each flowering weed depends on the fitness of that flowering weed in the colony. Those weeds that have better adoption to the environment and take more unused resources growfaster and produce more seeds. The new produced weeds are randomly spread over the field and grow to flowering weeds. This process continues till the maximum number of weeds is reached on the field due to the limited resources. Now, only those weeds with better fitness can survive and produce new weeds. This competitive contest between the weeds causes them to become well adapted and improved over the time[13-14].
The invasive weed optimization is a bio- inspired algorithm that simulates the natural behavior of seeds in colonizing and finding suitable places for growth and reproduction[13-14].
Each individual or agent, a set containing a value of each optimization variable, is called a seed.
Before considering the algorithm process, the new key terms used to describe this algorithm should be introduced. Table 1 shows some of these terms.
Each seed grows to a flowering plant in the colony. The meaning of a plant is one individual or agent after evaluating its fitness [13-14].

**Table 1- Some of the key terms used in the IWO**

| | |
|---|---|
| Agent/seed | Each individual in the colony containing a value of each optimization variable |
| Fitness | A value representing the goodness of the solution for each seed |
| Plant | one agent/seed after evaluating its fitness |
| Colony | The entire agents or seeds |
| Population Size | The number of plants in the colony |
| Maximum number of plants | The maximum number of plants allowed to produce new seeds in the colony |

Based on [13-14],The iterative approach of IWO can be described by the following steps:

**Step1:** Initialisation:
A population of initial seeds (N0) is randomly being dispread over the search space.
**Step2:** Evaluation:
The fitness of each weed in the population is calculated.
**Step3:** Reproduction:
Each weed in the population is allowed to produce seeds depending on its comparative fitness in the population. In other words, a weed will produce seeds based on its fitness, the worst fitness and the best fitness in the population. In such way, the increase of number of seeds produced is linear. The number of seeds for each weed varies linearly between $S_{min}$ for the worst plant and $S_{max}$ for the best plant. Figure 6 illustrates the procedure of reproduction.

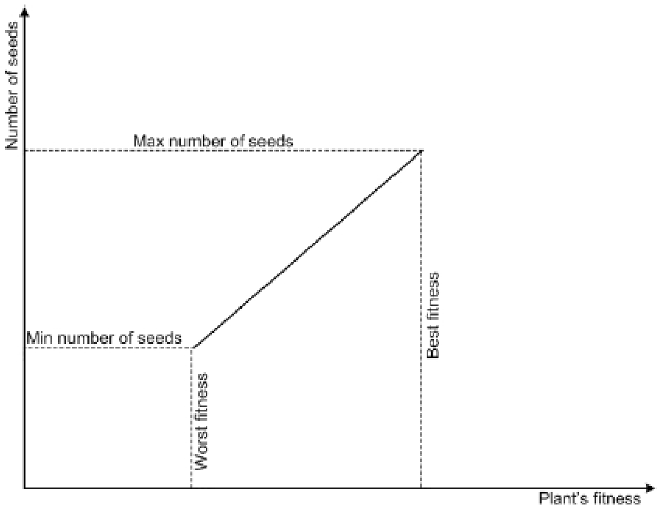



**Figure 6- Procedure of reproduction**

The equation for determining Weed $_{num}$ the number of seeds produced by each weed is presented in equation (1):

$$weed_{num} = S_{\min} + (S_{\max} - S_{\min}) \frac{f - f_{worst}}{f_{best} - f_{worst}} \quad (2)$$

Where f is the fitness of the weed considered, $f_{worst}$ and $f_{best}$ are respectively the worst and the best fitness in the population.
for a population containing five weeds, If we assume that weed Number 5 and weed Number 1 are the worst and best weeds between Respectively, So, the number of seeds around weed Number 5 is equal to $S_{min}$ and the number of seeds around weed Number 1 is equal to $S_{max}$.
**Step4:** Spatial dispersion:
This step ensures that the produced seeds will be generated around the parent weed, leading to a local search around each plant. The generated seeds are randomly spread out around the parent weeds according to a normal distribution with mean equal to zero and variance $s^2$. The standard deviation of the seed dispersion s decreases as a function of the number of iterations iter. The equation for determining the standard deviation for each generation is presented in equation (2):

$$\sigma_{iter} = \frac{(iter_{\max} - iter)^n}{(iter_{\max})^n} (\sigma_{initial} - \sigma_{final}) + \sigma_{final} \quad (3)$$

Where $iter_{max}$ is the maximum number of iterations.
$\sigma_{iter}$ is the standard deviation at the current iteration and " n" is the nonlinear modulation index. Obviously, the value of " σ " defines the exploration ability of the weeds. Therefore, as iter increases, the exploration ability of all weeds is gradually reduced. At the end of the optimization process, the exploration ability has diminished so much that every weed can only fine its position [9].
**Step5:**Competitive Exclusion:
After a number of iterations, the population reaches its maximum, and an elimination mechanism is adopted: The seeds and their parents are ranked together and only those with better fitness can survive and become reproductive. Others are being eliminated.
**Step6:**Termination Condition:
The whole process continues until the maximum number of iterations has been reached, and we hope that the plant with the best fitness is the closest one to the optimal solution.
The flowchart of the IWO is shown in Figure 7.

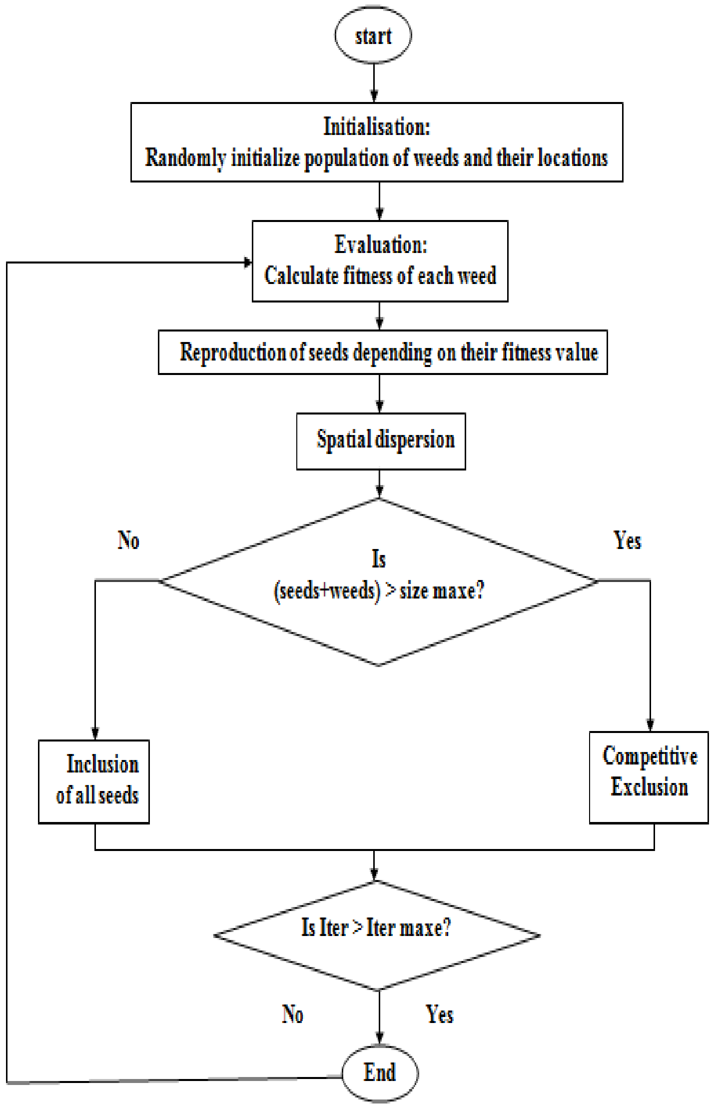



**Figure 7- Flowchart of the Invasive Weed Optimization (IWO)**

### 2.7. Biogeography-Based Optimization (BBO):

BBO algorithm is derived from biogeography, whose main contents is to establish mathematical models for a series of events include residence, migration routes, production of new species, and extinction of species in nature.The mathematical model of biogeography describe how species migrate from one island to another, how new species arise and how species become extinct. Geographical areas that are well suited as residences for biological species are said to have a high habitat suitability index (HSI). The variables that characterize habitability are called suitability index variables (SIV). SIVs are the independent variables of the habitat and HSI can be considered as the dependent variable. Habitats with a high HSI tend to have a large number of species and those with a low HSI have a small number of species. Habitats with a high HSI have a low species immigration rate because they are already nearly saturated with species. high HSI habitats are more static in their species distribution than low HSI habitats.so the large number of species on high HSI islands will tend to emigrate to neighbouring habitats. The algorithm of BBO consists of two main stages, migration and mutation[15-16]. Based on [15-16],The iterative approach of BBO can be described by the following steps:

**Step 1:** Initialize the parameters of BBO algorithm and the $H_i$ vector of any habitat.
**Step 2:** For different suitability $H_i$ , sort the habitats from good to bad. Generally the update rate of habitats i = 1.

**Step 3:** By comparison, judge whether the desired optimum is satisfied or not. If it is satisfied, the optimum is output and algorithm procedure is terminated. Otherwise, turn to Step 4.
**Step 4:** Suppose the maximum number of a specie in a habitat $S_{max} = n$. So by means of $S_i = S_{max} - i$ ($i = 1,2,...,n$), the populations value $S_i$ of habitat i is obtained, which is further brought into the migration model to obtain its $\lambda_i$ and $\mu_i$ .
**Step 5:** After the cyclic operation of $P_{mod}$ , whether i has entered into the immigration pattern (the number n of i is defined as the number of cycles) can be determined. If habitat i is carried out the immigration operation, the habitat immigration rate $\lambda_i$ (the dimension D of SIV as the number of cycles) is used to judge its characterized component $SIV_{ij}$ whether to be immigrated or not. If $SIV_{ij}$ is implemented with the immigration, then, through its emigration rate $\mu_m$ ( $m=1,2,...,n$, $m \neq i$ ) it can be performed by selecting, and then the feature component $SIV_{ij}$ of the i is replaced by a component of selected m .
**Step 6:** By calculating $M_i$ of the corresponding habitat, the related variable of the Habitat i is judged to see whether the mutation has occurred. The results are compared and turn to Step 2.
The flowchart of BBO algorithm is shown in Figure 8.

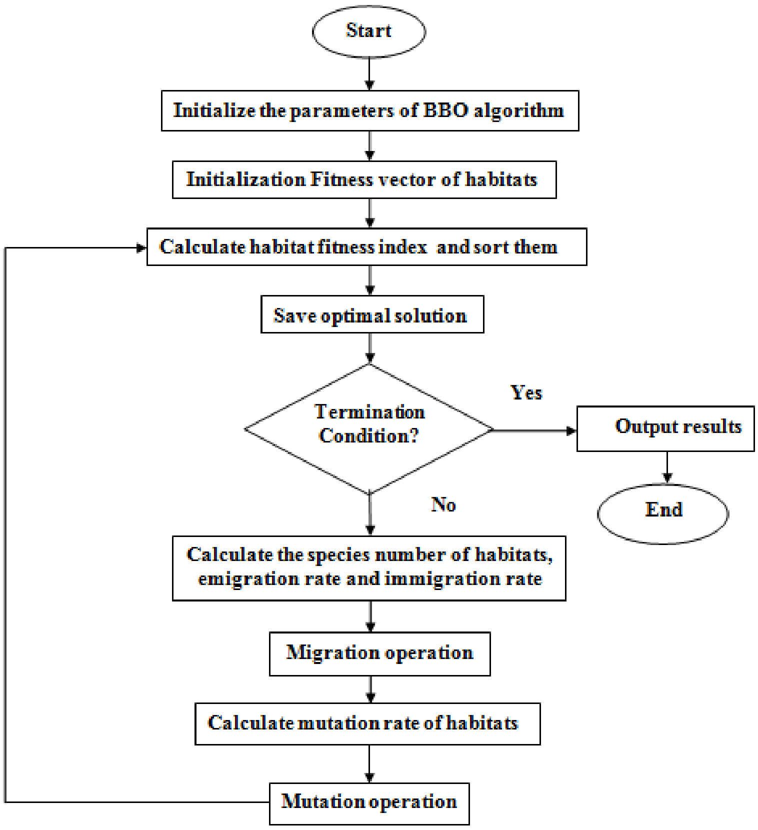



**Figure 8- Flowchart of the Biogeography-Based Optimization (BBO)**

### 2.8. Teaching-Learning-Based Optimization (TLBO):

This method works on the effect of influence of a teacher on learners. Like other nature-inspired algorithms, TLBO is also a population-based method and uses a population of solutions to proceed to the global solution. The population is considered as a group of learners or a class of learners. The process of TLBO is divided into two parts: the first part consists of the 'Teacher Phase' and the second part consists of the 'Learner Phase'. 'Teacher Phase' means learning from the teacher and 'Learner Phase' means learning by the interaction between learners[17]. Based on [17],The iterative approach of TLBO can be described by the following steps:

**Step 1 :** Initialize the population size or number of students in the class ( N ), number of generations (G ), number of design variables or subjects (courses) offered which coincides with the number of units to place in the distribution system (D ) and limits of design variables (upper, $U_L$ and lower, $L_L$ of each case).
Define the optimization problem as: Minimize f(X) , where f (X) is the objective function, X is a vector for design variables such that $L_L \leq X \leq U_L$.
**Step 2 :** Generate a random population according to the number of students in the class (N ) and number of subjects offered (D). This population is mathematically expressed as

$$\begin{bmatrix} X_{1,1} & X_{1,2} & . & . & X_{1,D} \\ X_{2,1} & X_{2,2} & . & . & X_{2,D} \\ . & . & . & . & . \\ . & . & . & . & . \\ X_{N,1} & X_{N,2} & . & . & X_{N,D} \end{bmatrix} \tag{4}$$

Where $X_{i,j}$ is the initial grade of the $j^{th}$ subject of the $i^{th}$ student.
**Step 3:** Evaluate the average grade of each subject offered in the class. The average grade of the j th subject at generation g is given by:

$$M^{g} = mean(X_{1,j}, X_{2,j}, ...X_{i,j}) \tag{5}$$

**Step 4:** Based on the grade point (objective value) sort the students (population) from best to worst. The best solution is considered as teacher and is given by:

$$X_{teacher} = X \,/_{f(x)=\min} \tag{6}$$

**Step 5:** Modify the grade point of each subject (control variables) of each of the individual student. Modified grade point of the $j^{th}$ subject of the $i^{th}$ student is given by:

$$\begin{aligned} X^{g}_{new(i)} &= X^{g}_{(i)} + rand \times (X^{g}_{Teacher} - T_F M^{g}) \\ X^{g}_{new(i)} &= X^{g}_{(i)} + r_1 \times [X^{g}_{Teacher} - (round(1+r_2) \times M^{g})] \end{aligned} \tag{7}$$

Where $r_1$ , $r_2$ are random numbers between [0, 1].
**Step 6 :** Every learner improves grade point of each subject through the mutual interaction with the other learners. Each learner interacts randomly with other learners and hence facilitates knowledge sharing. For a given learner, $X^{g}_{(i)}$ another learner $X^{g}_{(r)}$ is randomly selected (i≠r). The grade point of the $j^{th}$ subject of the $i^{th}$ learner is modified by:

$$X^{g}_{new(i)} = \begin{cases} X^{g}_{(i)} + rand \times (X^{g}_{(i)} - X^{g}_{(r)}) & if\ (X^{g}_{(i)} \langle X^{g}_{(r)}) \\ X^{g}_{(i)} + rand \times (X^{g}_{(r)} - X^{g}_{(i)}) & otherwise \end{cases} \tag{8}$$

The flowchart of TLBO algorithm is shown in Figure 9.

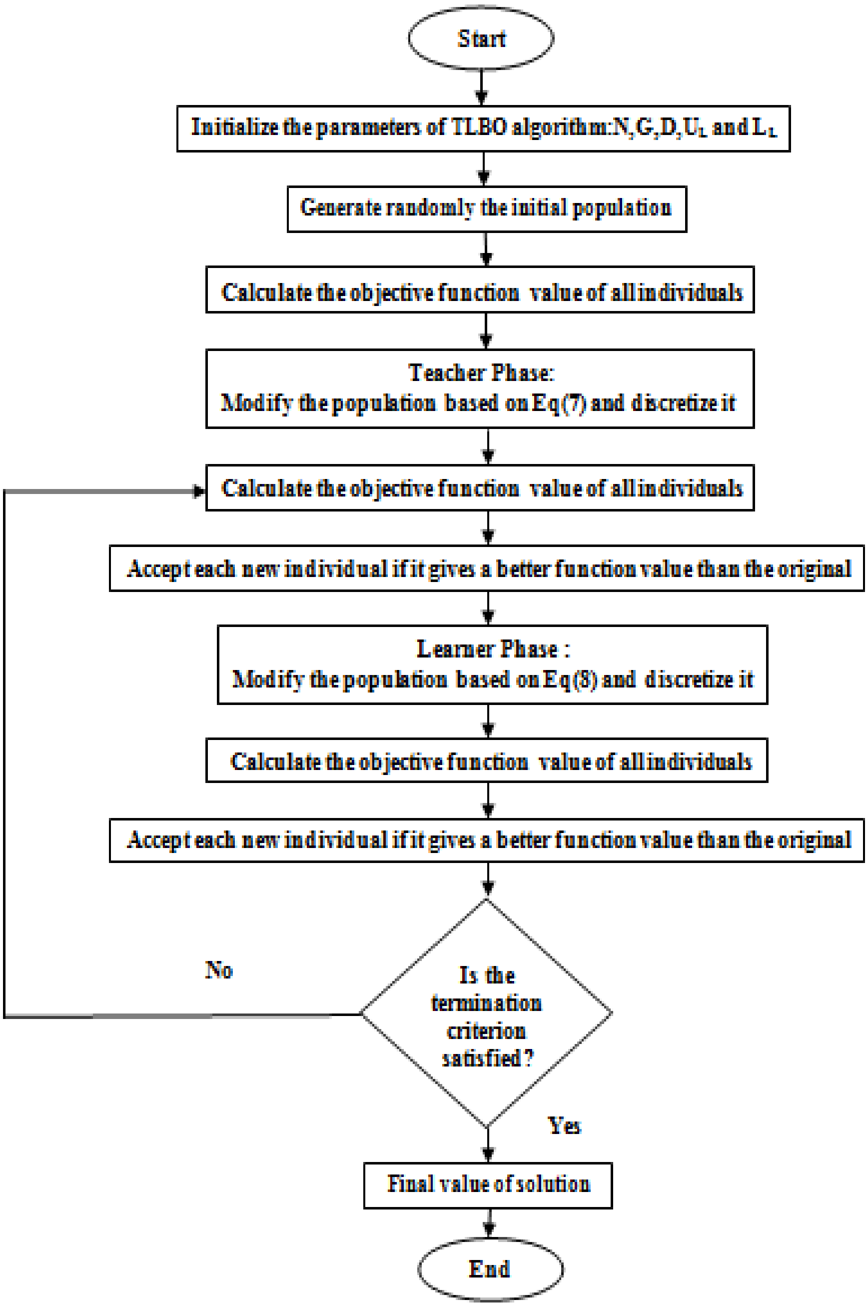



**Fig. 9- Flowchart of the TLBO algorithm**

#### 2.9. bat algorithm (BA):

Based on the echolocation behavior of microbats the Bat Algorithm has built on which in complete darkness they can find their prey.Based on the echolocation they also avoid an obstacle, which is an important characteristic when flying in the dark, the frequency return for which is different. There are some assumptions that the algorithm takes, those are[18]:
1)All bats use echolocation to sense distance, and they are also ‘aware’ of the difference between food/prey and background barriers in some magical way;
2)Bats fly randomly with a fixed frequency of varying wavelength and loudness to search for prey. They can automatically adjust the wavelength of their emitted pulses and adjust the rate of pulse emission depending on the proximity of their target.
3)Although the loudness can vary in many ways, but can assume that the loudness varies from a large (positive) to a minimum constant value.
Based on [18].The iterative approach of BA can be described by the following steps:

**Step 1:** initialization.
In this step, we initialize the parameters of algorithm, generate and also evaluate the initial population, and then determine the best solution $x_{best}$ in the population.
**Step 2:** generate the new solution.
Here, virtual bats are moved in the search space according to updating rules of the bat algorithm.
**Step 3:** local search .
The best solution is being improved using random walks.
**Step 4:**
evaluate the new solution.
The evaluation of the new solution is carried out.
**Step 5:**
save the best solution conditionaly .
conditional archiving of the best solution takes place.
**Step 6:**
find the best solution .
the current best solution is updated.

The flowchart of BA algorithm is shown in Figure 10.

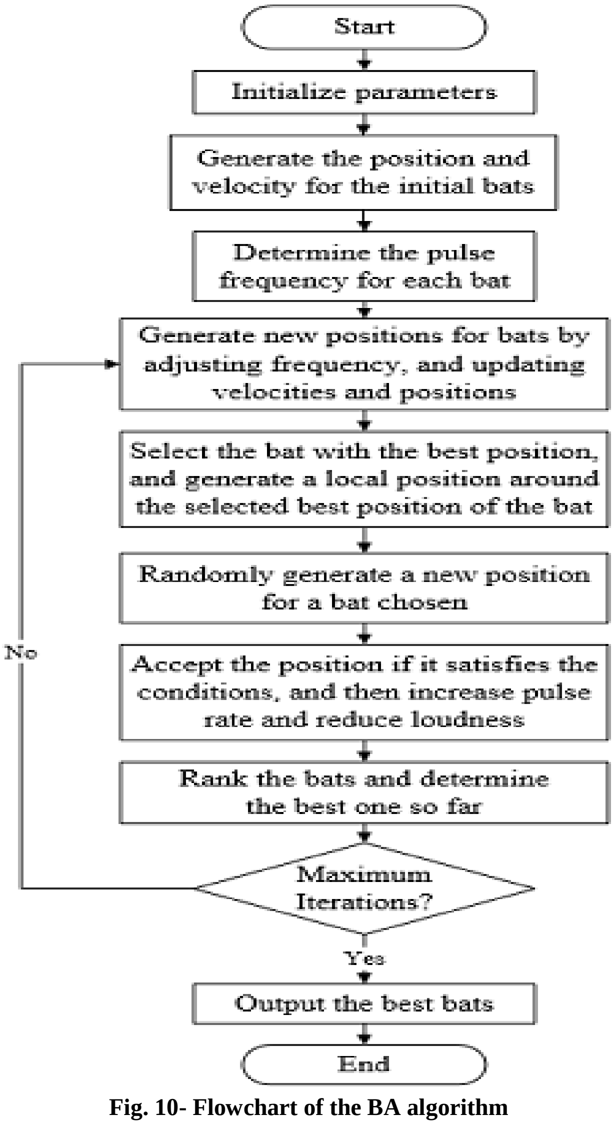



**Fig. 10- Flowchart of the BA algorithm**

### 2.10. Artificial Fish Swarm algorithm (AFSA):

artificial fish-swarm algorithm is one of the best methods of optimization among the swarm intelligence algorithms. This algorithm is inspired by the collective movement of the fish and their various social behaviors. Based on a series of instinctive behaviors, the fish always try to maintain their colonies and accordingly demonstrate intelligent behaviors. Searching for food, immigration and dealing with dangers all happen in a social form and interactions between all fish in a group will result in an intelligent social behavior.This algorithm has many advantages including high convergence speed, flexibility, fault tolerance and high accuracy[19-20]. More information In this caseIs available in [19-20].
A simple flowchart of AFSA algorithm is shown in Figure 11.

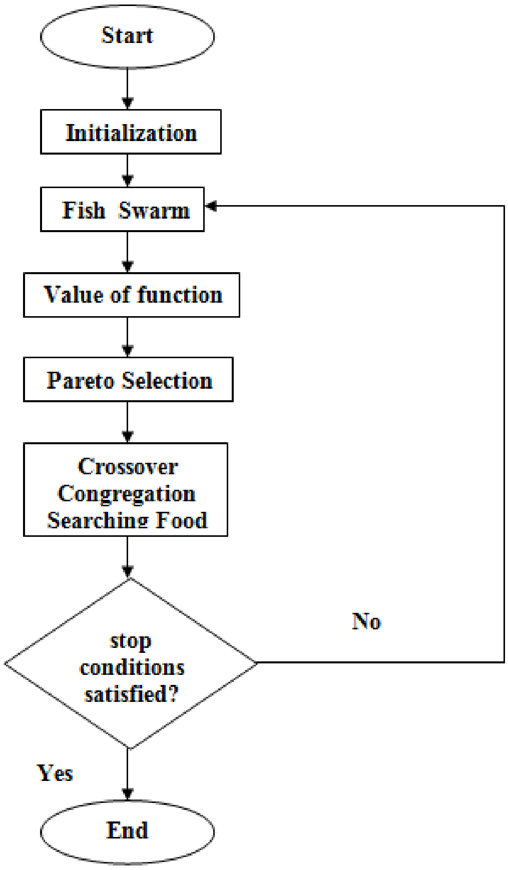



**Fig.11-Flowchart of the AFSA algorithm**

## 3. Conclusion

optimization is the process of adjusting the inputs to or characteristics of a device, mathematical process, or experiment to find the minimum or maximum output or result.In the simplest case, an optimization problem consists of maximizing or minimizing a real function by systematically choosing input values from within an allowed set and computing the value of the function.
Optimization may be single-objective or multi-objective. And this depends on the optimization problem.
In this paper, We reviewed the latest optimization methods, Which are used in electrical engineering articles and related sciences,Then For each optimization method, its algorithm is also plotted.The optimization methods introduced in this paper extend those engineering economics methods. Some are discrete, some are continuous.We also have two types of optimization problem:
Unconstrained Optimization and constrained Optimization.

The optimization methods studied are: Imperialist Competitive Algorithm (ICA), Harmony Search Algorithm(HS), Artificial Immune System (AIS), Differential Evolution (DE), Shuffled Frog Leaping Algorithm (SFLA), Invasive Weed Optimization(IWO), Biogeography-Based Optimization (BBO), Teaching-Learning-Based Optimization (TLBO), bat algorithm (BA), Artificial Fish Swarm algorithm (AFSA). Some of these methods have faster convergence and some other higher precision. The use of any method depends on the nature of the optimization problem and the tastes of individuals.

## 7. References


1. Boyd, Stephen P.; Vandenberghe, Lieven (2004). Convex Optimization . Cambridge University Press. pp. 129
2. Fogel,D.B.(1997 )."The Advantages of Evolutionary Computation"Biocomputing and emergent computation: Proceedings of BCEC97 .1-11.
3. Najafi.B," The Study of the most Important Intelligent Optimization Methods in Electrical Engineering", 4th National Conference on Information Technology,Computer & TeleCommunication, University of Torbat Heydarieh,July 2017.
4. Kunlei, Lian, Chaoyong, Zhang, Liang, Gaoa, “Single row facility layout problem using an imperialist competitive algorithm”, Proceedings of the 41st International Conference on Computers & Industrial Engineering
5. Zanganeh.m & Mirabedini.s.j, (2015)” Comparing Imperialist Competitive Algorithm with Backpropagation Algorithms for Training Feedforward Neural Network” Journal of mathematics and computer science,pp. 193-204
6. Geem, Zong Woo, Joong Hoon Kim, and G. V. Loganathan (2001). "A new heuristic optimization algorithm: harmony search." Simulation 76.2: pp. 60-68.
7. Wang. X, Gao. X & Zenger. K, (2015)” An Introduction to Harmony Search Optimization Method” SpringerBriefs in Computational Intelligence
8. Chang.A, Fu Liao.J, Chang.P.G, Teng.C.H & Chen.M.H(2014)” Application of Artificial Immune Systems combines Collaborative Filtering in Movie Recommendation System” IEEE 18th International Conference on Computer Supported Cooperative Work in Design (CSCWD),pp. 277 – 282.
9. Yin Wu.C & Ying Tseng.K,(2010),” Topology Optimization of Structure Using Differential Evolution” Structural and Multidisciplinary Optimization,PP. 939–953.
10. Wei Chien.C, Rong Hsu.Z & Ping Lee.W,(2009)” Improving the Performance of Differential Evolution Algorithm with Modified Mutation Factor” International Conference on Machine Learning and Computing.
11. Jadidoleslam.M, Bijami.E, Amiri.N, Ebrahimi.A & Askari.J,(2012),” Application of Shuffled Frog Leaping Algorithm to Long Term Generation Expansion Planning” International Journal of Computer and Electrical Engineering, Vol.4, No.2.
12. Karakoyun.M & Babalik.A,(2015),”Data Clustering with Shuffled Leaping Frog Algorithm (SFLA) for Classification” Int'l Conference on Intelligent Computing, Electronics Systems and Information Technology (ICESIT-15),
13. Mekni.S & Fayech.B.C,(2014),”A MODIFIED INVASIVE WEED OPTIMIZATION ALGORITHM FOR MULTIOBJECTIVE FLEXIBLE JOB SHOP SCHEDULING PROBLEMS” Third International Conference on Advanced Information Technologies & Applications,
14. Karimkashi.S & Kishk.A,(2010),” Invasive Weed Optimization and its Features in Electromagnetics” IEEE Transactions on Antennas and Propagation,PP. 1269 - 1278
15. Wang.J.S & Song.J.D,(2017),” Application and Performance Comparison of Biogeography-based Optimization Algorithm on Unconstrained Function Optimization Problem” International Journal of Applied Mathematics,pp. 82-88.
16. Malik.S, Dutta.P, Chakrabarti.S & Barman.A, (2017),” Survey on Biogeography Based Optimization Algorithm and Application of Biogeography Based Optimization to determine Parameters of PID Controller” International Journal of Advanced Research in Computer and Communication Engineering, Vol. 3, Issue 2.
17. Venkata Rao.R, (2016)” Teaching Learning Based Optimization Algorithm And Its Engineering Applications” Springer International Publishing

18. Anand.R, Rahman.A, (2014)" Solution of Commitment problem Using BAT Algorithm" International Journal of Engineering & Technology Innovations, Vol. 1 Issue 2
19. Xueqin.T, Jingshun.D, Liya.J & Zongchang.X, (2011)" WNN Optimization Design Based on Artificial Fish-Swarm Algorithm" IEEE International Conference on Computer Science and Network Technology,pp. 2747 – 2750
20. Shi.L, Guo.R & Ma.Y, (2011)," A novel artificial fish swarm algorithm for pattern recognition with convex optimization" IEEE International Conference on Communication and Electronics Systems (ICCES),pp. 1 – 4.
21. Parkinson, A.R., Balling, R., & J.D. Hedengren,( 2013.) "Optimization Methods for Engineering Design", Brigham Young University.
22. Belegundu A. and T. Chandrupatla,(1999),"Optimization Concepts & Applications in Engineering", Prentice Hall,.
23. Gen, M. & R. Cheng,(2000)," Genetic Algorithms and Engineering Optimization", Wiley,
24. Fletcher R.,(1980-1981)," Practical Methods of Optimization Volumes 1,2", John Wiley .
25. Luenberger & Ye,(2008)," Linear and Nonlinear Programming "Third Edition, Springer.
26. Rothlauf F. (2011)," Optimization Methods. In: Design of Modern Heuristics". Natural Computing Series. Springer, Berlin, Heidelberg.
27. Marler R.T & Arora J.S,(2004)" Survey of Multi-Objective Optimization Methods for Engineering", Structural and Multidisciplinary Optimization 26, 369–395.