International Journal of Enhanced Research in Science, Technology & Engineering
ISSN: 2319-7463, Vol. 7 Issue 5, May-2018, Impact Factor: 4.059

Using Polyak's Log-Sigmoid Method with Firefly Swarm

Rana Z. Al-Kawaz[1], Huda I. Ahmed[2], Hamsa T. Saaed[3]

[1]Department of Mathematics, College of Basic Education, University of Telafer, Iraq
[2]Department of Operation Research and Intelligent Techniques, College of Computer Sciences and Mathematics, University of Mosul, Iraq
[3] Department of Mathematics, College of Computer Sciences and Mathematics, University of Mosul, Iraq

ABSTRACT

Recently modern optimization methods are often met heuristics and they are very promising in solving NP-difficult optimization problems. The firefly algorithm (FA) from the met heuristics algorithms have become an increasingly important tool of Swarm Intelligence that has been applied in almost all areas of optimization, in addition to engineering practice engineering practice. Many problems from different fields have been successfully solved using the firefly algorithm and its variants. In this paper, we aim to extend the firefly algorithm (EFA) by mixed it with Polyak's log-sigmoid function and The second and third algorithms were using the hybridization of the penalty function with Polyak's log-sigmoid function (H1FA) and the latter with a augmented Lagrangian function (H2FA) The latest algorithm was given a new style by splitting Polyak's log-sigmoid function on penalty function All the algorithms proposed in the search to solve nonlinear and engineering problems. We have analyzed it similarities and difference with particle swarm optimization. These algorithms were implemented and compared to get a good numerical results.

HOW TO CITE THIS ARTICLE

Rana Z. Al-Kawaz, Huda I. Ahmed, Hamsa T. Saaed, “Using Polyak's Log-Sigmoid Method with Firefly Swarm”, International Journal of Enhanced Research in Science, Technology & Engineering, ISSN: 2319-7463, Vol. 7 Issue 5, May -2018.

1. INTRODUCTION

Most optimization problems in nonlinear engineering have many constrained. Therefore, an optimal solution to these problems would require effective optimization algorithms . In general, optimization algorithms can be divided into two main categories: determinism and stochastic. Class I: The determinism algorithms such as climbing the same set of solutions if the duplicates begin with the same initial guess. Class II: stochastic algorithms often produce different solutions up to the same point. However, the final results, although slightly different, will usually converge to the same optimal solutions within a certain accuracy. The determinism algorithms are almost all local search algorithms, and they are very effective in finding a local optima. However, there is a risk that algorithms may be confined to the local Optima, while Optima Global is out of reach. It is common practice to introduce some random elements into an algorithm until it becomes possible to jump out of this area. In this case, the algorithms become random. Random algorithms often include an inevitable component and a random component.

The latter can take as many forms as simple random distribution by random sampling of the search area or by random walking. Most random algorithms can be considered meteoritic. Metaheuristics algorithm, especially those based on the intelligence of the squadron (swarm intelligence), form an important part of contemporary global optimization algorithms[14,29,3,2,5,17,18], and good examples are genetic algorithms (GA) [9,7] and particle swarm optimization (PSO) [13,12]. Many modern metaheuristics algorithms were developed based on the swarm intelligence in nature [13,6]. New modern metaheuristics algorithms are being developed and show their power and effectiveness. For example, the Firefly Algorithm (FA) developed by the author shows its superiority over some traditional algorithms [25,15,4,26,30].

The paper is organized as follows: we will review the main idea of the Firefly Algorithm in Section 2, and we then describe a few advantage for the Polyak's log-sigmoid penalty method in Section 3. We proposed the new algorithms so that the first algorithm extended the Firefly algorithm (EFA) and the second and third was a hybridization of the Firefly algorithm (H1FA) and (H2FA)The third algorithm was the development of the same algorithm (MFA). All the new algorithms with the access feature to solve the problems of variable structural improvement mixed in Section 4. In Section 5, we will use our proposed algorithms to find the optimal solution of an some famous and engineering design problem. Finally, we will discuss the topics for further studies.

2 FOREWORD TO THE FIREFLY ALGORITHM

Firefly Algorithm

The firefly algorithm was developed by the author [27,28] and was based on the ideal behavior of flame retardant properties. In the firefly algorithm, the objective function of a particular improvement problem is related to this flashing light or light intensity that helps the firefly squad move to brighter and more attractive locations than to obtain the most effective solution [16].

To illustrate, we can achieve these brilliant characteristics such as the following three rules: s.t Because of the nature of the two sexes for all fireflies one firefly will be drawn to another despite his sex and because the gravity is directly proportional to the brightness of the lighters, so the less bright fireflies will be attracted by the narrower mattress. Finally, the brightness of the firefly is calculated by the the objective function.

If the distance between fireflies increases, both gravity and brightness will decrease significantly. Also if the firefly does not find anyone in its vicinity, it will go in a random direction[16]. In the FA, there are two important issues: the contrast of light intensity and attractiveness formulation.

Attractiveness:

In the firefly algorithm, the form of attractiveness function of a firefly is the following monotonically decreasing function [15]:

$$\beta(r) = \beta_0 e^{-\gamma r^m} \quad , (m \geq 1) \qquad ..(1)$$

Most often, it is used

$$\beta(r) = \beta_0 e^{-\gamma r^2} \qquad ..(2)$$

where, r is the distance between any two fireflies, β_0 is the initial attractiveness at r=0 and γ is an absorption coefficient which controls the decrease of the light intensity.

Distance:

The distance between any two fireflies i and j, at positions x_i and x_j, respectively, can be defined as a Cartesian distance [11]:

$$r_{ij} = \|x_i - x_j\|_2 = \sqrt{\sum_{k=1}^{d}(x_{i,k} - x_{j,k})^2} \qquad ..(3)$$

where $x_{i,k}$ is the kth component of the spatial coordinate xi of the ith firefly and d is the number of dimensions we have, for d = 2, we have

$$r_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \qquad ..(4)$$

However, the calculation of distance *r* can also be defined using other distance metrics, based on the nature of the problem.

Movement

The movement of a firefly *i* which is attracted by a more attractive i.e., brighter firefly *j* is given by the following equation :

$$x_{i+1} = x_i + \beta_0 e^{-\gamma r^2}(x_i - x_j) + \alpha\,(rand - \tfrac{1}{2}) \qquad ..(5)$$

Where the first chapter is the current state of a firefly, where $\beta_0 e^{-\gamma r^2}(x_i - x_j)$ is due to the attraction of the firefly x_j; so if β_0= 0 then it turned out to be a simple random movement. Firefly gravity is compared with the previous position.

The algorithm will update the position of the firefly to a higher gravity value; otherwise the firefly will remain in the current state. The algorithm's termination criterion depends on the number of duplicates given previously [29,10].

The coefficient α is a randomization parameter determined by the problem of interest, while randis a random number generator uniformly distributed in the space [0,1]. As we will see in this implementation of the algorithm, we will use β_0=1.0, $\alpha\in$[0,1] and the attractiveness or absorption coefficient γ=1.0, which guarantees a quick convergence of the algorithm to the optimal solution [1].

Firefly Algorithm Flow Chart:

Firefly Algorithm (FA) was developed by Xin-She Yang in 2008**.** The flow chart of Firefly Algorithm is shown in (Figure 1 and 2). Every new position should be evaluated by fitness function which is assumed as Integral Square Error (ISE) [8].

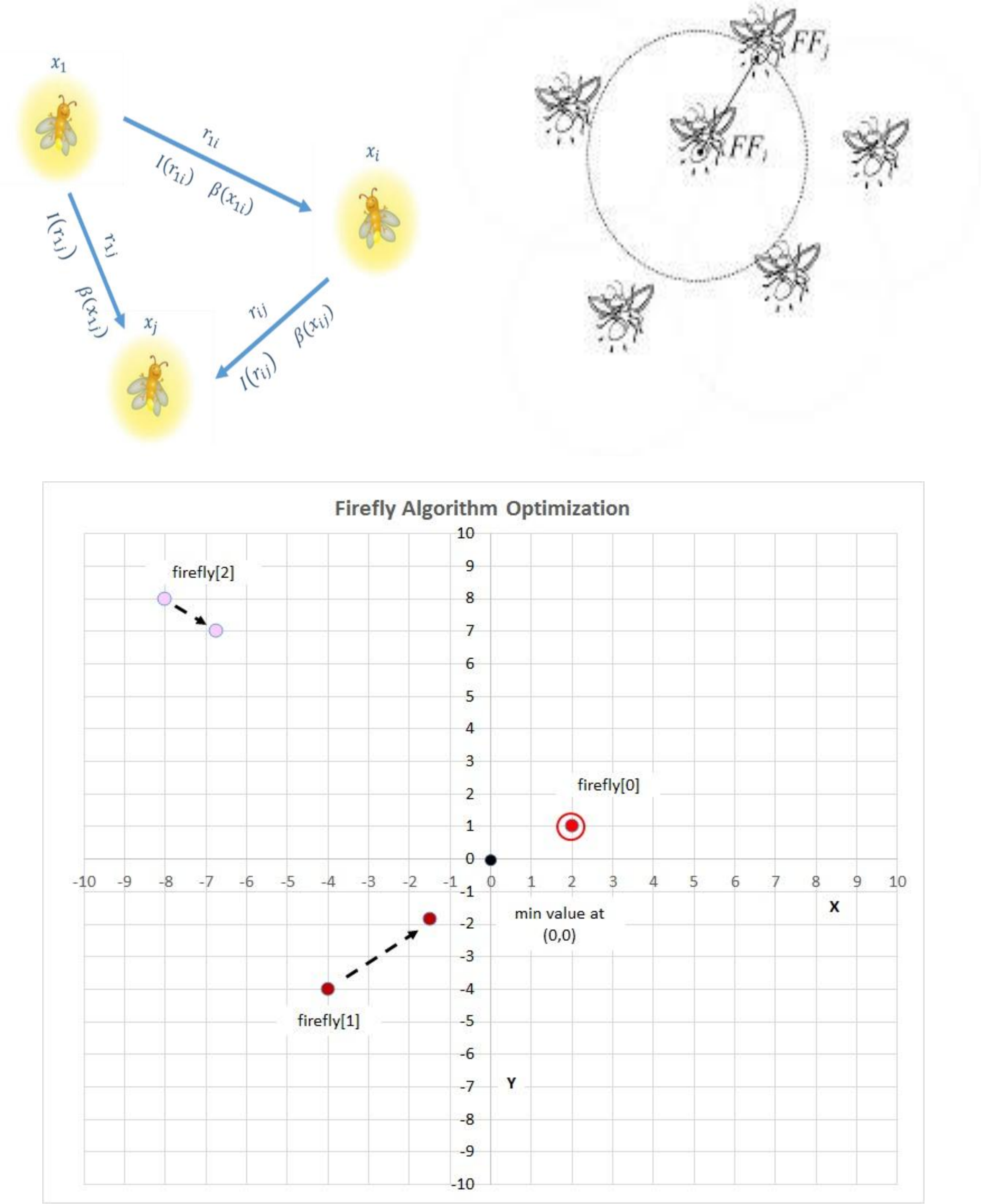

Fig. 1: Illustrate Firefly Algorithm [31].

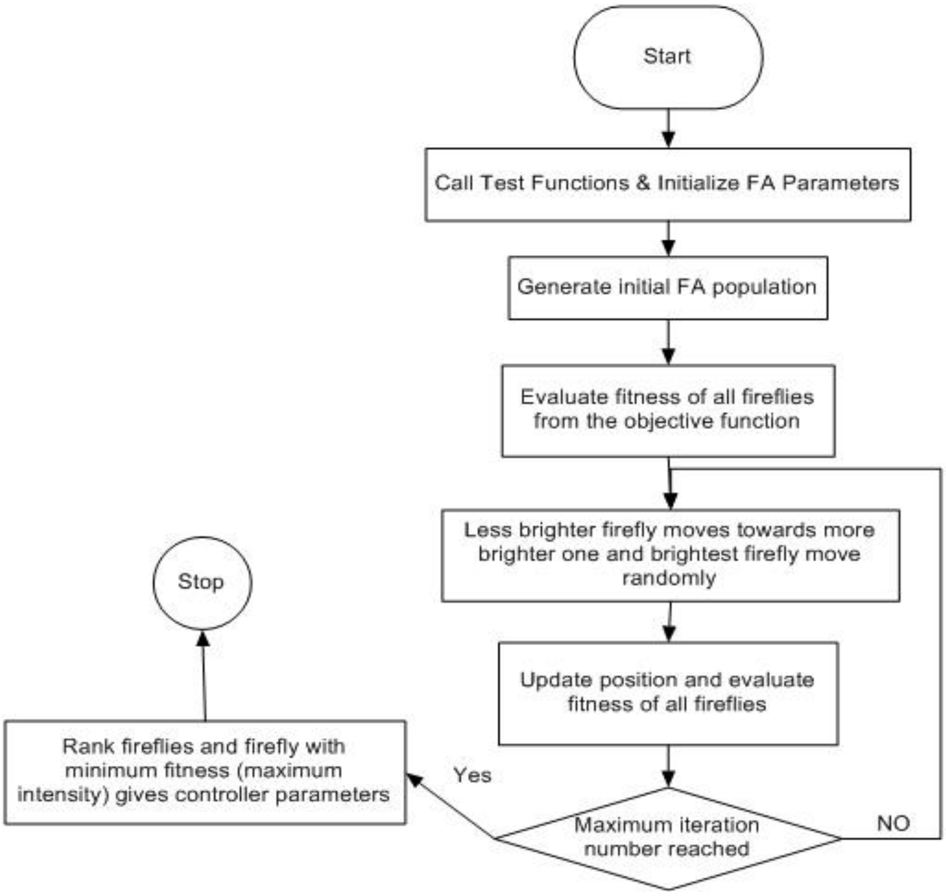

Fig. 2: Flow chart of Firefly Algorithm.

3. CONSTRAINED HANDLING APPROACH

The famous method in community optimization metaheuristic to deal with constraints is to use the penalty method. The main idea of this method is to convert a constrained optimization problem into an unconstrained problem by adding a specific value to the target function focus on the number of the constraint violation that occurred in a particular solution. This technique, known as the penalty method, is one of the famous methods of dealing with evolutionary algorithms. The current work has been used the same method:

Constrained Handling

The SUMT (Sequential Unconstrained Minimization Technique) was a very popular approach in the 1960's, where the original constrained problem was transformed to a sequence of unconstrained problems. This was a natural approach because unconstrained minimization techniques were evenly well developed. As research in optimization algorithms progressed, other methods were shown to be more efficient and reliable for "Typical" optimization problems. However, as problem has increase in size, it has been found that the more modem methods can become computationally inefficient. Thus, especially for structural optimization using approximation techniques, a new look at SUMT is appropriate[24]. The basic idea of this method is to transform a constrained optimization problem into an unconstrained one by adding a certain value to the objective function based on the number of constraint violation occurred in a certain solution. The SUMT approach is:

Minimize $\varphi(x) = f(x) + p(x)$ (6)

Where φ(x) is called the pseudo-objective function and the penalty term, $p(x)$, depends on the method being used. This general approach is described in some detail in [23]. The key distinction of each method is the form of $p(x)$. The most common approach in the metaheuristic optimization community to handle constraints is to use the penalty method. During this project, the following methods were considered:

Exterior Penalty Function [23]

The exterior penalty method, is one of the most popular methods of constraint handling in the evolutionary algorithms. If the optimization problem consists of minimization of cost function f subjected to the inequality constraints $c_i \le 0$, (i

=1 to p) and equality constraints h_i=0, (i =1 to q) , then in the penalty function approach, the constraints can be collapsed with the cost function into a penalty function defined as follows:

$$p(x) = \sum_{i=1}^{p} \lambda_i \ [c_i^+(x)]^2 + \sum_{i=1}^{q} k_i \ [h_i \ (x)]^2 \qquad (7)$$

$$c_i^+(x) = \max(c_i \ (x), 0)$$

The subscript p is the outer loop counter which we will call the cycle number. We begin with a small value of the penalty parameter, k_i, and minimize the pseudo-objective function,$\varphi(x)$. We then increase k_i and repeat the process until convergence.

Interior Penalty Function, Polyak's Log-Sigmoid [24]

In recent years the penalty function has been updated by Polyak, et.al.[20] appears to have better properties than the log-barrier method by eliminating the barrier. Here, they introduced the Log-Sigmoid penalty function (LSP) as:

$$p(x) = 2k_p^{-1} \sum_{i=1}^{m} \lambda_i^{p+1} \{\ln[1 + e^{k_p c_i(x)}] - \ln(2)\} \qquad (8)$$

Where,

$$\lambda_i^{p+1} = \frac{2\lambda_i^p}{[1 + e^{-k_p c_i(x)}]}$$

When compensating by equation (6), the efficiency algorithm increases the efficiency of the traditional algorithm of the penalty function.

Augmented Lagrange Multiplier Method [24]

The Augmented Lagrange Multiplier (ALM) method uses the augmented Lagrangian in the form:

$$A(x, \lambda, k_p) = f(x) + \sum_{i=1}^{m} [\lambda_i \ \psi_i + k_p \psi_i^2] + \sum_{j=1}^{l} [\lambda_{j+m} h_j(x) + k_p h_j^2(x)] \qquad (7)$$

where

$$\psi_i = max\left[c_i(x), \frac{-\lambda_i}{2k_p}\right]$$

The updated replication formulas for Lagrange multipliers are

$$\lambda_i^{p+1} = \lambda_i^p + 2k_p \left\{max\left[c_i(x), \frac{-\lambda_i^p}{2k_p}\right]\right\} \qquad i=1,m$$

and

$$\lambda_{j+m}^{p+1} = \lambda_{j+m}^p + 2k_p h_j(x) j = 1, l$$

The ALM method is overall considered to be more robust than the interior or exterior methods of the sequential unconstrained minimization technique[22,21]. After several years Polyak[19] introduced and analyzed the new Log-Sigmoid (LS) multipliers method for constrained optimization. The LS method is a homogenization technique recently developed as a Lagrange addition to the penalty method. At the same time, the LS method contains some specific characteristics, making them fundamentally different from other non-quadrature Lagrangian techniques. He established and estimated the approximation rate of the LS method under very moderate assumptions on input data and under the second standard stability condition for accurate and inaccurate minimization.

4. NEW PROPOSED ALGORITHMS

Firefly Based on the above interpretation of the two functions, the approach of the penalty function method for Polyak's log-sigmoid and the augmented Lagrange function approach was incorporated in the FA. The illustrations of EFA , H1FA , H2FA and MFA are presented in **(Figures 3,4,5 and 6**) respectively. Firefly algorithm is better than GA and PSO in terms of efficiency reach the solution and rate of convergence as described in Yang[28].To illustrate the new idea in the algorithms presented in this paper, we can look at the following points:

Extended Firefly Algorithm (EFA):

Through the review presented in the previous section we have observed the possibility of improvement of the firefly algorithm through the exterior penalty function, we have tried the same subject, but through the interior penalty functionand specifically using the function Polyak's Log-Sigmoid, a stronger improvement of the previous (penalty firefly)as given by the results of the next section and **(Table 1a-1b)** and we can write the link relationship as follows:. We can write the link relationship as:

Polyak's Log-Sigmoid + Firefly algorithm = EFA algorithm

The First Hybrid Algorithm (H1FA):
The second algorithm was a hybrid between two functions in the convex structure method to take advantage of both functions and properties. This algorithm gave a good performance through the results of the next section and **(Table 2a-2b**) and we can write the correlation relationship as follows:

θ (Polyak's Log-Sigmoid)+(1-θ)(Penalty)+Firefly algorithm = H1FA algorithm

The Second Hybrid Algorithm (H2FA):
After the good results obtained from hybridization in the second algorithm we experimented with hybridization of a different type between Augmented Lagrange Multiplier and Polyak's Log-Sigmoid functions also using the convex structure. The results obtained in the (**Table 3a-3b)** guide the efficient performance of the third algorithm and we can write the correlation relationship as follows:

θ (Polyak's Log-Sigmoid)+(1-θ)(Augmented Lagrange Multiplier)+Firefly algorithm = H2FA algorithm

Modified Firefly Algorithm (MFA):
Finally, the fourth algorithm was inspired by the idea of hybridization but in a completely new style and significantly changed the results as in **(Table 4a-4b**) of the next chapter with the use of two values to multiply them by the numerator and denominator. We can write the link relationship as follows:

(τ1 Polyak's Log-Sigmoid /τ2 Penalty)+Firefly algorithm=MFA algorithm

Now we review the detailed schemas of the new algorithms:

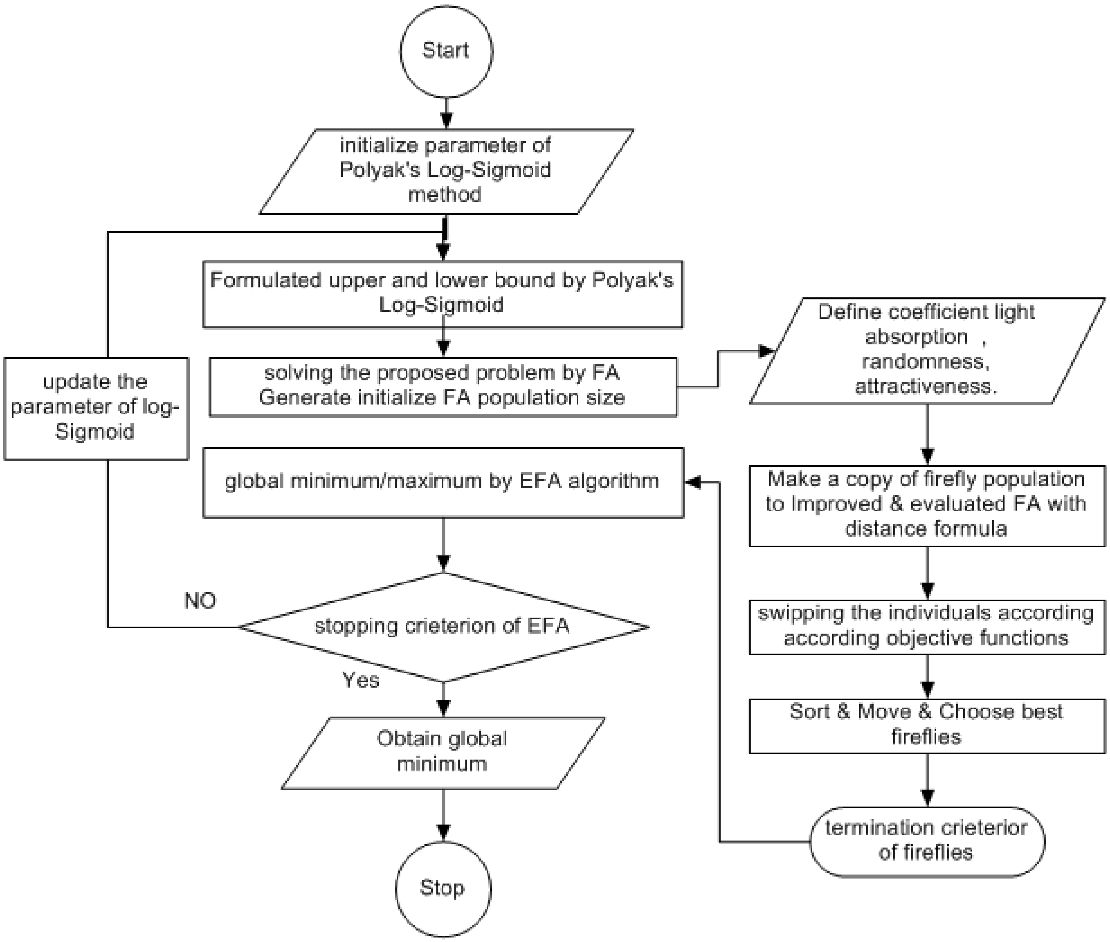

Fig. 3: Flow chart of proposed EFA

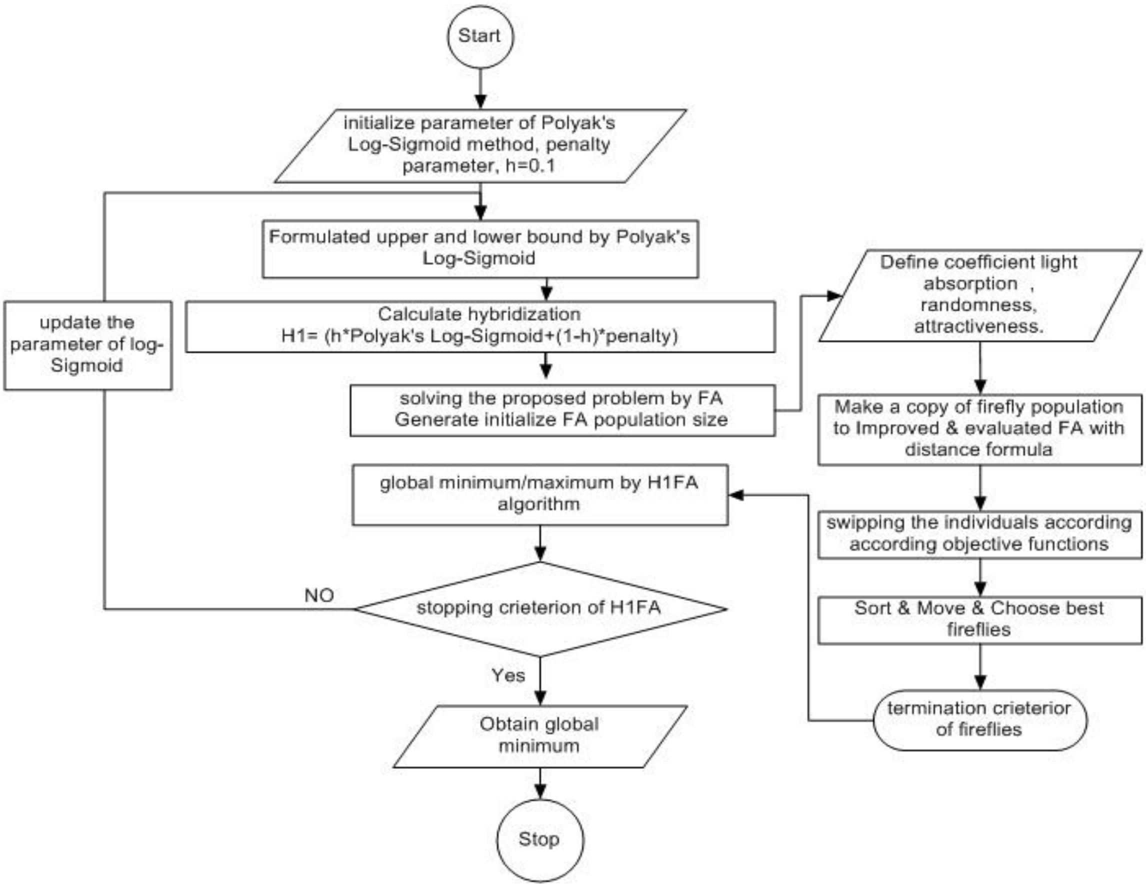

Fig. 4: Flow chart of proposed H1FA

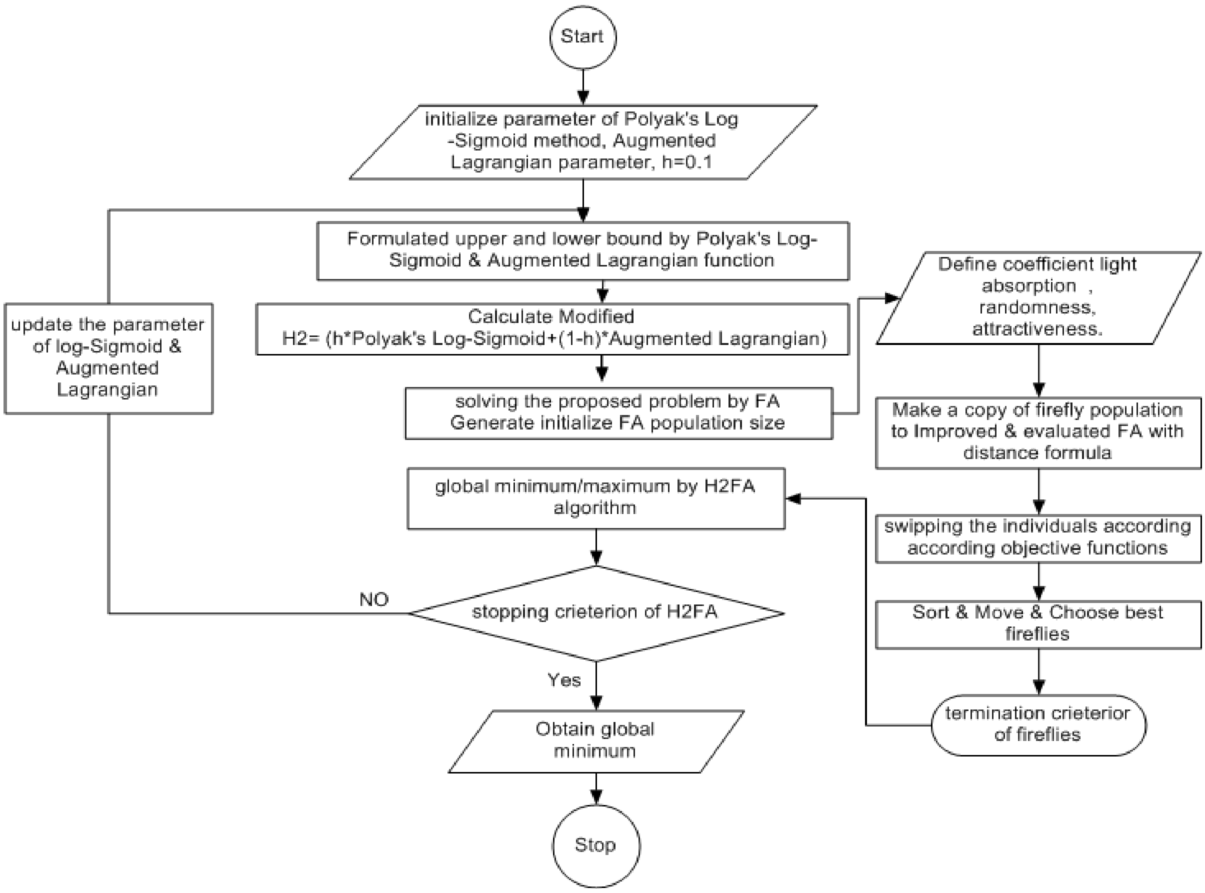

Fig. 5: Flow chart of proposed H2FA

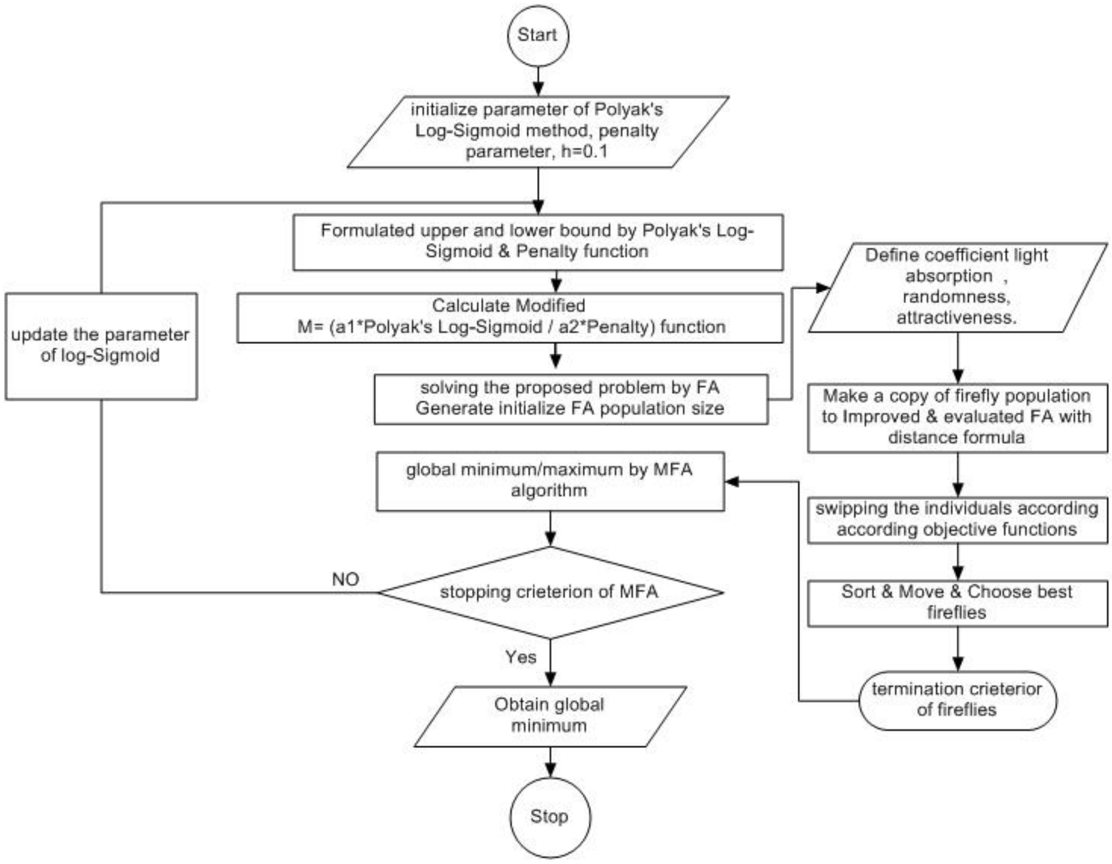

Fig. 6: Flow chart of proposed MFA

5. NUMERICAL EXAMPLES

From the pseudo code, it is relatively straight forward to implement the Firefly Algorithm using a popular programming language such as Matlab The proposed (EFA, H1FA, H2FA and MFA) has been implemented in Matlab 2011 programming language and run on a PC with Intel(R) Core(TM) i5. An improved firefly algorithm and Windows 7 x 32 Pro operating system. The optimization results obtained by the E-FA were compared with the same of the original FA [28]. The results of the algorithms used for comparison with the proposed (EFA, H1FA, H2FA and MFA). We have tested it against more than a dozen test

Nonlinear test problem for optimization

A number of standard nonlinear functions (see **Appendix**) were selected to compare the new algorithms and the FA algorithm. To demonstrate the efficiency of the new algorithms, the standard FA algorithm was compared with the algorithms proposed in the research. The results were compared using statistical analysis to evaluate the efficiency of new algorithms based on:

1- Mean: is defined by the following law

Mean= $\frac{\sum_{i=1}^{n} x_i}{n}$

2- Standerd Diviation: is defined by the following law

s.t(x)=$\sqrt{\frac{\sum_{i=1}^{n}(x_1-\bar{x})^2}{n-1}}$

3- Covariance: is defined by the following law

Cov(x, y)=$\frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{n-1}$

The statistical results in all Tables are showed that the functions (1-18) have a small standard deviation of the new method compared to the standard method and that the amount of dispersion Little fireflies.

Engineering Applications

Now we can apply Firefly algorithm to perform various design optimization tasks. In principle, any improvement problems can be solved by genetic algorithms and particle swarm enhancements can be solved by Firefly algorithm. For simplicity to be effective in improving the real world, we use the Football Association to find the best solution for the standard, but it is very difficult [5] . This is a well-known test problem for optimization and it can be defined as:

1. **Compression spring design optimization problem**

This problem[4] minimizes the weight of a compression spring (**Figure 7**), subject to Minimum deviation constraints, shear stress, frequency of mutation, limits on external diameter and design variables. There are three design variables: wire diameter x_1, the mean coil diameter x_2, and the number of active coils x_3.

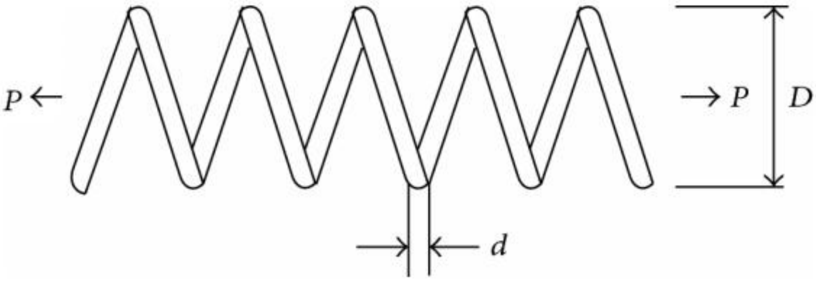

Fig. 7: Compression Spring.

2. Speed Reducer design optimization problem

The speed reduction design [4] shown in (**Figure 8**), with the x_1 face width, is the tooth unit x_2, number of teeth on pinion x_3, length of the first shaft between bearings x_4, length of the second shaft between bearings x_5, diameter of the first shaft x_6, and diameter of the first shaft x_7 (all variables continuous except x3 that is integer). The weight of the reducer should be reduced in accordance with the gear bend limits of the gear teeth, surface stress, and the occasional deviations of columns and pressures in the column.

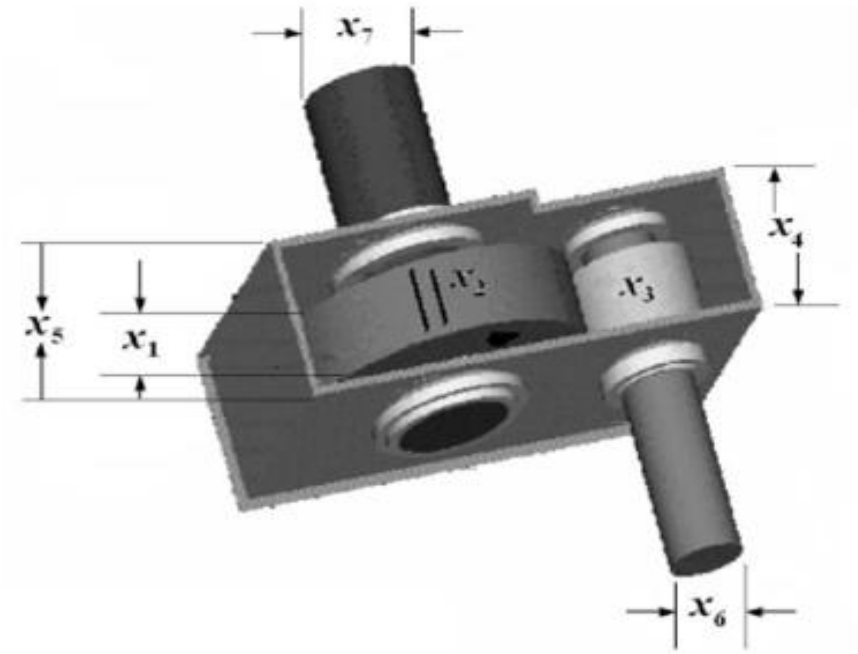

Fig. 8: Speed Reducer.

3. Welded beam design optimization problem

The problem is to design a welded beam for minimum cost, subject to some constraints[4]. (Figure 9) shows the welded beam structure which consists of a beam A and the weld required to hold it to member B. The objective is to find the minimum fabrication cost, consideration four design variables: x_1, x_2, x_3, x_4 and constraints of shear stress τ , bending stress in the beam σ, buckling load on the bar Pc, and end deflection on the beam δ.

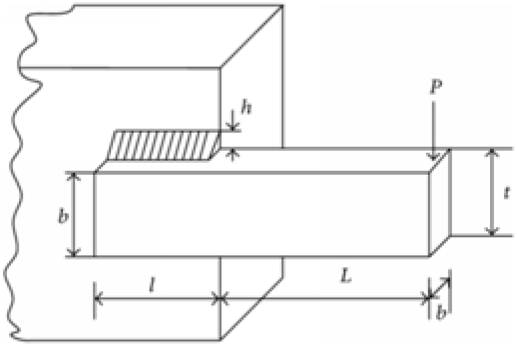

Fig. 9: Welded Beam.

Table 1a: Comparing FA Algorithm against EFA in 2-dimension models for (best objective, mean, Standard Division, Covariance) total N of function evaluation=20000

function	FA				EFA			
	Best obj.	Mean	Std	Covariance	Best obj.	mean	Std	covariance
F1	3.3746e+015	1.6842	0.8969	0.8044	4.9517e+005	3.3444	0.5286	0.2794
F2	3.4471e+016	2.025	0.0354	0.0012	8.4686e+016	2.5250	0.6718	0.4513
F3	1.9315e+015	2.2152	1.1098	1.2317	1.4640e+004	1.9904	0.4669	0.2180
F5	5.1453e+017	3.025	2.7931	7.8012	1.2155e+015	1.5250	0.6718	0.4512
F6	5.1816e+005	1.7626	1.2020	1.4448	1.0572e+005	1.6918	1.2464	1.5535
F7	2.5000e+012	1.3761	0.4612	0.2127	-0.0469	1.4006	0.2767	0.0766
F8	2.8046	1.2966	0.3487	0.1216	4.7939e+004	1.4088	0.1753	0.0307
F10	8.6963e+003	0.7279	0.7071	0.5	2.2778e+003	1.6135	0.7071	0.5000
F11	1.0000e+015	1.2771	1.0223	1.0450	1.0000e+015	1.7508	0.3525	0.1242
F12	1.0000e+015	2	0	0	1.0000e+015	2.9588	1.3559	1.8385
F13	6.7089e+004	0.8698	0.4936	0.2436	992.9370	0.8769	0.5435	0.2954
F14	5.7584e+014	0.9123	0.5484	0.3007	4.4100e+017	1.5250	0.6718	0.4512
F15	7.2810e+003	0.9713	0.3366	0.1133	6.1294e+003	0.8865	0.6543	0.4282

Table 1b: Comparing FA Algorithm against EFA in 3-dimension and Engineering models for (total N of function evaluation=20000 & n of fireflies=40)

function	FA				EFA			
	Best obj.	Mean	Std	covariance	Best obj.	Mean	Std	covariance
F4	2.0100e+015	1.0167	0.9751	0.9508	1.5688e+015	1.1	0.9836	0.9675
F9	4.3213	3.7667	2.0404	4.1633	-103.2000	4.7667	1.0786	1.1633
Spring	0.0127	3.5812	5.1576	26.6	90000	3.8333	3.2532	10.5833
Speed	2.504	8.024	9.1471	83.5497	1.345	6.6509	5.8031	33.6761
Beam	1.766	3.3505	4.2119	0.1877	0.7508	0.8007	0.501	0.251

Table 2a: Comparing FA Algorithm against H1FA in 2-dimension models for (best objective, mean, Standard Division, Covariance) total N of function evaluation=20000

function	FA				H1FA			
	Best obj.	Mean	Std	Covariance	Best obj.	Mean	Std	covariance
F1	3.3746e+015	1.6842	3.9175	0.8044	1.3534e+016	2.3708	0.8898	0.7918
F2	3.4471e+016	2.025	3.8633	0.0012	1.8383e+018	4.7750	1.0253	1.0513
F3	1.9315e+015	2.2152	10.5625	1.2317	1.5894e+015	4.4537	2.1868	4.7821
F5	5.1453e+017	3.025	10.9354	7.8012	8.0684e+017	3.9000	0.5657	0.3200
F6	5.1816e+005	1.7626	1.6301	1.4448	2.1774e+005	2.2289	0.8954	0.8018
F7	2.5000e+012	1.3761	3.9175	0.2127	1.1688e+017	2.6500	0.8485	0.7200
F8	2.8046	1.2966	3.8633	0.1216	3.7521e+016	2.1500	0.1414	0.0200

F10	8.6963e+003	0.7279	10.5625	0.5	1.5610e+015	1.6500	0.8485	0.7200
F11	1.0000e+015	1.2771	10.9354	1.0450	2.2500e+014	2.7500	0.3535	0.1250
F12	1.0000e+015	2	1.6301	0	4.7610e+015	3.6500	0.9192	0.8450
F13	6.7089e+004	0.8698	3.9175	0.2436	3.4225e+015	1.1500	0.1414	0.0200
F14	5.7584e+014	0.9123	3.8633	0.3007	4.5066e+017	1.5250	0.6718	0.4512
F15	7.2810e+003	0.9713	10.5625	0.1133	1.1248e+004	0.9996	0.0394	0.0016

Table 2b: Comparing FA Algorithm against H1FA in 3-dimension and Engineering models for (total N of function evaluation=20000 & n of fireflies=40)

function	FA				H1FA			
	Best obj.	Mean	Std	covariance	Best obj.	Mean	Std	covariance
F4	2.0100e+015	1.0167	0.9751	0.9508	1.7256e+018	4.1000	1.9793	3.9175
F9	4.3213	3.7667	2.0404	4.1633	3.4083e+018	4.7667	1.9655	3.8633
Spring	0.0127	3.5812	5.1576	26.6	1.0406e+016	4.2500	3.2500	10.5625
Speed	2.504	8.024	9.1471	83.5497	2.155	5.9755	3.3069	10.9354
Beam	1.766	3.3505	4.2119	0.1877	6.2061	2.0439	1.2767	1.6301

Table 3a: Comparing FA Algorithm against H2FA in 2-dimension models for (best objective, mean, Standard Division, Covariance) total N of function evaluation=20000

Function	FA				H2FA			
	Best obj.	Mean	Std	covariance	Best obj.	Mean	Std	Covariance
F1	3.3746e+015	1.6842	0.8969	0.8044	1.2476e+015	1.5215	1.1618	1.3498
F2	3.4471e+016	2.025	0.0354	0.0012	1.6925e+015	1.2750	0.3182	0.1012
F3	1.9315e+015	2.2152	1.1098	1.2317	1.4174e+015	3.6456	1.9153	3.6685
F5	5.1453e+017	3.025	2.7931	7.8012	3.5830e+016	3.9000	0.5657	0.3200
F6	5.1816e+005	1.7626	1.2020	1.4448	1.2502e+012	2.7346	0.5388	0.2903
F7	2.5000e+012	1.3761	0.4612	0.2127	1.1463e+016	2.6250	0.8839	0.7813
F8	2.8046	1.2966	0.3487	0.1216	1.5770e+015	1.3761	0.4612	0.2127
F10	8.6963e+003	0.7279	0.7071	0.5	1.0750e+015	3.1250	1.2374	1.5313
F11	1.0000e+015	1.2771	1.0223	1.0450	1.4081e+011	2.5000	0.7072	0.5001
F12	1.0000e+015	2	0	0	1.0000e+015	2	0	0
F13	6.7089e+004	0.8698	0.4936	0.2436	1.0980e+016	1.3500	1.2021	1.4450
F14	5.7584e+014	0.9123	0.5484	0.3007	2.3364e+006	0.8027	0.5385	0.2900
F15	7.2810e+003	0.9713	0.3366	0.1133	2.4734e+015	1.3750	0.1768	0.0313

Table 3b: Comparing FA Algorithm against H2FA in 3-dimension and Engineering models for(total N of function evaluation=20000 & n of fireflies=40)

function	FA				H2FA			
	Best obj.	Mean	Std	covariance	Best obj.	Mean	Std	Covariance
F4	2.0100e+015	1.0167	0.9751	0.9508	1.0920e+016	2.1000	0.9836	0.9675
F9	4.3213	3.7667	2.0404	4.1633	1.0539e+017	3.4333	0.5132	0.2633
Spring	0.0127	3.5812	5.1576	26.6	2.5733	1.8333	1.6042	2.5733
Speed	2.504	8.024	9.1471	83.5497	1.504	3.0716	0.5109	0.2611
Beam	1.766	3.3505	4.2119	0.1877	1.463	1.0914	0.4586	0.2103

Table 4a: Comparing FA Algorithm against MFA in 2-dimension models for(best objective, mean, Standard Division, Covariance) total N of function evaluation=20000

Function	FA				MFA			
	Best obj.	Mean	Std	covariance	Best obj.	Mean	Std	Covariance
F1	3.3746e+015	1.6842	0.8969	0.8044	2.6433e+006	2.4984	0.7076	0.5006
F2	3.4471e+016	2.025	0.0354	0.0012	1.1900e+006	1.4114	0.1252	0.0157
F3	1.9315e+015	2.2152	1.1098	1.2317	1.4286e+006	2.7021	0.8024	0.6438

F5	5.1453e+017	3.025	2.7931	7.8012	1.5524e+016	1.9000	0.8485	0.7200
F6	5.1816e+005	1.7626	1.2020	1.4448	1.0279e+007	3.4717	0.0173	0.0003
F7	2.5000e+012	1.3761	0.4612	0.2127	1.1157e+017	2.6250	0.8839	0.7813
F8	2.8046	1.2966	0.3487	0.1216	2.4067e+006	1.3053	0.7697	0.5925
F10	8.6963e+003	0.7279	0.7071	0.5	2.8391e+005	1.9757	0.7071	0.5
F11	1.0000e+015	1.2771	1.0223	1.045	1.1025e+015	4.0500	0	0
F12	1.0000e+015	2	0	0	1.0000e+015	1.9441	0.0791	0.0063
F13	6.7089e+004	0.8698	0.4936	0.2436	1.1236e+017	2.8500	0.4950	0.2450
F14	5.7584e+014	0.9123	0.5484	0.3007	4.3867e+006	0.7928	0.6371	0.4058
F15	7.2810e+003	0.9713	0.3366	0.1133	3.2852e+015	1.3750	0.1768	0.0313

Table 4b: Comparing FA Algorithm against MFA in 3-dimension and Engineering models for(total N of function evaluation=20000 & n of fireflies=40)

Function	FA				MFA			
	Best obj.	Mean	Std	covariance	Best obj.	Mean	Std	Covariance
F4	2.0100e+015	1.0167	0.9751	0.9508	1.5688e+015	1.1000	0.9836	0.9675
F9	4.3213	3.7667	2.0404	4.1633	1.1105e+010	3.4333	0.5132	0.2633
Spring	0.0127	3.5812	5.1576	26.6	0.0027	1.3721	1.1859	0.2079
Speed	2.504	8.024	9.1471	83.5497	2.004	3.4703	1.6358	2.6757
Beam	1.766	3.3505	4.2119	0.1877	0.1549	1.6256	1.9534	3.8159

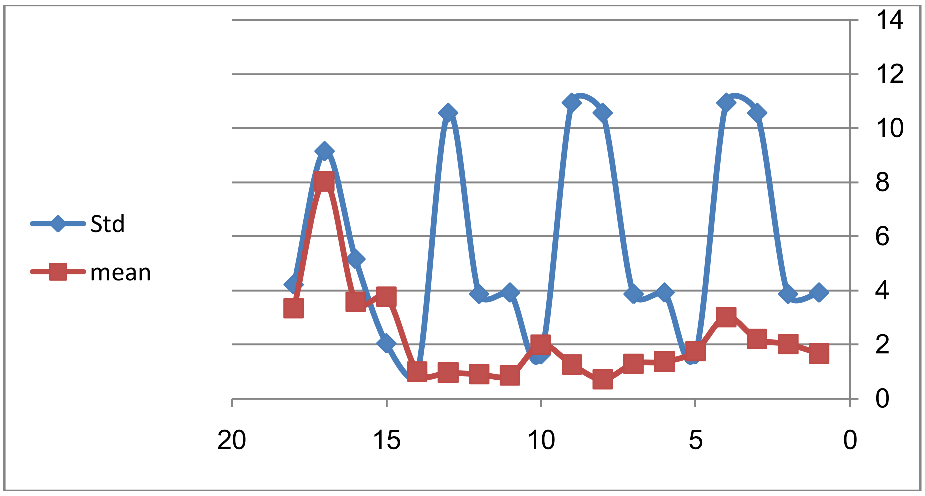

Fig. 10: Traditional FA

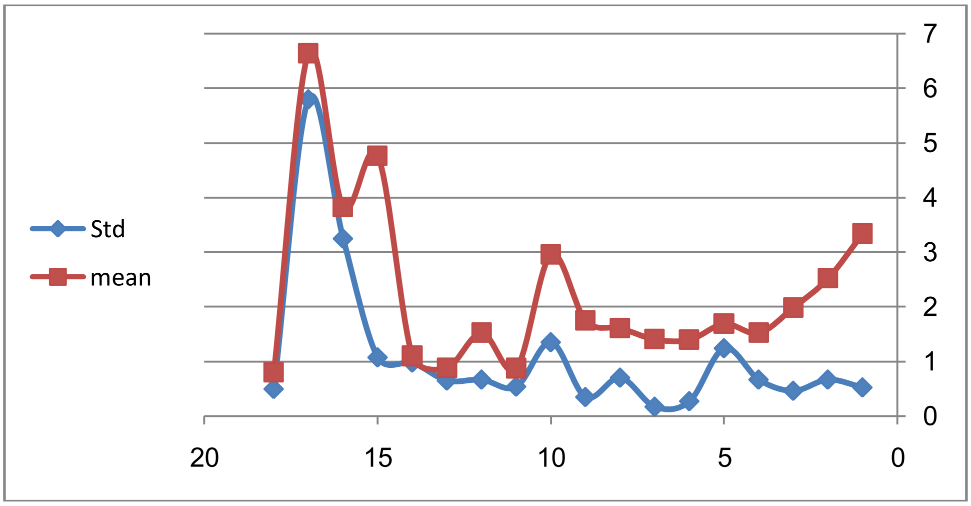

Fig. 11: Extended EFA

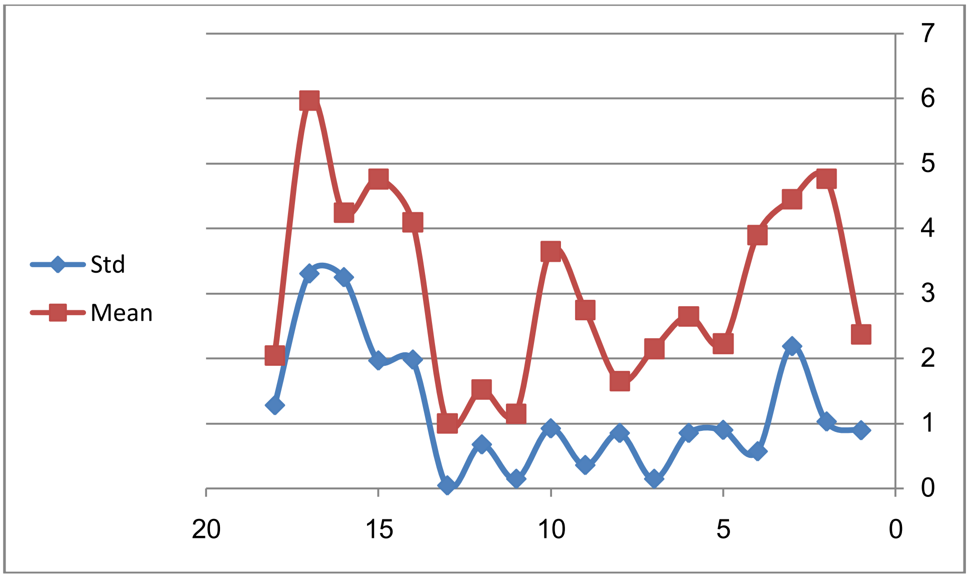

Fig. 12: Hybrid (1) H1FA

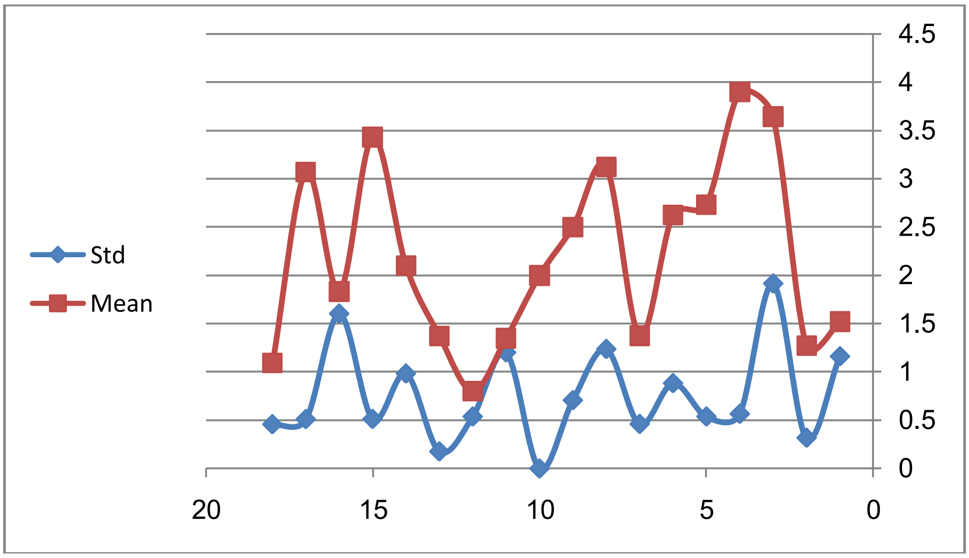

Fig. 13: Hybrid (2) H2FA

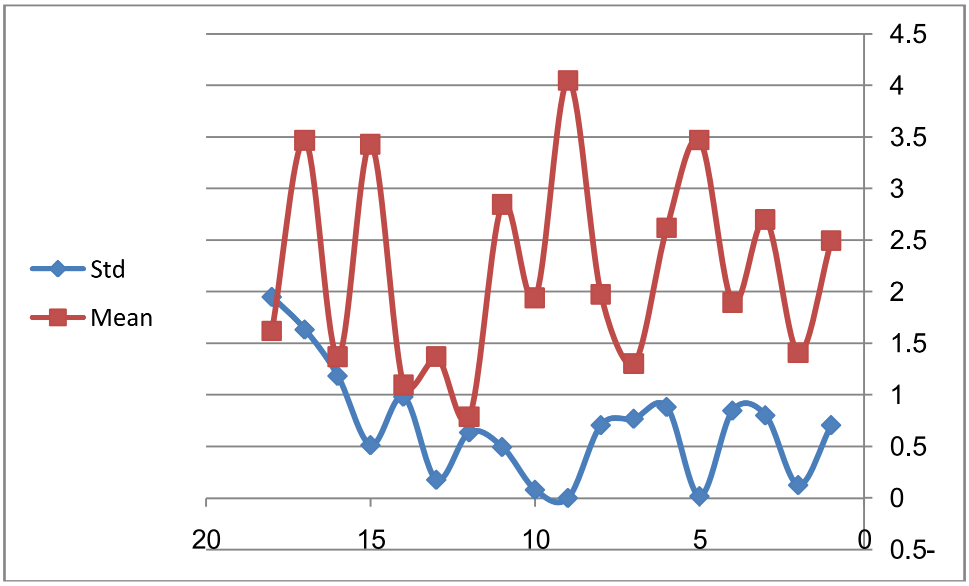

Fig. 14: Modified MFA

CONCLUSION

The statistical results in (**Tables 1-4**) showed that the functions (1-18) have a small standard deviation of the new method compared to the standard method and that the amount of dispersion of fireflies is small. In this way we have been able to demonstrate the ability of the algorithms provided in the research to overcome the firefly algorithm in the value of the optimal solution to the target function. The final forms of the new algorithms showed a good performance compared to the traditional firearm algorithm and the nonlinear and geometric functions in this study.

Appendix

1-$min f(x) = (x_1 - 3)^2 + (x_2 - 2)^2$
s.t

$$x_1^2 - x_2^2 + 5 = 0$$
$$x_1 + 2x_2 - 4 \leq 0$$

2-$min f(x) = x_1^2 - x_1 x_2 + x_2^2$
s.t

$$x_1^2 + x_2^2 - 4 = 0$$
$$2x_1 - x_2 + 2 < 0$$

3-$min f(x) = (x_1 - 2)^2 + (x_2 - 1)^2$
s.t

$$x_1 - 2x_2 = -1$$
$$-\frac{x_1^2}{4} + x_2^2 + 1 \geq 0$$

4-$min f(x) = x_1^3 + 2x_2^2 x_3 + 2x_3$
s.t

$$x_1^2 + x_2 + x_3^2 = 4$$
$$x_1^2 - x_2 + 2x_3 \leq 2$$

5-$min f(x) = e^1 - x_1 x_2 + x_2^2$
s.t

$$x_1^2 + x_2^2 = 4$$
$$2x_1 + x_2 \leq 2$$

6-$min f(x) = x_1^3 - 3x_1 x_2 + 4$
s.t

$$-2x_1 + x_2^2 = 5$$
$$5x_1 + 2x_2 \geq 18$$

7-$min f(x) = -e^{-x_1 - x_2}$
s.t

$$x_1^2 + x_2^2 - 4$$
$$x_1 - 1 \geq 0$$

8-$min f(x) = -x_1^2 + 2x_1 x_2 + x_2^2 - e^{-x_1 - x_2}$
s.t

$$x_1^2 + x_2^2 - 4 = 0$$
$$x_1 + x_2 \leq 1$$

9-$min f(x) = -x_1 x_2 x_3$
s.t
$20 - x_1 \geq 0$

$$11 - x_2 \geq 0$$
$$42 - x_3 \geq 0$$
$$72 - x_1 - 2x_2 - 2x_3 \geq 0$$

10-$min f(x) = (x_1 - 1)^2 + x_2 - 2$
s.t

$$x_2 - x_1 = 1$$

$x_1 + x_2 \geq 2$
11-$min f(x) = {x_1}^2 + {x_2}^2$
s.t

$$x_1 - 3 = 0$$
$$x_2 - 2 \leq 0$$

12-$minf(x) = \frac{1}{4000}({x_1}^2 + {x_2}^2) - \cos(x_1)\cos\left(\frac{x_2}{\sqrt{2}}\right) + 1$

s.t

$$x_1 - 3 = 0$$
$$x_2 - 2 \leq 0$$

13-$minf(x) = (x_1 - 2)^2 + \frac{1}{4}{x_2}^2$

s.t

$$2x_1 + 3x_2 = 4$$
$$x_1 - \frac{7}{2}x_2 \leq 1$$

14-$minf(x) = -x_1x_2$

s.t

$$20x_1 + 15x_2 - 30 = 0$$
$$\frac{{x_1}^2}{4} + {x_2}^2 - 1 \leq 0$$

15-$minf(x) = {x_1}^4 - 2{x_1}^2x_2 + {x_1}^2 + x_1{x_2}^2 - 2x_1 + 4$

s.t

$${x_1}^2 + {x_2}^2 - 2 = 0$$
$$0.25\,{x_1}^2 + 0.75{x_2}^2 - 1 \leq 0$$

REFERENCES

[1]. Apostolopoulos T. and Vlachos A., (2011), "Application of the Firefly Algorithm for Solving the Economic Emissions Load Dispatch Problem", International Journal of Combinatorics, Hindawi Publishing Corporation, ID: 523806, 23.

[2]. Auger A. and Doerr B.,(2010): "Theory of Randomized Search Heuristics: Foundations and Recent Developments", Series on (Vol 1) Theoretical Computer Science, World Scientific, ISSN: 1793-849X .

[3]. Auger A. and Teytaud O., (2010), "Continuous lunches are free plus the design of optimal optimization algorithms". Algorithmica, 57, 121–146.

[4]. Cagnina L.C., Esquivel S.C. and Coello C.A., (2008) "Solving engineering optimization problems with the simple constrained particle swarm optimizer", Informatica, 32, 319-326.

[5]. Deb K., (1995), "Optimization for Engineering Design", Prentice-Hall, New Delhi.

[6]. Dorigo M. and Stutzle T., (2004), "Ant Colony Optimization", MIT Press.

[7]. Goldberg D.E., (1989), "Genetic Algorithms in Search", Optimization and Machine Learning, Reading, Mass.: Addison Wesley.

[8]. Gupta A. and Padhy P.K., (2016), "Modified Firefly Algorithm based controller design for integrating and unstable delay processes", Engineering Science and Technology, an International Journal 19 , 548–558.

[9]. Holland J. H., (1975), "Adaptation in Natural and Artificial Systems", University of Michigan Press, Ann Arbor.

[10]. Johari N. F., Zain A. M., Mustaffa N. H. and Udin A., (2017), "Machining Parameters Optimization using Hybrid Firefly Algorithm and Particle Swarm Optimization", IOP Conf. Series: Journal of Physics: Conf. Series 892 , doi :10.1088/1742-6596/892/1/012005.

[11]. Karthikeyan S., Asokan P. and Nickolas S., (2014), "A hybrid discrete firefly algorithm for multi-objective flexible job shop scheduling problem with limited resource constraints", Int J Adv Manuf Technol, 72:1567–1579.

[12]. Kennedy J., Eberhart R. and Shi Y., (2001), "Swarm intelligence", Academic Press.

[13]. Kennedy J. and Eberhart R.C., (1995), "Particle swarm optimization", Proc. of IEEE International Conference on Neural Networks, Piscataway, NJ., 1942-1948.

[14]. Kennedy J. and Eberhart R., (1995): "Particle swarm optimization", In: Proc. of the IEEE Int. Conf. on Neural Networks, Piscataway, NJ, 1942–1948.

[15]. Lukasik S. and Zak S., (2009), "Firefly algorithm for continuous constrained optimization tasks", Lecture Notes in Artificial Intelligence (Eds. N. T. Ngugen, R. Kowalczyk, S. M. Chen), 5796, 97-100.

[16]. Nayak J., Naik B. and Behera H.S., (2016), "A novel nature inspired firefly algorithm with higher order neural network: Performance analysis", Engineering Science and Technology, an International Journal, 19, 197–211.

[17]. Neumann F., Witt C., (2010): "Bioinspired Computation in Combinatorial Optimization", Algorithms and Their Computational Complexity. Springer.

[18]. Parpinelli R. S., Lopes H.S., (2011), "New inspirations in swarm intelligence", a survey. Int. J. Bio-Inspired Computation 3, 1–16**.**

[19]. Polyak R.A., (2001), "Log-Sigmoid Multipliers Method in Constrained Optimization", Annals of Operations Research, Kluwer Academic Publishers, Manufactured in the Netherlands. 101, 427-460.

[20]. Polyak R. A., Griva I. and Sobieski J., (1998), "The Newton Log-Sigmoid Method in Constrained Optimization", Proc. 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, St. Louis, MI, Sept. 2-4, AIAA paper, AIAA-98-4797.

[21]. Powell M.J.D., (1981), "Optimization Algorithm sin 1979", Committeeon Algorithms Newsletter, Mathematical Programming Society, 5, 2-16.

[22]. Rockafellar R.T., (1973), "The Multiplier Method of Hestenes and Powell Applied to Convex Programming", J. Optim. Theory Appl., 12, 6, 555-562.
[23]. Vanderplaats G.N., (1999): "Numerical Optimization Techniques for Engineering Design-With Applications", McGraw-Hill, 3rd edition published by VR&D.
[24]. Vanderplaats G. N., (2002): "Very Large Scale Optimization", Vanderplaats Research and Development, Inc., Colorado Springs, Colorado, NASA/CR-2002-211768 McGraw-Hill.
[25]. Yang X.S., (2010), "Engineering Optimization: An Introduction with Metaheuristic Applications". John Wiley and Sons, Hoboken, NJ, USA.
[26]. Yang X.S., (2005), "Biology-derived algorithms in engineering optimization", in Handbook of Bioinspired Algorithms and Applications (eds S. Olarius & A. Y. Zomaya), Chapman & Hall /CRC.
[27]. Yang X.S., (2008), "Firefly algorithm, Nature-Inspired Metaheuristic Algorithms" Luniver Press, UK., 79–90.
[28]. Yang X.S., (2009), "Firefly algorithms for multimodal optimization", in: Stochastic Algorithms: Foundations and Appplications (Eds O. Watanabe and T. Zeugmann), SAGA, Lecture Notes in Computer Science, 5792, Springer-Verlag, Berlin, 169-178.
[29]. Yang X.S., (2010), "Firefly algorithm, stochastic test functions and design optimization", International Journal of Bio-Inspired Computation, 2, 2, 78–84.
[30]. Yelghi A. and Köse C., (2018), "A modified firefly algorithm for global minimum optimization", Applied Soft Computing, Elsevier publisher, Science Direct, 62, 29–44.
[31]. LouzazniM. ,Khouya A., Amechnoue K., Gandelli A.,Mussetta M., andCrăciunescu A., (2018), " Metaheuristic Algorithm for Photovoltaic Parameters: Comparative Study and Prediction with a Firefly Algorithm",Applied Science,8, 3, 339, doi:org/10.3390/app8030339.
[32]. McCaffrey J.,(2015), "Test Run - Firefly Algorithm Optimization", MSDN Magazines, 30, 6.