

An Experimental Study of Moth-Flame Algorithm using Benchmark Functions

Davinder Kaur¹, Harinder Pal Singh²P.G. Student, Department of Electrical Engineering, SBS State Technical Campus, Ferozepur, Punjab, India¹Assistant Professor, Department of Electrical Engineering, SBS State Technical Campus, Ferozepur, Punjab, India²

ABSTRACT: In this paper, a nature-inspired population based optimization algorithm proposed by Mirjalili in 2015, popularly known as Moth Flame Optimization (MFO), is demonstrated. The main inspiration of this paradigm is navigation manner of moths in cosmos known as transverse orientation. This orientation is used to solve the various optimization problems by mathematical modeling the behavior of the moths to execute optimization. To demonstrate the performance, the MFA (Moth-Flame Algorithm) is applied on six benchmark functions and are compared with the Flower Pollination Algorithm (FPA), Genetic Algorithm (GA), Firefly Algorithm (FA), Particle Swarm optimization (PSO). The statistical output derived using benchmark functions delineates the algorithm's efficiency is competitive and promising with another algorithm.

KEYWORDS: Optimization, MFO Algorithm, Benchmark Functions.

I. INTRODUCTION

A nature-inspired population based Moth Flame Algorithm (MFA) [1] based on the transverse orientation of moths in cosmos. This transverse orientation means maintaining a fixed angle with respect to the moon for the navigation in straight direction during the night. In MFA, moths fly in a logarithmic spiral and finally converge towards the flame. The spiral method delineates the exploration area and it guarantees to exploit the optimum solution.

Classical methods such as Lambda Iteration, Newton's method, and Lagrange Multiplier method [2] can solve the numerous optimization problems. However, in reality, there is distinct non-convexity of optimization issues for obtaining the solutions. Classical calculus-based methods cannot address this type of problem adequately and lead to suboptimal solutions. Dynamic programming [3] can be used to solve the optimization problems with cost discontinuous and nonlinear in nature, but it is computationally expensive and suffers from finding only local optima owing to premature convergence. To overcome the problems of—restriction in shape of cost curves, unidirectional search, premature and slow convergence, suboptimal solution and significant computational overhead, non-conventional stochastic and intelligent techniques are now used to resolve the complexities of problems reasonably well. These techniques include Particle Swarm Optimization (PSO) [4-6], Genetic Algorithm (GA) [7], Firefly Algorithm (FA)[8], Flower Pollination Algorithm (FPA)[9], besides others. Unlike classical optimization methods, the intelligent stochastic techniques such as Artificial Neural Network (ANN) [10], Fuzzy Logic Control (FLC) [11], Hopfield Neural Network (HNN)[12] etc work on a population of possible solutions in the search space and create an advantage of getting multiple suitable solutions in a single run, they are easy to implement, robust, and computationally less expensive. When applied to complex optimization problems like the ELD problem, these techniques have a high probability of finding the optimal quickly through competition and collaboration among the possible solutions. According to NFL (No-Free-Launch) theorem [1] there is no optimization algorithm for solving all optimization problems. Therefore, there are still problems that can be solved by new optimizers better than the existing one.

Paper is organized as follows. Section II describes the mathematical modelling of MFA and also the position updating criterion of moths corresponding to flames. The test problems using various types of benchmark functions for testing is given in Section III. Section IV presents experimental results showing results of MFA tested using benchmarks. Finally, Section V presents conclusion.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 8, August 2017

II. MOTH FLAME ALGORITHM

The Moth-Flame Algorithm is proposed by Seyedali Mirjalili in 2015[1].MFA is population based algorithm. The MFO algorithm is three-tuple that estimates the global optimal of the optimization problems and defined as follows:

$$MFO = (I, P, T) \quad \dots\dots\dots (2.1)$$

I is a function that forms a random population of moths and related fitness values. The function P is the main function, moves the moths around the search area. This function received the matrix of M and returns its updated one in end.

The function T returns true if the termination criterion is fulfilled and false if the termination criterion is not fulfilled. After the initialization, the P function iteratively runs until the function T returns true. The P function is the main function that moves the moths around the search scope. As transverse orientation is the main stimulation of this algorithm.

we represent the set of moths as follows:

$$M = \begin{bmatrix} m_{1,1} & m_{1,2} & \dots & \dots & m_{1,d} \\ m_{2,1} & m_{2,2} & \dots & \dots & m_{2,d} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ m_{n,1} & m_{n,2} & \dots & \dots & m_{n,d} \end{bmatrix} \quad \dots\dots\dots (2.2)$$

Where n is the number of moths and d is the number of variables (dimensions).

For all moths, we also suppose that there is an array for storing the corresponding fitness values as follows:

$$OM = \begin{bmatrix} OM_1 \\ OM_2 \\ \vdots \\ OM_n \end{bmatrix} \quad \dots\dots\dots (2.3)$$

Where n is the number of moths.

Note that the fitness value is return value of the objective (fitness) function for each moth. The position vector (first row in the matrix M for example) of each moth is passed to the fitness function and the output of the fitness function is assigned to the corresponding moth as its fitness function (OM1 in the matrix OM for example).

Other main components in the algorithm are flames. We suppose a matrix similar to the moth matrix as follows:

$$F = \begin{bmatrix} F_{1,1} & F_{1,2} & \dots & \dots & F_{1,d} \\ F_{2,1} & F_{2,2} & \dots & \dots & F_{2,d} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ F_{n,1} & F_{n,2} & \dots & \dots & F_{n,d} \end{bmatrix} \quad \dots\dots\dots (2.4)$$

Where n is the number of moths and d is the number of variables (dimension).

It can be observable from above equation that the dimension of M and F arrays are equal. For the flames, we also suppose that there is an array for storing the corresponding fitness values as follows:

$$OF = \begin{bmatrix} OF_1 \\ OF_2 \\ \vdots \\ OF_n \end{bmatrix} \quad \dots\dots\dots (2.5)$$

Where n is the number of moths.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 8, August 2017

It should be observed here that moths and flames are both solutions. The difference between them is the way we treat and update them in each iteration.

After the mathematical model of this behavior, position update of each moth with respect to a flame is done by using the following equation:

$$M_i = S(M_i, F_j) \quad \dots\dots\dots (2.6)$$

Where M_i point out the i-th moth, F_j point out the j-th flame, and S is the spiral function.

Logarithmic spiral is taken as the main update method of moths. The logarithmic spiral for MFO algorithm is expressed as follows:

$$S(M_i, F_j) = D_i \cdot e^{bt} \cdot \cos(2\pi t) + F_j \quad \dots\dots\dots (2.7)$$

Where D_i show the distance of the ith moth for the jth flame, b is a constant which defines the shape of the logarithmic spiral, and t is a random number whose range is [-1, 1]. D is formulated as follows:

$$D_i = |F_j - M_i| \quad \dots\dots\dots (2.8)$$

Where M_i point out the i-th moth, F_j point out the jth flame, and D_i show the distance of the ith moth for the j-th flame.

The logarithmic spiral, area around the flame and the position assuming distinct t on the curve are represented in Figure.2 shows a conceptual representation of position updating of a moth around a flame.

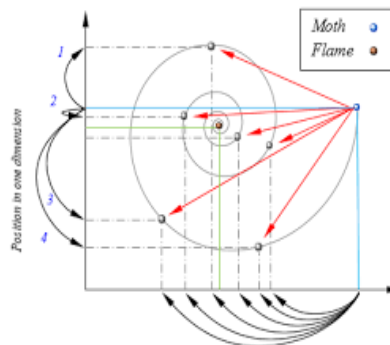


Fig.1: Possible positions reached by a moth with respect to flame.

The equation (2.7) representing the position update phenomenon of moth only around a flame, but it stimulates the MFO algorithm to be trapped in local optima. So to avoid this, each moth is forced to an individual flame for updating the position. After updating the list of flames in each iteration, the flames are sorted in order of their best fit values. The sorting is done by using the quick sort algorithm. Hence the moths will update their positions respectively according to related flames. The initial moth always updates its position with the best flame, on the other hand, the last moth updates its position with worst flame.

By this method, exploration of search scope is guaranteed around the best locations obtained so far. Another concern is that the position update mechanism of moths with respect to a different location may lead to degradation of exploitation of best solutions. To neglect this problem an adaptive mechanism is introduced. The mechanism represents how the number of flames is adaptively decreased over the course of iterations. According to this, following formula is computed:

$$\text{flame no} = \text{round} \left(N - 1 * \frac{N - 1}{T} \right) \quad (2.9)$$

Where l is the current number of iteration, N is the maximum number of flames, and T represents the maximum number of iterations. The gradual decrement in number of flames balances the exploitation and exploration of search scope.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 8, August 2017

III. TEST PROBLEMS

The performance of Moth Flame Algorithm is tested on six benchmark functions[13]. These test functions are divided into three types:-unimodal,multimodal and composite benchmark functions. The uni-modal functions are fit for benchmarking the exploitation of algorithm since they do not have local optima and have global optimum. On the other hand, multi-modal functions are helpful to observe exploration and local optima avoidance of algorithms. Eventually, composite functions are merger of shifted rotated and biased multi-modal test functions.

Uni-modal benchmark functions.

Test function 1: Sphere Function: The test function is represented as following:

$$f_1(x) = \sum_{i=1}^n x_i^2$$

The variable bounds and dimension for this function are taken as [-100,100] and 10 respectively.

Test function 2: Quadratic Function: The test function is represented as following:

$$f_2(x) = \sum_{i=1}^n ix_i^4 + rand(0,1)$$

The variable bounds and dimension for this function are taken as [-1.28, 1.28] and 10 respectively.

Multi-modal benchmark functions.

Test function 3: Schwefel Function: The test function is represented as following:

$$f_3(x) = \sum_{i=1}^n -x_i \sin(\sqrt{|x_i|})$$

The variable bounds and dimension for this function are taken as [-500,500] and 10 respectively.

Test function 4: Rastrigin Function: The test function is represented as following:

$$f_4(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$$

The variable bounds and dimension for this function are taken as [-5.12, 5.12] and 10 respectively.

Composite benchmark functions.

Test function 5: The test function is represented as following (Shekel's Foxholes Function):

$$f_5 = \left[\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6} \right]^{-1}$$

The variable bounds and dimension for this function are taken as [-65.536, 65.536] and 2 respectively.

Test function 6: The test function is represented as following (Kowalik's Function):

$$f_6 = \sum_{i=1}^{11} \left[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right]^2$$

The variable bounds and dimension for this function are taken as [-5, 5] and 4 respectively.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 8, August 2017

IV. RESULTS AND DISCUSSION

The simulation of MFO is carried out on MATLAB (R2013a) environment. In order to demonstrate the capability of MFO algorithm, some well-known algorithms have been chosen for comparison: PSO [14], FPA [15], GA [16], and FA [8]. It's noteworthy that we assume 1000 iterations and 30 number of search agents for each of the algorithms. The aim is not to represent the MFO is better or worse the other algorithm, but it is just to check the capability of algorithm. As according to NFL (No Free Launch) theorem, there is no algorithm to solve the optimization problem. The average values of results computed using six benchmark functions are represented in Table 1. The figure delineates the test function and convergence curve. Hence, the uni-modal functions are shown by Figure.2 and 3 respectively. The figure.4 and 5 represents the multi-modal functions. The composite functions are shown by Figure.6 and 7 respectively.

The results obtained after computations are represented by bold face. Results show that the algorithm is effective and capable to handle the optimization problem effectively, shows the better convergence and also able to deal with the constraints consisting problems.

Table.1: Results of benchmark functions.

Functions		MFO	PSO	FPA	FA	GA
Unimodal	F1	0.00002492	1.321152	203.6389	7480.746	21886.03
	F2	0.0026147	7.715564	11.1687	39.32533	56.51757
Multimodal	F3	-2941.6643	-3571	-8086.74	-3662.05	-6331.19
	F4	55.7175	124.2973	92.69172	214.8951	236.8264
Composite	F5	7.2592e⁻¹⁶	137.7789	10.09454	175.9715	92.13909
	F6	27.827	166.6643	11.41158	353.6269	96.70927

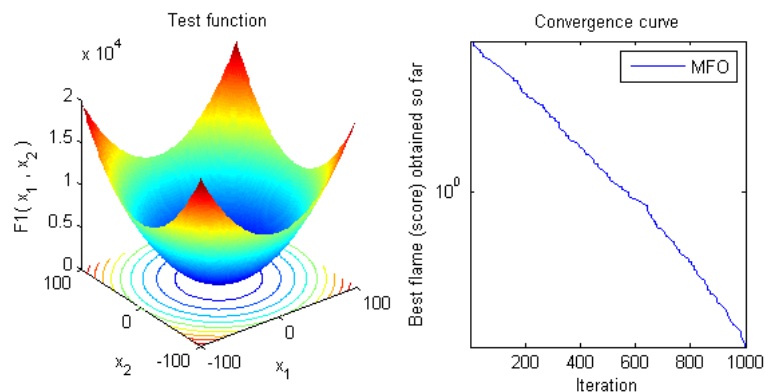


Fig.2. For Sphere Function

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 8, August 2017

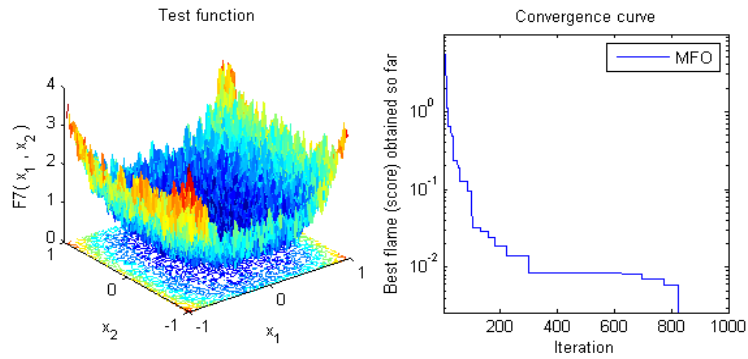


Fig.3. For Quadratic Function

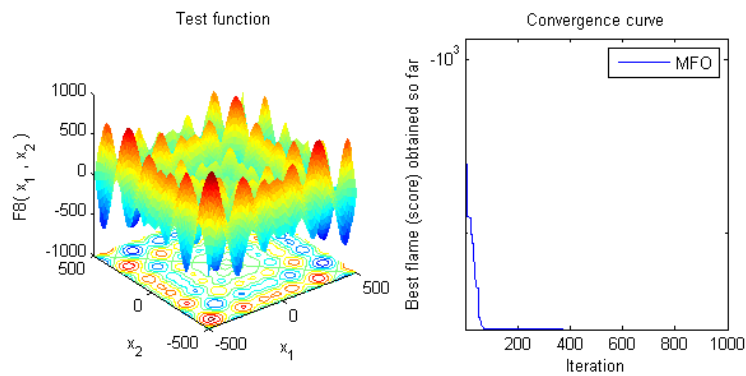


Fig.4. For Schwefel Function.

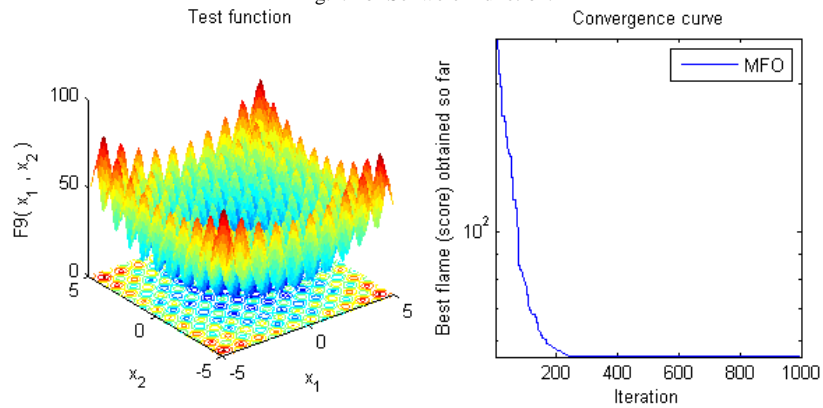


Fig.5. For Rastrigin Function.

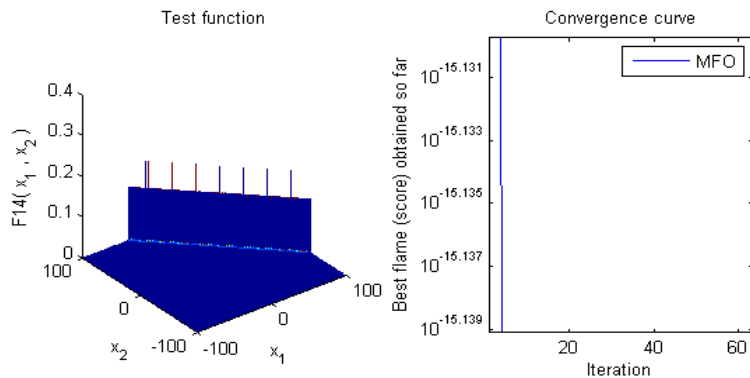


Fig.6. For Shekel's Foxholes Function

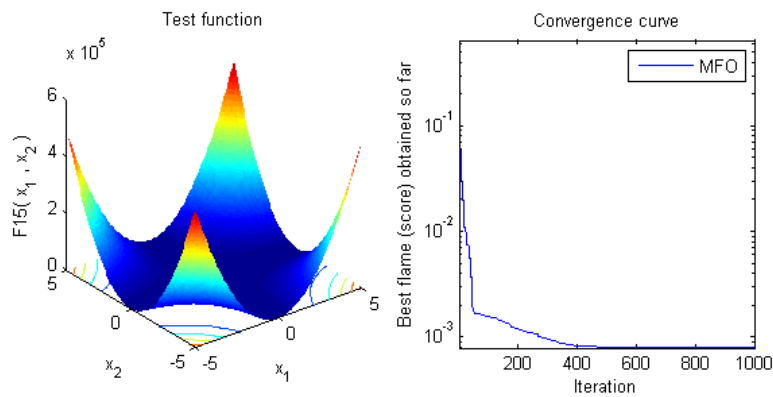


Fig.7. For Kowalik's Function

V. CONCLUSION

In this manuscript the nature-inspired Moth Flame Algorithm evaluates its performance on six benchmark functions which are unimodal, multimodal and composite type. Performance analysis on six benchmarks test functions reveals that algorithm's accuracy and superiority of the approach over PSO, FPA, GA and FA. The theoretical analysis is supported by the additional empirical evidence in which the range of initial values of x , dimension and f_{\min} was changed. The algorithm performs better exploration and exploitation. The above analysis also analyses that MFA is very simple yet effective. Thus the approach is a good alternative in optimization domain where the optima and accuracy are the crucial issue in solving the real life problems. Future work on MFA includes comparison with other algorithms using optimization problem either dual-objective or single objective such as Load Dispatch in Electrical field and many more.

REFERENCES

- [1] M. Seyedali, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm." Knowledge-Based Systems, 69 (2015): 46-61.
- [2] A.J Wood and B.F. Wollenberg, "Power Generation, Operation, and Control", John Wiley and Sons, New York, 1984.
- [3] Z. X. Liang and J. D. Glover, "A zoom feature for a dynamic programming solution to economic dispatch including transmission losses", IEEE Trans. on Power Systems, Vol. 7, No. 2, pp. 544-550, May 1992
- [4] R.C. Eberhart, J. Kennedy, "A new optimizer using particle swarm theory," in Proc of the 6th International Symposium on MMHS, 1995, pp. 39-43.
- [5] H. P Singh, Y. S Brar, & D. P. Kothari, "Combined active and reactive power dispatch using particle swarm optimization." In Proc. of Informing Science & IT Education Conference (InSITE) 2014 (295-304).
- [6] H. P Singh, Y. S. Brar, & D. P Kothari, "Multiobjective load dispatch using particle swarm optimization." IEEE 2013.
- [7] J.H. Holland, "Genetic algorithms," Sci. Am. 267 (1992) 66-72.
- [8] X.S. Yang, "Firefly algorithm," in: Engineering Optimization, 2010, pp. 221-230.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 8, August 2017

- [9] A.Y Abdelaziz. , E.S Ali., S.M Abd Elazim, "Combined economic and emission dispatch solution using Flower Pollination Algorithm." *Electrical Power and Energy Systems* 80 (2016) 264–274.
- [10] CT Su, CT Lin, "New approach with a Hopfield modeling framework to economic dispatch." *IEEE Trans Power Syst* 2000; 15(2):541–5.
- [11] YS.Brar, JS. Dhillon, DP Kothari, "Multiobjective load dispatch by fuzzy logic searching weightage pattern." *Elect Power Syst Res* 2002; 63:149–60.
- [12] D Mishra, A Shukla, PK Kalra, "OR-Neuron based Hopfield neural network for solving economic load dispatch problem." *Neural Inform. Process. Lett. Rev.*2006;10(11):249–59.
- [13] X. Yao, Y. Liu, G. Lin, "Evolutionary programming made faster," *IEEE Trans. Evol. Comput.* 3 (1999) 82-102.
- [14] J. Kennedy, R. Eberhart, "Particle swarm optimization," in: *Proceedings of the 1323 IEEE International Conference on Neural Networks*, 1995, 1995, pp. 1942– 13241948.
- [15] X.-S. Yang, "Flower pollination algorithm for global optimization," in: *1294 Unconventional Computation and Natural Computation*, Springer, 2012, pp. 1295, 240–249.
- [16] H. J Holland, "Adaptation in Natural and Artificial Systems," MIT Press, 1326 Cambridge, MA, 1992.