Performance Research on Firefly Optimization Algorithm with Mutation

Sankalap Arora[1] and Satvir Singh[2]
[1]*Department of Computer Science & Engineering,*
SUS College of Engineering and Technology, Mohali, India
[2]*Department of Electronics & Communication Engineering,*
Shaheed Bhagat Singh State Technical Campus, Ferozpur, India
E-mail: [1]sankalap.arora@gmail.com, [2]drsatvir.in@gmail.com

***Abstract*—Firefly algorithm is an optimization algorithm which mimics the behavior of fireflies to solve problems. In this paper, firefly algorithm with mutation is researched and the performance effect of parameter settings is studied in order to show which setting is more suitable for solving optimization problems. It is tested on ten standard function problems and compared with original firefly algorithm. Experiment results show that firefly with mutation is effective for solving most of the benchmark functions. And the firefly algorithm with mutation has superior performance to the compared method on all ten standard benchmark functions.**

Keywords—Optimization, Firefly, Mutation, Algorithm, Performance

I. INTRODUCTION

An optimization problem is the problem of finding the best solution from all feasible solutions. Classical methods of optimization are generally not used for their impracticality in complicated real life situation. They are generally deterministic in nature. Nature-inspired metaheuristic algorithms such as Particle Swarm Optimization (PSO), Genetic Algorithm (GA), Ant Colony Optimization (ACO) and Firefly Algorithm (FA) are most powerful algorithms for optimization [1]. The goal is to develop more proficient and better optimization techniques that might involve more and more sophistication of algorithm. Nature has remained a great source of inspiration to mankind to develop novel methods of optimization techniques. Bio mimicking of several natural events have given birth to modern day metaheuristic algorithm. The main essence of metaheuristic algorithm is to exploit the method of trial and error. Meta-heuristics have been remarkably successful because of four main reasons: simplicity, flexibility, derivation free mechanism, and local optima avoidance [2].

First, meta-heuristics are fairly simple as they are inspired by very simple concepts. The inspirations are typically related to physical phenomena, animals' behaviors, or evolutionary concepts [3]. The simplicity allows researchers to simulate different natural concepts, propose new meta-heuristics, hybridize two or more meta-heuristics, or improve the current meta-heuristics. Moreover, the simplicity assists researchers to learn metaheuristic quickly and apply them to their problems [4], [5].

Second, flexibility refers to the ease of applicability of metaheuristics to different problems without any major changes in the algorithm. Meta-heuristics are readily applicable to different problems since they mostly assume problems as black boxes. In other words, only the input(s) and output(s) of a system are important for a meta-heuristic which change according to the problem.

Third, in contrast to gradient-based optimization approaches, meta-heuristics optimize problems stochastically [6]. The optimization process starts with random solution(s) and there is no need to calculate the derivative of search spaces to find the optimum. This makes meta-heuristics highly suitable for real world problems [7], [8].

Lastly, meta-heuristics have capability to avoid local optima because of the stochastic nature of meta-heuristics which allow them to avoid stagnation in local solutions and search the entire search space extensively. The search space of real problems is usually unknown and very complex with a massive number of local optima, so meta-heuristics are good options for optimizing these challenging real world problems [9].

The strength of standard firefly algorithm lies in the attractiveness of less brighter firefly towards the brighter firefly [10]. The less brighter firefly improvises its position according to brighter firefly but it does not add good features or attributes from the better firefly. So if the less brighter firefly can add features or attributes from the better firefly, it can converge to optima quickly in less number of iterations. [11]

This paper aims to research on Firefly Algorithm with mutation (MFA) and provide comparison study of the MFA with FA. We will first outline the Firefly Algorithm, then formulate the firefly algorithm with mutation and then demonstrate the comparison of these algorithms focusing on critical factors like convergence and time consumption. The MFA optimization seems more promising in the sense that MFA converges quickly than firefly algorithm optimization. [12]

II. FIREFLY ALGORITHM

A. *Standard Firefly algorithm*

1) *Behavior of fireflie*

There are around two thousand species of firefly algorithm, usually found in tropical and temperate regions. Most species of fireflies produce unique, short and rhythmic flashes. Bioluminescence is the process responsible for flashing of light. These flashes are used to attract mating partners and potential prey. These rhythmic flashes are different from each other on the basis of the rate of flashing and amount of time. Females respond to unique pattern of flashing of a male which forms a signal system bringing both sexes together. [13].

When a light source emits light intensity from a particular distance r it obeys inverse square law. The light intensity I decreases with increase in the distance r in terms of $I \propto 1/r^2$ [14]. Air acts as an absorbent and light becomes weaker as the distance increase [15]. The flashing light is formulated in such a way that it can associated with objective function to be optimized, which opens gateway in formulation of new optimization algorithms [10].

2) Firefly Algorithm

To develop firefly inspired algorithm, it is mandatory to idealize some of the characteristics of fireflies. For simplicity in describing Firefly algorithm, three assumptions have been made [16]:

1. All fireflies are of same sex which means every firefly will be attracted to other fireflies regardless of their sex.
2. Attractiveness is proportional to their brightness, thus for any two flashing fireflies, the less brighter one will movetowards the brighter one.
3. The brightness of a firefly is affected or determined by the landscape of the objective function. Based on these three rules, the basic steps of the firefly algorithm (FA) can be summarized as the pseudocode shown in Algorithm 1 [17].

Algorithm 1 Firefly algorithm

Objective Function f (X), X= $(x_1, x_2 \dots \dots . x_d)$
Generate the initial population of n fireflies, X_i,
i = 1, 2,…, n
Light intensity I_i at X_i is determined by f (X_i)
Define the light absorption coefficient γ
while (t < MaxGeneration)
    for$=1{:}\,n$, all n fireflies
        for j= $1{:}\,n$,all n fireflies (inner loop)
        if $(I_i < I_j)$, Move firefly i towards j;
        end if
    Vary attractiveness with distance i via $\exp[-\gamma r^2]$
        end for j
    end for i
Rank the fireflies and find the current global best solution g*
end while
Post-process the results

3) Attractiveness and Distance

Firefly algorithm is based on two important factors: variation of light intensity and formulation of the attractiveness. An assumption is made that attractiveness of a firefly is calculated according its brightness which is associated further with the encoded objective function [18]. In maximum optimization problems, the brightness of a firefly can be chosen as $I(x) \propto f(x)$where I is the intensity of a firefly and x is a particular location. Attractiveness β is relative and it will change according to distance r_{ij} between firefly i^{th} and fireflyj^{th}. Light is also absorbed by the air and it also get decreased with increasing distance so attractiveness is allowed to vary with degree of absorption. Light intensity $I(r)$ varies according to inverse square law and for a given medium with fixed light absorbtion coefficientγ the light intensity I varies with distance r [19].So attractiveness β of firefly is defined by

$$\beta = \beta_o \exp(-\gamma r^2) \tag{1}$$

where β_o is the attractiveness at distance $r = 0, \beta$ is the fixed light absorption coefficient for a specific medium and γ is light absorption coefficient. The distance r_{ij} between any two fireflies i^{th} and j^{th} located at X_i and X_j, respectively, is determined using the Euclidean norm and movement of a less brighter fireflyi^{th} towards brighter firefly j^{th} is determinedby

$$x_i = x_i + \beta_0^{e\left(-\gamma r_{ij}^2\right)}\left(x_j - x_i\right) + \alpha\left(rand - \tfrac{1}{2}\right) \tag{2}$$

In (2) the second term is due to relative attraction and third term is a randomization parameter. α is randomization parameter normally selected within range [0,1] and $rand$ is a random number uniformly distributed in [0, 1]. Now to introduce the variation of attractiveness, γ parameter is used and its range is 0.01 to 10. The initial locations of n fireflies are distributed uniformly in the search space whenever the number of fireflies are greater than number of local optima [20]. During the execution, the fireflies converge into all of these local optima, the global optima is determined. The algorithm will approach the global optima when $n \to \infty$ and number of iterations are greater than 1 but in reality it converge quickly [21].

In (2) the second term is due to relative attraction and third term is a randomization parameter. $\alpha$is randomization parameter normally selected within range[0,1] and ϵ_i is a vector of random numbers drawn either a Gaussian or uniform distribution. Now to introduce the variation of attractiveness, γ parameter is used and its range is 0.1 to 10. In optimization problem where number of fireflies are greater than number of local optima, the initial locations of the *n* fireflies should be distributed relatively uniformly throughout the entire search space. During the execution, the fireflies converge into all of these local optima, the global optima is determined. FA will approach the global optima when $n \to \infty$ and number of iterations are greater than 1 but in reality it converge extremely quickly.

B. *Firefly Algorithm with Mutation*

The strength of any optimization algorithm lies in how faster the algorithm explores the new possible solutions and how efficiently it exploit the solutions to make them better. FA algorithm performs a move step which contains the exploration and exploitation concept. There is a need by which exploration and exploitation can be enhanced and the algorithm can work more efficiently. So mutation is added to firefly algorithm to achieve better results. By using mutation the basic concept of searching solutions is modified. In standard firefly algorithm, space is searched by moving the less brighter firefly moves towards the more brighter firefly. Firefly algorithm with mutation searches the search space by adding features to less brighter firefly from more brighter firefly. The extent of features to be added is decided by calculating the mutation probability of each firefly. The better the firefly, lesser the mutation probability and viceversa. By using features of better fireflies, the algorithm will converge faster and avoid falling into the local optimum.

In the firefly algorithm with mutation all fireflies do not participate in mutation, but some. The underlying principle of MFA is to adapt features from other fireflies and achieve best values in minimum amount of time. The mutation concept is also modified for FA. To better explain it lets suppose there are 100 fireflies, only top good 40 percent individuals will donate their features because they are good because of their good features. Similarly, the need of the good features from top 40 per cent will be needed by last 40 per cent of fireflies i.e. worst 40 percent of solutions. The better the firefly, more the mutation probability and worse the firefly, less the mutation probability. Additional to it, the in between 20 percent individuals which are average i.e. neither good nor bad, do not participate in the mutation process and they have very low mutation probability. The basic principle which is followed in the firefly algorithm with mutation is that there is better probability of good solutions becoming better and there is low probability of bad solutions becoming very good. So to make bad solutions, better solutions, mutation can be applied. As mutation gave them chance to modify themselves and attain good features.

The mutation operator is used to change some elements in selected individuals with a probability p_m (mutation probability) leading to additional firefly diversity to help the search process escape from local optimal traps. Each firefly has its mutation probability$p_m$necessary for it to mutation. The choice of p_m will critically affect the behavior and performance. Typical values of p_m are same as in GA i.e. 0.001 to 0.05. The mutation probability (MP) in firefly algorithm with mutation is calculated using (3)

$$MP = f_{new} - f_{old} \tag{3}$$

where f_{new} is the fitness of the new firefly and f_{new} the fitness of the original firefly. For a generation that undergoes n_m mutation operations, the average mutation progress value $\acute{M}P$ is given by (4)

$$\acute{M}P = \frac{1}{n_m} \sum MP \tag{4}$$

Before the end of each generation, mutation rates are adjusted using these average progress values. Based on these equations (3) and (4), the steps of the firefly algorithm with mutation (MFA) can be summarized as the pseudocode shown in Algorithm 2.

Algorithm 1 Firefly algorithm with Mutation

```
Objective Function f (X), X= (x_1, x_2 … … . x_d)
Generate the initial population of n fireflies, X_i,
i = 1, 2,…, n
Light intensity I_i at X_i is determined by f (X_i)
Define the light absorption coefficient γ
while (t < MaxGeneration)
        for= 1:n, all n fireflies
        for j= 1:n,all n fireflies (inner loop)
        if (I_i < I_j), Move firefly i towards j;
        Calculate Mutation probability
        Perform Mutation
        end if
        Vary attractiveness with distance i via
    exp[− γr^2]
        end for j
        end for i
    Rank the fireflies and find the current global best
    solution g*
    end while
    Post-process the results
```

III. Simulation & Experiments

In this paper, ten standard benchmark functions are used for testing the success of firefly algorithm with mutation against standard firefly algorithm, which are described in Table I.

There are many ways to carry out the comparison of algorithm performance and two the obvious approaches are: to compare the numbers of function evaluations for a given tolerance or accuracy or to compare their accuracies for a fixed number of function evaluations. Here the second approach is used. In simulations, maximum number of evaluations is fixed to 1000, so that meaningful statistical analysis can be done.

For both the algorithms, same standard of learning parameters i.e. $\alpha = 0.1$ and $\gamma = 0.01$ are used. For better comparison between the two algorithms Mean, Median, Best values and Worst values for different n are also being considered. Different number of fireflies(n) are used having values 50, 100, 250 and 500. Number of dimensions was as per the standard benchmark function, as discussed in Table I.

TABLE I VARIOUS STANDARD BENCHMARK FUNCTIONS

Benchmark Function	Formula	Dimension (n)	Range	Optimal Value
Ackley	$f(\vec{x}) = -20.exp\left(^{-0.2}\sqrt{\frac{i}{n}\sum_{i=1}^{n} x_i^2}\right) - exp\left(\frac{1}{n}\sum_{i=1}^{n} cos(2\pi.x_i)\right) + 20 + e$	30	(-32,32)	0
Sphere	$f_0(\vec{x}) = \sum_{i=1}^{n} x_i^2$	30	(-100,100)	0
Griewank	$f(\vec{x}) = \frac{1}{4000}\sum_{i=1}^{n-1}(x_i - 100)^2 - \prod_{i=1}^{n-1}\cos\left(\frac{x_i - 100}{\sqrt{i-1}}\right) + 1$	10	(-600, +600)	0
Michalewiz	$f(x) = -\sum_{i=0}^{n}(\sin(x_i) sin^{20}\left(\frac{ix_i^2}{\pi}\right)$	10	(0,π)	-0.966n
Rastrigin	$f(\vec{x}) = \sum_{i=1}^{n}(x_i^2 - 10\cos(2\pi x_i) + 10)$	30	(-5.12, 5.12)	0
Schaffer	$f(x) = (x_0^2 + x_1^2)^{\frac{1}{4}}(50(x_0^2 + x_1^2)^{0.1} + 1)$	2	(-100,100)	0
Schewel	$\sum_{i=0}^{n-1}\lvert x_i\rvert + \prod_{i-0}^{n-1}\lvert x_i\rvert$	30	(-10,10)	0

These algorithms are implemented in QT Creator. Table II shows best solution, mean solution, median solution, worst solution and time taken to complete defined number of iterations. Table II reveals the striking potential of the MFA in obtaining the high precision optimal solutions better than the standard FA solutions.

TABLE II VARIOUS STANDARD BENCHMARK FUNCTIONS

Functions	No. of Particles	Comparison of Simulation Results								Time Taken	
		Mean		Median		Best Results		Worst Results			
		FA	MFA	FA	MFA	FA	MFA	FA	MFA	FA	MFA
Ackley	50	9.89E-04	9.11E-04	9.85E-04	8.83E-04	9.01E-04	7.93E-04	1.13E-03	1.05E-03	4	1.3
	100	8.86E-04	7.66E-04	8.80E-04	7.60E-04	8.08E-04	6.75E-04	9.66E-04	8.27E-04	9	4.7
	250	8.19E-04	7.05E-04	8.54E-04	7.16E-04	7.47E-04	5.75E-04	9.03E-04	7.61E-04	37	25.3
	500	7.42E-04	6.63E-04	7.26E-04	6.48E-04	6.97E-04	6.28E-04	8.07E-04	6.99E-04	125	98
Sphere	50	1.91E-05	1.45E-05	1.90E-05	1.40E-05	1.60E-05	1.20E-05	2.20E-05	1.70E-05	1	1
	100	1.43E-05	1.21E-05	1.40E-05	1.20E-05	1.10E-05	1.00E-05	1.70E-05	1.60E-05	4	3.9
	250	1.19E-05	9.30E-06	1.20E-05	1.00E-05	1.00E-05	7.00E-05	1.40E-05	1.10E-05	29	24
	500	9.60E-06	8.90E-05	1.00E-05	9.00E-05	7.00E-06	7.00E-06	1.10E-05	1.10E-05	120	98
Griewank	50	1.44E-01	1.24E-01	9.35E-02	1.79E-01	3.60E-02	2.04E-01	4.06E-01	1.03E-01	1	0.55
	100	1.49E-01	1.51E-01	1.03E-01	7.63E-02	4.19E-02	2.36E-01	4.36E-01	1.75E-01	3	2.03
	250	1.53E-01	1.54E-01	1.40E-01	6.40E-02	4.67E-02	1.99E-01	3.91E-01	1.48E-01	12	9.6
	500	1.74E-01	1.44E-01	1.50E-01	2.12E-01	8.37E-02	1.94E-01	3.62E-01	1.23E-01	58	44.7
Michalewiz	50	-7.34E+00	-8.88724	-7.31E+00	-9.2088	-8.65E+00	-9.49491	-4.28E+00	-8.16183	2	0.62
	100	-7.86E+00	-8.89E+00	-8.13E+00	-9.21E+00	-8.70E+00	-9.49E+00	-6.16E+00	-8.16E+00	5	2.04
	250	-7.78E+00	-8.30E+00	-7.94E+00	-8.45E+00	-9.03E+00	-9.29E+00	-6.67E+00	-6.63E+00	15	11.8
	500	-7.56E+00	-8.88E+00	-7.64E+00	-9.40E+00	-8.40E+00	-9.02E+00	-7.11E+00	-9.26E+00	58	45.8
Rastrigin	50	4.71E+01	4.26E+01	4.58E+01	4.38E+01	2.79E+01	2.48E+01	6.17E+01	6.07E+01	2	1.2
	100	4.97E+01	4.38E+01	3.98E+01	3.98E+01	2.98E+01	2.49E+01	8.76E+01	6.57E+01	7	4.3
	250	4.32E+01	4.09E+01	4.08E+01	3.28E+01	3.18E+01	2.49E+01	6.57E+01	7.96E+01	32	25.4

Table II (Contd.)…

…Table II (Various Standard Benchmark Functions)

Functions	No. of Particles	Comparison of Simulation Results								Time Taken	
		Mean		Median		Best Results		Worst Results			
		FA	MFA	FA	MFA	FA	MFA	FA	MFA	FA	MFA
	500	4.01E+01	4.05E+01	3.68E+01	3.38E+01	2.89E+01	2.29E+01	5.27E+01	6.37E+01	122	99
Schaffer	50	2.76E-02	3.03E-02	3.32E-02	2.93E-02	1.76E-02	2.68E-02	4.00E-02	3.23E-02	0.7	0.2
	100	1.90E-02	2.60E-02	1.74E-02	2.53E-02	6.50E-03	1.95E-02	3.32E-02	3.90E-02	1.5	0.8
	250	1.61E-02	1.59E-02	1.54E-02	1.38E-02	4.66E-03	8.43E-03	2.47E-02	2.84E-02	7.5	5.1
	500	1.65E-02	1.68E-02	1.52E-02	1.36E-02	8.51E-03	7.09E-03	2.53E-02	2.92E-02	22	21.1
Schewel	50	1.91E-03	1.73E-03	1.85E-03	1.70E-03	1.79E-03	1.44E-03	2.09E-03	2.00E-03	4	1
	100	1.79E-03	1.50E-03	1.76E-03	1.51E-03	1.41E-03	1.36E-03	2.13E-03	1.61E-03	9	3.9
	250	1.53E-03	1.29E-03	1.54E-03	1.34E-03	1.36E-03	1.02E-03	1.74E-03	1.47E-03	36	24.4
	500	1.44E-03	1.28E-03	1.41E-03	1.27E-03	1.34E-03	1.19E-03	1.51E-03	1.36E-03	129	103

IV. Conclusion and Future Scope

In the paper, firefly algorithm with mutation(MFA) is researched which considers mutation probability and then perform mutation on fireflies to better explore search space. Simulation results demonstrated the potential of MFA. Simulation results suggests that the proposed algorithm is superior to standard firefly algorithm in terms of both efficiency and success rate. The standard firefly algorithm is efficient but solutions still change as the optima are approaching. So the solution quality is improved by reducing randomness by introducing mutation probability concept. Further, convergence is also improved. The reason for these better results lies in the modified mutation concept which gave bad fireflies/solutions more chance to adapt from good fireflies and eventually become better in less amount of time. The amount of time consumption is less because of the fact that all solutions do not participate in mutation but only those who can donate good features and those who need good features which is calculated by mutation probability for each individual. Considering more iterations information of the algorithm and its application in combination with other algorithms could be an exciting direction in the future.

References

[1] E. Bonabeau, M. Dorigo, and G. Theraulaz, Swarm intelligence. Oxford, 1999.

[2] D.E. Goldberg and J.H. Holland, "Genetic algorithms and machinelearning," Machine learning, Vol. 3, No. 2, pp. 95–99, 1988.

[3] V. Gazi and K.M. Passino, "Stability analysis of social foragingswarms," Systems, Man, and Cybernetics, Part B: Cybernetics, IEEETransactions on, Vol. 34, No. 1, pp. 539–557, 2004.

[4] J. Kennedy and R. Eberhart, "Particle swarm optimization," 1995.

[5] K.M. Passino, "Biomimicry of bacterial foraging for distributed optimizationand control," Control Systems, IEEE, Vol. 22, No. 3, pp. 52–67, 2002.

[6] T. B¨ack, D.B. Fogel, and Z. Michalewicz, Evolutionary computation 1: Basic algorithms and operators. CRC Press, 2000, Vol. 1.

[7] R.C. Eberhart, Y. Shi, and J. Kennedy, Swarm intelligence. Elsevier, 2001.

[8] J. Liang, P. Suganthan, and K. Deb, "Novel composition test functionsfor numerical global optimization," in Swarm Intelligence Symposium, 2005. SIS 2005. Proceedings 2005 IEEE. IEEE, 2005, pp. 68–75.

[9] X.S. Yang, Nature-inspired metaheuristic algorithms. Luniver press, 2010.

[10] S. Łukasik and S. Zak, "Firefly algorithm for continuous constrainedoptimization tasks," in Computational Collective Intelligence. Semantic Web, Social Networks and Multiagent Systems. Springer, 2009, pp. 97–106.

[11] X.S. Yang, Engineering optimization: an introduction with metaheuristicapplications. John Wiley & Sons, 2010.

[12] S. Arora and S. Singh, "The firefly optimization algorithm: Convergenceanalysis and parameter selection," International Journal of ComputerApplications, Vol. 69, No. 3, pp. 48–52, 2013.

[13] X.S. Yang, "Firefly algorithm, levy flights and global optimization,"in Research and Development in Intelligent Systems XXVI. Springer, 2010, pp. 209–218.

[14] S. Arora and S. Singh, "A conceptual comparison of firefly algorithm, batalgorithm and cuckoo search," in Control Computing Communication & Materials (ICCCCM), 2013 International Conference on. IEEE, 2013, pp. 1–4.

[15] X.S. Yang, "Chaos-enhanced firefly algorithm with automatic parametertuning," International Journal of Swarm Intelligence Research (IJSIR), Vol. 2, No. 4, pp. 1–11, 2011.

[16] K. Deb, Optimization for engineering design: Algorithms and examples. PHI Learning Pvt. Ltd., 2012.

[17] X.S. Yang, "Firefly algorithm, stochastic test functions and designoptimisation," International Journal of Bio-Inspired Computation, Vol. 2, No. 2, pp. 78–84, 2010.

[18] T. Apostolopoulos and A. Vlachos, "Application of the firefly algorithmfor solving the economic emissions load dispatch problem," International Journal of Combinatorics, Vol. 2011, 2010.

[19] X.S. Yang, "Firefly algorithms for multimodal optimization," in Stochastic algorithms: foundations and applications. Springer, 2009, pp. 169–178.

[20] Z.W. Geem, J.H. Kim, and G. Loganathan, "A new heuristic optimizationalgorithm: harmony search," Simulation, Vol. 76, No. 2, pp. 60–68, 2001.

[21] X.S. Yang, "Biology-derived algorithms in engineering optimization," arXiv preprint arXiv: 1003. 1888, 2010.