INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 4.295
(Volume 4, Issue 6)
Available online at: www.ijariit.com

Parameter tuning in firefly algorithm

Kavitha Rathore
kavirathore1@gmail.com
Amity University, Kant, Rajasthan

ABSTRACT

Optimization means to find the best solution for any situation under given constraints. In today's era, the problems are huge and complex. Nature always finds a way to deal with such problems efficiently in an optimized way. The computational algorithms which are inspired by nature to find solutions for such problems are called Nature inspired optimization algorithms. There are various nature-inspired algorithms and Firefly Algorithm (FA) is one among them. FA is a bio-inspired population-based stochastic algorithm which imitates the behavior of fireflies shown when they attract other fireflies. FA is an algorithm with many parameters that affect the accuracy and the convergence speed. A number of variants and parameter tuning related papers are available in the literature. In this paper first an introduction of Optimization, specifically Nature inspired optimization has been provided. Then, a detailed discussion about FA has been given. It is followed by a brief literature survey in which the work has been compared in tabular form to provide the readers with a better understanding. Further, we intend to improve the accuracy of Firefly algorithm by tuning the parameters namely α and β_{min}. A range of values of the above parameters is tested by forming their combinations to find out the mutual effect of both these parameters. These values are tested on a test bed of nine benchmark functions. The result is a combination of optimized values of both the parameters. The results are quite clear and provide a pair of optimized values of both the parameters.

Keywords— *Meta-heuristic algorithm, Firefly algorithms, Parameter tuning, Nature-inspired algorithms*

1. INTRODUCTION

1.1 Optimization

Optimization is to search for the best possible option from the available set of options under certain constraints and assumptions. It consists of mainly three components: Objective function, variables, and constraints. The objective function is said to be the function of mathematical optimization problem which can either be maximized or minimized over a feasible set of solutions. The function can be a single objective function or multiple objectives. The single objective function is one in which the target is to achieve one objective by setting the variable(s) [1]. The multi-objective function has more than one objective to be achieved. It can be redeveloped into a single objective function by either the different objectives forming a weighted combination or by considering few objectives as constraints. Variables are such values which vary over the given range and accordingly vary the objective function value. If variables take continuous value in range then it is known as a continuous optimization problem but it takes from discrete set it is known as a discrete optimization problem. Constraints allow the variables to take certain values but exclude others. The variable values are to be found such that it should satisfy the constraints and also it should either maximize or minimize the objective function. This is known as the optimization problem. Following is an example of a minimization problem:

$$\text{Minimize } x \in < R^n \quad f_a(x), \qquad (a = 1, 2, \ldots, M)$$
$$\text{subject to } h_b(x) = 0, \qquad (b = 1, 2, \ldots, B),$$
$$g_c(x) \leq 0, \qquad (c = 1, 2, \ldots, C),$$

where $f_a(x)$, $h_b(x)$ and $g_c(x)$ are functions of the design vector and $x = (x_1, x_2, \ldots, x_n)^T$
Here the components x_i of x is called design or decision variables and they can either be real continuous, discrete or the mixed of these two.

The functions $f_a(x)$ where $a = 1, 2, \ldots,$ M are called the objective functions or simply cost functions, and in the case of $M = 1$, there is only a single objective. Space spanned by the decision variables is called the design space or search space $< n$, while the space formed by the objective function values is called the solution space or response space. The equalities for h_j and inequalities for g_k are called constraints.

1.2 Optimization approach

Optimization approach can be broadly classified as Traditional and heuristic. In traditional approach, it tries to find the optimum solution for continuous function where sometimes it is trapped in the sub optimum solution whereas in the heuristic approach it tries to find the quickest approximate solution which need not be the best or accurate one. Heuristic algorithms are best to be used when a quick solution is needed rather than an optimum solution. Evolutionary algorithms are algorithms which take advantage of biological evolution like selection, reproduction, mutation, and recombination. In this, the possible solutions for the problem form the population and the objective function will decide the quality of the solution. In this, after the initial population generation, the best individuals are selected as parents to generate next generation through mutation and crossover. Again their fitness is checked and the best one is selected. This repeats until we find the best solution.

1.3 Nature inspired algorithm

Nature finds efficient methods to solve complex problems and so the algorithms inspired by these nature problem-solving techniques or methods are called nature-inspired algorithm. These algorithms try to relate their problem with nature and try to find an optimal solution. We have many such algorithms like Ant colony optimization algorithm, Harmony Search, Firefly algorithm, Artificial Bee Colony algorithm, Cuckoo search etc.

Most of the nature-inspired algorithms have a very similar working procedure which is shown in Figure 2. They have three major operations namely Selection, Crossover, and Mutation. A small modification in any of these operations provides a huge change in the working of the NIA.

Swarm Intelligence algorithm is inspired by the collective behavior of many individuals together with one another and also with its surroundings. So the overall result will depend on the interaction of the individuals with one another and also with the environment. The individuals need to self organize themselves according to the group. A colony of Ants, schools of fish, flock of birds, beehives etc all display follow-up behavior according to which small change done by an individual in group will result different behavior by other members and thus forming a new pattern.
Genetic Algorithms [4], Particle Swarm Optimization [5], Differential Evolution [6], Ant Colony Optimization [7] and the Firefly Algorithm are a few examples of nature-inspired optimization methods.

2. FIREFLY ALGORITHM

2.1 Fundamentals of firefly algorithm

Firefly Algorithm (FA) [8] is based upon the flashing behavior of fireflies. In this algorithm, search agents mimic fireflies which flashes instances of light to attract their mating partner. The attractiveness observed by a firefly depends upon many factors such as intensity of the light emitted, distance from the source firefly and absorption properties of the atmosphere. Following are a few principles fundamental formed for FA

1) Lesser bright firefly will be attracted and will move towards a more attractive firefly.
2) Intensity (I) of a firefly will decrease with distance r and it is inversely proportional i.e. $I \propto 1/r^2$
3) Atmospheric factors also affect the resultant attractiveness of a firefly at a given distance. The property is modeled using an absorption coefficient.
4) The brightness of a firefly depends upon the type of optimization problem. For example, for maximization problems, objective function value can simply be termed as brightness and for minimization problems, it can be the inverse of the same objective function value

2.2 Basic firefly algorithm

The basic pseudo code of firefly algorithm is as follows:

```
Objective function f(x),  x = (x1, ..., xd)^T
Initialization of fireflies xi (i = 1, 2, ..., n)
Light intensity Ii at xi is determined by f (xi)
Light absorption coefficient γ is defined
While (t <MaxGeneration)
for i = 1 : n for all n fireflies
        for j = 1 : n for all n fireflies (inner loop)
                if (Ii< Ij), Move firefly i towards j; end if
                Attractiveness varies with distance r via exp[−γr]
                Evaluating new solutions & updating light intensity
        end for j
end for i
Ranking of fireflies & find the current global best g*
end while
```

Fig. 1: Pseudo code of the firefly algorithm (FA)

The complete process of FA is depicted in Figure 2.

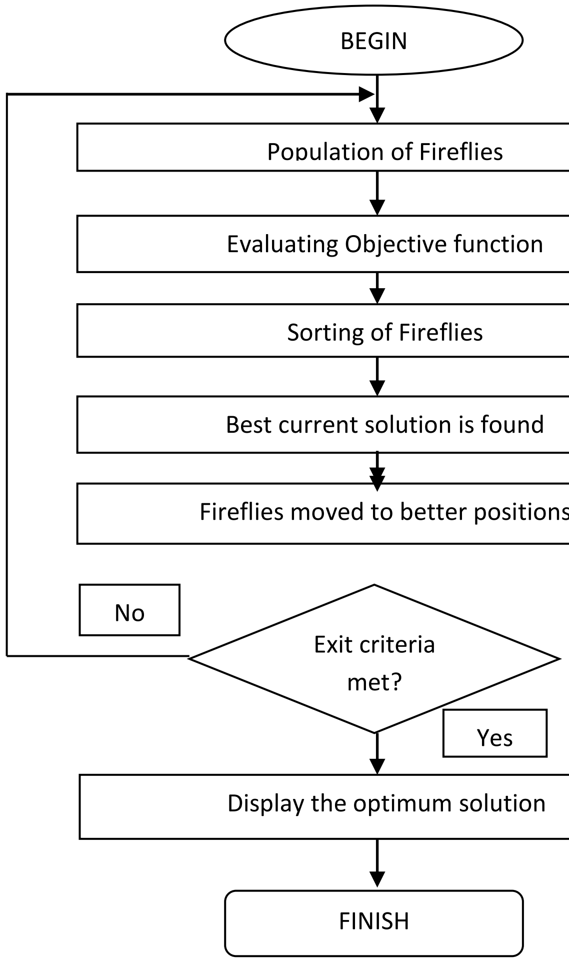

Fig. 2: Flowchart of firefly algorithm

The variation in light intensity follows the inverse square law and varies according to the following rule:

$$I(r) = \frac{I_s}{r^2} \quad (1)$$

Where I_s is the source intensity and $I(r)$ is the resultant intensity at a distance r.
Absorption due to atmospheric factors is modeled by equation 2

$$I = I_0 e^{-\gamma r} \quad (2)$$

Where I_0 is the source intensity, γ is the absorption coefficient and I is the resultant intensity at a distance r.

The attractiveness β is observed by other fireflies and can be defined as

$$\beta = \beta_0 e^{-\gamma r^2} \quad (3)$$

Where β_0 is the attractiveness at the source. The term in equation (3) can further be modified as

$$\beta = \beta_{min} + (\beta_0 - \beta_{min}) e^{-\gamma r^2} \quad (4)$$

The distance r between the two fireflies i and j is computed as

$$r_{ij} = \left\|x_i - x_j\right\| = \sqrt{\sum_{k=1}^{d} (x_{i,k} - x_{j,k})^2} \quad (5)$$

The less bright firefly is moved towards a brighter firefly according to the following expression

$$x_i = x_i + \beta\left(x_j - x_i\right) + \alpha \in_i \quad (6)$$

The attraction is defined by the second term. The third term is randomization with α being the randomization parameter and is a vector of random numbers drawn from a Gaussian distribution or uniform distribution.

3. LITERATURE SURVEY

Following table represents the survey of a few selected variants of the firefly algorithm in compact form:

Table 1: Survey table of variants of Firefly algorithm

S. no	Author	Year	Title	Modification	Applications
1	Xin-She Yang [8]	2010	Firefly Algorithm, Levy Flights, and Global Optimization	The movement is calculated by the attractiveness of light Intensity and randomization through Levy Flights instead of randomization.	Benchmark Functions
2	MO Yuan-bin, MA Yan-zhui, ZHENG Qiao-yan[9]	2013	Optimal Choice of Parameters for Firefly Algorithm	This paper suggests the range in which the parameters should lie. Randomization parameter α should be in [0.1, 0.2] and absorption co-efficient	Benchmark Functions

				γ between [1, 30] and the size of population between 20 to 40 for the smaller problem and not more than 50 for a complicated problem.	
3	Ivona Brajevi, Jelena Ignjatovi [10]	2018	An upgraded firefly algorithm with feasibility-based rules for constrained engineering optimization problems	Upgraded Firefly algorithm (UFA) has the following changes fine adjustments of controlled parameters, constraint handling mechanisms, Boundary constraint handling mechanisms, Usage of mutation operator and Equality constraint handling technique.	Engineering Problems
4	A. Manju, M.J. Nigam [11]	2012	Firefly Algorithm with Fireflies having Quantum Behavior	Based on probability theory, a quantum version of the firefly algorithm is introduced.	Benchmark Functions
5	Changnian Liu, Yafei Tian, Qiang Zhang, Jie Yuan, Binbin Xue [12]	2013	Adaptive Firefly Optimization Algorithm Based On Stochastic Inertia Weight	Adaptive Firefly Optimization algorithm (AFA) is based on stochastic inertia weight which varies as the position of fireflies change and fitness value differs. Convergence speed and performance are better than standard FA.	PID parameters
6	Shruti Goel, V. K. Panchal[13]	2014	Performance Evaluation of a New Modified Firefly Algorithm	New Modified Firefly Algorithm (NMFA) linearly decreases the randomization parameter thus it giving a balance between exploration and exploitation. It performs better than SFA and its both variants.	Benchmark Functions
7	Mukesh Gupta, S.N.Tazi, Akansha Jain, Deepika [14]	2014	Edge Detection Using Modified Firefly Algorithm	This Modified Firefly algorithm detects number of edges than other algorithms and performs better than already present algorithms and is less tender to noise.	Standard Images
8	Shubhendu Kumar Sarangi, Rutuparna Panda [15]	2016	A New Modified Firefly Algorithm for Function Optimization	The Modified Firefly algorithm uses the improved velocity concept of particle swarm optimization so that the searching behavior of Standard algorithm is improved.	Standard benchmark functions through simulations.
9	Irwan Mahmudi, Ratna Ika Putri, Margo Pujiantara, Ardyono Priyadi, Mauridhi Hery Purnomo [16]	2016	Modified Firefly Algorithm for Maximum Power Point Tracking Of a Small Stand Alone Wind Power System	In this algorithm, in each iteration, the coefficient β is updated and α and γ are neglected to get the maximum power from wind turbines using permanent magnet synchronous generator. The efficiency reaches 90.033% using this algorithm.	Wind Speed
10	Xiangbo Qi, Sihan Zhu, Hao Zhang [17]	2017	A Hybrid Firefly Algorithm	In SFA the fireflies are unisex and flashing behavior is simulated but here in this Hybrid algorithm, the mating behavior is also added.	Benchmark Functions
11	Mahdi Bidar, Samira Sadaoui, Malek Mouhoub, Mohsen Bidar [18]	2017	Improving Firefly Algorithm Performance using Fuzzy Logic	In this algorithm, the firefly algorithm parameters are adjusted by the Fuzzy system which acts as a controller based on Fuzzy rules.	High dimensional Benchmark functions
12	Shubhendu Kumar Sarangi, Rutuparna Panda [19]	2017	Crazy Firefly Algorithm for Function Optimization	This paper increases the diversity of the FA by adding a craziness operator.	Benchmark Functions
13	Xiuqin Pan, Limiao Xue, Ruixiang Li [20]	2018	A new and efficient firefly algorithm for numerical optimization problems	In New and efficient FA (NEFA) three changes are made to FA. A new attraction model and a new search operator are added to FA. Also, the step factor is dynamically updated.	Benchmark Functions
14	Huali Xu, Shuhao Yu, Jiajun Chen, Xukun Zuo [21]	2018	An Improved Firefly Algorithm for Feature Selection in classification	Discrete coding is used to convert the continuous version of the firefly algorithm to binary. For initial population opposition based learning is used and for searching opposition strategy is used. This improves the convergence rate to reach global optima.	Datasets

4. PROPOSED WORK

In standard firefly algorithm, the movement towards the brighter firefly is controlled by the following equation

$$x_i = x_i + \beta_0 e^{-\gamma r_{i,j}^2}(x_j - x_i) + \alpha \in_i \qquad (7)$$

Where γ is light absorption co-efficient, β the attractiveness between two fireflies and α the step factor. The objective function which decides the movement of firefly is based on the current position of fira efly x_i , attractiveness between both the fireflies considering the distance between them to be zero β_0 ,the ab sorption coefficient which when increases the attractiveness decreases and vice versa γ, the distance between both the fireflies r, the step factor α and randomization factor. For our proposed analysis, we have used a slightly different variant. In this variant, the new position of a firefly is given by the following equation.

$$x_i = x_i + \beta_{min} + (\beta_0 - \beta_{min})e^{-\gamma r^2} + \alpha * (rand(1,d) - 0.5) * \text{scale} \quad (8)$$

Where β_{min} is an additive coefficient to control the amount of the movement of the firefly. Scale is a d-dimensional vector which enables the firefly to move within the limits of each dimension. The scale is calculated by the difference between uppethe r bound and lower bound for each dimension.

The objective of this proposed work is to analyze the effect of values of α and β_{min} on the efficiency of the search results. This analysis has been done through extensive experimentation. Various combinations of values of abovethe two parameters are tested and results analyzed aiming to find out a set of values that provides the best results. All the possible combinations of the values 0.2, 0.4, 0.6 and 0.8 for each of the two parameters has been executed and results are noted.

4.1 Experimental set-up

The experimental set up for this execution is as follows:

1) Population Size (n)=20
2) α = 0.2 – 0.8
3) β_{min} =0.2 – 0.8
4) β_0 = 1
5) Termination Criterion: 2000 iterations (40000 NFE)
6) Experiments were done on an Intel core i5 processor with 4 GB RAM and 2.40 GHz frequency.
7) A tool used is Matlab R2013a.

4.2 Results

The experiments were repeated 25 times on the set of 9 benchmark functions and the average values of all the rounds were noted and are listed in table 3. Definitions of these functions are provided in table 2.

Table 2: Definitions of benchmark functions

Function name	Function Definition	Bounds	Optimization point	Optimized Variable Values
Sphere	$f(x) = \sum_{i=1}^{d} x_i^2$	[-100, 100]	f(x*)=0	**x*=(0,0,0....)**
Hyper-Ellipsoid	$f(x) = \sum_{i=1}^{n} x_i^2 i^2$	[-100, 100]	f(x*)=0	**x*=(0,0,0....)**
Rastrigin	$f(x) = 10d + \sum_{i=2}^{d} [x_i^2 - 10\cos(2\pi x_i)]$	[-5.12,5.12]	f(x*)=0	**x*=(0,0,0....)**
Ackley	$f(x) = -a\exp\left(-b\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right) - \exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos(cx_i)\right) + a + \exp(1)$	[-32,32]	f(x*)=0	**x*=(0,0,0....)**
Griewank	$f(x) = \sum_{i=1}^{d} \frac{x_i^2}{4000} - \prod_{i=1}^{d} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	[-600,600]	f(x*)=0	**x*=(0,0,0....)**
Rosenbrock	$f(x) = \sum_{i=1}^{d-1} \left[100x\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2\right]$	[-2.14, 2.14]	f(x*)=0	**x*=(0,0,0....)**
Salomon Function	$f(x) = 1 - \cos\left(2\pi\sqrt{\sum_{i=1}^{D} x_i^2}\right) + 0.1\sqrt{\sum_{i=1}^{D} x_i^2}$	[-100,100]	f(x*)=0	**x*=(0,0,0....)**
Schwefel .2.22	$f(x) = \sum_{i=1}^{n} \lvert x_i\rvert + \prod_{i=1}^{n} \lvert x_i\rvert$	[–10, 10]	f(x*)=0	**F(0, ,0 ,0)=0**
Bent-Cigar Function	$f(x) = x_1^2 + 10^6 \sum_{i=2}^{n} x_i^2$	[-100,100]	f(x*)=0	**F(0, ,0 ,0)=0**

The best values observed for each function are shown in bold font in Table 3.

Table 3: Average values of best solutions for benchmark functions for various combinations of α and β_{min} for 30 dimensions

α	0.2	0.4	0.6	0.8
β_{min}	0.2			
Sphere [-100, 100]	**1.97E-04**	5.88E-04	0.0013	0.0023
Hyper-ellipsoid [-100,100]	1.83E+03	5.21E+03	**1.68E+03**	1.89E+03
Rastrigin [-5.12, 5.12]	53.7278	50.7431	**48.7537**	52.735
Ackley [-32, 32]	**1.55E-06**	6.54E-06	1.53E-05	3.22E-05
Griewank [-600, 600]	**2.38E-04**	0.0012	0.0018	0.003
RosenBrock	**2.66E+01**	2.79E+01	2.84E+01	2.85E+01
Salomon Function	**2.00E-01**	3.00E-01	4.00E-01	0.2999
Schwefel.2.22	**-8.16E+09**	-8.17E+09	-8.09E+09	-8.16E+09
Bent Cigar	**1.28E-01**	3.52E-01	6.38E-01	1.24E+00
β_{min}	0.4			
Sphere[-100, 100]	**1.19E-04**	8.06E-04	0.0026	0.0044
Hyper-ellipsoid [-100,100]	**149.2239**	4.29E+03	2.71E+03	511.3158
Rastrigin [-5.12, 5.12]	35.8186	59.698	**32.8347**	70.6433
Ackley [-32, 32]	**1.07E-06**	0.0123	2.06E-05	2.06E-05
Griewank [-600, 600]	**4.32E-04**	0.0017	0.003	0.0049
RosenBrock	**2.59E+01**	2.82E+01	2.87E+01	2.94E+01
Salomon Function	**2.00E-01**	3.00E-01	3.00E-01	4.00E-01
Schwefel.2.22	**-7.94E+09**	-7.99E+09	-8.16E+09	-7.90E+09
Bent Cigar	**1.13E-01**	4.17E-01	8.38E-01	1.68E+00
β_{min}	0.6			
Sphere [-100, 100]	**2.19E-04**	5.74E-04	0.0021	0.0039
Hyper-ellipsoid [-100,100]	1.62E+03	2.54E+03	**578.0956**	948.7035
Rastrigin [-5.12, 5.12]	50.743	53.728	61.6885	**41.7897**
Ackley [-32, 32]	**1.82E-06**	4.90E-06	0.0148	2.04E-05
Griewank[-600, 600]	**3.59E-04**	0.0011	0.0028	0.0028
RosenBrock	**2.89E+01**	2.81E+01	2.92E+01	2.94E+01
Salomon Function	**2.00E-01**	2.00E-01	3.00E-01	3.00E-01
Schwefel.2.22	-8.12E+09	-8.17E+09	-8.15E+09	-8.07E+09
Bent Cigar	0.0722	0.3853	1.0447	2.3656
β_{min}	0.8			
Sphere [-100, 100]	**2.26E-04**	9.37E-04	0.0018	0.0025
Hyper-ellipsoid [-100,100]	7.85E+03	4.69E+03	5.69E+03	**3.85E+03**
Rastrigin [-5.12, 5.12]	52.7329	47.7583	**37.8093**	79.5982
Ackley [-32, 32]	**1.52E-06**	4.56E-06	1.45E-05	1.94E-05
Griewank [-600, 600]	**5.25E-04**	0.0191	0.003	0.0051
RosenBrock	2.82E+01	2.86E+01	2.92E+01	**2.69E+01**
Salomon Function	**3.00E-01**	3.00E-01	3.00E-01	3.00E-01
Schwefel.2.22	**-7.63E+09**	-7.86E+09	-7.98E+09	-8.15E+09
Bent Cigar	**1.52E-01**	4.71E-01	8.09E-01	2.23E+00

To draw a conclusion from table 3, first, the results were observed to find out the best value of α regardless of the value of β_{min}. This optimized value of α was found to be 0.2. Then the table was observed to find out the value of β_{min} that provides best result ts for α = 0.2 only. The best value of β_{min} found was 0.4.

5. CONCLUSION AND FUTURE WORK

This paper aimed at providing an optimized set of values for two parameters α and β_{min} in an existing variant of firefly algorithm. Extensive experimentation was done on a set of 9 diverse bench mark functions. For most of the problems, best values of objective functions were obtained at values α = 0.2 and β_{min} = 0.4. To validate the performance of the proposed algorithm, nine benchmark functions were used. The future work would be focused on continuous improvement and also to find practical applications of the newly optimized value set of parameters.

6. REFERENCES

[1] https://neos-guide.org/optimization-tree

[2] Xin-She Yang,(2010),” Nature-Inspired Metaheuristic Algorithms”, Luniver Press , Second Edition

[3] Jitendra Rajpurohit, Tarun Kumar Sharma, Ajith Abraham, Vaishali,(2016),” Glossary of Evolutionary Algorithms”,

[4] D. E. Goldberg, (1989), “Genetic Algorithms in Search, Optimization, and Machine Learning”, ADDISON-WESLEY PUBLISHING COMPANY.

[5] R. Eberhart, J. Kennedy,(1995), "A New Optimizer Using Particle Swarm Theory", In *proceedings of the Sixth International Symposium on Machine and Human Science*, pp. 39-43.
[6] Storn R, Price K., "Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces", Journal of Global Optimization, 11(4), pp 341–359, 1997
[7] A. Colorni, M. Dorigo et V. Maniezzo, Distributed Optimization by Ant Colonies, proceedings of the First European Conference on Artificial Life, Paris, France, Elsevier Publishing, 134-142, 1991
[8] Xin-She Yang, (2010)," Firefly Algorithm, Levy Flights, and Global Optimization", Springer-Verlag London Limited.
[9] MO Yuan-bin, MA Yan-zhui, ZHENG Qiao-yan, (2013), Optimal Choice of Parameters for Firefly Algorithm, IEEE, Fourth International Conference on Digital Manufacturing & Automation.
[10] Ivona Brajevi, Jelena Ignjatovi, (2018), An upgraded firefly algorithm with feasibility-based rules for constrained engineering optimization problems, Springer, SpringerNature2018.
[11] A. Manju , M.J. Nigam,(2012), Firefly Algorithm with Fireflies having Quantum Behavior, IEEE, 2012 International Conference on Radar, Communication, and Computing (ICRCC).
[12] Changnian Liu, Yafei Tian, Qiang Zhang, Jie Yuan, Binbin Xue,(2013), Adaptive Firefly Optimization Algorithm Based On Stochastic Inertia Weight, IEEE, 2013 Sixth International Symposium on Computational Intelligence and Design.
[13] Shruti Goel, V. K. Panchal, (2014), Performance Evaluation of a New Modified Firefly Algorithm, IEEE.
[14] Mukesh Gupta, S.N.Tazi, Akansha Jain, Deepika, (2014), Edge Detection Using Modified Firefly Algorithm, IEEE, Sixth International Conference on Computational Intelligence and Communication Networks.
[15] Shubhendu Kumar Sarangi, Rutuparna Panda, (2016), A New Modified Firefly Algorithm for Function Optimization, IEEE, International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT).
[16] Irwan Mahmudi, Ratna Ika Putri, Margo Pujiantara, Ardyono Priyadi, Mauridhi Hery Purnomo,(2016), Modified Firefly Algorithm for Maximum Power Point Tracking Of a Small Stand Alone Wind Power System, IEEE, International Seminar on Intelligent Technology and its Application.
[17] Xiangbo Qi, Sihan Zhu, Hao Zhang,(2017), A Hybrid Firefly Algorithm, IEEE.
[18] Mahdi Bidar, Samira Sadaoui, Malek Mouhoub, Mohsen Bidar,(2017), Improving Firefly Algorithm Performance using Fuzzy Logic, IEEE, International Conference on Systems, Man, and Cybernetics (SMC).
[19] Shubhendu Kumar Sarangi, Rutuparna Panda,(2017), Crazy Firefly Algorithm for Function Optimization, IEEE, 2nd International Conference on Man and Machine Interfacing (MAMI).
[20] Xiuqin Pan, Limiao Xue, Ruixiang Li,(2018), A new and efficient firefly algorithm for numerical optimization problems, Springer, The Natural Computing Applications Forum 2018.
[21] Huali Xu, Shuhao Yu, Jiajun Chen, Xukun Zuo,(2018), An Improved Firefly Algorithm for Feature Selection in Classification, Springer Nature 2018.

