Performance evaluation of firefly algorithm with variation in sorting for non-linear benchmark problems

Cite as: AIP Conference Proceedings **1836**, 020032 (2017); https://doi.org/10.1063/1.4981972
Published Online: 05 June 2017

A. J. Umbarkar, U. T. Balande and P. D. Seth

Performance Evaluation of Firefly Algorithm with Variation in Sorting for Non-Linear Benchmark Problems

A. J. Umbarkar[1, a)] U. T. Balande[2] and P.D.Seth[3, b)]
(Use the Microsoft Word template style: *Paper Author*)

[1, 2]*Department of Information Technology,*
Walchand College of Engineering, Sangli, 416416,
MS, India.
[3]*Department of Master of Computer Applications,*
Government College of Engineering, Karad, 415124,
MS, India.
[a)]Corresponding author: anantumbarkar@rediffmail.com
[b)]pranalisheth@gmail.com

Abstract. The field of nature inspired computing and optimization techniques have evolved to solve difficult optimization problems in diverse fields of engineering, science and technology. The firefly attraction process is mimicked in the algorithm for solving optimization problems. In Firefly Algorithm (FA) sorting of fireflies is done by using sorting algorithm. The original FA is proposed with bubble sort for ranking the fireflies. In this paper, the quick sort replaces bubble sort to decrease the time complexity of FA. The dataset used is unconstrained benchmark functions from CEC 2005 [22]. The comparison of FA using bubble sort and FA using quick sort is performed with respect to best, worst, mean, standard deviation, number of comparisons and execution time. The experimental result shows that FA using quick sort requires less number of comparisons but requires more execution time. The increased number of fireflies helps to converge into optimal solution whereas by varying dimension for algorithm performed better at a lower dimension than higher dimension.

INTRODUCTION

Optimization is an active research area from few decades. To find optimal value to such non-linear problems requires efficient algorithms. Optimization algorithm are divided into two categories: deterministic algorithms and stochastic algorithms [1]. Deterministic algorithms comprise almost all local search algorithms. Deterministic algorithms are efficient in finding local optima. Hill-climbing is a good example of deterministic algorithm. Stochastic algorithms have a deterministic component and a random component. Genetic algorithm is good example of this algorithm. [2].

Mostly nature inspired algorithms are population based optimization algorithms and have proved to be very efficient to find a solution for constrained and unconstrained optimization problems. Evolutionary algorithms [3] and Swarm Intelligence [4] based algorithms are two important classes of population based optimization algorithms. The most popular example of one such Evolutionary Algorithm is Genetic Algorithm (GA). Many modern meta-heuristic algorithms are swarm intelligent in nature. A swarm intelligence based algorithm has gained interest of many researchers. Our prime focus is solving unconstrained problem through swarm based algorithms.

The Firefly Algorithm (FA) [5-7] is a nature- inspired swarm intelligence based optimization algorithm which is based on social (flashing) behavior of fireflies or lighting bugs, at the summer sky in tropical temperature regions. In general, the FA has resemblance with different algorithms which is based on the swarm intelligence, such as Particle Swarm Optimization (PSO) [8], Bacterial Foraging (BFA) Algorithm, Ant Colony Optimization (ACO) [9] and Artificial Bee Colony (ABC) [10] etc. The latest swarm intelligence based optimization techniques are Firefly Algorithm, Cuckoo search (CS) [11], Bat algorithm (BA) [12] and Krill herd bio-inspired optimization algorithm

Applied Mathematics and Computer Science
AIP Conf. Proc. 1836, 020032-1–020032-9; doi: 10.1063/1.4981972
Published by AIP Publishing. 978-0-7354-1506-5/$30.00

[13] which is emerged recently. Furthermore, according to recent bibliography, swarm intelligence algorithms are very powerful and can perform better than other conventional algorithms, such as GA, for solving various optimization problems.

Optimization of benchmark functions using nature inspired Firefly Algorithm is an application in computational optimization. Benchmark functions [14] are used to control and review the performance of the algorithms. The standard benchmark functions used to review the performance of algorithms are unimodal or multimodal [15] with low and high dimensionality. The complexity of solving a NP-hard combinational problem is higher than any other optimization problems with no guarantee of reaching a global optimum.

Paper is organized as follows – introduction, followed by literature review then the fundamentals of Firefly Algorithm. Next Section describes Firefly Algorithm with variation in sorting followed by experimental results, conclusion and future scope for work.

LITERATURE REVIEW

Modern meta-heuristic algorithms are being developed and begin to show their power and efficiency. Firefly Algorithm (FA) is one of them. It was introduced by Yang at Cambridge University in 2007. Recently, Swarm Intelligence has received more attention from the research community because of changing computing infrastructures like single core CPU to multi-core GPGPU (General Purpose Computing on Graphic Hardware). FA is intended to provide following benefits [5]:

- FA has a fast convergence rate.
- It can be used as a general, global problem solver.
- It can serve as a local search heuristic problem.
- FA is simple, flexible and versatile, which is very efficient in solving a wide range of diverse real–world problems.

Bhushan et al. [8] compared FA and PSO with result carried out on standard benchmark functions. Senthilnath et al. [16] applied the FA for clustering and its performance compared with the ABC, PSO and other nine methods. Farahani et al. [17] stabilize movement of fireflies to find global best and convergence speed with using Gaussian distribution. Gandomi et al. [18] studied a set of non-linear constraint optimization problems in engineering. Results show that FA provides better results than PSO. Lukasik et al. [19] experimented the classical FA for constrained continuous optimization. The obtained results were compared with the PSO algorithm. Adil et al. [20] used FA for unconstrained optimization.

FIREFLY ALGORITHM

Concepts

Firefly Algorithm uses, three idealized rules for development of algorithm [21]. The following rules are obtained by observing the natural behavior of fireflies.

1. All fireflies are unisexual, and so therefore potentially attracted to any of the other fireflies disregarding of their sex.
2. Attractiveness is proportional to their brightness thus for any two fireflies, the less bright firefly is attracted and moved to the brighter one. The brightness and attractiveness decrease as the distance between fireflies is increased. If there is no brighter one than a particular firefly, it will move randomly.
3. The brightness of a firefly is determined by the landscape of the objective function.

Table 1 describes the parameters and notations used in implementation of Firefly Algorithm for solving optimization functions. Figure 1 at the bottom of paper describes the methodology adopted in implementation of FA.

Attractiveness and Light Intensity

Firefly Algorithm has two parameters, first, the variation of light intensity and second, attractiveness of firefly. The light intensity $I(r)$ differs according to inverse square law.

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

Where, Is is the intensity of the source. For a fixed absorption coefficient ‘γ’, light intensity I varies with distance ‘r'.

$$I = I_0 e^{-\gamma r} \tag{2}$$

Where, I_o is original light intensity. In order to avoid singularity at $r = 0$ in Eq.(1). The combined effect of both 'inverse square law' and 'absorption law' can be approximated using Gaussian form.

$$I = I_0 e^{-\gamma r^2} \tag{3}$$

The attractiveness β of any firefly is proportionate to its light intensity of the other fireflies adjacent to it. It is given in Eq. (4). Where β0 is the attractiveness at $r = 0$.

$$\beta = \beta_0 e^{-\gamma r^2} \tag{4}$$

Distance and Movement

The distance between any two fireflies i and j at X_i and X_j by Cartesian distance is given by:

$$r_{ij} = | x_i - x_j | = \sqrt{\sum_{k=1}^{d} (Xi,k - Xj,k)^2} \tag{5}$$

Where, d is number of dimensions and k represents component in spatial coordinate. The movement of any firefly i towards more attractive firefly j is determined in Eq. (6)

$$x_i = x_i + \beta_0 e^{-\gamma r^2} (x_j - x_i) + \alpha\ (rand\text{-}0.5) \tag{6}$$

Where, α is randomization parameter and rand function generator.

TABLE 1. Parameter and notation of FA.

Sr. No.	Notation	
	Parameter	**Notation in algorithm**
1	Brightness	Objective Function
2	Beta(β)	Attractiveness
3	Alpha(α)	Randomization parameter
4	Gamma(γ)	Absorption coefficient
5	Number of generations(g)	Iteration
6	Number of Fireflies(n)	population
7	Problem Dimension(D)	Dimension

FA WITH VARIATIONS IN SORTING

The behavior of Firefly Algorithm is described in Fig 1. In the traditional basic FA sorting of number of fireflies has carried with bubble sort which has all cases time complexity of $O(n^2)$. This method will have large processing time.

This paper replaces the bubble sort by quick sort. Quick sort algorithm is selected from various sorting algorithm because of less time complexity. Quick sort is a divide and conquer based sorting algorithm suitable for parallelization of FA. Quick sort based FA algorithm will have the average case time complexity of $O(n \log n)$, where n is equal to number of fireflies. Figure 2 describes FA with bubble sort for firefly sorting, where the exchange of fitness values done according to attractiveness of fireflies. Figure 3 describes FA with quick sort for firefly sorting, in which pivot element assigned to the first element of fitness value.

RESULTS AND DISCUSSION

The Firefly Algorithm is implemented in C programming language on desktop machine with-AMD FX(tm)-8320 8 core processor CPU 3.51 GHz, 16 GB RAM, and 500 GB HDD Table 2 gives the parameter setting of FA used in experimentation.

To test the performance of the Firefly Algorithm, problem set (15 functions), F1 to F15 from Congress on Evolutionary Computation-2005 [22] (CEC 2005) is used for experimentation.

There are two experiments are performed. In the first experiment the performance evaluation of FA using bubble sort and that by using quick sort is carried out. In the second experiment analysis of FA by varying number of fireflies (n) and varying dimension (D) is performed.

TABLE 2. Parameter and notation of FA.Parameter Setting of FA

Sr. No.	Setting of Parameter of FA	
	Parameter	**Values**
1	Beta(β)	0.2
2	Alpha(α)	0.5
3	Gamma(γ)	1.0
4	Number of generation (G)	8000
5	Number of Fireflies(n)	10,20,30,40
6	Problem Dimension(D)	10,20,30,40

Experiment 1: Performance evaluation of FA using bubble sort and quick sort

The performance of FA with bubble sort and quick sort calculated in 30 independent runs for 15 benchmark unconstrained function.

In the first experiment, the objective function value has been calculated using bubble sort for all 15 functions over a fixed number of generations, number of fireflies and dimension. The Table 3 shows that, FA with quick sort which finds fitness values close to the optimal fitness values given in CEC 2005 [22]. The result is calculated using statistical measures such as of best, worst, mean solution, standard deviation, and standard error of mean. The number of comparisons required using quick sort is found lesser than that of bubble sort.

TABLE 3. Result obtained for FA with bubble sort and FA with quick sort

Funtion	Best	Worst	Mean	Standard Deviation (SD)	Standard Error of Mean (SEM)	Number of comparisons Bubble sort	Number of comparisons Quick sort
F1	7.70E-02	1.02E+00	8.65E-01	1.77E-01	3.24E-02		2802027
F2	0.00E+00	7.15E-02	3.51E-02	1.98E-02	3.61E-03		3012142
F3	1.64E-01	7.26E+00	2.49E+00	2.28E+00	4.16E-01		3005135
F4	3.01E-01	3.18E+00	1.97E+00	1.02E+00	1.86E-01		2813214
F5	1.86E-02	1.06E+00	7.60E-02	1.87E-01	3.42E-02		3014009
F6	2.66E-01	1.03E+00	6.00E-01	2.77E-01	5.05E-02		2966534
F7	1.64E-01	1.33E+01	3.02E+00	3.25E+00	5.93E-01		2953825
F8	7.81E-02	2.47E-01	1.50E-01	6.98E-02	1.27E-02		2836785
F9	2.42E-02	1.02E+00	7.87E-02	1.78E-01	3.26E-02	6240000	2979699
F10	2.75E-02	1.08E+00	9.63E-02	1.87E-01	3.42E-02		3014849
F11	0.00E+00	1.10E-05	1.18E-04	6.33E-04	1.16E-04		2853885
F12	6.67E-01	1.87E+00	1.04E+00	3.87E-01	7.06E-02		2999350
F13	1.46E-01	9.68E+00	2.65E+00	3.16E+00	4.88E-01		2781390
F14	2.83E-01	1.28E+00	5.51E-01	5.84E-01	1.07E-01		2809502
F15	0.00E+00	1.10E-05	2.04E-06	3.76E-06	6.86E-07		2774829

Experiment 2: Analyze FA by varying number of fireflies (n) and varying dimension (D)

The results is taken on each of 15 benchmark functions by varying the number of fireflies (n) with fixed number of iteration and fixed dimension (D). Figure 4 shows that, FA's performance increases with increase in the number of fireflies for function F3.

In case of variation in the dimension for F3 by keeping number of fireflies (n) and number of iterations fixed (result plotted in Fig. 5), the performance of FA degrades with increase in dimensions.

CONCLUSION AND FUTURE WORK

The firefly algorithm uses the process of attraction based on the brightness of fireflies to optimize an objective function. Prior research has shown that the algorithm can solve optimization problems.

In this work, experimentation is performed with FA with bubble sort and quick sort to check the performance in terms of best, worst and mean solution, execution time, and number of comparisons. The performance of FA with bubble sort and FA with quick sort, both found similar and satisfactory in terms of best, worst, mean solution. The performance of FA with bubble sort takes less execution than the FA with quick sort. FA with quick sort is effective in terms of the time complexity as compare to FA with bubble sort. FA's performance in terms of best, worst and mean solution, convergence rate depends on the number of fireflies.

In future, researchers could experiment the various sorting algorithms for solving real life optimization problem. The parallel sorting algorithm could replace by traditional sorting algorithms for getting speedup on the latest computing infrastructure.

ACKNOWLEDGMENT

We express our sincere thanks to the all authors, whose papers in the area of swarm intelligence published in various journals and conference proceedings.

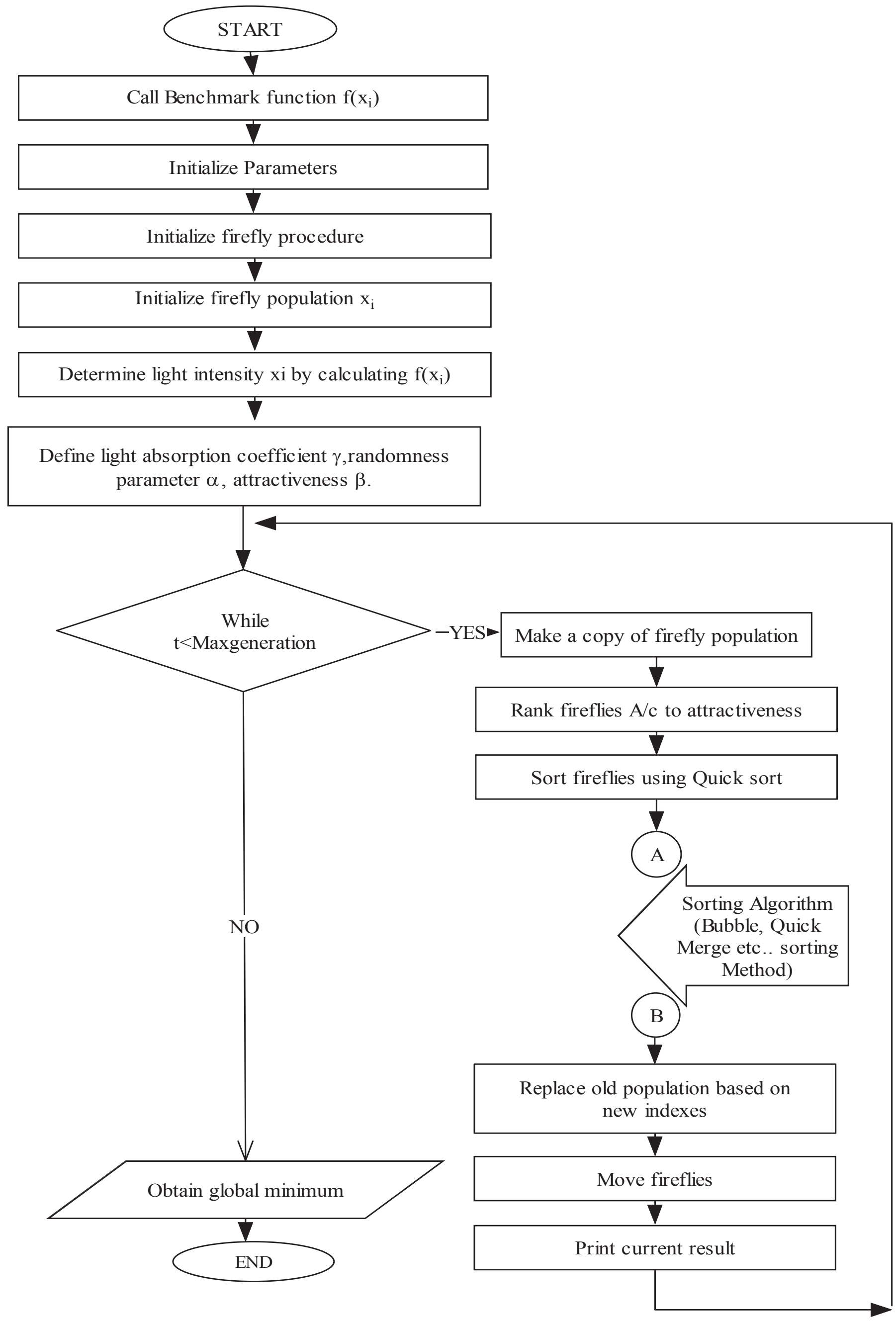

FIGURE 1. Flow chart of Firefly Algoritm

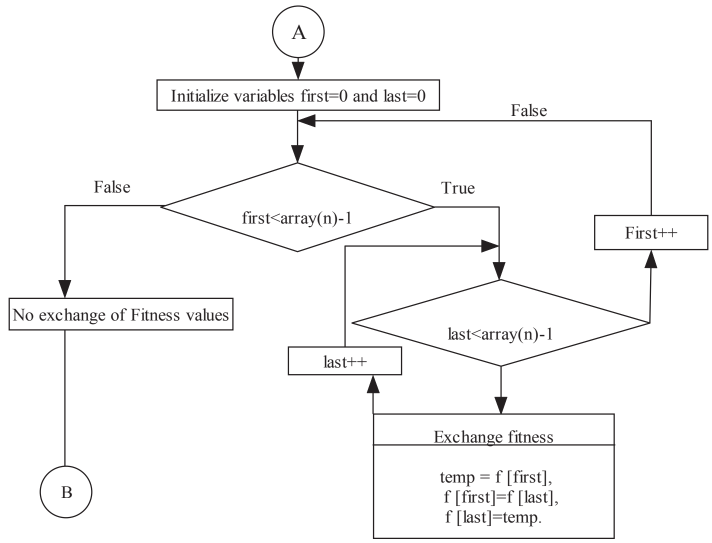

FIGURE 2. FA using bubble sort for firefly sorting

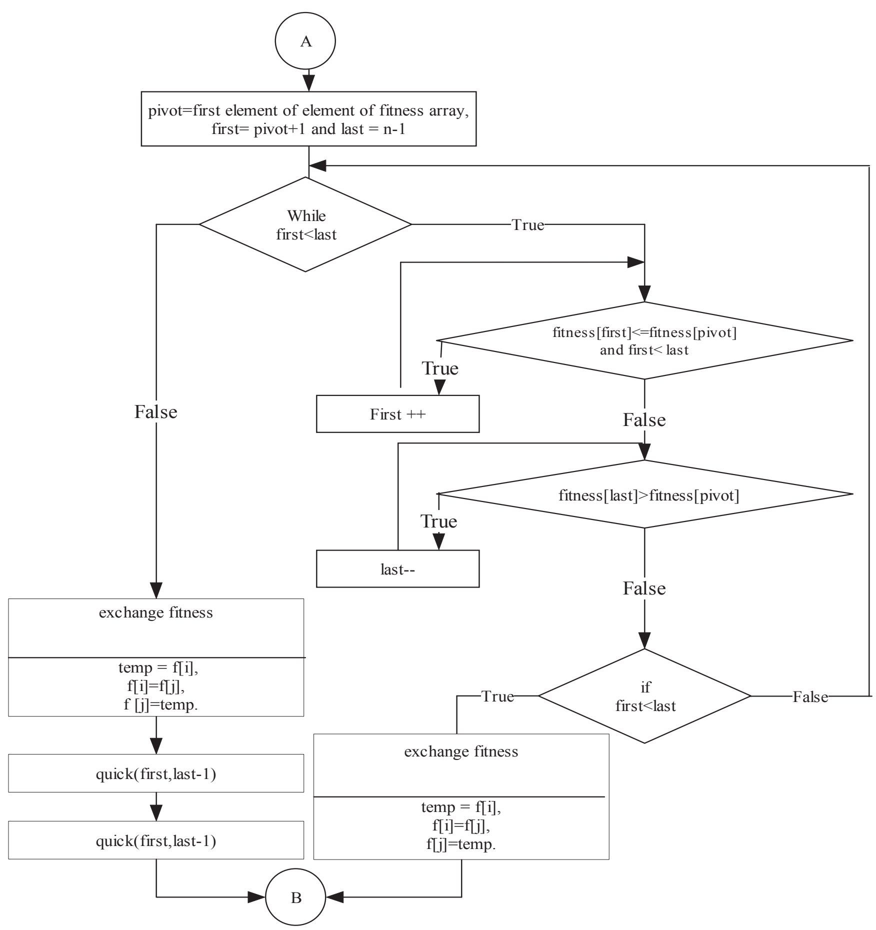

FIGURE 3. FA using quick sort for firefly sorting

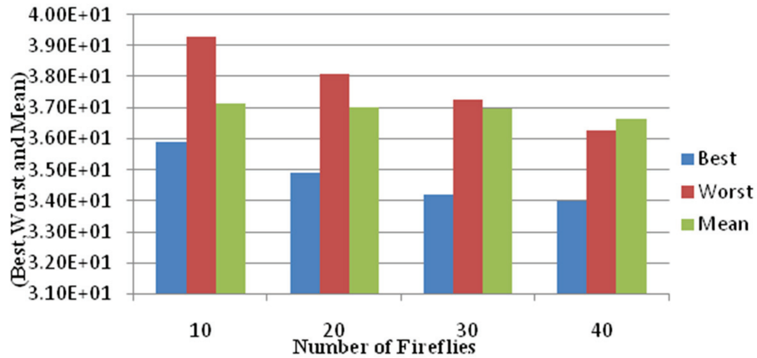

FIGURE 4. By varying number of fireflies of Rosenbrock test function (F3)

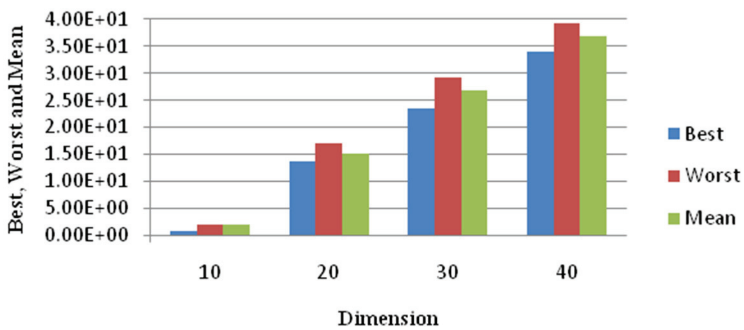

FIGURE 5. By varying dimension of Rosenbrock test function (F3).

REFERENCES

1. T. Kanimozhi and K. Latha, "Stochastic firefly for image optimization," in *2013 International Conference on Communications and Signal Processing (ICCSP)*, pp. 592-596,2013.
2. X.S.Yang and S.Deb, "Eagle strategy using levy walk and firefly algorithms for stochastic optimization," in *Nature Inspired Cooperative Strategies for Optimization (NICSO2010)*, pp. 101-111,2010,
3. A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing: SpringerVerlag*, 2003.
4. C.Blum and X.Li, "Swarm Intelligence: Introduction and Applications," *Swarm intelligence in optimization*, in: C. Blum, D. Merkle(Eds.), pp. 43-86, 2008.
5. X. S. Yang, *Nature-Inspired Metaheuristic Algorithms, Second ed.* Luniver Press Frome, BA11 6TT, United Kingdom: Luniver Press, 2008.
6. I. Fister, X.-S. Yang, D. Fister, and I. Fister, Jr., "Firefly Algorithm: A Brief Review of the Expanding Literature," in *Cuckoo Search and Firefly Algorithm*, vol. 516, Yang, Ed., ed: Springer International Publishing, pp. 347-360, 2014.
7. I. Fister, I. Fister Jr, X.-S. Yang, and J. Brest, "A comprehensive review of firefly algorithms," *Swarm and Evolutionary Computation*, vol. 13, pp. 34-46, 2013.
8. B. Bhushan and S. S. Pillai, "Particle Swarm Optimization and Firefly Algorithm: Performance analysis," in *2013 IEEE 3rd International Advance Computing Conference (IACC)* , pp. 746-751, 2013.
9. M. Dorigo and G. D. Caro, *The ant colony optimization meta-heuristic*, pp. 11-32, 1999.
10. D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of Global Optimization*, vol. 39, pp. 459-471, 2007.
11. Y. Xin-She and S. Deb, "Cuckoo Search via Levy flights," in *2013 International Conference on Nature & Biologically Inspired Computing, ICCCCM)*, pp. 1-4, 2013.
12. A. H. Gandomi and A. H. Alavi, "Krill herd: A new bio-inspired optimization algorithm," *Communications in Nonlinear Science and Numerical Simulation*, vol. 17, pp. 4831-4845, 2012.
13. A. Manju and M. J. Nigam, "Application of exponential atmosphere concept in improving Firefly Algorithm," in *2012 Third International Conference on Computing Communication & Networking Technologies (ICCCNT)*, pp. 1-6 , 2012.
14. S. Arora and S. Singh, "A conceptual comparison of firefly algorithm, bat algorithm and cuckoo search," in *2013 International Conference on Control, Computing, Communication and Materials (ICCCCM),* 2013.
15. L. F. F. Miguel, R. H. Lopez, and L. F. F. Miguel, "Multimodal size, shape, and topology optimization of truss structures using the Firefly algorithm," *Advances in Engineering Software*, vol. 56, pp. 23-37, 2013.
16. J. Senthilnath, S. N. Omkar, and V. Mani, "Clustering using firefly algorithm: Performance study," *Swarm and Evolutionary Computation*, vol. 1, pp. 164-171, 2011.
17. S. M. Farahani, A. A. Abshouri, B. Nasiri, and M. R. Meybodi, "A Gaussian Firefly Algorithm," *International Journal of Machine Learning and Computing*, vol. 1, No. 5, pp. 448-453, 2011.
18. A. H. Gandomi, X.-S. Yang, and A. H. Alavi, "Mixed variable structural optimization using Firefly Algorithm," *Comput. Struct.*, vol. 89, pp. 2325-2336, 2011.
19. S. Łukasik and S. Żak, "Firefly Algorithm for Continuous Constrained Optimization Tasks," *in Computational Collective Intelligence. Semantic Web, Social Networks and Multiagent Systems*. vol. 5796, N. Nguyen, R. Kowalczyk, and S.-M. Chen, Eds., ed: Springer Berlin Heidelberg, , pp. 97-106, 2009.
20. A. Hashmi, N. Goel, and Shruti, "Firefly Algorithm for Unconstrained Optimization," *IOSR Journal of Computer Engineering,* pp. 75-78, 2013.
21. X. S. Yang, "Multiobjective firefly algorithm for continuous optimization," *Engineering with Computers*, vol. 29, pp. 175-184, 2013.
22. P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y.-P. Chen, A. Auger and S. Tiwari, "Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization", *Technical Report*, Nanyang Technological University, Singapore, May 2005 AND KanGAL Report #2005005, IIT Kanpur, India.