Volume 3, Issue 5, May 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Performance of Firefly and Bat Algorithm for Unconstrained Optimization Problems

Nishant Goel,* Divya Gupta, Shruti Goel
Department of Computer Science
Maharaja Surajmal Institute of Technology
GGSIP university
C-4, Janakpuri, New Delhi, India

Abstract— Nature inspired meta-heuristic algorithms studies the emergent collective intelligence of groups of simple agents. This behaviour can efficiently be used to find the solutions of various global optimization problems. This paper compares performances of two latest of these algorithms namely Bat and Firefly algorithm for unconstrained optimization problems. Global optima are found using various test functions of different characteristics on the basis of convergence speed and precision.

Keywords— Firefly Algorithm, Bat Algorithm, Unconstrained Optimization, Benchmark Functions, Nature-Inspired Algorithms.

I. INTRODUCTION

Swarm based algorithms [1] are very efficient in solving various optimization problems [2]. Various swarm based algorithms have been developed and are implemented for solving unconstrained optimization problems [3]. Swarm Intelligence is a collective behaviour of decentralized, self-organized natural or artificial systems. The concept has been taken from the natural swarms which are the aggregation of animals such as fish schools, bird flocks and insect colonies. This swarm performs collective behaviour of two types-Self Organization and Stigmergy [4]. Self-Organization is a set of dynamical mechanisms whereby structures appear at the global level of a system from interactions of its lower level components [5]. Stigmergy is a mechanism of indirect coordination between agents or actions [6]. Firefly and Bat Algorithms are checked for their performance in solving unconstrained optimization problem for single objective function [7]. In single objective optimization problem only one optimum solution exists. However, the search may have many local optimum solutions but the goal is to find one global solution only. In these problems if a new solution has a better objective function value than the old solution, the new solution is accepted. A single objective problem is concerned with only one objective function and has a single solution space [4]. An algorithm works in this space by accepting or rejecting the solutions based on their respective function values. This paper shows the performance of Firefly and Bat Algorithm in solving these unconstrained optimization problems. The performance is calculated by implementing various benchmark functions [8] on the basis of convergence speed and precision. Section 2 summarizes swarm-based algorithms- Firefly and Bat Algorithm. Section 3briefly describes the test functions used along with their simulation results. Section 4 reports experimental settings of the algorithm and experimental analysis on the two algorithms. Finally, Section 5 concludes the work done.

II. SWARM BASED OPTIMIZATION ALGORITHMS

Firefly and Bat Algorithms are well known for their efficiency in the field of optimization. These swarm inspired algorithms are well researched in providing good results for unconstrained optimization problems.

A. *Firefly Algorithm(FA)*

Xin She Yang in 2008 proposed the Firefly algorithm which is inspired by the flashing behavior of fireflies. *Fireflies or lightning bugs* belong to family of insects that are capable to produce natural light. The firefly uses its flash characteristic either to attract a mate (for mating) or to attract a prey (for eating). It glows brighter whenever it is hungry or in searches for a mate making the attraction of insects or mates more effective. The brightness of the bioluminescent light depends on the available quantity of a pigment called *luciferin*, and more pigment means more light. These characteristics of fireflies were used by Yang to draw the following three idealized rules for the firefly algorithm:

1. A firefly will be attracted by other fireflies regardless of their sex.
2. Attractiveness is proportional to their brightness and decreases as the distance among them increases.
3. The landscape of the objective function determines the brightness of a firefly [9].

For a maximization problem, attractiveness β of a firefly is determined by its brightness or light intensity I, which is turn, is proportional to the value of objective function i.e.

$$I(x) \propto f(x) \qquad \text{for a particular position x.}$$

As the light intensity I decreases with the distance from its source, the attractiveness β differ with the distance r_{ij} between firefly i and firefly j. Light is also absorbed by the media, so we allow the attractiveness to vary with the varying degree of absorption i.e.

$$I(r) = I_0 e^{-\gamma r^2} \tag{1}$$

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

where γ is the absorption coefficient.

The distance between any two fireflies i and j at x_i and x_j respectively, the Cartesian distance is determined by equation (3) where $x_{i,k}$ is the kth component of the spatial coordinate x_i of the ith firefly and d is the number of dimensions.

$$r_{ij} = \sqrt{\sum_{k=1}^{d} (x_{i,k} - x_{j,k})^2} \tag{3}$$

The movement of a firefly i is attracted to another more attractive (brighter) firefly j is determined by

$$x_i = x_i + \beta_0 e^{-\gamma r_{ij}^2} \left(x_j - x_i\right) + \alpha\epsilon \tag{4}$$

Where the second term is due to the attraction while the third term is randomization with α being the randomization parameter and ϵ being the vector of random numbers drawn from a Gaussian distribution.

The parameter γ characterizes the contrast of the attractiveness and its value varies from 0.1 to 10 determining the convergence speed of the Firefly algorithm.

Pseudo code for FA:

1. Objective function of f(x), where $x=(x_1,........,x_d)$
2. Generate initial population of fireflies;
3. Formulate light intensity I;
4. Define absorption coefficient γ;
5. While (t<MaxGeneration)
6. For i = 1 to n (all n fireflies);
7. For j=1 to n (all n fireflies)
8. If ($I_j > I_i$), move firefly i towards j;
9. end if
10. Evaluate new solutions and update light intensity;
11. End for j;
12. End for i;
13. Rank the fireflies and find the current best;
14. End while;
15. Post process results and visualization;
16. End procedure;

B. Bat Algorithm(BA)

Xin She Yang in 2010 proposed the Bat algorithm which is inspired by the echolocation behavior of bats. A type of sonar is used by microbats to detect prey, avoid obstacles and locate their roosting crevices in the dark. Yang used these characteristics of bat to draw three important rules:

1. All bats use echolocation to sense distance, and they also `know' the difference between food/prey and background barriers in some magical way.
2. Bats fly randomly with velocity v_i at position x_i with a fixed frequency f_{min}, varying wavelength λ and loudness A_0 to search for prey. They can automatically adjust the wavelength (or frequency) of their emitted pulses and adjust the rate of pulse emission r in the range of [0, 1], depending on the proximity of their target.
3. Although the loudness can vary in many ways, we assume that the loudness varies from a large (positive) A_0 to a minimum constant value A_{min} [10].

For simplicity, the frequency f Є [0, f_{max}], the new solutions x_i^t and velocity v_i^t at a specific time step t are represented by a random vector drawn from a uniform distribution.

Pseudo code for Bat Algorithm:

1. Objective function f(x), $x = (x_1, ..., x_d)$;
2. Initialize the bat population xi (i = 1,2, ..., n) and v_i;
3. Define pulse frequency f_i at x_i;
4. Initialize pulse rates r_i and the loudness A_i;
5. while (t <Max number of iterations)
6. Generate new solutions by adjusting frequency, and updating velocities and locations/solutions;
7. if (rand > r_i)
8. Select a solution among the best solutions;
9. Generate a local solution around the selected best solution;
10. end if;
11. Generate a new solution by flying randomly;
12. if (rand < A_i & $f(x_i) < f(x_*)$)
13. Accept the new solutions;
14. Increase r_i and reduce A_i;
15. end if;
16. Rank the bats and find the current best x_*;
17. end while

III. Benchmark Functions

To test the performance of the two algorithms, six well known benchmark functions are used. These functions are useful to evaluate characteristics of optimization algorithms, such as convergence speed, precision, robustness and general performance [11]. Some of the benchmark functions are unimodal, while others are multimodal. A function is called unimodal if it has only one optimum position. The multimodal functions have two or more local optima.

TABLE 1

TEST FUNCTIONS: (D-DIMENSION OF THE FUNCTION)

S.No	Range	Function	Formulation	F(x)	Characteristics
1	[-10,10]	Step	$f(x) = \sum_{i=1}^{n} (\lvert x_i + 0.5 \rvert)$	0	US
2	[-2,2]	Sphere	$f(x) = \sum_{i=1}^{n} ix_i^2$	0	US
3	[-10,10]	Sum square	$f(x) = \sum_{i=1}^{n} ix_i^2$	0	US
4	[-100,100]	Trid10	$f(x) = \sum_{i=1}^{n} (x_i - 1)^2 - \sum_{i=2}^{n} x_i x_{i-1}$	-210	UN
5	[-0.5,10]	Zakharov	$f(x) = \sum_{i=1}^{n} x_i^2 + (\sum_{i=1} 0.5ix_i)^2 + (i = 1 \sum_{i=1} 0.5ix_i)^4$	0	UN
6	[-5,10]	Rosenbrock	$f(x) = \sum_{i=1}^{n-1} [100 \left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2]$	0	UN

IV. Experimental Analysis

The experimental environment is implemented in MATLAB programs and executed on a DELL Studio15 computer with the configuration of Intel Core I3 CPU M370 at 2.40GHz and 4GB RAM. In each experimental run, for each population size the six standard functions are processed ten times to measure the elapsed time taken and to find the average of the best objective function values. The parameter settings for benchmark functions for both the algorithms-Bat and Firefly are given in Table 2 whereas the experimental results for objective function values and processing time are shown in Table 3 and Table 4 respectively. The number of variable i.e. the dimension of both the algorithms has been kept constant equal to 10.

TABLE 2

PARAMETERS SET FOR BAT AND FIREFLY ALGORITHMS

BAT ALGORITHM	**FIREFLY ALGORITHM**
A (loudness): 0.5	α (randomness): 0.2
R (pulse rate): 0.5	γ (absorption): 1.0
Q_{min} (frequency minimum) : 0	β : 0

Q_{max} (frequency maximum): 2	β_0 : 0
Population Size : 2,5,10,20,40	Population Size : 2,5,10,20,40

TABLE3

COMPARISON OF ALGORITHMS WITH RESPECT TO OBJECTIVE FUNCTION VALUES. (N-POPULATION SIZE)

Test functions	N	Bat Algorithm			Firefly Algorithm		
		Best	Worst	Mean	Best	Worst	Mean
Sphere function	2	0.9914	9.0211	5.7316	$2.25e^{-8}$	$8.99e^{-8}$	$6.28e^{-8}$
	5	0.3825	4.4809	2.0963	$2.67e^{-8}$	$8.43e^{-8}$	$4.40e^{-8}$
	10	$1.312e^{-6}$	0.0325	0.0050	$2.45e^{-8}$	$6.76e^{-8}$	$4.17e^{-8}$
	20	$6.139e^{-7}$	$1.180e^{-6}$	9.0460	$1.17e^{-8}$	$4.45e^{-8}$	$2.09e^{-8}$
	40	$5.555e^{-7}$	$1.187e^{-6}$	6.2461	$1.78e^{-8}$	$7.57e^{-8}$	$4.39e^{-8}$
Sum Square Function	2	1.0344	25.211	9.4361	$4.96e^{-6}$	$1.47e^{-5}$	$1.14e^{-5}$
	5	$2.30e^{-5}$	0.6636	0.8141	$6.02e^{-6}$	$1.82e^{-5}$	$1.06e^{-5}$
	10	$8.01e^{-6}$	$1.695e^{-5}$	$9.219e^{-5}$	$3.24e^{-6}$	$1.54e^{-5}$	$9.23e^{-7}$
	20	$7.42e^{-6}$	$1.246e^{-5}$	$25.67e^{-5}$	$5.01e^{-6}$	$1.48e^{-5}$	$1.40e^{-5}$
	40	$0.10e^{-6}$	$9.844e^{-6}$	$5.458e^{-6}$	$3.23e^{-6}$	$1.26e^{-5}$	$0.87e^{-5}$
Step Function	2	113.50	312.36	190.720	$4.10e^{-7}$	$1.39e^{-6}$	$5.07e^{-7}$
	5	20.980	135.41	78.4565	$6.29e^{-7}$	$5.27e^{-6}$	$7.43e^{-7}$
	10	8.9720	125.42	45.4720	$4.26e^{-7}$	$1.31e^{-6}$	$9.26e^{-7}$
	20	0.7910	36.676	20.1534	$4.52e^{-7}$	$1.95e^{-6}$	$9.72e^{-7}$
	40	$1.07e^{-6}$	0.0104	0.00105	$4.22e^{-7}$	$1.71e^{-6}$	$9.37e^{-7}$
Trid10 Function	2	4642.0	24054.3	15908.9	-209.6	974.2	33.10
	5	3311.1	23145.5	11555.7	-209.9	-205.4	-208.0
	10	967.38	12915.7	5825.42	-209.9	-209.9	-209.9
	20	657.56	5120.91	2815.86	-209.9	-209.9	-209.9
	40	511.19	2674.69	2031.53	-209.9	-209.9	-209.9
Zakharov Function	2	481.83	274688.7	53342.0	-0.090	25.17	4.000
	5	156.07	88666.4	19489.3	-0.090	-0.090	-0.090
	10	96.340	1689.21	633.858	-0.090	-0.090	-0.090
	20	71.440	1864.08	546.703	-0.090	-0.090	-0.090
	40	24.430	629.610	230.117	-0.090	-0.090	-0.090
Rosenbrock Function	2	-6191.0	-Inf	-Inf	$-9.91e^{+3}$	$-9.98e^{+3}$	$-9.94e^{+3}$
	5	$-850e^{+153}$	-Inf	-Inf	$-9.98e^{+3}$	$-9.99e^{+3}$	$-9.99e^{+3}$
	10	$-571e^{+153}$	-Inf	-Inf	$-9.99e^{+3}$	$-9.99e^{+3}$	$-9.99e^{+3}$
	20	$-167e^{+153}$	-Inf	-Inf	$-9.99e^{+3}$	$-9.99e^{+3}$	$-9.99e^{+3}$
	40	$-149e^{+153}$	-Inf	-Inf	$-9.99e^{+3}$	$-9.99e^{+3}$	$-9.99e^{+3}$

Table 3 gives the best, mean and worst values of the optima obtained by both the algorithms for each benchmark functions. As we can see, the Firefly algorithm has outperformed the Bat algorithm in all six benchmark functions. For unimodal-separable functions (Step, Sphere and Sum-Square), Firefly algorithm values for almost all test functions are better by factor of 10^{-6}. For unimodal-nonseparable functions, we found different results. For Trid10 and Zakharov functions, the Firefly algorithm achieved the global optima unlike the Bat algorithm whereas for Rosenbrock function, the Firefly algorithm showed slightly better results than the bat algorithm but the global optima was still not achieved.

Population size has also played a vital role in the comparison as we could clearly see that as we increase the population size both the algorithms reach a more optimum value. We can also estimate from the table that population size of 2 and 5 do not give required results for bat algorithm and hence it is advised to keep the population size above 10 for this algorithm (typically between 10 to 40).

TABLE 4

COMPARISON OF ALGORITHMS WITH RESPECT TO PROCESSING TIME IN SECONDS(N-POPULATION SIZE)

Test functions	N	Bat Algorithm			Firefly Algorithm		
		Best	Worst	Mean	Best	Worst	Mean
Sphere function	2	0.070	0.212	0.088	0.086	0.192	0.113
	5	0.154	0.217	0.178	0.188	0.296	0.224
	10	0.304	0.362	0.326	0.456	0.655	0.534

	20	0.553	0.677	0.606	1.495	1.784	1.677
	40	1.129	1.238	1.186	5.587	6.253	5.900
Sum Square Function	2	0.097	0.234	0.119	0.116	0.183	0.144
	5	0.223	0.310	0.256	0.156	0.323	0.220
	10	0.463	0.568	0.498	0.359	0.487	0.438
	20	0.874	0.963	0.919	0.971	1.140	1.068
	40	1.771	1.943	1.827	3.071	4.187	3.825
Step Function	2	0.068	0.193	0.088	0.078	0.327	0.107
	5	0.146	0.262	0.177	0.172	0.242	0.217
	10	0.263	0.406	0.312	0.421	0.749	0.529
	20	0.158	0.590	0.546	1.490	1.817	1.482
	40	1.012	1.193	1.107	5.663	6.606	6.058
Trid10 Function	2	0.058	0.164	0.074	0.078	0.094	0.103
	5	0.145	0.193	0.160	0.172	0.258	0.215
	10	0.272	0.371	0.314	0.452	0.648	0.525
	20	0.534	0.654	0.597	1.560	2.122	1.73
	40	1.074	1.240	1.162	5.819	5.990	6.489
Zakharov Function	2	0.107	0.257	0.136	0.094	0.328	0.271
	5	0.236	0.309	0.264	0.203	0.390	0.236
	10	0.359	0.458	0.392	0.562	0.765	0.649
	20	0.942	1.078	1.014	1.763	1.997	1.910
	40	1.802	1.957	1.867	4.946	6.817	6.370
Rosenbrock Function	2	0.067	0.203	0.090	0.093	0.203	0.099
	5	0.163	0.289	0.216	0.183	0.225	0.212
	10	0.298	0.493	0.368	0.478	0.646	0.555
	20	0.582	0.780	0.575	1.411	1.723	1.627
	40	1.081	2.586	1.441	4.940	5.898	5.218

Table 4 gives the best, mean and worst values of the processing time in seconds. The elapsed time is measured as the time taken to run the algorithms from start to finish. Here we see that the bat algorithm converges to an optimization problem much faster than firefly algorithm. The difference in the best, mean and the worst times clearly indicates the winner.

V. CONCLUSIONS

In this paper, two swarm based algorithms were compared on accuracy and convergence speed parameters. Although there are improved versions of both the algorithms, we have used standard version only with set control parameters. It was observed that firefly algorithm gave better results to most of the test functions than bat algorithm. But the bat algorithm had a faster convergence speed than firefly algorithm. This conclusion is based on the six benchmark functions used in our experiment and the results may vary for some other set of benchmark functions.

ACKNOWLEDGMENT

Authors are thankful to the Director, MSIT for giving us the opportunity for doing our research work.

REFERENCES

[1] Beni, G., Wang, J. Swarm Intelligence in Cellular Robotic Systems, Proceed. NATO Advanced Workshop on Robots and Biological Systems, Tuscany, Italy, June 26–30 (1989).
[2] Boyd, Stephen P., Vandenberghe, Lieven, Convex Optimization (pdf), Cambridge University Press. p. 129. ISBN 978-0-521-83378-3.
[3] http://www.mathworks.in/help/optim/ug/unconstrained-nonlinear-optimization-algorithms.html.
[4] Er. A.K.Mishra, Dr. M. N. Das, Dr. T. C. Panda3, Swarm Intelligence Optimization: Editorial Survey, International Journal of Emerging Technology and Advanced Engineering, Volume 3, Issue 1, January 2013.
[5] D. Karaboga and B. Basturk, “On the performance of artificial bee colony (ABC) algorithm”, Applied Soft Computing 8(2008), pp. 687-697, 2008.
[6] Marsh, L. & Onof, C. (2007) "Stigmergic epistemology, stigmergic cognition." Cognitive Systems Research / doi: 10.1016/j.cogsys.2007.06.009.
[7] Bäck, Thomas (1995). Evolutionary algorithms in theory and practice : evolution strategies, evolutionary programming, genetic algorithms. Oxford: Oxford University Press. p. 328.
[8] Castor, Kevin (2006). "Hardware Testing and Benchmarking Methodology". Retrieved 2008-02-24.
[9] Yang, X. S., (2010) ,Firefly Algorithm, Stochastic Test Functions and Design Optimization, Int. J.Bio-Inspired Computation, Vol. 2, No. 2, pp.78–84.
[10] Xin-She Yang and Amir H. Gandomi, Bat Algorithm: A Novel Approach for Global Engineering Optimization, Engineering Computations, Vol. 29, Issue 5, 2012, pp.464 - 483.
[11] Marcin Molga, Czesław Smutnicki, Test functions for optimization needs, 3 kwietnia 2005.

