International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 10 (2015) pp. 27191-27204

http://www.ripublication.com

Performance Evaluation and Optimization of Wireless Sensor Networks Using Firefly Algorithm

Y.Chalapathi Rao
Associate Professor, Dept. of ECE, Anurag Engineering College
Kodad, Telangana, India, chalu.8421@gmail.com
Dr.Ch.Santhi Rani
Professor, Dept. of ECE, DMSSVH College of Engineering
Machilipatnam, AP, India, santhirani.ece@gmail.com

Abstract

The design challenges in Wireless Sensor Networks (WSNs) are to maximizing the network lifetime and power consumption. To answer this challenge, the key design is to effective action the sensor node at the optimal location and scheduling of those sensor nodes. In sensor troops, coverage improvement is one of the main issues. Maximum coverage ensures a better performance in WSNs. These are addressed and optimized through bio-inspired techniques namely Particle Swarm Optimization (PSO), Ant Colony Optimization Algorithm (ACO) and Firefly Algorithm (FFA). These are simple, effective and efficient optimization algorithms. The Firefly algorithm is one of the assorted character inspired algorithm that have been developed in the recent and it is inspired from the flashing behavior of the fireflies. The flashing behavior of the fireflies is to attract other fireflies in the cluster for mating. The low brighter firefly will be attracted by the brighter one. This process is imitated in the algorithm to find the solution of impartial function. In this paper, we evaluate the five multi-dimensional benchmark functions. The results of the simulation are acceptable showing the algorithm to have good performance abilities.

Keywords: ACO, Benchmark Functions, FFA, PSO, WSNs.

I. Introduction

Wireless Sensor Networks (WSNs) represent one of the most challenging areas in present days to monitoring industries, environments, hospitals, plants and etc. These networks are expected to be self governing, low-power effort, fact context and flexible. A final application may have many of sensor nodes spread out in an environment, making the deployment and the support of WSNs a complex task. A

WSN is to provide the users with access to the quality of arousing from the data gathered by spatially distributed sensors. The energy of the sensor nodes is usually supplied by battery with limited energy. The active wide applications of WSNs are lower energy consumption and to maximize the network lifetime [1].

A fundamental criterion for evaluating a WSN is the network lifetime, which is defined as the period that the network satisfies the application requirements. Since most devices of WSNs are powered by nonrenewable batteries, studies of maximizing the network lifetime have become one of the most important and demanding task in WSNs. In a WSN where devices are densely deployed, a subset of the devices can already address the coverage and connectivity issues. The rest of the devices can be switched to a sleep state for conserving energy, for that reason the lifetime of a WSN can be prolonged by planning the active intervals of devices, at every point during the network lifetime, the active devices must form a connected cover to fulfill sensing coverage and network connectivity. The devices in a WSN carry out both monitoring and communication issues. The monitoring requires devices to offer satisfying sensing coverage to the target. The communication requires devices to form a connected network for collecting and disseminating information via radio transmissions.

The problem of finding the maximum number of connected covers is difficult because each connected cover must fulfill sensing coverage and network connectivity simultaneously [2]. A common type of different WSNs is believed and a novel activity planning approach for minimizing power consumption and prolonging the network lifetime. The approach can be used in both cases of discrete point coverage and area coverage. We focus on the area coverage; the considered WSNs comprise two types of devices: sensors and sinks. The sensors monitor the target and transmit the monitoring results to the sinks. The sinks relay the monitoring results to the destination. Therefore, a connected covers in the WSNs must satisfy the following three constraints: 1) the sensors form complete coverage to the target. 2) All the monitoring results obtained by the sensors are transmitted to the sinks and 3) the sinks compose a connected wireless network. These three constraints interact with each other as the second constraint involves both sensors and sinks. Finding the maximum number of connected covers is thus more difficult than either the problem of maximizing the number of sensor subsets under the coverage constraint or the problem of maximizing the number of sink subsets under the connectivity constraint [3].

There are several ways of computing muli-dimentional benchmark functions. Bio-inspired algorithms prove to be effective for solving optimization problems. In this paper, we are evaluating the firefly algorithm (FFA) in terms of objective value with regards to the number of generations, and increasing dimensions using five benchmark functions.

The remainder of this paper is organized as follows: Section II discusses related work. Section III describes the firefly algorithm and the approach that has been followed. Section IV discusses the experimental results that have been obtained using the evaluation procedure implementation. Section V concludes this study with ideas for future work.

II. Related Work

An optimization problem is discovering the best solution from all feasible solutions. Classical methods of optimization are generally not used for their impracticality an intricate real life situation. They are generally deterministic in nature. Nature-inspired meta-heuristic algorithms such as Particle Swarm Optimization (PSO), Genetic Algorithm (GA), Ant Colony Optimization (ACO) and Firefly Algorithm (FFA) are most powerful algorithms for optimization. The goal is to develop more competent 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 imitating 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 [4].

First, meta-heuristics are fairly simple as they are inspired by very simple concepts. The inspirations are typically related to physical phenomena, animals' behaviours, or evolutionary concepts. 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.

Second, flexibility refers to the ease of applicability of meta-heuristics 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 inputs and outputs 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. The optimization process starts with random solutions 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.

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. The strength of standard firefly algorithm lies in the attractiveness of less brighter firefly towards the brighter firefly. The low 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 low brighter firefly can add features or attributes from the better firefly, it can converge to optima quickly in less number of iterations.

The heuristic algorithms are used to find solutions towards all possibilities, but they do not promise the exact one, so may be considered as approximately and not accurate algorithms, usually to find the fast and easy solution instead of exact one. The heuristic algorithms sometimes can be accurate and find the exact solution, but it is still heuristic until to get the exact the solution should be proved. The method of

scheme is used in heuristic algorithm is one of the familiar, such as avidity, but to be make easy and fast the algorithm by ignores the problem's demands. It may also used in conjunction with optimization algorithm to produce the optimized results.

Deterministic algorithms are almost all local search algorithms, and they are quite efficient and aim in finding the local optima. Each particle keeps track of its coordinates in the problem space which are associated with the best solution (fitness) it has achieved so far. The goal is to discover solutions that are not dominated by any other in the objective space of the optimization problems. The particle swarm optimization approach consists of, on each time step, varying the velocity of (accelerating) each particle toward its *pbest* and *gbest* locations [5]. Each swarm moves in the search space in a cooperative search procedure. Random term is weighted to acceleration to generate separate random numbers among *pbest* and g*best* locations. The swarms are set of particles for possible solutions of a group [6].

Particle Swarm optimization is a stochastic optimization algorithm introduced in the first half of 90's by Eberhart and Kennedy. PSO is inspired by birds flocking. The aim of the development was to simulate graphically the pattern of a flock of birds flying in tandem and execute swift direction changes. The initialization in PSO is similar to that of genetic algorithm; a random population is generated to represent the members of the swarm. The search for optimum goes on for generations by updating the swarm throughout the generations [7]. The particles fly through the solution based on their own best value and the swarm's best value.

PSO has been applied to a quality of problems including multi-objective problems in which more than one objective function is optimized simultaneously.

PSO based algorithm is used to locate the optimal sink position to the nodes to make the network is more efficient. PSO initializes with a cluster of random particles or solutions and searches for optima through updating of generations. PSO is more robust and easy to reach the solution for real world environmental monitoring and data aggregation problems.

Another important of the swarm intelligence algorithm is the Ant colony optimization algorithm. Ant colony optimization is a population based algorithm developed for combinatorial problems. It is inspired from the trail setting technique used by the ants when moving around in the search for food. The chemical deposit increases as the distance to the food decreases. This is imitated by using weights in computational environment. Software agents called artificial ants search for optimal solutions in a weighted graph. Local minimums are also avoided as the chemical left by the ants on their trail tend to evaporate quickly [8].

Ant colony optimization applications include scheduling, assignment and routing problems, thus making it more specific to routing and scheduling problems.

III. Firefly Algorithm (FFA)

A Firefly Algorithm (FFA) is a Meta heuristic algorithm inspired by the flashing behavior of fireflies. The first aim of firefly's flash is to act as a signal system to attract other fireflies. The firefly algorithm simulates the flash design and characteristics of fireflies. Most of fireflies produced short and metrical flashes and

have different flashing behavior. Fireflies use these flashes for communication and attracting the potential prey [9]. Firefly algorithm is powerful local search in which the brightness should be associated with the objective function but sometimes it may trap into several local optimums as result it cannot search globally well.

Formulation:

The attractiveness of different fireflies leads to the movement of firefly towards the other. This is the basic idea of the formulation of the algorithm. Luminescence or light intensity is defined as the amount of light energy transmitted and it varies with the distance. The attractiveness of the fireflies varies with the brightness which is in turn related to the objective function in the mathematical domain. The intensity decreases with the increase in distance, and hence, a given firefly will be attracted to a firefly that is close to it even though it is low brighter than a farther but brighter firefly. The intensity of light is known to vary inversely with the square of increasing distance or radius given by

$$I(r) = \frac{I_0}{r^2}$$

Where *I(r)* represent the light intensity as a function of distance and *r* is the radius. This can be converted to equality by adding a constant *I(s)*, which is intensity at the source. In real life conditions, every different form of energy is affected by atmospheric components, for example, sound traveling in a medium can be affected by the direction of wind and the amount of natural noise generated in the surrounding environment. Light is no exception to that and hence the intensity of light can also decrease depending on several environmental constraints. The intensity of light in real situations also depends on a factor called the absorption coefficient, and therefore, the inclusion of the absorption coefficient (α) changes the equation to:

$$I(r) = I_0 \exp(-\alpha.r^2)$$

Attractiveness of the firefly (Figure.1) is dependent and is directly proportional to the intensity of light. Hence the intensity equation can be transformed to represent the attractiveness as follows:

$$\beta = \beta_0 \exp(-\alpha.r^2)$$

Here β represents the attractiveness of the firefly and β_0 is the attractiveness at a radius 0. The above function, if necessary, can be approximated as

$$\beta = \frac{\beta_0}{1+\alpha.r^2}$$

The distance between any two fireflies is calculated using Cartesian distance method.

$$r_{a,b} = \sqrt{\sum_{k=1}^{d} (x_{a,k} - x_{b,k})^2}$$

Where $x_{a,k}$ is the k^{th} component of the spatial coordinate x_a of the a^{th} firefly, $x_{b,k}$ is the k^{th} component of the spatial coordinate x_b of the b^{th} firefly and *d* is the number of dimensions.

Firefly *a* is attracted to brighter firefly *b* and its movement is determined by

$$x_a = x_a + \beta_0 e^{-\alpha . r_{a,b}^2}(x_b - x_a) + \gamma \in$$

Where γ is randomization parameter.

Algorithm: FFA

1: Object of the function *f(X)*, where $X=(x_1,.....,x_d)$
2: Produce initial fireflies population;
3: Define and Compute intensity of light I;
4: Define absorption coefficient α;
5: **while** (*t<MaxProduction*)
6: **for** *a = 1* to n (all *n* fireflies);
7: **for** *b=1* to *n* (all *n* fireflies)
8: **if** ($I_b > I_a$), move firefly *a* towards *b*;
9: **end if**
10: Analyze new solutions and revise intensity of light;
11: **end for** *b*;
12. **end for** *a*;
13: Class the fireflies and know the current best;
14: **end while**;
15: Post proceeding results and visualization;
16: **end procedure**;

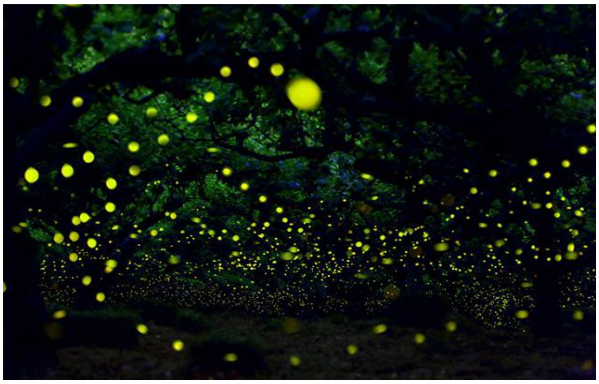

Figure 1: Fireflies

The firefly algorithm starts by initializing a population of fireflies and each firefly is different from the other in the swarm. The action is based on the brightness of the firefly. The brightness of the firefly is what ascertains the internal movement of the fireflies.

During the iterative process, the brightness of one firefly is compared with the others in the swarm and the difference in the brightness triggers the movement. The distance travelled depends on the attractiveness between the fireflies. During the iterative process the best solution thus far is continuously updated and the process goes on until certain stopping conditions are accepted. After the iterative process comes to a halt the best solution of the evaluation is determined and the post process is initiated to obtain the results. The flowchart diagram is shown in Figure 2.

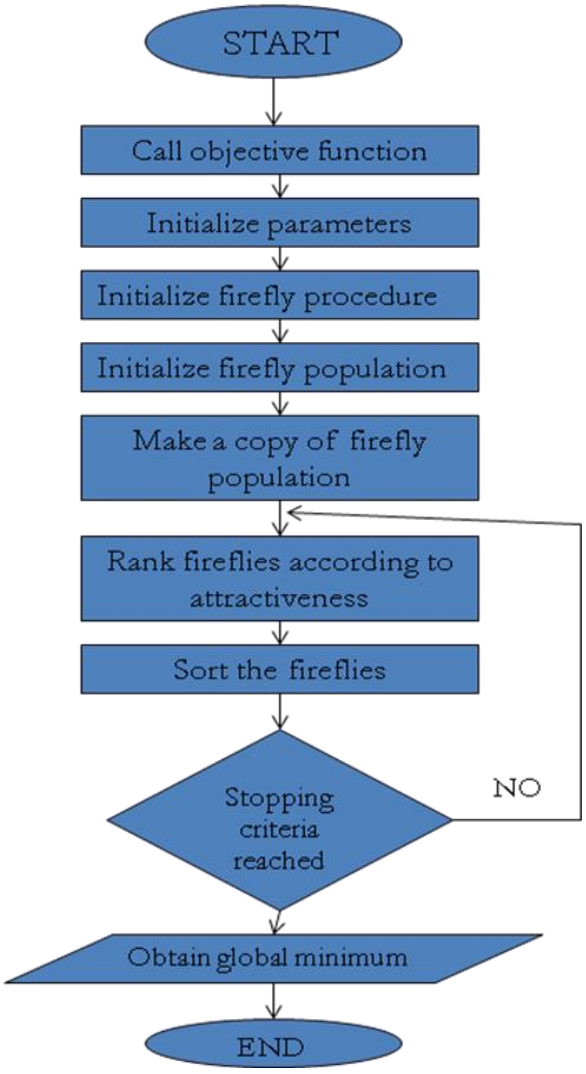

Figure 2: Firefly flowchart

IV. Experimental Results

Global optimization techniques are required to be formalized by a published set of benchmark problems. Benchmark functions are a way to test the optimization techniques [10].

Benchmark functions: There are four different types of benchmark functions designed for testing the global optimization problem [11].

1. Unimodal, convex, multidimensional: The set of functions designed for this class are simple to solve and have one global optimum. These functions can result in poor convergence to global optimum.
2. Multimodal and two dimensional functions with a few local optima: The set of functions under this class are of medium complexity and are serviceable to functions that have a few local optima and one global optimum.
3. Multimodal and two dimensional with many local optima: The set of functions under this class are more complex than the previous one as they are applicable to functions with large number of local optima.
4. Multimodal and multidimensional with many local optima: The functions under this set are more complex compared to the previous ones and are applied to intelligent optimization algorithms in combination with the previous class. Two dimensional problems are very rare in real world situations and hence multidimensional, multimodal benchmark problems are widespread in such situations.

Simple introductions to the functions that used to test the firefly algorithm have been defined with their mathematical definitions.

In this paper, five standard benchmark functions are used for testing the success of firefly algorithm, which are described in Table 1.

The simulations have been made with the following parameter setup for all functions. Population = 80 & Number of dimensions = 6. The number of iterations has been kept constant at 8,000 unless stated otherwise. The parameters alpha, beta, gamma and epsilon have been set at 0.1, 0.20, 0.25 and 0.001 respectively. The values of all the parameters remain constant unless stated otherwise.

Independent simulations have been run for each of the five functions and the following results have been collected. The outputs for 30 independent runs are: Best value, Worst value, Mean Value, Standard deviation of mean value and Time elapsed.

The firefly algorithm has been tested using five different benchmark functions via 4 different test cases. In the first test case is used to evaluate the best objective value for all functions over a fixed number of iterations. The second test case is used to evaluate the variation of the mean versus varying dimensions. The third test case is used to evaluate the variation of the mean versus varying iterations. The fourth test case is used to evaluate the variation of the best objective value versus varying iterations.

Table 1: Standard Benchmark Functions

Benchmark Function	**Formula**	**Dimen-sion (n)**	**Search Space Range**
Ackley	$f\left(\vec{x}\right) = -20.\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{a=1}^{n} x_a^2}\right) - \exp\left(\frac{1}{n}\sum_{a=1}^{n}\cos(2\pi x_i)\right) + 20 + e$	30	-32.768 to +32.768
DeJong	$f_0\left(\vec{x}\right) = \sum_{a=1}^{n} x_a^2$	30	-100 to +100
Michalewiz	$f\left(\vec{x}\right) = -\sum\left(\sin(x_a).\sin^{20}\left(\frac{ax_a^2}{\pi}\right)\right)$	10	0 to π
Rosenbrock	$f\left(\vec{x}\right) = 100\left(x_{a+1} + x_a^2\right)^2 + \left(-x_a\right)^2$	10	-2.048 to +2.048
Schwfel	$f\left(\vec{x}\right) = \sum_{a=1}^{n} -x_a \sin\left(\sqrt{\lvert x_a \rvert}\right)$	30	-500 to +500

Best objective value versus fixed number of iterations:

Table 2: Simulation Results for Standard Benchmark Functions

Function Name	**No. of Iterations**	**Best Value**	**Worst Value**	**Mean Value**	**Standard Deviation**	**Time elapsed (Sec.)**
Ackley	8000	0.00072	9.35306	2.25507	2.27691	470
De Jong	8000	0	0.00004	0.00001	0.00001	178
Michalewiz	8000	-2.65692	-4.68698	-4.10826	0.55496	186
Rosenbrock	8000	0.00774	3.99466	0.18517	0.77768	180
Schwfel	8000	0.00003	217.1397	91.69742	69.49743	102

Simulations performed by varying the problem dimension:

The results obtained by varying the problem dimension over a constant number of iterations. 30 independent repetitions have been performed for five functions and the mean of all the thirty independent runs has been calculated. Table 3 shows the collected results.

Table 3: Average Results For Function on Varying Dimension

Function Name	**Dim = 6**	**Dim = 8**	**Dim = 10**
Ackley	16.17964	17.8368	18.63649
DeJong	1.95E-05	4.04E-05	7.36E-05
Michalewiz	4.287659	6.095856	7.941764
Rosenbrock	0.164495	0.174542	0.343287
Schwfel	1360.356	1890.222	2630.118

Simulations performed by varying number of generations:
The results obtained by varying the number of iterations and recording the best value for one single independent run. The simulations have been performed on five different functions and the results obtained are illustrated in the figures below.

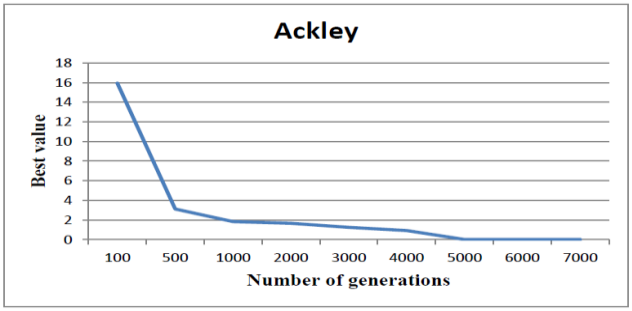

Figure 3: Variation of best value with number of generations for Ackley.

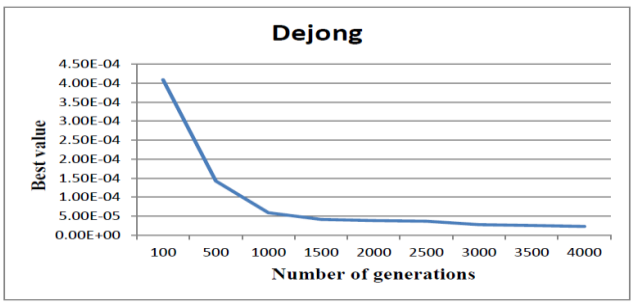

Figure 4: Variation of best value with number of generations for DeJong.

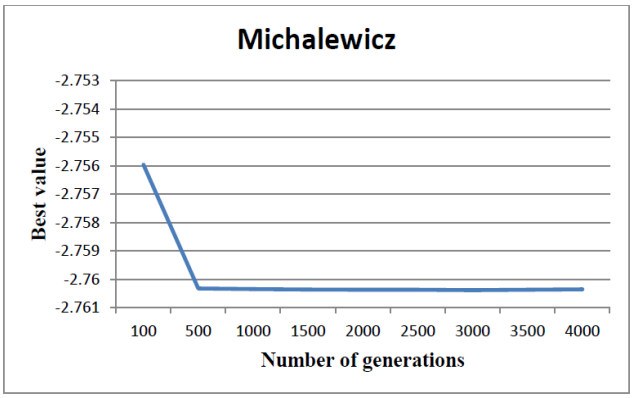

Figure 5: Variation of best value against number of generations for Michalewicz.

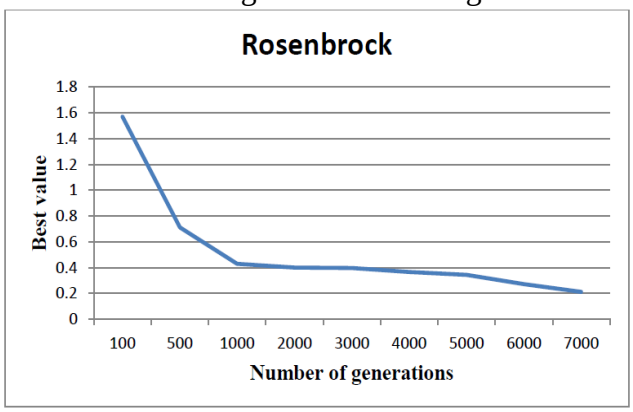

Figure 6: Variation of best value with number of generations for Rosenbrock.

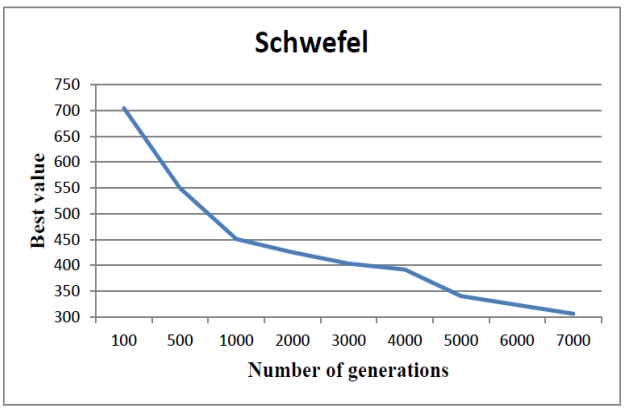

Figure 7: Variation of best value with number of generations for Schwefel.

Simulations performed by varying number of generations against the average value:

Five functions were evaluated by the firefly algorithm on a constant number of dimensions, by varying the number of iterations. Thirty independent runs for each

function have been made, the average over the 30 runs has been calculated and the results have been tabulated.

Table 4: Varying number of iterations and calculating average value

Function Name	**Generations =1,000**	**Generations =10,000**	**Generations =100,000**	**Generations =1,000,000**
Ackley	5.090657	4.266571	4.096618	4.055079
De Jong	0.000163	4.91E-05	1.68E-05	4.69E-06
Michalewiz	-2.73496	-2.74938	-2.73324	-2.73326
Rosenbrock	0.726725	0.420028	0.330758	0.238947
Schwfel	724.5558	560.4997	339.9028	183.3132

V. Conclusion and Future Work

The firefly algorithm uses the process of attraction based on the brightness in fireflies to optimize an objective function. In this paper, the firefly algorithm has been tested using five different benchmark functions both unimodal and multimodal in class. Three different test cases have been designed to test the algorithm using the functions.

Simulation experiments are conducted with three parameter sets, three parameter set of fireflies in domain and five benchmark functions. Better results are achieved in small number of fireflies in domain and smaller value of α and γ. Further research is needed to clarify effect of randomness and their effect on convergence. Better parameter sets probably await discovery in the outlined algorithm convergence domain.

References

[1] MdNafeesRahman, M A Matin, "Efficient Algorithm for Prolonging Network Lifetime of Wireless Sensor Networks,"*Tsinghua science and technology, pp561-568Vol. 16, No. 6*, December 2011, pp.561-568.

[2] J. Wang, R. Ghosh, and S. Das, "A survey on sensor localization," *Journal of Control Theory Appl.*, Springer,vol. 8, no. 1, 2010, pp. 2–11.

[3] S. Mini, Siba K. Udgata, and Samrat L. Sabat, "Sensor Deployment and Scheduling for Target Coverage Problem in Wireless Sensor Networks", *IEEE Sensors journal,* Vol. 14, no. 3, MARCH 2014,pp.636 – 644.

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

[5] Nor Azlina Ab. Aziz, Ammar W. Mohemmed, Mohamad Yusoff Alias, Kamarulzaman Ab. Aziz, Syabeela Syahali,"Coverage Maximization and Energy Conservation for Mobile Wireless Sensor Networks: A Two Phase Particle Swarm", IEEE,Sep 2011,pp.64-69.

[6] F. Van den Bergh and A. P. Engelbrecht, "A cooperative approach to particle swarm optimization," IEEE Trans. Evol. Comput., vol. 8, no. 3, 2004, pp. 225–239.

[7] J. Kennedy, R. C. Eberhart, "Particle swarm optimization", IEEE International Conference on Neural Networks, Piscataway, NJ. , 1995, pp. 942-1948.

[8] M. Dorigo, L.M. Gambardella (1997) Ant colony system: a cooperative learning approach to the traveling salesman problem, *IEEE Transaction on Evolutionary Computation* 1:53-66.

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

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

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

Y Chalapathi Rao was born in Krishna Dist. Andhra Pradesh, India, in 1984. He received B.Tech degree in ECE from Nagarjuna University, Guntur, India, M.Tech degree in VLSI System Design from JNTU, Kakinada, Andhra Pradesh and doing Ph.D. in Wireless Communications from Acharya Nagarjuna University (ANU), Guntur, India. He is an Associate Professor since 2014 at the Anurag Engineering College, Kodad, Telangana, India. His research interests are focused mainly in Wireless Sensor Networks, VLSI, and embedded systems design.

Professor Santhi Rani Chava received her B.Tech degree from Velagapudi Ramakrishna Siddhartha Engineering College, Nagarjuna University, Vijayawada, Andhra Pradesh, India. Her M.Tech degree was received from Jawaharlal Nehru Technological University College of Engineering, Ananthapur. She received Ph.D degree from Jawaharlal Nehru Technological University, Hyderabad. She had published around 24 papers in various national, international conferences and journals. She is currently working as a professor in Electronics and Communications Engineering Department, at D.M.S S.V.H College of Engineering, Machilipatnam, Andhra Pradesh, India. Her research areas include Smart Antennas, Low Power VSLI & Advanced Wireless Communications.