PERFORMANCE EVALUATION OF NATURE INSPIRED META-HEURISTIC ALGORITHMS USING ROSENBROCK, RASTRIGIN AND SPHERE TEST FUNCTION FOR OPTIMIZATION

[1]Deepika Dhawan, [2]Rajeshwar Singh
[1]Ph.D. Reseach Scholar,
[1]IKG Punjab Technical University, Jallandhar, Punjab, India.
[2]Doaba Khalsa Trust Group of Institutions, SBS Nagar, Punjab, India.

Abstract : Due to a large pool of optimization algorithms available, it is a daunting task for a researcher to select the best algorithm for their optimization problem. Nature inspired meta-heuristic algorithms have emerged and are very popular among researchers for their research. The meta-heuristic algorithm are used for finding and generating partial search algorithm that is supposed to provide a sufficiently good solution to any optimization problem especially with imperfect information. In this paper, we have chosen two widely researched and rapidly developing algorithms namely Artificial Bee Colony (ABC) and Firefly algorithm (FA) to test on different test functions. Therefore, this paper the efficiency of Artificial Bee Colony and Firefly algorithm are rigorously tested on three most used test function for optimization namely Rosenbrock test function, Rastrigin test function and Sphere test function. The best cost for both the algorithm on all three test functions are presented and compared to find which algorithm is best along with the test function for finding optimal solution to meta-heuristic implementation.

***Index Terms* - Nature-inspired, Meta-heuristic, Artificial Bee Colony, Firefly algorithm, Optimization.**

I. INTRODUCTION

Computer optimization algorithms which are used to solve optimization problems can be roughly categorised into two parts: Exact algorithms or deterministic algorithm and heuristics algorithms. The Exact algorithms yield the solutions in a determined time and does not put a load on the computation complexity, they yield a solution in a finite time. Whereas, Heuristic algorithms are used to solve problems which are NP-Hard problems. The solution to these problems are not found in a deterministic time, hence the name NP-Hard where NP stands for Non deterministic Polynomial. In this category the time required for a problem to be solved cannot be guaranteed. Also these algorithms find the solution which are "good" in a reasonable amount of time. Heuristic algorithms are very specific for a problem, whereas, Meta-heuristic is a high language framework which provides guidelines to develop a heuristic optimization problem. Meta-heuristic algorithms are used for finding and generating partial search algorithms which provide solutions to mostly imperfect information or limited capacity of computation. Unlike heuristic algorithms meta-heuristic algorithm do not have to be fully modified to work on any other optimization problem, they are hardly needed to modify to execute on other optimization problems. In meta-heuristic algorithms there are many metaphor based algorithms that are used to solve optimization problem in computer science. To list a few, Simulated Annealing (SA), Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), Artificial Bee Colony (ABC), Firefly Algorithm (FA) etc.

In this research paper we are using the two of the above stated metaphor based metaheuristic to evaluate on various test function for optimization as they are widely used among researcher for their research. The purpose of this paper is to evaluate the algorithms on various test function so that we can compare the results (best cost) of the algorithms and we can infer from the results that which algorithm would behave good on certain test function.

For this purpose we have taken Artificial Bee Colony (ABC) [1] and Firefly Algorithm (FA) [2], as they have gained some serious attention of researchers in the past few years [3, 4, 5, 6, 7]. Artificial Bee Colony is a metaphor based metaheuristic algorithm which mimics the behaviour on honey bee swarm. It was proposed by Derviş Karaboğain the year 2005 [1]. The Firefly Algorithm is also a metaphor based metaheuristic algorithm which mimics the behaviour of firefly swarm and the way they act is that the firefly's flash acts as a signal system to attract other fireflies. This algorithm was formulated by Xin-She Yang in the year 2008 [2].

II. RELATED WORK

In paper [8], authors have introduced a new algorithm to tackle multicast QoS problem. The key point of FA algorithm is to compute most feasible path which successfully full-fills the requirements of the system and helps to find the optimal multicast tree. The simulation results show the efficacy of the algorithm are drawing attention towards Multicast routing problem which occurs in many multimedia applications and it is really very difficult to find a solution for QoS multicast routing problem because it is nonlinear combination optimization problem.

In paper [9], authors are presenting a brilliant approach which is based on ABC to overcome challenges of routing. The artificial bees are used to alter the position of food sources in order to discover new food places. Authors are using Fitness function to calculate the food source with highest nectar value. The food source with highest nectar value is considered as the solution.

The authors in paper [10] are discussing on improvements which can be implemented with Artificial Bee Colony Algorithm to solve Vehicle Routing Problem with Time Window. Authors are proposing improvements over weak search ability and slow search speed of Artificial Bee Colony Algorithm. First of all the single search mode is changed into a three-way search method, which enhances the optimization depth of the algorithm. Secondly, by using multiple neighbour-hood searches of new food sources has enhanced the survival of new food sources and increased the diversity of populations. Next to it is the global optimal solution which is recorded by setting and updating the bulletin board. The simulation results show that the improved discrete ABC algorithm has multifarious advantages in solving large-scale problems.

In paper [11], authors are presenting modified Firefly algorithm based on Firefly algorithm and improved particle swarm optimization. Firefly algorithm falls under the category of nature enthused algorithm of swarm intelligence, it means the outcomes of algorithm depends upon the response of a firefly to the light of other fireflies. In order to enhance the searching behaviour of standard algorithm the improved velocity concept of particle swarm optimization is used in modified algorithm.

The paper [12] focuses on Function Optimization (FO) which is the well-studied continuous optimization task to find best suited parameter values to get optimal value of a function. Nature Inspired Algorithm (NIA) become famous to solve FO. In this paper, authors are simply comparing the results of different NIAs by using simulation on standard benchmark functions.

The paper [13], attracts attention towards two major issues in multicast routing, one is load balancing and other one is transmission delay. Authors are taking bi-objective optimization challenges for network routing using multi-cast, where the average bandwidth utilization ratio and the average transmission delay are two objectives for minimization. Authors compare outcomes based on Pareto dominance, which helps to enhance the local exploitation.

III. ABC META-HEURISTIC [1]

It is a process that matches the activities of honey bees, as the these bees collectively accomplish their task through social cooperation. In the ABC algorithm there are three types of bees: employed bees, onlooker bees and scout bees. The job of the employed bees is to search food around the food source.Also it is assumed that there is only one artificial employed bee for each food source, i.e. the number of employed bees in the colony is equal to the number of food sources [3, 4]. Then the information for the food source is then forwarded onto the onlooker bees. The onlooker bees then select the higher quality (fitness) food source rather than the lower quality ones. The scout bees are the employed bees who have abandoned their food source and search for a new one, once they find a new food source they memorize the location of the new food source and forget the old food source's location. The number of onlooker bees and the employed bees is equal to the number of solution in the swarm. The artificial bee colony generates a randomly distributed initial population (which means the bees in simulation are placed randomly, it may or may not be near the food source) of SN food sources (solution).

The process for the beesis to exchange all the information about the food sources to the other bees is that when the employed bees go to their food source they roam around in the area near the food source, the onlooker bees then watches the employed bee flying and dancing near the food source and get the information by co-operation that food is present at that source, and then chooses the food source from the probability value $p_i$associated with it, which is calculated by the following expression $P_i = fit_i / \sum fit_n$, where value of n is1 to SN (Number of food source), fit_i is the fitness value and i is evaluated by its employed bee

ABC Algorithms steps are:

1. Food sources are initialised for each employed bee
2. Repeat
    a. Every employed bee goes to the food source they memorized, then evaluates its nectar amount and dances in the hive.
    b. Each onlooker bee watches the dance of the employed bee and chooses the source depending on the dance, and then goes to the source, and evaluates its nectar amount.
    c. The abandoned food sources' left permanently and the scout bees find a new food source.
    d. The best food source found so far is registered, i.e. the new food source should have high quality than the previous one.
3. Until requirements are met.

3.1 Mathematically ABC Defined

Mathematically the working of ABC algorithm can be written as, in the case of employed bees who are searching around the food resources at x_i will search for a better food resource at new location v_i. the identification of the new food source will evaluated by the equation (1) [6]

$v_{ij} = x_{ij} + \Phi_{ij}(x_{ij} - x_{kj})$ j = 1,2,...,n; k = 1,2,...,SN (1)

The new location vector of the bees is $v_i = [v_{i1}, v_{i2}, \ldots\ldots, v_{in}]$, $x_i = [x_{i1}, x_{i2}, \ldots\ldots, x_{in}]$ is the location vector of the ith bee, k where k not equal to j is a correct random number in [1, SN] and Φ_{ij}a random number uniformly distributed between [-1, 1] The selection of the $x_{ij}$is done by the equation (2) [6]

$x_{ij}= L_j + rand(0,1) \times (U_j - L_j)$ (2)

The $U_j$and L_j are the upper bound and lower bound in equation (2). When a new location of the food source is found, its optimization/befitting rate is calculated by the equation (3). [6]

$Fit_i = 1/1+f_i$, if $f_i \geq 0$ else,

$Fit_i = 1+abs(f_i)$ (3)

In the case of onlooker bees, the selection of the food source is calculated by equation (4) [6]

$P_i = fit_i / \sum fit_n$

(4)

3.2 Parameters during implementation

We have implemented the meta-heuristic in MATLAB. In the implementation of the ABC algorithm we have set the number of maximum iterations to 50, population size to 50.

The other value needed is the co-efficient for Acceleration, we have taken it as 1, and the abandonment limit parameter is evaluated by (0.6*Decision Variable*population size). The code for Ant Bee Colony is as appended below:

Objective function f(x), $x=(x^1, x^2,....., x^n)^T$
Encode f(x) into virtual nectar levels.
Define dance routine (strength, direction) or protocol.
While(condition)
    For loop over all n dimensions
        Generate new solutions and evaluate the best solution found so far
    end for
Communicate and update the optimal solution set.
end while
Decode and output the best results

IV. FIREFLY ALGORITHM

Firefly Algorithm [2] is an algorithm which mimics the behaviour of the fireflies. As the ABC this algorithm is also metaphor based. It is a metaheuristic algorithm which is inspired by the flashing of the light from fireflies. The main purpose of the firefly's flash is to act as a signal system to attract other fireflies. This algorithm simulates the flash pattern of a firefly. The fireflies make short and rhythmic flashes and have different behaviour of flashing. They use flashing as a signal to attract other mates and also search for prey. This algorithm has brightness associated with the objective function, but sometimes it gets trapped into the local optimums. As a result it may not perform best all the time. There are two parameters of the algorithm, first is attractiveness coefficient and the other is randomization coefficient. Both play a crucial role in determining the optimal solution in the search space, as these values are used in calculating the speed of the convergence and the behaviour of the algorithm. The algorithm's parameters shall not change during iterations/execution of the code. This algorithm was developed by Xin-She Yang in the year 2008 at Cambridge. While formulating the algorithm the following assumptions were made:

1. All Fireflies are unisexual, so that any single firefly will be attracted to all other fireflies.
2. The attractiveness of the fireflies is directly proportional to the brightness of the fireflies' light, i.e. the more rhythmic light's intensity is the more other fireflies will be attracted to it.
3. The less bright firefly will be attracted to and will move towards the brighter one. However the intensity of the light decreases as the space between the two fireflies increases, i.e. if the brighter firefly is flying away from another firefly the intensity of the light will decrease.
4. If in case there are no firefly brighter than a given firefly, then it will move randomly.

We can reach an optimum result using the above stated assumptions.

4.1 FA defined mathematically

The two factors in the Firefly algorithm are change in luminous density and creation of attraction for other fire flies. That is to say that a firefly with high/low intensity will attract other firefly with high low intensity or the low light fireflies get attracted to other high light fireflies and this process continues until all of the fireflies have gathered in one place, which is supposed to be a global optimum. The distance between the fireflies i and j are denoted by r_{ij}, in addition the light intensity decreases as the distance increases. The light intensity I(r) changes according to the inverse square law relation. As mentioned above luminous density is in proportion to creation of attraction to other fireflies. Therefore, we can write

$\beta = \beta_0 e^{-\gamma r2}$ (5)

In equation (5)γ is the light absorption coefficient and β_0 is the attractiveness at r = 0.

Also r_{ij} is the Euclid distance between two fireflies, which is evaluated as shown in equation (6)

$$r_{ij} = \|x_i - x_j\| = \sqrt{\sum_{k=1}^{n}(x_{ik} - x_{jk})^2} \qquad (6)$$

The update of the movements of the fireflies following the above stated procedure takes place using equation (7)

$x_i \leftarrow x_i + \beta = \beta_0 e^{-\gamma r2} (xj - xi) + a(rand - 1/2)$ (7)

In equation (7) α, β_0 and γ are considered constant, α, β_0 select from [0, 1] and γ is selected from [0, ∞]. [7]

4.2 Implementation of Firefly Algorithms

Firefly Algorithms is implemented as described in the following paragraphs. Maximum iterations is set to 100, population size is set to 50, light absorption coefficient is set to 1, attraction coefficient is set to 2, mutation coefficient is set to 0.2, mutation coefficient damping ratio is set to 0.98 and the uniform mutation range is defined as 0.8 * (variable upper bound – variable lower bound). Therefore, Firefly Algorithm can be written as appended below:

Begin:

i. Objective function: f(x) = x = [x1, x2, …… xn]
ii. Generate initial population of fireflies x_i, where i = 1,2 3…n
iii. Formulate light intensity that is associated with the objective function
iv. Define absorption coefficient γ.

```
While (t <maxGeneration)
    For i =1:n (for all n fireflies)
        For j = 1:i (n fireflies)
            if (Ij> Ii),
                Vary attractiveness with the distance r via e^(-γr2)
                Move firefly i towards j
    Evaluate new solution and update light intensity
```

```
                End if
            End for j
        End for i
        Rank fireflies and find the current best.
        End while
        End
```

V. VARIOUS TEST FUNCTION FOR OPTIMIZATION

Test functions in mathematics and computer science are used useful for evaluation of characteristics of optimization algorithms such as convergence rate, precision, robustness. In this part we have compared the algorithms on three different test function first is Rosenbrock function, it is a non-convex function used to evaluate the performance of the optimization algorithm. Next is the sphere function which is uni-modal, has d dimensions, it is also used to evaluate the performance of the optimization algorithm. The last function we have used is the Rastrigin function; this function is non-convex and is also used to evaluate the performance of optimization algorithm.

5.1 Rosenbrock Test function

The Rosenbrock is a non-convex function, which is used for performance test problem for optimization algorithm. It was proposed by Howard H. Rosenbrock. This function is also known as Rosenbrock valley or Rosenbrock banana function. The function is defined by f(x) = $\sum_{i=1}^{N-1}[100\,(x_{i+1}-x^2)^2+(1-x_i)^2$.A sample plot of the Rosenbrock function is shown in Fig.1.

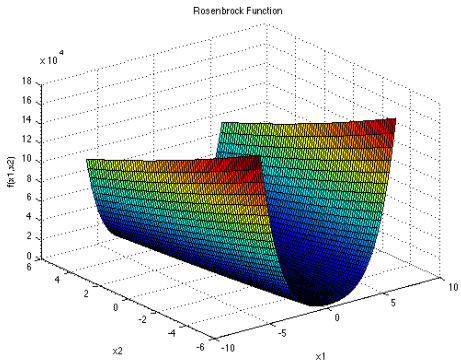

Fig.1: Rosenbrock Function

5.2 SphereTest function:

These Test functions have dimensions d and it is local value except for the global one. It is defined by the formula f(x) = $\sum_{i=1}^{d} {x_i}^2$. A sample plot of the function is shown in Fig.2

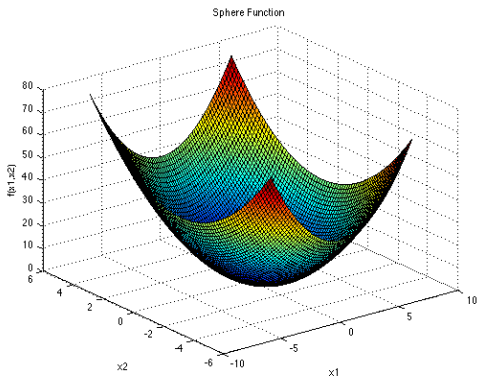

Fig.2: Sphere Test Function

5.3 Rastrigin Test function

Rastrigin function is non-convex function which is used for performance testing in optimization algorithms. It is an example of non-linear multimodal function. Proposed by Rastrigin as a 2 dimensional function, later it was generalised by others. It is defined by f(x) = An+ $\sum_{i=1}^{n}[{x_i}^2 - Acos(2\pi x_i)]$. A sample plot of the function is shown in Fig.3.

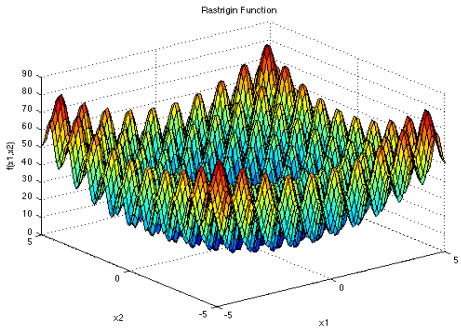

Fig.3: Rastrigin Test Function

VI. EVALUATION AND RESULTS

ABC and Firefly are tested using the Rosenbrock, Sphere and the Rastrigin test function. The implementation of the algorithms are very well explained above, the parameters used are also stated along with the implementation. The test function are briefly described above along with the figures to have an idea of the plot of the function. Now we are going to implement the test functions in MATLAB.

The Test function code in MATLAB is implemented as follows:

Rosenbrock test function:

```
function z=Rosenbrock(x)
    n=numel(x);
    z=sum((1-x(1:n-1)).^2)+100*sum((x(2:n)-x(1:n-1).^2).^2);
end
```

Sphere function:

```
function z=Sphere(x)
     z=sum(x.^2);
 end
```

Rastrigin function:

```
function f = rastriginfcn(x)
    n = size(x, 2);
    A = 10;
    f = (A * n) + (sum(x .^2 - A * cos(2 * pi * x), 2));
end
```

ABC and FA algorithms are very delicate because of their parameters and any change in the parameter can affect the outcome of the algorithm, so, the optimal parameters are set in order to get the optimal result. In the evaluation we have tested the algorithms after setting the maximum iteration and population/swarm size to 100 and 50. The population is a constant as we have set it to 50 but the iterations are incremented with 10 each time i.e. first it is simulated on 10, then 20, then 30 and so on up until 100.

However we have presented the output data from 50 to 100 iteration so that our data looks clean

Execution with test function set to Rosenbrock:

Upon execution of algorithms with Rosenbrock test function, the data is retrieved is as appended in table.1.

Table.1: Rosenbrock Test Function Convergence data for ABC and FA.

Algorithms	**Iterations**					
	50	60	70	80	90	100
ABC	1.6268	1.6411	1.2016	1.5757	1.5264	1.4621
FA	0.1874	0.0881	0.1422	0.0662	0.07576	0.0425

The convergence graphs with maximum iterations for both the algorithms areas shown in Fig.4(a) and Fig.4(b).

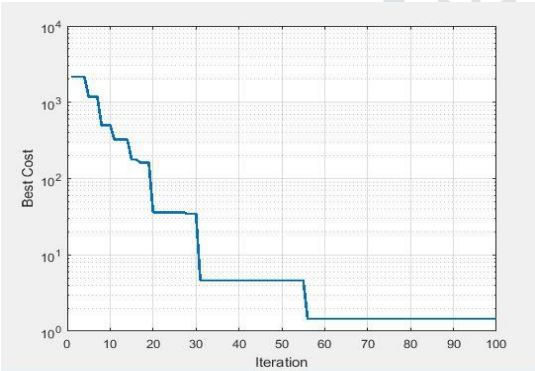

Fig.4(a): ABC Algorithms.

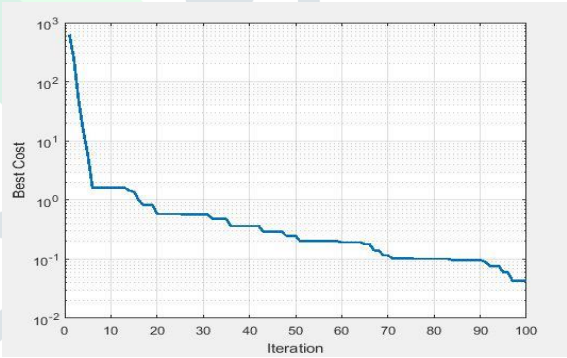

Fig.4(b): Firefly Algorithms.

Comparison diagram from the data of table.1 is shown in fig.4(c).

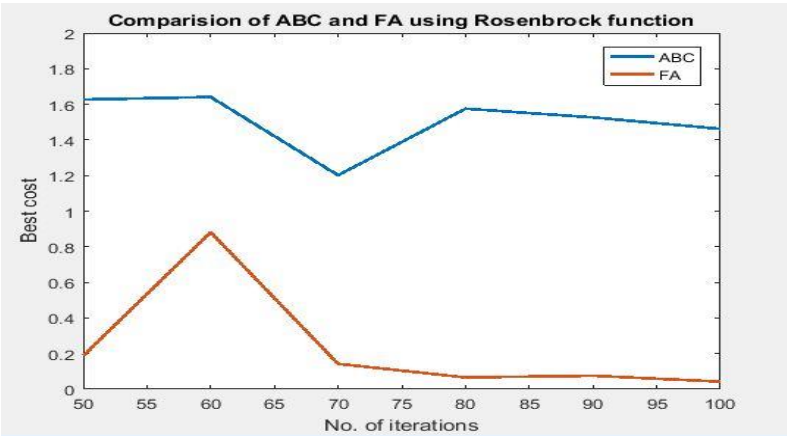

Fig.4(c): Comparison of ABC and FA using Rosenbrcok Function

Execution with test function set to Sphere:

Upon execution of algorithms with Sphere test function, the data obtained is as appended in table.2.

Table.2: Sphere Test Function Convergence data for ABC and FA.

Algorithms	**Iterations**					
	50	60	70	80	90	100
ABC	$3.0144e^{-5}$	$7.5238e^{-7}$	$1.7323e^{-7}$	$8.5798e^{-9}$	$2.2461e^{-9}$	$4.0359e^{-10}$
FA	$2.9248e^{-4}$	$2.3697e^{-4}$	$1.1092e^{-4}$	$8.5121e^{-5}$	$5.0446e^{-5}$	$4.1717e^{-5}$

The convergence graph with maximum iterations for both the algorithms are in Fig.5(a) and Fig.5(b).

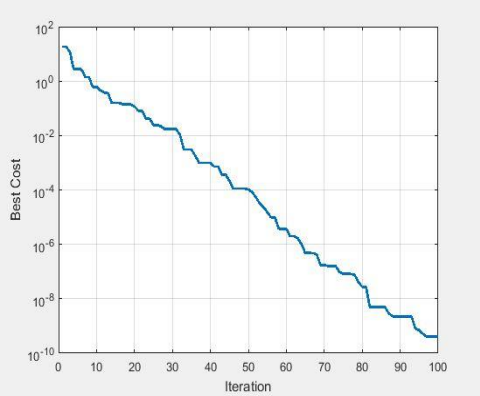

Fig.5(a): ABC Algorithms.

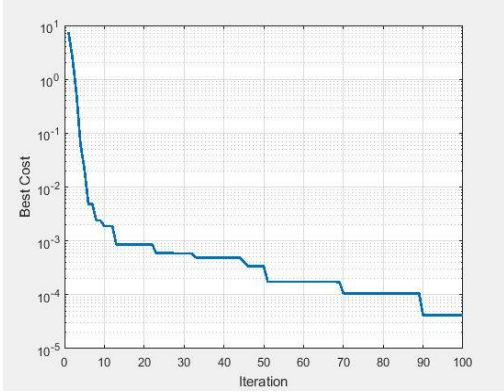

Fig.5(b): Firefly Algorithms.

Fig.5(a) represents the convergence graph of ABC algorithm and fig.5(b) represents the convergence graph of Firefly Algorithm.Comparison of ABC and FA using Sphere Test Function is carried out for data of table.2 and is as shown in fig.5(c).

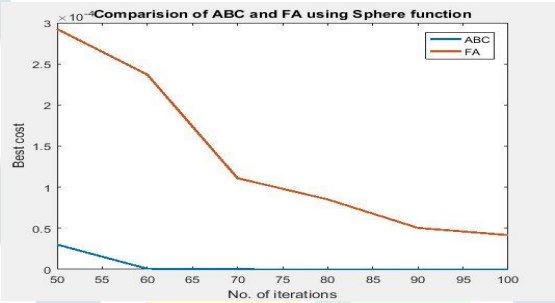

Fig.5(c): Comparison of ABC and FA using Sphare Test Function

Execution with test function set to Rastrigin:

Upon execution of algorithms with Rastrigin test function, the data obtained is as appended in table.3.

Table.3: Rastrigin Test Function Convergence data for ABC Algorithm and FA.

Algorithms	**Iterations**					
	50	60	70	80	90	100
ABC	3.9224	4.1649	3.7726	3.9081	6.3575	4.2936
FA	1.0284	0.1919	0.0246	0.0123	0.0084	0.0037

The convergence graph with maximum iterations for both the algorithms are in Fig.6(a) and Fig.6(b)

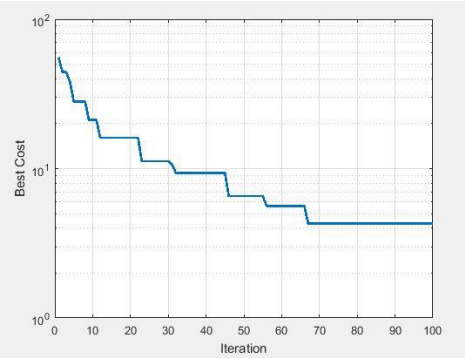

Fig.6(a): ABC Algorithms

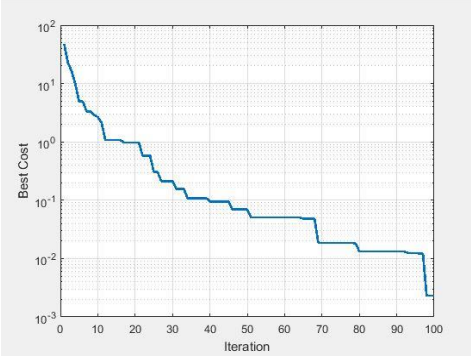

Fig.6(b): Firefly Algorithms.

The data obtained for Rastrigin Test Function is as shown in table 3. The comparison is carried out and graph obtained is as shown in Fig.6(c).

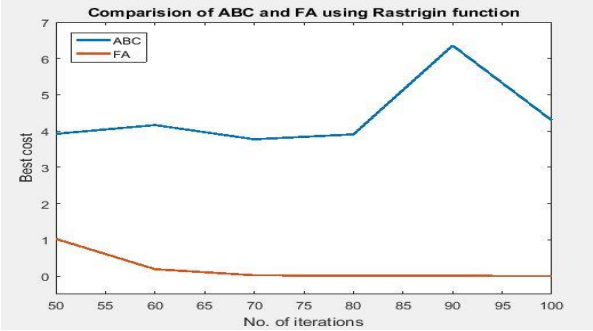

Fig.6(c): Comparison of ABC and FA using Rastrigin Test Function

VII. CONCLUSION

Performance evaluation of nature inspired ABC Algorithm and Firefly Algorithmhas been carried out on three test functions Sphere, Rastrigin and Rosenbrock. It is found that with the rosenbrock test function, Firefly algorithm has yielded a much better cost than the Artificial Bee Colony algorithm. In the convergence diagram we can see that the FA's plot has decreased rapidly approximately up to 0.05, whereas, the cost yielded by the ABC is near 1.2, which reveals that the FA performs better in comparison when using the rosenbrock test function. The plots and convergence diagram of ABC and FA using the sphere test function reveal that the ABC algorithm has proven to be the best in this particular case, as it has revealed the best cost that is much more impressive than the FA algorithm has revealed. We can see in the convergence diagrams (5 (a) and 5(b)) that initially FA algorithm was starting to converge faster but then it could not converge as much as the ABC. And in the graph 5(c), we can see that near to iteration 70 the FA's cost risen up. Hence we can be sure and say that the ABC algorithm will reveal better results when using with Sphere test function.The plot 6(c) and convergence diagram 6(a) and 6(b) of ABC and FA using the rastrigin test function reveal that the FA algorithm gives better results as compared to ABC using rastrigin test function. The FA algorithm gives the best cost around 0.00038 while the ABC gives 3.78. Hence we can be sure that the FA algorithm performs good using the rastrigin function in comparison with the ABC algorithm.

REFERENCES:

[1] *Karaboğa, Derviş,*"An Idea Based on Honey Bee Swarm For Numerical Optimization" *, 2005.*
[2] Yang, X. S., "Nature-Inspired Metaheuristic Algorithms"..2008.
[3]S. Nakrani and C. Tovey, "On honey bees and dynamic server allocation in internet hosting centres, Adaptive Behaviour", 12, pp 233-240, 2004.
[4]WengKee Wong, "Nature-Inspired Metaheuristic Algorithms for Generating Optimal Experimental Designs", 2011.
[5] DervisKaraboga · BahriyeBasturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm", Springer April, 13, 2007.
[6] W. Xiang, M.Q. An, "An Efficient and Robust Artificial Bee Colony Algorithm for Numerical Optimization", Computers & Operations Research", Vol. 40, pp. 1256-1265, Elsevier Ltd, 2013.
[7]: P.R. Srivatsava, B. Mallikarjun, X.S. Yang, "Optimal Test Sequence Generation using Firefly Algorithm", Swarm and Evolutionary Computation, Vol. 8, Elsevier B.V, 2013, pp. 44-53.
[8] Jie Yuan, Yafei Tian, Shan Wang, Changnian Liu "QoS Multicast Routing Based on FireflyAlgorithm,"IEEE Computer Society, 2013, pp. 160-163.
[9] Wei Zheng, Di Luo, "Routing in wireless sensor network using Artificial Bee Colony algorithm," IEEE International Conference on Wireless Communication and Sensor Networks, 2014, pp. 280-284.
[10] Cong Chen, Kang Zhou,"Application of Artificial Bee Colony Algorithm in Vehicle Routing Problem with Time Windows," IEEE International Conference on Sensing, Diagnostics, Prognostics and Control, 2018, pp. 781-785.
[11] Shubhendu Kumar Sarangi, Rutuparna Panda,"A New Modified Firefly Algorithm for Function Optimization," IEEE International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), 2016, pp. 2944-2949.
[12] Md. Julfikar Islam, Md. Siddiqur Rahman Tanveer and M. A. H. Akhand, "A Comparative Study on Prominent Nature Inspired Algorithms for Function Optimization," IEEE International Conference on Informatics, Electronics and Vision, 2016, pp. 803-808.
[13] Huanlai Xing, Fuhong Song, Lianshan Yan, Wei Pan," On Multicast Routing With Network Coding: A Multiobjective Artificial Bee Colony Algorithm," Communication Theories and Systems,2019, pp.160-176.