International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 02, February-2022

Optimization Performance Analysis of Firefly Algorithm using Standard Benchmark Functions

Sandeep Dhawan
Senior Director of IT
Food Authority
New York, USA.

Abstract— **Optimization is necessary for ensuring the quality of every software system. While solving optimization problems, it's a challenge to find optimal solutions from a huge number of solutions. Various types of optimization techniques have been developed to reduce complexities in optimization. Metaheuristic algorithms have been very effective in handling optimization problems. There are many efficient metaheuristic algorithms for solving optimization problems. However, for solving different optimization problems, different algorithms are suitable. It's important to evaluate every metaheuristic algorithm's performance. Optimization efficiencies of many metaheuristic algorithms haven't been fully analyzed yet. Therefore, this paper has evaluated the optimization performance of a slightly adapted Firefly Algorithm against some existing metaheuristic strategies like Bat Algorithm, Bacteria Foraging Algorithm, and Cuckoo Search Algorithm. The evaluation has been done using 11 benchmark functions. The results after the implementation have shown that Firefly Algorithm performs competitively against other metaheuristic algorithms in terms of finding optimal solutions.**

Keywords— Optimization, metaheuristic algorithms, Firefly Algorithm, optimal solutions.

I. INTRODUCTION

Optimization is a procedure for determining the most optimal solutions to make a system as effective and functional as possible by maximizing or minimizing the involved parameters in the problems. The field of optimization is huge and full of complexities. There have been investigations on optimization using metaheuristic algorithms for many years. As time goes on, more advanced software systems are being developed, and so complexities in optimization keep increasing. Optimizing a large and complex software system is quite a challenging task as the system will need testing a large number of complex solutions. As complexities in optimization keep increasing [1], many existing metaheuristic algorithms may struggle to handle these problems. Therefore, adaptations of metaheuristic algorithms and identifications of new algorithms [2] are common practices in order to make optimization simpler.

In optimization, there have been implementations of many metaheuristic algorithms like Genetic Algorithm (GA) [3], Simulated Annealing (SA) [4], Flower Pollination Algorithm (FPA) [5], Particle Swarm Optimization (PSO) [6], Hill Climbing (HC) [7], Cuckoo Search (CS) Algorithm [8], Ant Colony Algorithm (ACA) [9], Tabu Search (TC) [10], Bee Algorithm [11], Jaya Algorithm (JA) [12], Bacterial Foraging Algorithm (BFA) [13], Bat Algorithm (BA) [14], Harmony Search (HS) Algorithm [15], African Buffalo Optimization (ABO) Algorithm [16], etc. Optimization experts often get confused while making a decision for using suitable metaheuristic algorithms for solving their optimization problems. The efficiency of an algorithm depends on various factors like the ability to control local and global optima [17], computational heaviness, convergence rate [18], etc.

Some algorithms are great in handling local and global optima, but they aren't computationally light. GA efficiently finds optimal solutions in optimization, but GA is computationally heavy [19], so if someone looks for a computationally light algorithm, GA may not be very suitable. Again, HC is computationally light, but HC often struggles while handling local optima [20]. Also, some algorithms have simple implementation procedures, but they struggle to avoid local optima. For every metaheuristic algorithm, it's important to keep a balanced ratio between exploitation and exploration. Failing to handle exploration and exploitation results in bad optimization performance. However, convergence rate towards the global optima is also important to ensure a high optimization performance. All these factors matter for every metaheuristic algorithm during any optimization process. Each of the most metaheuristic algorithms shows strong performance in terms of some of these factors and struggles in terms of other factors. So, it's essential to evaluate the robustness of metaheuristic algorithms based on different factors.

As there's a necessity to analyze the characteristics of every metaheuristic algorithm in order to make optimization processes easier, this paper has analyzed the performance of the Firefly Algorithm (FA) using some benchmark functions. Some adaptations have been made to improve the performance of the original FA. FA's performance has been compared with other metaheuristic algorithms like BFA, BA, and CS. In the evaluation process, the main consideration has been the ability to find optimal solutions in optimization.

The rest of the paper is arranged as follows. Section 2 will discuss the related works of FA in the field of optimization. Section 3 will discuss the characteristics of FA and how it works for finding optimal solutions. Section 4 will discuss the working procedure through which FA finds optimal solutions from the benchmark functions. Section 5 will discuss the obtained results of FA from its implementation and compare the results with the results of other metaheuristic algorithms. Section 6 will have a discussion on the obtained results. Finally, section 7 will conclude this research work.

II. RELATED WORKS

FA was developed in 2008 [21]. Since then, there have been some implementations of FA in optimization. There have been implementations of FA for multimodal optimization. In multimodal optimization, there have been comparisons among FA and other metaheuristic algorithms like PSO. Results and simulations have indicated the superiority of FA against many existing metaheuristic algorithms [22]. In multimodal optimization, FA has potential for future applications.

FA is useful in optimizing machining parameters (spindle speed, depth of cut, feed rate, etc.). A hybridization of FA with PSO discovered optimal solutions by exploring search spaces. The hybrid FA used selected objective functions for machining parameter optimization in turning operations. FA also estimated the parameters of the machining cutting and led to reduced surface roughness, and it was validated by the ANOVA test [23].

FA has also been successful in mobile robot navigating applications. Luminosity and brightness are major variables in mobile navigation. An optimized FA based strategy was developed for obstacle avoidance using co-in-centric sphere-based geometrical strategy and optimum path generation. This strategy comprises the optimum path finding to objective function as well as constraints to obstacles and paths as the algebraic-geometry co-relation function. While compared with other existing approaches, FA has been found to be efficient for mobile robot navigation [24].

FA has been successfully implemented for scheduling optimization of heating, cooling, and power with many objectives. According to the recent multi-system optimization strategy of combined heating, cooling, and power, an analysis of limiting factors like energy utilization, cost, environmental protection, etc., was conducted. FA was implemented for single-target searching. The implementation results showed that the search speed was improved due to using FA. Also, FA was useful in finding an efficient scheduling scheme of heating, cooling, and power system with many targets in a short time [25]. The application of FA was very beneficial to solve system optimization problems.

The exploration to exploitation ratio is significant considering the performance of every metaheuristic algorithm in optimization. It has been possible to use a fuzzy system for the efficient and dynamic tuning of the parameters of FA so that exploration and exploitation stay in balance in the searching steps. The tuned FA has been successful in handling local optima. For the evaluation of the performance of the fuzzy-based FA, there have also been experiments on selected low and high dimensional benchmark functions and two engineering constrained problems [26]. The original FA and other metaheuristic algorithms have been present in the performance comparison. The experimental results have proved the efficiency and strength of the fuzzy-based FA against the other metaheuristic algorithms.

FA has also been effective in supply chain network optimization for industrial plants under a policy of vendor-managed inventory (VMI) [27]. FA's exploitation and exploration were utilized to solve the optimization problems in supply chain networks. Tuning of the algorithm parameters was done by the response surface methodology. The simulation results showed the strong performance of FA in providing better outcomes for supply chain network optimization.

Solar photovoltaic cell modelling is a major requirement in the applications of solar photovoltaic systems. It's necessary to determine cell parameters. FA has been used for identifying cell parameters to generate the solar cell characteristics under different temperature conditions and solar irradiation. The implementation showed that there was a fair agreement between data sheet values and computed values [28]. The proposed method using FA is useful for solar photovoltaic researchers, designers, and simulators.

Feature selection in machine learning is quite a challenging task. Feature selection, a part of the process of dimension reduction, helps in feature selection from data sets having the biggest impacts on the machine learning model's accuracy and performance [29]. As the feature selection looks for an optimal feature set in a wide search space, FA can be implemented here as a wrapper technique for feature selection. Therefore, an improved edition of the FA was adapted for tackling the problems faced in feature selection. The improved FA used a learning procedure based on quasi-reflection and overcame the original FA's observed drawbacks. The proposed FA-based strategy was validated, and there were k-nearest neighbors as a classification model. The developed strategy has performed better than other existing strategies in terms of the number of features and classification accuracy.

An accelerated variant of FA, named Fast Firefly Algorithm (FFA), has shown strong optimization performance by solving some benchmark functions. FFA has been superior to the original FA considering convergence rate towards the global solution following a similar precision. In the application of controlling a BLDC electric motor, FFA showed its optimization efficiency by optimizing the motor's Proportional Integral regulator parameters using the IAE, ISTE, ITAE, and ISE performance criteria [30]. In this optimization, FFA has been competitive against other metaheuristic algorithms like FA, PSO, GA, and ABC.

A modified Firefly Algorithm has been successful in minimizing operational costs of distributed green data centers (DGDCs) [31]. Green data centers integrate renewable sources and provide clean energy at reduced operating costs. In DGDCs, it's necessary to do energy efficiency and cost based scheduling of many heterogeneous applications by verifying different tasks' delay bound constraints. The effectiveness of the modified FA has been evaluated by conducting experiments using real-life data.

A multifactorial FA has successfully integrated FA's robust exploitation capability to enhance every task's self-evolution when handling low-similarity tasks for ensuring better inter-task information transfers by providing high-quality solutions [32]. The algorithm has a better encoding and decoding technique to focus on search areas on sparse and complete graphs. The test results showed that the developed encoding scheme helped the algorithms improve solutions by around 32% on average. This implementation has been successful in enhancing FA's optimization performance.

III. FIREFLY ALGORITHM (FA)

The developments of many metaheuristic algorithms have been based on inspirations form the nature. Nature has been able to discover solutions to various problems just through experiences and without being told. Early metaheuristic algorithm's main motivation has been natural selection and the fittest solution's survival. Every type of animals has its own mode of communications. Natural characteristics of these animals inspire various metaheuristic algorithm techniques.

In addition to watching and enjoying the charming view of the sky with flashing fireflies, these insects have been a motivation and center for scientific researches. In the field of metaheuristic algorithm based optimization where we need to find optimal solutions from a huge number of solutions, we can think of fireflies as solutions in the solution search space. In this way, the movement and attraction of flashing fireflies can be an inspiration for an optimization metaheuristic algorithm where solutions follow stronger (brighter) solutions. These properties mainly inspire FA.

FA is based on the characteristics of fireflies. Fireflies are winged insects or beetles that can blink or produce light at night. There's no ultraviolet frequency or infrared in the light. The lower abdomen, called bioluminescence, is responsible for producing ultraviolet frequency. Fireflies attract prey or mates by using flash light. Depending on the light pattern produced by the firefly, a suitable mate responds by mimicking the same or a similar pattern or uses a specific pattern. The flashing property of fireflies is the base of their communications. There are nearly 2000 species of fireflies having their own distinct patterns of flash. The light intensity due to flashing becomes weak for the other firefly if the distance between two fireflies is high. So, flashing should be within visual range of fireflies for proper communication. A firefly's flash light also works as a safety warning technique to warn the fireflies about potential dangers or predators.

Fireflies' flashing behaviors and the bioluminescent communication phenomenon inspire FA [33]. The formulation of FA has the following assumptions:

- The attraction between fireflies isn't based on their sex, as fireflies are unisexual.
- Attractiveness proportionally depends on their brightness. A firefly with less brightness will get attracted to a firefly with higher brightness. However, if the distance between two fireflies increases, the attractiveness decreases.
- If two fireflies have the same brightness, they will move randomly.

Constructing new solutions takes place by fireflies' random explores and attractions [34]. Fireflies' brightness is usually associated with the related problem's objective function. Their attractiveness helps them to get subdivided into small groups, and every subgroup swarm around local models.

FA is a very robust strategy to solve NP-hard problems and constrained optimization problems [35]. Although there have been wide applications of FA for solving continuous mathematical functions, there haven't been enough reports [36]. For applied mathematics, FA is a simple mathematic and logic [37]. FA's behavior is simple and suitable for solving continuous mathematical functions.

FA can outperform many conventional algorithms in terms of statistical performances measured by standard stochastic testing functions. It works using global communications among fireflies. Therefore, FA can find local and global optima simultaneously. Also, FA mainly uses real random numbers [21]. Various fireflies function independently, and the strategy is great for parallel implementation.

A. *Firefly's Attractiveness*

A firefly's attractiveness is its brightness. Suppose the brightness, I of a firefly i on another firefly j is according to the firefly's degree of brightness and the two fireflies' distance r_{ij} [22]. Equation 1 shows this scenario.

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

Suppose n fireflies are present. Also, xi corresponds to firefly i's solution. Firefly i's brightness is associated with f(xi), which is the objective function. A firefly's brightness, I is chosen to express its current position of its objective function or fitness value f(x), as shown in equation 2.

$$I_i = f\ (x_i) \qquad (2)$$

The firefly with less brightness (attractiveness) gets attracted and moves to the firefly with higher brightness, and every firefly has a particular attractiveness value β. The attractiveness value (β) is relative, and it's according to the distance between two fireflies. Equation 3 shows the firefly's attractiveness function.

$$\beta\ (r) = \beta_o\ e^{-\gamma r^2} \qquad (3)$$

In equation 3, βo is the firefly's attractiveness at r=0. However, γ is the absorption coefficient of media light.

However, the distance between firefly i and firefly j located at xi and xj respectively can be determined using the Cartesian distance, as shown in the following equation:

$$r_{ij} = |x_i - x_j| = \{ \sum_{k=1}^{d} (x_i - x_j)\}^{1/2} \qquad (4)$$

In equation 4, d denotes the number of dimensions. K denotes component in spatial coordinate.

B. *Moving Towards Attractive Fireflies*

When a firefly i moves from the position xi to another firefly j located at xj, the scenario can be explained by equation 5.

$$x_i(t+1) = x_i(t) + \beta_o\ e^{-\gamma r^2}\ (x_j - x_i) + \alpha\varepsilon_i \qquad (5)$$

In equation 5, $\beta_o\ e^{-\gamma r^2}\ (x_j - x_i)$ is because of the firefly j's attraction. In optimization, x_j is the best solution in the current population. $\alpha\varepsilon_i$ is a randomization parameter. ε_i has a range of 0 to 1. The parameter r also has a range of 0 to 1 [38]. In case βo = 0, then it becomes a normal random movement. However, α has a range of 0 to 1, and in experiments, βo can be used as equal to 1 [39]. In some works, γ has a value of 1 [38]. However, the ranges of these parameters can be varied for modifying the algorithm.

FA compares the new firefly's attractiveness with the old one. The firefly moves to a new position if that new position generates a higher attractiveness value; otherwise, the firefly stays in its current position. FA's termination criterion is according to a predefined fitness value or an arbitrarily chosen number of iterations. However, the firefly with the highest brightness moves randomly, as shown in equation 6.

$$x_i(t+1) = xi(t) + \alpha\varepsilon_i \qquad (6)$$

C. *Modification of FA to Increase Optimization Performance*

FA is a simple and efficient metaheuristic algorithm. Also, it's great for parallel implementation. Some researches have shown that FA isn't quick in convergence and often get stuck to local optima while solving optimization problems. The updates rely on present performance and don't keep any memory of previous optimal solutions and performances. It may result in losing better solutions. Also, as the parameter ranges are fixed throughout an optimization process, the search behavior stays the same for all conditions in all iterations. Therefore, modification of the standard FA to improve its performance is an important research issue. Also, the standard FA is mainly developed for solving continuous optimization problems, so modification and adjustment of FA can also be great for non-continuous problems [40].

Basically, 3 classes of modifications are common for metaheuristic algorithms. Class 1 modifications involve parameter modifications. In this category, the algorithm's parameters go through modifications and the same formulas or mechanisms are used. However, class 2 modifications have new updating mechanisms. This type of modification technique changes the whole or part of the updating mechanism or formulas. Class 3 modifies the search space or region, perhaps using the same updating formulas or mechanism, and changes in probability distribution while generating random numbers. However, some modifications may involve multiple classes. A class is selected according to the need of a specific optimization problem.

In the standard FA, the used parameters in the equations are user-defined constants. FA's performance greatly relies on its parameter values like other metaheuristic algorithms. These parameters have impacts on the degree of exploitation and exploration.

Some of the FA's modifications involve making the parameters adaptive and variable. Recent research works on FA have modified the parameters α, γ, and r. Modifying α affects the firefly's random moment, whereas modification of either r or γ affects the attraction level between fireflies. Some research works have also done adjustment on the brightness, β_0.

IV. METHODOLOGY

This section will discuss how FA has found optimal solutions using the benchmark functions. Figure 1 shows the working principle of this process. Depending on maximization, minimization, or any other factor, every function has the best function value. FA finds the best function value of various benchmark functions by a random search method. Each of the functions has one or more parameters, and these parameters have specific ranges. FA selects these parameter values within their ranges by random search. This process goes through a number of steps, as shown in figure 1. An optimization tool has been developed for finding the optimal solutions of the benchmark functions.

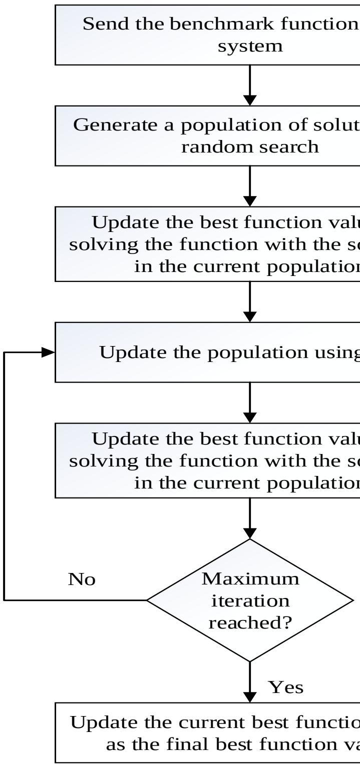

Send the benchmark function to the system
Generate a population of solutions by random search
Update the best function value by solving the function with the solutions in the current population
Update the population using FA
Update the best function value by solving the function with the solutions in the current population
Maximum iteration reached?
No
Yes
Update the current best function value as the final best function value

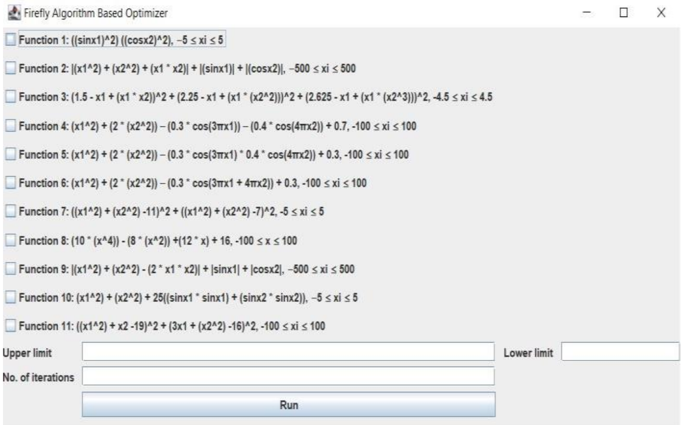
Fig. 1. Process flow for the working procedure of FA for solving benchmark functions

A. *Send the Benchmark Function to the System*

Firstly, the FA-based optimization tool receives the benchmark function from the user. After opening the FA based optimizer tool, the user will see the options of 11 benchmark functions and he just needs to select one function from there. There are other inputs like upper and lower limits of the function parameters and the number of iterations. Figure 2 shows the tool. There are 11 benchmark functions. The user has to choose any of the functions. Also, there are other options, such as "Upper limit", "Lower limit", and "No. of iterations". Every benchmark function has one or more parameters and each of these parameters has an upper limit and a lower limit. For getting more optimal solutions, it's better to keep higher number of iterations. However, it's necessary to consider that keeping the number of iterations too high may result in an increased execution time.

B. *Generate an Initial Population*

Now, the tool will generate an initial population by random search. Within the range of the parameters of the function, the strategy will generate the population of solutions. As seen in figure 2, the first benchmark function has two parameters: x1 and x2. Here, x1 and x2 have an upper limit of 500 and a lower limit of -500. The strategy will generate a population of x1 and x2 within this range.



```
Firefly Algorithm Based Optimizer
☐ Function 1: ((sinx1)^2) ((cosx2)^2), −5 ≤ xi ≤ 5
☐ Function 2: |(x1^2) + (x2^2) + (x1 * x2)| + |(sinx1)| + |(cosx2)|, −500 ≤ xi ≤ 500
☐ Function 3: (1.5 - x1 + (x1 * x2))^2 + (2.25 - x1 + (x1 * (x2^2)))^2 + (2.625 - x1 + (x1 * (x2^3)))^2, -4.5 ≤ xi ≤ 4.5
☐ Function 4: (x1^2) + (2 * (x2^2)) – (0.3 * cos(3πx1)) – (0.4 * cos(4πx2)) + 0.7, -100 ≤ xi ≤ 100
☐ Function 5: (x1^2) + (2 * (x2^2)) – (0.3 * cos(3πx1) * 0.4 * cos(4πx2)) + 0.3, -100 ≤ xi ≤ 100
☐ Function 6: (x1^2) + (2 * (x2^2)) – (0.3 * cos(3πx1 + 4πx2)) + 0.3, -100 ≤ xi ≤ 100
☐ Function 7: ((x1^2) + (x2^2) -11)^2 + ((x1^2) + (x2^2) -7)^2, -5 ≤ xi ≤ 5
☐ Function 8: (10 * (x^4)) - (8 * (x^2)) +(12 * x) + 16, -100 ≤ x ≤ 100
☐ Function 9: |(x1^2) + (x2^2) - (2 * x1 * x2)| + |sinx1| + |cosx2|, −500 ≤ xi ≤ 500
☐ Function 10: (x1^2) + (x2^2) + 25((sinx1 * sinx1) + (sinx2 * sinx2)), −5 ≤ xi ≤ 5
☐ Function 11: ((x1^2) + x2 -19)^2 + (3x1 + (x2^2) -16)^2, -100 ≤ xi ≤ 100
Upper limit [          ]   Lower limit [          ]
No. of iterations [          ]
[Run]
```

Fig. 2. FA based optimizer

C. *Update the Best Function Value Using the Solutions in the Current Population*

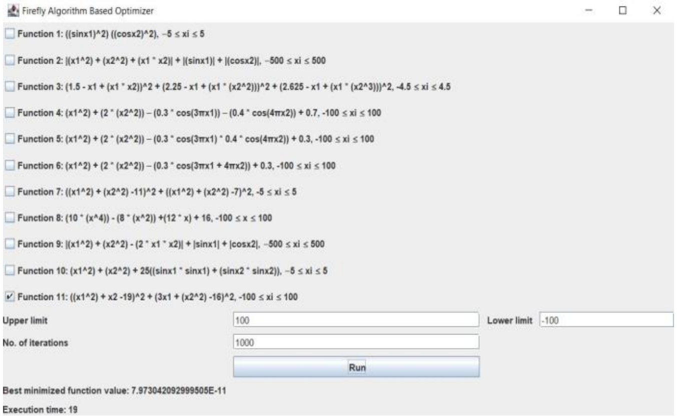
Using the solutions in the current population, the tool will solve the function and update the best function value. The best function value can be updated based on considerations like maximization, minimization, etc. In our research, we consider minimization of the benchmark functions.

D. *Update the Population Using FA and Update the Best Function Value*

Now, the tool will update the population by implementing FA. FA will update every solution in the current population. If any updated solution falls outside the range with the given upper limit and lower limit, the solution will be updated again to keep it within the range. The solutions from the updated population will be used again to solve the functions. Thus, if the tool finds a better function value, the best function value will be updated.

E. *Termination Criterion and Declaring the Final Best Function Value*

The process will go on like this, and once the maximum iteration is reached, the process will end. Now, the tool will update the most recently updated best function value as the final best function value. Figure 3 shows an obtained output in the FA base optimizer tool.



```
Firefly Algorithm Based Optimizer
☐ Function 1: ((sinx1)^2) ((cosx2)^2), −5 ≤ xi ≤ 5
☐ Function 2: |(x1^2) + (x2^2) + (x1 * x2)| + |(sinx1)| + |(cosx2)|, −500 ≤ xi ≤ 500
☐ Function 3: (1.5 - x1 + (x1 * x2))^2 + (2.25 - x1 + (x1 * (x2^2)))^2 + (2.625 - x1 + (x1 * (x2^3)))^2, -4.5 ≤ xi ≤ 4.5
☐ Function 4: (x1^2) + (2 * (x2^2)) – (0.3 * cos(3πx1)) – (0.4 * cos(4πx2)) + 0.7, -100 ≤ xi ≤ 100
☐ Function 5: (x1^2) + (2 * (x2^2)) – (0.3 * cos(3πx1) * 0.4 * cos(4πx2)) + 0.3, -100 ≤ xi ≤ 100
☐ Function 6: (x1^2) + (2 * (x2^2)) – (0.3 * cos(3πx1 + 4πx2)) + 0.3, -100 ≤ xi ≤ 100
☐ Function 7: ((x1^2) + (x2^2) -11)^2 + ((x1^2) + (x2^2) -7)^2, -5 ≤ xi ≤ 5
☐ Function 8: (10 * (x^4)) - (8 * (x^2)) +(12 * x) + 16, -100 ≤ x ≤ 100
☐ Function 9: |(x1^2) + (x2^2) - (2 * x1 * x2)| + |sinx1| + |cosx2|, −500 ≤ xi ≤ 500
☐ Function 10: (x1^2) + (x2^2) + 25((sinx1 * sinx1) + (sinx2 * sinx2)), −5 ≤ xi ≤ 5
☑ Function 11: ((x1^2) + x2 -19)^2 + (3x1 + (x2^2) -16)^2, -100 ≤ xi ≤ 100
Upper limit [100]   Lower limit [-100]
No. of iterations [1000]
[Run]
Best minimized function value: 7.973042092999505E-11
Execution time: 19
```

Fig. 3. Output in FA based the optimizer tool

V. TEST RESULTS

We have run optimization tests of selected 11 benchmark functions using FA. The tests were performed using Intel(R) Core(TM) i5-8250U with 8 GB DDR4 RAM on the operating system of Windows 10. The code has been written in Java using the Intellij IDEA Community Edition 2020.2.3. For the 11 benchmark functions, we have done minimization processes and selected the best function values based on the minimized function values. 10000 iterations were used while solving the functions. To bring better results, we modified the equations of FA a little. As an example, in equation 4, the range of α is 0 to 1 in many existing works. However, we used the range of -1 to 1 for α. This modification has helped to obtain more optimal solutions. It's also possible change other parameters from the equations of FA. But we have run tests adapting only one parameter (α).

We have solved 11 benchmark functions using FA and compared the test results with the results of other metaheuristic algorithms like BA, BFA, CS. The benchmark functions have been selected from existing works. We have collected the results of BA, BFA, and CS from existing works. By comparing these results, we can evaluate how FA performs while finding optimal solutions in optimization. The 11 benchmark functions are as following.

1) Parsopoulos Function:

$$f_1(x) = (\sin x_1)^2\,(\cos x_2)^2 \tag{7}$$

x_i has a range: $-5 \le x_i \le 5$.

2) Bartels Conn function:

$$f_2(x) = |x_1^2 + x_2^2 + x_1x_2| + |\sin x_1| + |\cos x_2| \tag{8}$$

x_i has a range: $-500 \le x_i \le 500$.

3) Beale Function:

$$f_3(x) = (1.5-x_1+x_1x_2)^2 + (2.25-x_1+x_1x_2^2)^2 + (2.625-x_1+x_1x_2^3) \tag{9}$$

x_i has a range: $-4.5 \le x_i \le 4.5$.

4) Bohachevsky Function 1:

$$f_4(x)= x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7 \tag{10}$$

x_i has a range: $-100 \le x_i \le 100$.

5) Bohachevsky Function 2:

$$f_5(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) * 0.4\cos(4\pi x_2) + 0.3 \tag{11}$$

x_i has a range: $-100 \le x_i \le 100$.

6) Bohachevsky Function 3:

$$f_6(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1 + 4\pi x_2) + 0.3 \tag{12}$$

x_i has a range: $-100 \le x_i \le 100$.

7) Himmelblau Function:

$$f_7(x) = (x_1^2 + x_2 -11)^2 + (x_1 + x_2^2 -7)^2 \tag{13}$$

x_i has a range: $-5 \le x_i \le 5$.

8) 8th Benchmark Function:

$$f_8(x) = 10x^4 - 8x^2 +12x + 16 \tag{14}$$

x has a range: $-100 \le x \le 100$.

9) 9th Benchmark Function:

$$f_9(x) = |x_1^2 + x_2^2 - 2x_1x_2| + |\sin x_1| + |\cos x_2| \tag{15}$$

x_i has a range: $-500 \le x_i \le 500$.

10) Egg Crate Function:

$$f_{10}(x) = x_1^2 + x_2^2 + 25(\sin^2 x_1 + \sin^2 x_2) \tag{16}$$

x_i has a range: $-5 \le x_i \le 5$.

11) 9th Benchmark Function:

$$f_{11}(x) = (x_1^2 + x_2 -19)^2 + (3x_1 + x_2^2 -16)^2 \tag{17}$$

x_i has a range: $-100 \le x_i \le 100$.

Table 1 shows the test results. Here, the mean values are the main factors based on which performances of the algorithms can be evaluated. Considering the mean values, FA outperforms BA in 5 functions, and BA outperforms FA in 3 functions. FA outperforms BFA in 6 functions, and BFA outperforms FA in 2 functions. However, FA and CS didn't outperform each other; these two algorithms performed at a similar level. Figure 4 shows the ranks of algorithms based on the mean values.

TABLE I. TEST RESULTS

Functions	BA				BFA				CS				FA			
	Best	Worst	Mean	Time(ms)	Best	Worst	Mean	Time(ms)	Best	Worst	Mean	Time(ms)	Best	Worst	Mean	Time(ms)
f1	1.0E-5	0.025	0.0065	175	1.72E-10	4.23E-8	1.27E-8	506	NA	NA	NA	NA	0	0	0	123
f2	1.29	260.3	91.73	252	1.00	10862	3273.97	724	NA	NA	NA	NA	1.38	13688	2725	147
f3	1.18E-6	8.41E-3	3.74E-3	335	2.68E-10	2.86E-7	1.01E-7	777	NA	NA	NA	NA	0	1.76	0.64	939
f4	.08	13.01	5.15	245	0.24	314.87	41.80	689	NA	NA	NA	NA	0.05	10.55	1.41	203
f5	-.002	17.38	5.17	250	0.013	314.47	30.38	689	NA	NA	NA	NA	0.38	13.92	4.96	71
f6	0.011	26.12	7.05	212	4.61E-7	212.20	30.98	600	NA	NA	NA	NA	0	60.14	5.68	101
f7	1.05E-5	0.361	0.0745	301	8.59E-9	1.21E-6	2.99	652	0	0	0	32.9	0	3.22	0.30	59
f8	NA	NA	NA	NA	NA	NA	NA	NA	5.23	5.23	5.23	29.6	5.23	5.23	5.23	48.4
f9	NA	NA	NA	NA	NA	NA	NA	NA	1	1.15	1.055	151.5	1	1.04	1.01	79
f10	1.27E-4	1.30	.318	354	1.98E-9	2.12E-6	4.41E-7	712	NA	NA	NA	NA	0	0.35	.035	198
f11	NA	NA	NA	NA	NA	NA	NA	NA	0	0	0	40.5	0	0	0	49.2

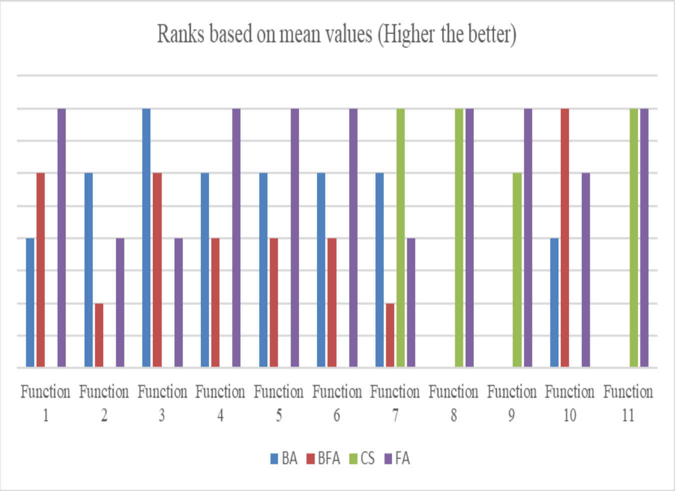


Fig. 4. Ranks of algorithms based on mean values

Figure 5 shows the ranks of algorithms based on the best values. As seen from the figure, FA outperforms BA in 6 functions, and BA outperforms FA in 2 functions. FA outperforms BFA in 6 functions, and BFA outperforms FA in 2 functions. Again, BA and CS perform at a similar level. These results prove the strong performance of FA considering the best function values.

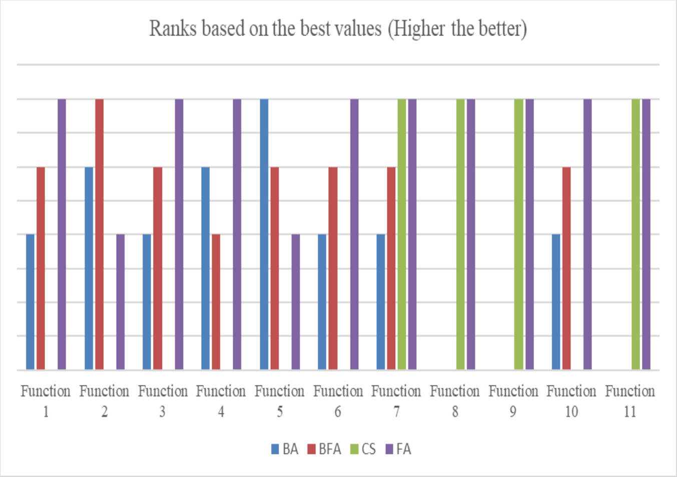


Fig. 5. Ranks of algorithms based on the best values

VI. DISCUSSION

The obtained test results of FA have shown promising results in determining obtaining optimal solutions while solving the benchmark functions. Considering the mean function values and the best function values, FA outperformed both BA and BFA. However, FA couldn't outperform CS, but FA wasn't outperformed by CS also. FA and CS performed at a similar level. These test results indicate that FA is an efficient metaheuristic algorithm in the field of optimization. The promising results of FA indicate that the algorithm is highly efficient in obtaining solutions while handling optimization problems.

FA also determined the optimal solutions with lower execution times, but execution times weren't considered in this research because the test results of other metaheuristic algorithms were obtained from other existing works, and different devices were used for those experiments. However, lower execution times indicate that FA is computationally light, but it can still obtain optimal solutions in optimization. Despite having 10000 iterations while solving the functions, FA took quite shorter execution times.

The optimality of the solutions obtained by FA indicates that the slight adaptation of FA has ensured better convergence rate as compared to the original FA. Unlike the original FA, the slightly adapted FA in this research didn't struggle much while handling local optima, and it can be realized due to the optimality of the obtained solutions. The adaptation has also enhanced the simplicity of the random search process and it has resulted in short execution times.

VII. CONCLUSION

FA is an efficient metaheuristic algorithm mimicking fireflies' behaviors at night. This paper has evaluated the optimization performance of FA by solving 11 benchmark functions. Parameter adaptation of FA has been successful in ensuring higher optimization performance. The performances of FA have been compared to other algorithms, and FA showed promising results. These results prove that FA is a very suitable metaheuristic algorithm to handle optimization problems.

However, there can be more future research works on FA as further improvement of FA's performance is still possible. As an example, FA performed really well compared to other 3 metaheuristic algorithms but FA couldn't completely outperform CS in terms of optimality of the solutions. FA and CS performed at a similar level. Therefore, further adaptation of FA can improve FA's dominance over other metaheuristic algorithms.

Software professionals can rely on FA for solving optimization problems. FA has great potential in optimization. FA can be implemented for other optimization works like multi-objective optimization, t-way testing, etc. Many advanced optimization works are possible using FA.

REFERENCES

[1] [1] R. R. O. A. A. H. L. Z. N. R. Khin Maung Htay, "A Pairwise T-Way Test Suite Generation Strategy Using Gravitational Search Algorithm," in 2021 International Conference on Artificial Intelligence and Computer Science Technology (ICAICST), 2021.

[2] [2] P. S. A. P. E. T. Johann Dréo, Metaheuristics for Hard Optimization, Berlin, Heidelberg: Springer, 2006.

[3] [3] Xin-SheYang, "Chapter 6 - Genetic Algorithms," in Nature-Inspired Optimization Algorithms (Second edition), 2021, pp. 91-100.

[4] [4] S. H. J. A. W. J. Darrall Henderson, "The Theory and Practice of Simulated Annealing," in Handbook of Metaheuristics, 2003, pp. 287-319.

[5] [5] X.-S. Yang, "Chapter 2 - Analysis of Algorithms," in Nature-Inspired Optimization Algorithms, 2014, pp. 23-44.

[6] [6] Xin-SheYang, "Chapter 8 - Particle Swarm Optimization," in Nature-Inspired Optimization Algorithms (Second edition), 2021.

[7] [7] A. A.-D. M. A. A.-B. A. A. S. O. Mohammed Alweshah, "β -Hill climbing algorithm with probabilistic neural network for classification problems," Journal of Ambient Intelligence and Humanized Computing, vol. 11, pp. 3405-3416, 2020.

[8] [8] M. K. Xin-She Yang, "Chapter 1 - Nature-inspired computation and swarm intelligence: a state-of-the-art overview," in Nature-Inspired Computation and Swarm Intelligence, 2020, pp. 3-18.

[9] [9] T. S. M. D. Manuel López-Ibáñez, "Ant Colony Optimization: A Component-Wise Overview," in Handbook of Heuristics, 2018, pp. 371-407.

[10] [10] J. K. D. G. A. B. Kamil Musiał, "Tabu Search and Greedy Algorithm Adaptation to Logistic Task," in Computer Information Systems and Industrial Management, 2017.

[11] [11] N. P. M. Floriano De Rango, "Chapter 19 - Multirobot coordination through bio-inspired strategies," in Nature-Inspired Computation and Swarm Intelligence, 2020, pp. 361-390.

[12] [12] A. B. N. K. Z. Z. S. H. Mashuk Ahmed, "Comparison of Performances of Jaya Algorithm and Cuckoo Search Algorithm Using Benchmark Functions," in Proceedings of International Conference on Emerging Technologies and Intelligent Systems, 2021.

[13] [13] L. W. J. D. S. P. Huang Chen, "Bacterial Foraging Optimization Based on Self-Adaptive Chemotaxis Strategy," Computational Intelligence and Neuroscience, vol. 2020, 2020.

[14] [14] H. S. A. A. M. N. M. A. M. K. Z. Z. Yazan A. Alsariera, "Comparative Performance Analysis of Bat Algorithm and Bacterial Foraging Optimization Algorithm using Standard Benchmark Functions," in 2014 8th Malaysian Software Engineering Conference (MySEC), 2015.

[15] [15] V. G. H. X. X. W. K. Z. X. Z. Gao, "Harmony Search Method: Theory and Applications," Computational Intelligence and Neuroscience, 2015.

[16] [16] A. B. N. A. N. M. H. A. W. M. A. Julius B. Odili, "African Buffalo Optimization Algorithm Based T-Way Test Suite Generation Strategy for Electronic-Payment Transactions," in Proceedings of International Conference on Emerging Technologies and Intelligent Systems, 2021.

[17] [17] R. S. Dunia Sattar, "A smart metaheuristic algorithm for solving engineering problems," Engineering with Computers, vol. 37, pp. 2389-2417, 2020.

[18] [18] T. J. K. Tatjana Davidović, "Convergence Analysis of Swarm Intelligence Metaheuristic Methods," in Optimization Problems and Their Applications, Springer International Publishing, 2018.

[19] [19] S. S. C. V. K. Sourabh Katoch, "A review on genetic algorithm: past,," Multimedia Tools and Applications, vol. 80, pp. 8091-8126, 2021.

[20] [20] S. S. Stefan Edelkamp, "Selective Search," in Heuristic Search: Theory and Applications, 2012, pp. 633-669.

[21] [21] X.-S. Yang, Nature-Inspired Metaheuristic Algorithms, Luniver Press, 2008.

[22] [22] X.-S. Yang, "Firefly Algorithms for Multimodal Optimization," Stochastic Algorithms: Foundations and Applications, pp. 169-178, 2009.

[23] [23] A. M. Z. N. H. M. A. U. Nur Farahlina Johari, "Machining Parameters Optimization using Hybrid Firefly Algorithm and Particle Swarm Optimization," Journal of Physics: Conference Series, 2017.

[24] [24] D. R. P. A. J. S. K. K. B.K. Patle, "On firefly algorithm: optimization and application in mobile robot navigation," World Journal of Engineering, vol. 14, no. 1, 2017.

[25] [25] Z. J. C. L. H. Y. Ren Changan, "Application of Firefly Algorithm in Scheduling Optimization of Combined Cooling, Heating and Power with Multiple Objectives," A Journal of the Italian Association of Chemical Engineering, vol. 67, 2018.

[26] [26] S. S. M. M. M. B. Mahdi Bidar, "Enhanced Firefly Algorithm Using Fuzzy Parameter Tuner," Computer and Information Science, vol. 11, no. 1, 2018.

[27] [27] M. Z. R. O. D. A. Abdulhadi Altherwi, "A Modified Firefly Algorithm for Global Optimization of Supply Chain Networks," in Proceedings of the 5th NA International Conference on Industrial Engineering and Operations Management, Detroit, Michigan, USA, 2020.

[28] [28] K. A. K. Rajkumar K, "Application of firefly algorithm for power estimations of solar photovoltaic power plants," Energy Sources, Part A: Recovery, Utilization, and Environmental Effects, 2021.

[29] [29] D. C. L. G. M. Ź. I. S. N. B. Timea Bezdan, "Feature Selection by Firefly Algorithm with Improved Initialization Strategy," in 7th Conference on the Engineering of Computer Based Systems, 2021.

[30] [30] R. B. Y. B. M. M. A. R. Smail Bazi, "A Fast Firefly Algorithm for Function Optimization: Application to the Control of BLDC Motor," Sensors, vol. 21, no. 16, 2021.

[31] [31] W. L. F. M. H. Y. M. Z. M. S. Ahmed Chiheb Ammari, "Firefly Algorithm and Learning-based Geographical Task Scheduling for

Operational Cost Minimization in Distributed Green Data Centers," Neurocomputing, 2022.

[32] [32] H. T. T. B. Ta Bao Thang, "A hybrid Multifactorial Evolutionary Algorithm and Firefly Algorithm for the Clustered Minimum Routing Cost Tree Problem," Knowledge-Based Systems, 2022.

[33] [33] A. Altherwi, "Application of the Firefly Algorithm for Optimal Production and Demand Forecasting at SelectedIndustrial Plant," Open Journal of Business and Management, vol. 8, no. 6, 2020.

[34] [34] X.-S. Yang, "Metaheuristic Optimization: Algorithm Analysis and Open Problems," in Experimental Algorithms, 2011.

[35] [35] A. V. Theofanis Apostolopoulos, "Application of the Firefly Algorithm for Solving the Economic Emissions Load Dispatch Problem," International Journal of Combinatorics, vol. 2011, 2010.

[36] [36] S. Ż. Szymon Łukasik, "Firefly Algorithm for Continuous Constrained Optimization Tasks," in Computational Collective Intelligence. Semantic Web, Social Networks and Multiagent Systems, Wrocław, Poland, 2009.

[37] [37] P. Azar, "Fireflies & Oscillators," APPLIED MATHEMATICS CORNER, 2009.

[38] [38] Z. C. H. S. S. R. X.-S. Y. Hui Wang, "Randomly attracted firefly algorithm with neighborhood search and dynamic parameer adjustment mechanism," METHODOLOGIES AND APPLICATION, 2016.

[39] [39] H. C. O. Surafel Luleseged Tilahun, "Modified Firefly Algorithm," Journal of Applied Mathematics, vol. 2012, 2012.

[40] [40] N. N. H. S. L. T. J. M. T. N. Waqar A. Khan, "A Review and Comparative Study of Firefly Algorithm and its Modified Versions," in Optimization Algorithms - Methods and Applications, 2016.