Article

The Enhanced Firefly Algorithm Based on Modified Exploitation and Exploration Mechanism

Moath Sababha *, Mohamed Zohdy and Maged Kafafy

Electrical and Computer Engineering, Oakland University, Rochester, MI 48309, USA; zohdyma@oakland.edu (M.Z.); makafafy@oakland.edu (M.K.)

* Correspondence: msababha@oakland.edu; Tel.: +1-73-4834-8391

check for updates

Abstract: As a nature-inspired search algorithm, the Firefly algorithm (being a naturally outstanding search algorithm with few control parameters) may have a considerable influential performance. In this paper, we present a new firefly algorithm to address the parameter selection and adaptation strategy in the standard firefly algorithm. The proposed firefly algorithm introduces a modified exploration and exploitation mechanism, with adaptive randomness and absorption coefficients. The proposed method employs the adaptation of the randomness and absorption coefficients to be a function of time/iterations. Moreover, gray relational analysis advancing fireflies is used to allocate different information from appealing ones effectively. Standard benchmark functions are applied to verify the effects of these improvements and it is illustrated that, in most situations, the performance of the proposed firefly algorithm is superior to (or at least highly competitive with) the standard firefly algorithm, and state-of-the-art approaches in terms of performance.

Keywords: firefly algorithm; nature-inspired search algorithm; exploration and exploitation mechanism; optimization problem

1. Introduction

Optimization is something that is involved in all our activities. Everything from the simple decision of what time to leave for work all the way to the more complex decisions such as how to budget a daily cost of living allowance requires optimizing procedures. The process of optimization is finding an optimal solution for a function. All probable values can be obtained solutions and the optimal solution is assumed to be the extreme value. There are two categories for optimization algorithms. These are: stochastic and deterministic. Deterministic methods include classical optimization modes such as Golden mean, Newton method, modified Newton method, gradient method, along with Lagrange methods. These are largely dependent on gradient information and is most useful in unimodal functions with one global optimum. This procedure however has problem when gradients are small or part of flat regions [1]. Therefore, stochastic algorithms are preferred in most instances as it can get away from local minima for a better performance [2].

Stochastic algorithms are a subset of metaheuristic algorithms. Literature tend to indicate to stochastic methods as metaheuristics [3,4]. Heuristic refers 'to discover solution by trial and error'. When referring to a "higher–level heuristic" that refers to Meta-heuristic, whose search is the result of a trade-off between local search and randomization. The balance among exploration and exploitation or diversification and intensification is focused on bio-inspired algorithm [5].

Recently, the biologically inspired algorithms have exhibited as powerful tool to solve complex engineering optimization problems through the use of hard optimization problems [6]. These methods are successful because the programs can maintain the proper balance of exploration and exploitations via a set candidate for solutions with generational improvement. In [2], they refer to this as exploitation

in reference to the ability of the algorithm to use previous information in order to update its solutions. Animal swarm behaviors have been said to been inspired by swarm-based algorithms. These are defined as bio-inspired methods where the intelligence is part of social behavior in animals and insects. Relatively, all metaheuristic algorithms seek for achieving the balance between local search and randomization [7].

Inspired by nature, optimization algorithms contain all searching for optimal value problems. Generally, metaheuristic methods perform on a population of solutions to obtain the best solutions. Computer scientists studied the possibility of making the conception of evolution as an optimization method and this created a subset of gradient free methods called genetic algorithms (GA) [8]. Since then many other nature-inspired metaheuristic methods have been proposed, such as differential evolution (DE) [9,10], cuckoo search (CS) [11], particle swarm optimization (PSO) [12,13], and more recently, the firefly algorithm (FA) [14,15] that is inspired by fireflies' behavior in nature.

Firefly algorithm is a part of the family of swarm intelligence algorithms. This algorithm is based upon the theory that the bioluminescence of an insect's body can be used to interact and communicate. It allows for the development of communication as part of a group social behavior. Firstly, proposed by Yang in 2008, the firefly algorithm or firefly-inspired algorithm [16], is a metaheuristic optimization algorithm, inspired by the flashing behavior of fireflies. The fundamental purpose for a firefly's flash is applied as a signal system to appeal to other fireflies. Recent researches demonstrate that the firefly algorithm is quite powerful and efficient, and the performance of firefly algorithm can be improved with feasible promising results [17–21]. Firefly algorithm is an effective algorithm in exploitation (i.e., local search) but sometimes it settles into some local optima so that it fails to perform well in the global search. Also, the search in the firefly method relies totally on random walks, so a convergence is not guaranteed. Firstly, presented here, a main improvement of including effective adaptive parameters relies on the iteration processes to enhance the exploration and exploitation mechanisms in the original firefly method, thus making the approach more feasible for a wider range of practical applications while preserving the attractive characteristics of the basic FA. The method proposed in this paper is based on the measurement taken to counteract the limitations and weaknesses of the basic firefly method. However, to improve the execution of the presented algorithm, the exploitation and exploration of the search enhanced by altering the values of the fixed parameters. The proposed method is exhibited to be effective in getting satisfactory results since it aids in providing an equilibrium on exploitation and exploration abilities. Proposed approach is evaluated on six standard benchmarking functions that have ever been applied to verify optimization algorithms in continuous optimization problems. Simulation results show that the proposed method performs more efficiently and accurately than basic FA, PSO, DE and other state-of-the-art approaches.

This paper is organized as follows: In Section 2, the Firefly method is briefly presented. The proposed firefly method is presented and discussed in Section 3. Simulation and examples of the proposed firefly algorithm presented in Section 4. Finally, discussion and conclusions are outlined in Sections 5 and 6, respectively.

2. Firefly Algorithm

The firefly algorithm is a new ecology intelligence metaheuristic method [19–21] for solving optimization problems, in which the search algorithm is inspired by social behavior of fireflies and the phenomenon of bioluminescent communication. There are two crucial issues in FA that are the formulation of attractiveness and modification of light intensity. The firefly algorithm imitates the social behavior of fireflies flying in the tropical summer sky. Fireflies communicate, hunt for pray and attract other fireflies (especially, the opposite sex fireflies) using bioluminescence with various flashing patterns. By mimicking nature, various metaheuristic algorithms can be designed. For simplicity, some of the flashing characteristics of fireflies are idealized so as to design a firefly-inspired algorithm, which are three idealized rules described as follows: (1) All fireflies are the same sex so that one firefly will be attracted by other fireflies despite their sex. (2) Attractiveness is proportional to the brightness

which declines with increasing distance between fireflies. For any couple of flashing fireflies, the less bright one will move towards the brighter one. If there are no brighter fireflies than a particular firefly, this individual will move at random in the search space. (3) The brightness of a firefly is determined or influenced by the objective function.

For a maximization problem, brightness can simply be proportional to the value of the cost function. Other forms of brightness can be defined in a similar way to the fitness function in PSO algorithm. The main update formula for any couple of two fireflies i and j at x_i, x_j, is:

$$x_i^{Itr+1} = x_i^{Itr} + \beta\left(x_i^{Itr} - x_j^{Itr}\right) + \alpha\epsilon_i^{Itr} \tag{1}$$

where α is a parameter controlling the step size, $\beta = \beta_0 \exp(-\gamma r^2)$ is the attractiveness with β_0 represents the attractiveness at distance $(r = 0)$ and γ represents the light absorption coefficient. ϵ_i is randomization where the vector of random variables being drawn from a distribution (e.g., Gaussian distribution). The distance between any pair of fireflies i and j at x_i, x_j, can be the Cartesian distance $r_{ij} =\parallel x_i - x_j \parallel_2$ relying on the practical application problems. In this paper, we take $\beta_0 = 1$, $\gamma = 1$ and $\alpha \in [0, 1]$.

3. The Proposed Adaptive Approach for Firefly Algorithm

It is important to mention that Firefly algorithm (FA) is better than the evolutionary algorithm in terms providing solution to complex non-linear optimization. This is because FA is feasible, the time of execution is less, and lastly, the stability factor is high. However, the FA algorithm has got some weaknesses and imitations in spite of it considered to execute efficiently than other metaheuristic algorithms. Firstly, the parameters that form the algorithm are set fixed [20]. In this case, they cannot be modified in regard to time or iteration. In the literature, they have outlined that the performance of FA predetermined with distinguishing features is best on functions of narrow variable range and low dimensions. Moreover, the sizes or variable range may increase if the problems are more complicated. Hence the issues might not be fit, and their execution might be dropped. The changes help in the enhancement of the diversification area and the algorithm speed to get rid of premature convergence of the algorithm. Another limitation of FA is that it does not have the mechanism to remember the historical data, which may be significant. It also asserts that considering the early simulation, there are still some changes in the solution because of fixed randomness value as the most favorable condition is approaching. Therefore, an active research ability potential is needed to enhance the aspect.

The method proposed in this paper is based on the measurement taken to counteract the FA limitations and weaknesses. One way to improve the execution of the algorithm is that the exploitation and exploration of the search need to be enhanced by altering the values of the fixed parameters. In regard to the FA original, many values of the parameter in the equation of renewal movement are predetermined and fixed. Also, in the suggested algorithm, the randomization and attractiveness parameter are set in advance at the stage of initialization and later modified in the process of optimization. During the repetition process, the capability of using the local knowledge is another aspect that could be used in enhancing the original algorithm. The fact that the algorithm is still having the significant factor of randomization is determined by FA having less memory and jumping out of the furthest end in the first repetition, and it may be an issue. It means that it is in a mode of movement. Therefore, the objectives of the modification are to improve local exploration and search at the local extreme and end of the search process, to avoid any casual pace. The last is to make the algorithm to move fast to the optimum point.

The focus of the proposed FA algorithm is the enhancement in the movement area of Firefly. In regard to exploitation and exploration mechanism, the mathematical expression used to enhance the movement of the firefly is given as:

$$x_{i+1} = \beta(t)x_i + x_j(1 - \beta(t)) + \alpha(t)\epsilon_i \tag{2}$$

In this case, the random number is represented by ϵ_i; attractiveness coefficient is represented by, $\beta(t)$, the time t, and the randomness coefficient at time (t) is represented by $\alpha(t)$. Considering the movement equation in Equation (2), an exploitation method is represented by the first two terms to show a better solution to the rest of the agents while exploration processes are described by the last term for search space exploration. Furthermore, attractiveness $\beta(t)$, and the randomization $\alpha(t)$ coefficients are incorporated to modify the adaptive functions. Therefore, in the search process, the parameter of attractiveness is increased adaptively, while the randomization is reduced over time/iteration.

At the early stage of the search process, the randomization $\alpha(t)$ can be assumed at a high value, while the attractiveness $\beta(t)$ can be considered in a low value. They are assumed to make sure that all the fireflies are to move randomly around the early stages so that global searching process can be fine-tuned. The strategy of changing the parameters adaptively over time will aid in overlapping the potential and balance of intense local and global search so that at the early stage, the strategy would not be trapped into a local extreme point.

In regard to the end of optimization stage, the light intensity is increased by setting the attractiveness to a higher value so that firefly with the optimum location to produce lighter making others move closer. On the other hand, by setting the randomization to a low value allows fireflies to move to a nearer excellent location, escalating the fireflies' intensity. To improve the accuracy of the algorithm of the extreme value, then their movements need to be targeted.

Three parameters are used in standard FA in solving the optimization, and the parameters might result in the notably different performance of FA. For example, there is the coefficient of γ, α, which represents absorption, and a represent randomization parameter, respectively. However, it is hard to manually tune the parameters by looking at various problems with distinguish characteristics. In this case, to improve FA, several adaptive functions can be employed, and two mechanisms are incorporated to eradicate premature convergence of classical FA. Ideally, to create an equilibrium between the exploration and the exploitation based on the proposed adaptive function. Moreover, the mechanisms are gray-based coefficient for improving the heterogeneous search efficiently [22], and the distance-based adaptive, which is for information sharing. Both the mechanism can exchange messages and is used to tune the control parameters in FA method. Also, a new strategy was suggested for the selection of the parameter of randomization. Therefore, the randomness coefficient α equation can be expressed as:

$$\alpha(Itr_i) = \exp\left(1 - \left(\frac{Itr_{max}}{Itr_{max} - Itr_i}\right)^c\right) \tag{3}$$

where c represents the integer number to determine the speed of decaying of the randomness, Itr_{max} represents maximum number of iteration and Itr_i is the current iterations number.

Moreover, parameter γ is a vital aspect in characterizing the differences of the speed of the convergence and the attractiveness. The outcome is that, when a constant c is applied in solving the problems of optimization, then FA execution will be noticeably constrained as done in traditional FA. As well known that during the search process attractiveness should vary with different distances among the population, and it must also be connected with distance among the fireflies. The importance of information of the distances is to promise search adaptively. Consequently, the ratio of distance, which is to trace the promising flight direction adaptively is the determinant factor in the definition of an adaptive coefficient. Its definition is as follows:

$$\gamma(Itr_i) = 1 - \exp\left(1 - \left(\frac{Itr_{max}}{Itr_{max} - Itr_i}\right)^c\right) \tag{4}$$

Based on the adjustment, the proposed firefly algorithm will not get trapped in the local extreme, and it will escalate the convergence speed making the solution at optimum better a better since there will be an equilibrium between local and global search.

Heterogeneous is applied in the updating rule based on Gray relational analysis (GRA), which is the same as the measure for the finite sequence having incomplete information, suggested in [22,23].

Therefore, it is carried to enhance the fireflies search capabilities. In the search process, two updating equations are explained and chosen randomly and are presented in Equation (4):

$$x_{i+1} = \begin{cases} \beta(i)x_i + x_j(1 - \beta(i)) + \alpha(i)\epsilon_i & rand > 0.5 \\ \frac{NG-i}{NG}(1 - \delta)x_i + \delta x_{best} & ELSEWHERE \end{cases} \tag{5}$$

where δ represents the gray coefficient and NG is the number of generations.

The fireflies are given a chance to acquire more beneficial information from others and change the directions of flight adaptively. At the final point, the step measurement is meant to make the algorithm have an equilibrium between exploitation and exploration mechanisms, and to enhance the algorithm in case the search conditions are in sophisticated and ample space and high conditions. Hence, applying all the suggested changes and considerations, the FA is enhanced and can be outlined in the pseudo-code as follows in Algorithm 1.

Algorithm 1: The Proposed Firefly Algorithm Improved

Input: Cost Function $f(x)$, Initial population of Fireflies x_0, the Max number of Iteration ITR_{max}, light absorption coefficient γ, the randomness coefficient α.
Generate initial random population x_i, $(i = 1, 2, \ldots, n)$, initialize the Light Intensity I_0 at x_i by $f(x_i)$.
While Loop: till $Itr_i \leq ITR_{max}$ do
Determine the value of the adaptive parameters (γ, α) in Equations (3) and (4)
   For loop: for each $i = 1 \ldots n$ all n fireflies do
   Inner loop: for each $j = 1 \ldots n$ all n fireflies do
   Evaluate the distance, r, between the two particles (x_i, x_j)
   Evaluate the attractiveness $\beta = \exp(-\gamma r^2)$
              If $\left(I_i < I_j\right)$, move firefly i towards j;
          Update parameter values (γ, α) as Equations (3) and (4)
           Evaluate new solution x_{i+1} as in Equation (5)
end if
End For j
   End For i
Update light intensity $I_{x_{i+1}}$
Rank the fireflies and find the current global best values
End While
Output: the best fireflies solutions x and elapsed time

---

4. Accuracy and Convergence of the Presented Algorithm

The effect of tuning and choosing of the iteration parameters and a population of the presented algorithm are outlined in this section. Also, it observes and examines the convergence features and the effect on the accuracy of the optimal solution. It is because, when providing a solution for any optimization problem, the parameters of bio-inspired algorithms computation need to be found and selected correctly to get the most favorable results. The simulation part gives the comparison between the proposed algorithm and its predecessor. Also, the objective of the research is to investigate the role of the number of iteration and impact of the population size in various dimensions of the problem [24]. There are two unconstrained problems of single optimization outlined in the benchmark functions, which is applied in this study. Two types of optimization problems that form the functions are local optima and a single global optimum. They consist of multimodal and unimodal types.

The personal computer (PC) that has a processor provided a platform for the performance of the experimental testing. In this case, the PC had a CPU Intel (R) Core (TM) i7-2.4G that has Windows 10 operating system, and memory of 4.00 GB RAM. MATLAB R2013a (MathWorks, Natick, MA, USA) is used in coding the program. Some of the parameters are made similar is to provide an

honest comparison of the algorithms. Also, the two-benchmark function both have a global optimum. When the number of iteration and the population size is stationary at a specific value, then the number of function evaluations is given as:

$$NE = n \times Itr_{max} \tag{6}$$

In this case, the population size is represented by n, and the maximum iteration is represented by Itr_{max} during the process of iteration. The time was taken after allowable Itr_{max} was reached, and the best solution value found and is used to determine the performance results.

4.1. Unimodal Function

In this part, the objective of the study is to provide an understanding of the impacts of tuning distinguishing factors of the presented FA algorithm on solution and union of the unimodal functions. Also, the proposed algorithm is used in performing the simulation process with the Schwefel's Problem, as follows [23] (refer to Figure 1):

$$f(\boldsymbol{x}) = \sum_{i=1}^{d} |x_i| + \prod_{i=1}^{d} |x_i| \tag{7}$$
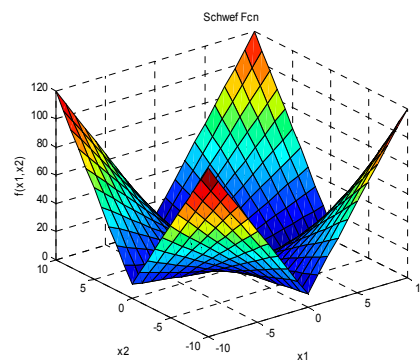

Figure 1. Schwefel's Problem in 3-D view.

Table 1 contains the comparison of the proposed FA method and the original FA in terms of accuracy and the corresponding CPU time.

Table 1. Results of the proposed Firefly algorithm (FA) for Schwefel's Problem.

Variables			The Original FA Method			The Proposed FA Method c = 5		
Population (*n*)	Dimension (*d*)	Itr_{max}	Cost Function $f(x)$		Time (s)	Cost Function $f(x)$		Time (s)
6	2	50	6.58824×10^{-4}	$+/- 2.03 \times 10^{-3}$	0.019	$\mathbf{4.22032 \times 10^{-11}}$	$+/- \mathbf{3.11 \times 10^{-8}}$	0.017649662
		500	6.13740×10^{-5}	$+/- 1.23 \times 10^{-4}$	0.068	$\mathbf{1.88611 \times 10^{-88}}$	$+/- \mathbf{5.23 \times 10^{-53}}$	0.146790572
	30	50	8.11338×10^{1}	$+/- 9.23$	0.017	$\mathbf{2.21701 \times 10^{-8}}$	$+/- \mathbf{4.01 \times 10^{-5}}$	0.017149309
		500	2.10145×10^{1}	$+/- 1.03 \times 10^{1}$	0.076	$\mathbf{3.77617 \times 10^{-88}}$	$+/- \mathbf{6.11 \times 10^{-51}}$	0.150035174
30	2	50	3.93047×10^{-4}	$+/- 5.01 \times 10^{-3}$	0.097	$\mathbf{3.05356 \times 10^{-17}}$	$+/- \mathbf{4.03 \times 10^{-11}}$	0.109078429
		500	7.42205×10^{-5}	$+/- 6.17 \times 10^{-4}$	0.859	$\mathbf{1.5336 \times 10^{-98}}$	$+/- \mathbf{5.12 \times 10^{-75}}$	1.049852353
	30	50	1.79201×10^{1}	$+/- 7.73$	0.106	$\mathbf{3.02683 \times 10^{-9}}$	$+/- \mathbf{1.22 \times 10^{-5}}$	0.119447301
		500	5.70892×10^{-1}	$+/- 9.13 \times 10^{-1}$	0.967	$\mathbf{5.9831 \times 10^{-93}}$	$+/- \mathbf{5.31 \times 10^{-73}}$	1.255127879
100	2	50	8.19849×10^{-4}	$+/- 9.52 \times 10^{-5}$	0.871	$\mathbf{2.66275 \times 10^{-24}}$	$+/- \mathbf{3.31 \times 10^{-16}}$	0.926686182
		500	7.36599×10^{-5}	$+/- 1.33 \times 10^{-4}$	8.388	$\mathbf{2.6357 \times 10^{-119}}$	$+/- \mathbf{1.11 \times 10^{-85}}$	9.85979571
	30	50	5.47774	$+/- 8.33$	0.997	$\mathbf{3.1563 \times 10^{-10}}$	$+/- \mathbf{5.01 \times 10^{-6}}$	1.096175286
		500	4.62718×10^{-1}	$+/- 8.35 \times 10^{-2}$	9.302	$\mathbf{5.0678 \times 10^{-100}}$	$+/- \mathbf{1.73 \times 10^{-83}}$	10.67208132

The results showed that there is a small improvement when population size of 100 is used, compared to when the size of 30 is used. Nevertheless, when the iteration is increased to 500, and the

problem dimension escalated to 30, the computational burden is presented clearly, based on the use of large population size.

4.2. Multimodal Function

The simulations involving the levy function, $f(x)$ as illustrated in Figure 2, as an example of multimodal function are executed to examine the impacts of parameters on solution accuracy of the suggested algorithm, and the parameter on the union. The $f(x)$ is given by:

$$f(\pmb{x}) = sin^2(\pi w_1) + \sum_{i=1}^{d-1}\left[(w_i-1)^2\Big(1+sin^2(\pi w_i+1)\Big)\right] + (w_d-1)^2\Big(1+sin^2(2\pi w_d)\Big) \qquad (8)$$

where

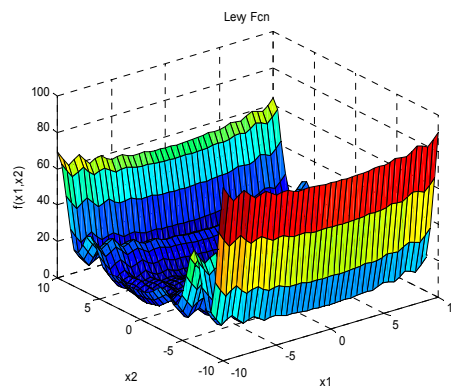$$w_i = 1 + \frac{x_i-1}{4}\ \forall\ i = 1,2,\ldots,d$$

Figure 2. Levy functions in 3-D view.

The comparison of the result is made, and conclusions determined on the most effective parameter condition for the algorithm. In regard to the multimodal problem of the Levy functions, the suggested approach and the result for FA original based on numerical simulation are indicated in Table 2.

Table 2. Results of the proposed FA methods for Levy function problem.

Variables			The Original FA Method			The Proposed FA Method $c = 5$		
Population (n)	Dimension (d)	Itr_{max}	Cost Function $f(x)$		Time (s)	Cost Function $f(x)$		Time (s)
6	2	50	2.456×10^{-1}	$+/- 9.21 \times 10^{-1}$	0.022	**8.70213 × 10⁻⁴**	**+/− 5.63 × 10⁻²**	0.017334055
		500	2.76×10^{-2}	$+/- 7.03 \times 10^{-2}$	0.082	**2.360659 × 10⁻³**	**+/− 3.11 × 10⁻¹**	0.166245785
	30	50	2.32×10^{2}	$+/- 9.13 \times 10^{1}$	0.022	**3.25949207**	**+/− 8.75 × 10¹**	0.023739867
		500	2.15×10^{2}	$+/- 5.21 \times 10^{1}$	0.118	**2.586502461**	**+/− 7.05 × 10¹**	0.195115349
30	2	50	3.64×10^{-2}	$+/- 7.03 \times 10^{-2}$	0.128	**1.053531 × 10⁻³**	**+/− 3.51 × 10⁻²**	0.112729302
		500	7.68×10^{-4}	$+/- 5.19 \times 10^{-4}$	1.184	**1.22458 × 10⁻⁴**	**+/− 1.33 × 10⁻³**	1.038401091
	30	50	2.06×10^{2}	$+/- 9.25 \times 10^{1}$	0.133	**1.989531603**	**+/− 8.23**	0.135518497
		500	1.56×10^{2}	$+/- 9.72 \times 10^{1}$	1.163	**1.815948116**	**+/− 7.33**	1.421920632
100	2	50	4.73×10^{-3}	$+/- 8.78 \times 10^{-3}$	0.903	**1.24 × 10⁻⁴**	**+/− 7.12 × 10⁻³**	0.958555
		500	8.84×10^{-4}	$+/- 6.15 \times 10^{-4}$	8.501	**4.12474 × 10⁻⁵**	**+/− 2.33 × 10⁻⁵**	9.402039477
	30	50	2.22×10^{2}	$+/- 9.25 \times 10^{2}$	1.356	**1.94852802**	**+/− 7.77**	1.120368893
		500	1.33×10^{2}	$+/- 7.88 \times 10^{2}$	9.971	**1.797004014**	**+/− 5.53**	11.20427867

The proposed algorithm showed good result to solve the problem and outperforms the original FA method. The result further noted that a small enhancement could be explicitly observed on population

sizes of 30 and 100 when high population sizes are used. Also, as the time of computation increased proportionally with the population size.

4.3. Other Optimization Methods

In this subsection, we present the performance of the proposed firefly algorithm with other optimization approaches, such as Standard particle swarm optimization (SPSO) [24], Differential Evolution (DE) [9], [25] Variable Step Size Firefly Algorithm (VSSFA) [26], Memetic Firefly Algorithm (MFA) [27] and original FA, in global numeric optimization problem. Well-known classical benchmark problems used in the following experiments for studying and comparing the performance of proposed optimization method with other methods in this section [23,28]. Specifically, we have used six well-defined objective functions to test the proposed Firefly method. Table 3 presents the selected problem sets. All these functions are continuous. For more details, please refer to [23,28].

Table 3. Benchmark problem sets.

Function's Number	Name	Expression
F1	Ackley Function	$f_1(x) = \left[-20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^{n} (x_i^2)} \right) - \exp\left(\frac{1}{n} \sum_{i=1}^{n} \cos(2\pi x_i) \right) + 20 + e \right],$ $-32 \leq x_i \leq 32$ $\min(f_1) = f_1(0,\ldots,0) = 0$
F2	Sphere Function	$f_2(x) = \sum_{i=1}^{n} (x_i)^2, \; -100 \leq x_i \leq 100$ $\min(f_2) = f_2(0,\ldots,0) = 0$
F3	Rosenbrock Function	$f_3(x) = \sum_{i=1}^{n-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right], \; -10 \leq x_i \leq 10$ $\min(f_3) = f_3(1,\ldots,1) = 0$
F4	Rastrigin Function	$f_4(x) = \sum_{i=1}^{n} \left(x_i^2 - 10\cos(2\pi x_i) + 10 \right), \; -5.12 \leq x_i \leq 5.12$ $\min(f_4) = f_4(0,\ldots,0) = 0$
F5	Schwefel Problem 2.22	$f_5(x) = \sum_{i=1}^{n} |x_i| + \prod_{i=1}^{n} |x_i|, \; -10 \leq x_i \leq 10$ $\min(f_5) = f_5(0,\ldots,0) = 0$
F6	Griewank Function	$f_6(x) = \frac{1}{4000} \sum_{i=1}^{n} (x_i^2) - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}} \right) + 1, \; -600 \leq x_i \leq 600$ $\min(f_6) = f_6(0,\ldots,0) = 0$

In this section, we consider the dimension of the problems of 30 ($d = 30$). For fair comparison, we use the same population size ($n = 30$) and the number of function evaluations of 15×10^3. For each benchmark function, the mean value of 30 Monte Carlo simulations of each algorithm has been reported. Nevertheless, we take $\beta_0 = 1$, $\gamma = 1$ and $\alpha = 0.2$ for the FA and VSSFA methods as in [26]. For MFA method [27], we use the following setting: $\beta_{min} = 0.2$, $\beta_0 = 1$, $\gamma = 1$ and $\alpha = 0.2$. For the SPSO and DE methods, we have used the following setting parameters as in [25]. Moreover, we define a weighting factor (f = 0.5), and a crossover constant (CR) = 0.5 for DE method. While for SPSO, we set the inertial constant $w = \frac{1}{2\log(2)}$, a cognitive constant $c_1 = 0.5 + \log(2)$, and a social constant for swarm interaction $c_2 = 0.5 + \log(2)$ as in [24].

Table 4 shows the numerical result values of the proposed FA method compared with other methods on the different problem sets. As illustrated in Table 4, the proposed FA method obtained better results than the other five methods on six benchmark problem sets (F1–F6). The proposed FA method obtained the target solution on problem set F6, although the other five methods did not as well as the proposed FA method, they obtained the comparable results to each other as well. Furthermore, it is clearly that the VSSFA method trapped and failed to achieve reasonable solutions on all problem sets.

It is also worthwhile to study the proposed FA method with various values of c parameter. Thus, Table 5 presents the performance of the proposed FA method with different values of c. It is obvious that the performance of the proposed FA method improves more consistently as c increases, and is better than that of the FA method. However, the proposed FA method significantly outperforms the other five algorithms for achieving reasonable solutions on all benchmark problem sets.

Table 4. Results obtained by the standard FA, Variable Step Size Firefly Algorithm (VSSFA), Memetic Firefly Algorithm MFA, particle swarm optimization (PSO), differential evolution (DE), and the proposed FA on the test suite.

Function	Standard FA		VSSFA		MFA		SPSO		DE		The proposed FA with $c = 5$	
	Mean		Mean		Mean		Mean		Mean		Mean	
F1	3.21×10^{-2}	$+/- 1.09 \times 10^{-2}$	2.31×10^{1}	$+/- 4.5 \times 10^{1}$	7.31×10^{-4}	$+/- 8.55 \times 10^{-3}$	8.75×10^{0}	$+/- 9.77 \times 10^{0}$	1.03×10^{1}	$+/- 4.1 \times 10^{1}$	0	$+/- 0$
F2	7.87×10^{1}	$+/- 5.3 \times 10^{1}$	6.83×10^{1}	$+/- 9.21 \times 10^{1}$	2.71×10^{-4}	$+/- 3.34 \times 10^{-4}$	4.31×10^{-4}	$+/- 5.33 \times 10^{-4}$	5.11×10^{1}	$+/- 4.1 \times 10^{1}$	1.12×10^{-185}	$+/- 7.03 \times 10^{-123}$
PF3	5.71×10^{-2}	$+/- 9.55 \times 10^{-2}$	$4.75 \times 10^{+3}$	$+/- 7.1 \times 10^{3}$	3.95×10^{1}	$+/- 8.3 \times 10^{1}$	5.31×10^{0}	$+/- 5.2 \times 10^{0}$	7.71×10^{1}	$+/- 8.5 \times 10^{1}$	3.30×10^{-187}	$+/- 3.65 \times 10^{-153}$
F4	5.12×10^{2}	$+/- 8.7 \times 10^{1}$	4.23×10^{2}	$+/- 7.1 \times 10^{2}$	3.53×10^{1}	$+/- 2.3 \times 10^{1}$	1.92×10^{3}	$+/- 2.7 \times 10^{3}$	2.13×10^{4}	$+/- 2.2 \times 10^{4}$	28.88×10^{0}	$+/- 2.01 \times 10^{-2}$
F5	4.87×10^{-2}	$+/- 7.23 \times 10^{-2}$	9.28×10^{1}	$+/- 9.3 \times 10^{1}$	8.01×10^{-4}	$+/- 7.33 \times 10^{-4}$	1.95×10^{0}	$+/- 8.7 \times 10^{0}$	7.27×10^{1}	$+/- 1.3 \times 10^{1}$	5.98×10^{-93}	$+/- 3.25 \times 10^{-78}$
F6	1.41×10^{-3}	$+/- 9.88 \times 10^{-3}$	8.75×10^{1}	$+/- 9.2 \times 10^{1}$	5.33×10^{-3}	$+/- 7.35 \times 10^{-3}$	7.72×10^{-3}	$+/- 8.75 \times 10^{-3}$	8.52×10^{0}	$+/- 9.6 \times 10^{0}$	1.48×10^{-5}	$+/- 8.22 \times 10^{-4}$

Table 5. Results obtained by the standard FA and the proposed FA with different values of c on the test suite.

Function	Standard FA		Proposed FA $c = 1$		Proposed FA $c = 2$		Proposed FA $c = 3$		Proposed FA $c = 5$	
	Mean		Mean		Mean		Mean		Mean	
F1	3.21×10^{-2}	$+/- 1.09 \times 10^{-2}$	0	$+/- 0$	0	$+/- 0$	0	$+/- 0$	0	$+/- 0$
F2	7.87×10^{1}	$+/- 5.3 \times 10^{1}$	2.72×10^{-48}	$+/- 4.11 \times 10^{-23}$	1.93×10^{-112}	$+/- 3.33 \times 10^{-83}$	2.85×10^{-155}	$+/- 1.71 \times 10^{-95}$	1.12×10^{-185}	$+/- 7.03 \times 10^{-123}$
F3	5.71×10^{-2}	$+/- 9.55 \times 10^{-2}$	1.51×10^{-48}	$+/- 2.57 \times 10^{-25}$	2.87×10^{-113}	$+/- 7.23 \times 10^{-92}$	7.25×10^{-151}	$+/- 2.22 \times 10^{-105}$	3.30×10^{-187}	$+/- 3.65 \times 10^{-153}$
F4	5.12×10^{2}	$+/- 8.7 \times 10^{1}$	28.86×10^{0}	$+/- 2.00 \times 10^{-2}$	28.86×10^{0}	$+/- 2.00 \times 10^{-2}$	28.87×10^{0}	$+/- 2.01 \times 10^{-2}$	28.88×10^{0}	$+/- 2.01 \times 10^{-2}$
F5	4.87×10^{-2}	$+/- 7.23 \times 10^{-2}$	2.21×10^{-24}	$+/- 8.11 \times 10^{-15}$	1.66×10^{-58}	$+/- 5.88 \times 10^{-37}$	2.52×10^{-78}	$+/- 6.21 \times 10^{-52}$	5.98×10^{-93}	$+/- 3.25 \times 10^{-78}$
F6	1.41×10^{-3}	$+/- 9.88 \times 10^{-3}$	1.26×10^{-4}	$+/- 3.83 \times 10^{-3}$	2.48×10^{-4}	$+/- 8.57 \times 10^{-3}$	1.47×10^{-4}	$+/- 4.51 \times 10^{-4}$	1.48×10^{-5}	$+/- 8.22 \times 10^{-4}$

4.4. Computational Complexity

In this subsection, we obtained the average CPU time to evaluate the computational complexity of each method. For each method, the computational complexity based on the CPU time for each approach is defined as follows [20],

$$EFF = \frac{T_c(F)}{T_{tot}(F)} \times 100\% \tag{9}$$

where EFF represents the computational efficiency for each method, $T_c(F)$ denotes the computational time of an algorithm on the benchmark problem set F and $T_{tot}(F)$ is the total time of all the methods on the problem set F.

As shown in Table 6, DE and the standard FA methods obtain the best results among all other methods, and they are followed by MFA method. Furthermore, the computational efficiency of the presented FA method is comparable to SPSO method. However, the proposed method is worse than the standard FA method in terms of computational complexity because of the new exploration and exploitation approach have been used.

Table 6. The computational efficiency (EEF) of each method on the six problem sets.

Functions	Standard FA	VSSFA	MFA	SPSO	DE	The Proposed FA with $c = 5$
F1	10%	25%	11%	22%	9%	23%
F2	11%	26%	12%	20%	10%	21%
F3	10%	23%	12%	20%	11%	24%
F4	10%	25%	11%	22%	9%	23%
F5	9%	26%	11%	20%	10%	24%
F6	11%	24%	12%	20%	10%	23%

4.5. Statistical Test

In this subsection, we employ the typical non-parametric tests; namely, the Quade, Friedman, and Aligned Friedman tests [29,30], as a methodology for comparing and evaluating the results of all the algorithms in this paper. Table 6 shows the average rankings obtained from the non-parametric test. Each method and its score are reported in descending order. Furthermore, we reported the statistics and the corresponding *p*-values of the tests at the bottom of the Table 7.

Table 7. Descending ranks of all compared methods over all problem sets.

Average	Quade		Friedman		Aligned Friedman	
Rank	**Method**	**Score**	**Method**	**Score**	**Method**	**Score**
1	VSSFA	7.6235	VSSFA	7.5026	VSSFA	82.5832
2	SPSO	6.8235	SPSO	6.8125	SPSO	75.1472
3	DE	6.5121	DE	6.5011	DE	72.1322
4	FA	6.4157	FA	6.4113	FA	68.3582
5	MFA	5.5201	MFA	5.1090	MFA	52.4235
6	Proposed FA	3.1052	Proposed FA	3.3075	Proposed FA	38.8885
Statistic	4.011302		35.2354		12.75773	
***p*-value**	0.0056273		0.000271		0.367715	

5. Discussion

This paper examines the newly proposed improvements regarding FA method. By including effective adaptive parameters relies on the iteration processes to enhance the exploration and exploitation mechanisms in the original FA method. The proposed method is exhibited to be effective in getting satisfactory results since it aids in providing an equilibrium on exploitation and exploration

abilities. In the proposed FA, the fireflies fly in the sky to find food/prey (i.e., best solutions). Two parameters are: The step size (α) that acts in a similar role as the cooling schedule in the traditional simulated annealing optimization method, and the absorption coefficient (β) that regulates the attractiveness. The appropriately update for the step size (α) and absorption coefficient (β) balances the exploration and exploitation behavior of each firefly, respectively. As the attractiveness usually decrease once a firefly has found its prey/solution, the light intensity increases in order to raise the attack accuracy. The results found in this paper provides a conclusion on apprehending the impacts of tuning distinguishing features of the proposed FA algorithm on their accuracy of solution and union. The multimodal Levy function problem and the unimodal Schwefel's Problem 2.22 were used in the simulation execution along with other benchmark functions. It was noted that a large number of repetitions yielded a better-quality solution. Also, the vast iteration is needed and essential in comparing the algorithm scrutinized. It is because all the algorithms proposed have shown good capability in obtaining better global extreme value by overlapping the iteration. The quality of the solution can also be increased by increasing the population size, and this will also increase the time for computation. Consequently, a competitive proportion of the population is applied in comparing the algorithm proposed to determine the best result. Moreover, the proposed method is worse than the standard FA method in terms of computational complexity because of the new exploration and exploitation approach have been used. Nevertheless, with the advent of GPUs (see Nvidia.com, e.g.), multi-thread excursions, and cloud computing, metrics based on the timing of a single CPU are no longer relevant. The critical and most important metric is really performance quality of the algorithm.

6. Conclusions

This paper proposed an improved meta-heuristic Firefly method for optimization problems. A novel type of FA model has been presented, and an improvement is applied to exchange information between fireflies during the process of the light intensity updating. This new method can enhance the performance of the original firefly method rate without losing the speed convergence of the basic FA. The detailed implementation procedure for this proposed method is also described. Compared with the basic FA, PSO, DE, and two other FA variants, the simulation results illustrate that this approach is a feasible and effective way in numerical optimization problems. Regarding the results found in the unimodal problem, as the dimensions of the problem increase, the population size also increases, which was better than the original FA. Furthermore, due to increased iteration, the proposed algorithm has obtained important enhancement. For the case of multimode study, the proposed algorithm obtains a good improvement in the accuracy of the solution with an increase in dimension and population size. Our future work will focus on developing a new meta-hybrid approach to solve optimization problem.

References

1. Yang, X.-S. *Engineering Optimization: An Introduction with Metaheuristic Applications*; John Wiley & Sons: Hoboken, NJ, USA, 2010.
2. Tang, W.J.; Wu, Q.H. Biologically inspired optimization: A review. *Trans. Inst. Meas. Control* **2009**, *31*, 495–515. [CrossRef]
3. Simon, D. *Evolutionary Optimization Algorithms*; John Wiley & Sons: Hoboken, Hoboken, NJ, USA, 2013.
4. Singiresu, S.R. *Engineering Optimization Theory and Practice*, 4th ed.; John Wiley & Sons: Hoboken, NJ, USA, 2009.
5. Talbi, E.G. *Metaheuristics: From Design to Implementation*; John Wiley & Sons: Hoboken, NJ, USA, 2009.

6. Castro, L.N. *Nature-Inspired Computing Design, Development, and Applications*; IGI Global: Hershey, PA, USA, 2012.
7. Blum, C.; Roli, A. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Comput. Surv.* **2003**, *35*, 268–308. [CrossRef]
8. Russell, C.E.; Shi, Y. Comparison between genetic algorithms and particle swarm optimization. In Proceedings of the Evolutionary Programming VII, 7th International Conference, San Diego, CA, USA, 25–27 March 1998.
9. Das, S.; Suganthan, P.N. Differential evolution: A survey of the state-of-the-art. *IEEE Trans. Evol. Comput.* **2010**, *15*, 4–31. [CrossRef]
10. Albataineh, Z.; Salem, F.; Ababneh, J.I. Linear phase FIR low pass filter design using hybrid differential evolution. *Int. J. Res. Wirel. Syst.* **2012**, *1*, 43–49.
11. Yang, X.-S.; Deb, S. Cuckoo search: Recent advances and applications. *Neural Comput. Appl.* **2014**, *9*, 169–174. [CrossRef]
12. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95—International Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995.
13. Shi, Y.; Eberhart, R.C. Parameter selection in particle swarm optimization. In Proceedings of the 7th International Conference on Evolutionary Programming VII, San Diego, CA, USA, 25–27 March 1998.
14. Yang, X.-S.; He, X. Firefly algorithm: Recent advances and applications. *Int. J. Swarm Intell.* **2013**, *1*, 36–50. [CrossRef]
15. Binitha, S.; Sathya, S.S. A survey of bio inspired optimization algorithms. *Int. J. Soft Comput. Eng.* **2012**, *2*, 137–151.
16. Yang, X.-S. Firefly algorithms for multimodal optimization. In Proceedings of the 5th International Conference on Stochastic Algorithms: Foundations and Applications, Sapporo, Japan, 26–28 October 2009.
17. Wang, H.; Wang, W.; Sun, H.; Zhao, J.; Zhang, H.; Liu, J.; Zhou, X. A new firefly algorithm with local search for numerical optimization. In Proceedings of the Computational Intelligence and Intelligent Systems: 7th International Symposium (ISICA), Guangzhou, China, 21–22 November 2015.
18. Wang, H.; Wang, W.; Sun, H.; Rahnamayan, S. Firefly algorithm with random attraction. *Int. J. Bio-Inspired Comput.* **2016**, *8*, 33–41. [CrossRef]
19. Wang, G.-G.; Guo, L.; Duan, H.; Wang, H. A new improved firefly algorithm for global numerical optimization. *J. Comput. Theor. Nanosci.* **2014**, *11*, 477–485. [CrossRef]
20. Cheung, N.J.; Ding, X.M.; Shen, H.B. Adaptive firefly algorithm: Parameter analysis and its application. *PLoS ONE* **2014**, *9*, e112634. [CrossRef] [PubMed]
21. Yang, X.-S. Firefly algorithm, stochastic test functions and design optimization. *Int. J. Bio-Inspired Comput.* **2010**, *2*, 78–84. [CrossRef]
22. Engelbrecht, A.P. Heterogeneous particle swarm optimization. In Proceedings of the International Conference on Swarm Intelligence, Brussels, Belgium, 8–10 September 2010.
23. Virtual Library of Simulation Experiments: Test Functions and Database. Available online: https://www.sfu.ca/~ssurjano/ (accessed on 20 March 2018).
24. Clerc, M. Standard Particle Swarm Optimization. Available online: http://clerc.maurice.free.fr/pso/SPSO_descriptions.pdf (accessed on 10 March 2018).
25. Albataineh, Z.; Salem, F. New blind multiuser detection in DS-CDMA using H-DE and ICA algorithms. In Proceedings of the International Conference on 2013 4th Intelligent Systems Modelling & Simulation (ISMS), Bangkok, Thailand, 29–31 January 2013.
26. Yu, S.H.; Zhu, S.L.; Ma, Y.; Mao, D.M. A variable step size firefly algorithm for numerical optimization. *Appl. Math. Comput.* **2015**, *263*, 214–220. [CrossRef]
27. Fister, I., Jr.; Yang, X.S.; Fister, I.; Brest, J. Memetic firefly algorithm for combinatorial optimization. In *Bioinspired Optimization Methods and their Applications (BIOMA 2012)*; Filipic, B., Silc, J., Eds.; Jozef Stefan Institute: Ljubljana, Slovenia, 2012.
28. Wang, H.; Rahnamayan, S.; Sun, H.; Omran, M.G.H. Gaussian bare-bones differential evolution. *IEEE Trans. Cybern.* **2013**, *43*, 634–647. [CrossRef] [PubMed]

29. Derrac, J.; García, S.; Molina, D.; Herrera, F. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm Evol. Comput.* **2011**, *1*, 3–18. [CrossRef]

30. Chih, M. Three pseudo-utility ratio-inspired particle swarm optimization with local search for multidimensional knapsack problem. *Swarm Evol. Comput.* **2018**, *39*, 279–296. [CrossRef]