Research Article
Modified Firefly Algorithm

Surafel Luleseged Tilahun and Hong Choon Ong

School of Mathematical Sciences, Universiti Sains Malaysia, 11800 Minden, Penang, Malaysia

Correspondence should be addressed to Surafel Luleseged Tilahun, surafelaau@yahoo.com

Firefly algorithm is one of the new metaheuristic algorithms for optimization problems. The algorithm is inspired by the flashing behavior of fireflies. In the algorithm, randomly generated solutions will be considered as fireflies, and brightness is assigned depending on their performance on the objective function. One of the rules used to construct the algorithm is, a firefly will be attracted to a brighter firefly, and if there is no brighter firefly, it will move randomly. In this paper we modify this random movement of the brighter firefly by generating random directions in order to determine the best direction in which the brightness increases. If such a direction is not generated, it will remain in its current position. Furthermore the assignment of attractiveness is modified in such a way that the effect of the objective function is magnified. From the simulation result it is shown that the modified firefly algorithm performs better than the standard one in finding the best solution with smaller CPU time.

1. Introduction

An optimization problem is a problem of finding values for the variables of a function to optimize the function. These kinds of problems exist in many disciplines [1–4]. Whenever a decision needs to be made and the problem is formulated using mathematical terms, optimization solution methods will be used to solve the formulated problem. The solution methods exist depending on the behavior of the problem. For example, if both the objective function and the functions which construct the feasible region are linear, it is called a linear programming problem, and methods like simplex algorithm will be used to solve it [5]. Some of the solution methods depend on the derivatives of the objective function [6, 7]. Even though there are many solution methods, there are many problems which need special attention and are hard to solve using the deterministic solution methods.

Metaheuristic algorithms are optimization algorithms which try to improve the quality of solution members iteratively with some randomness properties. Most of these algorithms

are inspired by biological aspects. Unlike deterministic solution methods metaheuristic algorithms are not affected by the behavior of the optimization problem [8]. This makes the algorithm to be used widely in different fields [9–11]. Since the introduction of genetic algorithm in 1975, many metaheuristic algorithms are introduced [12]. Firefly algorithm is one of the new algorithms introduced in 2009 by Yang [13]. The algorithm mimics the behavior of fireflies. It has three ideal rules. One among these rules is that a firefly will follow the brighter firefly, and if there is no brighter firefly, it will move randomly.

In this paper we modify this random movement of the brighter firefly, and we also modify the way how the attractiveness, which is used to compute the brightness, is computed. Simulation is also done on selected benchmark problems. From the simulation result it is shown that the modified firefly algorithm performs better than the standard firefly algorithm. In the next section basic concepts will be discussed followed by a discussion on the modified firefly algorithm in Section 3. Simulation results will be presented in Section 4 followed by the conclusion in Section 5.

2. Basic Concepts

2.1. Optimization Problem

An optimization problem has basically three components: a function to optimize, possible solution set to choose a value for the variable from, and the optimization rule, which will be either maximized or minimized. Since one can switch between minimization and maximization problems by multiplying the objective function by negative one, analyzing either minimization or maximization problem is enough. A typical maximization problem can be given as follows:

$$
\begin{aligned}
\max \quad & f(x), \\
\text{s.t.} \quad & x \in S \subseteq \mathfrak{R}^n.
\end{aligned}
\tag{2.1}
$$

A solution for this problem is a member of set S which gives the maximum value of $f(x)$ compared to all elements in S. Mathematically, x^* is a solution if and only if $x^* \in S$ and $f(x^*) \geq f(x)$, for all $x \in S$.

The aim of all solution methods and algorithms is to find such x^*.

2.2. Firefly Algorithm

Firefly algorithm is a bio-inspired metaheuristic algorithm for optimization problems. It was introduced in 2009 at Cambridge University by Yang [14]. The algorithm is inspired by the flashing behavior of fireflies at night. One of the three rules used to construct the algorithm is that all fireflies are unisex, which means any firefly can be attracted to any other brighter one. The second rule is that the brightness of a firefly is determined from the encoded objective function. The last rule is that attractiveness is directly proportional to brightness but decreases with distance, and a firefly will move towards the brighter one, and if there is no brighter one, it will move randomly.

From elementary physics it is clear that the intensity of light is inversely proportional to the square of the distance, say r, from the source. Furthermore when light passes through

a medium with light absorption coefficient of λ the light intensity, I varies with distance r as given below:

$$I(r) = I_0 e^{-\lambda r},\tag{2.2}$$

where I_0 is the intensity at the source point.

These can be combined as

$$I(r) = I_0 e^{-\lambda r^2}.\tag{2.3}$$

However, computationally computing $1/(1 + \lambda r^2)$ is easier than computing $e^{-\lambda r^2}$ [14]. Hence the intensity can be calculated using

$$I(r) = \frac{I_0}{1 + \lambda r^2}.\tag{2.4}$$

Similarly, the attractiveness of a firefly can be defined as follows:

$$A(r) = \frac{A_0}{1 + \lambda r^2},\tag{2.5}$$

where A_0 is the attractiveness at $r = 0$.

If a firefly located at $x' = (x'_1, x'_2, \ldots, x'_n)$ is brighter than another firefly located at $x = (x_1, x_2, \ldots, x_n)$, the firefly located at x will move towards x'. The location update of the firefly located at x will be done as follows:

$$x := x + A_0 e^{-\lambda r^2}(x' - x) + \alpha\varepsilon.\tag{2.6}$$

The last term is a randomization term with α being a randomization parameter with $0 \leq \alpha \leq 1$ and ε vector of random numbers, whereas the second term is due to the attraction of x towards x'. According to [13] for practical uses A_0 can be taken as one, $A_0 = 1$.

The algorithm can be summarized as follows.

(1) Generate a random solution set, $\{x_1, x_2, \ldots, x_k\}$.

(2) Compute intensity for each solution member, $\{I_1, I_2, \ldots, I_k\}$.

(3) Move each firefly i towards other brighter fireflies, and if there is no other brighter firefly, move it randomly.

(4) Update the solution set.

(5) Terminate if a termination criterion is fulfilled otherwise go back to step 2.

3. Modified Firefly Algorithm

The brightest firefly is a firefly with current global best solution. If this brightest firefly moves randomly as in the standard firefly algorithm, its brightness may decrease depending on the direction. This leads to the decrement of the performance of the algorithm in that particular iteration. However, if this brightest firefly is allowed to move only if in a direction in which its brightness improves, it will not decrease the performance of the algorithm in terms of global best solution for the solutions in that particular iteration. The modification proposed in this paper is as follows. To determine the movement direction of the brightest firefly, randomly generate m unit vectors, say $u_1, u_2, \ldots, u_m$. Then choose a direction, U, among the randomly generated m directions in which the brightness of the brightest firefly increases if the firefly moves in that direction. Hence, the movement of the brightest firefly can be described as follows:

$$x := x + \alpha U, \tag{3.1}$$

where α is a random step length.

If such direction does not exist among the randomly generated solutions, the brightest firefly will stay in its current position.

Furthermore, rather than taking $A_0^i = 1$ for each firefly i, it is better to assign a source attractiveness which depends on the intensity of the firefly, which in turn depends on the objective function. One possible way to do so is to assign the ratio of the intensity of fireflies. Suppose firefly i, located at x', is brighter than a firefly j, which is located at x. Then the firefly located at x will move towards firefly i, as given in (2.6), but A_0 is given by

$$A_0 = \frac{I_0'}{I_0}, \tag{3.2}$$

where I_0' is the intensity at $r = 0$ for firefly i and I_0 is the intensity at $r = 0$ for firefly j and $I_0 \neq 0$. For practical cases, to eliminate the singularity case when $I_0 = 0$, A_0 can also be described as $e^{I_0' - I_0}$.

If we take $A_0 = I_0'$ and if the intensity is large, the movement of firefly j towards i may be long. However depending on the solution space it is better to adjust A_0. In either case it should be directly proportional to the intensity at the source, I_0'.

4. Simulation Results

To test the performance of modified firefly algorithm and compare it with the standard firefly algorithm, seven test problems are selected. The test functions include stochastic, multimodal, unimodal, continuous, and discontinuous benchmark problems. For each test problem the same randomly generated initial solution set is used for both the modified and the standard firefly algorithms with absorption coefficient, λ, equaling 2, and the number of random directions for local search in the case of modified firefly algorithm is 20. Furthermore the iteration number is set to be 500.
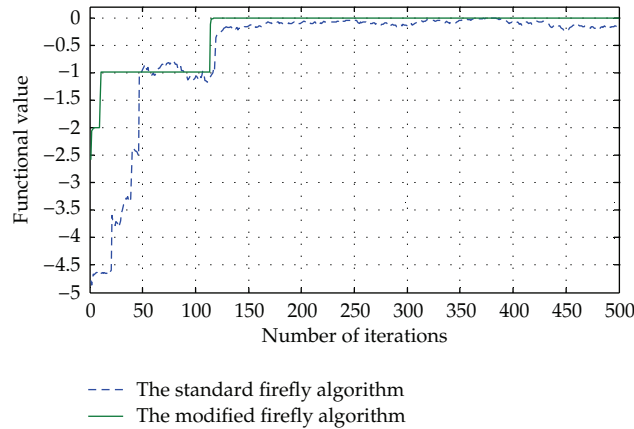
Figure 1: Simulation result of the first test problem.

4.1. The First Test Problem

The objective function of the first test problem used for the simulation purpose is the negative of second function of De Jong, and the problem is given as in [15]:

$$\max \quad f_1 = -\left(20 + \left(x_1^2 - 10\cos 2\pi x_1\right) + \left(x_2^2 - 10\cos 2\pi x_2\right)\right),$$
$$\text{s.t.} \quad -2.048 \leq x_1, \quad x_2 \leq 2.048. \tag{4.1}$$

It is one of the well-known benchmark optimization problems with many local optimums and a unique global optimum of 0 at (0, 0).

The modified firefly algorithm and the standard firefly algorithm are used to solve this maximization problem with $\alpha = 1.2$, and the result is recorded as shown in Figure 1.

After running the program the solutions found using the standard firefly algorithm and the modified firefly algorithm are -0.1239 and -0.0001 at $(-0.0207, 0.0140)$ and $(0.0000, 0.0005)$, respectively.

4.2. The Second Test Problem

The second test problem used is based on Esom's function [16], and the problem is given as

$$\max \quad f_2 = (\cos x_1 \cos x_2)e^{-(x_1-\pi)^2-(x_2-\pi)^2},$$
$$\text{s.t.} \quad -20 \leq x_1, \ x_2 \leq 20. \tag{4.2}$$

This is an optimization problem with a unique global solution. However, the behavior of the objective function with wide flat surface and an optimum of 1 at (π, π) makes it a hard optimization problem.

The result after running the simulation, with $\alpha = 10$, is recorded in Figure 2.

According to the simulation result, the standard firefly algorithm gives a result of 0.9782 at $(3.2044, 3.0378)$, whereas the modified firefly algorithm gives 1 at $(3.1420, 3.1413)$.
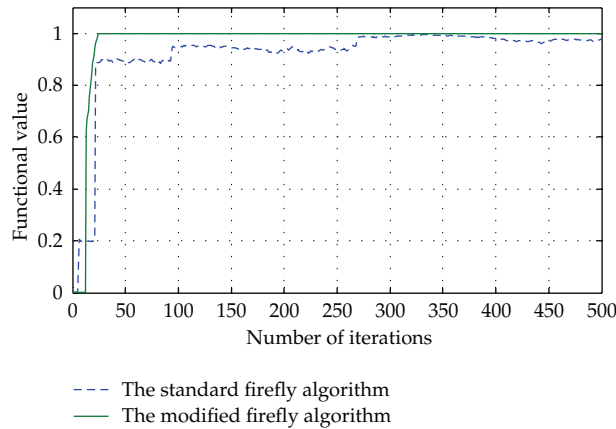
Figure 2: Simulation result of the second test problem.

4.3. The Third Test Problem

The third test problem is a discontinuous test problem taken from [17]. It is given as follows:

$$
\max \quad f_3 = \sum_{i=1}^{2} \mathrm{int}(x_i),
$$

$$
\text{s.t.} \quad -5.12 \le x_1, \quad x_2 \le 5.12.
$$

(4.3)

The optimum solution is 10 and is found when $x_i \ge 4.5$, for $i = 1, 2$. If a suitable step length is not assigned, many algorithms tend to be trapped on one of the flat surfaces rather than moving towards the optimal solution.

After the 500 iterations the final functional value for the two algorithms is found to be 10 at $(5.1149, 5.0934)$ for the standard firefly algorithm and $(5.1200, 5.1200)$ for the modified firefly algorithm. The algorithm parameter α is set to be 2.4. The performance graph of the two algorithm overlaps as shown in **Figure 3**.

4.4. The Fourth Test Problem

The fourth test problem is taken from Mishra [18]. It is given as follows

$$
\max \quad f_4 = \ln\left\{ \left[\left(\sin\left(\cos x_1 + \cos x_2\right)^2 \right)^2 - \left(\cos\left(\sin x_1 + \sin x_2\right)^2 \right)^2 + x_1 \right]^2 \right\}
$$

$$
- 0.1\left((x_1 - 1)^2 + (x_2 - 1)^2 \right),
$$

$$
\text{s.t.} \quad -10 \le x_1, \quad x_2 \le 10.
$$

(4.4)

This is a multimodal function with optimal functional value of 2.28395 at $(2.8863, 1.8233)$.
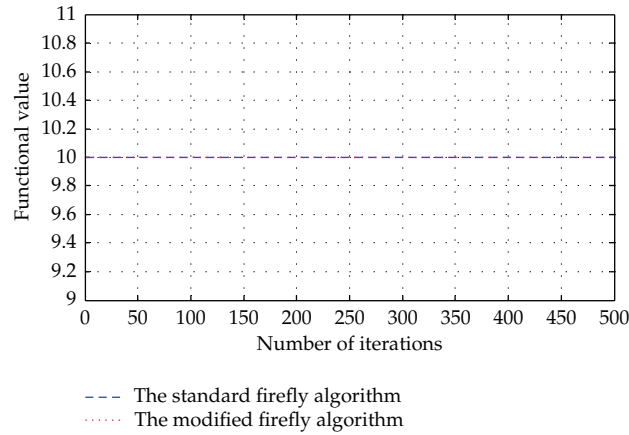
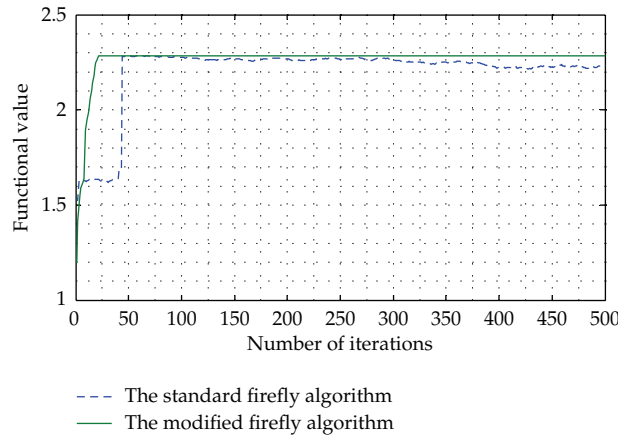Figure 3: Simulation result of the third test problem.

Figure 4: Simulation result of the fourth test problem.

The simulation result for the fourth test problem, with $\alpha = 5$, is given in Figure 4.

From the simulation result the final functional value for the standard and modified firefly algorithm is found to be 2.2308 at (2.8765, 1.9518) and 2.2839 at (2.8864, 1.8234), respectively.

4.5. The Fifth Test Problem

The fifth test problem is a stochastic test. Owing to the random noise parameter it is unlikely to get the same result at a point in different iterations [19]. It is given as follows:

$$\max \quad f_5 = 5e^{-(x_1-\pi)^2-(x_2-\pi)^2} + \sum_{j=1}^{2}\sum_{i=1}^{2}\varepsilon_{ij}e^{-(x_1-i)^2-(x_2-j)^2},$$

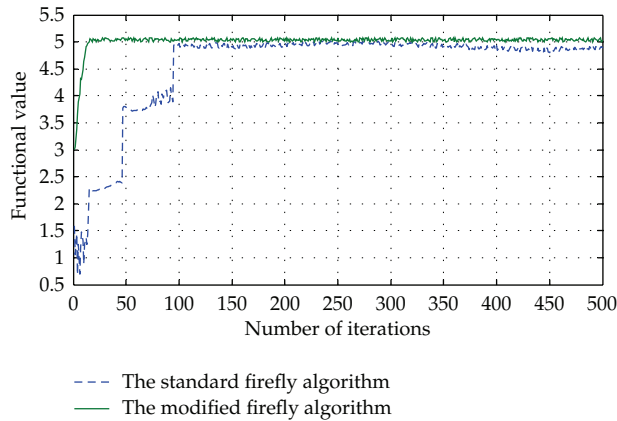$$\text{s.t.} \quad 0 \leq x_1, \quad x_2 \leq 10, \tag{4.5}$$

Figure 5: Simulation result of the fifth test problem.

where ε_{ij} is a random number between 0 and 1. The optimal value is found at (π, π) with the functional value between 5 and 5.0794 depending on the random number ε_{ij}.

4.8828 and 5.0279 are the final functional values obtained after the simulation for the standard and modified firefly algorithm at $(2.9504, 3.1789)$ and $(3.1455, 3.1368)$, respectively. Figure 5 shows the functional values for the two algorithms in the 500 iterations with $\alpha = 2.4$.

4.6. The Sixth Test Problem

The sixth test problem is a maximization problem with an objective function called the Styblinski function [20]. It is multimodal problem with optimum solution of 358.3323 at $(-2.9035, -2.9035)$. It is given as follows:

$$\max \quad f_6 = 280 - \left(\frac{x_1^4 - 16x_1^2 + 5x_1}{2} + \frac{x_2^4 - 16x_2^2 + 5x_2}{2} \right),$$

$$\text{s.t.} \quad -5 \leq x_1, \quad x_2 \leq 5. \tag{4.6}$$

After the simulation the final functional value for the standard and modified firefly algorithm is found to be 358.2775 at $(-2.9495, -2.8714)$ and 358.3323 at $(-2.9040, -2.9040)$, respectively. Like the previous simulation α is set to be 2.4. Figure 6 shows the performance of the two algorithms in the 500 iterations.
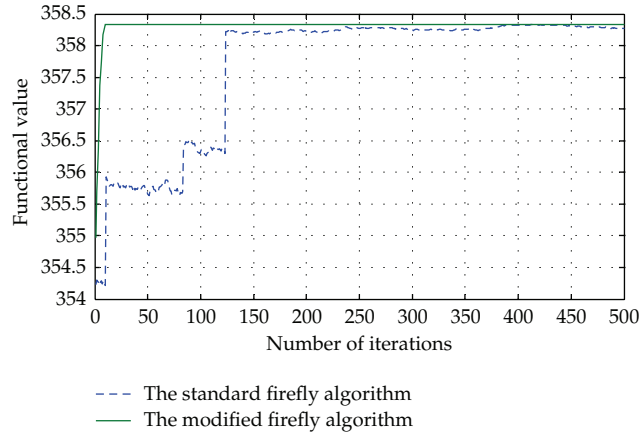
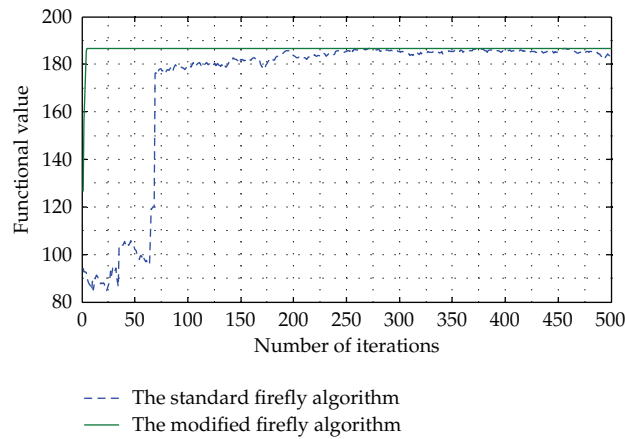**Figure 6:** Simulation result of the sixth test problem.

Figure 7: Simulation result of the seventh test problem.

4.7. The Seventh Test Problem

The objective function of the last test problem is taken to be the negative of Shubert's function. It is a multimodal test function. It is given as

$$
\max \quad f_7 = -\left[\sum_{i=1}^{5} i\cos((i+1)x_1 + i)\right]\left[\sum_{i=1}^{5} i\cos((i+1)x_2 + i)\right],
$$

$$
\text{s.t.} \quad -10 \leq x_1, \quad x_2 \leq 10. \tag{4.7}
$$

It has 760 optimum solutions among which 18 are global optimal with functional optimum value of 186.73067 [21].

Figure 7 shows the performance of the two algorithms for the seventh test problem, with $\alpha = 5$.

Table 1: Summery of simulation results, with maximum number of iterations as a termination criterion.

	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$	$f_5(x)$	$f_6(x)$	$f_7(x)$
x^*	$(0,0)$	(π,π)	$x_1, x_2 \geq 4.5$	$(2.8863, 1.8233)$	(π,π)	$(-2.9035, -2.9035)$	multiple
$f^* = f(x^*)$	0	1	10	2.28395	$[5, 5.0794]$	358.3323	186.73067
x_s^*	$(-0.0207, 0.0140)$	$(3.2044, 3.0378)$	$(5.1149, 5.0934)$	$(2.8765, 1.9518)$	$(2.9504, 3.1789)$	$(-2.9495, -2.8714)$	$(0.3895, -0.1890)$
$f_s^* = f(x_s^*)$	-0.1239	0.9782	10	2.2308	4.8828	358.2775	183.4958
x_m^*	$(0.0.0005)$	$(3.1420, 3.1413)$	$(5.12, 5.12)$	$(2.8864, 1.8234)$	$(3.1455, 3.1368)$	$(-2.904, -2.904)$	$(-0.1989, -5.8586)$
$f_m^* = f(\mathbf{x}_m^*)$	-0.0001	1	10	2.2839	5.0279	358.3323	186.7290
CPU_s	0.1248	0.1404	0.1404	0.1560	0.2184	0.1716	0.1248
CPU_m	0.1404	0.1560	0.1404	0.1716	0.2340	0.1872	0.2028

x^* the optimal solution with f^* as the global optimal value.
x_s^* is the solution of the ordinary firefly algorithm.
f_s^* is the final functional value in the ordinary firefly algorithm.
x_m^* is the solution of the modified firefly algorithm.
f_m^* is the final functional value in the modified algorithm.
CPU_s CPU time for the standard firefly algorithm.
CPU_m CPU time for the modified firefly algorithm.

Table 2: Summary of simulation results, with an achievement level as a termination criterion.

	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$	$f_5(x)$	$f_6(x)$	$f_7(x)$
f'	-0.01	0.99	10	2.28	4.9	358.2	186
CPU_s	0.1243	0.1569	0.1100	0.1601	0.1892	0.1687	0.1304
CPU_m	0.0936	0.0321	0.0122	0.0982	0.0789	0.0654	0.0776
Itr._s	152.45	235.2	11.2	61.25	87.31	99.54	121.25
Itr._m	57.9	46.8	6.28	31.64	35.48	40.68	29.96

f': the target for termination criterion.
CPU_s: average CPU time for the standard firefly algorithm.
CPU_m: average CPU time for the modified firefly algorithm.
Itr._s: average iteration number for the standard firefly algorithm before termination.
Itr._m: average iteration number for the modified firefly algorithm before termination.

From the simulation the final functional values for the standard and modified firefly algorithms are found to be 183.4958 at $(0.3895, -0.1890)$ and 186.7290 at $(-0.1989, -5.8586)$, respectively.

The simulation results of the test problems can be summarized in Table 1.

For the simulation results it is clearly shown that the modified firefly algorithm performs better than the standard firefly algorithm. From the figures, the standard firefly algorithms performance is not always improving but sometimes gets down from the current optimal solution which is not the case in the modified firefly algorithm for all the test problems. Even though the CPU time for the modified firefly algorithm is slightly higher it gives better solution after the 500 iterations. If the termination criterion is set to achieve a certain target, say f', which is very near to the global optimal solution, the modified firefly algorithm will terminate before the standard firefly algorithm, as shown in Table 2. The two algorithms are run 100 times, and the average CPU time and average iterations number before termination are recorded, with the termination criterion are achieving at least f'. This means that the algorithm keep running until the objective function becomes f' or more, where f' is chosen to be a very close number to the global optimal solution.

From Table 2, it is shown that when the termination criterion is not the maximum number of iterations but a certain achievement level, the modified firefly algorithm achieves the target with less number of iterations which results in less CPU time as compared to the standard firefly algorithm.

5. Conclusion

Firefly algorithm is one of the newly introduced metaheuristic algorithms which mimic the behavior of fireflies at night. In this paper a discussion is given on modifying the algorithm so that the performance will be better. One of the modifications is identifying the best direction in which the brightness of the brightest firefly increases. If such a direction is found, from randomly generated directions, the brightest firefly will move in that direction. However, if such a direction is not found among the randomly generated directions, it will stay in its current position. Furthermore a modification is suggested in computing the attractiveness of each firefly at $r = 0$. According to the simulation results done on selected seven benchmark problems with different properties, the modified firefly algorithm performs better in achieving the near optimal solution. Furthermore, when the termination criterion is to achieve at least a certain functional level, modified firefly algorithm achieves the target with less CPU time and iteration numbers.

Acknowledgments

References

[1] N. Svilen and Z. Tsvetan, "Load optimization in training the olympic rowing (Skiff) champion from Beijing, 2008: case study," *Serbian Journal of Sports Sciences*, vol. 5, pp. 41–50, 2011.

[2] X. Gong, S. A. Vorobyov, and C. Tellambura, "Optimal bandwidth and power allocation for sum ergodic capacity under fading channels in cognitive radio networks," *IEEE Transactions on Signal Processing*, vol. 59, no. 4, pp. 1814–1826, 2011.

[3] S. S. Rao, *Engineering Optimization: Theory and Practice*, John Wiley & Sons, 3rd edition, 2009.

[4] D. W. K. Yeung and L. A. Petrosyan, *Subgame Consistent Economic Optimization: An Advanced Cooperative Dynamic Game Analysis*, Birkhäuser, 2011.

[5] R. J. Vanderbei, *Linear Programming: Foundations and Extensions*, International Series in Operations Research & Management Science, Springer, 3rd edition, 2008.

[6] J. F. Bonnans, J. C. Gilbert, C. Lemaréchal, and C. A. Sagastizábal, *Numerical Optimization: Theoretical and Practical Aspects*, Springer, 2003.

[7] R. Fletcher, *Practical Methods of Optimization*, John Wiley & Sons, 2000.

[8] E. G. Talbi, *Metaheuristics: From Design to Implementation*, John Wiley & Sons, 2009.

[9] T. Apostolopoulos and A. Vlachos, "Application of the firefly algorithm for solving the economic emissions load dispatch problem," *International Journal of Combinatorics*, vol. 2011, Article ID 523806, 23 pages, 2011.

[10] S. L. Tilahun and H. C. Ong, "Bus timetabling as a fuzzy multiobjective optimization problem using preference based genetic algorithm," *Promet Traffic & Transportation*, vol. 24, no. 3, 2012.

[11] J. Y. Wang and C. M. Lin, "Mass transit route network design using genetic algorithm," *Journal of the Chinese Institute of Engineers*, vol. 33, no. 2, pp. 301–315, 2010.

[12] M. Negnevitsky, *Artificial Intelligence: A Guide To Intelligent System*, Henry Ling Limited, Harlow, England, 2005.

[13] X. S. Yang, "Firefly algorithms for multimodal optimization," in *Proceedings of the 5th International Conference on Stochastic Algorithms: Foundation and Applications (SAGA '09)*, vol. 5792 of *Lecture notes in Computer Science*, pp. 169–178, 2009.

[14] X. S. Yang, *Nature-Inspired Metaheuristic Algorithm*, Luniver Press, Beckington, UK:, 2nd edition, 2010.

[15] M. Molga and C. Smutnicki, Test functions for optimization needs, 2005, http://www.zsd.ict.pwr .wroc.pl/files/docs/functions.pdf.

[16] E. J. S. Pires, J. A. T. MacHado, P. B. De Moura Oliveira, J. B. Cunha, and L. Mendes, "Particle swarm optimization with fractional-order velocity," *Nonlinear Dynamics*, vol. 61, no. 1-2, pp. 295–301, 2010.

[17] J. G. Digalalis and K. G. Margaritis, "On benchmarking functions for genetic algorithm," *International Journal of Computer Mathematics*, vol. 1, pp. 1–27, 2000.

[18] S. K. Mishra, *Repulsive Particle Swarm Method on Some Diffcult Test Problems of Global Optimization*, North-Eastern Hill University (NEHU), Meghalaya, India, 2006, http://ssrn.com/abstract=928538.

[19] X. S. Yang and Firefly Algorithm, "Stochastic test functions and design optimisation," *International Journal of Bio-Inspired Computation*, vol. 2, no. 2, p. 7884, 2010.

[20] P. Ittipong, P. Luangpaiboon, and P. Pongcharoen, "Solving nonlinear continuous mathematical models using shuffled frog leaping and memetic algorithms," in *Operations Research Network in Thailand (OR-NET '08)*, vol. 25, pp. 79–85, 2008, http://as.nida.ac.th/ornet/ORNET2008/08papers/79-85.pdf.

[21] M. Molga and C. Smutnicki, Test functions for optimization needs, 2005, http://www. bioinformaticslaboratory.nl/twikidata/pub/Education/NBICResearchSchool/Optimization/VanK-ampen/BackgroundInformation/TestFunctions-Optimization.pdf.