Vol. 5(16) Jul. 2015, PP. 2347-2354

Contents list available at IJMEC
International Journal of Mechatronics, Electrical and Computer Technology (IJMEC)
Journal Homepage: www.aeuso.org
PISSN: 2411-6173 - EISSN: 2305-0543

MFASA: A New Memetic Firefly Algorithm Based on Simulated Annealing

Nadia Nekouie[1*] and Mahdi Yaghoobi[2]

[1]*Department of Computer Engineering, Mashhad branch, Islamic Azad University, Mashhad, Iran*

[2]*Department of Computer Engineering, Mashhad branch, Islamic Azad University, Mashhad, Iran*

***Corresponding Author's E-mail:** nadia_nekouie@yahoo.com*

Abstract

Firefly algorithm is a global optimization algorithm. Unlike other algorithms such as genetic algorithm and particle swarm optimization algorithm, it benefits a competitive potential. However, it fails in finding global optimization solution for some cases. This paper introduces a new hybrid algorithm based on a simulated annealing algorithm to improve the functionality of the firefly algorithm. The simulated annealing algorithm is a algorithm with remarkable local search ability. Simultaneous application of the firefly algorithm and simulated annealing algorithm enables us to effectively benefit from the global search and local search capacity of those algorithms. To test the efficiency of the proposed method, five benchmark functions were examined. The results proved that the suggested method provides a competitive efficiency compared to other methods.

***Keywords**: simulated annealing algorithm, metaheuristic algorithm, firefly algorithm, global search, local search.*

1. Introduction

The non-linearity of many optimization problems often results in multiple local optima. To cope with this issue, global algorithms are widely used [1]. Metaheuristic techniques are well-known global optimization methods. These techniques attempt to reproduce social behavior [2] or natural phenomena [3]. Several novel metaheuristic algorithms are proposed for global search. Such algorithms can increase the computational efficiency, solve larger problems, and implement robust optimization codes [2, 3]. The main objective of all these algorithms is to perform a global optimization. To this end, each metaheuristic algorithm should possess two main characteristics of the exploration and exploitation. The exploration is the ability of searching all the space of the problem, whereas the exploration deals with the convergence ability toward the best point. The ultimate purpose of all metaheuristic algorithms is to reach equilibrium between these two. In many cases, the exploration and the exploitation are in opposition, i. e. with increase in one the other one decreases. Firefly algorithm is a new metaheuristic algorithm inspired from flashing behavior of Firefly [4]. This algorithm uses a population-based iterative procedure to perform the optimization. In this algorithm, solutions are simulated through the firefly. Each firefly emits light to attract peers.

The life of fireflies follow three basic rules:

1) All fireflies are unisex so that one firefly will be attracted at other fireflies regardless of their sex.
2) Attractiveness is proportional to their brightness. For any couple of flashing fireflies, the less bright one will move towards the brighter one. Attractiveness is proportional to the brightness which decreases with increasing distance between fireflies. If there are no brighter fireflies than a particular firefly, it will move randomly in the space.

Article History:
Received Date: Mar. 10, 2015
Accepted Date: Jun. 24, 2015
Available Online: Jul. 09, 2015

3) The brightness of a firefly is somehow related with the analytical form of the cost 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 genetic algorithms.

Recently, this algorithm is employed in many problems. Senthilnath et al. applied it to cluster data [5]. Their proposed method overcame the particle optimization algorithm and bee colonies. In another research, Horng and Liou used this algorithm to perform a threshold-based multilayer thresholding for the segmentation of images [6]. Their method compares with four methods of exhaustive search, particle optimization, quantum particle optimization and bee mating. Also, this algorithm is used to address the multi-objective optimization problems [7]. Besides, other applications of this algorithm, such as handling traveling salesman problem [8], knapsack problem [9], scheduling problem [10], and low-frequency prediction [11], are also available in the related literature. However, despite widespread use of the firefly algorithm, it lacks a high local search ability, though it has a high global search ability.

Simulated annealing algorithm is another metaheuristic algorithm which, unlike firefly algorithm, is single population. Simulated annealing algorithm is inspired by the process of physically melting of an object. It denotes a physical process in which the object is heated to its melting temperature and then it is cooled based on specific rules during which the energy of the object diminishes to a minimum value. This algorithm lacks a high global search ability, though its local search ability is desirable.

The objective of this paper is to offer a mixed method to solve optimization problems. In fact, the proposed method combines the global search property of the firefly algorithm and the local search property of the simulated annealing algorithm. The firefly algorithm is initially delineated. Then, the simulated annealing algorithm is described, followed by introducing the proposed method. Finally, the results will be analyzed by their application on five case benchmark functions. The results demonstrated the ability of the proposed method to enhance the functionality of firefly algorithm in a competitive way.

2. Firefly algorithm

Firefly algorithm was designed by Yang in 2008 by benchmarking the flashing qualities of the firefly [12]. There are two important questions in the firefly algorithm to be considered. Change in the intensity of the emitted light and the formulation of level of attractiveness. In lay terms, we may imagine that the attractiveness of a firefly depends on the intensity of the emitted light, as it is specified by the objective function .

In the most rudimentary of its form, for maximizing problems, the glowing of *i* for a firefly is proportional to the objective function. The attractiveness of a firefly or β is relative and should be observed through the eyes of other fireflies. Hence, this value changes based on the distance r_{ij} between two fireflies of *i* and *j*. Since the attractiveness of a firefly is relevant to the intensity of the light observed by close-by fireflies, the level of attractiveness of firefly i, as observed by firefly j, can be determined using the following equation:

$$\beta_{i,j} = \beta_0 e^{-\gamma r_{ij}^2} \tag{1}$$

where β0 is the attractiveness at r = 0.and γ is the light absorption coefficient. The distance between two fireflies of *i* and *j* is calculated using Euclidean distance or any other distance. The movement of firefly *i* as it's being attracted by the brighter firefly *j* is calculated as follows:

$$\Delta x_i = \beta_0 e^{-\gamma r_{ij}^2}\left(x_j^{\,t} - x_i^{\,t}\right) + \alpha\varepsilon_i, \quad (2)$$

where the second term is due to the attraction. The third term is randomization, with α being the randomization parameter, and ε_i is a vector of random numbers drawn from a Gaussian distribution or uniform distribution. The parameter γ now characterizes the variation of the attractiveness, and its value is crucially important in determining the speed of the convergence and how the FA algorithm behaves. In theory this relation is in force: $\gamma \in [0,\infty)$. When $\gamma \to 0$ the level of attractiveness remains constant, i. e. $\beta = \beta_0$. This is equivalent to that the intensity of light in the ideal condition never diminishes. Hence, a glowing firefly can be in any spot of the amplitude. Thus, the global optimization (the best firefly) can be detected by the other fireflies. On the other hand, if $\gamma \to \infty$ then $\beta(r) \to \delta(r)$ meaning that the attractiveness is almost zero in the sight of other fireflies. This is equivalent to the case where the fireflies roam randomly in a very thick foggy region randomly. No other fireflies can be seen, and each firefly roams in a completely random way, which leads to simulated annealing (SA). The Firefly algorithm can be summarized as the pseudo code shown in Figure 1.

Objective function $f(x), x = (x_1, \ldots, x_d)^T$
Initialize a population of fireflies
$x_i (i = 1,2, \ldots, n)$
Define light absorption coefficient
while (t <MaxGeneration)
for i = 1 : n all n fireflies
for j = 1 : i all n fireflies
Light intensity I_i at x_i is determined by
$f(x_i)$
if $(I_j > I_i)$
Move firefly i towards j in all d dimensions
end if
Attractiveness varies with distance r via
$\exp[-\gamma r]$
Evaluate new solutions and update light
intensity
end for j
end for i
Rank the fireflies and find the current best
end while
Postprocess results and visualization

Fig. 1: The firefly algorithm pseudo code [4].

3. Simulated annealing algorithm

One of the earliest and yet most popular metaheuristic algorithms is simulated annealing (SA), which is a trajectory-based, random search technique for global optimization. It mimics the annealing process in materials processing when a metal cools and freezes into a crystalline state with the minimum energy and larger crystal sizes so as to reduce the defects in metallic structures. The annealing process involves the careful control of temperature and its cooling rate, often called the annealing schedule. SA has been successfully applied in many areas. Since the first development of simulated annealing by Kirkpatrick et al. [13], SA has been applied in almost every area of optimization. This algorithm starts with a primary solution, finds a neighbor answer for that solution, and in the case of non-improvement in the objective function also be accepted with a probability *p*. where ΔE is the difference between the objective function of the current response and the neighbor's response, and T is the temperature. At each temperature, several repetitions are implemented and then the temperature is lowered slowly. In early steps, the temperature is kept extremely high to increase the possibility of receiving bad responses. By gradual decreases in the temperature in the final steps, the chances of receiving bad responses decrease and hence, the algorithm converges to a good response. This technique prevents being limited into a local optimized position. This ultimately leads the system to move toward a lower energy.

Indeed, in this algorithm, at each level of the solution x with a fitness function of f(x), a neighbor x' in the neighborhood of x, N(x), is selected. In each step the difference between the objective functions is as follows:

$$\Delta = f(x) - f(x') \tag{3}$$

X' can be computed using the following equation:

$$P_s = exp\left(-\frac{\Delta}{T}\right) \tag{4}$$

Then, the possibility of accepting a random number $r \in (0,1)$ is compared and x' is accepted if P>r. T is the temperature controlled by cooling schema [14]. However, the simulated annealing algorithm includes accessories such as primary temperature, a procedure for changing the temperature, and a cooling schema until it stops. The basic structure of SA algorithm is presented in Figure 2.

- Select an initial solution x_0
- REPEAT
- Generate solution x' in the neighborhood of x_0
- Calculate Δ, and calculate the probability by Eqn. (4)
- Decide the new solution by P_s
- Memorize the best solution found so far
- Reduce the temperature
- UNTILE requirements are met

Fig. 2: The basic structure of simulated annealing algorithm [14].

4. Firefly algorithm based on simulated annealing

This paper presents a new technique to solve the optimization problems. The developed technique is a combination of firefly algorithm and simulated annealing algorithm. Firefly algorithm is a population-oriented one. Such algorithms show a considerable capacity in global search, though they prove weak in local searches. As stated in previous section, the simulated annealing algorithm includes a process for neighbor search. This process results in a high local search ability for the proposed method; though it is still weak in global search. Combination of these two algorithms allows the search space to be searched globally using the firefly algorithm and locally using the simulated annealing algorithm, with the simultaneous utilization of the advantages of both algorithms. In the presented method, n primary population of fireflies is first generated and the level of brightness for each firefly is computed. In the second step, each firefly moves toward the firefly with the greatest attractiveness. In this step, the simulated annealing algorithm for local search is used around the generated solutions. The overall structure of the proposed method is as below:

1. Primary initialization of the firefly algorithm parameters such as number of primary population N_{Pop}, the number of maximum repetitions, and the attraction coefficient.
2. Primary initialization of the simulated annealing algorithm parameters such as the number of repetitions and the primary temperature (T).
3. Generation of N_{Pop} fireflies (primary solution).
4. For each pair of fireflies (solutions) take the following steps:
4.1. If the attractiveness of firefly *i* is greater than that of firefly *j* (or if the fitness function *i* is better than the fitness function *j*), the firefly *i* moves toward the firefly *j* based on the following equation:

$$\Delta x_i = \beta_0 e^{-\gamma r_{ij}^2}\left(x_j{}^t - x_i{}^t\right) + \alpha\varepsilon_i, \qquad (5)$$

4.2. Change the level of attractiveness in accordance with the distance.
4.3. For each generated solution x:
4.3.1. Find the neighbors of x.
4.3.2. If its energy is decreased $\Delta E < 0$, accept the solution, otherwise accept it if $\exp(-\Delta E / T)$.
4.3.3. If the balance is not reached, the temperature needs to be decreased and shift to be made to step 4.3.1.
4.4. If the condition of the algorithm termination is not met, move to step 4.

5. Results

In this section, the results of implementation of proposed method on five benchmark functions are examined. These functions are addressed in a 10-demensional state. A summary of the employed functions' characteristics are given in table 1.

Table 1: Characteristics of the employed benchmark function

Name	Benchmark Functions	Lower bound	Upper bound
Ackley	$f(X) = -a\ exp\left(-b\sqrt{\frac{1}{d}\sum_{i=1}^{d} \mathrm{x_i}^2}\right) - exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos(cx_i)\right) + a + \exp(1)$	-32.768	+32.768
Grienwank	$f(X) = \sum_{i=1}^{d}\frac{x_i{}^2}{4000} - \prod_{i=1}^{d} cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	-600	+600
Rastrigin	$f(X) = 10d + \sum_{i=1}^{d}[x_i{}^2 - 10cos(2\pi x_i)]$	-5.12	+5.12
Schwefel	$f(X) = 418.9829d - \sum_{i=1}^{d} x_i sin(\sqrt{\lvert x_i\rvert})$	-10	+10
Sphere	$f(X) = \sum_{i=1}^{d} x_i{}^2$	5.12	-5.12

According to the random nature of metaheuristic algorithm, the implementation of the algorithm results in different answers. Thus, the proposed algorithm was implemented 30 times and the statistic results of these implementations such as the mean and the standard deviation are reported. Increasing the value of the mean and decreasing the standard deviation rises the efficiency of the algorithm. The results of proposed method are demonstrated in table 2. Also, the statistical results related to the functionality of the firefly algorithm and simulated annealing algorithm are given in tables 3 and 4, respectively. The convergence diagram for the firefly method and the proposed method is presented in Figure 3,4. As can be seen, the proposed method demonstrates a very high convergence speed. The suggested method, in a very effective way, can improve the result of the firefly algorithm on every benchmark function .

Table 2: Statistical results of firefly algorithm

Test functions	mean	Median	maximum	minimum	Standard deviation
Ackley	2.81E-01	2.50E-01	1.30E-01	7.04E-01	1.76E-01
Grienwank	3.12E-01	1.27E-01	9.34E-01	6.85E-03	3.50E-01
Rastrigin	2.17E+01	1.45E+01	4.85E+00	5.10E+01	1.59E+01
Schwefel	2.89E+02	2.60E+02	4.33E+02	1.47E+02	1.27E+01
Sphere	7.79E-03	1.10E-02	2.76E-02	8.35E-03	2.04E-03

Table 3: Statistical results of simulated annealing algorithm

Test function	Mean	median	maximum	minimum	Standard deviation
Ackley	1.06E-01	2.18E-04	9.96E-01	1.33E-05	3.13E-01
Grienwank	2.13E-01	3.99E-04	9.95E-01	1.32E-06	3.62E-01
Rastrigin	4.18E-02	5.85E-04	3.13E-01	5.92E-05	9.76E-02
Schwefel	8.37E-03	3.05E-01	9.95E-01	4.77E-01	4.29E-05
Sphere	4.94E-04	2.39E-01	9.95E-01	4.18E-01	8.12E-06

Table 4: Statistical results of proposed method

Test Function	GA	PSO	SA	FA	Proposed Method
Ackley	5.26E-01	4.32E-3	2.81E-01	2.81E-01	9.76E-10
Grienwank	6.61E-01	3.25E-1	3.12E-01	3.12E-01	2.18E-10
Rastrigin	6.15E+02	6.32E-01	2.17E+01	2.17E+01	7.47E-11
Schwefel	9.69E+01	3.56E+2	2.89E+02	2.89E+02	1.30E-04
Sphere	9.65E-02	8.1E-03	7.79E-03	7.79E-03	8.91E-10

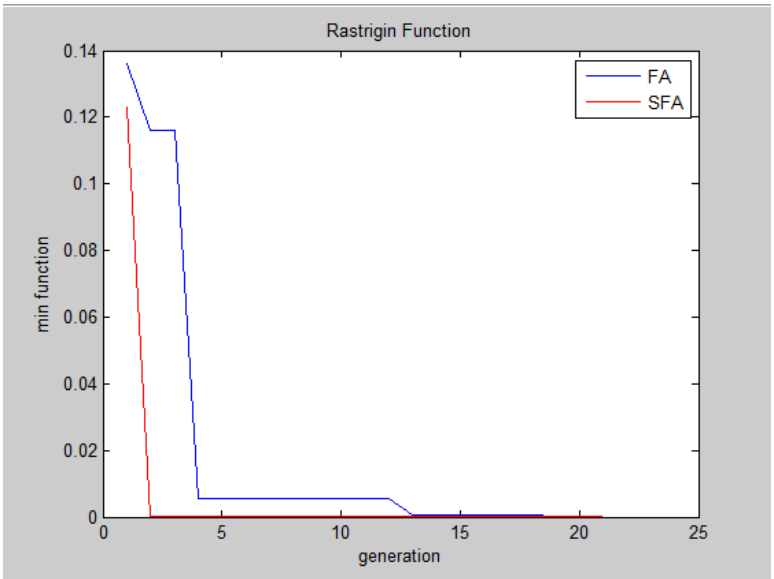

Fig. 3: The comparison between firefly algorithm convergence speed and proposed method on rastrigin function

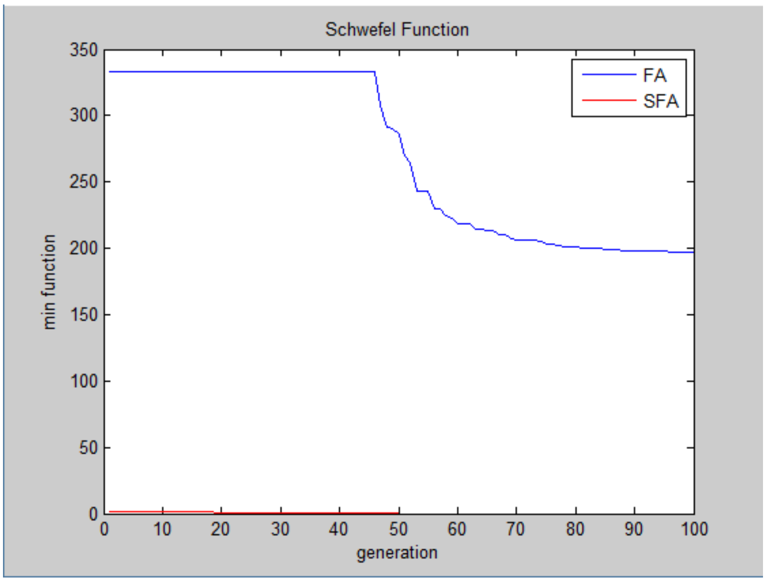

Fig. 4: The comparison between firefly algorithm convergence speed and proposed method on schwefel function

Conclusions

In this research, a new method for improvement of the functionality of the firefly algorithm is presented. The firefly algorithm demonstrates a great potential for the global search, though it proves modest in the local search. Hence, the local search ability of simulated annealing algorithm was implemented. The proposed algorithm was tested on five benchmark functions. The results proved that this algorithm, in comparison with firefly algorithm, possesses an extremely high ability in approximation of the global optimized position. As an extension to this research, the proposed method can be developed to use multiple functions.

References

[1] Y. XS, "Nature-inspired metaheuristic algorithms," *Luniver Press,* 2008.
[2] A. A. Gandomi AH, "Krill herd: a new bio-inspired optimization algorithm. Commun Nonlinear Sci," *Numer Simulat, in press. http://dx.doi.org/10.1016/j.cnsns.2012.05.010,* 2012.
[3] T. S. Kaveh A, "A novel heuristic optimization method: charged system search," *Acta Mech,* vol. 213(3–4), pp. 267–89, 2010.
[4] X.-S. YANG, "Firefly algorithm, stochastic test functions and design optimisation," *International Journal of Bio-Inspired Computation,* vol. 2, pp. 78-84, 2010.
[5] J. SENTHILNATH, OMKAR, S. & MANI, V, "Clustering using firefly algorithm: Performance study," *Swarm and Evolutionary Computation,* vol. 1, pp. 164-171, 2011.
[6] M.-H. L. HORNG, R.-J, "Multilevel minimum cross entropy threshold selection based on the firefly algorithm," *Expert Systems with Applications,* vol. 38, pp. 14805-14811, 2011.
[7] X.-S. YANG, "Multiobjective firefly algorithm for continuous optimization," *Engineering with Computers,* vol. 29, pp. 175-184, 2013.
[8] L. ZHOU, DING, L. & QIANG, X, "A Multi-population Discrete Firefly Algorithm to Solve TSP," *Bio-Inspired Computing-Theories and Applications. Springer,* 2014.
[9] A. O. BAYKASOĞLU, F. B, "An improved firefly algorithm for solving dynamic multidimensional knapsack problems," *Expert Systems with Applications,* vol. 41, pp. 3712-3725, 2014.
[10] S. KARTHIKEYAN, ASOKAN, P. & NICKOLAS, S, "A hybrid discrete firefly algorithm for multi-objective flexible job shop scheduling problem with limited resource constraints," *The International Journal of Advanced Manufacturing Technology,* vol. 72, pp. 1567-1579, 2014.
[11] A. KAVOUSI-FARD, SAMET, H. & MARZBANI, F, "A new hybrid Modified Firefly Algorithm and Support Vector Regression model for accurate Short Term Load Forecasting," *Expert Systems with Applications,* vol. 41, pp. 6047-6056, 2014.
[12] X.-S. YANG, "Nature-inspired metaheuristic algorithms," *Luniver press,* 2010.
[13] G. C. Kirkpatrick S, Vecchi MP, "Optimization by simulated annealing," *Science* vol. 220(4598), pp. 671–80, 1983.
[14] S.-M. CHEN, SAROSH, A. & DONG, Y.-F, "Simulated annealing based artificial bee colony algorithm for global numerical optimization," *Applied mathematics and computation,* vol. 219, pp. 3575-3589, 2012.