International Journal of Hybrid Information Technology
Vol. 9, No.7 (2016), pp. 397-410
http://dx.doi.org/10.14257/ijhit.2016.9.7.36

An Improved Firefly Algorithm Based on Nonlinear Time-varying Step-size

Shuhao Yu[1,2], Shenglong Zhu[3] and Xiancun Zhou[1]

[1]*School of Information Engineering, West Anhui University, Lu'an, 237012, China*
[2]*School of Management, Hefei University of Technology, Hefei, 230009, China*
[3]*State Grid Anhui Electric Power Research Institute, 230601 Hefei, China*
yush@wxc.edu.cn

Abstract

Firefly algorithm (FA) is a novel population-based stochastic optimization algorithm and has been shown to yield good performance for solving varieties of optimization problems. Meanwhile, it sustains premature convergence because it is easily to fall into the local optima which may generate a low accuracy of solution or even fail. To overcome this defect, a nonlinear time-varying step strategy for firefly algorithm (NTSFA) is presented. It uses a nonlinear decreasing and time-varying step-size for fireflies to better balance the algorithm's ability of exploration and exploitation. Numerical simulation on 20 test benchmark functions display that the proposed algorithm can increase the accuracy of the original FA. Finally, we apply NTSFA to integrate into k-means clustering for mouse dataset. The results show that NTSFA is an effective optimization algorithm.

Keywords: *firefly algorithm; local optimum; nonlinear time-varying; step-size*

1. Introduction

The fireflies flash in order to attract a mating partner or to resist predators' aggression. With the inspiration motived by the research results on the modeling and the simulations of the behavior of the fireflies, Yang proposed the firefly algorithm (FA)[1]. Firefly algorithm is built on a physical feature of light intensity that reduces with the increase of the distance. As the distance between the firefly and the light source augments, the light absorption leads to the light becomes increasingly weak. [2].

Similar to other metaheuristic search algorithms, firefly algorithm tends to suffer from the premature convergence issue, which is primarily due to the fast convergence feature and diversity loss of the fireflies' population during the search process. In recent years, many researchers focused on developing new firefly algorithm variants that avoid the premature convergence problem [3-12]. These variants of the FA can be stated in four areas: parameter tuning and parameter control, algorithms use of learning strategies, hybridization with other search techniques and discrete FA variants. A preliminary summary of the above categories of the FA are listed below.

1) Parameter setting and control: The purpose of parameter setting and parameter control is to find more perfect parameter so as to help an algorithm perform well for range of issues. In basic firefly algorithm, the method of parameter setting is fixed which cannot be adjusted. Therefore, the proper selection of control parameters such as step value and light absorption coefficient significantly influences the convergence. There are many variants of firefly algorithm with variable step such as fuzzy settings[4, 13], chaotic tuning[3, 14], and self-adaptive step settings[8] during iterations.

2) Algorithms based on learning strategies: In [15], they proposed to use quaternion for the statement of each firefly in FA in order to improve the performance of the original FA and keep away from any stagnation. Trunfio et al. [16] used a cooperative coevolutionary

ISSN: 1738-9968 IJHIT
Copyright © 2016 SERSC

method for strengthening firefly algorithm with the purpose of bettering it much more competent under the condition of search spaces with multiple dimensions. In [17], a robust firefly algorithm to solve global numerical optimization problems was proposed, in which the improvement included the addition of information-exchange strategy between the top fireflies, or the optimal solutions during the process of the light intensity updating. In [18], the developer of the firefly algorithm, Yang formulated a new variant of algorithm by integrating Lévy flights with the search strategy via the FA.

3) Hybridization with other search techniques: In order to improve the performance of the basic FA, research focus has been aimed to alleviate the weaknesses of the original FA by integrating it with useful attributes of other evolutionary algorithms, such as genetic algorithm[19], harmony search[20], sequential quadratic programming[21], simulated annealing algorithm[22], particle swarm optimization[23-24], and *etc*.

4) Discrete firefly algorithm versions: The basic FA is designed for continuous optimization problems. However, many practical problems are formulated as discrete optimization problems. So the researchers proposed the discrete firefly algorithm to solve discrete optimization problems. Sayadi et al.[25] presented the discrete FA to solve the manufacturing cell formation problem. In [26], they proposed a hybrid discrete firefly algorithm for multi-objective flexible job shop scheduling problem with limited resource constraints, and in [27], they developed a discrete FA to actualize in the loading pattern optimization of nuclear reactor core. Farhoodnea et al.[28] presented a novel solution for the optimal placement and sizing of active power conditioners by utilizing the dynamic discrete FA.

In this paper, we propose an improved FA with a nonlinear time-varying step strategy. The proposed algorithm uses nonlinear decreasing step size with time-varying for all fireflies, which can balance the algorithm's ability of exploration and exploitation. Experiments on 20 benchmark test functions show the proposed algorithm has demonstrated a surprising effectiveness and accuracy in obtaining the optimal solution.

The rest of the paper is organized as follows. Section 2 describes briefly the basic firefly algorithm. Section 3 provides the detailed description of the nonlinear time-varying step strategy proposed in this paper. Section 4 presents the experimental settings and simulation results. Finally, the conclusion is drawn in Section 5.

2. A Brief Overview of the Firefly Algorithm

This section provides a brief description of the basic firefly algorithm.

2.1. The Biological Foundations of Firefly Algorithm

The firefly algorithm has two important factors: the brightness and attractiveness. Brightness shows the advantages or disadvantages of a firefly's position and determines its moving direction, while attractiveness determines the moving distance. The firefly algorithm achieves the objective optimization through the constantly updated brightness and attractiveness. As Yang [1] demonstrated, the firefly algorithm is based upon three idealized formulas as following:

1) All fireflies are unsexing so that one individual is attracted to others without regard to their sexes. The brightness of a firefly is computed by the scenario of the objective function, the better position, the higher brightness;

2) Attractiveness is proportional to a firefly's brightness. Therefore, for any two fireflies, the brighter one will attract the less bright one and the attractiveness reduces with their distance increases;

3) A particular firefly will move randomly if there is no brighter one than it.

 Copyright © 2016 SERSC

2.2. The Description and Analysis of Firefly Algorithm

In general, We presume that the attractiveness of a firefly depends on its brightness which is equivalent to the encoded objective function[29]. As the distance increases from the source, the variations of brightness and attractiveness should be monotonically decreasing functions. The distance between any two individuals can be calculated based on Cartesian distance as follows.

$$r_{ij} = \left\| x_i - x_j \right\| = \sqrt{\sum_{k=1}^{d} (x_{ik} - x_{jk})^2} \tag{1}$$

where x_{ik} is the k-th element of the i-th firefly and d is the dimension of the object function to be optimized. In practical applications, the attractiveness can be computed by the following form.

$$I(r) = I_0 e^{-\gamma r^2} \tag{2}$$

here, γ is the light absorption factor which can be treated as a constant. Since a firefly's attractiveness is proportional to the brightness, the attractiveness β of a firefly defines as follows.

$$\beta(r) = \beta_0 e^{-\gamma^2} \tag{3}$$

where β_0 is the attraction at $r = 0$. Thus, the movement of a firefly i is attracted to the other more attractive one j is computed by

$$x_{i+1} = x_i + \beta(x_j - x_i) + \alpha(rand - 0.5) \tag{4}$$

where x_i and x_j are the spatial coordinates of fireflies i and j. The second term is associated with the attractiveness. The third term is randomization with the factor α which is step size factor generated from interval [0, 1]. The *rand* is a random number generator uniformly distributed in [0, 1]. The main steps of the firefly algorithm are summarized in Figure 1.

```
begin
   Objective function f(X), X=(x_1, ...,x_d)^T
   Generate initial population of fireflies X_i(i=1, ...,n)
   Brightness I_i at X_i is determined by f(X_i)
   Define light absorption coefficient γ
   while (t<MaxGeneration)
   for i=1:n all n fireflies
      for j=1:n all n fireflies
         if (I_j>I_i)
            Move firefly i towards j in d-dimension
         end if
         Attractiveness varies with distance via exp[- γr^2]
         Evaluate new solutions and update brightness
      end for j
    end for i
   Rank the fireflies and find the current best
   end while
   Post process results and visualization
end
```

Figure 1. Pseudo Code of the Firefly Algorithm

2.3. Algorithm Complexity

Figure 1 shows that the firefly algorithm has one surrounding loop and two inner loops for iteration t when going through the population n. So the complexity of the firefly algorithm is $O(n^2 t)$ at the extreme case. As [1, 29] suggested, the population (n=40) is small and the iteration (t=5000) is large. Therefore, the computation cost is relatively low-cost because the algorithm complexity is linear on the basis of the iteration. If the population is relatively large, we can utilize one inner loop by sorting the attractiveness or brightness of all fireflies using sorting algorithms. On this occasion, the computational complexity of FA will be $O(n \cdot t \cdot \log(n))$.

The main computational cost is to evaluate the objective functions, peculiarly for exterior black-box type objectives. This latter case is also true for all metaheuristic algorithms. All in all, for all optimization problems, the most computationally vast part is objective evaluations[30].

3. The Nonlinear Time-Varying Step Strategy for Firefly Algorithm

3.1. Motivation

In the basic FA, the third term of equation (4) is randomization with step α. In general, the method of setting step size is fixed ($\alpha = 0.2$), which cannot variable to all fireflies. With a large step size, it is helpful for fireflies to explore new search space, but it is not useful to the convergence of global optimum. If the step size has a small value, the result is contrary. Therefore, the step α has a great effect on the exploration and convergence of the algorithm.

In general, an ideal step size setting should be as following: at early iteration, the step should adopt large value which can improve the algorithm’s ability of exploration, and at later period the step should decrease which can enhance the algorithm’s ability of exploitation.

To take into account the above issues, a nonlinear time-varying step strategy for firefly algorithm (NTSFA) is proposed.

3.2. Nonlinear Time-Varying Step Settings

As mentioned above, in firefly algorithm step α plays a key role in regulating the equilibrium of exploration and exploitation. In NTSFA, to balance the algorithm’s ability between exploration and exploitation, the best firefly executes self-regulation of its step to accelerate its seeking for the global optimum. The rest of the fireflies employ the standard procedure and use the step decreasing linearly. The personalized step strategy is defined as follows.

$$\alpha(t) = 1 - \left(\frac{t}{MaxGeneration}\right)^{\frac{1}{\pi^2}} \tag{5}$$

where $\alpha(t)$ is the step at t iteration, *MaxGeneration* is the maximum of iterations. The decreasing nature of variable and dynamic strategy is graphically illustrated in Figure 2.

 Copyright © 2016 SERSC

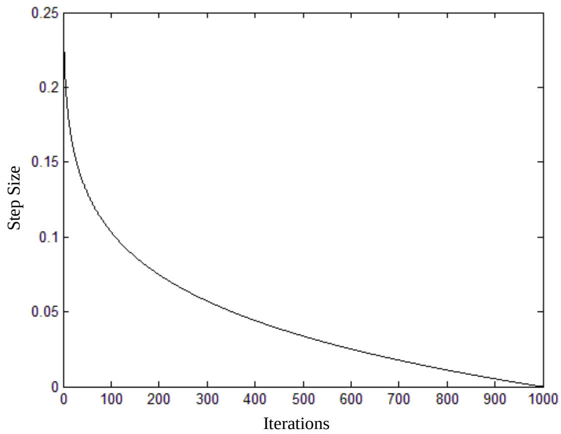


Figure 2. Variation of Step Over 1000 Iterations

From this figure we can see clearly that the step is large at early stage, and then decreases with the iteration increases. This strategy can help the proposed algorithm to improve its ability of exploration at early iteration, and enhance its ability of exploitation at later period.

3.3. Program Flow of the Proposed Algorithm

The implementation procedure of our proposed nonlinear time-varying step-size firefly algorithm can be described as follows:

Step 1: Create the initial population of fireflies, $\{x_1, x_2, ..., x_n\}$;

Step 2: Compute intensity for each firefly member, $\{I_1, I_2, ..., I_n\}$;

Step 3: Calculate the step size by equation (5);

Step 4: Move every firefly to other brighter fireflies, the position of fireflies are computed by equation (4) ;

Step 5: Update the solution set;

Step 6: Terminate if a termination criterion is satisfied, or else go to Step 2.

4. Experimental Results and Discussion

This section gives the experimental settings that were utilized to evaluate the proposed NTSFA and presents the experimental results.

4.1. Benchmark Functions and Experiment Settings

We choose 20 benchmark functions to test the performance of NTSFA, and compare the results obtained by NTSFA and the basic firefly algorithm. All test functions are minimization problems and listed in Table I.

All the programs were executed in Matlab 2010b under Windows 7 with 2 GB RAM. We adopt 100 independent runs so as to eliminate stochastic unconformity in each case study. As it suggested by literature [1], the number of fireflies was 30, the dimension for f_{11}-f_{20} was 20 and the maximum iteration number was 1000. Other parameters are set as follows: the light absorption coefficient $\gamma = 1.0$, the attractiveness $\beta_0 = 1.0$ and the step α is calculated by equation (5).

Table I. Benchmark Functions

Functions	Formulations	Limits
f1	$f(x) = x_1^2 + x_2^2 - \cos(18x_1) - \cos(18x_2)$	[-1, 1]
f2	$f(x) = (4 - 2.1x_1^2 + \frac{x_1^4}{3})x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2$	[-3, 3]
f3	$f(x) = (\lvert x_1 \rvert - 5)^2 + (\lvert x_2 \rvert + 5)^2$	[-10, 10]
f4	$f(x) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi})\cos x_1 + 10$	[-5, 5]
f5	$f(x) = 100(x_2 - x_1)^2 + (6.4(x_2 - 0.5)^2 - x_1 - 0.6)^2$	[-5, 5]
f6	$f(x) = 100(x_2 - x_1^2) + (x_1 - 1)^2$	[-2.048, 2.048]
f7	$f(x) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$	[-10, 10]
f8	$f(x) = (1 + (x_1 + x_2 + 1))^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)^2$ $(30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2))$	[-2, 2]
f9	$f(x) = 0.5 + \frac{(\sin\sqrt{x_1^2 + x_2^2})^2 - 0.5}{1 + 0.001(x_1^2 + x_2^2)^2}$	[-10, 10]
f10	$f(x) = (x_2 + x_1^2 - 11)^2 + (x_1 + x_2^2 - 7)^2 + x_1$	[-5, 5]
f11	$f(x) = \sum_{i=1}^{D} x_i^2$	[-100, 100]
f12	$f(x) = \sum_{i=1}^{D} (x_i^2 - 10\cos(2\pi x_i) + 10)$	[-5.12, 5.12]
f13	$f(x) = \sum_{i=1}^{D} (-x_i \sin(\sqrt{\lvert x_i \rvert}))$	[-500, 500]
f14	$f(x) = 418.9829D - \sum_{i=1}^{D} (-x_i \sin(\sqrt{\lvert x_i \rvert}))$	[-500, 500]
f15	$f(x) = \sum_{i=1}^{D} \lvert x_i \rvert + \prod_{i=1}^{D} \lvert x_i \rvert$	[-10, 10]
f16	$f(x) = \sum_{i=1}^{D} x_i^2 + (\frac{1}{2}\sum_{i=1}^{D} ix_i)^2 + (\frac{1}{2}\sum_{i=1}^{D} ix_i)^4$	[-5, 10]
f17	$f(x) = \sum_{i=1}^{D} \lvert x_i + 0.5 \rvert^2$	[-100, 100]
f18	$f(x) = -20\exp(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}) - \exp(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)) + 20 + e$	[-32, 32]
f19	$f(x) = \sum_{i=1}^{D-1} (100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2)$	[-2.048, 2.048]
f20	$f(x) = \sum_{i=1}^{D} \frac{x_i^2}{4000} - \prod_{i=1}^{D} \cos(\frac{x_i}{\sqrt{i}}) + 1$	[-600, 600]

4.2. Result Comparisons on Solution Accuracy

The best value, the worst value, the mean fitness and the standard deviation obtained by two algorithms are provided in Table II. The best results obtained for each function are

Copyright © 2016 SERSC

highlighted in bold. The results show that NTSFA has a stronger ability to obtain the global optimum on all of the tested functions and outperforms the basic firefly algorithm signally only except functions f2 and f4.

To observe the convergence features of the NTSFA, both the basic firefly algorithm and NTSFA algorithms are tested on all tested functions. The fitness versus iteration for both the algorithms is shown in Figure 3 to Figure 22.

From the figures, we can see that NTSFA converges to the global optimum closely and with more faster rate of convergence. As all the functions are minimization problems, the lower point it has when the algorithm stops, the better solution it gains in the end. In order to see clearly, we adopt the iteration *Maxgeneration*=300. The figures show that NTSFA is unlikely to be trapped into local optima but can achieve the global optimum gradually during the process, particularly on the functions f5 to f7, f11 to f20.

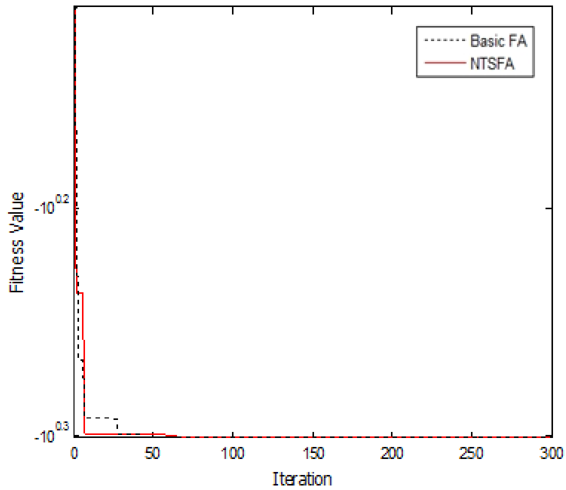


Figure 3. Convergence Graphs of F1

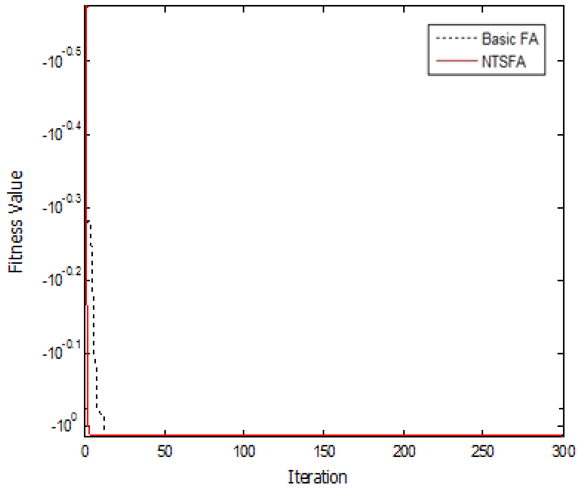


Figure 4. Convergence Graphs of F2

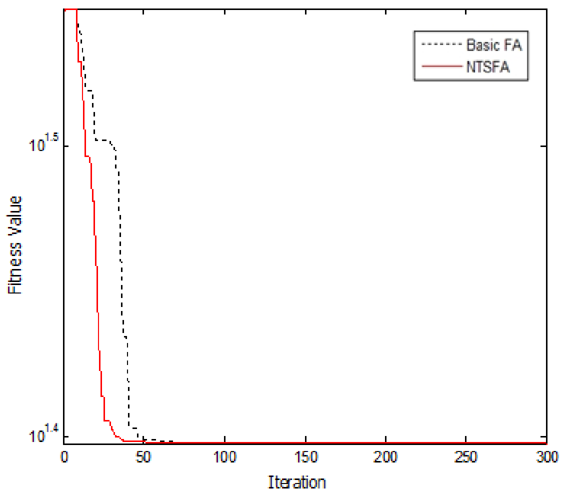


Figure 5. Convergence Graphs of F3

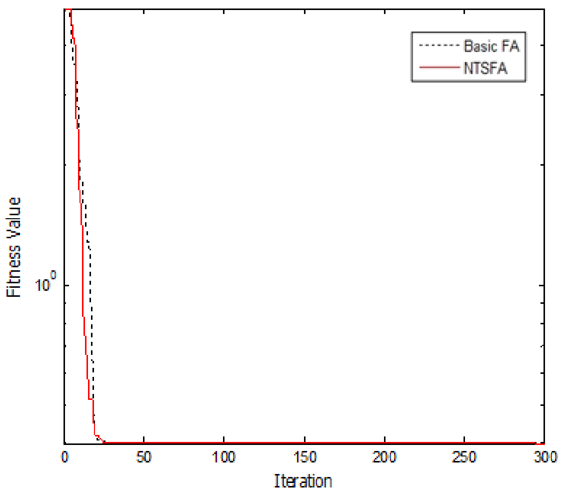
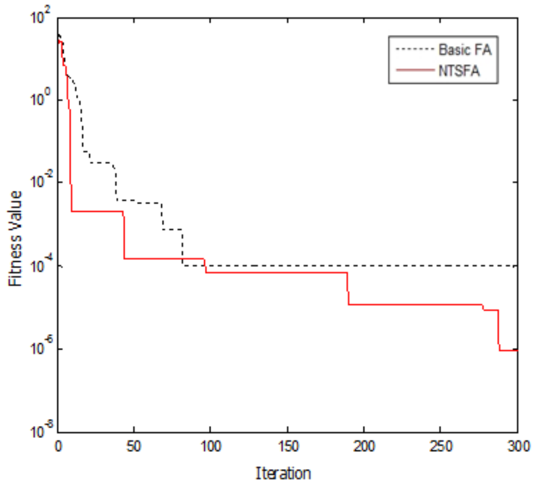


Figure 6. Convergence Graphs of F4



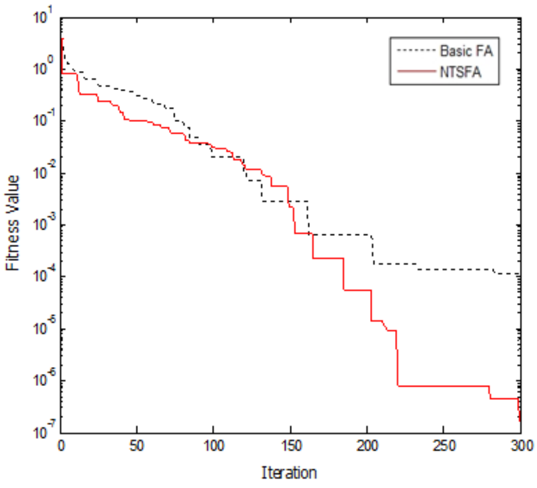
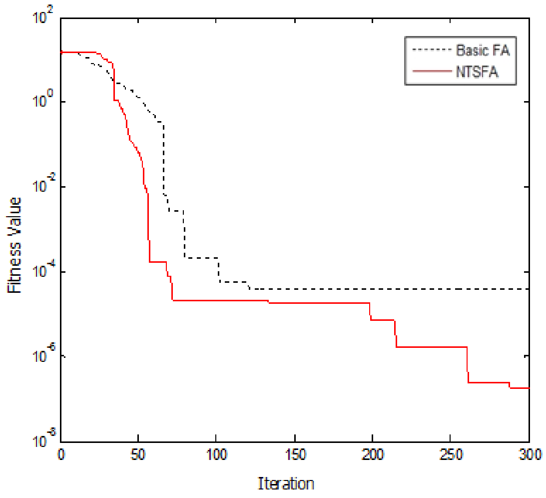


Figure 7. Convergence Graphs of F5 **Figure 8. Convergence Graphs of F6**



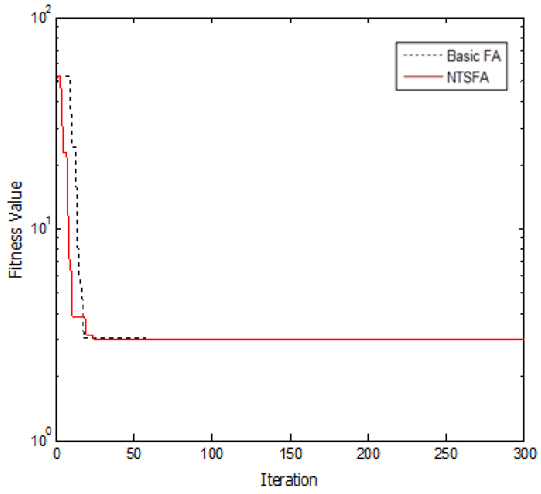
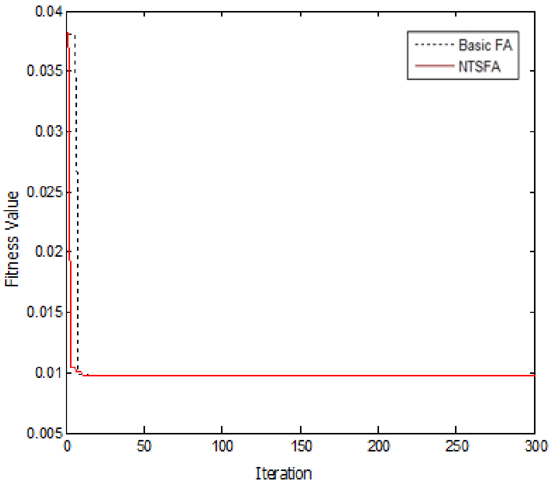


Figure 9. Convergence Graphs of F7 **Figure 10. Convergence Graphs of F8**



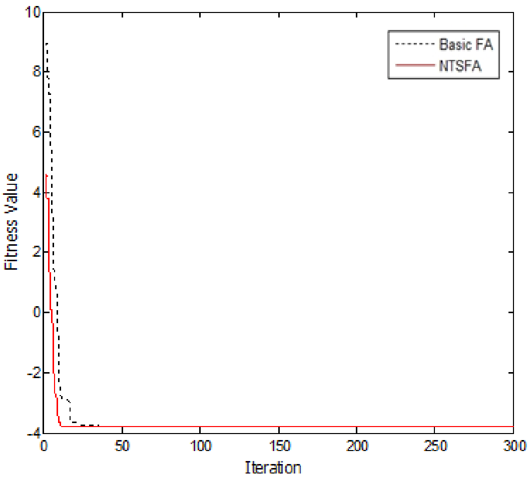


Figure 11. Convergence Graphs of F9 **Figure 12. Convergence Graphs of F10**

 Copyright © 2016 SERSC

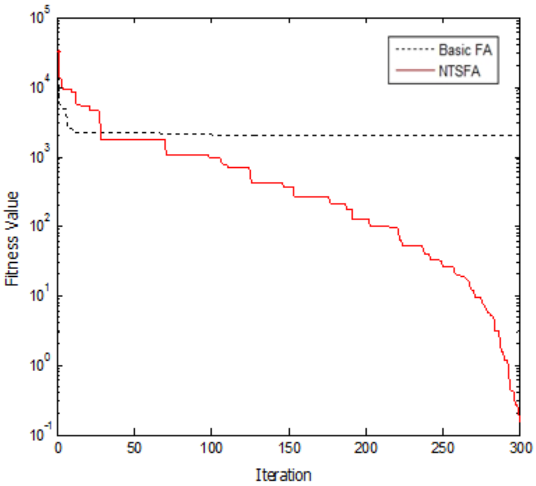


Figure 13. Convergence Graphs of F1

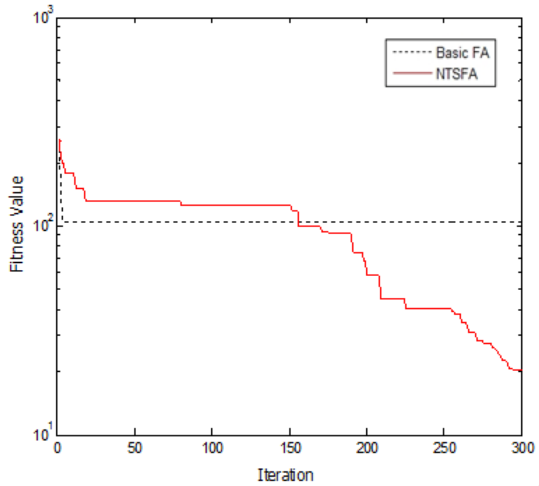


Figure 14. Convergence Graphs of F12

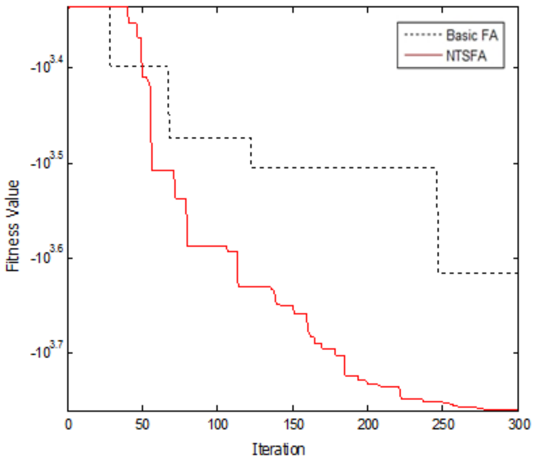


Figure 15. Convergence Graphs of F13

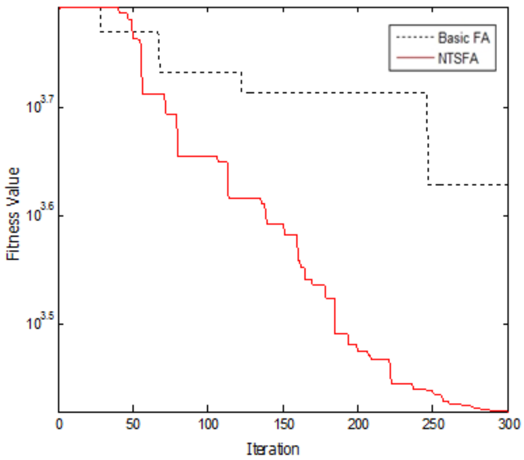


Figure 16. Convergence Graphs of F14

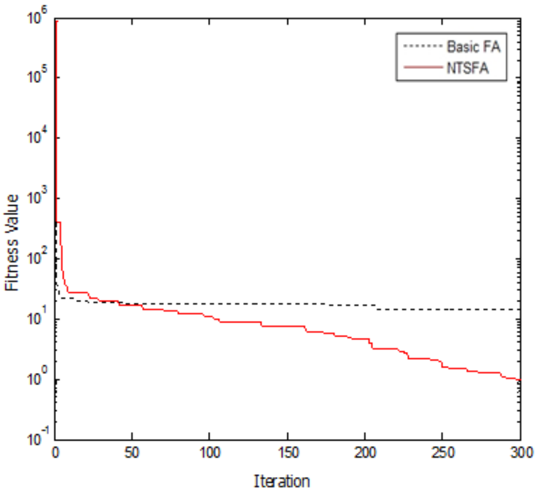


Figure 17. Convergence Graphs of F15

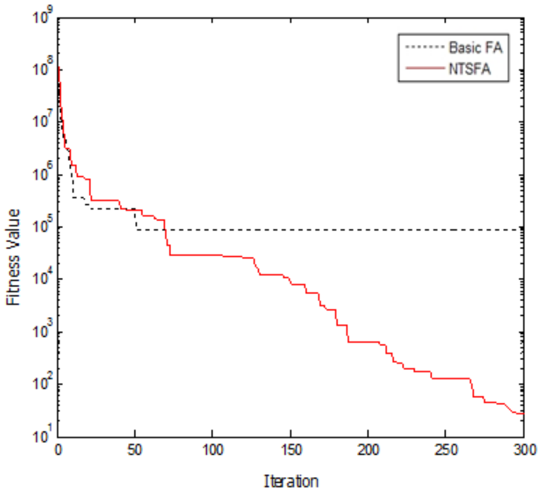
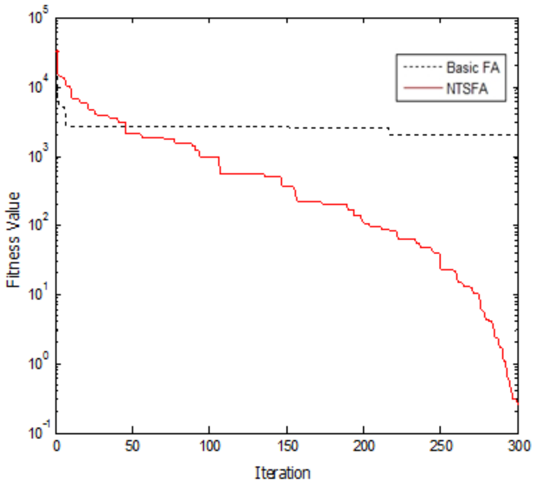


Figure 18. Convergence Graphs of F16

Copyright © 2016 SERSC



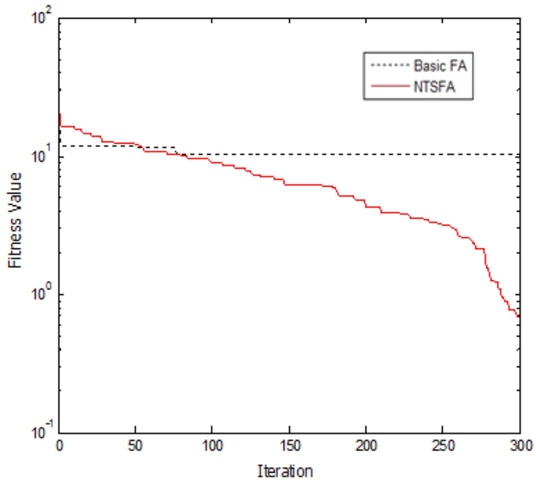
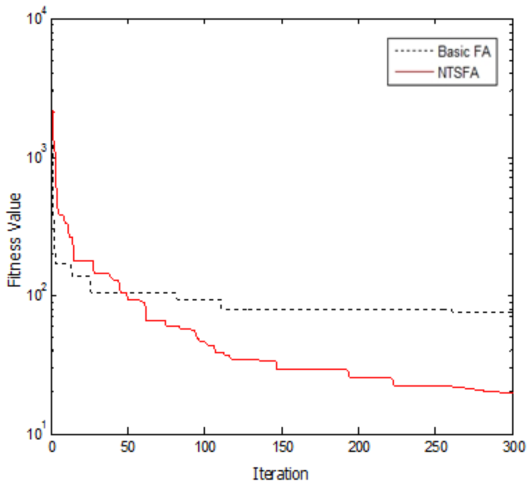


Figure 19. Convergence Graphs of F17 **Figure 20. Convergence Graphs of F18**



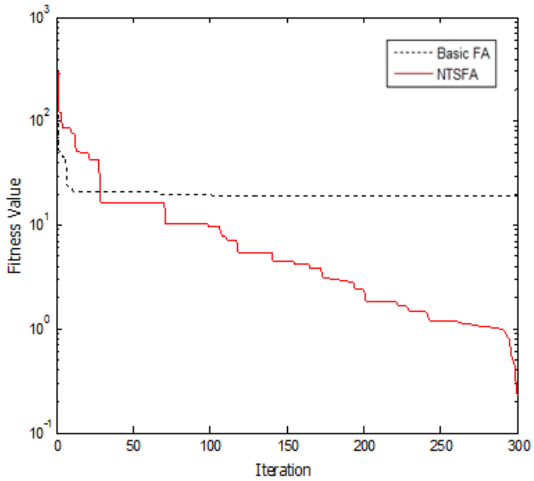


Figure 21. Convergence Graphs of F19 **Figure 22. Convergence Graphs of F20**

4.3. Non-Parametric Test for Analyzing the Algorithms

In order to further evaluate the performance of the proposed algorithm, a non-parametric statistical tool, Wilcoxon's test is also conducted. Table III shows the results of using Wilcoxon's test for functions f1 to f20. The level of significance considered is α=0.05.

From Table III, it's easy to see that Wilcoxon's test obtain the p-values smaller than the level of significance α=0.05. According to the suggestion given in literature[31], the smaller the p-value, the stronger the evidence against the null hypothesis, and the results in Table III are therefore showing a significant difference in the performance of the proposed NTSFA and the basic firefly algorithm for all of the functions. For the functions with large p-values the test is showing no statistical significant difference in the performance of the algorithms. Overall it is apparent from the results of the Wilcoxon's tests that the proposed NTSFA has significantly better performance than the basic firefly algorithm.

 Copyright © 2016 SERSC

Table III. Results of Benchmark Functions in 100 Runs

Function	Best Value		Worst Value		Mean Value		Standard Deviation	
	Basic FA	NTSFA	Basic FA	NTSFA	Basic FA	NTSFA	Basic FA	NTSFA
f1	-1.99999E+00	**-2.00000E+00**	-1.99923E+00	**-1.99990E+00**	-1.99985E+00	**-1.99999E+00**	1.47032E-04	**9.92751E-06**
f2	-1.03163E+00	-1.03163E+00	-2.1546E-01	-1.02155E+00	-1.01888E+00	**-1.03153E+00**	9.31831E-02	**1.00821E-03**
f3	2.500002E+01	**2.500000E+01**	2.500189E+01	**2.500014E+01**	2.500064E+01	**2.500003E+01**	4.211480E-04	**2.737638E-05**
f4	3.978874E-01	3.978874E-01	3.979011E-01	**3.978874E-01**	3.978908E-01	**3.978874E-01**	3.103210E-06	**3.267788E-09**
f5	7.949154E-07	**7.208494E-10**	3.176062E-04	**4.862455E-07**	5.073456E-05	**7.467843E-08**	5.748401E-05	**8.596316E-08**
f6	1.720344E-07	**2.246149E-10**	4.772365E-05	**3.627762E-07**	1.266809E-05	**4.237888E-08**	1.101537E-05	**5.706483E-08**
f7	1.451270E-08	**3.360701E-11**	1.215915E-05	**4.435371E-08**	3.145345E-06	**4.809479E-09**	2.943494E-06	**6.273583E-09**
f8	3.000004E+00	**3.000000E+00**	3.000090E+01	**3.000003E+00**	3.540377E+00	**3.000001E+00**	3.799086E+00	**7.054787E-07**
f9	9.109657E-07	**4.540417E-10**	1.978356E-02	**9.903703E-03**	7.920619E-03	8.709271E-03	4.063768E-03	**2.702575E-03**
f10	-3.78396E+00	**-3.78396E+00**	-2.81272E+00	**-2.81278E+00**	-3.74967E+00	**-3.76724E+00**	1.726934E-01	**1.192112E-01**
f11	1.031110E+03	**5.927894E-02**	2.028277E+03	**1.691856E-01**	1.479882E+03	**1.137935E-01**	2.083085E+02	**2.716365E-02**
f12	8.979507E+01	**8.487045E+00**	1.229886E+02	**3.954261E+01**	1.060949E+02	**2.157631E+01**	6.703514E+00	**6.696686E+00**
f13	-4.22775E+03	**-6.24765E+03**	-2.50831E+03	**-3.85876E+03**	-3.43251E+03	**-5.16263E+03**	3.129276E+02	5.260082E+02
f14	3.581983E+03	**1.697717E+03**	5.834437E+03	**4.441905E+03**	4.928885E+03	**3.202206E+03**	3.731002E+02	5.252242E+02
f15	1.165543E+01	**2.970858E-01**	1.599610E+01	**8.646516E-01**	1.398104E+01	**5.079072E-01**	9.678164E-01	**9.516405E-02**
f16	2.311397E+04	**9.821317E-01**	1.290938E+05	**1.196974E+02**	6.437487E+04	**3.125772E+01**	2.044253E+04	**2.424396E+01**
f17	1.030751E+03	**5.768389E-02**	1.988745E+03	**2.277659E-01**	1.539028E+03	**1.153348E-01**	2.098559E+02	**2.809820E-02**
f18	8.501686E+00	**1.631931E-01**	1.111004E+01	**5.122384E-01**	1.017628E+01	**3.525345E-01**	4.732518E-01	**7.656788E-02**
f19	4.958953E+01	**1.493210E+01**	8.978232E+01	**1.968795E+01**	7.508255E+01	**1.821777E+01**	7.945404E+00	**9.831894E-01**
f20	7.424893E+00	**3.385581E-02**	1.871803E+01	**4.276894E-01**	1.467941E+01	**1.260871E-01**	2.195112E+00	**6.241278E-02**

4.4. NTSFA Performance on a Real-World Problem

K-means clustering algorithm is built on a partitioning way by virtue of data points which are continuously relocated to the nearest centroid. Though K-means clustering algorithm is popular, it has an essential weakness of getting into local optima that relies on the casually generated initial centroid values. In literature[32], the constructs of the integration of biomimetic optimization algorithms into K-means clustering are proposed. In their case, the clustering process is driven by the biomimetic optimization way; the relocation of centroids in each procedure is the variable. The algorithms are aimed at minimizing an objective function which is defined as follows.

$$f = \sum_{j=1}^{K} \sum_{i=1}^{N} \left\| x_{i,j} - cen_j \right\|^2 \tag{6}$$

In this paper, we apply NTSFA to integration into k-means clustering for mouse dataset. All the experimental setting is implemented according to the literature [32]. The experimental results are shown in Table IV and the best results obtained for each function are highlighted in bold.

Table IV. Performance Comparisons of Algorithms for Mouse Dataset

Algorithm	Best value	Worst Value	Mean Value
K-means	8.1132	8.4131	8.3255
C-ant	101.8220	111.9023	105.3098
C-firefly	8.1791	8.6702	8.3212
C-cuckoo	8.1132	8.1132	8.1132
C-bat	8.1132	8.1132	8.1132
C-wolf	8.1132	8.3248	8.2571
NTSFA	**2.2176**	**2.9222**	**2.6939**

The results in the Table IV give the significance of the proposed NTSFA as it is the best optimizer for the functions. From the table, it can be easily summarized that NTSFA has significantly outperformed the six other algorithms. All the values obtained by NTSFA are very low which denotes the searching accuracy Therefore, it can be concluded that NTSFA is an efficient optimization algorithm for practical problems.

Copyright © 2016 SERSC

5. Conclusion

In order to overcome the defaults of the basic firefly algorithm, a nonlinear time-varying step strategy firefly algorithm is presented. It uses nonlinear variable step size for all fireflies to better balance the algorithm's ability of exploration and exploitation. The performance of NTSFA has been tested on 20 test functions. The results clearly highlight that NTSFA behaves better. Finally, to further evaluate the performance of NTSFA, we apply it to integration into k-means clustering for mouse dataset. It is easy to see that NTSFA is an effective optimization tool.

Acknowledgements

The authors gratefully acknowledge the support of the National Natural Science Foundation of China (NSFC) (No. 61303209), the Universities Natural Science Foundation of Anhui Province (No.KJ2011A268) and the Universities Excellent Young Talents Foundation of Anhui Province (No.gxyqZD2016249).

References

[1] X.-S. Yang, "Nature-inspired metaheuristic algorithms", **(2010)**, Luniver press.
[2] I. Fister, "A comprehensive review of firefly algorithms," Swarm and Evolutionary Computation, vol. 13, no. 0, **(2013)**, pp. 34-46.
[3] A.H. Gandomi, "Firefly algorithm with chaos", Communications in Nonlinear Science and Numerical Simulation, vol. 18, no. 1, **(2013)**, pp. 89-98.
[4] T. Hassanzadeh and H.R. Kanan, "Fuzzy Fa: A Modified Firefly Algorithm", Applied Artificial Intelligence, vol. 28, no. 1, **(2014)**, pp. 47-65.
[5] M.Tuba and N. Bacanin, "Artificial Bee Colony Algorithm Hybridized with Firefly Algorithm for Cardinality Constrained Mean-Variance Portfolio Selection Problem," Applied Mathematics & Information Sciences, vol. 8,no. 6, **(2014)**, pp. 2831-2844.
[6] S. Yu, "A simple diversity guided firefly algorithm", Kybernetes, vol. 44, no. 1, **(2015)**, pp. 43-56.
[7] S.Yu, "A novel wise step strategy for firefly algorithm", International Journal of Computer Mathematics, vol. 91, no. 12, **(2014)**, pp. 2507-2513.
[8] S.Yu, S. Yang and S. Su, "Self-Adaptive Step Firefly Algorithm", Journal of Applied Mathematics, **(2013)**.
[9] J.L. Guerrero, A. Berlanga and J.M. Molina, "A Guided Mutation Operator for Dynamic Diversity Enhancement in Evolutionary Strategies", International Journal of Natural Computing Research (IJNCR), vol. 4, no. 2, **(2014)**, pp. 20-39.
[10] X.S. Yang, Firefly Algorithm, "Levy Flights and Global Optimization", Research and Development in Intelligent Systems Xxvi, **(2010)**, pp. 209-218.
[11] M.K. Marichelvam, T. Prabaharan and X.S. Yang, "A Discrete Firefly Algorithm for the Multi-Objective Hybrid Flowshop Scheduling Problems", Ieee Transactions on Evolutionary Computation, vol. 18, no. 2, **(2014)**, pp. 301-305.
[12] G.K. Jati and Suyanto, "Evolutionary Discrete Firefly Algorithm for Travelling Salesman Problem", Adaptive and Intelligent Systems, vol. 6943, **(2011)**, pp. 393-403.
[13] M. Bidar and H.R. Kanan, "Modified Firefly Algorithm Using Fuzzy Tuned Parameters", 2013 13th Iranian Conference on Fuzzy Systems (Ifsc), **(2013)**.
[14] L.D. Coelho, D.L.D. Bernert and V.C. Mariani, "A Chaotic Firefly Algorithm Applied to Reliability-Redundancy Optimization", 2011 Ieee Congress on Evolutionary Computation (Cec), **(2011)**, pp. 517-521.
[15] I.Fister, "Modified firefly algorithm using quaternion representation", Expert Systems with Applications, vol. 40, no. 18, **(2013)**, pp. 7220-7230.
[16] G.A. Trunfio, "Enhancing the firefly algorithm through a cooperative coevolutionary approach: an empirical study on benchmark optimisation problems", International Journal of Bio-Inspired Computation, vol. 6, no. 2, **(2014)**, pp. 108-125.
[17] G.G. Wang, "A New Improved Firefly Algorithm for Global Numerical Optimization", Journal of Computational and Theoretical Nanoscience, vol. 11, no. 2, **(2014)**, pp. 477-485.
[18] X.-S. Yang, Firefly algorithm, "Levy flights and global optimization", in Research and Development in Intelligent Systems XXVI, Springer, **(2010)**, pp. 209-218.
[19] A. Rahmani and S.A. MirHassani, "A hybrid Firefly-Genetic Algorithm for the capacitated facility location problem", Information Sciences, vol. 283, no. 0, **(2014)**, pp. 70-78.
[20] L.H. Guo, "An Effective Hybrid Firefly Algorithm with Harmony Search for Global Numerical Optimization", Scientific World Journal, **(2013)**.

 Copyright © 2016 SERSC

[21] M. Khajehzadeh, M.R. Taha and M. Eslami, "A New Hybrid Firefly Algorithm for Foundation Optimization", National Academy Science Letters-India, vol. 36, no. 3, **(2013)**, pp. 279-288.
[22] B.V. Nouri, P. Fattahi and R. Ramezanian, "Hybrid firefly-simulated annealing algorithm for the flow shop problem with learning effects and flexible maintenance activities", International Journal of Production Research, vol. 51, no. 12, **(2013)**, pp. 3501-3515.
[23] C. Man and G. Sheng, "An Improved Algorithm Based on Cooperative Particle Swarm Optimization", Journal of Harbin University of Science and Technology, vol. 15, no. 6, **(2010)**, pp. 51-53, 58.
[24] J. Liu, T. Yu and X. Wu, "PID Control Strategy Based on Modified PSO for AC Servo System", Journal of Harbin University of Science and Technology, vol. 18, **(2013)**, no. 1, pp.85-89.
[25] M.K. Sayadi, A. Hafezalkotob and S.G.J. Naini, "Firefly-inspired algorithm for discrete optimization problems: An application to manufacturing cell formation", Journal of Manufacturing Systems, vol. 32, no. 1, **(2013)**, pp. 78-84.
[26] S. Karthikeyan, P. Asokan and S. Nickolas, "A hybrid discrete firefly algorithm for multi-objective flexible job shop scheduling problem with limited resource constraints", International Journal of Advanced Manufacturing Technology, vol. 72, no. 9-12, **(2014)**, pp. 1567-1579.
[27] N. Poursalehi, A. Zolfaghari and A. Minuchehr, "Multi-objective loading pattern enhancement of PWR based on the Discrete Firefly Algorithm", Annals of Nuclear Energy, **(2013)**, pp. 57, no. 0, pp. 151-163.
[28] M. Farhoodnea, "Optimum placement of active power conditioners by a dynamic discrete firefly algorithm to mitigate the negative power quality effects of renewable energy-based generators", International Journal of Electrical Power & Energy Systems, vol. 61, **(2014)**, pp. 305-317.
[29] X.S. Yang, "Firefly algorithm, stochastic test functions and design optimisation", International Journal of Bio-Inspired Computation, vol. 2, no. 2, **(2010)**, pp. 78-84.
[30] X.-S. Yang, "Cuckoo Search and Firefly Algorithm: Overview and Analysis", in Cuckoo Search and Firefly Algorithm, Springer, **(2014)**, pp. 1-26.
[31] S. García, "A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 special session on real parameter optimization", Journal of Heuristics, vol. 15, no. 6, **(2009)**, pp. 617-644.
[32] R. Tang, "Integrating nature-inspired optimization algorithms to K-means clustering", in Digital Information Management (ICDIM), 2012 Seventh International Conference on 2012. IEEE.

Copyright © 2016 SERSC