Journal of Computational and Theoretical Nanoscience
Vol. 11, 477–485, 2014

RESEARCH ARTICLE

A New Improved Firefly Algorithm for Global Numerical Optimization

Gai-Ge Wang[1,*], Lihong Guo[2], Hong Duan[3], and Heqi Wang[2]

[1] *School of Computer Science and Technology, Jiangsu Normal University, Xuzhou, Jiangsu, 221116, China*
[2] *Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033, China*
[3] *School of Computer Science and Information Technology, Northeast Normal University, Changchun 130117, China*

A novel robust meta-heuristic optimization algorithm, which can be considered as an improvement of the recently developed firefly algorithm, is proposed to solve global numerical optimization problems. The improvement includes the addition of information exchange between the top fireflies, or the optimal solutions during the process of the light intensity updating. The detailed implementation procedure for this improved meta-heuristic method is also described. Standard benchmarking functions are applied to verify the effects of these improvements and it is illustrated that, in most situations, the performance of this improved firefly algorithm (IFA) is superior to or at least highly competitive with the standard firefly algorithm, a differential evolution method, a particle swarm optimizer, and a biogeography-based optimizer. Especially, this new method can accelerate the global convergence speed to the true global optimum while preserving the main feature of the basic FA.

Keywords: Global Optimization Problem, Benchmark Functions, Firefly Algorithm (FA), Biogeography-Based Optimization (BBO), Differential Evolution (DE), Particle Swarm Optimization (PSO), Top Fireflies, Lévy Flight.

1. INTRODUCTION

The process of optimization is searching a parameter in a function that produces an optimal solution. All of feasible values are available solutions and the extreme value is optimal solution. In general, optimization algorithms are applied to solve optimization problems. A simple classification way for optimization algorithms is considering the nature of the algorithms, and optimization algorithms can be divided into two main categories: deterministic algorithms, and stochastic algorithms. Deterministic algorithms using gradient such as hill-climbing have a rigorous move, and will generate the same set of solutions if the iterations commence with the same initial starting point. On the other hand, stochastic algorithms without using gradient often generate different solutions even with the same initial value. However, generally speaking, the final values, though slightly different, will converge to the same optimal solutions within a given accuracy. Generally, stochastic algorithms have two types: heuristic and metaheuristic.[1] Recently, nature-inspired metaheuristic algorithms perform powerfully and efficiently in solving modern non-linear numerical global optimization problems. To some extent, all metaheuristic algorithms strive for making balance between randomization and local search.[2]

Inspired by nature, these strong metaheuristic algorithms are applied to solve NP-hard problems such as UCAV path planning,[3,4] test-sheet composition,[5] dynamic WSN deployment,[6] and water, geotechnical and transport engineering.[7] Optimization algorithms cover all searching for extreme value problems. These kinds of metaheuristic algorithms carry out on a population of solutions and always find best solutions. During the 1950s and 1960s, computer scientists studied the possibility of making the conception of evolution as an optimization tool and this generated a subset of gradient free approaches named genetic algorithms (GA).[8,9] Since then many other nature-inspired metaheuristic algorithms have been produced, such as harmony search,[10] differential evolution (DE),[11–13] cuckoo search (CS),[14,15] particle swarm optimization (PSO),[16–18] biogeography-based optimization (BBO),[19–21] krill herd (KH) algorithm,[22] and more recently,

*Author to whom correspondence should be addressed.

1546-1955/2014/11/477/009 doi:10.1166/jctn.2014.3383

the firefly algorithm (FA)[23,24] that is inspired by firefly behavior in nature.

Firstly proposed by Yang in 2008, the firefly algorithm or firefly-inspired algorithm[25,26] 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 FA is quite powerful and efficient, and the performance of FA can be improved with feasible promising results.[27]

FA is a powerful algorithm in exploitation (i.e., local search) but at times it may trap into some local optima so that it cannot perform global search well. For firefly algorithm, the search depends completely on random walks, so a fast convergence cannot be guaranteed. Firstly presented here, a main improvement of adding the handling of top fireflies is made to the FA, which is made to speed up convergence, thus making the approach more feasible for a wider range of practical applications while preserving the attractive characteristics of the basic FA. Proposed approach is evaluated on ten standard benchmarking functions that have ever been applied to verify optimization algorithms in continuous optimization problems. Experimental results show that the IFA performs more efficiently and accurately than basic FA, BBO, PSO and DE.

The structure of this paper is organized as follows. Section 2 describes basic FA in brief. Our proposed approach IFA is presented in detail in Section 3. Subsequently, our method is evaluated through ten benchmarking functions in Section 4. In addition, the IFA is also compared with DE, BBO, FA and PSO in this section. Finally, Section 5 concludes the paper and discusses the future path of our work.

2. FIREFLY ALGORITHM

The firefly algorithm is a new ecology intelligence metaheuristic method[28] 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.

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 GA. The main update formula for any couple of two fireflies i and j at x_i and x_j is

$$x_i^{t+1} = x_i^t + \beta_0 e^{-\gamma r_{ij}^2}(x_i^t - x_j^t) + \alpha \varepsilon_i^t \tag{1}$$

where α is a parameter controlling the step size, β_0 is the attractiveness at $r = 0$, the second term is due to the attraction, while the third term is randomization with the vector of random variables ε_i being drawn from a distribution (e.g., Gaussian distribution). The distance between any pair of fireflies i and j at x_i and x_j can be the Cartesian distance $r_{ij} = \|x_i - x_j\|_2$ or the l_2-norm, relying on the practical application problems. In our present work, we take $\beta_0 = 1$, $\alpha \in [0, 1]$, and $\gamma = 1$. It can be shown that the limiting case $\gamma \to 0$ corresponds to the standard PSO.[16] In practice, if the inner loop (for j) is deleted and the brightness I_j is substituted by the current global best g^*, then FA essentially declines to the standard PSO.

3. OUR APPROACH: IMPROVED FIREFLY ALGORITHM (IFA)

Due to different fireflies working almost independently,[2] it may lack the exchange information between top fireflies. Therefore, in this paper, we add information exchange between top fireflies to the FA during the process of the light intensity updating. And then, based on exchange information between top fireflies, the IFA is proposed to solve the global numerical optimization problem.

For firefly algorithm, the search depends completely on random walks, so a fast convergence cannot be guaranteed. Firstly presented here, a main improvement of adding the handling of top fireflies is made to the basic FA, including two minor improvements, which are made to speed up convergence, thus making the approach more feasible for a wider range of real-world applications while preserving the attractive characteristics of the basic FA.

Algorithm 1 The algorithm of exchanging information among top fireflies

Begin

Step 1: *Set max Lévy flight step size A and golden ratio ϕ.*

Step 2: *for $i = 1$:NoTop (all top fireflies) do*

Current firefly at position x_i

Pick another firefly from the top fireflies

at random x_j
if $x_i = x_j$ ***then***
Calculate Lévy flight step size
$\phi \leftarrow A/G^2$
Perform Lévy flight from x_i *to generate new firefly* x_k
Evaluate the light intensity I_k *for firefly* x_k *by* $I_k = f(x_k)$
Choose a random firefly l from all fireflies
if $(I_k > I_l)$
Move firefly k towards l;
end if
else
$dx = |x_i - x_j|/\phi$
Move distance dx from the worst firefly to the best firefly to find x_k
Evaluate the light intensity I_k *for firefly* x_k *by* $I_k = f(x_k)$
Choose a random firefly l from all fireflies
if $(I_k > I_l)$ ***then***
Move firefly k towards l;
end if
end if
Step **3**: ***end for*** *i*
End.

The first improvement is adding Lévy flight to the FA with the step size α. Moreover, in the IFA, the value of α declines as the procedure proceeds (increasing generations). This is done for the similar reasons that the inertia constant is declined in the PSO,[29] i.e., to stimulate more localized searching as the fireflies, get closer to the optimal solution. The second improvement is to add information exchange between the fireflies striving for accelerating the convergence speed to the best solution. In the FA, there is no information exchange between fireflies and the searches are implemented independently in essence, i.e., different fireflies work almost independently.[2] In the IFA, a portion of the fireflies with the brightest light intensity are made up of a group of top fireflies. For every top firefly, a second firefly in this group is selected randomly and a novel firefly is then produced on the line connecting this pair of top fireflies. The distance along this line at which the novel firefly is situated is computed with the inverse of the golden ratio $\varphi = (1+\sqrt{5})/2$, so that it is much closer to the firefly with the brightest light intensity. In this rare case, firefly i and firefly j have the same light intensity, the new firefly is produced and located at the middle point. In this step, there is a probability that, the same firefly is selected twice. If so, a local Lévy flight search is carried out from the randomly selected firefly with step size $\alpha = A/G^2$. The detailed procedure for exchanging information between top fireflies contained in the improved firefly algorithm is presented in Algorithm 1, and the basic framework of improved firefly algorithm can be simply described as Algorithm 2. There are two parameters, the parameter of light absorption coefficient and the fraction of fireflies to make up the top fireflies, which need to be regulated in the IFA. Through testing on benchmarking functions, we can see that setting the parameter of light absorption coefficient to 1.0 and the fraction of fireflies placed in the top fireflies group to 0.25 produced the optimal results through a series of simulation experiments.

Algorithm 2 The algorithm of improved firefly algorithm
Begin
Step **1**: ***Initialization***. *Set the generation counter* $G = 1$; *Initialize the population of NP fireflies P randomly and each firefly corresponding to a potential solution to the given problem*; *define light absorption coefficient* Υ; *set controlling the step size a and the initial attractiveness* β_0 *at* $r = 0$; *set max Levy flight step size A, golden ratio* ϕ *and the fraction of fireflies placed in the top fireflies group* p_a.
Step **2**: *Evaluate the light intensity I for each firefly in P determined by* $f(x)$.
Step **3**: ***While*** *the termination criteria is not satisfied* ***or*** $G < MaxGeneration$ ***do***
Sort the population of fireflies P from best to worst by order of light intensity I for each firefly;
Exchange information between top fireflies by **Algorithm 1**
for $i = 1{:}NP$ *(all NP fireflies)* ***do***
for $j = 1{:}NP$ *(NP fireflies)* ***do***
if $(I_j < I_i)$,
Move firefly i towards j;
end if
Vary attractiveness with distance r via $exp[-\Upsilon r^2]$;
Evaluate new solutions and update light intensity;
end for *j*
end for *i*
Evaluate the light intensity I for each firefly in P determined by $f(x)$.
Sort the population of fireflies P from best to worst by order of light intensity I for each firefly;
$G = G + 1$;
Step **4**: ***end while***
Step **5**: *Post-processing the results and visualization*;
End.

The basic idea behind the IFA is that a population of *NP* fireflies (possible solutions) of dimension d use flashing behavior to communicate, attract other fireflies, hunt

RESEARCH ARTICLE

for pray and fly randomly through a d-dimensional search space updating their attractiveness β_i and light intensity I_i. Initially, we let NP fireflies be in the starting node at random, and each solution/firefly $x_i = [x_{i1}, x_{i2}, \ldots, x_{id}]$ is evaluated by a fitness function $I(x_i)$ which denoted by I_i. And then, the fireflies will choose the next solution in the search space according to exchanging information operation among top fireflies instead flying randomly used in FA. Consequently, this progress will enhance the original quality of the candidate solution.

4. SIMULATION EXPERIMENTS

In this section, we test the performance of the proposed meta-heuristic IFA to global numerical optimization through a series of experiments conducted on benchmarking functions.

To allow a fair comparison of running times, all the experiments were conducted on a PC with a Pentium IV processor running at 2.0 GHz, 512 MB of RAM and a hard drive of 160 Gbytes. Our implementation was compiled using MATLAB R2012a (7.14) running under Windows XP3. No commercial FA tools were used in the following experiments.

4.1. General Performance of IFA

In order to explore the benefits of IFA, in this subsection we compared its performance in global numeric optimization problem with four other population-based optimization methods, which are BBO, DE, FA, and PSO.

In all experiments, we will use the same set of FA parameter that are the fraction of fireflies placed in the top fireflies group $p_a = 0.25$, absorption coefficient $\lambda = 1.0$, max Lévy flight step size $A = 1.0$. For DE, BBO and PSO, we used the following parameters used in Ref. [30]. For BBO, habitat modification probability = 1, immigration probability bounds per gene = [0, 1], step size for numerical integration of probabilities = 1, maximum immigration and migration rates for each island = 1 and mutation probability = 0.005; for DE, a weighting factor $F = 0.5$ and a crossover constant $CR = 0.5$; For PSO, an inertial constant = 0.3, a cognitive constant = 1, and a social constant for swarm interaction = 1.

Well-defined problem sets are favorable for evaluating the performance of optimization methods proposed in this paper. Based on mathematical functions, benchmarking functions can be applied as objective functions to perform such tests. The properties of these benchmarking functions can be easily achieved from their definitions. Ten different benchmarking functions are applied to verify our proposed meta-heuristic algorithm IFA.

The benchmarking functions described in Table I are continuous functions. More details of all the benchmark functions can be found in Ref. [31].

Each algorithm had a population size $NP = 50$ and maximum generation $Maxgen = 50$. We ran 100 Monte Carlo simulations of each algorithm on each benchmarking function to get representative performances. Table II and Table III illustrate the results of the simulations. Table III shows the average minima found by each algorithm, averaged over 100 Monte Carlo runs. Table III shows the absolute best minima found by each algorithm over 100 Monte

Table I. Benchmark functions.

No.	Name	Definition
F01	Ackley	$f(\vec{x}) = 20 + e - 20 \cdot e^{-0.2\cdot\sqrt{(1/n)\sum_{i=1}^{n} x_i^2}} - e^{(1/n)\sum_{i=1}^{n}\cos(2\pi x_i)}$
F02	Fletcher-Powell	$f(\vec{x}) = \sum_{i=1}^{n}(A_i - B_i)^2$, $A_i = \sum_{j=1}^{n}(a_{ij}\sin\alpha_j + b_{ij}\cos\alpha_j)$, $B_i = \sum_{j=1}^{n}(a_{ij}\sin x_j + b_{ij}\cos x_j)$
F03	Griewank	$f(\vec{x}) = \sum_{i=1}^{n}\frac{x_i^2}{4000} - \prod\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$
F04	Quartic *with noise*	$f(\vec{x}) = \sum_{i=1}^{n}(i\cdot x_i^4 + U(0,1))$
F05	Rastrigin	$f(\vec{x}) = 10\cdot n + \sum_{i=1}^{n}(x_i^2 - 10\cdot\cos(2\pi x_i))$
F06	Rosenbrock	$f(\vec{x}) = \sum_{i=1}^{n-1}100\cdot(x_{i+1} - x_i^2)^2 + (x_i - 1)^2$
F07	Schwefel 1.2	$f(\vec{x}) = \sum_{i=1}^{n}\left(\sum_{j=1}^{i}x_j\right)^2$
F08	Schwefel 2.22	$f(\vec{x}) = \sum_{i=1}^{n}\lvert x_i\rvert + \prod_{i=1}^{n}\lvert x_i\rvert$
F09	Sphere	$f(\vec{x}) = \sum_{i=1}^{n}x_i^2$
F10	Step	$f(\vec{x}) = 6\cdot n + \sum_{i=1}^{n}\lfloor x_i\rfloor$

Note: *In benchmark function F02, the matrix elements $\mathbf{a}_{n\times n}, \mathbf{b}_{n\times n} \in (-100, 100)$, $\alpha_{n\times 1} \in (-\pi, \pi)$ are draw from uniform distribution.[32]

Table II. Mean normalized optimization results on benchmarking functions. The values shown are the minimum objective function values found by each algorithm, averaged over 100 Monte Carlo simulations.

	BBO	DE	FA	IFA	PSO
F01	1.77	2.60	1.09	**1.00**	3.46
F02	**1.00**	3.20	2.65	1.81	6.97
F03	4.53	10.85	1.49	**1.00**	40.33
F04	32.22	134.22	3.64	**1.00**	1.00E3
F05	**1.00**	3.92	1.50	1.11	4.63
F06	3.80	9.28	1.52	**1.00**	19.79
F07	6.32	14.89	1.74	**1.00**	11.71
F08	**1.00**	2.83	1.65	1.12	6.32
F09	7.31	18.96	1.49	**1.00**	71.92
F10	4.82	13.00	1.63	**1.00**	49.80

Notes: *The values are normalized so that the minimum in each row is 1.00. These are not the absolute minima found by each algorithm, but the average minima found by each algorithm.

Table III. Best normalized optimization results on benchmark functions. The values shown are the minimum objective function values found by each algorithm, averaged over 100 Monte Carlo simulations.

	BBO	DE	FA	IFA	PSO
F01	3.94	5.57	1.17	**1.00**	9.32
F02	1.48	6.67	1.15	**1.00**	14.68
F03	4.12	8.08	**1.00**	**1.00**	42.81
F04	1.43E3	1.60E4	1.20	**1.00**	7.52E4
F05	1.11	6.30	1.20	**1.00**	6.98
F06	3.01	10.09	1.18	**1.00**	10.37
F07	10.48	38.60	1.81	**1.00**	23.83
F08	3.35	11.31	**1.00**	1.42	22.95
F09	75.36	378.00	1.56	**1.00**	1.53E3
F10	43.80	234.60	**1.00**	2.00	1.07E3

Notes: *The values are normalized so that the minimum in each row is 1.00. These are the absolute best minima found by each algorithm.

Carlo runs. In other words, shows the average performance of each algorithm, while Table III shows the best performance of each algorithm. Note that the normalizations in the tables are based on different scales, so values cannot be compared between the two tables.

From Table II, we see that, on average, IFA is the most effective at finding objective function minimum on seven of the ten benchmarks (F01, F03, F04, F06, F07, F09, and F10). BBO is the second most effective, performing best on the other three of the ten benchmarks (F02, F05, and F08) when multiple runs are made. Table III shows that IFA performs the best on seven of the ten benchmarks (F01–F07, and F09), while FA performs the best at finding objective function minimum on the other three benchmarks (F03, F08, and F10) when multiple runs are made. By carefully studying the results in Table III, we can recognize that for F03 FA performs equally to IFA.

Furthermore, convergence graphs of BBO, DE, FA, IFA and PSO are shown in Figures 1–10 which represent the process of optimization. The values shown in Figures 1–10 are the best objective function optimum achieved from

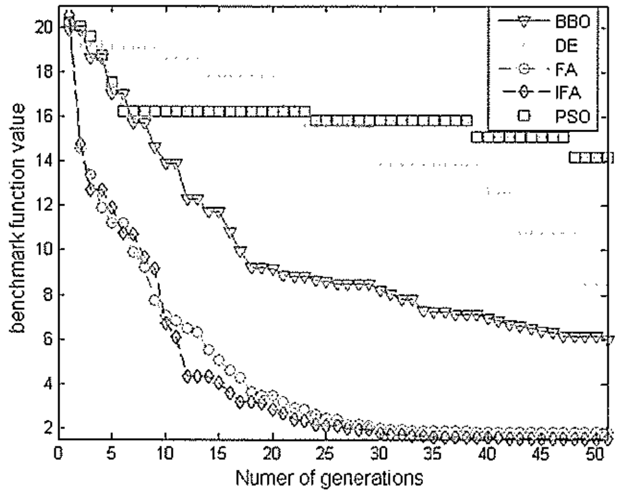

Fig. 1. Comparison of the performance of the different methods for the F01 Ackley function.

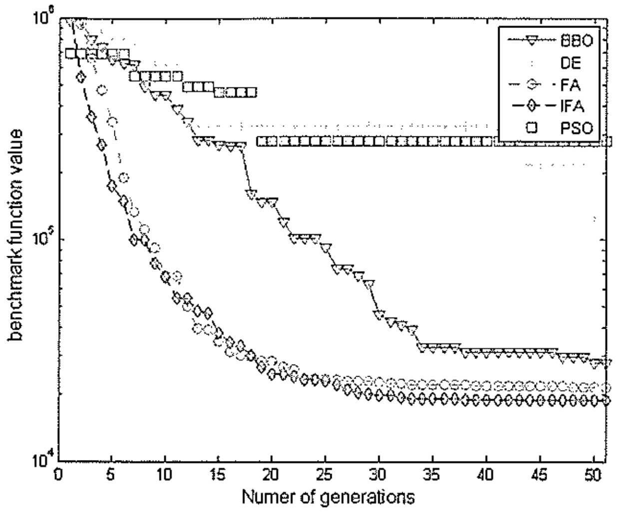

Fig. 2. Comparison of the performance of the different methods for the F02 Fletcher-Powell function.

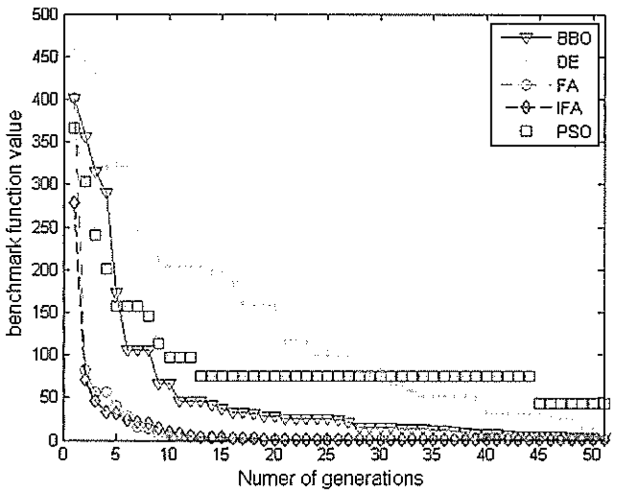

Fig. 3. Comparison of the performance of the different methods for the F03 Griewank function.

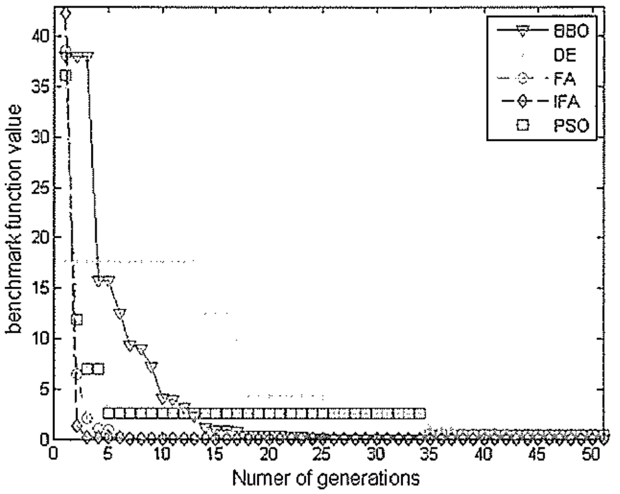

Fig. 4. Comparison of the performance of the different methods for the F04 quartic (*with noise*) function.

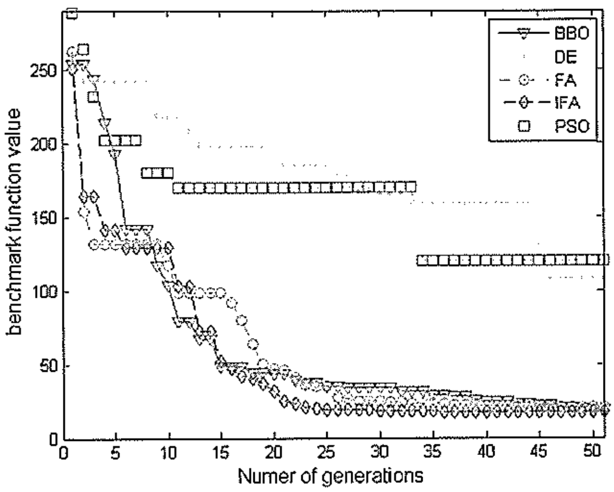

Fig. 5. Comparison of the performance of the different methods for the F05 Rastrigin function.

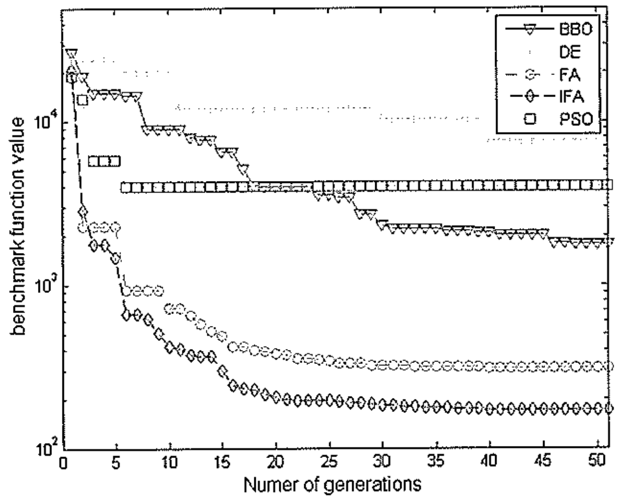

Fig. 7. Comparison of the performance of the different methods for the F07 Schwefel 1.2 function.

100 Monte Carlo simulations, which are the true objective function value, not normalized.

Figure 1 shows the results obtained for the five methods when the F01 Ackley function is applied. From Figure 1 and Table IV, we can draw the conclusion that, IFA is slightly superior to basic FA during the process of optimization, while outperforming all other methods in this multimodal benchmarking function. Here, FA shows the almost same fast convergence rate initially, however it is outperformed by IFA after 10 generations. Obviously, IFA and FA outperform BBO, PSO and DE during the whole process of searching the global minimum.

Figure 2 illustrates the optimization results for F02 Fletcher-Powell function. In this benchmarking problem, it is obvious that IFA outperforms all other methods during all progress of optimization.

Figure 3 shows the optimization results for F03 Griewank function. From Figure 3 and Table IV, we can see that, IFA and FA have the same convergence rate during the whole optimization process in this multimodal function. All though slower, DE and BBO eventually find the global minimum, especially BBO.

Figure 4 shows the results for F04 Quartic (*with noise*) function. For this benchmarking function, the figure shows that there is little difference between the performance of BBO, DE, FA, PSO and IFA at the end of optimization. But from Figure 4 and Table IV, we can draw the conclusion that, IFA is slightly superior to basic FA during the optimization process in this multimodal benchmarking function. All though slower, DE and PSO eventually find the global minimum.

Figure 5 shows the performance achieved for F05 Rastrigin function. For this unimodal function, IFA is slightly superior to basic FA and BBO, while outperforming all other methods. Here, FA and IFA show an almost same fast convergence rate initially, however IFA

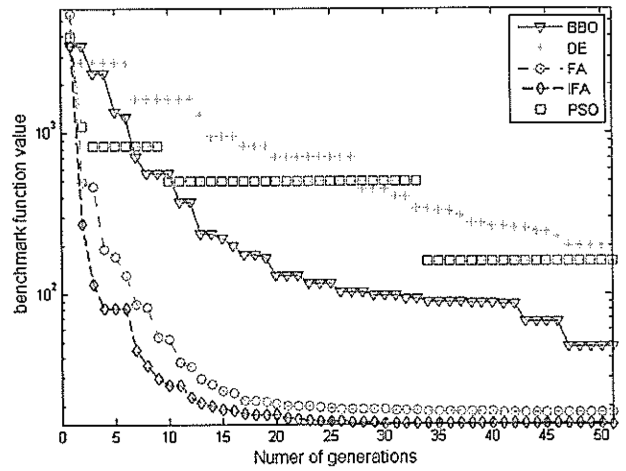

Fig. 6. Comparison of the performance of the different methods for the F06 Rosenbrock function.

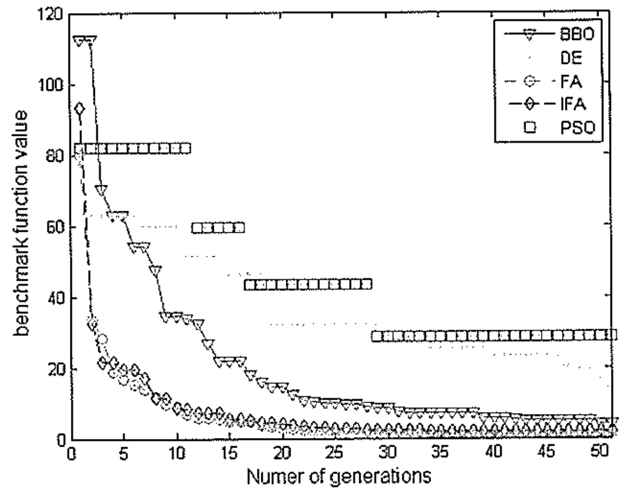

Fig. 8. Comparison of the performance of the different methods for the F08 Schwefel 2.22 function.

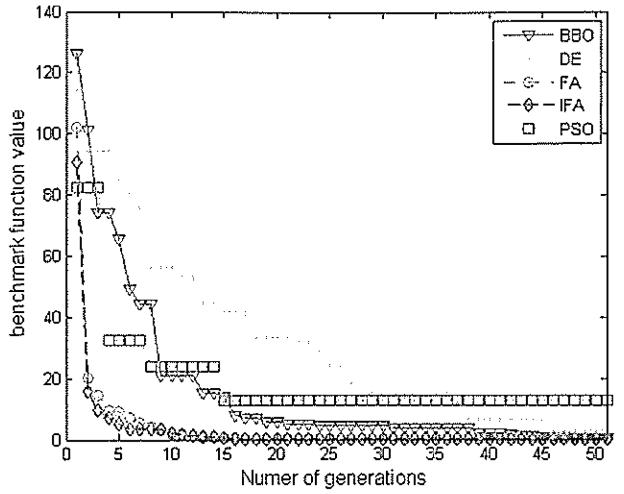

Fig. 9. Comparison of the performance of the different methods for the F09 sphere function.

outperforms FA after 10 generations. IFA outperforms both PSO and DE during all optimization process. It also shows a much higher convergence rate than DE and PSO. The BBO has a slower initial convergence rate but does finally match the performance of IFA.

Figure 6 shows the results achieved for the five methods when using the F06 Rosenbrock function. In this relatively simple unimodal benchmarking function, FA and IFA show an initial fast convergence towards the known minimum, as the procedure proceeds, IFA gets closer to the minimum than FA. The IFA methods clearly outperform the FA, BBO, DE and PSO algorithm, especially BBO, DE and PSO. BBO, DE, FA and PSO do not manage to succeed in this benchmark function within the maximum number of generations. At last, DE and PSO converge to the same value.

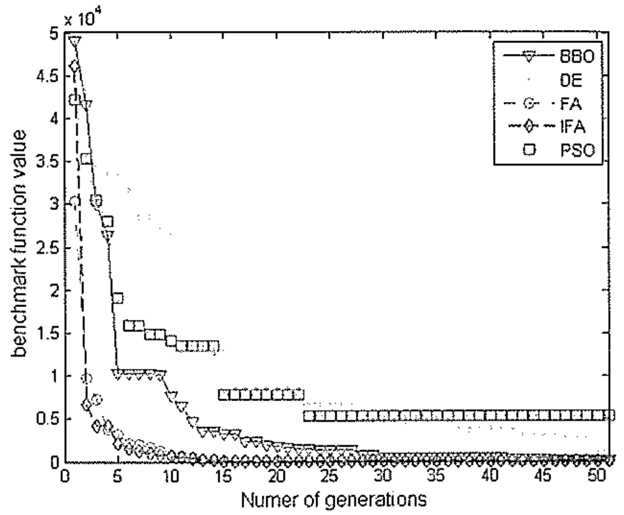

Fig. 10. Comparison of the performance of the different methods for the F10 step function.

Figure 7 shows the optimization results for the F07 Schwefel 1.2 function. Very clearly, IFA has the fastest convergence rate finding the global minimum after 4 generations. BBO, PSO, DE and FA do not manage to succeed in this problem within the maximum number of generations. This highlights the lack of information exchange between fireflies in the standard FA, meaning FA cannot take advantage of the symmetry of this function as the IFA methods can. Here, IFA significantly outperforms all other approaches.

Figure 8 shows the results for F08 Schwefel 2.22 function. From Figure 8 and Table IV, we can see that, FA is slightly superior to basic IFA during the optimization process in this relatively simple unimodal benchmarking function. All though slower, BBO eventually find the global minimum more or less. It is clear that IFA, FA outperform BBO, PSO and DE during the process of searching the global minimum, and BBO, PSO and DE do not manage to succeed in this relatively simple problem within the maximum number of generations, especially PSO and DE.

Figure 9 shows the equivalent results for the F09 Sphere function. From Figure 9 and Table IV, we can see that, IFA is slightly superior to basic FA during the optimization process in this relative simple unimodal benchmarking function. Clearly, IFA and FA outperform all the other methods in this benchmarking function. IFA and FA outperform both BBO and DE initially, all though slower, BBO and DE eventually find the global minimum at the end of optimization process, especially BBO, which has a slower initial convergence rate but does finally match the performance of IFA. PSO does not manage to succeed in this relatively simple problem within the maximum number of generations.

Figure 10 shows the results for F10 Step function. From Figure 10 and Table IV, we can see that, FA is slightly superior to IFA during the optimization process in this relative simple unimodal benchmarking function. All though slower, BBO eventually find the global minimum. It is clear that IFA, FA, DE and BBO outperform PSO during the process of searching the global minimum, and PSO does not manage to succeed in this relatively simple problem within the maximum number of generations.

4.2. Discussion

In the IFA, the fireflies fly in the sky to find food/prey (i.e., best solutions). Three other parameters are: the step size (α) that acts in a similar role as the cooling schedule in the traditional simulated annealing optimization method,[33] the fraction of fireflies (p_a) to make up the top fireflies, and the absorption coefficient (γ) that regulates the attractiveness. The appropriately update for the step size (α) and absorption coefficient (γ_i) 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.

> For all of the standard benchmarking functions that have been considered, the IFA has been demonstrated to perform better than or equally to the standard FA. For all the considered functions, the IFA's performance is at least quite competitive with or superior to the BBO, with the IFA performing significantly better in some functions. The IFA performs excellently and efficiently because of its ability to simultaneously carry out a local search, still searching globally at the same time. It succeeds in doing this due to the information exchange between the top fireflies and global search via Lévy flights. A similar behavior may be performed in the PSO by using multi-swarm from a particle population initially. However, IFA's advantages include performing simply and easily, and only have three parameters to regulate. When comparing IFA to DE in all benchmarking functions, IFA converges with a significantly faster speed. At last, DE does reach as close (or in some cases closer) to the optimum as IFA but demands many more generations to get this. The work carried out here demonstrates the IFA to be robust over all kinds of benchmarking functions.

5. CONCLUSION AND FUTURE WORK

This paper proposed an improved meta-heuristic IFA method for optimization problem. A novel type of FA model has been presented, and an improvement is applied to exchange information between top fireflies during the process of the light intensity updating. This new method can speed up the global convergence rate without losing the strong robustness of the basic FA. The detailed implementation procedure for this meta-heuristic method is also described. Compared with the BBO, DE, basic FA and PSO, the experimental results illustrate that this approach is a feasible and effective way in global numerical optimization problems.

In the field of optimization, there are many issues worthy of further study, and efficient optimization method should be developed depending on the analysis of specific engineering problem. Our future work will focus on the two issues: on the one hand, we would apply our proposed approach IFA to solve practical engineering optimization problems, such as investigate transport properties of nanostructures,[34] displacement-momentum commutation,[35] influence of quantizing magnetic field,[36] pressure and temperature induced non-linear optical properties.[37] And, obviously, IFA can become a fascinating method for real-world engineering optimization problems; on the other hand, we would develop new meta-hybrid approach to solve optimization problem.

Acknowledgments: This work was supported by State Key Laboratory of Laser Interaction with Material Research Fund under Grant No. SKLLIM0902-01 and Key Research Technology of Electric-discharge Non-chain Pulsed DF Laser under Grant No. LXJJ-11-Q80.

References

1. X. S. Yang, A. H. Gandomi, S. Talatahari, and A. H. Alavi, Metaheuristics in Water, Geotechnical and Transport Engineering, Elsevier, London, UK (2013).
2. X. S. Yang, Nature-Inspired Metaheuristic Algorithms, 2nd ed, Luniver Press, Frome (2010).
3. G. Wang, L. Guo, H. Duan, L. Liu, and H. Wang, *Int. J. Hybrid Inf. Technol.* 5, 3 (2012).
4. G. Wang, L. Guo, H. Duan, L. Liu, and H. Wang, *ScientificWorldJournal* (2012).
5. H. Duan, W. Zhao, G. Wang, and X. Feng, *Math. Probl. Eng.* 2012, 22 (2012).
6. G. Wang, L. Guo, H. Duan, L. Liu, and H. Wang, *J. Sens. Actuator Netw.* 1, 2 (2012).
7. A. H. Gandomi, X. S. Yang, S. Talatahari, and A. H. Alavi, Metaheuristic Applications in Structures and Infrastructures, Elsevier, Waltham, MA (2013).
8. A. Jamshidnezhad and M. J. Nordin, *J. Comput. Theor. Nanosci.* 9, 8 (2012).
9. X. Xia, Q. Zhang, B. Wang, and Y. Zhao, *J. Comput. Theor. Nanosci.* 9, 7 (2012).
10. G. Wang and L. Guo, *J. Appl. Math.* 2013, 21 (2013).
11. R. Storn and K. Price, *J. Global Optim.* 11, 4 (1997).
12. A. H. Gandomi, X.-S. Yang, S. Talatahari, and S. Deb, *Comput. Math. Appl.* 63, 1 (2012).
13. A. Abraham, R. K. Jatoth, and A. Rajasekhar, *J. Comput. Theor. Nanosci.* 9, 2 (2012).
14. G. Wang, L. Guo, H. Duan, L. Liu, H. Wang, and W. Jianbo, *J. Inform. Comput. Sci.* 9, 16 (2012).
15. G. Wang, L. Guo, H. Duan, H. Wang, L. Liu, and M. Shao, *ScientificWorldJournal* 2012 (2012).
16. J. Kennedy and R. Eberhart, Particle swarm optimization, *Proceeding of the IEEE International Conference on Neural Networks*, IEEE, Perth, Australia (1995).
17. A. H. Gandomi, G. J. Yun, X.-S. Yang, and S. Talatahari, *Commun. Nonlinear Sci. Numer. Simulat.* 18, 2 (2013).
18. S. Talatahari, X. S. Yang, R. Sheikholeslami, and A. H. Gandomi, *J. Comput. Theor. Nanosci.*
19. D. Simon, *IEEE. T. Evolut. Comput.* 12, 6 (2008).
20. G. Wang, L. Guo, H. Duan, L. Liu, H. Wang, and M. Shao, *Adv. Sci. Eng. Med.* 4, 6 (2012).
21. G. Wang, L. Guo, H. Duan, L. Liu, H. Wang, and M. Shao, *J. Comput. Theor. Nanosci.* (2013).
22. G. Wang, L. Guo, A. H. Gandomi, L. Cao, A. H. Alavi, H. Duan, and J. Li, *Math. Probl. Eng.* 2013, 14 (2013).
23. X.-S. Yang, S. S. Sadat Hosseini, and A. H. Gandomi, *Appl. Soft Comput.* 12, 3 (2012).
24. A. H. Gandomi, X. S. Yang, S. Talatahari, and A. H. Alavi, *Commun. Nonlinear Sci. Numer. Simulat.* 18, 1 (2013).
25. X. S. Yang, Nature-Inspired Metaheuristic Algorithms, Luniver Press, Frome (2008).
26. A. H. Gandomi, X.-S. Yang, and A. H. Alavi, *Computers and Structures* 89, 23 (2011).
27. X. S. Yang, *Int. J. Bio-Inspired Computation* 2, 2 (2010).
28. E. Talatahari, S. Talatahari, A. H. Gandomi, and X. S. Yang, *J. Comput. Theor. Nanosci.*
29. D. Bratton and J. Kennedy, Defining a standard for particle swarm optimization, *Proceedings of the 2007 IEEE Swarm Intelligence Symposium* (*SIS* 2007) IEEE (2007).

30. G. Wang, L. Guo, H. Duan, H. Wang, L. Liu, and J. Li, *Neural Comput. Appl.* (2012).
31. X. Yao, Y. Liu, and G. Lin, *IEEE. T. Evolut. Comput.* 3, 2 (1999).
32. R. Fletcher and M. J. D. Powell, *The Computer Journal* 6, 2 (1963).
33. S. Kirkpatrick, C. D. Gelatt Jr, and M. P. Vecchi, *Science* 220, 4598 (1983).
34. T. Ono, Y. Fujimoto, and S. Tsukamoto, *Quantum Matter* 1, 1 (2012).
35. V. Sajfert, P. Mali, N. Bednar, N. Pop, D. Popov, and B. Tosic, *Quantum Matter* 1, 2 (2012).
36. N. Paitya, S. Bhattacharya, D. De, and K. Ghatak, *Quantum Matter* 1, 1 (2012).
37. M. Narayanan and A. J. Peter, *Quantum Matter* 1, 1 (2012).

Received: 29 November 2012. Accepted: 24 December 2012.