An Improved Flower Pollination Algorithm to Solve Function Optimization Problem

Xiao-Xu Ma, and Jie-Sheng Wang

Abstract—**Flower pollination algorithm (FPA) is a new swarm intelligence optimization algorithm which simulates flower pollination. For all intelligent optimization algorithms, the performance analysis of parameters can affect the convergence speed, convergence precision and global optimization ability. In this paper, an enhanced variation of flower pollination algorithm (MFPA) was proposed. The convergence speed and algorithm searching precision are determined by the switching probability and λ in Levy flight. The simulation experiments are carried out by using the six typical test functions to discuss this influence. The simulation results show that the switching probability is less sensitive to the MPFA algorithm. With reasonable setting of MFPA parameters, the search precision and convergence speed can be improved effectively.**

Index Terms—**flower pollination algorithm, clonal selection strategy, improved pollen pollination operator**

I. Introduction

The process of optimization is essentially the choice of a vector within a search space. The selected vector can maximize or minimize an objective function to provide the best solution. Generally, modern intelligent approaches are used to deal with these types of optimization problems. Such optimization approaches can be categorized into two groups in view of their natures: deterministic and random intelligent approaches[1].The function optimization presents a formalized framework for modelling and solving some certain problems. Given an objective function, it takes a number of parameters as its inputs, whose goal is to find the combination of parameters and return the best value. This framework is abstract enough that a wide variety of different problems can be interpreted as function optimization problems [2].

However, the traditional function optimization algorithm is used to solve the typical problem with small dimension, often not applicable in practice. So people focus on the nature. Nature provides rich models to solve these problems (such as fireflies, bats, ants). People discovered the swarm intelligence optimization algorithm by simulating natural biological systems. These models could stimulate computer scientists using household non-traditional tools to solve the application problems [3]. Now a lot of swarm intelligence optimization algorithm is proposed, such as particle swarm optimization (PSO) [4], ant colony algorithm (ACO) [5], bat algorithm (BA) [6], harmonious algorithm (SLO) [7], chicken swarm algorithm (CSO) [8], firefly algorithm [9] etc. They can be used in the dictionary learning remote sensing data, automotive safety integrity level positioning, economic dispatch, composition and examples of the Cloud Service Composition of QOS awareness. Obviously, the study of swarm intelligence optimization has become an important research direction.

Flower pollination algorithm (FPA) is a swarm intelligence optimization algorithm proposed by Cambridge university scholar Yang in 2012 to simulate the flower pollination [10]. Although this algorithm is simple and has few parameters, the superiority and convenience are more prominent. At present, many scholars at home and abroad are crazed about studying the algorithm. FPA has been used to solve the economic load dispatch and combined economic emission dispatch problems in power systems [11], the placement of distribution transformers in a low-voltage grid [12], the multi-objective flower pollination algorithm applied in electrical load forecasting [13], the assembly sequence optimization [14], the hybrid flower pollination algorithm (HFPA) used to solve the dynamic multi-objective optimization scheduling (DMOOD) thermal system [15] and the combined economic and emission dispatch solution [16]. In this paper, FPA and clone selection (CS) strategy are combined to form MFPA algorithm to solve the function optimization problems. In order to prove the superiority of this algorithm, the comparison and analysis of parameter performance are carried out through the simulation experiments. The paper is organized as follows. In Section 2, the improved FPA are introduced. The simulation experiments and results analysis are introduced in details in Section 3. Finally, the conclusion illustrates the last part.

II. Improved Flower Pollination Algorithm

A. Flower Pollination Algorithm

Flower pollination algorithm (FPA) is swarm intelligence optimization algorithm proposed by to simulate flower pollination [10]. The dynamic control on the process of global search and local search is realized by adjusting parameter *P*. This method solves the balance between global

Manuscript received December 20, 2017; revised April 14, 2018.This work was supported by the the Basic Scientific Research Project of Institution of Higher Learning of Liaoning Province (Grant No. 2017FWDF10), the Program for Research Special Foundation of University of Science and Technology of Liaoning (Grant No. 2015TD04) and the Opening Project of National Financial Security and System Equipment Engineering Research Center (Grant No. USTLKFGJ201502).

Xiao-Xu Ma is a postgraduate student in the School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, PR China (e-mail: 1193460040@qq.com).

Jie-Sheng Wang is with the School of International Finance and Banking, University of Science and Technology Liaoning, Anshan, 114051, PR China; National Financial Security and System Equipment Engineering Research Center, University of Science and Technology Liaoning. (phone: 86-0412-2538246; fax: 86-0412-2538244; e-mail: wang_jiesheng@126.com).

(Advance online publication: 28 August 2018)

search and local search and uses the Levy flight to make it have a good global optimization capability [17]. Flower pollination process is achieved through cross-pollination or self-pollination in the nature. The position of the pollinator is random or similar to random in the process of pollination. In order to simulate the way of flower pollination, the following four rules are set.

Rule 1: The biotic and cross-pollination can be recognized as a global pollination, where the pollinators follow the Levy distribution.

Rule 2: The abiotic and self-pollination can be interpreted as a local pollination.

Rule 3: The flower constancy property can be considered as a reproduction ratio that is proportional to the degree of similarity between two flowers.

Rule 4: Due to the physical proximity and wind, local pollination has a slight advantage over global pollination. Both are controlled by the value of the variable $P \in [0,1]$.

In the global pollination, the fittest reproduction is ensured through insects that can travel for long distances. If the fittest is represented as $g*$, the flower constancy and the first rule can be mathematically formulated as follows:

$$x_i^{t+1} = x_i^t + \gamma L(g* - x_i^t) \tag{1}$$

where, x_i^t is a solution vector at iteration t, $g*$ is the best found solution at iteration t, γ represents the step size scaling factor, and L is the pollination strength or the step size. The insect's long moves can be mimicked using Levy flight. For this reason, the step size L is derived from the Levy distribution.

$$L \approx \frac{\lambda\Gamma(\lambda)\sin(\frac{\gamma\pi}{2})}{\pi}\frac{1}{s^{1+\lambda}} \tag{2}$$

where, $\lambda = 1.5$ and $\lambda\Gamma$ represents the typical Gamma function.

The local pollination based on Rule 2 can be formulated as follows:

$$x_i^{t+1} = x_i^t + \in (x_j^t - x_k^t) \tag{3}$$

where, x_j^t and x_k^t are pollens (solution vectors) that are transferred from different flowers, but these flowers belong to a single plant species. It simulates the flower constancy in a small neighborhood. The variable $\in$ is derived from a uniform distribution in the range $[0,1]$.

The pollination process can be either local or global, so a switch probability p is presented to switch between the two types of pollination (Rule 4).

B. *Clonal Selection Strategy*

CSA is inspired by the clonal selection theory that is presented in 1959 [9]. The main characteristics of the immune system can be summarized as follows.

1) The immune system has a memory set that remembers the previous attacks.

2) The most stimulated antibodies are selected for cloning.

3) The poorly and nonstimulated antibodies are removed.

4) The activated immune cells have undergone a hypermutation process.

5) The human antibodies' diversity is maintained (repertoire diversity).

The resultant clones are enhanced through mutation, for a better matching with the antigens. CSA is a population-based algorithm. The population consists of a set of antibodies or solutions that are designed for a specific problem. The cloning process generates typical copies from the highest affinity antibodies, so there is a direct relationship between an antibody's affinity and its cloning ratio. The resultant clones are enhanced through mutation for a better matching with the antigens. The mutated antibodies (solutions) are added to the current population, then all antibodies are ranked. After that, the best antibodies are chosen as memory cells. The memory cells are considered the best set of solutions for the optimization problem we are intending to solve. Performing meta-dynamics is the last step in the CSA, where the lowest affinity solutions are replaced by randomly generated ones. This step is performed to keep the population diversity.

C. *Improved Pollen Pollination Operator*

On the basis of pollen pollination function, an enhanced variation of flower pollination algorithm MFPA was introduced. The running speed and the convergence speed are optimized. There are better capability for nonlinear problems or more complex problems in practice. Experimental results showed that the solutions generated from random walks, in the local pollination, converge faster than the Levy flight ones, so we replaced the Levy flights by random walks.

Random walks are drawn from random uniform distribution in $[0,1]$. Before applying local pollination, high affinity solutions are cloned proportional to their affinity; then local pollination is performed. The local pollination was modified by introducing a step-size scaling factor γ_2. A preliminary parametric study showed that γ_2 =3 works well for all test cases. Eq.(4) was used to analyze the modified pollen pollination algorithm.

$$MAE = \frac{\sum_{i=1}^{N} |m_i - Ki|}{N} \tag{4}$$

Where, m_i indicates the mean of optimal values, K_i is the corresponding global optimal value, and N represents the number of samples. In our case, N is the number of test functions.

In order to avoid sticking in the local minimum the algorithm checks if the best solution $g*$ is not changed for 100 successive iterations with a value not greater than 10e-6. If so, all the population Pop is replaced by a new randomly generated one after keeping the best found solution $g*$. This step increases the exploration to a high extent and is similar to the meta-dynamics step in the CSA.

III. Simulation Experiments and Results Analysis

A. Test Functions

In the simulation experiments, six typical functions are adopted to verify the performance of MFPA. The simulation environment adopts the WINDOW7 operating system, MATLAB software for simulation. The testing functions are shown in Tab.1, where f_1 - f_2 are unimodal functions and f_3 - f_6 are the multimodal functions.

B. Simulation Experiments and Results Analysis

(1) Change of Single Variable P

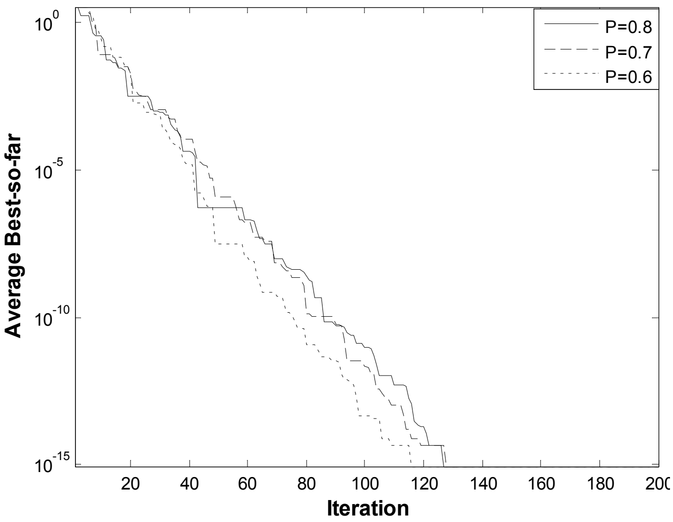
The initialization parameters of MFPA are set as: the population size n is 50, the number of iterations max_it is 500, $x_i = 2$, $\lambda = 1.5$, $\gamma_1 = 1$ and $\gamma_2 = 3$ .In order to reduce the influence of random disturbance, the independent operating for each test function is carried out 10 times. The optimal value and average values of MFPA in different probability of switch P are shown in Tab. 2. The simulation results of the six test functions are shown in Fig. 1.

TABLE 1. SIMULATION TESTING FUNCTIONS

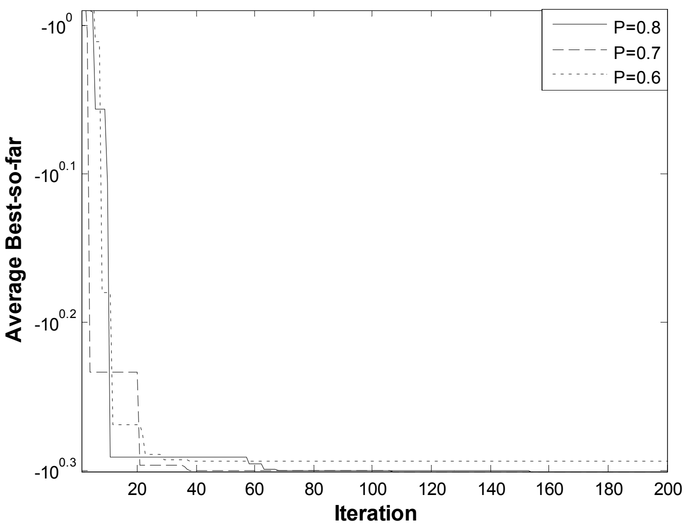
Function	**Name**	**Expression**	**Range**
f_1	Ackley	$f_1(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}X_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$	[-32,32]
f_2	Rotated Hyper-Ellipsoid	$f_2(x) = \sum_{i=1}^{d}\sum_{j=1}^{i} x_j^2$	[-5.536,65.536]
f_3	Schwefel	$f_3(x) = 418.9829d - \sum_{i=1}^{d} x_i \sin(\sqrt{\lvert x_i \rvert})$	[-100,100]
f_4	Michaelmas	$f_4(x) = -\sum_{i=1}^{d}\sin(x_i)\sin^{2m}(\frac{ix_i^2}{\pi})$	[0,π]
f_5	Drop-Wave	$f_5(x) = -\frac{1+\cos(12\sqrt{x_1^2+x_2^2})}{0.5(x_1^2+x_2^2)+2}$	[-5.12,5.12]
f_6	Rastrigin’	$f_6(x) = n*10 + \sum_{i=1}^{d}(x_i^2 - 10\cos(2\pi x_i))$	[-5.12,5.12]

TABLE 2. PERFORMANCE COMPARISON OF MFPA UNDER DIFFERENT

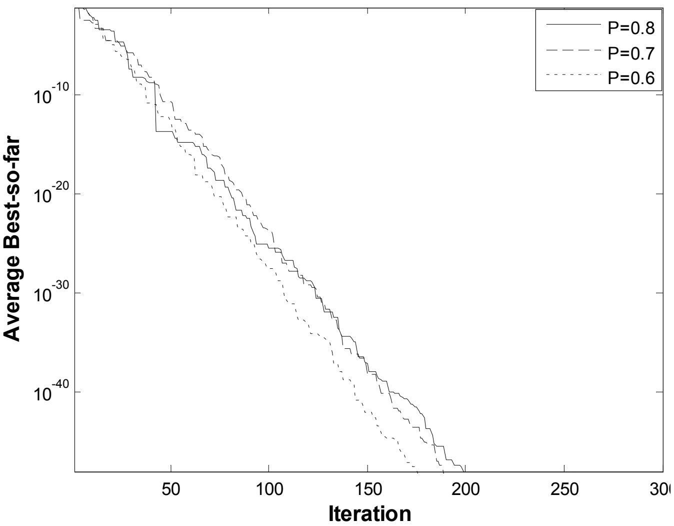
Function	Result	Simulation results of MFPA under different		
		0.8	0.7	0.6
f_1	optimum	8.8818e-016	8.8818e-016	8.8818e-016
	average	0.015709	0.012521	0.014927
	std	0.1811	0.16682	0.1775
f_2	optimum	3.1246e-046	2.4306e-048	9.8397e-052
	average	0.030934	0.014937	0.032911
	std	0.48667	0.26264	0.40367
f_3	optimum	2.5455e-005	2.5455e-005	2.5455e-005
	average	0.22287	0.33214	0.14301
	std	2.9657	8.0506	4.3608
f_4	optimum	-1.8013	-1.8013	-1.8013
	average	-1.8010	-1.8008	-1.8010
	std	0.0043	0.0103	0.0045
f_5	optimum	-1	-1	-1
	average	-0.9978	-0.9990	-0.9991
	std	0.0207	0.0072	0.0075
f_6	optimum	0	0	0
	average	0.0091	0.0104	0.0136
	std	0.2302	0.1239	0.2479



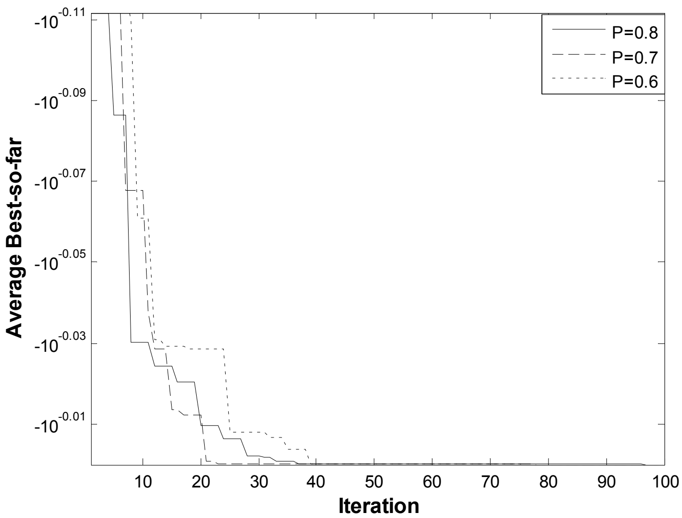
(a) Function f_1



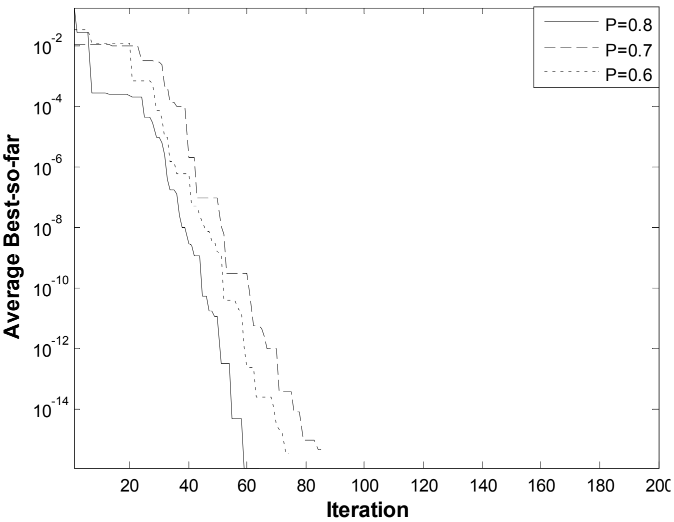
(d) Function f_4



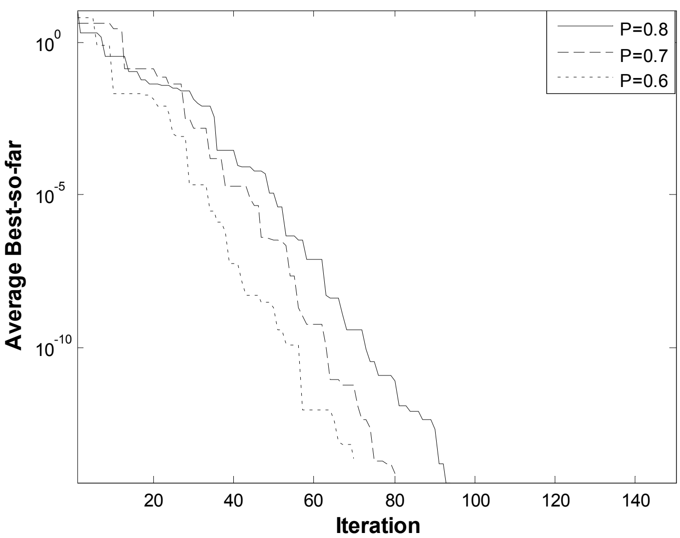
(b) Function f_2



(e) Function f_5



(c) Function f_3



(f) Function f_6

Fig. 1 Simulation results of MFPA under different.

It can be seen from the convergence curves and the numerical results of six functions after 500 iterations run independently 10 times. When P from 0.8 to 0.6, the optimal values of f_1 - f_6 are the same. The ability to search for the optimal is the same. In other words, the change of P will not affect the optimization result of the functions. With the decrease of P the optimal value of function f_2 gradually decreases and the searching ability gradually increases. The basic trend of the 6 function curves is almost the same. The most obvious is function f_1 and the worst is function f_4. Compared with other convergent curves, f_3 has the largest fluctuation. It can be seen from the trend of all curves that the convergence speed is not increased or decreased regularly with the increase of P. However, it affects the optimal solution of the spatial distribution of different functions and has some relationship with the solution space. The maximum or minimum value of parameter P has little effect on function optimization. So whatever the value of P, you can get the optimal value for f_1, f_3, f_4, f_5, f_6 and have no effect on the function itself. For the function f_2, when P =0.6, the optimization of function is the best.

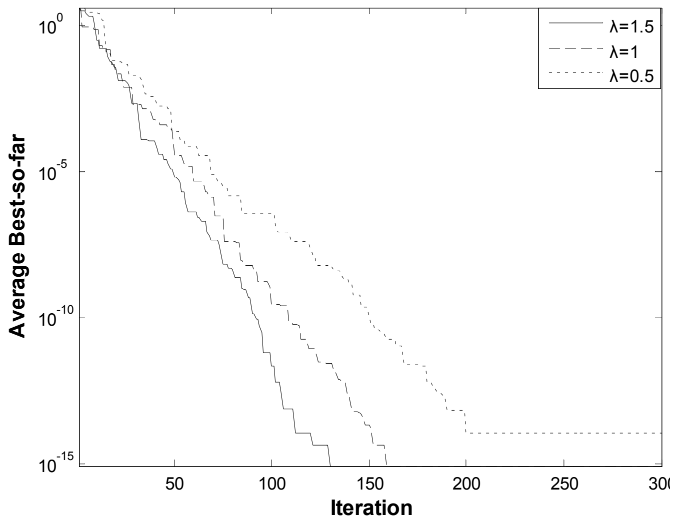
(2) Change of Single Variable

The initialization parameters of MFPA are set as: the population size n is 50, the number of iterations max_it is 500, $x_i = 2$, $P = 0.8$, $\gamma_1 = 1$, and $\gamma_2 = 3$. In order to reduce the influence of random disturbance, the independent operating for each test function is carried out 10 times. The optimal value and average values of MFPA in different Levy flight λ are shown in Tab. 3. The simulation results of the six test functions are shown in Fig. 2.

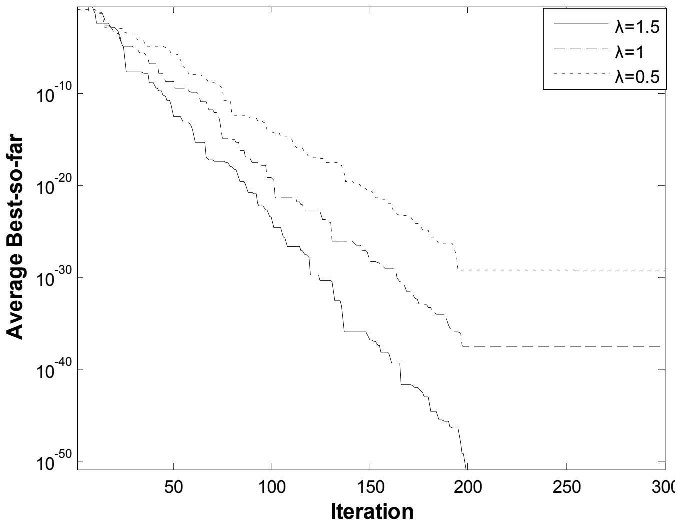
It can be seen from the convergence curves and the numerical results of six functions after 500 iterations run independently 10 times. With the decreasing of λ, the optimal solution and the precision of the optimization of function f_1 and f_2 gradually increase. However, the optimal solution and the precision of the optimization of function f_3 - f_6 remain the same. The biggest fluctuation is the function f_3. From the trend of the curves, the function f_2 is most obvious, and the function f_4 is worse.It can be seen from convergence curves that the rate of convergence rate becomes slower with the decrease of λ for unimodal functions. For multimodal functions, the change of λ has no regular change in its convergence speed but it affects the optimal solution of the spatial distribution of different functions and has some relationship with the solution space.

TABLE 3. PERFORMANCE COMPARISON OF MFPA UNDER DIFFERENT

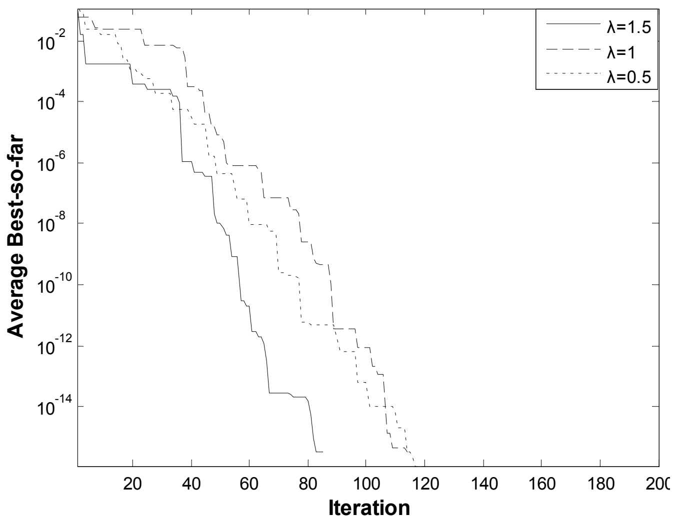
Function	Result	Simulation results of MFPA under different		
		1.5	1	0.5
f_1	optimum	8.8818e-016	8.8818e-016	2.2204e-014
	average	0.0189	0.0324	0.0319
	std	0.2548	0.3063	0.2907
f_2	optimum	6.6973e-045	1.3048e-038	5.6656e-023
	average	0.0081	0.0129	0.0194
	std	0.1054	0.1524	0.1317
f_3	optimum	2.5455e-005	2.5455e-005	2.5455e-005
	average	0.3125	0.0585	0.9943
	std	3.6581	1.0581	14.8181
f_4	optimum	-1.8013	-1.8013	-1.8013
	average	-1.8012	-1.8010	-1.8012
	std	0.0019	0.0040	8.2125e-004
f_5	optimum	-1	-1	-1
	average	-0.9987	-0.9989	-0.9986
	std	0.0088	0.0118	0.0118
f_6	optimum	0	0	0
	average	0.0073	0.0129	0.0293
	std	0.2171	0.1416	0.2484



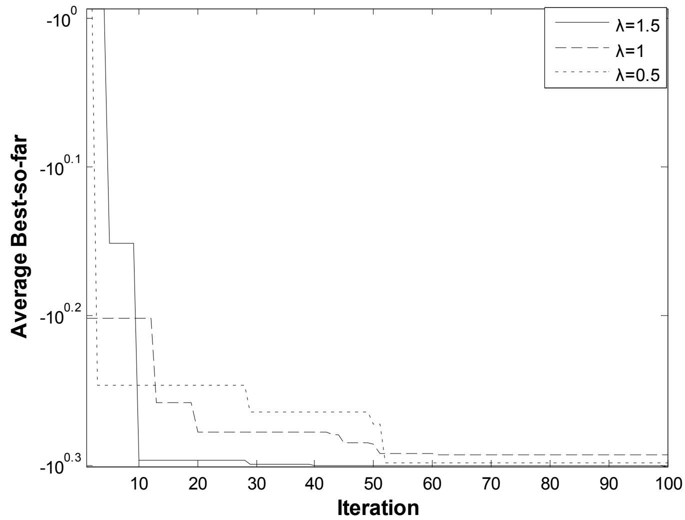
(a) Function f_1



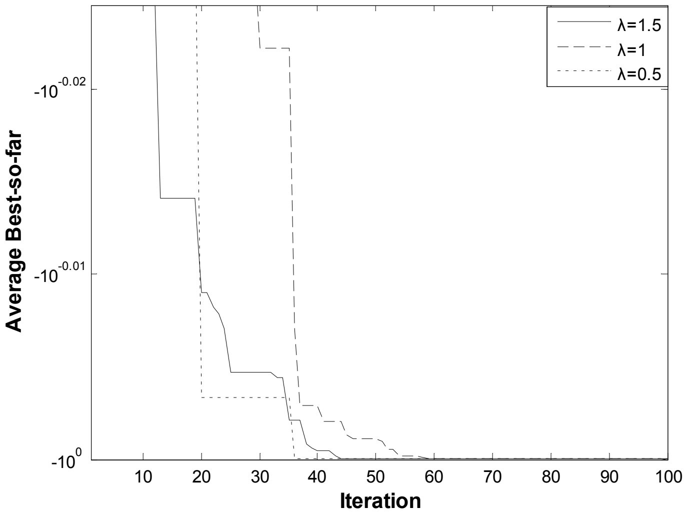
(b) Function f_2



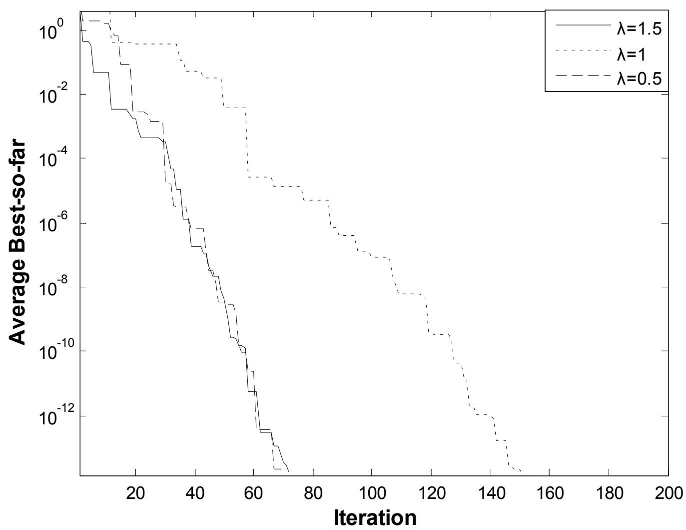
(c) Function f_3



(d) Function f_4



(e) Function f_5



(f) Function f_6

Fig. 2 Simulation results of MFPA under different.

IV. CONCLUSION

Based on the basic principle of the improved pollen pollination algorithm MFPA, the optimization performance is verified by carrying out the simulation experiments on six test functions. The value of parameter *P* has little influence on the convergence precision and the convergence speed of the functions. For unimodal functions, with the decreasing of parameters λ , the convergence precision gradually decreases and the convergence speed gradually slows down. For the multimodal functions, the convergence precision and the convergence speed of the function are not sensitive to the variation of parameters λ . Different functions have different requirements for parameters *P* and λ , so we need to get proper parameter setting according to different functions. In conclusion, the simulation results show that the convergence speed and optimization precision are closely related to the parameter setting.

REFERENCES

[1] G. G. Wang, L. Guo, A. H. Gandomi, G. S. Hao, and H. Wang, "Chaotic Krill Herd Algorithm," *Information Sciences*, vol. 274, no.274, pp. 17-34, 2014.

[2] N. Bell, and B. J. Oommen, "A Novel Abstraction for Swarm Intelligence: Particle Field Optimization," *Autonomous Agents and Multi-Agent Systems*, vol. 31, pp. 1-24, 2017.

[3] X. S. Yang, S. Deb, S. Fong, X. He， and Y. X. Zhao, "From Swarm Intelligence to Metaheuristics: Nature-Inspired Optimization Algorithms," *Computer*, vol. 49, no.9, pp. 52-59, 2016.

[4] L. Wang, H. Geng, P. Liu, K. Lu, Joanna Kolodziej, Rajiv Ranjan, and Albert Y. Zomaya, "Particle Swarm Optimization Based Dictionary Learning for Remote Sensing Big Data," *Knowledge-Based Systems*, vol. 79, no.C, pp. 43-50, 2015.

[5] Y. Ghanou, and G. Bencheikh, "Architecture Optimization and Training for the Multilayer Perceptron using Ant System," *IAENG International Journal of Computer Science*, vol. 43, no.1, pp. 20-26, 2016.

[6] A. P. Song, M. B. Li, X. H. Ding, W. Cao, and K. Pu, "Community Detection Using Discrete Bat Algorithm," *IAENG International Journal of Computer Science*, vol. 43, no.1, pp37-43, 2016.

[7] F. Huang, X. Li, S. Zhang, and J. Zhang, "Harmonious Genetic Clustering," *IEEE Transactions on Cybernetics,*vol. 48, no.1, pp. 199, 2017.

[8] D. Wu, S. Xu, and F. Kong, "Convergence Analysis and Improvement of the Chicken Swarm Optimization Algorithm," *IEEE Access*, vol. 4, pp. 9400-9412, 2017.

[9] L. Olatomiwa, S. Mekhilef, S. Shamshirband, K. Mohammadi, D. Petković, and C. Sudheer, "A Support Vector Machine–firefly Aalgorithm-based Model for Global Solar Radiation Prediction," *Solar Energy*, vol. 115, pp. 632-644, 2015.

[10] X. S. Yang, "Flower Pollination Algorithm for Global Optimization," *Proc of International Conference on Unconventional Computing and Natural Computation,* vol. 7445, pp. 240-249, 2012.

[11] A. Y. Abdelaziz, E. S. Ali, and S. M. A. Elazim, "Implementation of Flower Pollination Algorithm for Solving Economic Load Dispatch and Combined Economic Emission Dispatch Problems in Power Systems," *Energy*, vol. 101, pp. 506-518, 2016.

[12] S. J. Huang, P. H. Gu, W. F. Su, and X. Z. Liu, "Application of Flower Pollination Algorithm for Placement of Distribution Transformers in a Low-voltage Grid," *IEEE International Conference on Industrial Technology,* pp. 1280-1285, 2015.

[13] L. Xiao, W. Shao, M. Yu, J. Ma, and C. Jin, "Research and Application of a Combined Model Based on Multi-objective Optimization for Electrical Load Forecasting," *Energy*, vol. 119, pp. 1057-1074, 2017.

[14] A. Mishra, and S. Deb, Assembly Sequence Optimization Using a Flower Pollination Algorithm-based Approach," *Journal of Intelligent Manufacturing*, pp. 1-12, 2016.

[15] H. M. Dubey, M. Pandit, and B. K. Panigrahi, "Hybrid Flower Pollination Algorithm with Time-varying Fuzzy Selection Mechanism for Wind Integrated Multi-objective Dynamic Economic Dispatch," *Renewable Energy*, vol. 83, pp. 188-202, 2015.

[16] A. Y. Abdelaziz, E. S. Ali, and S. M. A. Elazim, "Combined Economic and Emission Dispatch Solution Using Flower Pollination Algorithm," *International Journal of Electrical Power & Energy Systems*, vol. 80, pp. 264-274, 2016.

[17] E. Nabil, "A Modified Flower Pollination Algorithm for Global Optimization," *Expert Systems with Applications*, vol. 57, pp. 192-203, 2016.

Xiao-Xu Ma is received her B. Sc. degree from University of Science and Technology Liaoning in 2012. She is currently a master student in School of Electronic and Information Engineering, University of Science and Technology Liaoning, China. Her main research interest is modeling methods of complex process and intelligent optimization algorithms.

Jie-sheng Wang received his B. Sc. And M. Sc. degrees in control science from University of Science and Technology Liaoning, China in 1999 and 2002, respectively, and his Ph. D. degree in control science from Dalian University of Technology, China in 2006. He is currently a professor and Master's Supervisor in School of Electronic and Information Engineering, University of Science and Technology Liaoning. His main research interest is modeling of complex industry process, intelligent control and Computer integrated manufacturing.