A Multi-Swarm Bat Algorithm for Global Optimization

Gai-Ge Wang
[1]School of Computer Science and Technology
Jiangsu Normal University
Xuzhou, China
[2]Institute of Algorithm and Big Data Analysis
Northeast Normal University
Changchun, China
[3]School of Computer Science and Information Technology
Northeast Normal University
Changchun, China
gaigewang@163.com

Bao Chang
Department of Information Science
Jiangsu Normal University Kewen College
Xuzhou, China

Zhaojun Zhang
School of Electrical Engineering and Automation
Jiangsu Normal University,
Xuzhou, China

***Abstract*—By simulating the echolocation behavior of bats in nature, bat algorithm (BA) is proposed for global optimization that is a recently developed nature-inspired algorithm. Since then, it has been widely used in various fields. Bat algorithm balance the global search and local search by adjusting loudness and pulse rate. However, there is so many loudness and pulse rate combinations that it is hard to choose the most proper one for different problems. In this paper, a multi-swarm algorithm, called multi-swarm bat algorithm (MBA), is proposed for global search problem. In MBA method, immigration operator is used to exchange information between different swarms with different parameter settings. Thus, this configuration can make a good trade-off between global and local search. In addition, the best individuals of every swarm is put into the elite swarm through selection operator. The bat individuals in elite swarm pass over next generation without performing any operators, and this can ensure these best solutions cannot be damaged during optimization process. In order to evaluate the efficiency of MBA method, MBA has been benchmarked by sixteen standard test functions by comparing with basic BA. The results show that the MBA method is able to search more satisfactory function values on most benchmark problems than BA.**

Keywords—Bat algorithm; multi-swarm; immigration operator; selection operator; benchmark problems

I. INTRODUCTION

In real-world life, numerous problems can be modeled into global optimization problems, and then they be solved by optimization methods. However, with the increment of complex of the problem, the traditional optimization methods are hardly to find the satisfactory solutions. Inspired by phenomenon in nature, various nature-inspired metaheuristic algorithms [1] have been put forward and successfully solved many real engineering application problems, such as parameter estimation [2], feature selection [3], nonlinear system modeling [4], reliability [5], scheduling [6], neural network training [7], and knapsack problem [8]. Among these methods, swarm-based algorithms, also called swarm intelligence (SI) method, is one of the most representative paradigms.

Inspired by the collective behavior of animals, various swarm intelligence methods have been proposed and widely used in various applications. Two of famous SI methods are particle swarm optimization (PSO) [9-12] and ant colony optimization (ACO) [13]. They originated from the social flocking behavior of bird when searching for the food [9] and the pheromone of ants [13-15], respectively. Since then, more state-of-the-art SI algorithms have been developed, such as artificial bee colony (ABC) [16], firefly algorithm (FA) [17-20], cuckoo search (CS) [21-24], bat algorithm (BA) [25-30], grey wolf optimizer (GSO) [31], and krill herd (KH) [32-36]. They are inspired by the swarm behavior of honey bees, fireflies, cuckoos, bats, grey wolves, and krill, respectively.

Developed by Yang in 2010, BA method [25, 27] is a meta-heuristic search algorithm by simulating the echolocation behavior of when searching for food. BA balances the global search and local search by adjusting loudness and pulse rate. However, sometimes, there is so many loudness and pulse rate combinations that it is hard to select the most proper one to solve certain problem.

Recently, a multi-swarm or multi-population technology [37, 38] has been incorporated into a series of metaheuristics, such as ABC [39], genetic algorithm (GA) [40], and PSO [41, 42]. Because different swarms/populations can have their own parameter settings and they can simultaneously implement search, they have significantly improve the performance of the original metaheuristics.

In this paper, a multi-swarm algorithm, called multi-swarm bat algorithm (MBA), is proposed for global optimization. In MBA method, different swarms with their own parameter settings simultaneously search the given domain and they can exchange information by immigration operator. By this way, this configuration can make a good balance between global and local search. And then, the best individuals of every swarm is collected to make up the elite swarm through selection operator.

This work was supported by Research Fund for the Doctoral Program of Jiangsu Normal University (No. 9213614102) and National Natural Science Foundation of China (No. 61305149).

The bat individuals in elite swarm will pass over next generation without performing any operators. This can ensure the bat swarms always include the best solutions, and make the swarm proceed towards the global optima. In order to evaluate the efficiency of MBA method, MBA has been benchmarked by sixteen standard test functions by comparing with basic BA. The results show that the MBA method is able to search more satisfactory function values on most benchmark problems than BA.

Section 2 reviews the idealization of bat algorithm, and Section 3 describes the mainframe of MBA. With the aim of the showing the performance of MBA method, several simulation results comparing MBA with BA method for general benchmark functions are presented in Section 4. Section 5 presents some concluding remarks and suggestions for further work.

II. BAT ALGORITHM

The bat algorithm [1] is a new swarm intelligence optimization method for solving optimization problems, and it is inspired by social behavior of bats and the phenomenon of echolocation to sense distance. In BA, each bat is defined by its position x_i^t, velocity v_i^t, frequency f_i, loudness A_i^t and the emission pulse rate r_i^t in a d-dimensional search space. For each bat individual, frequency, velocity and position are updated by

$$f_i = f_{\min} + (f_{\max} - f_{\min})\beta \tag{1}$$

$$\mathbf{v}_i^t = \mathbf{v}_i^{t-1} + (\mathbf{x}_i^{t-1} - \mathbf{x}_*) f_i \tag{2}$$

$$\mathbf{x}_i^t = \mathbf{x}_i^{t-1} + \mathbf{v}_i^t \tag{3}$$

where $\beta \in [0, 1]$ is a random vector drawn from a uniform distribution. Here $\mathbf{x}_*$ is the current global best solution. Generally speaking, , the frequency f is assigned according to the domain size of the problem of interest.

For the local search part, once a solution is selected among the current best solutions, a new solution for each bat is generated locally using random walk

$$\mathbf{x}_{new} = \mathbf{x}_{old} + \varepsilon A^t \tag{4}$$

where $\boldsymbol{\varepsilon} \in [-1, 1]$ is a scaling factor which is a random number, while $A_t = \langle A_i^t \rangle$ is the average loudness of all the bats at current generation.

Furthermore, the loudness A_i and the rate r_i of pulse emission update accordingly as the iterations proceed show as Eq. (5).

$$A_i^{t+1} = \alpha A_i^t, r_i^{t+1} = r_i^0[1 - \exp(-\gamma t)] \tag{5}$$

where α and γ are constants.

BA balances the global search and local search by adjusting loudness A and pulse rate r. However, sometimes, there is so many loudness and pulse rate combinations that it is hard to select the most proper one to solve certain problem.

Based on the above description, the basic steps of BA can be shown in Algorithm 1.

Algorithm 1 Bat Algorithm

Begin

Step 1: Initialization. Set the generation counter t=1; randomly initialize the population; define loudness A_i, pulse frequency Q_i *and* the initial velocities v_i ($i = 1,2,\cdots,NP$); set pulse rate r_i.

Step 2: While t<*MaxGeneration* **do**

Generate new solutions by adjusting frequency, and updating velocities and locations

if ($rand>r_i$) **then**

Select a solution among the best solutions

Generate a local solution around the selected best solution

end if

Generate a new solution by flying randomly

if ($rand<A_i$ & $f(x_i)<f(x_*)$) **then**

Accept the new solutions

Increase r_i and reduce A_i

end if

Rank the bats and find the current best x_*

$t = t+1;$

Step 3: end while

End

III. MULTI-SWARM BAT ALGORITHM

In view of the above problems in bat algorithm, a kind of new multiple swarm cooperation algorithm, called multi-swarm bat algorithm (MBA), is proposed and can be used to replace the conventional standard bat algorithm.

Based on BA, the following concepts are mainly introduced into the MBA.

(1) Frequency, velocity and location updating operators are only performed on a single swarm in BA, while frequency, velocity and location updating operators can be simultaneously implemented on many swarms in MBA. Different swarms have different control parameter settings, thus they can achieve different search purposes.

(2) Different swarms exchange information through immigration operator to achieve multi-swarm co-evolution, and the final optimal solution is the aggregated results of multi-swarm co-evolution.

(3) The best individual in each population is saved on each generation through selection operator, which can also be a criterion for judging convergence.

Based on the three above rules, the frame of MBA can be given in Fig. 1. In Fig. 1, the regular BA is used in swarm 1~N that performs frequency, velocity and location updating operators.

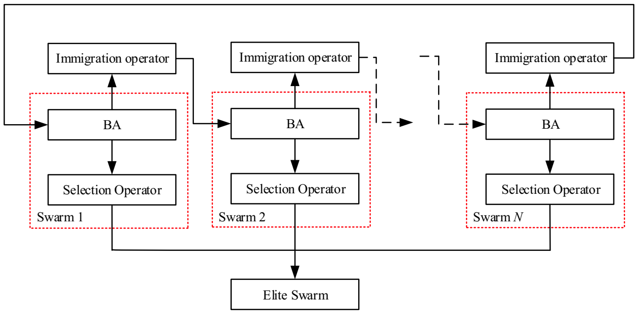

Fig. 1. Flowchart of multi-swarm bat algorithm.

Various swarms have different control parameter settings. The value of loudness A, and pulse rate r can make a good trade-off between global search and local search. In BA, the loudness and pulse updating operators are the main operators when generating new individuals, and they determine the search ability of bat algorithm. However, the values of A and r have so numerous choices that it is hard to select the most proper parameter settings for different problems, and the optimization results have great difference for different A and r. MBA can evolve simultaneously on multiple swarms with different control parameters. This framework can overcome BA's weaknesses and make a good balance between global search and local search.

Various swarms are relatively independent, and they can exchange information through immigration operator. By immigration operator, the best individuals in various swarms can be transferred to other swarms periodically (every certain evolution generations) in order to realize information exchange between different swarms. This can be done by replacing the worst individual in target swarm with the best individual in source swarm. Immigration operator is an indispensable and important operator in MBA. If there is no immigration operator, different swarms lose contact between them. MBA will be equivalent to multiple repeated implementations of BA with various control parameters, thus losing the features of MBA.

The elite swarm has significant difference from other swarms. The optimal individual selected by selection operator is put into the elite swarm each generation. The elite swarm doesn't perform frequency, velocity and location updating operators, and this can ensure the best individual in each population is not damaged during the optimization process. At the same time, the elite swarm can be also used to judge the termination of the algorithm.

IV. Simulation Results

The performance of MBA was compared with the basic BA method on sixteen benchmark problems (see Table 1). More detailed descriptions of all the benchmarks can be referred as [43-45].

TABLE I. BENCHMARK FUNCTIONS.

No.	*Name*	*Definition*
F01	Ackley	$f(\vec{x}) = 20 + e - 20\cdot e^{-0.2\cdot\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}} - e^{\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)}$
F02	Alpine	$f(\vec{x}) = \sum_{i=1}^{n} \lvert x_i \sin(x_i) + 0.1 x_i \rvert$
F03	Brown	$f(\vec{x}) = \sum_{i=1}^{n-1}\left[\left(x_i^2\right)^{x_{i+1}^2+1} + \left(x_{i+1}^2\right)^{x_i^2+1}\right]$
F04	Dixon & Price	$f(\vec{x}) = (x_1 - 1)^2 + \sum_{i=2}^{n} i\left(2x_i^2 - x_{i-1}\right)^2$
F05	Fletcher-Powell	$f(\vec{x}) = \sum_{i=1}^{n}(A_i - B_i)^2,\ A_i = \sum_{j=1}^{n}\left(a_{ij}\sin\alpha_j + b_{ij}\cos\alpha_j\right)$ $B_i = \sum_{j=1}^{n}\left(a_{ij}\sin x_j + b_{ij}\cos x_j\right)$
F06	Griewank	$f(\vec{x}) = \sum_{i=1}^{n}\frac{x_i^2}{4000} - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$
F07	Holzman 2 function	$f(\vec{x}) = \sum_{i=1}^{n} i x_i^4$
F08	Levy 8	$f(\vec{x}) = \sin^2(\pi y_1) + \sum_{i=1}^{n-1}\left[(y_i - 1)^2\left(1 + 10\sin^2(\pi y_i + 1)\right)\right]$ $+(y_n - 1)^2\left(1 + 10\sin^2(2\pi y_n)\right),\ y_i = 1 + (x_i - 1)/4$
F09	Pathological function	$f(\vec{x}) = \sum_{i=1}^{n}\left(0.5 + \frac{\sin^2\sqrt{100x_i^2 + x_{i+1}^2} - 0.5}{1 + 0.001\left(x_i^2 - 2x_i x_{i+1} + x_{i+1}^2\right)^2}\right)$
F10	Penalty #1	$f(\vec{x}) = \frac{\pi}{30}\left\{10\sin^2(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2\left[1 + 10\sin^2(\pi y_{i+1})\right]\right.$ $\left.+(y_n - 1)^2\right\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4),\ y_i = 1 + 0.25(x_i + 1)$
F11	Penalty #2	$f(\vec{x}) = 0.1\left\{\sin^2(3\pi x_1) + \sum_{i=1}^{n-1}(x_i - 1)^2 \square \left[1 + \sin^2(3\pi x_{i+1})\right]\right.$ $\left.+(x_n - 1)^2\left[1 + \sin^2(2\pi x_n)\right]\right\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$
F12	Perm #1	$f(\vec{x}) = \sum_{i=1}^{n/4}(x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2$ $+(x_{4i-2} - x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^4$
F13	Perm #2	$f(\vec{x}) = \sum_{i=1}^{n}(i \cdot x_i^4 + U(0,1))$
F14	Powell	$f(\vec{x}) = \sum_{i=1}^{n/4}(x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2$ $+(x_{4i-2} - x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^4$
F15	Quartic *with noise*	$f(\vec{x}) = \sum_{i=1}^{n}(i \cdot x_i^4 + U(0,1))$
F16	Rastrigin	$f(\vec{x}) = 10\cdot n + \sum_{i=1}^{n}(x_i^2 - 10\cdot\cos(2\pi x_i))$

In order to make a fair comparison, the population size and maximum generation for BA and MBA are set to 30 and 50, respectively. Noted that, for MBA, we set five swarms (*i.e.*, N=5), and thus each swarm has six bat individuals. For other parameters used in BA, they are set as follows: loudness A

=0.25, pulse rate r =0.5, scaling factor ε=0.1, the minimum frequency $f_{\min}$=0.7 and the maximum frequency $f_{\max}$=0.9. For MBA, the values of A and r are dynamically adjusted in the range [0.7, 1.0] and [0.5, 0.8], respectively. For other parameters used in MBA, they are the same as BA.

The BA and MBA are coded by MATLAB, and they are implemented under the same conditions as [46, 47].

Because BA and MBA are stochastic algorithms, 100 independent runs are done in order to get representative results. In this section, the best function values obtained by BA and MBA are highlighted by bold font (see Table 2).

From Table 2, for the best solutions, though BA and MBA have the same final optimal results on F13, MBA is significantly superior to BA on the others. For the average value, MBA has the absolute advantage over BA method on all the benchmark problems. For the worst performance, it is more complex than the best and average solutions. After 50-generation optimization, BA is well capable of finding the same worst values with MBA on F01, F07, F09, F13, F15 and F16. MBA is able to find more satisfactory results on F02-F05, and F10-F12 than BA, while BA can search more satisfactory on F06, F08 and F14 than MBA. Based on the above analyses, MBA has the better performance than basic BA on most benchmarks. For standard deviation (Std), MBA has much smaller Std on fourteen benchmarks (except F01 and F09). Smaller Std indicates that MBA implements more stably than BA.

TABLE II. FUNCTION VALUES OBTAINED BY BA AND MBA ON SIXTEEN BENCHMARKS.

		BA	MBA			BA	MBA
F01	Best	17.04	**11.91**	F09	Best	1.04	**0.54**
	Mean	19.43	**15.79**		Mean	3.69	**3.09**
	Worst	**19.95**	**19.95**		Worst	**1.1E6**	**1.1E6**
	Std	**0.57**	1.44		Std	**0.97**	1.05
F02	Best	25.67	**6.07**	F10	Best	7.5E6	**3.3E4**
	Mean	36.49	**14.31**		Mean	1.5E8	**1.0E7**
	Worst	31.88	**19.95**		Worst	5.2E7	**1.9E7**
	Std	4.90	**3.35**		Std	9.3E7	**2.1E7**
F03	Best	68.46	**21.30**	F11	Best	3.5E7	**1.4E6**
	Mean	2.3E6	**176.26**		Mean	3.0E8	**2.7E7**
	Worst	641.74	**324.29**		Worst	2.9E8	**1.2E7**
	Std	2.2E7	**463.44**		Std	1.8E8	**3.2E7**
F04	Best	5.2E4	**3.0E3**	F12	Best	1.6E39	**1.2E39**
	Mean	3.7E5	**4.1E4**		Mean	8.5E50	**1.4E50**
	Worst	3.4E5	**4.2E4**		Worst	4.5E50	**5.5E41**
	Std	1.8E5	**3.1E4**		Std	1.8E51	**1.4E51**
F05	Best	4.5E3	**1.8E5**	F13	Best	**1.00**	**1.00**
	Mean	1.0E6	**4.6E5**		Mean	7.1E3	**2.21**
	Worst	1.1E6	**4.0E5**		Worst	**3.4E5**	**3.4E5**
	Std	3.1E5	**1.8E5**		Std	6.9E4	**12.09**
F06	Best	120.66	**30.53**	F14	Best	843.70	**77.47**
	Mean	257.51	**91.67**		Mean	6.1E3	**847.42**
	Worst	**317.48**	324.29		Worst	**4.0E3**	3.4E5
	Std	77.14	**36.06**		Std	3.1E3	**655.31**
F07	Best	1.5E4	**1.0E3**	F15	Best	2.03	**0.24**
	Mean	1.0E5	**1.2E4**		Mean	26.52	**3.46**
	Worst	**3.4E5**	**3.4E5**		Worst	**85.31**	**85.31**
	Std	5.4E4	**1.2E4**		Std	14.21	**3.51**
F08	Best	40.13	**6.86**	F16	Best	179.98	**51.11**
	Mean	93.88	**25.02**		Mean	252.51	**93.20**
	Worst	**85.31**	4.2E4		Worst	**303.88**	**303.88**
	Std	29.20	**9.51**		Std	26.33	**20.69**

In addition, to future show the convergent process of MBA method, limited by the length of paper, convergence trajectories of BA and MBA on some functions are also illustrated in the current work (see Figs. 2-3).

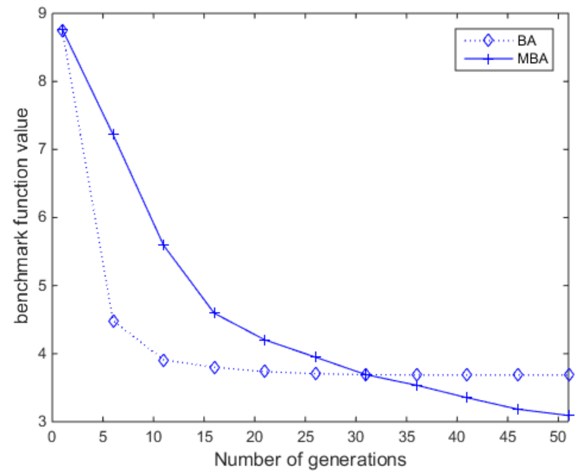

Fig. 2. Convergent history for the F09 Pathological function with 50 generations.

From Fig. 2, clearly, both BA and MBA start optimization process from the same fitness value. BA has a more fast convergent speed than MBA within generation 30. At generation 30, MBA and BA have the similar fitness value. Subsequently, BA may trap into a local optimal value, while MBA continue to find better fitness value till the end of the optimization process. This can show that MBA can escape from local minimum through dynamically adjusting loudness A and pulse rate r.

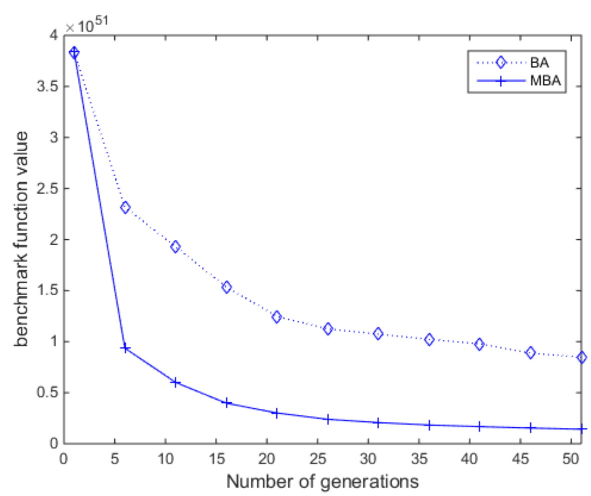

Fig. 3. Convergent history for the F12 Perm #1 function with 50 generations.

For this test problem, the optimization process is more simple than before. MBA has the better fitness value than BA during all the optimization process. This means, the framework used in MBA, is more efficient than BA, and it is more suitable for optimization problem.

V. DISCUSSION AND CONCLUSION

By incorporating the multi-swarm technology into basic BA method, a new kind of heuristic search method, called multi-swarm bat algorithm (MBA), is proposed for global optimization. In MBA method, different swarms with their own parameter settings simultaneously search the given domain and they can exchange information by immigration operator. This configuration can make a good balance between global and local search by dynamically adjusting loudness and pulse rate. After all the bat individuals in every swarm are updated, the best individuals in each swarm is collected to make up the elite swarm through selection operator. In order to prevent the bat individuals in elite swarm being damaged, they perform no operators and directly pass over next generation. Through an array of experiments, the results show that MBA performs better than basic BA.

In addition, MBA algorithm is simple, and has no complicated calculation and operators. This makes MBA algorithm implement easily and fast.

Despite various advantages of the MBA method, the following points should be clarified and focused on in the future research.

Firstly, computational requirements are of vital importance for a metaheuristic method. How to make the MBA method implement much faster is worthy of further study, too.

Secondly, limited by the paper, MBA has been only benchmarked by sixteen functions. In future, more benchmark problems, especially real-world applications, should be used to verify the MBA method.

Thirdly, the number of swarm is also an interesting spot that is worthy of further study.

At last, some theoretical analysis should be done in order to prove convergence of the method. This analysis can also significantly improve the reliability of the method when solving real-world application problem.

REFERENCES

[1] X. S. Yang, *Nature-inspired metaheuristic algorithms*, 2nd ed., Luniver Press, Frome, 2010.

[2] X. Li, and M. Yin, “Parameter estimation for chaotic systems by hybrid differential evolution algorithm and artificial bee colony algorithm,” *Nonlinear Dynamics,* vol. 77, no. 1-2, pp. 61-71, 2014.

[3] X. Li, and M. Yin, “Multiobjective binary biogeography based optimization for feature selection using gene expression data,” *IEEE Transactions on NanoBioscience,* vol. 12, no. 4, pp. 343-353, 2013.

[4] A. H. Gandomi, and A. H. Alavi, “Multi-stage genetic programming: A new strategy to nonlinear system modeling,” *Information Sciences,* vol. 181, no. 23, pp. 5227-5239, 2011.

[5] D. Zou, L. Gao, J. Wu, S. Li, and Y. Li, “A novel global harmony search algorithm for reliability problems,” *Computers & Industrial Engineering,* vol. 58, no. 2, pp. 307-316, 2010.

[6] X. Li, and M. Yin, “An opposition-based differential evolution algorithm for permutation flow shop scheduling based on diversity measure,” *Advances in Engineering Software,* vol. 55, pp. 10-31, 2013.

[7] S. Mirjalili, S. M. Mirjalili, and A. Lewis, “Let a biogeography-based optimizer train your Multi-Layer Perceptron,” *Information Sciences,* vol. 269, pp. 188-209, 2014.

[8] D. Zou, L. Gao, S. Li, and J. Wu, “Solving 0-1 knapsack problem by a novel global harmony search algorithm,” *Applied Soft Computing,* vol. 11, no. 2, pp. 1556-1564, 2011.

[9] J. Kennedy, and R. Eberhart, “Particle swarm optimization,” in Proceeding of the IEEE International Conference on Neural Networks, Perth, Australia, 1995, pp. 1942-1948.

[10] S. Mirjalili, G.-G. Wang, and L. d. S. Coelho, “Binary optimization using hybrid particle swarm optimization and gravitational search algorithm,” *Neural Computing and Applications,* vol. 25, no. 6, pp. 1423-1435, 2014.

[11] G.-G. Wang, A. H. Gandomi, X.-S. Yang, and A. H. Alavi, “A novel improved accelerated particle swarm optimization algorithm for global numerical optimization,” *Engineering Computations,* vol. 31, no. 7, pp. 1198-1220, 2014.

[12] G.-G. Wang, A. H. Gandomi, and A. H. Alavi, “A chaotic particle-swarm krill herd algorithm for global numerical optimization,” *Kybernetes,* vol. 42, no. 6, pp. 962-978, 2013.

[13] M. Dorigo, V. Maniezzo, and A. Colorni, “Ant system: optimization by a colony of cooperating agents,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics,* vol. 26, no. 1, pp. 29-41, 1996.

[14] Z. Zhang, N. Zhang, and Z. Feng, “Multi-satellite control resource scheduling based on ant colony optimization,” *Expert Systems with Applications,* vol. 41, no. 6, pp. 2816-2823, 2014.

[15] Z. Zhang, and Z. Feng, “Two-stage updating pheromone for invariant ant colony optimization algorithm,” *Expert Systems with Applications,* vol. 39, no. 1, pp. 706-712, 2012.

[16] D. Karaboga, and B. Basturk, “A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm,” *Journal of Global Optimization,* vol. 39, no. 3, pp. 459-471, 2007.

[17] X. S. Yang, “Firefly algorithm, stochastic test functions and design optimisation,” *International Journal of Bio-Inspired Computation,* vol. 2, no. 2, pp. 78-84, 2010.

[18] G.-G. Wang, L. Guo, H. Duan, and H. Wang, “A new improved firefly algorithm for global numerical optimization,” *Journal of Computational and Theoretical Nanoscience,* vol. 11, no. 2, pp. 477-485, 2014.

[19] G. Wang, L. Guo, H. Duan, L. Liu, and H. Wang, “A modified firefly algorithm for UCAV path planning,” *International Journal of Hybrid Information Technology,* vol. 5, no. 3, pp. 123-144, 2012.

[20] I. F. Jr., S. Fong, J. Brest, and I. Fister, "Towards the self-adaptation in the bat algorithm." Proceedings of the 13th IASTED International Conference on Artificial Intelligence and Applications, 2014. doi:10.2316/P.2014.816-011

[21] X. S. Yang, and S. Deb, "Cuckoo search via Lévy flights." World Congress on Nature & Biologically Inspired Computing (NABIC 2009), IEEE Publications, Coimbatore, vol. 4, pp. 210-214, 2009. arXiv:1003.1594v1.

[22] X. Li, and M. Yin, “Modified cuckoo search algorithm with self adaptive parameter method,” *Information Sciences,* vol. 298, pp. 80-97, 2015.

[23] G.-G. Wang, A. H. Gandomi, X. Zhao, and H. E. Chu, “Hybridizing harmony search algorithm with cuckoo search for global numerical optimization,” *Soft Computing*, in press..

[24] X. Li, J. Wang, and M. Yin, “Enhancing the performance of cuckoo search algorithm using orthogonal learning method,” *Neural Computing and Applications,* vol. 24, no. 6, pp. 1233-1247, 2013.

[25] X. S. Yang, "A new metaheuristic bat-inspired algorithm," *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, Studies in Computational Intelligence J. R. González, D. A. Pelta, C. Cruz, G. Terrazas and N. Krasnogor, eds., pp. 65-74, Heidelberg: Springer, 2010.

[26] S. Mirjalili, S. M. Mirjalili, and X.-S. Yang, “Binary bat algorithm,” *Neural Computing and Applications,* vol. 25, no. 3-4, pp. 663-681, 2013.

[27] A. H. Gandomi, X.-S. Yang, A. H. Alavi, and S. Talatahari, “Bat algorithm for constrained optimization tasks,” *Neural Computing & Applications,* vol. 22, no. 6, pp. 1239-1255, 2013.

[28] X. S. Yang, and A. H. Gandomi, “Bat algorithm: a novel approach for global engineering optimization,” *Engineering Computations,* vol. 29, no. 5, pp. 464-483, 2012.

[29] I. Fister Jr, S. Fong, J. Brest, and I. Fister, “A novel hybrid self-adaptive bat algorithm,” *The Scientific World Journal,* vol. 2014, pp. 1-12, 2014.

[30] I. J. Fister, I. Fister, Y. Xin-She, S. Fong, and Z. Yan, "Bat algorithm: Recent advances." 2014 IEEE 15th International Symposium on Computational Intelligence and Informatics (CINTI), pp. 163-167.

[31] S. Mirjalili, S. M. Mirjalili, and A. Lewis, “Grey wolf optimizer,” *Advances in Engineering Software,* vol. 69, pp. 46-61, 2014.

[32] A. H. Gandomi, and A. H. Alavi, “Krill herd: A new bio-inspired optimization algorithm,” *Communications in Nonlinear Science and Numerical Simulation,* vol. 17, no. 12, pp. 4831-4845, 2012.

[33] G.-G. Wang, A. H. Gandomi, and A. H. Alavi, “Stud krill herd

algorithm," *Neurocomputing,* vol. 128, pp. 363-370, 2014.
[34] G.-G. Wang, A. H. Gandomi, and A. H. Alavi, "An effective krill herd algorithm with migration operator in biogeography-based optimization," *Applied Mathematical Modelling,* vol. 38, no. 9-10, pp. 2454-2462, 2014.
[35] G.-G. Wang, A. H. Gandomi, A. H. Alavi, and G.-S. Hao, "Hybrid krill herd algorithm with differential evolution for global numerical optimization," *Neural Computing and Applications,* vol. 25, no. 2, pp. 297-308, 2014.
[36] L. Guo, G.-G. Wang, A. H. Gandomi, A. H. Alavi, and H. Duan, "A new improved krill herd algorithm for global numerical optimization," *Neurocomputing,* vol. 138, pp. 392-402, 2014.
[37] C. Li, S. Yang, and M. Yang, "An adaptive multi-swarm optimizer for dynamic optimization problems," *Evolutionary Computation,* vol. 22, no. 4, pp. 559-594, 2014.
[38] C. Li, T. T. Nguyen, M. Yang, S. Yang, and S. Zeng, "Multi-population methods in unconstrained continuous dynamic environments: The challenges," *Information Sciences,* vol. 296, pp. 95-118, 2015.
[39] S. Biswas, S. Das, S. Debchoudhury, and S. Kundu, "Co-evolving bee colonies by forager migration: A multi-swarm based Artificial Bee Colony algorithm for global search space," *Applied Mathematics and Computation,* vol. 232, pp. 216-234, 2014.
[40] H. Pourvaziri, and B. Naderi, "A hybrid multi-population genetic algorithm for the dynamic facility layout problem," *Applied Soft Computing,* vol. 24, pp. 457-469, 2014.
[41] D. Yazdani, B. Nasiri, A. Sepas-Moghaddam, and M. R. Meybodi, "A novel multi-swarm algorithm for optimization in dynamic environments based on particle swarm optimization," *Applied Soft Computing,* vol. 13, no. 4, pp. 2144-2158, 2013.
[42] X. Zhao, Z. Liu, and X. Yang, "A multi-swarm cooperative multistage perturbation guiding particle swarm optimizer," *Applied Soft Computing,* vol. 22, pp. 77-93, 2014.
[43] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Transactions on Evolutionary Computation,* vol. 3, no. 2, pp. 82-102, 1999.
[44] D. Simon, "Biogeography-based optimization," *IEEE Transactions on Evolutionary Computation,* vol. 12, no. 6, pp. 702-713, 2008.
[45] X.-S. Yang, Z. Cui, R. Xiao, A. H. Gandomi, and M. Karamanoglu, *Swarm Intelligence and Bio-Inspired Computation*, Elsevier, Waltham, MA, 2013.
[46] G.-G. Wang, L. Guo, A. H. Gandomi, G.-S. Hao, and H. Wang, "Chaotic krill herd algorithm," *Information Sciences,* vol. 274, pp. 17-34, 2014.
[47] G. Wang, L. Guo, H. Wang, H. Duan, L. Liu, and J. Li, "Incorporating mutation scheme into krill herd algorithm for global numerical optimization," *Neural Computing and Applications,* vol. 24, no. 3-4, pp. 853-871, 2014.