A new artificial bee colony algorithm based on modified search strategy

Kai Li, Minyang Xu, Tao Zeng, Tingyu Ye and Luqi Zhang

School of Information Engineering,
Nanchang Institute of Technology,
Nanchang, 330099, China
Email: Aha15380251715@gmail.com
Email: ncxmy2019@163.com
Email: nitztao@163.com
Email: yty01122@163.com
Email: nczlq2020@126.com

Wenjun Wang

School of Business Administration,
Nanchang Institute of Technology,
Nanchang, 330099, China
Email: wangwenjun881@126.com

Hui Wang*

School of Information Engineering,
Nanchang Institute of Technology,
Nanchang, 330099, China
Email: huiwang@whu.edu.cn
*Corresponding author

Abstract: Artificial bee colony (ABC) is an efficient global optimisation algorithm. It has attracted the attention of many researchers because of its simple concept and strong exploration. However, it exhibits weak exploitation capability. To improve this case, a novel ABC with modified search strategy (namely MSABC) is proposed in this work. In MSABC, some modified elite solutions are preserved and used to guide the search. In addition, MSABC uses the modified elite solutions to generate offspring to replace the probability selection in the onlooker bee phase. To evaluate the capability of MSABC, 22 classical problems are tested. Results demonstrate MSABC achieves superior performance than five other ABC variants.

Keywords: ABC; artificial bee colony; elite solution; search strategy; probability selection; modified search strategy; Euclidean distance; weak exploitation; strong exploration; global optimisation; function optimisation; optimisation.

Reference to this paper should be made as follows: Li, K., Xu, M., Zeng, T., Ye, T., Zhang, L., Wang, W.. and Wang, H. (2022) 'A new artificial bee colony algorithm based on modified search strategy', *Int. J. Computing Science and Mathematics*, Vol. 15, No. 4, pp.387–395.

Biographical notes: Kai Li is an undergraduate at the School of Information Engineering, Nanchang Institute of Technology, Nanchang, China. His research interest includes swarm intelligence and global optimisation.

Minyang Xu is a graduate student at the School of Information Engineering, Nanchang Institute of Technology, Nanchang, China. His research interest includes intelligent optimisation and application.

Tao Zeng is a graduate student at the School of Information Engineering, Nanchang Institute of Technology, Nanchang, China. His research interest includes intelligent optimisation and application.

Tingyu Ye is a graduate student at the School of Information Engineering, Nanchang Institute of Technology, Nanchang, China. His research interest includes intelligent optimisation and application.

Luqi Zhang is a graduate student at the School of Information Engineering, Nanchang Institute of Technology, Nanchang, China. His research interest includes intelligent optimisation and application.

Wenjun Wang is an Associated Professor at the School of Business Administration, Nanchang Institute of Technology, Nanchang, China. Her research interest includes swarm intelligence and decision optimisation. She has published more than 20 international journal/conference papers.

Hui Wang is a Professor at the School of Information Engineering, Nanchang Institute of Technology, China. His research interests include evolutionary optimisation, swarm intelligence, and large-scale optimisation. He has published more than 50 research papers in these fields.

1 Introduction

Many global optimisation problems exist in real world (Bangyal et al., 2020; Nedjah et al., 2020; Wang et al., 2020b). For complex optimisation problems, the performance of traditional optimisation methods are not ideal. To tackle this case, different types of intelligent optimisation algorithms were proposed, such as genetic algorithm (GA) (Zhang et al., 2020), bat algorithm (BA) (Boudjemaa et al., 2020), artificial bee colony (ABC) (Karaboga, 2005) and differential evolution (DE) (Yang et al., 2020).

Among the above algorithms, ABC has competitive performance and simple concept. However, ABC suffers from some problems. For instance, ABC shows weak exploitation capability and the convergence speed is slow. To improve the performance of the original ABC, many ABC variants were designed by modifying the search strategies (Banharnsakun et al., 2011; Ma et al., 2019; Peng et al., 2019; Wang et al., 2020a). In order to overcome the shortcomings of ABC, a novel ABC based on modified search strategy (namely MSABC) is proposed in this work, in which modified elite solutions are preserved and used to guide the search. In addition, the probability selection of the original ABC is replaced by the modified elite solutions in the onlooker bee phase. To evaluate the capability of MSABC, 22 classical problems are tested. Results demonstrate MSABC achieves superior performance than five other ABC variants.

The remaining parts of the paper are arranged as follows. Section 2 describes the original ABC. Our MSABC is given in Section 3. Section 4 presents the simulation experiments. Finally, this work is concluded in Section 5.

2 Artificial bee colony

Artificial bee colony (ABC) is a bio-inspired optimisation algorithm (Karaboga, 2005). It simulates the foraging behaviour of bees. The procedure of ABC is described as below.

Initialisation: In this step, SN solutions are randomly produced to form the initial population, and SN is the amount of nectars (population size). Each initial solution can be generated as below.

$$x_{i,j} = low + rand(0,1) \cdot (up - low), \tag{1}$$

where D is the dimension size, $i = 1, 2, \ldots, SN, j = 1, 2, \ldots, D, rand(0,1)$ is randomly generated in $[0,1]$, and $[low, up]$ is the search range.

Employed bee stage: In the population, each employed bee update an existing food source to find a better one.

$$v_{i,j} = x_{i,j} + \phi_{i,j} \cdot (x_{i,j} - x_{k,j}), \tag{2}$$

where $j \in \{1, 2, \ldots, D\}$ and $k \in \{1, 2, \ldots, SN\}$ are two random integers, and $i \neq k$. The factor $\phi_{i,j} \in [-1,1]$ is a random value. If the newly generated V_i wins X_i, X_i is substituted for V_i and $trial_i$ is updated. The related equations are described as follows.

$$X_i = \begin{cases} V_i, & \text{if } V_i \text{ better than } X_i \\ X_i, & \text{otherwise} \end{cases}, \tag{3}$$

$$trial_i = \begin{cases} 0, & \text{if } V_i \text{ better than } X_i \\ trial_i + 1, & \text{otherwise} \end{cases}, \tag{4}$$

Onlooker bee stage: In this step, the probability selection is employed to select better solutions for updating. The selected probability $prob(i)$ of each solution is defined by

$$prob(i) = \frac{fit(i)}{\sum_{i=1}^{SN} fit(i)}, \tag{5}$$

where $fit(i)$ is the fitness value of X_i.

Scout bee stage: If the maximal value of $trial_i$ exceeds $limit$, then the corresponding solution X_i will be initialised by equation (1).

3 The proposed algorithm MSABC

In this section, a modified search strategy is used to improve ABC. The new approach is called MSABC, which employs modified elite solutions to guide the search. Moreover, the modified elite solutions are used to replace the original probability selection in the onlooker bee phase.

Unlike other elite solutions, MSABC defines a modified elite solution. For the current X_i, its modified elite solution X_i^* should satisfy two conditions:

1 Establish a set E_i to store better solutions which are better than X_i

2 X_i^* is the shortest Euclidean distance between X_i and solutions in E_i.

Based on the concept of the modified elite solution, a modified search strategy is designed as follows.

$$v_{i,j} = x_{best,j} + \phi_{i,j} \cdot (x_{i,j}^* - x_{k,j}), \tag{6}$$

where X^* is the modified elite solution of X_i. X_k and $\phi_{i,j}$ are the same with the original ABC.

It is obvious that the modified elite solution X_i^* is better than X_i. In the onlooker bee phase, a probability selection is employed to choose better solutions. Then, we can use modified elite solutions to replace the probability selection to select better solutions.

The framework of MSABC is similar to the original ABC, and both of them have the same complexity. Compared with the original ABC, MSABC makes two modifications. Firstly, the search strategy of the employed and onlooker bees is equation (6), but not equation (2). Secondly, MSABC does not calculate the selection probability $prob(i)$ of each solution X_i. For each X_i, its modified elite solution X_i^* is selected for further search in the onlooker bee phase.

4 Experimental study

To assess the capability of MSABC, 22 benchmark problems are tested. Results of MSABC are compared with five other ABC algorithms. The details of these problems can be found in Wang et al. (2020a, 2021). The six ABC variants used for comparisons are ABC (Karaboga, 2005), GABC (Zhu and Kwong, 2010), MEABC (Wang et al., 2014), MABC (Gao et al., 2012), CABC (Dao et al., 2014) and our approach MSABC.

To make a fair comparison, the same parameter settings are employed. SN = 50 and $limit$ = 100 are used (Cui et al., 2016). For D = 30 and 100, the maximum number of function evaluations is set to 1.5×10^5 and 5.0×10^5, respectively. Each ABC algorithm is run 100 times on every problem. The coefficient C is equal to 1.5 in GABC (Zhu and Kwong, 2010). For CABC, MABC and MEABC, please refer to their corresponding literature (Dao et al., 2014; Gao et al., 2012; Wang et al., 2014).

Table 1 lists the computational results of all ABC variants on $D = 30$. The symbol $w/t/l$ in the last line is used to describe the overall competitive results of MSABC and other ABCs. The definitions of $w/t/l$ can be found in Wang et al. (2013a). From the results, MSABC has the same performance as the original ABC on two problems, and it surpasses the original ABC on other problems. MSABC and GABC obtain the same performance on six problems. For the rest of problems, GABC is worse than MSABC. MSABC is worse than MABC, MEABC and CABC on f_{18}. For the remaining problems, MSABC is not inferior to them. Among the six ABC algorithms, MSABC achieves the best or the same performance on 21 problems.

Table 1 Comparison results ($D = 30$)

Problem	*ABC*	*GABC*	*MEABC*	*MABC*	*CABC*	*MSABC*
	Average	*Average*	*Average*	*Average*	*Average*	*Average*
f_1	8.31E-16	6.53E-16	2.17E-41	7.16E-42	1.71E-50	*3.91E-62*
f_2	2.76E-15	5.74E-16	8.23E-37	3.05E-37	1.37E-42	*2.77E-59*
f_3	7.85E-16	4.24E-16	4.56E-41	5.32E-42	8.44E-52	*1.07E-62*
f_4	5.27E-14	2.41E-17	2.97E-87	1.68E-73	2.08E-56	*3.13E-127*
f_5	3.22E-11	2.08E-15	1.64E-20	1.26E-22	1.85E-26	*1.32E-32*
f_6	4.01E+01	9.03E-01	4.92E+00	5.07E+00	2.17E+01	*2.10E-01*
f_7	*0.00E+00*	*0.00E+00*	*0.00E+00*	*0.00E+00*	*0.00E+00*	*0.00E+00*
f_8	1.46E-20	8.38E-20	*7.18E-66*	*7.18E-66*	*7.18E-66*	*7.18E-66*
f_9	2.06E-01	1.94E-02	2.43E-02	3.23E-02	6.25E-02	*1.69E-02*
f_{10}	5.53E-01	2.21E+00	2.40E+00	5.90E+00	1.67E-01	*1.25E-01*
f_{11}	2.31E-15	*0.00E+00*	*0.00E+00*	*0.00E+00*	*0.00E+00*	*0.00E+00*
f_{12}	1.34E-02	*0.00E+00*	*0.00E+00*	*0.00E+00*	*0.00E+00*	*0.00E+00*
f_{13}	2.24E-15	1.73E-15	*0.00E+00*	*0.00E+00*	*0.00E+00*	*0.00E+00*
f_{14}	-12514.4	-12568.3	*–12569.5*	*–12569.5*	*–12569.5*	*–12569.5*
f_{15}	6.33E-07	3.21E-14	4.19E-14	3.91E+00	3.01E-14	*2.50E-14*
f_{16}	6.91E-16	5.46E-16	*1.57E-32*	*1.57E-32*	*1.57E-32*	*1.57E-32*
f_{17}	6.88E-16	4.95E-16	*1.35E-32*	*1.35E-32*	*1.35E-32*	*1.35E-32*
f_{18}	5.24E-07	1.99E-09	*7.58E-17*	*3.15E-17*	*2.11E-17*	1.40E-15
f_{19}	9.61E-13	7.29E-17	*1.35E-31*	*1.35E-31*	*1.35E-31*	*1.35E-31*
f_{20}	2.63E-01	*0.00E+00*	*0.00E+00*	*0.00E+00*	*0.00E+00*	*0.00E+00*
f_{21}	*–7.83E+01*	*–7.83E+01*	*–7.83E+01*	*–7.83E+01*	*–7.83E+01*	*–7.83E+01*
f_{22}	-2.92E+01	*–2.96E+01*	*–2.96E+01*	*–2.96E+01*	*–2.96E+01*	*–2.96E+01*
$w/t/l$	20/2/0	16/6/0	9/12/1	9/12/1	9/12/1	–

Table 2 presents the results of the six ABCs on 22 classical problems for $D = 100$. Compared with $D = 30$, the increase of dimensionality does not influence the overall performance of MSABC. All six ABC algorithms obtain the same performance on f_{21}. MSABC performs better than the original ABC on 21 problems. GABC outperforms MSABC on f_{15}. However, GABC is worse than MSABC on 15 problems. MEABC, MABC and MSABC have the same performance on 12 problems. MEABC is better than MSABC on f_{15} and f_{18}. MSABC surpasses MEABC on 8 problems. MABC and CABC are better than MSABC on f_{15}. MSABC wins MABC on 9 problems. CABC is not as good as MSABC on 11 problems. Among the six ABC algorithms, MSABC achieved the best or the same performance on 20 problems.

Friedman test and Wilcoxon test can be applied to further evaluate the overall capability of the six ABCs (Wang et al., 2013b; Xiao et al., 2021). Figures 1 and 2 illustrate that MSABC has the best performance among the six ABCs (D=30 and 100). Table 3 shows that MSABC is significantly better than five other ABCs. In the case of $D = 100$, MSABC is not significantly better than MEABC.

Table 2 Comparison results ($D = 100$)

	ABC	*GABC*	*MEABC*	*MABC*	*CABC*	*MSABC*
Problem	*Average*	*Average*	*Average*	*Average*	*Average*	*Average*
f_1	3.65E-15	6.18E-15	3.18E-37	8.02E-38	7.11E-49	*5.44E-63*
f_2	2.06E-14	4.22E-15	8.21E-32	1.49E-33	1.43E-40	*2.38E-61*
f_3	5.47E-15	1.60E-15	8.49E-36	2.93E-38	3.50E-49	*3.91E-63*
f_4	2.74E-09	4.34E-16	2.98E-83	2.86E-73	8.73E-35	*7.03E-130*
f_5	7.71E-09	2.65E-15	7.96E-20	2.73E-20	1.34E-25	*5.67E-30*
f_6	9.82E+01	3.36E+01	4.02E+01	5.61E+01	8.23E+01	*1.72E+00*
f_7	3.06E+00	*0.00E+00*	*0.00E+00*	*0.00E+00*	*0.00E+00*	*0.00E+00*
f_8	8.29E-39	9.41E-30	*7.12E-218*	*7.12E-218*	*7.12E-218*	*7.12E-218*
f_9	2.09E+00	*1.06E-01*	1.67E-01	1.70E-01	7.22E-01	*1.06E-01*
f_{10}	6.16E+00	4.61E+00	6.23E+00	4.29E+01	4.42E+00	*1.13E+00*
f_{11}	4.24E-07	*0.00E+00*	*0.00E+00*	*0.00E+00*	*0.00E+00*	*0.00E+00*
f_{12}	3.05E+00	*0.00E+00*	*0.00E+00*	*0.00E+00*	*0.00E+00*	*0.00E+00*
f_{13}	1.83E-15	1.39E-15	*0.00E+00*	*0.00E+00*	*0.00E+00*	*0.00E+00*
f_{14}	-41255.1	-41799.4	*–41898.3*	*–41898.3*	*–41898.3*	*–41898.3*
f_{15}	*2.92E-05*	*3.49E-13*	*3.67E-13*	5.05E+00	6.03E+00	8.33E-05
f_{16}	4.59E-15	3.31E-15	*4.71E-33*	*4.71E-33*	*4.71E-33*	*4.71E-33*
f_{17}	7.11E-15	3.29E-15	1.35E-32	1.35E-32	*1.37E-32*	1.35E-32
f_{18}	2.68E-04	1.12E-05	*9.20E-17*	*2.95E-18*	*1.17E-26*	2.77E-07
f_{19}	8.63E-13	1.09E-15	*1.35E-31*	*1.35E-31*	*1.35E-31*	*1.35E-31*
f_{20}	5.83E+00	*0.00E+00*	*0.00E+00*	*0.00E+00*	*0.00E+00*	*0.00E+00*
f_{21}	*–7.83E+01*	*–7.83E+01*	*–7.83E+01*	*–7.83E+01*	*–7.83E+01*	*–7.83E+01*
f_{22}	-9.01E+01	-9.79E+01	*–9.95E+01*	*–9.95E+01*	*–9.95E+01*	*–9.95E+01*
$\boldsymbol{w/t/l}$	20/1/1	15/6/1	8/12/2	9/12/1	11/10/1	——

Figure 1 Mean rank values achieved by all ABCs ($D = 30$)

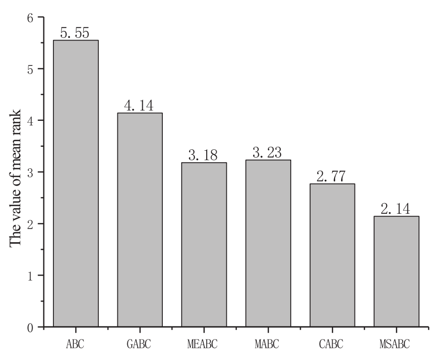

Figure 3 gives the convergence features of MSABC and other algorithms on f_1, f_2, f_5, f_6, f_9 as well as f_{15} ($D = 30$). For problem f_9, MSABC, GABC and MEABC have similar convergence performance. MSABC converges much faster than five other ABCs on five problems (f_1, f_2, f_5, f_6, and f_{15}).

Figure 2 Mean rank values achieved by all ABCs (D=100)

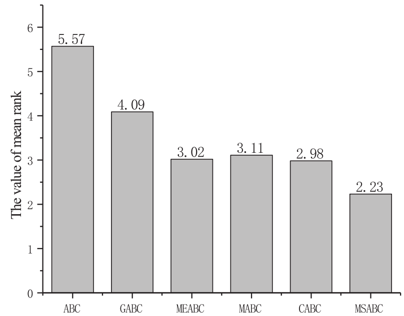

Figure 3 Convergence curves of six ABCs on f_1, f_2, f_5, f_6, f_9 and f_{15} ($D = 30$): (a) sphere (f_1); (b) elliptic (f_2); (c) Schwefel 2.22 (f_5); (d) Schwefel 2.21 (f_6); (e) quartic (f_9) and (f) Ackley (f_{15})

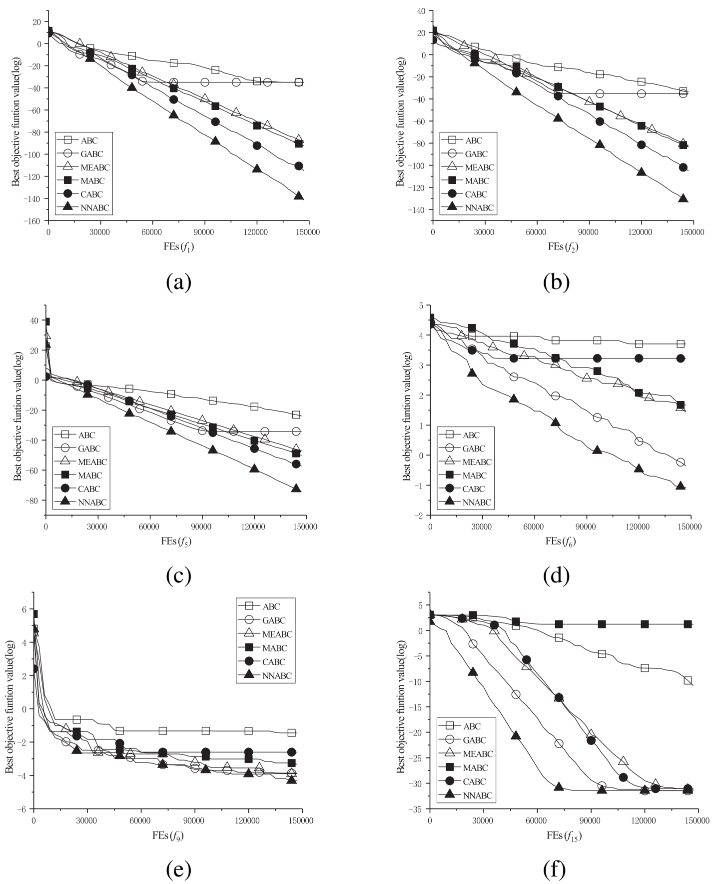

Table 3 Wilcoxon test results

MSABC vs.	$D = 30$	$D = 100$
	p-value	*p-value*
ABC	*8.90E-05*	*3.21E-04*
GABC	*4.38E-04*	*3.78E-03*
MEABC	*2.84E-02*	1.39E-01
MABC	*2.84E-02*	*2.84E-02*
CABC	*2.84E-02*	*2.08E-02*

5 Conclusion

To strengthen the optimisation performance of ABC, a modified search strategy is utilised to improve ABC. The new algorithm MSABC has two modifications:

1 a new search strategy is designed based on the modified elite solution

2 the modified elite solution is used to replace the probability selection.

In the experiments, 22 classical benchmark problems are used to test the performance of MSABC, ABC, GABC, MEABC, MABC, and CABC. Computational results demonstrate that the proposed strategies can significantly improve the solution quality and convergence speed.

Acknowledgement

This work was supported by National Natural Science Foundation of China (No. 62166027), and Jiangxi Provincial Natural Science Foundation (Nos. 20212ACB212004 and 20212BAB202023).

References

Bangyal, W.H., Ahmed, J. and Rauf, H.T. (2020) 'A modified bat algorithm with torus walk for solving global optimisation problems', *International Journal of Bio-Inspired Computation*, Vol. 15, No. 1, pp.1–13.

Banharnsakun, A., Achalakul, T. and Sirinaovakul, B. (2011) 'The best-so-far selection in artificial bee colony algorithm', *Appllied Soft Computing*, Vol. 11, No. 2, pp.2888–2901.

Boudjemaa, R., D Oliva and Ouaar, F. (2020) 'Fractional Lévy flight bat algorithm for global optimisation', *International Journal of Bio-Inspired Computation*, Vol. 15, No. 2, pp.100–112.

Cui, L., Li, G., Lin, Q., Du, Z., Gao, W., Chen, J. and Lu, N. (2016) 'A novel artificial bee colony algorithm with depth-first search framework and elite-guided search equation', *Information Sciences*, Vol. 367, pp.1012–1044.

Dao, T.K., Chu, S.C., Nguyen, T.T., Shieh, C.S. and Horng, M.F. (2014) 'Compact artificial bee colony', *Modern Advances in Applied Intelligence*, pp.96–105.

Gao, W.F. and Liu, S.Y. (2012) 'A modified artificial bee colony algorithm', *Comput. Oper. Res.*, Vol. 39, pp.687–697.

Karaboga, D. (2005) *An Idea based on Honey Bee Swarm for Numerical Optimization*, Technical Report-TR06, Erciyes University, Kayseri, Turkey.

Ma, L.B., Wang, X.W., Shen, H. and Huang, M. (2019) 'A novel artificial bee colony optimiser with dynamic population size for multi-level threshold image segmentation', *International Journal of Bio-Inspired Computation*, Vol. 13, No. 1, pp.32–44.

Nedjah, N., Mourelle, L. and Morais, R.G. (2020) 'Inspiration-wise swarm intelligence meta-heuristics for continuous optimisation: a survey-part I', *International Journal of Bio-Inspired Computation*, Vol. 15, No. 4, pp.207–223.

Peng H., Deng, C., Wu, Z. (2019) 'Best neighbor guided artificial bee colony algorithm for continuous optimization problems', *Soft Computing*, Vol. 23, No. 18, pp.8723–8740.

Wang, H., Wang, W.J., Xiao, S., Cui, Z. and Zhou, X. (2020a) 'Improving artificial bee colony algorithm using a new neighborhood selection mechanism', *Information Sciences*, Vol. 527, pp.227–240.

Wang, F., Li, Y., Zhou, A. and Tang, K. (2020b) 'An estimation of distribution algorithm for mixed-variable newsvendor problems', *IEEE Trans. EVol. Comput.*, Vol. 24, No. 4, pp.479–493.

Wang, H., Wang, W.J., Zhou, X.Y., Zhao, J., Wang, Y., Xiao, S.Y. and Xu, M.Y. (2021) 'Artificial bee colony algorithm based on knowledge fusion', *Complex & Intelligent Systems*, Vol. 7, No. 3, pp.1139–1152.

Wang, H., Rahnamayan, S., Sun, H. and Omran, M.G.H. (2013a) 'Gaussian bare-bones differential evolution', *IEEE Transactions on Cybernetics*, Vol. 42, No. 2, pp.634–647.

Wang, H., Sun, H., Li, C., Rahnamayan, S. and Pan, J. (2013b) 'Diversity enhanced particle swarm optimization with neighborhood search', *Information Sciences*, Vol. 223, pp.119–135.

Wang, H., Wu, Z., Rahnamayan, S., Sun, H., Liu, Y. and Pan, J. S. (2014) 'Multi-strategy ensemble artificial bee colony algorithm', *Information Sciences*, Vol. 279, pp.587–603.

Xiao, S.Y., Wang, H., Wang, W.J., Huang, Z.K., Zhou, X.Y. and Xu, M.Y. (2021) 'Artificial bee colony algorithm based on adaptive neighborhood search and Gaussian perturbation', *Applied Soft Computing*, Vol. 100, Article 106955.

Yang, S., Huang, Q.L., Cui, L., Xu, K.K., Ming, Z. and Wen, Z.K. (2020) 'Variable-grouping-based exponential crossover for differential evolution algorithm', *International Journal of Bio-Inspired Computation*, Vol. 15, No. 3, pp.147–158.

Zhang, X., Li, X.T. and Yin, M.H. (2020) 'An enhanced genetic algorithm for the distributed assembly permutation flowshop scheduling problem', *International Journal of Bio-Inspired Computation*, Vol. 15, No. 2, pp.113–124.

Zhu, G. and Kwong, S. (2010) 'Gbest-guided artificial bee colony algorithm for numerical function optimization', *Appl. Math. Comput.*, Vol. 217, No. 7, pp.3166–3173.