An Improved Exhausted-Food-Sources-Identification Mechanism for the Artificial Bee Colony Algorithm

Jiaxu Ning[1,2](✉), Haitong Zhao[2], Peng Sun[3], Yunfei Feng[4], and Tianyu Zhao[5]

[1] Shenyang Ligong University, Shenyang 110159, People's Republic of China
739250969@qq.com
[2] Northeastern University, Shenyang 110819, People's Republic of China
[3] IOWA State University, Ames, IA 50010, USA
[4] Sam's Club Technology Walmart Inc., Bentonville, AR 72712, USA
[5] College of Electronic Science and Engineering, Jilin University, Changchun, China

Abstract. Artificial bee colony (ABC) algorithm has been widely used to solve the optimization problems. In the existing ABC algorithms, choosing which employed bee giving up its food source only based on its current trial number. It may cause some promising areas are exploited insufficiently and some non-significant areas are searched excessively. Thus, much more searching resources are wasted. To cope with this problem, an improved exhausted food source identification mechanism based on space partitioning (ISP) is designed, which considers the food source states both in the objective space and searching space simultaneously. Then, the proposed mechanism is applied to the basic ABC algorithm and a recently improved ABC algorithm. The experimental results have demonstrated that the ABC algorithms with the designed exhausted food source identification mechanism perform better than the original ABC algorithms in almost all the functions on the CEC2015 test suit.

Keywords: Swarm intelligence · Optimization problem · Artificial Bee Colony Algorithm

1 Introduction

Swarm intelligence (SI) has become a significant research subfield of artificial intelligence inspired by natural behavior of the swarm individuals [1]. The artificial bee colony (ABC) algorithm is a popular SI-based algorithm by simulating waggle dance and foraging behaviors of real honey bee colonies [2]. Due to the simple concept, easy implementation and fast convergence, ABC has attracted much attention and wide applications in numerical optimization domain and engineering applications [3]–[6].

In the ABC model, when the food sources are abandoned, the employed bee related to it becomes a scout. Then, a food source is produced for this scout, and the scout bee becomes an employed bee again to consume this food source. We can see that the worth of scout lie in that it can make the exhausted food source be abandoned timely to

H. Song et al. (Eds.): SIMUtools 2019, LNICST 295, pp. 633–642, 2019.
https://doi.org/10.1007/978-3-030-32216-8_62

save search effort. In term of simulating the behavior of scout bees, identify the accurate exhausted food source, most of the literatures are the same as the basic ABC algorithm. If there is more than one food source whose trail number exceeds the limit number, literatures [7]–[9] identified the food source with the largest trail number as an exhausted source of food, and [10]–[12] selected only one food source as an exhausted food source. Literature [11] proposed MNIIABC algorithm, where the scout bees also added a judgment mechanism to guarantee that the new generated food source was different from the abandoned exhausted food source. Literature [13] selected a food source according to their probabilities at first. And then judge it whether it is an exhausted food source by comparing its trail number with the limit number.

However, these identification mechanisms perform not well in identifying the exhausted food sources. The larger the area around a food source is not explored, that is, the larger the subspace volume of the food source is, the higher the search frequency of this food source is compared with other food sources, then the trail number will be higher. Therefore, according to the original identification mechanism, it is of great possibility to identify this food source as an exhausted food source, and this recognition is likely to be wrong.

In order to avoid this deficiency, this paper introduces a novel exhausted-food-source-identification mechanism based on space partitioning (ISP) in the scout bees phase, which identifies exhausted food sources more accurately by judging the volume and density of the subspace of every food source. For food sources with the same trail number, the food source with smaller subspace volume and greater subspace density is supposed to be identified as an exhausted food source with great possibility. In addition, the experimental results on the 15 test functions on CEC2015 test suit have demonstrated that ABC algorithms with ISP mechanism perform better than the ABC algorithms with the original identification mechanism.

2 The Artificial Bee Colony Optimization Model

Artificial Bee Colony (ABC) is one of the most recently defined algorithms by Karaboga [2], motivated by the intelligent forage behavior of honey bees. In ABC algorithm, the colony of artificial bees consists of three groups of bees: employed bees, onlookers and scouts. A food source represents a possible solution to the problem tobe optimized. The nectar amount of a food source corresponds to the quality of the solution represented by that food source. For every food source, there is only one employed bee. The basic structure of the ABC algorithm can be divided into the initialization stage, employed bee stage, onlooker bee stage and scoutstage.

At the initialization stage, it is supposed that the initial population of the food sources is made up of SN number of n-dimensional real valued vectors, and the ith solution of the population can be represented as $\mathbf{x}_i = \{x_{i1}, \ldots, x_{in}\}$. Then the SN candidate solutions are randomly generated by

$$x_{ij} = l_j + rand(0,1)(u_j - l_j) \tag{1}$$

where l_j and u_j are the lower and upper bound constraints of the jth variable of $\boldsymbol{x}_i$, respectively. At the employed bee stage, each employed bee flies to a food source and explores it by

$$v_{ij} = x_{ij} + \varphi_{ij}(x_{ij} - x_{kj}) \tag{2}$$

where φ_{ij} is uniformly distributed random real number in $[-1, 1]$, $k \in \{1, \ldots, i-1, i+1, \ldots, SN\}$ is randomly chosen and $j \in \{1, \ldots, n\}$ is a randomly chosen dimension. After generating the $\mathbf{v}_i$ by Eq. (2), there is a greedy selection between $\mathbf{x}_i$ and $\mathbf{v}_i$ by Eq. (3).

$$\mathrm{x}_i = \begin{cases} \mathrm{v}_i, & \textit{if fit}(\mathrm{v}_i) > \textit{fit}(\mathrm{x}_i), \\ \mathrm{x}_i, & \text{otherwise}, \end{cases} \tag{3}$$

Where $fit(\mathbf{x}_i)$ means the fitness value of $\mathbf{x}_i$. For a minimization problem, the fitness value of a solution can be defined as:

$$\textit{fit}(\mathrm{x}_i) = \begin{cases} \frac{1}{1+f(\mathrm{X}_i)}, & \textit{if} \quad f(\mathrm{x}_i) > 0, \\ 1 + |f(\mathrm{x}_i)|, & \textit{if} \quad f(\mathrm{x}_i) \leq 0, \end{cases} \tag{4}$$

Where $f(\mathbf{x}_i)$ is the value of the objective function at $\mathbf{x}_i$.

At the onlooker bee stage, every onlooker bee randomly selects a solution from the SN solutions with a probability p_i. The probability of a food source chosen by an onlooker bee can be calculated by:

$$p_i = \textit{fit}(\mathrm{x}_i) \Big/ \sum\nolimits_{i=1}^{SN} \textit{fit}(\mathrm{x}_i) \tag{5}$$

As can be seen from Eq. (5), the solution with a better fitness has a higher probability selected by an onlooker bee. Once the onlooker bee has chosen her food source, an exploitation is made on it by Eq. (2) to generate a new solution, then a greedy selection is made by Eq. (3) between the new solution and the old one to remain a better solution.

After all employed bees and onlooker bees have explored their food sources, there is a check to see whether there is an exhausted food source need tobe abandoned or not at the scout stage. Here the exhausted food means a food source that has not been improved over the last LIMIT cycles. The LIMIT is a predetermined control parameter of the ABC algorithm. If there is an exhausted food source $\mathbf{x}_i$, a scout bee discovers a new food source by Eq. (1) to replace it.

3 The Exhausted Food Source Identification Mechanism Based on Space Partitioning

As we can see from what have been mentioned in Sect. 2, there is a shortcoming in the ABC algorithm. It is of great possibility to identify an abundant food source as an exhausted food source if we only judge it by comparing the trail number with the Limit number. In the following, we introduced a novel exhausted-food-sources-identification (ISP) mechanism to identify the exhausted food sources, for not giving up a sufficient supply in a sense. That is, for food sources with the same trail number, the food source with smaller subspace volume and greater subspace density is supposed to be identified as an exhausted food source with greater possibility. Therefore, the greater the subspace density is, the greater the probability that this food source will be identified as an exhausted one is. Hence, scout bees utilize this searching space information to make the selection of food source to abandon more accurate.

As is shown in Fig. 1, assuming that the trail numbers in food sources X9 and X10 are both greater than the Limit number and the subspace density of the food source X9 is higher than that of X10, then X9 will be identified as the exhausted food source with a higher probability.

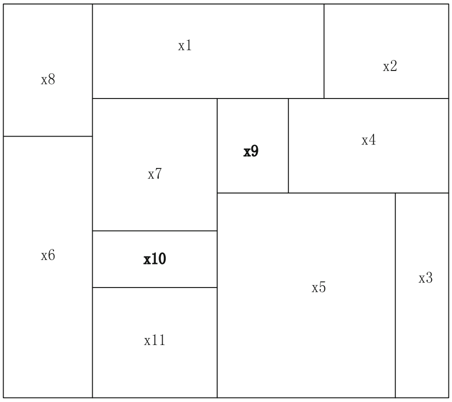


Fig. 1. The diagram of sub-regions of the exhausted food sources x9 and x10

Space partitioning in this paper is implemented through a Binary Space Partitioning (BSP) tree [14], which preserves all the historical food sources in the process of iteration. Each leaf node in the tree represents a subspace, the subspaces do not overlap, and the sum of all subspaces is the entire area of all food sources around a hive. While partitioning, the boundary values are determined by using two food sources that belong to the same subspace. The search density for the current subspace is evaluated by the super volume of the subspace.

Definition 1 (Subspace of x): Assume x is a food source in the search space A ($x \in A$), and A is partitioned as the subspace set H by BSP tree, we define the subspace $h \subseteq H$ as the "subspace of x" if $x \in h$ and h is represented by a leaf node of BSP tree.

Definition 2 (Subspace density of x): Supposing that there is only one food source x in a subspace h. The subspace density of x can be evaluated by figuring the subspace super volume value of h utilizing Eqs. (6) and (7), where v(x) refers to the subspace size, u_k and l_k are the upper and lower limits for dimension k respectively, and D (x) refers to the subspace density of x.

$$v(x) = \prod_{k=1}^{D} (u_k - l_k) \tag{6}$$

$$D(x) = \frac{1}{v(x)} \tag{7}$$

The steps of space partitioning are as follows:

Step1: Create an empty BSP tree, where there is only one root node in the tree.
Step2: After a new food source v is found, v will be saved in the root node if there is only the root node in the current tree, and the partitioning process will be completed. Otherwise, supposing h is the subspace of v, search the node corresponding to h and the unique solution x stored in it.
Step3: Calculate the partition dimension j according to the food sources v and x, and subspace h in the dimension j is divided into two parts.
Step4: Create two nodes n1 and n2, save the food sources x and z, the corresponding subspace boundary values. Then insert n1 and n2 as left and right children of node n into the tree.

Based on the above descriptions, the exhausted-food-sources-identification (ISP) mechanism is designed as follows in this paper. In scout bee phase, a scout bee first determines the number N of the candidate food sources whose trail number is greater than the limit number. If N = 0, the scout bees phase ends. If N = 1, identify this food source as an exhausted one and generate a new food source V utilizing Eq. (1) to substitute X meanwhile reset its trail = 0. If N > 1, first find the subspaces of these candidate food sources. Then calculate their subspace density values and the probability values P_{di} utilizing Eqs. (6), (7) and (8) respectively. After that, identify the exhausted food source X utilizing the greedy selection mechanism according to the subspace probability values P_{di}. Finally, generate a new food source V utilizing Eq. (1) to substitute X and reset its trail = 0.

$$p_{di} = \frac{D(\mathrm{X}_i)}{\sum D(\mathrm{X}_i)} \tag{8}$$

In the following, the paper integrates ABC algorithm with ISP to better identify the exhausted food sources, which is called ABC-ISP in the following parts. It is still performed into four steps: initialization phase, employed bees phase, onlooker bees phase and scout bees phase. Except that a BSP tree is created to preserve all the food sources and their subspace boundaries. It is significant to figure out that the ISP mechanism can also be easily integrated with other ABC variants. So, to verify the effectiveness of the ISP mechanism for the sending scout phrase, we also introduce it into the recently proposed improved ABC algorithm called ABCG [15]. And the pseudo-code of ABC algorithm with ISP is presented in Table 1 to better illustrate the integration way of it.

Table 1. The pseudo-code description of ABC algorithm with ISP

AlgorithmABC with ISP
Begin
1. Initialize density tree T which only has a root node
2. Initialization phrase
3. Memorize the best food source position
4. Insert food sources into T use algorithm Insert (T, x)
5.**while** not stopping criteria met **do**
6. Employed bees phrase
7. Insert food sources founded in the employed bees phrase into T
8. Onlooker bees phrase
9. Insert food sources founded in the onlooker bees phrase into T
10. Memorize the best food source position
/* Scout bee phrase*/
11. Count the number of candidate exhausted food sources N use trial number
12. **if** N=1 **then**
13. Generate a new solution randomly V use Eq.(1)
14. Replace the exhausted solution with V
15. Insert (T,V)
16. **end if**
17. **if** N>1 **then**
18. **for**i=1 to N **do**
19. Search (T, X_i)
20. Calculate super volume of subspace containing X_i use Eq. (6)
21. Calculate density probability P_{di} of subspace containing X_i utilizing Eq. (7)
22. **end for**
25. Identify the exhausted food source X by a greedy selection mechanism according to P_{di}
26. Generate a new food source V use Eq. (1)
27 Substitute food source V for X
28. Insert (T, V)
29. **end if**
30.**end while**
end

4 Experimental Results

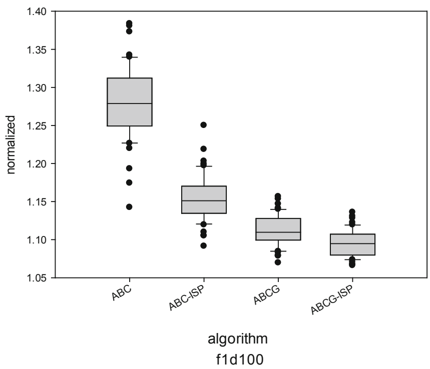
In this part, the ABC-ISP and the ABCG with ISP mechanism called ABCG-ISP algorithms are tested and computed with the original ABC and ABCG algorithms. All the experiments are tested on 15 test functions of the CEC2015 test suit. The dimension D = 100, the whole search space is set to $[-100, 100]^D$ for all the test functions. All algorithms are terminated when the maximum fitness evaluation number MFE = 10000*D is arrived, and the other parameters of these algorithms are set as the same as in [2] and [15]. All the experiments are done on the same computer (2.93 GHz CPU and 3 GB RAM) with Visual Studio 2008.

The main experiment results are that the algorithms solve the test functions and run 51 times independently to obtain the minimum function values. The achieved maximum, minimum and the mean values by the compared algorithms on each test instance are given in Table 2. The distributions of the obtained results for the compared algorithms are shown as Fig. 2. In this figure, there are five solid lines in every diagram, each solid line from the top to the bottom on behalf of: the maximum point (there may exists abnormal points), quarter-digit, median, three-quarters and the minimum point (there may exists abnormal). For what we care for is to obtain the minimum function optimization, therefore, the smaller the obtained function value is, the quality of the algorithm is better. The performances of all the algorithms can get directly through this box diagrams. And the convergences of the algorithms on the test instances are shown as in Fig. 3.

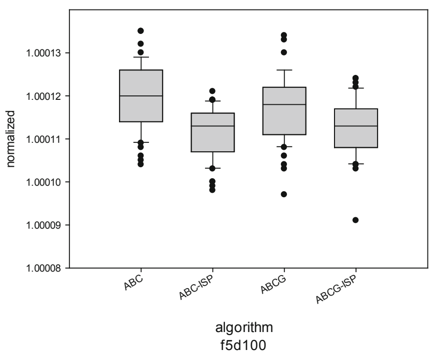
Table 2. Results of compared algorithm based on description statics way

Function	Result	ABC	ABC-ISP	ABCG	ABCG-ISP
F1	Min	1.40E+07	8.98E+06	6.84E+06	6.50E+06
	Max	3.77E+07	2.46E+07	1.54E+07	1.34E+07
	Mean	2.76E+07	1.51E+07	1.11E+07	9.32E+06
F5	Min	1.03E+04	9.75E+03	9.64E+03	9.05E+03
	Max	1.34E+04	1.20E+04	1.33E+04	1.23E+04
	Mean	1.19E+04	1.11E+04	1.16E+04	1.12E+04
F8	Min	7.07E+06	3.06E+06	3.48E+06	3.13E+06
	Max	1.48E+07	1.35E+07	1.16E+07	1.19E+07
	Mean	1.09E+07	8.33E+06	7.40E+06	7.24E+06
F10	Min	4.59E+05	1.04E+06	8.67E+04	5.12E+04
	Max	6.99E+06	3.11E+06	1.13E+06	8.81E+05
	Mean	4.06E+06	1.86E+06	5.30E+05	3.49E+05
F14	Min	9.79E+04	9.74E+04	9.82E+04	9.73E+04
	Max	1.10E+05	1.09E+05	1.08E+05	1.05E+05
	Mean	1.04E+05	1.01E+05	1.02E+05	1.00E+05

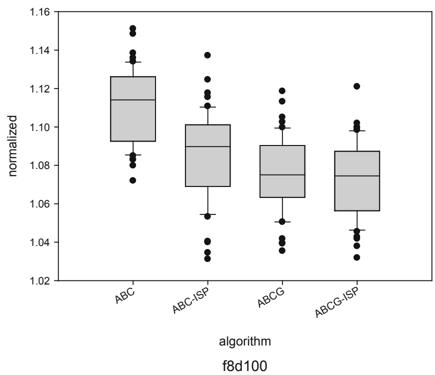
From the Table 2, it can be seen that the ABC-ISP and ABCG-ISP achieved smaller maximum, minimum and the mean fitness values than the ABC and ABCG algorithm respectively, which can be further proved by the Fig. 2. From the convergence curve of these algorithms on the test instances in Fig. 3, we can see that the ABC-ISP and ABCG-ISP not only converge faster than the ABC and ABCG algorithm, but also converge to a better position on each test function. So, we can conclude that the designed ISP mechanism works well.



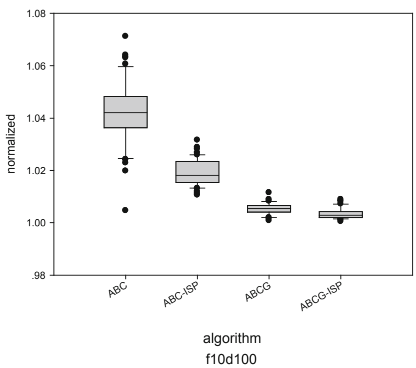
(a) F1



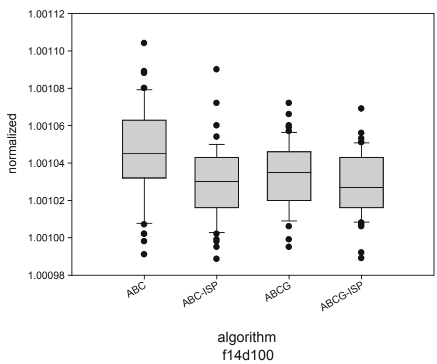
(b) F5



(c) F8

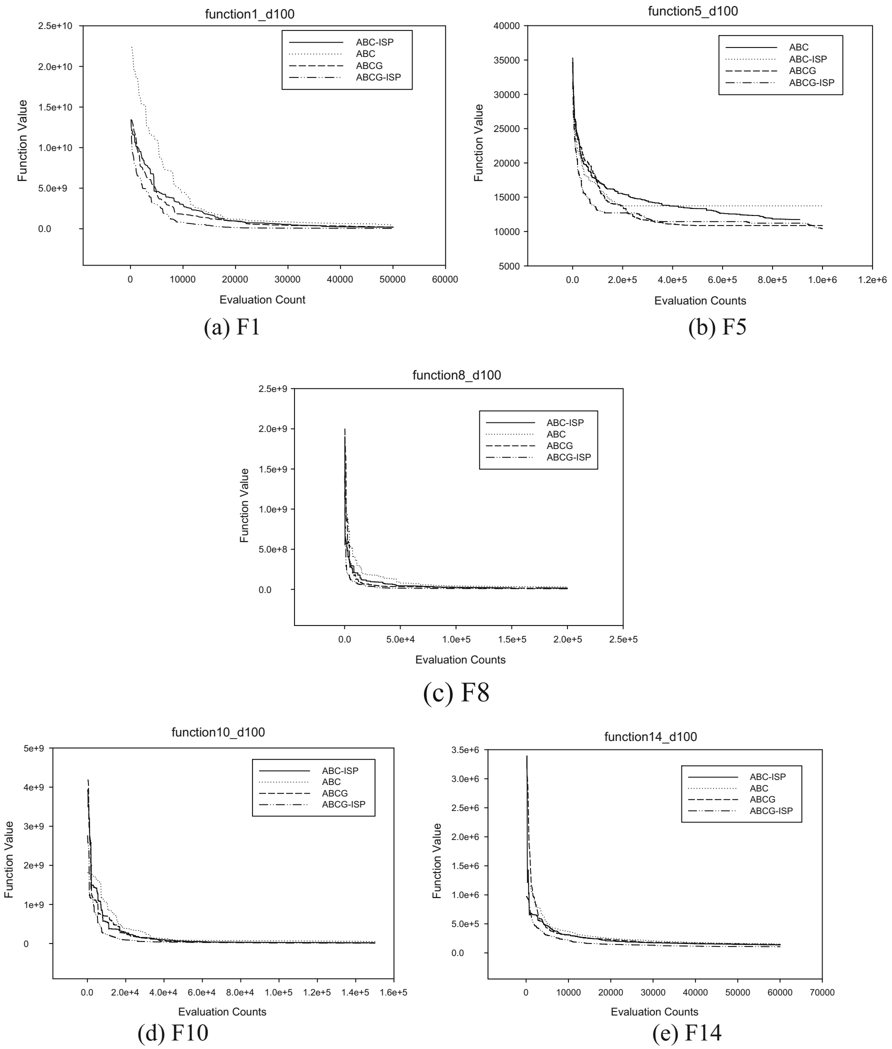


(d) F10



(e) F14

Fig. 2. The boxplots of the compared algorithms on CEC2015 test functions



(a) F1 (b) F5 (c) F8 (d) F10 (e) F14

Fig. 3. The convergence curves of the compared algorithms on CEC2015 test functions

5 Conclusion

In this paper, an improved exhausted food source identification mechanism based on space partitioning (ISP) is designed. Instead of presenting a new hybrid ABC algorithm or integrating an operator of an existing algorithm into ABC, our aim is to model the behavior of foragers in ABC more accurately. By using the new selection mechanism proposed for scoutbees, the ABC algorithms combined with it achieved a better performance than the original versions in terms of solution quality and convergence characteristics. Through this work, we can see that combining the searching space information and objective space information during optimizing process can effectively improve the algorithm performance.

Acknowledgement. This work is funded by Shenyang Dongda Emerging Industrial Technology Research Institute.

References

1. Zhang, S., Lee, C.K.M., Chan, H.K., Choy, K.L., Zhang, W.: Swarm intelligence applied in green logistics: a literature review. Eng. Appl. Artif. Intell. **37**, 154–169 (2015)
2. Karaboga, D., Basturk, B.: A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J. Glob. Opt. **39**(3), 459–471 (2007)
3. Karaboga, D., Beyza, G., Celal, O., Karaboga, N.: A comprehensive survey: artificial bee colony (ABC) algorithm and applications. Artif. Intell. Rev. **42**(1), 21–57 (2014)
4. Jiang, D., Huo, L., Lv, Z., et al.: A joint multi-criteria utility-based network selection approach for vehicle-to-infrastructure networking. IEEE Trans. Intell. Transp. Syst. **pp**(99), 1–15 (2018)
5. Jiang, D., Huo, L., Song, H.: Rethinking behaviors and activities of base stations in mobile cellular networks based on big data analysis. IEEE Trans. Netw. Sci. Eng. **1**(1), 1–12 (2018)
6. Jiang, D., Xu, Z., Li, W., et al.: Topology control-based collaborative multicast routing algorithm with minimum energy consumption. Int. J. Commun Syst **30**(1), 1–18 (2017)
7. Karaboga, D., Gorkemli, B.: A quick artificial bee colony (qABC) algorithm and its performance on optimization problems. Appl. Soft Comput. **23**(5), 227–238 (2014)
8. Gao, W.F., Liu, S.Y.: A modified artificial bee colony algorithm. Comput. Oper. Res. **39**(3), 687–697 (2012)
9. Cui, L., Zhang, K., Li, G., et al.: Modified Gbest-guided artificial bee colony algorithm with new probability model. Soft. Comput. **2017**(2), 1–27 (2017)
10. Yu, W.J., Zhan, Z.H., Zhang, J.: Artificial bee colony algorithm with an adaptive greedy position update strategy. Soft. Comput. **2016**, 1–15 (2016)
11. Zhong, F., Li, H., Zhong, S.: An improved artificial bee colony algorithm with modified-neighborhood-based update operator and independent-inheriting-search strategy for global optimization. Eng. Appl. Artif. Intell. **58**, 134–156 (2017)
12. Bai, W., Eke, I., Lee, K.Y.: An improved artificial bee colony optimization algorithm based on orthogonal learning for optimal power flow problem. Control Eng. Practice **61**, 163–172 (2017)
13. Kishor, A., Chandra, M., Singh, P.K.: An astute artificial bee colony algorithm. In: Deep, K., et al. (eds.) Proceedings of Sixth International Conference on Soft Computing for Problem Solving. AISC, vol. 546, pp. 153–162. Springer, Singapore (2017). https://doi.org/10.1007/978-981-10-3322-3_14
14. Hearn, D., Baker, M.P.: Computer Graphics with OpenGl, 3rd edn. Prentice-Hall, Upper Saddle River (2004)
15. Xiang, W., Meng, X., Li, Y., He, R.: An improved artificial bee colony algorithm based on the gravity model. Inf. Sci. **429**, 49–71 (2018)