

DOI: 10.11830/ISSN.1000-5013.201612038



# 平衡搜索的改进人工蜂群算法

刘晓芳<sup>1,2</sup>, 柳培忠<sup>1,2</sup>, 骆炎民<sup>3</sup>, 范宇凌<sup>1</sup>

(1. 华侨大学 工学院, 福建 泉州 362021;  
2. 华侨大学 工业智能化技术与系统福建省高校工程研究中心, 福建 泉州 362021;  
3. 华侨大学 计算机科学与技术学院, 福建 厦门 361021)

**摘要:** 针对人工蜂群(ABC)算法局部搜索能力弱的问题,提出一种平衡搜索的人工蜂群算法(BSABC). 首先,采用一种基于对数函数的的适应度评价方式,用于减小选择压力,在一定程度上避免陷入局部最优. 其次,受微分进化算法的启发,提出一种新的搜索策略,通过当前最优个体指导进化方向,使候选解的产生倾向于当前最优解,同时避免陷入局部最优. 对 6 个经典测试函数进行仿真实验,并与经典的改进人工蜂群算法对比测试,结果表明:所提出的算法在收敛速度和收敛精度上都有显著的提升.

**关键词:** 人工蜂群算法; 局部搜索; 群智能算法; 适应度评价; 搜索策略

**中图分类号:** TP 18      **文献标志码:** A      **文章编号:** 1000-5013(2019)01-0128-05

## Improved Artificial Bee Colony Algorithm Based on Balanced Search

LIU Xiaofang<sup>1,2</sup>, LIU Peizhong<sup>1,2</sup>,  
LUO Yanmin<sup>3</sup>, FAN Yuling<sup>1</sup>

(1. College of Engineering, Huaqiao University, Quanzhou 362021, China;  
2. Universities Engineering Research Center of Fujian Province Industrial Intelligent Technology and Systems,  
Huaqiao University, Quanzhou 362021, China;  
3. College of Computer Science and Technology, Huaqiao University, Xiamen 361021, China)

**Abstract:** Aim at the drawback of artificial bee colony (ABC) algorithm with weak local search capability, an artificial bee colony algorithm based on balanced search (BSABC) is proposed. Firstly, improved fitness evaluation methods based on the logarithmic function is introduced to minimize selection pressure and avoid to fall into local optimum to a certain extent. Secondly, enlightened by the differential evolution algorithm, a novel search strategy is proposed, which conducts the evolution direction of the candidate solution, tending to the current optimal solution, and at the same time avoiding to fall into the local optimum. The simulating experiments were conducted on a benchmark suite of 6 test functions, the results demonstrate that BSABC has significant enhancement in convergent speed and convergent accuracy compared with the basic ABC algorithm.

**Keywords:** artificial bee colony algorithm; local search; swarm intelligence algorithm; fitness evaluation; search strategy

**收稿日期:** 2016-12-20

**通信作者:** 柳培忠(1976-),男,讲师,博士,主要从事仿生智能计算、多维空间仿生信息学的研究. E-mail: pzliu@hqu.edu.cn.

**基金项目:** 国家自然科学基金资助项目(61203242); 福建省物联网云计算平台建设资助项目(2013H2002); 华侨大学研究生科研创新能力培育计划资助项目(1511322003)

与其他智能算法相比,人工蜂群算法<sup>[1-2]</sup>最大的优点在于开采和开发同时进行,增加了寻找到最优解的概率. 但仍然存在收敛速度慢,易陷入局部最优,开采和开发能力不平衡等问题. Zhu 等<sup>[3]</sup>受粒子群算法(PSO)的启发,提出一种改进算法(gbest-guide ABC,GABC),在搜索策略中引入全局最优用于加快收敛速度,并采用经典测试函数证明了改进策略的有效性. Banharnsakun 等<sup>[4]</sup>针对人工蜂群算法采用至今最优选择个体引导进化方向,使候选解向至今最优选择个体靠拢,并采用自适应调整搜索范围的方法,避免陷入局部最优. 高卫峰等<sup>[5]</sup>充分利用和平衡不同搜索方程的探索和开发能力,提出一种改进的人工蜂群算法(IABC),实验结果表明,该改进算法具有良好的处理复杂数值优化问题的性能. 宁爱平等<sup>[6]</sup>利用随机过程理论,对人工蜂群算法收敛性进行理论分析. Li 等<sup>[7]</sup>提出一种离散人工蜂群算法(DABC),该算法提出一种独特的解决方案,采用两个离散向量和禁忌搜索策略生成新的候选解,并用于解决多目标柔性作业车间调度问题. Kiran 等<sup>[8]</sup>为了有效地解决不同特性的优化问题,提出一种整合 5 个搜索策略,并采用计数器控制更新的解决方案. 目前,人工蜂群算法的改进版本各式各样,并有一定的性能提升,但仍存在收敛速度慢、局部搜索能力弱的缺点. 针对以上问题,本文提出一种收敛速度更快、局部搜索能力更强的人工蜂群算法.

## 1 基本的人工蜂群算法

人工蜂群算法是模拟蜜蜂群体搜索优良蜜源的优化算法,把蜜蜂种群分为雇佣蜂、观察蜂和侦察蜂 3 种类型,分工合作,以找到满意的蜜源<sup>[2]</sup>. 该算法随机生成初始食物源(初始解),其公式为

$$x_i^j = x_{\min}^j + \text{rand}(0,1)(x_{\max}^j - x_{\min}^j). \tag{1}$$

式(1)中: $i=1,2,\cdots,\text{SN}$ ,SN 为食物源(解)的个数; $j=1,2,\cdots,D$ , $D$  为解的维数; $x_{\min}^j$ 和  $x_{\max}^j$ 分别为解的下界和上界.

雇佣蜂在初始解附近邻域搜索,寻找优良蜜源的位置,其更新策略为

$$v_i^j = x_i^j + \varphi_i^j(x_i^j - x_k^j). \tag{2}$$

式(2)中: $x_i^j$ 为第  $i$  个食物源的第  $j$  维; $\varphi_i^j$ 为 $[-1,1]$ 范围内的随机数, $i \in [1,\text{SN}]$ , $j \in [1,D]$ ;  $k$  为随机选择的整数, $k \in [1,\text{SN}]$ 且  $k \neq i$ . 采用贪婪选择机制根据适应度值选择的候选解  $V_i$  和当前解  $X_i$ .

对于最小化问题,适应度值的计算公式为

$$\text{fit}_i = \begin{cases} 1/(1 + f_i), & f_i \geq 0, \\ 1 + |f_i|, & f_i \leq 0. \end{cases} \tag{3}$$

式(3)中: $f_i$  为  $V_i$  对应的函数值, $f_i$  越小,则  $\text{fit}_i$  越大.

## 2 改进的人工蜂群算法

### 2.1 基于对数的适应度评价方式

在原始人工蜂群算法中,观察蜂通过概率选择优秀个体跟随,然后,进行深度搜索. 概率大小与雇佣蜂携带蜜源的质量密切相关,适应度值大小反映蜜源质量,适应度值越大,蜜源质量越好,被选择的概率就越大. 但是,式(3)中,当种群中解的精度差异较大,而概率值却相同,体现不出个体之间的差异. 通过概率选择更新可行解的作用消失,易使种群“早熟”,出现停滞现象<sup>[9]</sup>.

针对该问题,引入基于对数的适应度评价方式,增大差异,对解相似但不同的个体加以区分,使优秀个体有更大几率被跟随开采<sup>[10]</sup>. 则适应度评价方式为

$$\text{fitness}_i = \frac{0.1}{0.1 + 1/|\lg f_i|}, \quad 0 \leq f_i \leq 10^{-\epsilon}. \tag{4}$$

式(4)中: $\epsilon$  由计算机的计算精度决定. 此处, $\epsilon$  取 8.

### 2.2 基于增强开采能力策略

搜索能力的平衡是影响优化结果的因素之一. 由原始人工蜂群算法的搜索方程(2)可以看出:随机系数  $\varphi_i^j$ ,随机参数  $j,k$  对算法的开发能力有利,而不利于开采能力. 因此,需引入一种增强开采能力的策略平衡搜索能力. 受微分进化算法<sup>[8,11]</sup>的启发,引入一种新的搜索策略,即

$$v_i^j = x_{\text{best}}^j + \varphi_i^j(x_i^j - x_r^j).$$

(5)

式(5)中: $x_{\text{best}}^j$ 为当前种群中最优个体的第  $j$  维; $\varphi_i^j$  为  $[-1,1]$  内的随机数; $x_r^j$  为第  $r$  个蜜蜂的第  $j$  维, $r$  为  $\{1,2,\cdots,\text{SN}\}$  内的随机整数,且  $r\neq 1$ .

### 3 算法性能分析

#### 3.1 测试函数

采用 Elliptic 函数(F01), Sphere 函数(F02), Weierstrass 函数(F03), Ackley 函数(F04), Griewanks 函数(F05)和 Alpine 函数(F06)等 6 个基准测试函数进行实验,测试改进人工蜂群(BSABC)算法的性能.其中:F01~F02 为单峰函数;F03~F06 为多峰函数;理论最优值均为 0. Sphere 函数和 Elliptic 函数的搜索范围均为  $[-100,100]^D$ , Weierstrass 函数、Ackley 函数、Griewanks 函数和 Alpine 函数的搜索范围分别为  $[-0.5,0.5]^D$ ,  $[-32.768,32.768]^D$ ,  $[-600,600]^D$ ,  $[-10,10]^D$ . 测试函数详细介绍参考文献[12]. 部分测试函数的三维曲线图,如图 1 所示.

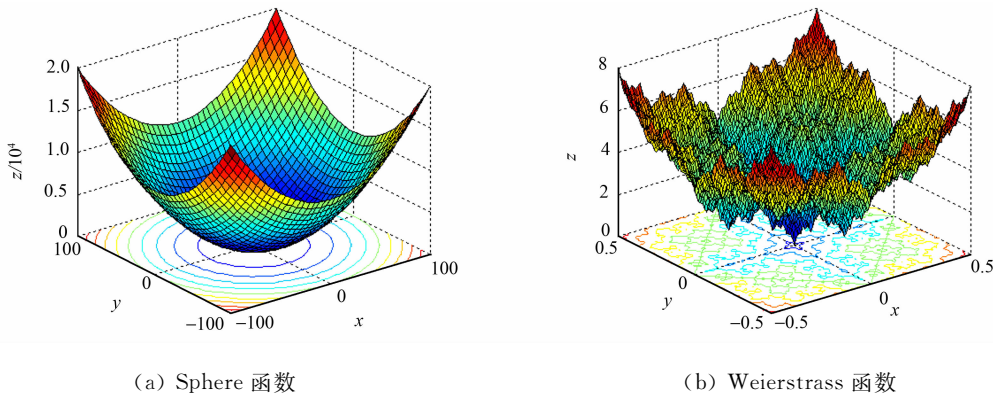


图 1 部分测试函数曲线  
Fig. 1 Partial test function curve

#### 3.2 实验分析

将所提 BSABC 算法与 ABC 算法<sup>[2]</sup>和 GABC 算法<sup>[3]</sup>进行比较. 设置参数如下:  $\text{SN}=40$ ,  $\text{limit}=100$ ,  $\text{MCN}=1\,000$ . 将测试函数分别采用这 3 个算法在相同的实验背景下运行,且每个测试函数独立运行 10 次以避免偶然性,并记录最优值、最差值、中值、平均值和方差.  $D=30$  的测试函数实验结果,如表 1 所示. 表 1 中:加粗字体表示 3 种算法中性能最优. 由表 1 可知:在 6 个测试函数中,BSABC 算法的解的精度最高,而且在解的稳定性方面性能最好. 特别地,由于开采能力的增强以基于对数的适应值函数减小选择压力,对测试函数 Weierstrass 和 Griewanks 性能最好.

表 1 测试函数的收敛实验结果( $D=30$ )

Tab. 1 Results of convergence of test functions ( $D=30$ )

函数	算法	最优值	最差值	中值	平均值	方差
Elliptic	ABC	$6.926\,66\times10^{-4}$	$1.132\,49\times10^{-2}$	$4.173\,08\times10^{-3}$	$4.889\,03\times10^{-3}$	$4.195\,18\times10^{-3}$
	GABC	$2.003\,53\times10^{-10}$	$7.211\,49\times10^{-10}$	$2.617\,17\times10^{-10}$	$3.532\,46\times10^{-10}$	$2.152\,08\times10^{-10}$
	BSABC	<b><math>5.195\,33\times10^{-16}</math></b>	<b><math>7.130\,02\times10^{-16}</math></b>	<b><math>5.510\,23\times10^{-16}</math></b>	<b><math>5.714\,88\times10^{-16}</math></b>	<b><math>8.060\,42\times10^{-17}</math></b>
Sphere	ABC	$2.219\,76\times10^{-10}$	$2.030\,98\times10^{-9}$	$5.130\,29\times10^{-10}$	$7.983\,28\times10^{-10}$	$7.168\,64\times10^{-10}$
	GABC	$9.376\,23\times10^{-16}$	$2.748\,01\times10^{-15}$	$2.093\,61\times10^{-15}$	$1.929\,22\times10^{-15}$	$6.931\,02\times10^{-16}$
	BSABC	<b><math>5.171\,34\times10^{-16}</math></b>	<b><math>7.590\,28\times10^{-16}</math></b>	<b><math>6.669\,92\times10^{-16}</math></b>	<b><math>6.430\,31\times10^{-16}</math></b>	<b><math>1.078\,97\times10^{-16}</math></b>
Weierstrass	ABC	$7.431\,64\times10^{-4}$	$9.037\,61\times10^{-4}$	$7.953\,40\times10^{-4}$	$8.206\,65\times10^{-4}$	$6.509\,59\times10^{-5}$
	GABC	$1.210\,69\times10^{-5}$	$3.634\,44\times10^{-5}$	$1.720\,74\times10^{-5}$	$1.953\,72\times10^{-5}$	$9.646\,90\times10^{-6}$
	BSABC	<b><math>1.342\,93\times10^{-12}</math></b>	<b><math>1.013\,23\times10^{-11}</math></b>	<b><math>7.759\,13\times10^{-12}</math></b>	<b><math>6.825\,47\times10^{-12}</math></b>	<b><math>3.305\,60\times10^{-12}</math></b>
Ackley	ABC	$2.512\,45\times10^{-5}$	$6.309\,58\times10^{-5}$	$3.297\,00\times10^{-5}$	$4.209\,28\times10^{-5}$	$1.769\,01\times10^{-5}$
	GABC	$4.436\,74\times10^{-8}$	$1.099\,10\times10^{-7}$	$7.470\,67\times10^{-8}$	$7.307\,51\times10^{-8}$	$2.696\,41\times10^{-8}$
	BSABC	<b><math>1.411\,85\times10^{-11}</math></b>	<b><math>5.521\,98\times10^{-11}</math></b>	<b><math>2.851\,76\times10^{-11}</math></b>	<b><math>3.203\,34\times10^{-11}</math></b>	<b><math>1.644\,40\times10^{-11}</math></b>

续表  
Conitune table

函数	算法	最优值	最差值	中值	平均值	方差
Griewanks	ABC	$4.959\ 13\times10^{-9}$	$8.824\ 75\times10^{-7}$	$1.919\ 75\times10^{-7}$	$2.712\ 98\times10^{-7}$	$3.608\ 93\times10^{-7}$
	GABC	$1.200\ 61\times10^{-11}$	$5.282\ 80\times10^{-6}$	$8.831\ 02\times10^{-10}$	$1.064\ 73\times10^{-6}$	$2.358\ 03\times10^{-6}$
	BSABC	<b>0</b>	<b><math>3.173\ 31\times10^{-10}</math></b>	<b><math>2.884\ 91\times10^{-12}</math></b>	<b><math>6.520\ 31\times10^{-11}</math></b>	<b><math>1.409\ 65\times10^{-10}</math></b>
Alpine	ABC	$3.174\ 84\times10^{-4}$	$3.880\ 31\times10^{-3}$	$7.976\ 60\times10^{-4}$	$1.703\ 19\times10^{-3}$	$1.564\ 93\times10^{-3}$
	GABC	$4.023\ 64\times10^{-5}$	$1.759\ 90\times10^{-4}$	$9.330\ 71\times10^{-5}$	$9.591\ 62\times10^{-5}$	$5.649\ 88\times10^{-5}$
	BSABC	<b><math>1.022\ 67\times10^{-12}</math></b>	<b><math>4.721\ 22\times10^{-12}</math></b>	<b><math>2.007\ 41\times10^{-12}</math></b>	<b><math>2.409\ 71\times10^{-12}</math></b>	<b><math>1.486\ 25\times10^{-12}</math></b>

为了更直观地观察算法的寻优过程,给出测试函数( $D=30$ )的收敛曲线,如图 2 所示.图 2 中: $\bar{e}$  为平均误差值; $N$  为迭代次数.

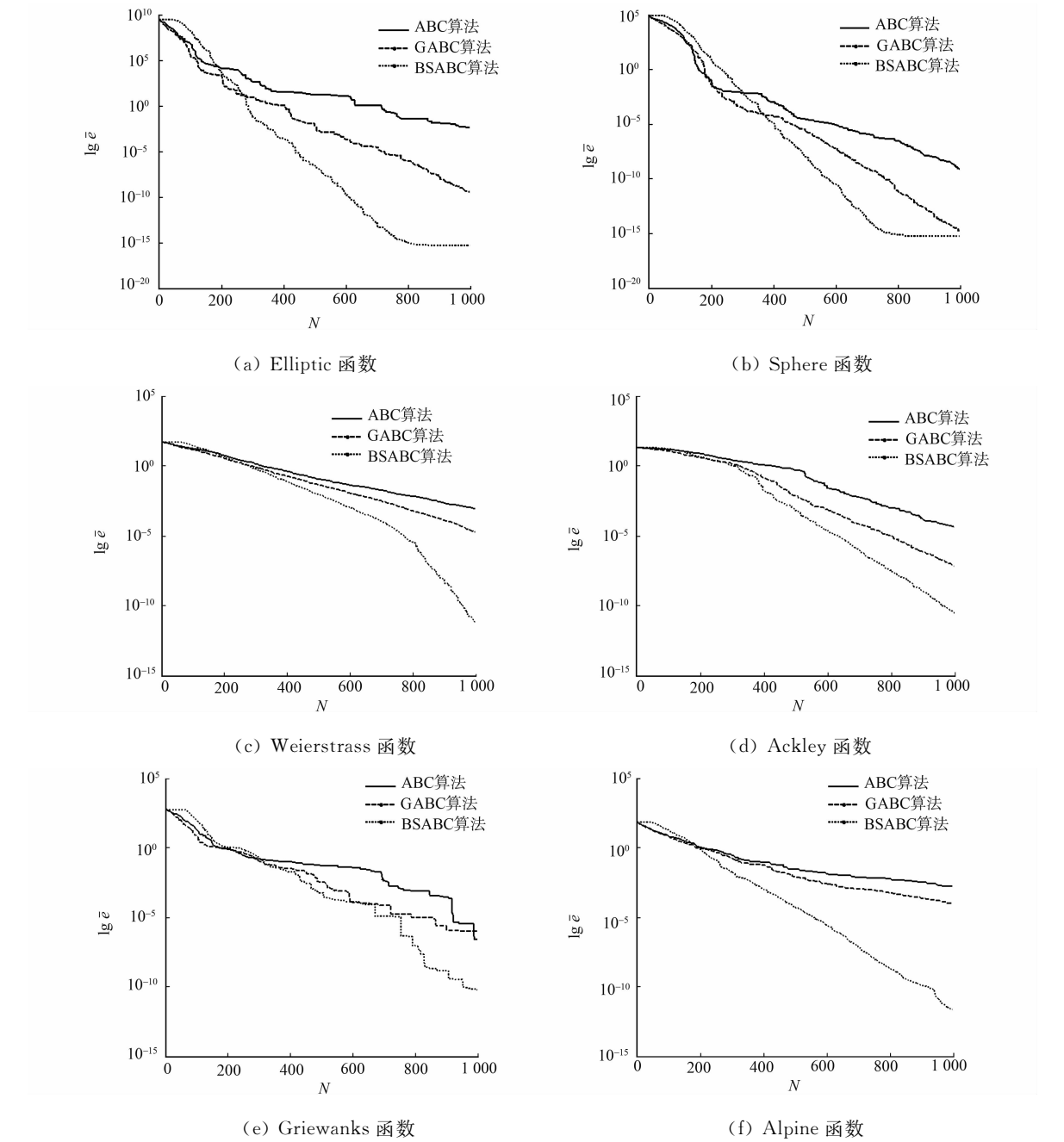


图 2 测试函数的收敛曲线( $D=30$ )

Fig. 2 Convergence curve of test function ( $D=30$ )

由图 2 可知:对于 Elliptic 函数,BSABC 算法在收敛精度和收敛速度方面均性能最优;对于 Sphere

函数,BSABC 具有较快的收敛速度,但是搜索后期 BSABC 算法陷入停滞状态;对于 Weierstrass 函数和 Griewanks 函数,BSABC 算法不仅有较高收敛精度,而且有较快的收敛速度;对于 Ackley 函数,BSABC 算法的收敛曲线几乎呈线性下降,这表明 BSABC 能够跳出局部最优,防止早熟;对于 Alpine 函数,虽然 BSABC 算法在进化前期收敛速度比 ABC 算法和 GABC 算法慢,但进化后期收敛速度明显快于另外两种算法快陷入局部最优。

综上所述,无论对于单峰函数,还是多峰函数,BSABC 算法都表现出良好的性能。

## 4 结束语

针对基本 ABC 算法局部搜索能力弱的问题<sup>[13-14]</sup>及类似算法的研究<sup>[15]</sup>,提出一种平衡搜索能力的人工蜂群算法.通过 6 个测试函数的仿真实验表明,所提算法可以提高优化解的精度和速度.但该算法也存在不足,对于复杂优化问题,效果不是很好.因此,如何使算法能够解决复杂问题,是下一步的研究工作。

## 参考文献:

- [1] KARABOGA D. An idea based on honey bee swarm for numerical optimization: Technical report-TR06[R]. [S.l.]: [s. n.], 2005:1-10.
- [2] KARABOGA D, BASTURK B. On the performance of artificial bee colony (ABC) algorithm[J]. Applied Soft Computing, 2008, 8(1): 687-697. DOI:10.1016/j.asoc.2007.05.007.
- [3] ZHU Guopu, KWONG S. Gbest-guided artificial bee colony algorithm for numerical function optimization[J]. Applied Mathematics and Computation, 2010, 217(7): 3166-3173. DOI:10.1016/j.amc.2010.08.049.
- [4] BANHARNSAKUN A, ACHALAKUL T, SIRINAOVAKUL B. The best-so-far selection in artificial bee colony algorithm[J]. Applied Soft Computing, 2011, 11(2): 2888-2901. DOI:10.1016/j.asoc.2010.11.025.
- [5] 高卫峰, 刘三阳, 黄玲玲. 受启发的人工蜂群算法在全局优化问题中的应用[J]. 电子学报, 2012, 40(12): 2396-2403. DOI:10.3969/j.issn.0372-2112.2012.12.007.
- [6] 宁爱平, 张雪英. 人工蜂群算法的收敛性分析[J]. 控制与决策, 2013, 28(9): 1554-1558.
- [7] LI Junqing, PAN Quanke, TASGETIREN M F. A discrete artificial bee colony algorithm for the multi-objective flexible job-shop scheduling problem with maintenance activities[J]. Applied Mathematical Modelling, 2014, 38(3): 1111-1132. DOI:10.1016/j.apm.2013.07.038.
- [8] KIRAN M S, HAKLI H, GUNDUZ M, *et al.* Artificial bee colony algorithm with variable search strategy for continuous optimization[J]. Information Sciences, 2015, 300: 140-157. DOI:10.1016/j.ins.2014.12.043.
- [9] GAO Weifeng, LIU Sanyang, HUANG Lingling. Enhancing artificial bee colony algorithm using more information-based search equations[J]. Information Sciences, 2014, 270(1): 112-133. DOI:10.1016/j.ins.2014.02.104.
- [10] 陈杰, 沈艳霞, 陆欣. 基于信息反馈和改进适应度评价的人工蜂群算法[J]. 智能系统学报, 2016, 11(2): 172-179. DOI:10.11992/tis.201506024.
- [11] YI Wenchao, GAO Liang, ZHOU Yinzi, *et al.* Differential evolution algorithm with variable neighborhood search for hybrid flow shop scheduling problem[C]//International Conference on Computer Supported Cooperative Work in Design. [S.l.]: IEEE Press, 2016: 233-238. DOI:10.1109/CSCWD.2016.7565994.
- [12] SUGANTHAN P N, HANSEN N, LIANG J J, *et al.* Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization[R]. Singapore: Nanyang Technological University, 2005: 341-357.
- [13] 王志刚, 王明刚. 基于符号函数的多搜索策略人工蜂群算法[J]. 控制与决策, 2016, 31(11): 2037-2044. DOI:10.13195/j.kzyjc.2015.1046.
- [14] 秦全德, 程适, 李丽, 等. 人工蜂群算法研究综述[J]. 智能系统学报, 2014, 9(2): 127-135. DOI:10.3969/j.issn.1673-4785.201309064.
- [15] 柴文光. CPSO 支持向量机红外瓦斯传感器动态补偿[J]. 华侨大学学报(自然科学版), 2016, 37(3): 316-319. DOI:10.11830/ISSN.1000-5013.2016.03.0316.

(责任编辑: 钱筠 英文审校: 吴逢铁)