

## 基于排序选择和精英引导的改进人工蜂群算法

孔德鹏, 常天庆<sup>†</sup>, 戴文君, 王全东, 孙皓泽

(陆军装甲兵学院 控制工程系, 北京 100072)

**摘要:** 针对人工蜂群算法收敛速度较慢、收敛精度不高的问题, 提出一种基于排序选择和精英引导的改进人工蜂群算法. 分析观察蜂概率选择方法在适应值变化时对于精英个体优选的不足, 提出一种排序选择方法, 用以替代概率选择方法, 从而提高算法的收敛速度. 利用精英个体对搜索的引导作用, 分别提出针对采蜜蜂和观察蜂的改进邻域搜索方程, 从而提高算法的搜索效率. 与其他人工蜂群算法的对比结果表明, 所提出的改进方法能够有效提升算法的收敛速度和收敛精度.

**关键词:** 人工蜂群算法; 排序选择; 精英引导; 搜索方程

**中图分类号:** TP18

**文献标志码:** A

### An improved artificial bee colony algorithm based on the ranking selection and the elite guidance

KONG De-peng, CHANG Tian-qing<sup>†</sup>, DAI Wen-jun, WANG Quan-dong, SUN Hao-ze

(Department of Control Engineering, Academy of Army Armored Forces, Beijing 100072, China)

**Abstract:** In order to solve the problem of low convergence speed and low convergence accuracy of an artificial bee colony algorithm, an improved artificial bee colony algorithm based on ranking selection and elite guidance is proposed. The probability selection method of onlooker bees is weak to select the elite individual when the fitness value is changing, therefore, a ranking selection method is proposed to replace that of probability selection for improving the convergence speed of the algorithm. To improve the search efficiency, two new neighborhood search equations for employed bees and onlooker bees respectively are proposed by using the elite guidance. Compared with other artificial bee colony algorithms, the results show that the proposed algorithm can effectively improve the convergence speed and convergence accuracy.

**Keywords:** artificial bee colony algorithm; ranking selection; elite guidance; search equation

## 0 引言

人工蜂群(Artificial bee colony, ABC)算法是一种模拟蜜蜂采蜜行为的群智能优化方法,是由土耳其学者 Karaboga 于 2005 年提出的,同时也是目前最为有效的优化算法之一<sup>[1]</sup>. 人工蜂群算法具有寻优效果好、控制参数少、实现简单等特点<sup>[2]</sup>. 与粒子群(PSO)算法<sup>[3]</sup>、遗传算法(GA)<sup>[4]</sup>、蚁群(ACO)算法<sup>[5]</sup>和差分进化(DE)算法<sup>[6]</sup>相比,ABC算法的寻优能力和算法精度都具有明显优势<sup>[7]</sup>. 人工蜂群算法自提出以后,受到了广泛关注,已应用在云服务组合<sup>[8]</sup>、功率流优化<sup>[9]</sup>、时-费-效权衡<sup>[10]</sup>、动态聚类<sup>[11]</sup>、最短路径问题(SP)<sup>[12]</sup>等多个方面.

虽然人工蜂群算法相比于其他一些智能计算方

法具有一定优势,但算法仍存在食物源开发与开采之间的矛盾,收敛速度较慢,局部求精能力较弱. 因此,众多学者针对人工蜂群算法存在的不足提出了一系列改进方法. Zhu 等<sup>[13]</sup>受 PSO 算法的启发提出了基于最优值引导的人工蜂群(GABC)算法,算法在邻域搜索公式中加入了全局最优 Gbest 的引导,提高了收敛速度; Gao 等<sup>[14]</sup>利用改进的邻域搜索方程和正交学习方法提出了改进人工蜂群(CABC)算法,提升了 ABC 算法的寻优能力; Karaboga 等<sup>[15]</sup>提出一种 ABC 的改进版算法,即快速人工蜂群(qABC)算法,该算法改进了观察蜂的搜索公式,加快了算法的收敛速度; Gao 等<sup>[16]</sup>受差分进化算法的启发,提出了一种改进的 ABC 算法,该算法利用搜索方程“ABC/best/1”

收稿日期: 2017-10-11; 修回日期: 2017-11-13.

基金项目: 军队院校创新工程项目(2015YY05).

责任编辑: 林崇.

作者简介: 孔德鹏(1990—),男,博士生,从事火力运用决策优化技术的研究;常天庆(1963—),男,教授,博士生导师,从事火控系统及其智能化技术等研究.

<sup>†</sup>通讯作者. E-mail: changtianqing@263.net.

和“ABC/rand/1”提高了算法的收敛速度;Li等<sup>[17]</sup>将遗传算法的基因重组引入ABC算法中,提出了基于基因重组的人工蜂群(GRABC)算法,提高了全局寻优能力;Gao等<sup>[18]</sup>研究了一种搜索方式,即算法只在最优解附近进行邻域搜索,并提出了改进人工蜂群(MABC)算法.上述这些改进方式使得ABC算法的性能有所提升,但对于实际优化问题仍需要更高性能的寻优算法,人工蜂群算法的收敛速度、收敛精度等仍有待进一步提高.

本文主要针对人工蜂群算法进行改进,旨在提高算法的精度和收敛速度.通过排序选择方法使得蜂群的搜索效率不受适应度值的影响,从而提高收敛速度;同时,采用基于精英引导的改进邻域搜索方程,既可保证算法快速收敛,也可获得较高的收敛精度.

## 1 人工蜂群算法

人工蜂群由3种蜂组成,即采蜜蜂(Employed bees, EB)、观察蜂(Onlooker bees, OB)和侦察蜂(Scout bees, SB).采蜜蜂在觅食过程中将食物源信息共享给观察蜂,观察蜂根据食物源的质量选择自己的食物源.当一个食物源枯竭时,相应的蜜蜂将舍弃这个食物源,变成侦察蜂,重新随机选择一个食物源.在人工蜂群算法中,采蜜蜂和观察蜂的种群数量都为SN,每个蜜蜂所在的食物源位置为优化问题的解,食物源的质量为解的评价值.

### 1.1 初始化

每个食物源的位置用 $D$ 维向量表示,即

$$X_i = [x_{i1}, x_{i2}, \dots, x_{iD}], \quad i = 1, 2, \dots, SN.$$

食物源的初始位置随机产生,解空间的上限为

$$UB = [UB_1, UB_2, \dots, UB_D],$$

解空间的下限为

$$LB = [LB_1, LB_2, \dots, LB_D],$$

则初始的食物源位置(即初始解)为

$$x_{id} = LB_d + (UB_d - LB_d) \cdot \text{rand}(0, 1). \quad (1)$$

其中: $i = 1, 2, \dots, SN$ ;  $d = 1, 2, \dots, D$ ;  $\text{rand}(0, 1)$ 是 $[0, 1]$ 上均匀分布的随机数.

### 1.2 采蜜蜂

每一个食物源 $x_i$ 对应一个采蜜蜂,每个采蜜蜂随机选择一个邻居进行邻域搜索来更新食物源,获得的新食物源为 $v_i$ ,则有

$$v_{id} = x_{id} + \phi_{id}(x_{id} - x_{kd}). \quad (2)$$

其中: $i = 1, 2, \dots, SN$ ;  $d = 1, 2, \dots, D$ ;  $\phi_{id}$ 是 $[-1, 1]$ 上均匀分布的随机数; $k \in \{1, 2, \dots, SN\}$ ,  $k \neq i$ 是随机选择的一个邻居.获得新食物源后,按照贪婪选择的方式对之前的食物源进行更新,如果新的食物源 $v_i$ 的质量高于原食物源 $x_i$ ,则 $v_i$ 取代 $x_i$ ;否则 $x_i$ 保持不变,迭代重复值 $\text{trial}(i)$ 加1.迭代重复值 $\text{trial}(i)$ 表示经过多次搜索,食物源质量也没有得到改善.

### 1.3 概率选择

当所有采蜜蜂进行邻域搜索后,将所在食物源信息共享给观察蜂,观察蜂根据各食物源的质量进行概率选择.第 $i$ 个食物源被观察蜂选择的概率为

$$p_i = \text{fit}_i / \sum_{j=1}^{SN} \text{fit}_j. \quad (3)$$

食物源的适应度值 $\text{fit}_i$ 按下式计算:

$$\text{fit}_i = \begin{cases} 1/(1 + f_i), & f_i \geq 0; \\ 1 + |f_i|, & \text{otherwise.} \end{cases} \quad (4)$$

其中: $f_i$ 是第 $i$ 个食物源的评价值,由所求解问题的目标函数计算得到.

### 1.4 观察蜂

观察蜂选择一个食物源后,与采蜜蜂相同,按照邻域搜索的方式更新食物源位置,即

$$v_{id} = x_{id} + \phi_{id}(x_{id} - x_{kd}). \quad (5)$$

### 1.5 侦察蜂

当采蜜蜂和观察蜂多次搜索同一食物源却没有获得更好的食物源时,放弃该食物源,变为侦察蜂,并随机搜索一个新食物源 $x_i$ 替代该食物源.

$$x_{id} = LB_d + (UB_d - LB_d) \cdot \text{rand}(0, 1). \quad (6)$$

## 2 改进人工蜂群算法

基于排序选择和精英引导策略,本文提出一种改进人工蜂群算法(REABC).排序选择策略是一种基于排序位置的概率选择方式,克服了基于适应度概率选择在迭代后期对于精英个体选择能力的不足.利用精英引导的方法分别对采蜜蜂和观察蜂的邻域搜索方程进行改进,以提高搜索效率,使得算法快速收敛.

### 2.1 排序选择策略

#### 2.1.1 基于适应度的概率选择方法的局限性

在ABC算法中,观察蜂以一定的概率选择采蜜蜂的食物源作为自己的食物源,选择概率由采蜜蜂的适应值计算得出,如式(3)所示.但在适应值差别较小时,概率选择方法并不能对精英个体进行优选.

在利用式(4)进行适应值计算的过程中,在函数值差别较大时,较差的个体被选择的概率很小(如 $EB_1$ 为0.007),较优的个体被选择的概率很大(如 $EB_5$ 为0.6017),从而使得算法容易陷入局部极值;当获得的结果精度越来越高,函数值差距较小时,不同个体适应值的差别越来越小,因此计算出的采蜜蜂被选择的概率基本相同,等同于随机选择,这使得算法的收敛能力下降.

表1 基于适应度的概率选择方法

参数	$EB_1$	$EB_2$	$EB_3$	$EB_4$	$EB_5$
评价值	$10^3$	$10^2$	$10^1$	$10^0$	$10^{-1}$
适应度值	0.0010	0.0099	0.0909	0.5000	0.9091
选择概率	0.0007	0.0066	0.0602	0.3309	0.6017
评价值	$10^{-4}$	$10^{-5}$	$10^{-6}$	$10^{-7}$	$10^{-8}$
适应度值	0.9999	1.0000	1.0000	1.0000	1.0000
选择概率	0.2000	0.2000	0.2000	0.2000	0.2000

2.1.2 排序选择

由于ABC算法后期对优势个体的选择能力降低,本文采用排序选择方法,按照个体的评价值进行排序,每个排序位置的选择概率固定,使得优势个体被选择的概率不受适应度值变化的影响,从而提高算法的寻优能力.设有SN个采蜜蜂 $X_1, X_2, \dots, X_{SN}$ ,对应的评价函数值为 $f(X_1), f(X_2), \dots, f(X_{SN})$ ,根据其评价值进行排序,可以获得按照评价值排序的序列 $X_{R(1)}, X_{R(2)}, \dots, X_{R(SN)}$ ,其中 $R(i)$ 表示按评价值排序为 $i$ 的个体的原序列排序号.选择反比例函数构造基于排序的选择策略,个体排序越靠前,被选择的概率越大.

$$p_i = \frac{1/i}{\sum_{i=1}^{SN} 1/i}, i = 1, 2, \dots, SN. \quad (7)$$

排序靠后的个体也有一定的被选择概率,如表2所示.

表2 基于适应度的概率选择方法

参数	$EB_1$	$EB_2$	$EB_3$	$EB_4$	$EB_5$
评价值	$10^{-4}$	$10^{-5}$	$10^{-6}$	$10^{-7}$	$10^{-8}$
评价值排序	5	4	3	2	1
排序选择概率	0.0876	0.1095	0.1460	0.2190	0.4380

通过这种方法提高了算法跳出局部极值的能力,保证了算法的全局搜索效能.

2.2 基于精英引导的改进邻域搜索方程

传统的邻域搜索方程如式(2)和(5)所示,利用随机选择的一个邻居进行搜索,这种方式具有较好的全

局搜索能力,但是没有对当前找到的最优解进行充分开发,局部优化能力较弱,总体的搜索效率较低,本文利用精英引导提高搜索效率.

2.2.1 采蜜蜂

本文针对采蜜蜂的邻域搜索方程进行改进,利用精英个体的引导使种群快速收敛.选择种群中评价值最好的一部分个体组成精英群体,精英的数量表示为

$$T = \text{ceil}(p \times SN). \quad (8)$$

其中: $p$ 表示精英个体占种群个体的比例, $\text{ceil}(\cdot)$ 表示向上取整函数.

采蜜蜂利用精英个体的引导,并以随机选择的邻居作为搜索中心,相对于ABC算法仅对自身邻域进行搜索,扩大了搜索范围,加快了搜索速度.对于第 $i$ 个采蜜蜂,其新的食物源为

$$v_{id} = x_{kd} + \phi_{id}(x_{rd}^e - x_{kd}). \quad (9)$$

其中: $d = 1, 2, \dots, D; x_k$ 是随机选取的一个邻居, $k \in \{1, 2, \dots, SN\}, k \neq i; x_r^e$ 是随机选择的精英个体, $r \in \{1, 2, \dots, T\}$ .

2.2.2 观察蜂

传统的人工蜂群中观察蜂以一定的概率选择采蜜蜂的食物源作为自己的食物源,其中质量好的食物源被选择的概率较大.然后利用式(5)所示的邻域搜索方程进行邻域搜索,寻找新的食物源.由于选择概率受适应值的影响,算法后期的收敛速度变慢,影响了算法的性能.

蜜蜂采蜜过程中一般都会向食物源质量好的地方搜索,这样可以有效提高搜索到的食物源的质量.因此,根据这个特征,本文提出一种新的观察蜂邻域搜索方程,即

$$v_{id} = x_{id} + \phi_{id}(x_{id} - x_{hd}). \quad (10)$$

其中: $d = 1, 2, \dots, D; x_h$ 是根据采蜜蜂的排序进行概率选择的一个精英, $h \in \{1, 2, \dots, SN\}$ .

观察蜂以基于排序选择的食物源为起点,向着另一个基于排序选择的食物源进行邻域搜索.与式(9)不同的是,观察蜂通过引入基于排序选择的精英个体进行邻域搜索,加快了收敛速度,同时对较差个体也保留一定的选择概率,防止陷入局部极值.

2.3 REABC算法

由2.1节和2.2节的改进方法可得REABC算法的伪代码如表3所示.

表3 REABC算法伪代码

```

01: Initialization: 按照式(1)随机生成初始种群;
02: 根据式(7)计算基于排序的选择概率;
03: while FES < Max FES
04:   for i=1: SN // employed bees phase
05:     从种群中随机选择一个邻居  $x_k$ ;
06:     从精英群体中随机选择一个精英;
07:     根据式(9)计算新的食物源  $v_i$ ;
08:     计算  $v_i$  的评价值  $f(v_i)$ ;
09:     if  $f(v_i) < f(x_i)$ 
10:       Set  $x_i = x_i, f(x_i) = f(v_i), trial(i)=0$ ;
11:     else
12:        $trial(i) = trial(i)+1$ ;
13:     end if
14:   end for //end employed bees phase
15:   FES = FES + SN;
16:   根据每个个体的评价值进行排序;
17:   Set  $t = 0$ ;
18:   while  $t < SN$  //onlooker bees phase
19:     根据排序选择策略选择一个蜜蜂  $x_i$ ;
20:     根据排序选择策略选择一个精英  $x_h$ ;
21:     根据式(10)计算新的食物源  $v_i$ ;
22:     计算  $v_i$  的评价值  $f(v_i)$ ;
23:     if  $f(v_i) < f(x_i)$ 
24:       Set  $x_i = x_i, f(x_i) = f(v_i), trial(i)=0$ ;
25:     else
26:        $trial(i) = trial(i)+1$ ;
27:     end if
28:      $t = t + 1$ ;
29:   end while //end onlooker
30:   FES = FES + SN;
31:   记录当前的最优解  $x$ ;
32:   if  $trial(i) > limit$  //scout bee phase
33:     根据式(6)随机生成一个解替换  $x_i$ ;
34:      $trial(i) = 0, FES = FES + 1$ ;
35:   end if //end scout bee phase
36: end while
37: Output: 全局最优解  $x$ 
    
```

2.4 算法复杂度分析

REABC算法相对于ABC算法多了一个排序过程,排序复杂度为 $O(SN \cdot \log(SN))$ ,而ABC算法的复杂度为 $O(SN \cdot D)^{[19]}$ ,因此,REABC算法的复杂度为 $O(SN \cdot D + SN \cdot \log(SN))$ .由于 $O(\log(SN)) \ll O(D)$ ,EAABC算法的复杂度可以表示为 $O(SN \cdot D)$ ,即相对于传统ABC算法,REABC算法具有相同的计算复杂度.此外,REABC算法取消了适应度值计算的环节,且排序的选择概率是定值,不需要每次迭代循环时都进行计算,这些改进可使算法的运行时间减少.

3 实验验证

利用22个标准测试函数<sup>[17]</sup>对所提出的REABC算法进行验证. $f_1 \sim f_6$ 和 $f_8$ 是连续单峰函数, $f_7$ 是非连续阶跃函数, $f_{11} \sim f_{22}$ 是连续多峰函数, $f_{10}$ 在 $D \leq 3$ 时是单峰函数,在 $D > 3$ 时是多峰函数.

3.1 几种ABC算法迭代结果对比

选择ABC<sup>[2]</sup>、GABC<sup>[13]</sup>、CABC<sup>[14]</sup>、MABC<sup>[18]</sup>、qABC<sup>[15]</sup>、GRABC<sup>[17]</sup>这6种代表性算法作为对比算

法.为了保证算法对比的公平性,本文按照ABC算法进行设置:种群 $SN = 50$ ;求解问题的维数 $D = 30$ ;截止条件 $FES = 5000 \times D$ ;FES表示评价函数的评估次数,即执行 $f(x)$ 的次数;最大限制重复次数 $limit = D \times SN$ .其他参数设置如下:GABC中 $C = 1.5$ ,MABC中 $P = 0.7$ ,qABC中 $r = 1.5$ ,GRABC中 $p = 0.1$ ,REABC中 $p = 0.1$ .分别计算7种算法在22个测试函数中的收敛关系,算法获得的最优值随迭代次数关系的收敛曲线如图1~图11所示.

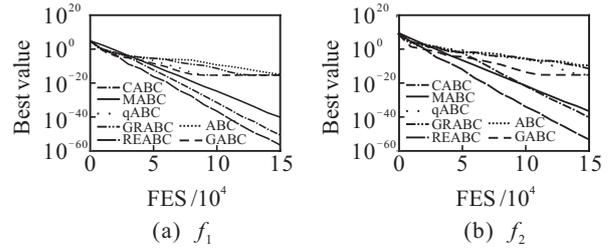


图1 函数 $f_1$ 和函数 $f_2$

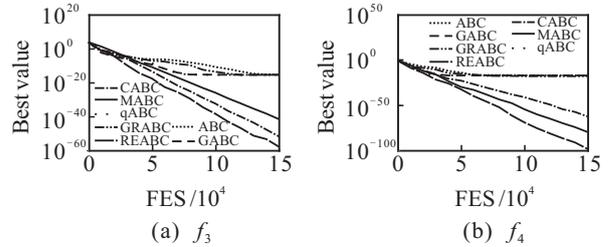


图2 函数 $f_3$ 和函数 $f_4$

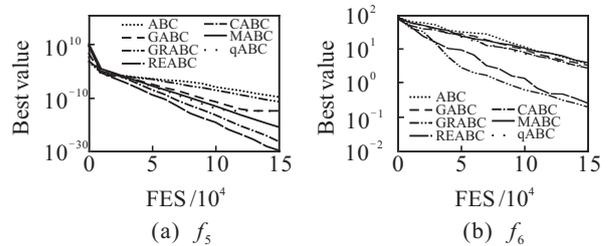


图3 函数 $f_5$ 和函数 $f_6$

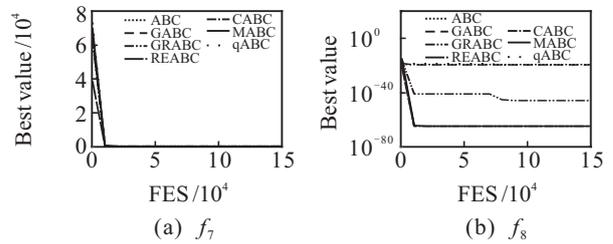


图4 函数 $f_7$ 和函数 $f_8$

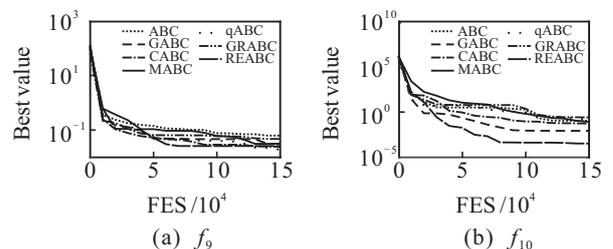


图5 函数 $f_9$ 和函数 $f_{10}$

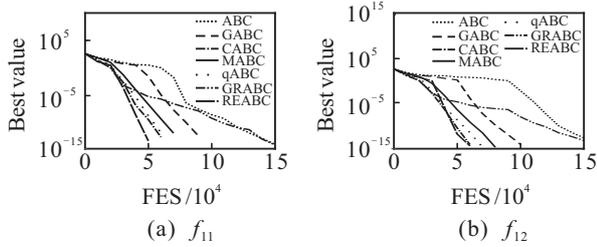


图6 函数  $f_{11}$  和函数  $f_{12}$

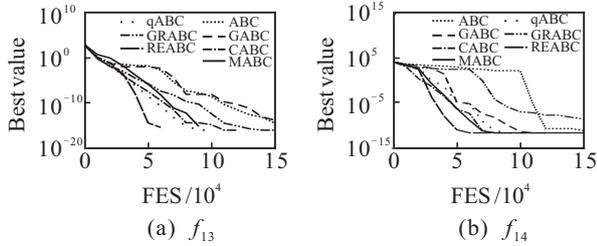


图7 函数  $f_{13}$  和函数  $f_{14}$

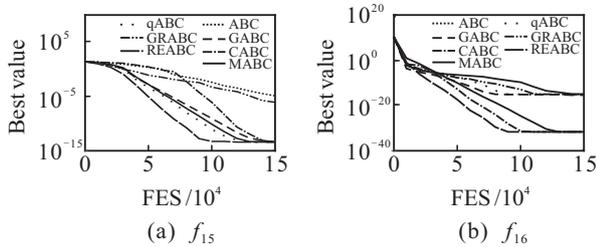


图8 函数  $f_{15}$  和函数  $f_{16}$

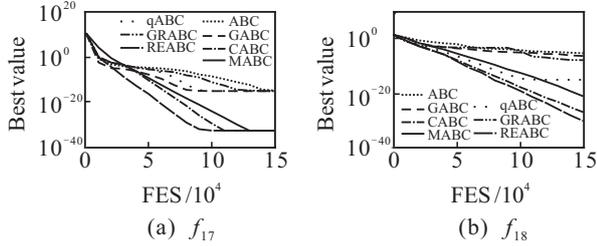


图9 函数  $f_{17}$  和函数  $f_{18}$

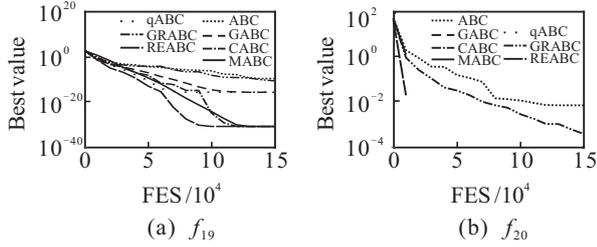


图10 函数  $f_{19}$  和函数  $f_{20}$

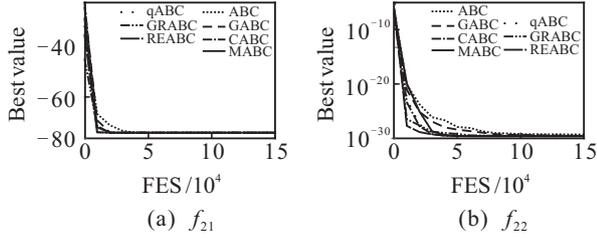


图11 函数  $f_{21}$  和函数  $f_{22}$

由图1~图11可以看出, REABC算法收敛速度较快,且能够获得比较优秀的解。相比于其他算法,除

测试函数  $f_6$  和  $f_9$  外, REABC算法都能够获得最好的寻优结果,说明了 REABC算法具有较好的收敛性能。

3.2 几种ABC算法寻优结果排序

同样与3.1节的6种算法进行对比,分别在22个测试函数上测试其收敛精度,在  $D = 30$  的情况下进行实验对比验证。算法独立运行25次,记录每次的最优值,根据22个测试函数的25次独立运行的结果给出基于Friedman非参数检验的算法综合性能排序,如表4所示。

表4 基于Friedman非参数检验的算法性能排序

	ABC	GABC	CABC	MABC	qABC	GRABC	REABC
$f_1$	6.64	4.80	2.00	3.00	4.64	5.92	1.00
$f_2$	7.00	4.28	2.00	3.00	4.72	6.00	1.00
$f_3$	6.60	5.68	2.00	3.00	5.00	4.72	1.00
$f_4$	6.36	6.32	3.00	2.00	5.04	4.28	1.00
$f_5$	7.00	4.64	2.00	3.00	4.36	6.00	1.00
$f_6$	6.96	4.12	3.36	5.92	4.64	1.36	1.64
$f_7$	4.00	4.00	4.00	4.00	4.00	4.00	4.00
$f_8$	6.80	5.80	2.00	2.00	5.40	4.00	2.00
$f_9$	7.00	4.12	1.80	4.04	4.76	4.00	2.28
$f_{10}$	4.04	3.40	5.32	4.52	3.64	4.36	2.72
$f_{11}$	6.38	3.04	3.04	3.04	3.04	6.42	3.04
$f_{12}$	6.44	3.00	3.00	3.00	3.00	6.56	3.00
$f_{13}$	6.74	4.26	2.78	2.44	4.36	4.98	2.44
$f_{14}$	5.94	3.04	2.60	3.72	3.50	6.50	2.70
$f_{15}$	7.00	4.08	2.02	4.34	2.56	6.00	2.00
$f_{16}$	6.52	5.40	2.00	2.00	5.08	5.00	2.00
$f_{17}$	6.80	4.76	2.00	2.00	4.96	5.48	2.00
$f_{18}$	6.80	5.96	1.80	2.72	4.00	5.24	1.48
$f_{19}$	6.36	4.20	3.32	1.78	4.40	6.20	1.74
$f_{20}$	6.92	3.02	3.02	3.02	3.02	5.98	3.02
$f_{21}$	3.84	4.40	3.02	6.62	3.84	3.14	3.14
$f_{22}$	7.00	5.88	3.12	1.08	3.80	4.16	2.96
Ave	6.32	4.46	2.69	3.19	4.17	5.01	2.14

由表4可以看出, REABC的寻优性能最好。

3.3 几种ABC算法的运行时间对比

为了验证算法的时间复杂度,计算每种算法在相同测试条件下的运行时间,算法参数设置与3.1节相同。在22个测试函数上运行的平均耗时结果如表5所示。

表5 算法运行时间对比

	ABC	GABC	CABC	MABC	qABC	GRABC	REABC
$T$	2.04	2.05	2.05	2.05	3.09	2.25	1.99

由表5可以看出,几种算法的时间消耗差距不大,与理论分析的结果一致,本文的REABC算法的时间消耗最小,与取消适应度计算和概率计算有关。该结果说明,本文方法在不增加计算量的情况下能够有效提升算法的寻优效果。

## 4 结论

针对人工蜂群算法存在收敛速度慢,寻优精度不高的问题,提出了一种改进的人工蜂群算法.采用排序选择方法以基于位置的固定概率进行选择,克服了基于适应度的概率选择方式对优秀个体选择能力较弱的不足.利用精英引导改进邻域搜索方程(采蜜蜂通过对精英个体与邻居的搜索更新食物源,观察蜂利用排序选择策略选取两个优秀个体进行更新食物源)提高了算法的搜索效率和收敛精度.在标准测试函数上的对比测试表明,本文提出的REABC算法具有最好的综合性能,从而验证了改进算法的有效性.

### 参考文献(References)

- [1] Rajasekhar A, Lynn N, Das S, et al. Computing with the collective intelligence of honey bees — A survey[J]. *Swarm and Evolutionary Computation*, 2017, 32(2): 25-48.
- [2] Karaboga D, Basturk B. A powerful and efficient algorithm for numerical function optimization: Artificial bee colony(ABC) algorithm[J]. *J of Global Optimization*, 2007, 39(3): 459-471.
- [3] Yu K, Wang X, Wang Z. Multiple learning particle swarm optimization with space transformation perturbation and its application in ethylene cracking furnace optimization[J]. *Knowledge-Based Systems*, 2016, 96(3): 156-170.
- [4] Chuang Y C, Chen C T, Hwang C. A real-coded genetic algorithm with a direction-based crossover operator[J]. *Information Sciences*, 2015, 305(6): 320-348.
- [5] Yang Q, Chen W N, Yu Z T, et al. Adaptive multimodal continuous ant colony optimization[J]. *IEEE Trans on Evolutionary Computation*, 2017, 21(2): 191-205.
- [6] Cheng J, Zhang G, Caraffini F, et al. Multicriteria adaptive differential evolution for global numerical optimization[J]. *Integrated Computer-Aided Engineering*, 2015, 22(2): 103-107.
- [7] Karaboga D, Gorkemli B, Ozturk C, et al. A comprehensive survey: Artificial bee colony(ABC) algorithm and applications[J]. *Artificial Intelligence Review*, 2012, 42(1): 21-57.
- [8] Huo Y, Zhuang Y, Gu J, et al. Discrete gbest-guided artificial bee colony algorithm for cloud service composition[J]. *Applied Intelligence*, 2014, 42(4): 661-678.
- [9] Bai W, Eke I, Lee K Y. An improved artificial bee colony optimization algorithm based on orthogonal learning for optimal power flow problem[J]. *Control Engineering Practice*, 2017, 61(4): 163-172.
- [10] Tran D H, Cheng M Y, Cao M T. Hybrid multiple objective artificial bee colony with differential evolution for the time-cost-quality tradeoff problem[J]. *Knowledge-Based Systems*, 2015, 74(1): 176-186.
- [11] Ozturk C, Hancer E, Karaboga D. Dynamic clustering with improved binary artificial bee colony algorithm[J]. *Applied Soft Computing*, 2015, 28(3): 69-80.
- [12] Ebrahimnejad A, Tavana M, Alrezaamiri H. A novel artificial bee colony algorithm for shortest path problems with fuzzy arc weights[J]. *Measurement*, 2016, 93(11): 48-56.
- [13] Zhu G, Kwong S. Gbest-guided artificial bee colony algorithm for numerical function optimization[J]. *Applied Mathematics and Computation*, 2010, 217(7): 3166-3173.
- [14] Gao W F, Liu S Y, Huang L L. A novel artificial bee colony algorithm based on modified search equation and orthogonal learning[J]. *IEEE Trans Cybern*, 2013, 43(3): 1011-1024.
- [15] Karaboga D, Gorkemli B. A quick artificial bee colony(qABC) algorithm and its performance on optimization problems[J]. *Applied Soft Computing*, 2014, 23(5): 227-238.
- [16] Gao W, Liu S. Improved artificial bee colony algorithm for global optimization[J]. *Information Processing Letters*, 2011, 111(17): 871-882.
- [17] Li G, Cui L, Fu X, et al. Artificial bee colony algorithm with gene recombination for numerical function optimization[J]. *Applied Soft Computing*, 2017, 52(3): 146-159.
- [18] Gao W F, Liu S Y. A modified artificial bee colony algorithm[J]. *Computers & Operations Research*, 2012, 39(3): 687-697.
- [19] Gao W F, Huang L L, Liu S Y, et al. Artificial bee colony algorithm based on information learning[J]. *IEEE Trans Cybern*, 2015, 45(12): 2827-2839.

(责任编辑: 闫妍)