

# 基于全局无偏搜索策略的精英人工蜂群算法

杜振鑫<sup>1,2</sup>, 刘广钟<sup>1</sup>, 韩德志<sup>1</sup>, 余学山<sup>1</sup>, 贾建鑫<sup>1</sup>

(1. 上海海事大学信息工程学院, 上海 201306; 2. 韩山师范学院计算机与信息工程学院, 广东潮州 521041)

**摘 要:** 针对精英人工蜂群算法(ABC\_elite)加速收敛和早熟停滞现象的矛盾, 提出一种改进算法(EABC\_elite). 该算法通过在雇佣蜂阶段引入全局最优解加速收敛, 同时通过普通个体平衡全局最优解过大的引导作用, 因此全局最优解与普通个体的信息都能得到利用而算法仍能较好平衡. 在观察蜂阶段引入普通个体避免算法早熟收敛. 改进算法在加速收敛与防止早熟之间取得很好的平衡, 总体上没有偏向任何方向, 增强了ABC\_elite的全局搜索能力. 实验表明, 改进算法的性能显著好于ABC\_elite以及最近提出的几种较高水平的改进人工蜂群算法.

**关键词:** 人工蜂群; 平衡策略; 搜索公式; 函数优化

**中图分类号:** TP18

**文献标识码:** A

**文章编号:** 0372-2112 (2018)0308-07

**电子学报 URL:** <http://www.ejournal.org.cn>

**DOI:** 10.3969/j.issn.0372-2112.2018.02.008

## Artificial Bee Colony Algorithm with Global and Unbiased Search Strategy

DU Zhen-xin<sup>1,2</sup>, LIU Guang-zhong<sup>1</sup>, HAN De-zhi<sup>1</sup>, YU Xue-shan<sup>1</sup>, JIA Jian-xin<sup>1</sup>

(1. College of Information Engineering, Shanghai Maritime University, Shanghai 201306, China;

2. School of Computer Information Engineering, Hanshan Normal University, Chaozhou, Guangdong 521041, China)

**Abstract:** To settle the contradictory between convergence speed and precocity in artificial bee colony with elite-guided equations (ABC\_elite), an enhanced ABC\_elite (EABC\_elite) is proposed. In the employed bee phase of EABC\_elite, the global best (gbest) individual is introduced to the search equation to accelerate convergence, while the ordinary individual is also introduced to the search equation to balance the gbest's great lead ability, thus the information of gbest and ordinary individuals can all be used and the balance between exploration and exploitation can still be maintained. In the onlooker bee phase of EABC\_elite, the ordinary individual is also introduced into the search equation to avoid the precocity. EABC\_elite can keep good balance between accelerating convergence and averting precocity. It has no bias to any search directions and the global search ability of ABC\_elite is enhanced. The experimental results demonstrate that EABC\_elite performs significantly better than ABC\_elite and other recently-developed state-of-the-art variants of artificial bee colony algorithm.

**Key words:** artificial bee colony; balance strategy; search equation; function optimization

### 1 引言

人工蜂群算法(ABC)是Karaboga<sup>[1]</sup>于2005年提出的一种群体智能优化算法,模拟了蜂群采蜜的智能行为,结构简单、参数较少、易于实现,受到了众多学者的关注和研究,目前已推广到很多领域,如文献[2]将其用于特征选择,文献[3]将其用于网络攻击,都取得了不错的效果.与其他进化算法类似,ABC也面临着收敛速度慢,易早熟等缺点.ABC的优化性能主要依赖于搜

索公式<sup>[4,8]</sup>,许多研究表明,由于搜索公式中的邻居选择机制随机性太强,导致ABC擅长勘探而不擅长开采<sup>[5,6]</sup>.文献[5]首先提出一种全局最优解(global best, gbest)引导的ABC算法,通过在ABC的搜索公式中添加gbest来增强ABC的开采能力.后来,许多研究发现<sup>[6,7]</sup>,通过简单的添加gbest项,虽然可以加快收敛速度,但是因为所有个体都向gbest学习,容易导致早熟收敛问题.为了解决这个问题,文献[7]提出一种没有任

何偏向的算法(CABC),取得了较好的效果.2016年,文献[8]提出,CABC虽然较好的解决了早熟收敛问题,却没有有效利用种群中的有用信息,特别是gbest的信息没有得到充分利用,因此提出一种精英引导的ABC算法(ABC\_elite),与GABC和CABC相比,ABC\_elite在加速收敛与避免早熟之间取得了较好平衡,因此取得了较好的效果.实验结果显示,ABC\_elite及加入深度优先(Depth First Strategy,DFS)策略的DFSABC\_elite好于许多最新提出的改进ABC及多种差分(DE)、粒子群(PSO)改进算法,是目前较高水平的进化算法之一.

ABC\_elite之所以取得较大优势的主要原因,在于其使用了新颖的搜索公式.但是,在ABC\_elite的搜索公式中,由于所有的候选解都偏向精英解,而种群中占多数的普通个体的信息没有得到充分利用,因此搜索区域较小,降低了算法的全局搜索能力.在雇佣蜂阶段未能利用gbest的信息引导算法进化,而在观察蜂阶段开采能力过强.为了解决上述问题,本文提出一种改进的算法EABC\_elite,在较大区域内产生候选解,增加了普通个体的搜索机会,在雇佣蜂阶段利用gbest引导算法进化,以加速收敛.在观察蜂阶段通过使用普通个体,进一步平衡算法的勘探与开采能力.EABC\_elite的最大特点是在总体上没有偏向任何方向,利用了更多的信息防止早熟收敛,同时增强了ABC\_elite的勘探与开采能力,并保持了ABC\_elite的简单性.通过对CEC2015<sup>[9]</sup>复杂多峰函数上的实验表明,改进算法在多数函数上明显优于ABC\_elite及DFSABC\_elite.另外,与最新提出的几种较高水平的改进ABC算法及PSO、DE改进算法相比,本文改进算法也具有明显的优势.

## 2 相关工作

### 2.1 基本人工蜂群算法

在ABC<sup>[1]</sup>中,人工蜂群包含了3种不同类型的蜜蜂:雇佣蜂、观察蜂以及侦察蜂.ABC采用一个随机生成的群体开始迭代搜索.设食物源的规模为SN,其中,食物源 $X_i = (X_{i1}, X_{i2}, \dots, X_{id})$ 代表一个候选解,首先按照下式生成SN个初始解:

$$X_{i,j} = X_j^L + rand_j(X_j^U - X_j^L) \quad (1)$$

式(1)中, $i=1,2,\dots,SN, j=1,2,\dots,D$ . $D$ 是变量的维数. $X_j^L$ 与 $X_j^U$ 分别是第 $j$ 维的下界与上界.算法开始迭代之后,ABC根据蜜蜂的类型将搜索过程分为三个阶段:

(1)雇佣蜂阶段:每只雇佣蜂在对应的食物源 $X_i$ 处根据式(2)生成一个候选解 $V_i = (V_{i1}, V_{i2}, \dots, V_{id})$ ,若 $V_i$ 的适应值比 $X_i$ 的适应值更优,则用 $V_i$ 替换 $X_i$ :

$$V_{i,j} = X_{i,j} + \phi_{i,j}(X_{i,j} - X_{k,j}) \quad (2)$$

其中, $\phi_{i,j}$ 是 $[-1,1]$ 之间的随机数, $k \in \{1,2,\dots,SN\}$ 是随机选择的一个食物源,且 $k \neq i, j \in \{1,2,\dots,D\}$ 是随机

选择的一个维度.

(2)观察蜂阶段:在所有雇佣蜂完成勘探之后,观察蜂按照下面式(3)定义的概率随机选择一个食物源 $i$ 进一步开采:

$$P_i = fit_i / \sum_{i=1}^{SN} fit_i \quad (3)$$

式中, $fit_i$ 是食物源 $X_i$ 的适应度值.从式(3)可见,食物源的适应度值越大,被观察蜂选中的概率越高.其中, $fit_i$ 按下式计算:

$$fit_i = \begin{cases} 1/(1+f_i), & f_i \geq 0 \\ 1+|f_i|, & f_i < 0 \end{cases} \quad (4)$$

这里, $f_i$ 是第 $i$ 个解的目标函数值.

(3)侦察蜂阶段:当雇佣蜂对应的食物源经过limit次未更新,说明该食物源已被开采耗尽.这种情况下,采用式(1)随机初始化一个新食物源代替 $X_i$ .

### 2.2 改进的人工蜂群算法

由于ABC具有较高的优化性能,从而吸引了许多研究人员的关注,目前已经提出了大量改进算法,这些工作大体可以分为三类:(1)ABC与辅助策略的融合.此类改进通过引入新的策略增强ABC的性能.如文献[6]提出按照适应度选择邻居、参数记忆的方法增强ABC的开采能力,文献[7]将正交实验设计方法引入ABC,文献[10]提出一种带记忆的ABC,文献[11]引入混沌鲶鱼效应增强ABC跳出局部最优解的能力等.(2)ABC与其他进化算法的融合.如文献[12]将PSO与ABC融合,文献[13]将思维进化算法与ABC融合,文献[19]与[28]将DE与ABC融合等.(3)搜索公式的改进.ABC最显著的特性在于其搜索公式,这是ABC区别于其他进化算法的主要特征<sup>[6]</sup>.但是,已有许多研究表明,ABC擅长勘探而不擅长开采<sup>[5,6,8]</sup>.所谓勘探能力,是指算法搜索未知区域的能力,而开采能力是指在已经搜索过的区域附近详细搜索的能力.如果勘探能力较强,可以帮助算法跳出局部最优解,也会导致算法具有较慢的收敛速度;如果开采能力较强,可以加快算法收敛速度,但也容易导致算法早熟收敛,陷入局部最优解.是否能够有效平衡这两个相互矛盾的方面,是一个算法取得较高性能的关键因素之一<sup>[8]</sup>.

2010年,文献[5]受到PSO搜索公式的启发,通过引入 $X_{best}$ 项,提出一种新的搜索公式(GABC):

$$V_{i,j} = X_{i,j} + \phi_{i,j}(X_{i,j} - X_{k,j}) + \psi_{i,j}(X_{best,j} - X_{i,j}) \quad (5)$$

式中, $\psi_{i,j}$ 是 $[0,1.5]$ 之间的随机数, $X_{best}$ 是全局最优解gbest.但是,正如文献[7]所指出的,GABC公式右边第二项与第三项可能位于相反的方向,可能导致算法振荡现象.为此,2011年文献[14]提出一种新的gbest引导的搜索公式(IABC):

$$V_{i,j} = X_{best,j} + \phi_{i,j} \cdot (X_{i,j} - X_{r1,j}) \quad (6)$$

式中,  $r1 \in \{1, 2, \dots, SN\}$  是随机选择的序号,  $r1 \neq i$ . 式(6)避免了 GABC 中的振荡问题, 取得了比 GABC 更好的性能. 但是, 在式(6)中, 所有个体都向 gbest 学习, 造成算法开采能力过强, 容易早熟收敛. 2013 年, 为了避免式(5)中的振荡问题, 文献[7]提出一种带交叉算子的 ABC(CABC):

$$V_{i,j} = X_{r1,j} + \phi_{i,j} \cdot (X_{r1,j} - X_{r2,j}) \quad (7)$$

式中,  $r1 \neq r2$ ,  $r1$  与  $r2$  是  $\{1, 2, \dots, SN\}$  中随机选择的食物源序号. 由于式(7)只有一个引导项, 因此避免了 GABC 中的振荡问题. CABC 的最大优点是去掉了  $X_{best}$  的引导, 没有偏向任何方向, 因此较好的平衡了算法的勘探与开采能力, 不容易早熟收敛. 2016 年, 文献[8]提出, 虽然搜索式(7)显著提高了 ABC 的性能, 但是种群中的成功经验并没有充分利用, 特别是种群中最重要的个体  $X_{best}$  没有得到利用, 通过结合 GABC、IABC、CABC 的优点, 提出一种精英 ABC 算法(ABC\_elite), 重新引入 gbest, 并通过精英解平衡 gbest 过大的引导作用, 避免算法早熟:

$$V_{i,j} = X_{e,j} + \phi_{i,j} \cdot (X_{e,j} - X_{k,j}) \quad (8)$$

$$V_{e,j} = \frac{1}{2}(X_{e,j} + X_{best,j}) + \phi_{e,j} \cdot (X_{best,j} - X_{k,j}) \quad (9)$$

式中,  $X_e$  是从种群最好的  $p \cdot SN$  个个体中随机选择的精英解,  $p \in (0, 1)$ ,  $X_k$  是从  $\{1, 2, \dots, SN\}$  中随机选择的普通个体,  $e \neq k \neq i$ ,  $\phi_{i,j}$  与  $\phi_{e,j}$  是  $[-1, 1]$  之间的随机数. 式(8)、(9)右边只有一个引导项, 避免了 GABC 中的振荡问题. 在 ABC\_elite 中, 雇佣蜂阶段使用式(8), 所有个体向精英解学习, 观察蜂阶段使用式(9), 所有个体向 gbest 学习. 式(9)中的基向量  $\frac{1}{2}(X_{e,j} + X_{best,j})$  通过加入  $X_e$  平衡 gbest 过大的引导作用, 避免算法早熟收敛. 2017 年, 同样为了避免 GABC 式(5)中的振荡问题, 文献[15]提出一种改进的 GABC 算法 MGABC, 将 GABC 式(5)的右边两项进行拆分:

$$v_{i,j} = \begin{cases} X_{i,j} + \phi_{i,j} \cdot (X_{i,j} - X_{k,j}), & \text{if } rand < P \\ X_{i,j} + \psi_{i,j} \cdot (X_{best,j} - X_{i,j}), & \text{otherwise} \end{cases} \quad (10)$$

式中,  $P$  是新引入的参数,  $0 < P < 1$ .

### 3 本文提出的算法

#### 3.1 ABC\_elite 算法的主要优点与缺点

ABC\_elite 的搜索公式的主要优势在于精英引导, 对比 IABC 与 GABC, 在避免震荡问题的同时, 避免 gbest 过大的引导作用导致早熟收敛. 但是, ABC\_elite 的上述优点恰好也是它的缺点. ABC\_elite 的搜索式(8)与式(9)可以看成基向量加扰动向量的形式. 例如, 在式(8)中, 基向量是  $X_e$ , 而扰动向量是  $X_e - X_k$ , 在式(9)中, 基

向量是  $(X_{best} + X_e)/2$ , 而扰动向量是  $X_{best} - X_k$ . 由于系数  $\phi$  在各种对比算法中都是相同的, 这里不做考虑.

根据式(8)与式(9), 所有的候选解都偏向精英解, 导致搜索区域仅限于精英所在区域, 而精英只占较小的比例  $p$  (文献[8]中  $p = 0.1$ ), 因此搜索区域较小. 在 ABC, GABC, CABC 的搜索式(2)、(5)、(7)中, 候选解都是以普通解作为基向量, 加上一定的扰动形成, 这样可以保证普通个体有足够的机会参与进化过程, 从而保证算法有较高的全局搜索能力. 而在 ABC\_elite 的式(8)、(9)中, 基向量都是精英解, 这造成普通个体的周围无法得到搜索, 普通个体  $X_k$  在整个进化过程中仅能以扰动向量出现. 另外, 在式(8)与式(9)中, 扰动成分  $X_{best,j} - X_{k,j}$  较大. 因为  $X_{best}$  是种群中的最优解, 而  $X_k$  仅仅是一个普通的解, 一般来说,  $X_k$  远差于  $X_{best}$ , 因此  $X_{best,j} - X_{k,j}$  是比较大的扰动.

#### 3.2 本文提出的算法 EABC\_elite

文献[16]总结了许多进化算法的发展规律, 认为: “在进化算法中, 有效使用的信息越多, 算法越不容易早熟收敛.” 在 PSO 中, 所有个体都向 gbest 学习, 容易早熟, 后来相继提出了大量改进算法, 如竞争与合作 PSO<sup>[16]</sup>、社会学习 PSO<sup>[17]</sup> 等, 都利用更多的信息以帮助算法跳出局部最优解. ABC 的发展也经历了相似的过程, GABC 最早提出利用 gbest 加速收敛, 但是所有个体都向 gbest 学习, 容易早熟收敛. CABC 没有偏向任何方向, 取得了勘探与开采能力的平衡, 而 ABC\_elite 虽引入了 gbest 的引导, 但同时通过精英解平衡 gbest 过大的引导作用.

为了克服 ABC\_elite 的上述缺点, 本文提出式(13)与式(14):

$$\mu = \frac{1}{3}(X_{best,j} + X_{e,j} + X_{k,j}) \quad (11)$$

$$\delta = \frac{1}{3}(|X_{best,j} - X_{e,j}| + |X_{e,j} - X_{k,j}| + |X_{best,j} - X_{k,j}|) \quad (12)$$

$$V_{i,j} = \mu + \phi_{i,j} \cdot \delta \quad (13)$$

$$V_{e,j} = \mu + \phi_{e,j} \cdot \delta \quad (14)$$

式中,  $\phi_{i,j}$  与  $\phi_{e,j}$  是  $[-1, 1]$  之间的随机数,  $|\cdot|$  是绝对值符号,  $\mu$  称为基向量,  $\delta$  称为扰动向量,  $X_e$  是从种群最好的  $p \cdot SN$  个个体中随机选择的精英解,  $p \in (0, 1)$ ,  $X_k$  是从  $\{1, 2, \dots, SN\}$  中随机选择的普通个体,  $e \neq k \neq i$ . 其中, 式(13)用于雇佣蜂阶段; 式(14)用于观察蜂阶段. 与 ABC\_elite 类似, 在式(14)中, 只对精英个体更新, 以增强算法的开采能力. 本文提出的算法称为 EABC\_elite (Enhanced ABC\_elite), 其伪代码见图 1.

```

01: Initialization; Generate  $SN$  solutions that contain  $D$  variables according to Eq. (1);  $Fes = 0$ 
02: while  $Fes < max\_Fes$ 
03:   Select the top  $T = p$ .  $SN$  solutions as elite solutions from population
04:   for  $i = 1$  to  $SN$  //employed bee phase
05:     Generate a new candidate solution  $V_i$  in the neighbourhood of  $X_i$  using Eq. (13), Evaluate  $V_i$ 
06:     if  $f(V_i) < f(X_i)$ 
07:       Replace  $X_i$  by  $V_i$  and  $counter(i) = 0$ 
08:     else
09:        $counter(i) = counter(i) + 1$ 
10:     end if
11:   end for //end employed bee phase
12:   for  $i = 1$  to  $r \cdot p \cdot SN$  //onlooker bee phase
13:     Select a solution  $X_e$  from elite solutions randomly to search
14:     Generate a new candidate solution  $V_e$  in the neighbourhood of  $X_e$  using Eq. (14), Evaluate  $V_e$ 
15:     if  $f(V_e) < f(X_e)$ 
16:       Replace  $X_e$  by  $V_e$  and  $counter(e) = 0$ 
17:     else
18:        $counter(e) = counter(e) + 1$ 
19:     end if
20:   end for //end onlooker bee phase
21:    $Fes = Fes + SN \times 2$ 
22:   Select the solution  $X_{max}$  with max counter value //Scout bee phase
23:   if  $counter(max) > limit$  //only one food source with max counter value can be initialized
24:     Replace  $X_{max}$  by a new solution generated according to Eq. (1),  $Fes = Fes + 1$ ,  $counter(max) = 0$ ;
25:   end if //end scout bee phase
26: end while

```

图 1 EABC\_elite 为伪代码

### 3.3 EABC\_elite 的特点

(1) EABC\_elite 使用了更多个体的信息, 增大了全局搜索能力, 没有偏向任何方向, 进一步平衡了算法的勘探与开此能力. 在雇佣蜂阶段使用了  $X_{best}$  加速收敛, 同时, 增加普通个体  $X_k$  实现勘探与开采的平衡, 避免早熟. 在 ABC\_elite 的雇佣蜂式 (8) 中, 基向量只有一项  $X_e$ , 而在 EABC\_elite 的雇佣蜂式 (13) 中, 基向量由 3 项组成: 全局最优解  $X_{best}$ 、精英个体  $X_e$ 、普通个体  $X_k$ . 由于  $X_{best}$  开采能力最强, 而普通个体  $X_k$  勘探能力较强, 因此  $X_{best}$  与  $X_k$  相互“中和”, 最终公式具有较合适的开采能力. 但是, 式 (13) 利用了更多的有用信息, 同时兼顾了加速收敛与避免早熟两个因素. 图 2 (a) 给出了 ABC\_elite 式 (8) 的平衡策略, 图 2 (b) 给出了 EABC\_elite 的式 (13) 的平衡策略. 式 (13) 虽然使用了开采能力更强的  $X_{best}$ , 但是因为同时使用了勘探能力更强的普通个体  $X_k$ , 所以仍然能够保持平衡. 同样的, EABC\_elite 通过在观察蜂式 (14) 的基向量中加入普通个体  $X_k$ , 避免算法早熟收敛.

(2) EABC\_elite 比 ABC\_elite 的搜索区域大  
ABC\_elite 的式 (9) 的搜索原理如图 3 (a) 所示, 候

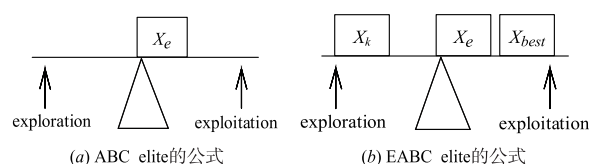


图 2 ABC\_elite 与 EABC\_elite 的平衡策略

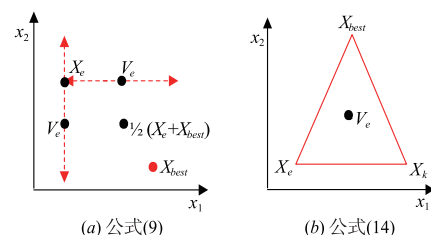


图 3 ABC\_elite 公式与 EABC\_elite 公式的搜索原理

选解  $V_e$  只能在红色虚线上生成, 搜索区域较小, 当公式中的扰动成分  $X_k$  较差的时候,  $X_{best} - X_k$  是一个较大的扰动, 候选解会沿着箭头延伸到较远的地方, 远离精英解. EABC\_elite 的式 (14) 的搜索原理如图 3 (b) 所示, 候选解在  $X_{best}$ ,  $X_k$  与  $X_e$  围成的三角形区域内生成, 搜索区域较大. 同时, 式 (13) 与式 (14) 由于只含有一个扰动

项  $\delta$ , 从而避免了 GABC 的振荡问题. 在进化的前期, 由于  $\delta$  较大, 算法以勘探为主, 尽可能搜索较大的区域, 随种群不断进化,  $X_{\text{best}}, X_k$  与  $X_e$  之间的差异逐渐减小,  $\delta$  随之减小, 因此算法逐渐转为以开采为主.

EABC\_elite 与 ABC-elite 的唯一区别是采用不同公式, ABC\_elite 在雇佣蜂阶段采用式(8), 在观察蜂阶段采用式(9); EABC\_elite 在雇佣蜂阶段采用式(13), 在观察蜂阶段采用式(14). 在 ABC\_elite 及 EABC-elite 的观察蜂阶段, 只有精英解有机会产生候选解, 普通解没有机会, 这样可将计算资源集中到最有潜力的区域, 增强开采能力.

显然, 对比 ABC\_elite, EABC\_elite 仅改变了搜索公式, 没有引入新的操作, 因此复杂度与 ABC\_elite 相同, 为  $O(D \cdot SN)^{[8]}$ , 与基本 ABC 的计算复杂度相同. EABC\_elite 的收敛性容易在形式上给出证明. 在式(13)与式(14)中, 由于基向量  $\mu$  与扰动向量  $\delta$  中的  $X_e$  与  $X_k$  随着算法不断迭代, 与  $X_{\text{best}}$  的差异逐渐减小, 因此  $\mu$  逐渐收敛于  $X_{\text{best}}$ , 而  $\delta$  逐渐收敛于 0, 因此式(13)与式(14)逐渐收敛于全局最优解  $X_{\text{best}}$ .

#### 4 实验与分析

为了检验 EABC\_elite 的性能, 在 CEC2015<sup>[9]</sup> 测试函数集上进行了测试. CEC2015 是一套求解难度较高的测试函数集, 所有函数在基本函数的基础上通过偏移、旋转增加求解难度, 其中  $F_1-F_2$  是单峰函数,  $F_3-F_5$  是简单多峰函数,  $F_6-F_8$  是混合函数,  $F_9-F_{15}$  是复合函数. 测试函数的维数  $D=30$ , 种群规模与文献[8]一致, 即  $SN=50$ . 按照 CEC2015 的建议, 最大函数评价次数  $\max\_Fes=10000 \cdot D$ , 其余参数见表 1, 这些参数均参照相应算法的原始文献设置.

表 1 参数设置

算法	发表年	参数设置
MGABC	2017	limit = $SN \cdot D, P=0.3$
BCABC	2015	limit = 200
DGABC	2016	limit = $SN \cdot D, C=1.5, F=0.2, CR=0.3$
ABC_elite	2016	limit = $SN \cdot D, p=0.1, r=1/p$
DFSABC_elite	2016	limit = $SN \cdot D, p=0.1, r=1/p$
EABC_elite	-	limit = $SN \cdot D, p=0.1, r=1/p$

在所有实验中, 每种算法独立运行 25 次, 记录其平均结果. 采用 Wilcoxon 非参数统计<sup>[18]</sup> 方法统计 EABC\_elite 与被比较算法的显著性差异, 显著性水平为 0.05, 符号“+”, “=”, “-”分别表示被比较的算法好于、等

于、差于 EABC\_elite.

EABC\_elite 算法与 MGABC<sup>[15]</sup>, BCABC<sup>[6]</sup>, DGABC<sup>[19]</sup>, ABC\_elite<sup>[8]</sup>, DFSABC\_elite<sup>[8]</sup> 比较. 之所以与这些算法比较, 是因为这些算法都是 GABC 算法的最新改进版本. 其中, BCABC 是按适应度选择邻居、参数记忆、骨干模块三种策略与 CABC 算法的集成, 文献[6]的实验结果表明, BCABC 具有较强的优化性能. DGABC<sup>[19]</sup> 是 DE 与 GABC 算法的集成, 并采用了竞争策略. 另外, DGABC 采用混沌反向初始化策略, 但是为了公平比较, 本文所有算法均采用随机初始化策略. DFSABC\_elite<sup>[8]</sup> 是 ABC\_elite 与 DFS 策略的集成, 由于 DFSABC-elite 算法已经与许多较好的算法比较, 例如 ABCVSS<sup>[20]</sup>, CABC<sup>[7]</sup>, Best-so-far ABC<sup>[21]</sup>, MABC<sup>[22]</sup>, qABC<sup>[23]</sup>, EABC<sup>[24]</sup>, BABC<sup>[6]</sup> 及 CoDE<sup>[25]</sup>, CLPSO<sup>[26]</sup> 等多种 DE 与 PSO 改进算法, 结果表明 DFSABC\_elite 明显好于这些算法, 因此本文不再与这些算法比较.

测试结果见表 2, 其中, Mean Error 表示平均误差值, 即算法在 25 次运行中得到的结果的均值与函数理论最优解的差, 该值越小, 表示求解质量越好, std 表示标准差. 最好结果用黑体标识. 从表 2 可见, 在单峰函数  $F_1-F_2$  上, EABC\_elite 明显好于另外几种算法, 且提高的幅度较大. 这主要得益于 EABC\_elite 在雇佣蜂阶段的搜索式(13)中采用 gbest 引导, 从而加快了算法的收敛速度. 在多峰函数  $F_3-F_{15}$  上, EABC\_elite 在大多数函数上都取得了最好的求解精度. 对比 MGABC, BCABC, DGABC, ABC\_elite, DFSABC\_elite, 分别只有 3, 2, 2, 1, 2 个函数没有取得最好结果, 在其余所有函数上, EABC\_elite 都取得了最好结果. 这主要得益于 EABC\_elite 的搜索式(13)与式(14)没有偏向任何解, 利用了更多个体的信息, 因此不容易陷入局部最优解. 总之, 由于 EABC\_elite 采用了更强的启发式信息的同时采用了更好的平衡策略, 使得算法能够在加速收敛的同时, 避免陷入局部最优解. EABC\_elite 无论在单峰函数还是多峰函数上, 都取得了最好的性能. 图 4 给出了 6 种算法在  $F1, F2, F4$  上的收敛曲线图, 曲线越陡表示算法收敛速度越快. 从图 4 可以看出, EABC\_elite 算法相对于其他算法具有更快的收敛速度和更高的求解精度.

另外, 本文也将 EABC\_elite 与最近提出的 SLP-SO<sup>[17]</sup>, sinDE<sup>[27]</sup>, ABCADE<sup>[28]</sup> 等进行了比较. 实验结果表明, EABC\_elite 显著好于这些算法, 但是由于篇幅所限, 本文不再给出实验结果.

表 2  $D=30$  时,6 种算法在 CEC2015 测试函数集上的误差均值与标准差

Alg	MGABC Mean Error(std)	BCABC Mean Error(std)	DGABC Mean Error(std)	ABC_elite Mean Error(std)	DFSABC_elite Mean Error(std)	EABC_elite Mean Error(std)
$F_1$	2.82e+06(1.14e+06) -	3.19e+06(1.96e+06) -	3.10e+06(1.53e+06) -	2.97e+06(1.51e+06) -	3.20e+06(1.08e+06) -	2.27e+06(9.86e+05)
$F_2$	1.96e+03(3.01e+03) -	3.24e+03(4.81e+03) -	2.51e+03(2.34e+03) -	2.67e+03(3.59e+03) -	1.64e+03(2.03e+03) -	1.20e+03(2.26e+03)
$F_3$	2.01e+01(4.05e-02) -	2.03e+01(7.43e-02) -	2.01e+01(3.91e-02) -	2.01e+01(3.96e-02) -	2.01e+01(4.61e-02) -	2.00e+01(3.83e-02)
$F_4$	5.60e+01(9.12e+00) -	3.92e+01(7.13e+00) -	4.09e+01(6.79e+00) -	4.24e+01(7.09e+00) -	4.21e+01(6.8e+00) -	3.73e+01(6.40e+00)
$F_5$	1.94e+03(2.58e+02) -	1.96e+03(4.09e+02) -	1.83e+03(2.06e+02) -	1.69e+03(2.58e+02) -	1.68e+03(2.47e+02) -	1.67e+03(2.40e+02)
$F_6$	1.23e+06(6.96e+05) -	1.10e+06(6.83e+05) -	8.01e+05(5.80e+05) -	1.07e+06(5.78e+05) -	1.03e+06(5.58e+05) -	6.34e+05(4.65e+05)
$F_7$	8.49e+00(1.90e+00) -	1.06e+01(3.78e+00) -	1.01e+01(7.93e-01) -	8.32e+00(1.41e+00) -	8.52e+00(1.31e+00) -	7.96e+00(1.30e+00)
$F_8$	2.27e+05(1.34e+05) +	2.39e+05(1.60e+05) +	1.73e+05(8.95e+04) +	2.74e+05(1.69e+05) -	2.36e+05(1.36e+05) +	2.50e+05(1.33e+05)
$F_9$	1.04e+02(3.13e-01) -	1.03e+02(2.73e-01) =	1.03e+02(2.72e-01) =	1.04e+02(2.24e-01) -	1.04e+02(2.63e-01) -	1.03e+02(2.41e-01)
$F_{10}$	6.39e+05(4.02e+05) -	5.08e+05(3.34e+05) -	4.17e+05(2.34e+05) -	6.31e+05(4.50e+05) -	6.99e+05(5.80e+05) -	4.09e+05(5.26e+05)
$F_{11}$	3.33e+02(8.18e+01) -	3.28e+02(7.05e+01) =	4.50e+02(1.49e+02) -	3.43e+02(8.09e+01) -	3.65e+02(1.30e+02) -	3.28e+02(9.80e+01)
$F_{12}$	1.04e+02(2.38e-01) =	1.04e+02(3.55e-01) =	1.05e+2(4.76e-01) -	1.05e+02(3.37e-01) -	1.05e+02(6.69e-01) -	1.04e+02(3.20e-01)
$F_{13}$	2.62e-02(2.70e-04) +	2.64e-02(5.54e-04) +	2.61e-02(7.67e-04) +	2.63e-02(3.38e-04) +	2.63e-02(3.87e-04) +	2.65e-02(5.12e-04)
$F_{14}$	3.13e+04(5.16e+03) +	3.31e+04(9.25e+02) -	3.28e+04(8.23e+02) -	3.21e+04(8.92e+02) -	3.21e+04(9.64e+02) -	3.19e+04(7.51e+02)
$F_{15}$	1.00e+02(2.02e-13) =	1.00e+02(1.18e-12) =	1.00e+02(7.22e-14) =	1.00e+02(6.83e-14) =	1.00e+02(4.25e-13) =	1.00e+02(3.23e-14)
+ / = / -	3/2/10	2/4/9	2/2/11	1/1/13	2/1/12	-

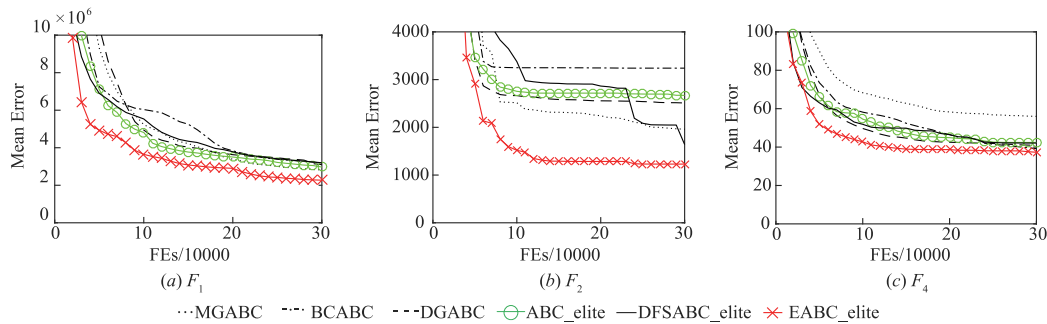


图4  $F_1$ 、 $F_2$ 与 $F_4$ 函数的收敛曲线图

### 5 结论

为了加快 ABC\_elite 的收敛速度并避免早熟收敛, 本文提出一种改进算法 EABC\_elite, 利用 gbest 加快算法收敛速度, 并利用普通个体避免算法失去平衡, 从而在加速收敛和防止早熟之间取得较好的平衡. 从总体上来说, EABC\_elite 没有偏向任何搜索方向, 充分利用了全局最优解 gbest, 精英解  $X_e$  与普通个体  $X_k$  的信息, 增强了算法的全局搜索能力, 不容易早熟收敛. 实际上, EABC\_elite 可以看成 CABC、IABC 与 ABC\_elite 算法的融合, 分别集成了 CABC 的无偏特性, IABC 利用 gbest 引导算法进化的优点, 以及 ABC\_elite 精英引导的优势. 在 CEC2015 测试函数集上的实验表明, EABC\_elite 与几种较高水平的 ABC 改进算法相比具有更快的收敛速度和更高的求解精度. 对比最新提出的多种 PSO、DE 改进算法, EABC\_elite 也具有明显优势.

#### 参考文献

[1] Karaboga D. An idea based on honey bee swarm for numeri-

cal optimization[R]. Kayseri:Engineering Faculty Computer Engineering Department, Ereyes University, 2005.

[2] Zorarpacl E, Özel S A. A hybrid approach of differential evolution and artificial bee colony for feature selection[J]. Expert Systems with Applications, 2016, 62(11): 91 - 103.

[3] Lozano M, García-Martínez C, Rodríguez F J, et al. Optimizing network attacks by artificial bee colony[J]. Information Sciences, 2017, 377(1): 30 - 50.

[4] Cui L, Li G, Zhu Z, et al. A novel artificial bee colony algorithm with an adaptive population size for numerical function optimization[J]. Information Sciences, 2017, 414(22): 53 - 67.

[5] Zhu G, Kwong S. Gbest-guided artificial bee colony algorithm for numerical function optimization[J]. Applied Mathematics and Computation, 2010, 217(7): 3166 - 3173.

[6] Gao W, Chan F T S, Huang L, et al. Bare bones artificial bee colony algorithm with parameter adaptation and fitness-based neighborhood[J]. Information Sciences, 2015, 316(18): 180 - 200.

[7] Gao W, Liu S, Huang L. A novel artificial bee colony algo-

- rithm based on modified search equation and orthogonal learning[J]. IEEE Transactions on Cybernetics, 2013, 43(3):1011–1024.
- [8] Cui L, Li G, Lin Q, et al. A novel artificial bee colony algorithm with depth-first search framework and elite-guided search equation[J]. Information Sciences, 2016, 367(22):1012–1044.
- [9] Liang J J, Qu B Y, Suganthan P N, et al. Problem definitions and evaluation criteria for the CEC 2015 competition on learning-based real-parameter single objective optimization[R]. Zhengzhou China: Computational Intelligence Laboratory, Zhengzhou University, 2014.
- [10] Li X, Yang G. Artificial bee colony algorithm with memory[J]. Applied Soft Computing, 2016, 41(4):362–372.
- [11] 王生生, 杨娟娟, 柴胜. 基于混沌鲶鱼效应的人工蜂群算法及应用[J]. 电子学报, 2014, 42(9):1731–1737.  
WANG S S, YANG J J, CHAI S. Artificial bee colony algorithm with chaotic catfish effect and its application[J]. Acta Electronica Sinica, 2014, 42(9):1731–1737. (in Chinese)
- [12] Li Z, Wang W, Yan Y, et al. PS-ABC: A hybrid algorithm based on particle swarm and artificial bee colony for high-dimensional optimization problems[J]. Expert Systems with Applications, 2015, 42(22):8881–8895.
- [13] 暴励. 一种思维进化蜂群算法[J]. 电子学报, 2015, 43(5):948–955.  
BAO Li. A mind evolutionary artificial bee colony algorithm[J]. Acta Electronica Sinica, 2015, 43(5):948–955. (in Chinese)
- [14] Gao W, Liu S. Improved artificial bee colony algorithm for global optimization[J]. Information Processing Letters, 2011, 111(17):871–882.
- [15] Cui L, Zhang K, Li G, et al. Modified Gbest-guided artificial bee colony algorithm with new probability model[J]. Soft Computing, 2017, 21(1):1–27.
- [16] Li Y, Zhan Z H, Lin S, et al. Competitive and cooperative particle swarm optimization with information sharing mechanism for global optimization problems[J]. Information Sciences, 2015, 293(4):370–382.
- [17] Cheng R, Jin Y. A social learning particle swarm optimization algorithm for scalable optimization[J]. Information Sciences, 2015, 291(1):43–60.
- [18] Derrac J, García S, Molina D, et al. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms[J]. Swarm and Evolutionary Computation, 2011, 1(1):3–18.
- [19] Gao W, Huang L, Wang J, et al. Enhanced artificial bee colony algorithm through differential evolution[J]. Applied Soft Computing, 2016, 48(11):137–150.
- [20] 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(8):140–157.
- [21] Banharsakun A, Achalakul T, Sirinaovakul B. The best-so-far selection in artificial bee colony algorithm[J]. Applied Soft Computing, 2011, 11(2):2888–2901.
- [22] Gao W, Liu S. A modified artificial bee colony algorithm[J]. Computers & Operations Research, 2012, 39(3):687–697.
- [23] Karaboga D, Gorkemli B. A quick artificial bee colony (qABC) algorithm and its performance on optimization problems[J]. Applied Soft Computing, 2014, 23(10):227–238.
- [24] Gao W, Liu S, Huang L. Enhancing artificial bee colony algorithm using more information-based search equations[J]. Information Sciences, 2014, 270(12):112–133.
- [25] Wang Y, Cai Z, Zhang Q. Differential evolution with composite trial vector generation strategies and control parameters[J]. IEEE Transactions on Evolutionary Computation, 2011, 15(1):55–66.
- [26] Liang J J, Qin A K, Suganthan P N, et al. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions[J]. IEEE Transactions on Evolutionary Computation, 2006, 10(3):281–295.
- [27] Draa A, Bouzoubia S, Boukhalfa I. A sinusoidal differential evolution algorithm for numerical optimisation[J]. Applied Soft Computing, 2015, 27(2):99–126.
- [28] Liang Z, Hu K, Zhu Q, et al. An enhanced artificial bee colony algorithm with adaptive differential operators[J]. Applied Soft Computing, 2017, 58(9):480–494.

#### 作者简介



**杜振鑫** 男, 1976 年出生, 博士研究生, 主要研究方向为进化计算, 智能信息处理。  
E-mail: duzhenxinmail@163.com

**刘广钟** 男, 1962 年出生, 博士, 教授、博导, 主要研究方向为无线传感器网络、智能信息处理。  
E-mail: gzhliu@shmtu.edu.cn

**韩德志(通信作者)** 男, 1966 年出生, 博士, 教授、博导, 主要研究方向为云计算, 智能信息处理。  
E-mail: dezhihan88@sina.com