

一种双种群差分蜂群算法

暴 励^{1,2}, 曾建潮¹

(1. 太原科技大学 复杂系统与计算智能实验室, 山西 太原 030024; 2. 广播电影电视管理干部学院 学生处, 山西 太原 030013)

摘要: 人工蜂群算法(ABC)是一种基于蜜蜂群智能搜索行为的随机优化算法. 为了有效改善人工蜂群算法的性能, 结合差分进化算法, 提出一种新的双种群差分蜂群算法(BDABC). 该算法首先通过基于反向学习的策略初始化种群, 使得初始化的个体尽可能均匀分布在搜索空间, 然后将种群中的个体随机分成两组, 每组采用不同的优化策略同时进行寻优, 并通过在两群体之间引入交互学习的思想, 来提高算法的收敛速度. 基于6个标准测试函数的仿真实验表明, BDABC算法能有效避免早熟收敛, 全局优化能力和收敛速率都有显著提高.

关键词: 人工蜂群算法; 双种群; 差分进化算法; 反向学习

中图分类号: TP18 **文献标识码:** A

A bi-group differential artificial bee colony algorithm

BAO Li^{1,2}, ZENG Jian-chao¹

(1. Complex System and Computational Intelligence Laboratory,

Taiyuan University of Science and Technology, Taiyuan Shanxi 030024, China;

2. Students' Affairs Department, the Professional Training Academy of Radio Film and Television, Taiyuan Shanxi 030013, China)

Abstract: Artificial bee colony(ABC) algorithm is a stochastic optimization algorithm based on the particular intelligent behavior of honeybee swarms. In order to improve the performance of artificial bee colony(ABC) algorithm, a novel bi-group differential artificial bee colony algorithm(BDABC) which is combined with differential evolution(DE) algorithm is proposed. In this algorithm, an initialization strategy based on the opposition-based learning is applied to diversify the initial individuals in the search space. All individuals are randomly divided into two populations, and the evolutions of two sub-groups are simultaneously performed with different optimization strategies. The interactive learning strategy is introduced to accelerate the convergence speed. Experimental results on six benchmark functions show that the BDABC algorithm not only effectively avoids the premature convergence, but also significantly improves the global optimization ability and the convergence speed.

Key words: artificial bee colony; bi-group; differential evolution; opposition-based learning

1 引言(Introduction)

群智能优化算法(swarm intelligence algorithm, SIA)是通过模拟自然界生物的群体行为而构造的随机优化算法, 如蚁群算法、微粒群算法、人工鱼群算、人工蜂群算法等. 他们为解决大量存在于计算机科学、管理科学、控制工程等科研领域的全局优化问题提供了新的途径, 因此成为科学界长期研究的热点.

人工蜂群算法^[1](artificial bee colony, ABC)是由 Karaboga 于 2005 年提出的一种基于蜜蜂群智能搜索行为的随机优化算法. 目前, 关于 ABC 算法研究与应用还处于初级阶段, 但由于其控制参数少、易于实现、计算简洁等优点, 已被越来越多的学者所关注. ABC 算法已成功的应用于解决函数的数值优化问题^[2~4], 而且在人工神经网络训练^[5,6]、滤波器设计^[7]、网络优化^[8]、机器人路径规划^[9]、生产调度^[10]

等工程领域也取得了良好的应用效果.

虽然 ABC 算法在许多优化问题中都表现出优于微粒群(PSO)、差分进化(DE)等算法的性能^[3,4], 较好的平衡了探索和开采的能力. 但是由于进化方式和选择策略的影响, 使算法在快速收敛的同时, 增加了陷入局部最优或早熟收敛的概率. 因此, 为了提高 ABC 算法的寻优能力、加快收敛速度、克服早熟收敛问题, 许多学者对其进行了改进. Singh^[11]提出了一种解决最小生成树问题的人工蜂群算法; 蔡沛伟^[12]提出了一种交互的人工蜂群算法; 丁海军等^[13]通过对 ABC 算法参数的改进, 提出了一种适合组合优化问题的改进蜂群算法, 并将其应用在 TSP 问题的求解上; 康飞等^[14]针对人工蜂群算法求解反演分析时存在收敛速度慢、长期停滞等缺陷, 引入了文化算法的双层进化结构和具有概率突跳特点的模拟退火操作, 提出了一种具有知识

引导功能的人工蜂群算法,为解决参数识别和系统优化提供了一种新方法. 本文提出了一种改进的ABC算法——双种群差分蜂群算法(bi-group differential artificial bee colony algorithm, BDABC). 将人工蜂群算法(ABC)与差分进化算法(differential evolution, DE)混合,采用双种群结构,两种群独立进化,在适当的时候两种群之间进行信息交换,从而在维持种群多样性的同时加速进化过程. 为了使初始种群尽可能均匀分布在搜索空间,采用了基于反向学习的策略来初始化种群,从而提高种群解的质量.

2 人工蜂群算法和差分进化算法(ABC algorithm and DE algorithm)

2.1 人工蜂群算法(ABC algorithm)

ABC算法中,人工蜂群主要由以下3部分组成: 引领蜂(employed bees)、跟随蜂(onlookers)和侦察蜂(scouts). ABC算法在求解优化问题时,食物源的位置被抽象成解空间中的点,蜜蜂采蜜(食物源)的过程也就是搜寻最优解的过程. 考虑全局优化问题: $\min \sigma = f(X)$ s.t. $X \in S \subset \mathbb{R}^n$, 则问题的每个可行解对应一个食物源位置,每个食物源的优劣程度取决于待优化问题所确定的适应值,解的个数(SN)等于引领蜂或跟随蜂的个数. 本文用 d 维向量 $X_i = (x_{i1}, x_{i2}, \dots, x_{id}) \in S$ 来表示第 i 个食物源的位置. 首先,ABC算法生成含有 SN 个解(食物源)的初始种群,每个解 $X_i (i = 1, 2, \dots, SN)$ 是一个 d 维的向量. 然后,蜜蜂对所有的食物源进行循环搜索,循环次数为 $C (C = 1, 2, \dots, MCN)$. 引领蜂首先对相应的食物源(解)进行一次邻域搜索,然后根据“贪婪”选择策略,如果搜索到的食物源(解)的花蜜数量(适应度)优于以前的,则用新的食物源位置代替旧的食物源位置,否则保持旧的食物源位置不变. 所有的引领蜂完成搜索之后,回到舞蹈区把食物源上花蜜数量的信息通过跳摇摆舞传达给跟随蜂. 跟随蜂根据得到的信息按照概率选择食物源. 花蜜越多的食物源,被选择的概率越大. 跟随蜂选中食物源后,也进行一次邻域搜索,并保留较好的解. ABC算法就是通过如此重复的搜索,最终来找到最优解.

引领蜂和跟随蜂依据式(1)进行食物源位置更新,

$$v_{ij} = x_{ij} + r_{ij}(x_{ij} - x_{kj}), \quad (1)$$

其中: $k \in \{1, 2, \dots, SN\}$, $j \in \{1, 2, \dots, d\}$, 这两个数都是随机选取的,但是 k 不能等于 i (k 是 i 邻域的一个解), $r_{ij} \in [-1, 1]$ 是一个随机数,它控制 x_{ij} 邻域的生成范围,随着搜索接近最优解,邻域的范围会逐渐减小.

ABC算法中跟随蜂对食物源的选择是通过观察完引领蜂的摇摆舞来判断食物源的收益率,并依据收益率大小来选择到哪个食物源采蜜. 收益率通过

适应度值来表示,选择概率 P_i 按照下式确定:

$$P_i = \text{fit}_i / \sum_{i=1}^{SN} \text{fit}_i, \quad (2)$$

其中: fit_i 是第 i 个解的适应度值, SN 是解的个数.

在ABC算法中,还有一个控制参数“limit”,它用来记录某个解被更新的次数. 假定某个解连续经过limit次循环之后没有得到改善,表明这个解陷入局部最优,那么这个位置就要被放弃,与这个解相对应的引领蜂也转变为侦察蜂. 假设被放弃的解是 x_i 且 $j \in \{1, 2, \dots, d\}$,那么就由侦察蜂通过式(3)随机产生一个新的解来代替 x_i .

$$x_i^j = x_{\min}^j + \text{rand}(0, 1)(x_{\max}^j - x_{\min}^j). \quad (3)$$

2.2 差分进化算法(DE algorithm)

差分进化^[15](differential evolution, DE)算法是由Rainer Storn和Kenneth Price于1995年共同提出的一种采用实数矢量编码在连续空间中进行随机搜索的优化算法,具有原理简单、受控参数少、鲁棒性强等特点. 在基本DE算法中,首先生成含有 NP 个解的初始种群,每个解 $X_i = (x_{i1}, x_{i2}, \dots, x_{iD}) (i = 1, 2, \dots, NP)$ 是一个 D 维的向量. 进化过程主要由变异、交叉和选择3个操作组成.

1) 变异操作: 在DE算法中,根据变异个体的生成方式不同,有多种不同的DE变异方式^[16],常见的变异方式有DE/rand/1/bin和DE/best/1/bin两种,变异公式如下:

$$v_{i,j} = x_{r1,j} + F * (x_{r2,j} - x_{r3,j}), \quad (4)$$

$$v_{i,j} = x_{\text{best},j} + F * (x_{r1,j} - x_{r2,j}), \quad (5)$$

其中: $x_{\text{best},j}(t)$ 是当前种群中最好个体的第 j 维向量, $i \neq r_1 \neq r_2 \neq r_3 \in [1, NP]$, $F \in (0, 1)$ 为缩放因子,用于控制差异向量的缩放程度. 在本文中采用DE/best/1/bin变异方式.

2) 交叉操作: 按照一定的概率,将变异得到的中间个体 v_i 与父代个体 x_i 之间进行交叉操作,生成新个体 u_i ,

$$u_{i,j} = \begin{cases} v_{i,j}, & \text{rand}(0, 1) \leq CR \text{ or } j = \text{rand}(1, D), \\ x_{i,j}, & \text{其他}. \end{cases} \quad (6)$$

其中: $CR \in [0, 1]$ 为交叉概率, $\text{rand}(0, 1)$ 为 $[0, 1]$ 之间均匀分布的随机数, $\text{rand}(1, D)$ 是 $[1, D]$ 之间的一个随机整数,这种交叉可以保证 u_i 中至少有一个分量由 v_i 提供.

3) 选择操作: DE算法同ABC算法一样,也采用“贪婪”选择策略,以保证适应度更优的个体进入到下一代. 将经过变异与交叉操作后生成的新个体 u_i 与父代个体 x_i 进行比较,如果 u_i 的适应度优于 x_i ,则 u_i 替代 x_i 进入下一代,否则,维持 x_i 不变,直接进入下一代.

2.3 ABC和DE的相关性分析(Correlation analysis of ABC and DE)

ABC算法和DE算法都是通过引入随机个体执行变异操作产生新个体,不同的是在ABC算法中变异操作对原个体的扰动较小,产生的新个体继承了原个体较多特性;而在DE算法中的变异操作吸收了更多其他个体的信息,产生的新个体比原个体有较大的变异.在两种算法中,他们对个体的评估都同样采用“贪婪”选择的策略,所以他们之间存在一定的相关性.由ABC的更新式(1)和DE的更新式(4)比较可知,当 $F \in (0, 1)$, $i = r_1 = r_2$ 时,式(4)转化为式(1).由于ABC算法中变异操作只在某一维进行,所以可以把ABC算法看作是 $CR = 0$ 或只有一维进行了交叉操作的DE算法.

3 双种群差分蜂群算法(Bi-group differential artificial bee colony algorithm)

由于ABC和DE算法产生新个体的方式不同决定了他们在寻优时的效果也不同.所以本文在对两种算法特性进行分析的基础上,将他们的优势相融合,提出了一种双种群差分蜂群算法(BDABC),其主要思想是:把种群随机地等分成两组,每组采用不同的优化策略进行寻优.其中一组采用ABC算法,个体更新采用式(1),变异个体由自身和另外一个不相同的随机个体组成,这样有利于保持种群的多样性,具有较强的全局搜索能力;而另一组采用DE/best/1/bin策略,个体更新采用式(5),变异个体由当前最优个体作引导,局部搜索能力较强,收敛速度快.通过把两种变异策略混合,两个种群并行进化,从而保证了在维持种群多样性的同时加快算法的收敛速度,使算法在合理的计算复杂度下具有较高的全局收敛率.

3.1 基于反向学习的群体初始化(Population initialization based on the opposition-based learning)

基本ABC算法的初始群体是随机产生的,这种方式不能保证所产生的初始群体均匀地分布在解空间内,在一定程度上会限制算法的求解效率.为了使初始种群具有尽可能的多样性,均匀分布在搜索空间,本文采用基于反向学习的群体初始化^[17].首先随机生成初始解,然后为每个初始解产生相对应的反向解,最后对这两类解进行排序选择,把适应度较优的解作为初始种群,这将有助于求解效率的提高与解的质量的改善.

3.2 BDABC中选择策略的确定(The confirmation to selection strategy of ABC algorithm)

在BDABC算法中,跟随蜂选择食物源的选择策略采用锦标赛^[18]的方法,因为锦标赛选择只把适应值的相对值作为选择的标准,而且对适应值的正负

也没有要求,从而避免了超级个体对算法的影响,在一定程度上可以避免算法过早收敛和停滞现象的发生.

3.3 DE中相关参数的选择(Parameter selection of DE)

3.3.1 自适应缩放因子(Self-adaptive zoom factor)

在DE算法中缩放因子 F 用于控制差异向量的缩放程度.它的取值在很大程度上影响着进化过程的多样性和收敛速度.在基本DE算法中缩放因子 F 一般取固定值, F 取值较小时,种群的多样性低、收敛速度较快; F 取较大值时,种群多样性高但收敛速度较慢.为此,本文采取自适应的缩放因子,取值范围在 $[0, 1]$ 之间,当种群内的个体适应度趋于一致或者收敛于局部最优解时, F 值适当增大,增加群体多样性;而当群体适应度比较分散时, F 值适当减小,提高收敛速度.自适应缩放因子公式如下:

$$F_i = \begin{cases} 1, & \text{fitness}_{\text{worst}} = \text{fitness}_{\text{best}}, \\ 1 - \frac{\text{fitness}_{\text{worst}} - \text{fitness}_i}{\text{fitness}_{\text{worst}} - \text{fitness}_{\text{best}}}, & \\ \text{其他}, & \end{cases} \quad (7)$$

其中: fitness_i 表示当前第 i 个个体的适应值, $\text{fitness}_{\text{worst}}$ 表示当前种群中最差的适应值, $\text{fitness}_{\text{best}}$ 表示当前种群中最优的适应值.

3.3.2 自适应交叉概率(Self-adaptive crossover probability)

在基本的DE算法中,交叉概率 CR 一般取 $[0, 1]$ 之间的固定实数. CR 越大,则有利于局部搜索和加速收敛速率; CR 越小,则有利于保持种群的多样性和全局搜索能力. CR 的调整需要一定的经验,文献[19]通过对测试函数的大量数值模拟实验,确定 CR 的最佳取值范围为 $[0.4, 0.9]$.为此,本文采用了动态调整的交叉概率,根据算法的循环代数来动态地改变 CR 的值,使 CR 随着循环代数的增加逐渐增大,这样使算法在初期能提高全局搜索能力,而在后期有较好的局部搜索能力.自适应交叉概率公式如下:

$$CR(t) = \frac{t}{2 \cdot MCN} + 0.4, \quad (8)$$

其中: t 表示当前循环代数, MCN 表示最大循环代数.当 $1 \leq t \leq MCN$ 时, CR 在 $(0.4, 0.9]$ 之间变化.

3.4 交互学习(Interactive learning)

为了加快算法的收敛速度,本文让两组群体每隔一定的代数进行交互学习,通过设置参数 G 来控制两群体间交互学习的时机,即每当进化 G 代后,比较两个群体的最优适应值,用较好的那个群体的最优适应值个体替换另一个群体的最差适应值个体.这样通过他们之间的信息交流来均衡全局探索能力

和收敛速度之间的矛盾, 从而在维持种群多样性的同时加快收敛速度, 提高算法收敛于全局最优解的性能. 参数 G 的取值如太小, 两群体的个体会逐渐趋同, 易陷入早熟收敛; 太大则不利于两群体个体之间的交互学习. 为了更好的发挥两群体中较好个体的特性, 通过对本文所选择的测试函数仿真实验, 确定了 G 取值范围在 $[40, 60]$ 之间为宜, 因此 G 的取值对测试函数所代表的这类问题是具有一定普适性的.

交互学习操作的步骤如下(以求解最小化问题为例):

- 1) 分别找出两种群的最小适应值 Fitness1 和 Fitness2;
- 2) 如果 Fitness1 < Fitness2, 则用种群1的最小适应值个体替代种群2的最大适应值个体; 否则用种群2的最小适应值个体替代种群1的最大适应值个体;
- 3) 交互学习操作结束, 继续下一步操作.

3.5 BDABC算法流程(The main steps of BDABC)

算法步骤如下:

Step 1 按照反向学习策略初始化种群的解 $x_i (i = 1, \dots, n)$; 计算每个解 x_i 的适应值, 并对其进行分组;

Step 2 判断是否满足交互学习条件, 如满足则进行交互操作.

Step 3 分别在两子种群中按照式(1)和式(5)两种优化策略产生新解, 并且计算其适应值;

Step 4 如果新解的适应值好于 x_i , 则用其替换 x_i , 否则保留 x_i 不变;

Step 5 分别在两子种群中根据锦标赛选择策

略利用式(2)计算与 x_i 相关的概率值 P_i ;

Step 6 分别在两子种群中让跟随蜂按照 P_i 选择食物源(解), 并按两种优化策略产生新解, 计算其适应值;

Step 7 如果新解的适应度值好于 x_i , 则用其替换 x_i , 否则保留 x_i 不变;

Step 8 判断是否有要放弃的解, 如果存在, 则根据公式(3)随机产生一个新解来代替它;

Step 9 记录迄今为止最好的解;

Step 10 判断是否满足循环终止条件, 如满足输出最优结果, 否则返回 Step 2.

4 仿真实验与结果分析(The experiment of simulation and the analysis of results)

4.1 实验参数设置(The Setting of experimental parameters)

为了验证以上分析的有效性及改进的算法的性能, 将本文提到的BDABC算法与基本ABC算法进行了比较实验. 在仿真实验中本文选择了6个测试函数^[20], 并根据函数性质分为具有单个极小点(单模态)和多个局部极小点(多模态)两大类. 表1列出了这些测试函数的定义、取值范围和理论全局最优解. 其中, $f_1 \sim f_2$ 是单模态函数, 主要用来测试算法的寻优精度, 考察算法的执行性能; $f_3 \sim f_6$ 是复杂的非线性多模态函数, 具有许多局部极值点, 一般算法都较难找到全局最优值, 因此可用来检验算法的全局搜索的性能和避免早熟的能力. 在固定的迭代次数下, 本文通过对测试函数的最优值、最差值、平均值和方差的考察来对算法进行评估.

表1 6个标准测试函数
Table 1 Six benchmark functions

函数	方程	取值范围	理论全局最优解
Sphere	$f_1(x) = \sum_{i=1}^n x_i^2$	$[-100, 100]$	0
Quartic	$f_2(x) = \sum_{i=1}^n ix_i^4 + \text{random}[0, 1)$	$[-1.28, 1.28]$	0
Schwefel Problem 2.26	$f_3(x) = -\sum_{i=1}^n (x_i \sin(\sqrt{ x_i }))$	$[-500, 500]$	$-418.9829n$
Rastrigin	$f_4(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	$[-5.12, 5.12]$	0
Ackley	$f_5(x) = -20 \exp(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}) - \exp(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i) + 20 + e)$	$[-32, 32]$	0
Griewank	$f_6(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1$	$[-600, 600]$	0

在实验中, 两种算法的蜂群个数(colony size)均设为100, 其中引领蜂的个数(n_e)和跟随蜂的个数(n_o)均为50; 解的个数 $SN = n_e = n_o = 50$, 测试函数的维数 D 分别取50和100, 相应的最大迭代次数 MCN 分别为2000和5000; 在ABC中, limit=

$SN \cdot D$, 在BDABC算法中, limit=50, 两个子种群的个体数分别为50, 则解的个数 $SN = n_e = n_o = 25$, 交互学习参数 $G = 50$. 针对每个测试函数各算法均随机运行30次求其平均值. 各函数在不同维数下的最优值、最差值、平均值和方差的比较见

表2, 为了进一步比较算法的性能, 图1~6给出每个函数50维时平均适应值的进化过程曲线(100维的进化过程曲线略). 图中的横坐标表示循环代数, 纵坐标表示平均适应值.

表2 6个标准测试函数的仿真结果比较(50维)

Table 2 Comparison of experimental results of six benchmark functions(50 dimensions)

函数	维数	算法	均值	方差	最大值	最小值
f_1 Sphere	50	ABC	7.431086E-011	6.691496E-011	2.869498E-010	1.025880E-011
		BDABC	4.477788E-024	6.233437E-024	3.454583E-023	4.096191E-025
	100	ABC	5.353588E-013	1.062128E-012	5.008864E-012	1.102194E-014
		BDABC	1.255630E-033	1.300959E-033	5.119839E-033	1.880861E-034
f_2 Quartic	50	ABC	7.701566E-001	1.368055E-001	1.133232E+000	4.898284E-001
		BDABC	1.226310E-001	3.462654E-002	1.917625E-001	6.800217E-002
	100	ABC	2.476103E+000	3.520698E-001	3.164215E+000	1.853860E+000
		BDABC	2.500413E-001	5.214447E-002	4.017494E-001	1.822192E-001
f_3 Schwefel Problem 2.26	50	ABC	-2.046301E+004	1.043607E+002	-2.020742E+004	-2.064856E+004
		BDABC	-2.093728E+004	3.613201E+001	-2.083071E+004	-2.094914E+004
	100	ABC	-4.112410E+004	1.855057E+002	-4.081883E+004	-4.142370E+004
		BDABC	-4.160215E+004	1.512783E+002	-4.130610E+004	-4.189829E+004
f_4 Rastrigin	50	ABC	3.043434E-006	1.168673E-005	5.964192E-005	2.070522E-011
		BDABC	3.197442E-015	8.427937E-015	4.618528E-014	0.000000E+000
	100	ABC	9.305371E-013	1.544835E-012	6.243006E-012	1.088019E-014
		BDABC	1.369275E-016	2.837024E-016	1.443290E-015	0.000000E+000
f_5 Ackley	50	ABC	4.688896E-007	1.891135E-007	8.994045E-007	1.233645E-007
		BDABC	3.264174E-012	1.580259E-012	7.777778E-012	6.794565E-013
	100	ABC	3.097513E-009	6.536969E-010	4.569487E-009	2.118687E-009
		BDABC	3.408829E-013	9.451118E-014	5.870859E-013	2.353673E-013
f_6 Griewank	50	ABC	3.002776E-010	1.539993E-009	8.453016E-009	5.714318E-013
		BDABC	3.700743E-018	2.026981E-017	1.110223E-016	0.000000E+000
	100	ABC	8.171797E-013	1.280251E-012	4.397926E-012	5.551115E-015
		BDABC	7.216450E-017	1.261999E-016	4.440892E-016	0.000000E+000

4.2 实验结果分析(The analysis of experimental results)

1) 由表2和图1, 2可知, 对于复杂的单模态函数Sphere, ABC算法的收敛速度较慢, 而BDABC算法的收敛速度明显快于ABC算法, 几乎线性递减的收敛到最优解. 对于单模态函数Quartic, 两种算法在进化早期收敛速度很快, 但在进化中后期收敛速度明显减慢, 但BDABC依然优于ABC算法, 且BDABC的稳定性也好于ABC算法.

2) 多模态函数Schwefel Problem 2.26的取值范围为[500,500], 当维数是50时, 它在点(420.9687, ..., 420.9687)处取得理论最优值-20949.145, 该函数的最优值位于搜索空间的边界附近. 由表2和图3可知, BDABC算法的准确性明显优于ABC算法. 由此可以看出, BDABC算法对于最优解位于搜索空间边缘的情况, 它也能较

好的找到最优解.

3) 多模态函数Rastrigin, Ackley, Griewank均是复杂的非线性全局优化问题, 主要用来测试算法的全局搜索性能, 从图4~6可以看出, BDABC算法的收敛速度和精度均高于ABC算法. 由于这些函数自身的特性, 他们很容易陷入局部最优而搜索停滞, 但BDABC算法对参数的动态调整, 使其尽可能的跳出局部最优, 提高了算法的收敛精度. 对于Ackley函数, 几乎线性递减的收敛到最优解, 对于Rastrigin函数, 收敛效果更加明显. 由此可见, BDABC算法具有良好的全局搜索性能和较快的搜索速度.

通过以上分析可知, BDABC比基本ABC在收敛精度和收敛速度上均有非常显著的提高, 其性能明显优于基本ABC算法, 并且随着维数的增加, BDABC也保持了较好的有效性和稳健性.

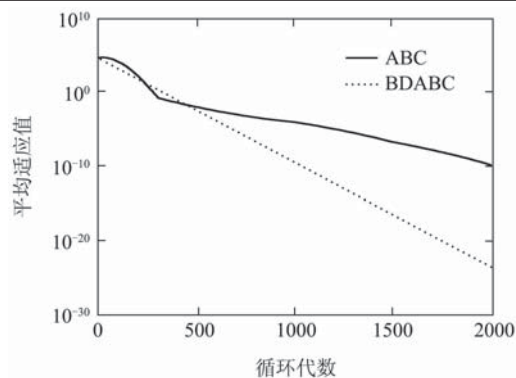


图1 Sphere Function进化过程曲线

Fig. 1 The progress curve of Sphere Function

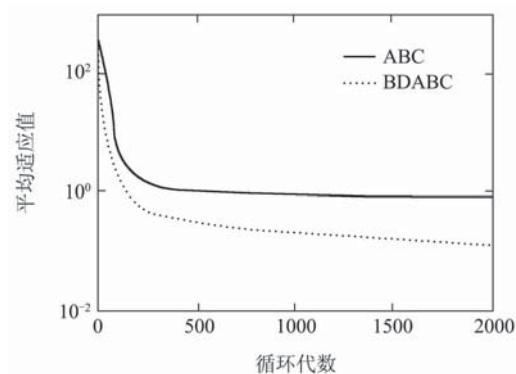


图2 Quartic Function进化过程曲线

Fig. 2 The progress curve of Quartic Function

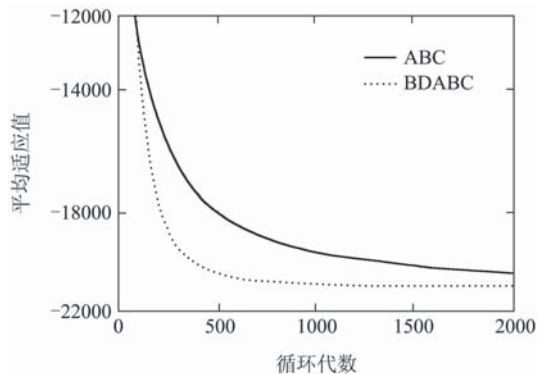


图3 Schwefel Problem 2.26进化过程曲线

Fig. 3 The progress curve of Schwefel Problem 2.26

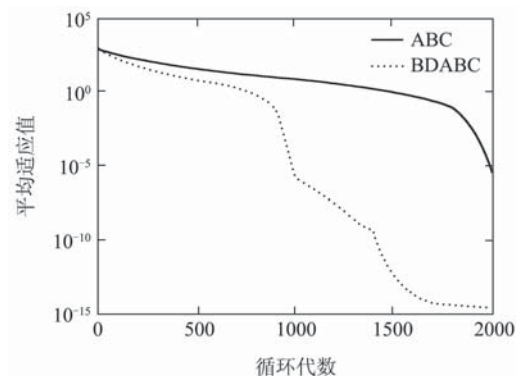


图4 Rastrigin Function进化过程曲线

Fig. 4 The progress curve of Rastrigin Function

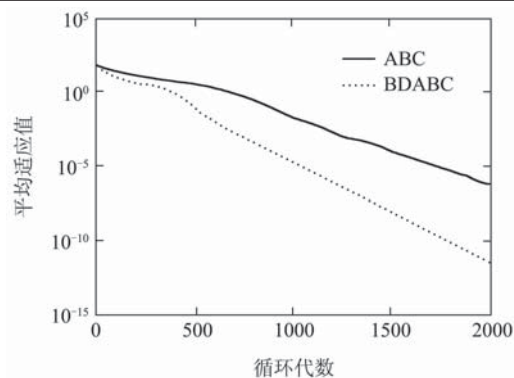


图5 Ackley Function进化过程曲线

Fig. 5 The progress curve of Ackley Function

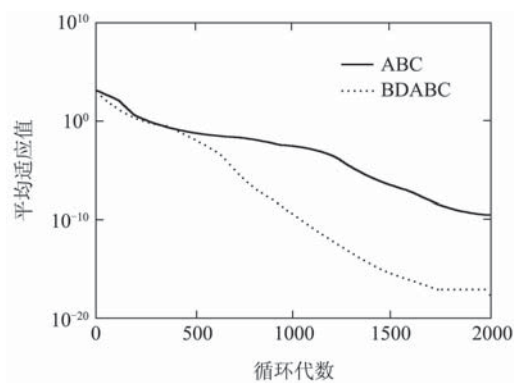


图6 Griewank Function进化过程曲线

Fig. 6 The progress curve of Griewank Function

5 结论(Conclusion)

本文针对ABC算法的不足, 结合差分进化算法, 提出了一种双种群差分蜂群算法(BDABC). 该算法主要有如下特点: 1) 通过基于反向学习的初始化策略, 提高了种群的多样性及解的质量; 2) 利用ABC和DE两种算法各自的变异优点, 使改进的BDABC算法在解的探索和开采两个方面保持了较好的平衡; 3) 引入交互学习来实现两种种群之间的信息交流, 提高算法的收敛速度. 通过对较复杂的全局优化问题的仿真实验表明, 该算法利用两群体并行进化, 在保证种群多样性的同时, 避免了早熟收敛, 全局搜索与跳出局部最优的能力均强于基本ABC算法, 而且这种良好的性能在高维的情况下也有很好的表现.

参考文献(References):

- [1] KARABOGA D. An idea based on honey bee swarm for numerical optimization[R]. Kayseri: Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.
- [2] KARABOGA D, BASTURK B. A powerful and efficient algorithm for numerical function optimization: artificial bee colony(ABC) algorithm[J]. *Journal of Global Optimization*, 2007, 39(3): 459 – 471.
- [3] KARABOGA D, BASTURK B. On the performance of artificial bee colony(abc) algorithm[J]. *Applied Soft Computing*, 2008, 8(1): 687 – 697.

- [4] KARABOGA D, BASTURK B. Artificial bee colony(ABC) optimization algorithm for solving constrained optimization problems[C] // *LNCS: Advances in Soft Computing: Foundations of Fuzzy Logic and Soft Computing*. Berlin: Springer-Verlag, 2007, 4529: 789 – 798.
- [5] KARABOGA D, AKAY B B. Artificial bee colony algorithm on training artificial neural networks[C] // *2007 IEEE 15th Signal Processing and Communications Applications Conference*. New York: IEEE, 2007: 818 – 821.
- [6] KARABOGA D, AKAY B B, OZTURK C. Artificial bee colony(ABC) optimization algorithm for training feed-forward neural networks[C] // *LNCS: Modeling Decisions for Artificial Intelligence*. Berlin: Springer-Verlag, 2007, 4617: 318 – 319.
- [7] KARABOGA N. A new design method based on artificial bee colony algorithm for digital IIR filters[J]. *Journal of the Franklin Institute*, 2009, 346(4): 328 – 348.
- [8] SRINIVASA RAO R, NARASIMHAM S V L, RAMALINGARAJU M. Optimization of distribution network configuration for loss reduction using artificial bee colony algorithm[J]. *International Journal of Electrical Power and Energy Systems Engineering*, 2008, 1(2): 709 – 715.
- [9] 胡中华, 赵敏. 基于人工蜂群算法的机器人路径规划[J]. 电焊机, 2009, 39(4): 93 – 96.
(HU Zhonghua, ZHAO Min. Research on robot path planning based on ABC algorithm[J]. *Electric Welding Machine*, 2009, 39(4): 93 – 96.)
- [10] 李端明, 程八一. 基于人工蜂群算法求解不同尺寸工件单机批调度问题[J]. 四川大学学报(自然科学版), 2009, 46(3): 657 – 662.
(LI Duanming, CHENG Bayi. Artificial bee colony algorithm for scheduling a single batch-processing machine with non-identical job sizes[J]. *Journal of Sichuan University(Natural Science Edition)*, 2009, 46(3): 657 – 662.)
- [11] SINGH A. An artificial bee colony algorithm for the leaf-constrained minimum spanning tree problem[J]. *Applied Soft Computing*, 2009, 9(2): 625 – 631.
- [12] TSAI P W, PAN J S, LIAO B Y, et al. Enhanced artificial bee colony optimization[J]. *International Journal of Innovative Computing, Information and Control*, 2009, 5(12): 5081 – 5092.
- [13] 丁海军, 李峰磊. 蜂群算法在TSP问题上的应用及参数改进[J]. 中国科技信息, 2008, (3): 241 – 243.
(DING Haijun, LI Fenglei. Bee colony algorithm for TSP problem and parameter improvement[J]. *China Science and Technology Information*, 2008, (3): 241 – 243.)
- [14] 康飞. 改进人工蜂群算法及其在反演分析中的应用[J]. 水电能源科学, 2009, 27(1): 126 – 129.
(KANG Fei. Improved artificial bee colony algorithm and its application in back analysis[J]. *Water Resources and Power*, 2009, 27(1): 126 – 129.)
- [15] STORN R, PRICE K. Differential evolution-a simple and efficient heuristic for global optimization over continuous space[J]. *Journal of Global Optimization*, 1997, 11(4): 341 – 359.
- [16] MENDES R, MOHAIS A S. DynDE: a differential evolution for dynamic optimization problems[C] // *2005 IEEE Congress on Evolutionary Computation*. New York: IEEE, 2005: 2808 – 2815.
- [17] 吴昱, 李元香, 徐星. 基于群智能的新型反向混合差分进化算法[J]. 小型微型计算机系统, 2009, 5(4): 903 – 907.
(WU Yu, LI Yuanxiang, XU Xing. Novel oppositional hybrid differential evolution algorithm based on swarm intelligence[J]. *Journal of Chinese Computer Systems*, 2009, 5(4): 903 – 907.)
- [18] BAO L, ZENG J C. Comparison and analysis of the selection mechanism in the artificial bee colony algorithm[C] // *2009 9th International Conference on Hybrid Intelligent Systems*. Los Alamitos, CA: IEEE Computer Society, 2009: 411 – 416.
- [19] 毛润宇, 王小平, 薛小平. 一种自适应差分演化算法[J]. 计算机应用与软件, 2008, 25(12): 7 – 8.
(MAO Runyu, WANG Xiaoping, XUE Xiaoping. An adaptive differential evolution algorithm[J]. *Computer Applications and Software*, 2008, 25(12): 7 – 8.)
- [20] YAO X, LIU Y, LIN G M. Evolutionary programming made faster[J]. *IEEE Transactions on Evolutionary Computation*, 1999, 3(2): 82 – 102.

作者简介:

暴 励 (1975—), 男, 主要研究方向为智能计算, E-mail: libao@sina.com;

曾建潮 (1963—), 男, 博士, 教授, 博士生导师, 主要研究方向为复杂系统建模、仿真与优化、智能计算等, E-mail: zengjianchao@263.net.