

8-13-2020

Adaptive Quick Artificial Bee Colony Algorithm Based on Opposition Learning

Xiaojian Yang

College of Electronics and Information Engineering, Nanjing Tech University, Nanjing 211816, China;

Yiwei Dong

College of Electronics and Information Engineering, Nanjing Tech University, Nanjing 211816, China;

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the Artificial Intelligence and Robotics Commons, Computer Engineering Commons, Numerical Analysis and Scientific Computing Commons, Operations Research, Systems Engineering and Industrial Engineering Commons, and the Systems Science Commons

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation.

Adaptive Quick Artificial Bee Colony Algorithm Based on Opposition Learning

Abstract

Abstract: *On the basis of analyzing such shortcomings of the artificial bee colony algorithm (ABC) as slow convergence, low convergence precision and premature convergence, the opposition-learning adaptive quick artificial bee colony algorithm (OAQABC) was proposed. A new step size was proposed, which made the around food source parameter of quick artificial bee colony algorithm (QABC) adaptive, and combined the opposition-based learning to improve the employed bee phase. The experimental results show that OAQABC has better performance than basic ABC and QABC. Also the optimization performance of OAQABC is better than particle swarm optimization (PSO) algorithm and Cuckoo Search (CS) algorithm obviously in the experiment.*

Keywords

artificial bee colony (ABC), adaptive, opposition-based learning, optimization

Recommended Citation

Yang Xiaojian, Dong Yiwei. Adaptive Quick Artificial Bee Colony Algorithm Based on Opposition Learning[J]. Journal of System Simulation, 2016, 28(11): 2684-2692.

基于反向学习的自适应快速人工蜂群算法

杨小健, 董毅伟

(南京工业大学电子与信息工程学院, 南京 211816)

摘要: 针对人工蜂群算法收敛速度慢、收敛精度不高、易陷入局部收敛等缺点, 提出一种基于反向学习的自适应快速人工蜂群算法。研究出一种新型自适应步长, 使得快速人工蜂群算法跟随蜂搜索阶段的周边食物源参数自适应化, 并结合反向学习策略来改进引领蜂搜索阶段。在保证收敛速度的前提下进一步提高解的全局性。在仿真实验中加入粒子群算法和布谷鸟搜索算法的函数优化实验。对标准函数的仿真结果表明, 基于反向学习的自适应快速人工蜂群算法相较于标准人工蜂群算法和快速人工蜂群算法在优化性能方面得到明显改善, 其优化性能也明显优于粒子群算法和布谷鸟搜索算法。

关键词: 人工蜂群算法; 自适应; 反向学习; 优化

中图分类号: TP301.6

文献标识码: A

文章编号: 1004-731X (2016) 11-2684-09

DOI: 10.16182/j.issn1004731x.joss.201611006

Adaptive Quick Artificial Bee Colony Algorithm Based on Opposition Learning

Yang Xiaojian, Dong Yiwei

(College of Electronics and Information Engineering, Nanjing Tech University, Nanjing 211816, China)

Abstract: On the basis of analyzing such shortcomings of the artificial bee colony algorithm (ABC) as slow convergence, low convergence precision and premature convergence, the opposition-learning adaptive quick artificial bee colony algorithm (OAQABC) was proposed. A new step size was proposed, which made the around food source parameter of quick artificial bee colony algorithm (QABC) adaptive, and combined the opposition-based learning to improve the employed bee phase. The experimental results show that OAQABC has better performance than basic ABC and QABC. Also the optimization performance of OAQABC is better than particle swarm optimization (PSO) algorithm and Cuckoo Search (CS) algorithm obviously in the experiment.

Keywords: artificial bee colony (ABC); adaptive; opposition-based learning; optimization

引言

近几年来优化问题在各个领域的应用都层出不穷, 传统的数值分析与计算在这些优化问题的解决上已经力不从心。而人们也已经在启发式算法中找到求解复杂优化问题的方法, 诸如遗传算法、粒

子群算法、差分进化算法等。这些算法在很多复杂的优化问题上都提供了比传统数值分析方法更好的优化解。

人工蜂群算法 (Artificial Bee Colony Algorithm, 简称 ABC) 是 Karaboga 提出的一种新的启发式算法, 并在 2007 年形成完整理论用来解决函数数值优化问题^[1]。该算法模拟了蜜蜂群在搜索附近蜜源时的行为, 并利用不同种类蜜蜂的分工合作快速有效地寻找到最优解。同时, ABC 算法已经成功应用于多种优化问题, 如 ABC 算法在约束条件优化中的应用^[2]、在混合流水线调度方面



收稿日期: 2015-02-12

修回日期: 2015-05-11;

作者简介: 杨小健(1963-), 男, 江苏南京, 博士, 教授, 研究方向为工业控制和优化; 董毅伟(1990-), 男, 江苏南京, 硕士生, 研究方向为人工智能与优化。

<http://www.china-simulation.com>

• 2684 •

的优化^[3]、在分布式电源规划^[4]、炼钢连铸调度的应用^[5]等。

为了提高 ABC 算法的寻优能力, 许多学者都针对 ABC 算法的缺点进行改进。Akay B 等^[6]在标准蜂群算法变异过程中加入扰动因子和比例因数, 使得改进算法比原算法在复杂的多维函数优化中得到更好的效果; Brajevic I 等^[7]改进了侦查蜂搜索阶段, 相较于原算法更简单、可靠、健壮, 更易于解决工程领域方面的问题。Vitorino L N 等^[8]将 ABC 算法与粒子群算法相结合, 暴励等^[9]将 ABC 算法与混沌优化思想结合并缩减了算法搜索空间, 二者都使得改进算法有了更快的收敛速度及精度。高卫峰等^[10]受差分进化算法启发用新的搜索方程提高算法的优化性能、优化效率及鲁棒性。而 ABC 算法的原作者 Karaboga 在 2014 年又根据真实蜜蜂采集蜂蜜的行为, 进一步改进了 ABC 算法, 并提出一种快速人工蜂群算法^[11](Quick Artificial Bee Colony Algorithm, 简称 QABC)。QABC 算法引入了一个周边食物源的新概念, 根据周边食物源参数来判别出周边食物源种群的大小, 从而提高蜜蜂采集效率。QABC 算法在收敛速度方面明显优于传统 ABC 算法, 该算法的周边食物源参数是给定的。当选择较好的周边食物源参数后算法会体现出良好的收敛速度, 但算法的整体精度并没有得到提高, 甚至相较原先的 ABC 算法有所下降。在文献[12]中可以看出 ABC 算法的一项改进是将反向学习机制运用到了侦查蜂搜索阶段, 用反向学习生成的解替换随机解来避免算法过早陷入局部最优。本文在研究中发现侦查蜂阶段引入反向学习并不能很好的防止算法的早熟, 因为侦查蜂阶段原本生成的随机解可以扩大整个算法的种群多样性, 从而有效避免早熟。而在引领蜂阶段引入反向学习机制不仅可以避免算法的早熟, 还能在一定程度上提高算法的运行精度。所以, 本文在 QABC 算法的基础上将周边食物源参数自适应化并结合反向学习机制优化引领蜂搜索阶段, 提出一种基于反向学习的自适应快速蜂群算法 (opposition-learning

adaptive quick artificial bee colony algorithm, 简称 OAQABC)。在标准函数实验中, 验证了 OAQABC 算法优于 ABC 算法和 QABC 算法的良好优化性能。

1 标准人工蜂群算法

标准人工蜂群算法是对蜜蜂寻找食物源的搜索过程进行模拟。其中, 蜂群主要包括两大类: 雇佣蜂和非雇佣蜂。而雇佣蜂又名引领蜂, 非雇佣蜂包含跟随蜂和侦查蜂。蜂群算法的搜索原理是在这三种蜂群的相互协调中进行的。引领蜂将食物源的相关信息与其他蜜蜂分享。跟随蜂在得到引领蜂的信息后负责确定食物源的优劣并判断是否跟随引领蜂采蜜。而侦查蜂则是在一些食物源枯竭时负责探索新的食物源, 并转变为引领蜂。具体步骤如下:

首先是初始化参数阶段, 包括确定蜂群数量 CZ 、确定最大迭代次数 MCN 、确定搜索上限 u 下限 l 、设置控制参数 $Limit$ 以及在搜索空间中根据公式(1)随机生成初始解等。

$$x_{i,j} = l_{i,j} + \text{rand}(0,1) * (u_{i,j} - l_{i,j}) \quad (1)$$

式中: $i \in \{1, 2, \dots, N\}$, $j \in \{1, 2, \dots, D\}$; N 是食物源的个数, 即解的个数, 一般取 $CZ/2$; D 是所要优化问题的维数。

初始化后, 整个蜂群将进行引领蜂、跟随蜂、侦查蜂搜索阶段, 重复循环这三个阶段直至达到最大迭代次数 MCN 。引领蜂阶段根据公式(2)产生一个新解, 再根据适应度优劣选择保留的解。

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

式中: $k \in \{1, 2, \dots, SN\}$ 且 $k \neq i$; $\phi_{i,j}$ 为 $[-1, 1]$ 之间的随机数。

之后, 引领蜂将解的信息与跟随蜂分享, 跟随蜂根据公式(3)计算每个解的选择概率 P_i 。

$$P_i = \frac{fit_i}{\sum_{k=1}^{SN} fit_k} \quad (3)$$

式中: fit 表示解的适应度值, 如公式(4)所示。

$$fit_i = \begin{cases} \frac{1}{1 + f_i(x)} & f_i > 0 \\ 1 + abs(f_i(x)) & f_i \leq 0 \end{cases} \quad (4)$$

然后在 $[0,1]$ 内随机产生一个数,如果 P_i 大于该随机数,则按公式(2)生成新解,并按公式(5)进行贪婪选择,判断是否保留新解。

$$v_{i,j} = \begin{cases} v_{i,j} & fit(v_{i,j}) > fit(x_{i,j}) \\ x_{i,j} & fit(v_{i,j}) \leq fit(x_{i,j}) \end{cases} \quad (5)$$

在所有跟随蜂完成搜索之后,侦查蜂将那些经过 Limit 次循环仍没有被更新的解舍弃,并根据公式(1)生成一个新解替换它。

2 基于反向学习快速人工蜂群算法(OAQABC)

2.1 快速人工蜂群算法

快速人工蜂群算法是 Karaboga 在 2014 年最新提出的一种关于人工蜂群算法的改进。在真实的蜂群采蜜行为中,跟随蜂根据引领蜂所分享的信息找到蜂蜜,并且在蜜源附近寻找是否有更优质的蜜源,再决定所要采集的蜜源。换言之,跟随蜂可能会与引领蜂所采集的蜜源不同。这点在标准人工蜂群算法中没有体现。因此,QAABC 算法主要改进如下:

在跟随蜂搜索阶段,如果 P_i 大于随机数则在生成新解之前,先计算 x_i 与其他解 x_m 之间的欧式距离 $d(i,m)$,在根据公式(6)算出 md_m ,即平均欧式距离。

$$md_i = \frac{\sum_m^{SN} d(i,m)}{SN - 1} \quad (6)$$

若 $d(i,m) \leq r \times md_i$ 则确定 x_m 是 x_i 的附近的食物源,并将其加入周边食物源种群 N,其中 r 为周边食物源参数,在 N 中再依据公式(7)决定将要采集的食物源。最后,根据公式(8)生成新解。

$$fit(x_{Ni}^{best}) = \max(fit(x_{Ni}^1), fit(x_{Ni}^2), \dots, fit(x_{Ni}^S)) \quad (7)$$

$$v_{Ni,j}^{best} = x_{Ni,j}^{best} + \varphi_{i,j}(x_{Ni,j}^{best} - x_{k,j}) \quad (8)$$

OABC 最主要的改进就引入周边食物源参数,并在跟随蜂搜索阶段建立新的食物源候选解来加速优化过程。相对于标准人工蜂群算法,快速人工蜂群算法有较快的收敛速度。但是也存在自身的不足,虽然在前期 QAABC 算法的收敛速度明显加快,但是后期容易陷入局部最优,收敛速度减慢,甚至不如 ABC 算法。为了克服 QAABC 算法的缺点,本文提出了 OAQABC 算法。

2.2 OAQABC 原理

在蜂群算法初始化阶段,初始种群的候选解都是随机生成的。那么,随机生成的解会直接影响到之后算法执行中的收敛速度。如果在初始化阶段引入反向学习机制^[13-14],会对初期收敛有一点改进,但对于整体收敛速度并没有实质性改进。文献[12]将反向学习机制引用到搜查蜂搜索阶段,而实际效果并不是很理想。因为相比于反向学习生成的解,随机解更能增加种群的多样性。所以,本文将反向学习引用到引领蜂搜索阶段,在引领蜂产生新解之后,再为每个解产生相对应的反向解,并保留较好的解作为本次循环最终解,减少初始种群的随机性所带来的不稳定性。

在概率学中,一个解有 50%的几率不如它自身的反向解优秀。那么,在引领蜂搜索阶段引入反向学习机制可以使函数的收敛更加快速和精确。反向学习的定义如下:

$$\tilde{x}_{i,j} = a_i + b_i - x_{i,j} \quad (9)$$

$x_{i,j}$ 是在候选种群 P 第 i 组中的其中一维的分量, $j \in \{1, 2, \dots, D\}$, $x_{i,j} \in \{a_i, b_i\}$, a_i 和 b_i 分别表示在第 i 组候选解的上限和下限。根据公式(9)可将每一维的反向解分量 $\tilde{x}_{i,j}$ 计算出,得到反向种群 OP。再根据 P 和 OP 适应度值的大小选定最终种群。由于是在引领蜂搜索阶段引入的反向种群机制,所以引领蜂所搜得的解会非常优秀,这会直接影响到跟随蜂的搜索效率,使得算法整体的搜索速度加快。但是在后期引领蜂阶段若是一直采用反向学习方

法去加速迭代反而会影响到整个种群的多样性。所以本算法将在前期迭代中使用反向学习机制加速算法的收敛速度, 在后期让引领蜂自由搜索, 增强整体种群多样性。

在 QABC 中周边食物源参数的值会直接影响到整个算法的收敛速度。当 r 的值为 0 时, QABC 会直接转化为 ABC; 当 r 的值很大时, QABC 虽然前期下降很快, 但后期非常容易陷入局部最优。由于算法多样性的缺失, QABC 在后期的搜索性能甚至不如 ABC。

所以, 为了弥补 QABC 的缺陷, 本文在 ABC 算法的跟随蜂阶段提出了自适应步长来优化 QABC 算法。

$$\omega_1 = \frac{\exp(MCN + 1 - iter)}{\exp(MCN)} \quad (10)$$

$$\omega_2 = \frac{abs(\text{fit}^{\text{best}} - \text{fit}^{\text{worst}})}{abs(\text{fit}_i - \text{fit}^{\text{worst}})} \quad (11)$$

$$d = \omega_1 \omega_2 \quad (12)$$

式中: fit_i 表示经过跟随蜂概率选择公式(3)所选出的一组候选解的适应度值; fit^{best} 和 $\text{fit}^{\text{worst}}$ 分别表示所有候选解中的最优适应度值和最差适应度值, 若 fit_i 为最差适应度值则用次差适应度值替换; MCN 表示最大迭代次数; $iter$ 表示当前迭代次数; ω_1 表示稳值项; ω_2 表示扰动项; d 表示自适应步长。

在自适应步长 d 中, ω_1 使得原本的 QABC 可以在前期拥有较快的收敛速度, 很快找到比较优秀的解。在算法运行的后期, 因为 ω_1 的减小使得 QABC 算法可以逐渐向 ABC 算法转变, 从而增加算法的多样性, 加强算法后期的勘探能力。 ω_2 可以有效防止算法早熟, 当算法在一个局部最优解附近徘徊时, ω_2 使得自适应步长 d 维持在一个较低的值, 增加 OAQABC 算法跳出局部最优的概率。根据上述方法来使 QABC 算法自适应化, 并使得自适应化的 QABC 算法与反向学习相结合得到 OAQABC 算法。

2.3 OAQABC 算法流程

算法流程如下:

Step 1 初始化 OAQABC 的各项参数。

Step 2 按式(2)变异产生新解 $v_{i,j}$, 保留适应度好的解。

Step 3 若 $iter < \frac{MCN}{2}$, 则跳转到 Step4, 否则跳转到 Step5。

Step 4 按式(9)反向学习产生反向解 $\tilde{x}_{i,j}$, 按式(5)选择保留 $x_{i,j}$ 或 $\tilde{x}_{i,j}$ 。

Step 5 按式(3)计算出每个候选解的 P_i , 并产生随机数 $a \in (0,1)$, 若满足 $P_i > a$ 则跳转到 Step6, 否则跳转到 Step10。

Step 6 按式(12)计算出自适应步长 d , 并根据 $d(i,m) \leq d \times r \times md_i$ 确定周边种群 N 。

Step 7 按式(7)找出需要变异的候选解 $x_{Ni,j}^{\text{best}}$, 并按式(8)进行变异 $v_{Ni,j}^{\text{best}}$ 。

Step 8 按式(5)选择保留 $x_{Ni,j}^{\text{best}}$ 或 $v_{Ni,j}^{\text{best}}$, 若保留 $v_{Ni,j}^{\text{best}}$ 则跳转到 Step9, 否则跳转到 Step10。

Step 9 将该候选解对应的 $Limit$ 值置 0, 跳转到 Step11。

Step 10 将该候选解对应的 $Limit$ 值加 1。

Step 11 若经过 $Limit$ 次循环, 候选解仍没被替换, 则按式(1)初始化候选解。

Step 12 若算法达到最大迭代次数 MCN 或最优解达到精度要求, 则输出最优解, 并结束算法。

3 实验仿真与分析

3.1 实验参数设置

为了验证文中所提及的改进算法的性能, 本文将 ABC, QABC, OAQABC 算法进行对比实验。

在仿真实验中选择了 6 个具有代表性的测试函数: Sphere 函数、Schwefel 函数、Rosenbrock 函数、Rastrigin 函数、Griewank 函数、Ackley 函数。Sphere 函数是比较简单的单峰函数, 比较容易

搜索到最优值。Schwefel 是比较复杂的单峰函数，Rosenbrock 函数是很难找到最优值的多峰函数。多模态函数 Rastrigin、Ackley，Griewank 是复杂的非线性全局优化函数，这对于测试算法的全局搜索性非常好。表 1 列出了这些函数的维数、取值范围、理论最优解以及函数的定义。

在实验中，ABC 算法、QABC 算法、OAQABC 算法的蜜蜂种群数 CS 均设为 50，其中引领蜂和跟随蜂的个数均为 25；所有测试函数的维数如表 1 所示均为 30；Limit 值均设置为 750；QABC 和

OAQABC 算法的周边食物源参数均设为 0.5；所有测试函数的最大迭代次数 MCN 设置为 2000。详细参数设置如表 2 所示。

针对每个测试函数各个算法均运行 20 次并求其平均值、标准差。最优值、最差值、平均值以及标准差的比较见表 3~8 所示。为了方便对比不同蜂群算法之间收敛速率的差异，图 1~6 给出了 ABC 算法、QABC 算法和 OAQABC 算法在 6 种不同测试函数下的进化过程曲线。图中横坐标为迭代次数，纵坐标为目标函数值。

表 1 标准测试函数

测试函数	维数	取值范围	理论最优值	函数定义
Sphere	30	[-100,100]	0	$f(x) = \sum_{i=1}^D x_i^2$
Schwefel	30	[-500,500]	-12 569.5	$f(x) = \sum_{i=1}^D -x_i \sin(\sqrt{ x_i })$
Rosenbrock	30	[-30,30]	0	$f(x) = \sum_{i=1}^{D-1} 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2$
Rastrigin	30	[-5.12,5.12]	0	$f(x) = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10)$
Griewank	30	[-600,600]	0	$f(x) = \frac{1}{4000} (\sum_{i=1}^D (x_i - 100)^2) - (\prod_{i=1}^D (\frac{x_i - 100}{\sqrt{i}})) + 1$
Ackley	30	[-32,32]	0	$f(x) = 20 + e - 20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i)\right)$

表 2 ABC、QABC、OAQABC 参数设置

算法	蜜蜂种群数(CS)	引领蜂个数	跟随蜂个数	函数维数	Limit 值	周边食物源参数
ABC	50	25	25	30	750	/
QABC	50	25	25	30	750	0.5
OAQABC	50	25	25	30	750	0.5

表 3 Sphere 函数仿真比较结果

算法	平均值	标准差	最优值	最差值
ABC	6.517 7E-16	1.131 6E-16	5.147 2E-16	7.440 1E-16
QABC	1.014 8E-15	1.178 4E-16	8.865 5E-16	1.182 3E-15
OAQABC	5.116 0E-16	8.765 8E-17	3.933 4E-16	6.576 5E-16

表 4 Schwefel 函数仿真比较结果

算法	平均值	标准差	最优值	最差值
ABC	-1 2569.5	0	-1 2569.5	-1 2569.5
QABC	-1 2569.5	0	-1 2569.5	-1 2569.5
OAQABC	-1 2569.5	0	-1 2569.5	-1 2569.5

表 5 Rosenbrock 函数仿真比较结果

算法	平均值	标准差	最优值	最差值
ABC	0.887 8	0.784 0	0.117 2	2.278 4
QABC	1.943 0	0.987 9	0.452 2	3.377 6
OAQABC	0.790 9	0.589 2	0.107 2	1.579 0

表 6 Rastrigin 函数仿真比较结果

算法	平均值	标准差	最优值	最差值
ABC	1.205 6E-08	3.118 7E-08	1.307 4E-12	9.455 9E-08
QABC	5.140 3E-08	1.097 5E-07	4.064 3E-11	3.372 6E-07
OAQABC	2.848 3E-11	3.195 3E-11	3.690 9E-13	8.816 4E-11

表 7 Griewank 函数仿真比较结果

算法	平均值	标准差	最优值	最差值
ABC	7.574 6E-11	1.502 2E-10	1.576 5E-13	3.761 9E-10
QABC	1.576 9E-03	3.136 1E-03	2.345 4E-06	7.849 3E-03
OAQABC	2.027 5E-13	2.108 8E-13	1.711 5E-14	5.873 1E-13

表 8 Ackley 函数仿真比较结果

算法	平均值	标准差	最优值	最差值
ABC	8.014 3E-09	4.779 6E-09	3.916 9E-09	1.7986 E-08
QABC	6.044 5E-05	3.316 4E-05	4.763 0E-06	1.017 0E-04
OAQABC	3.293 5E-09	2.225 1E-09	5.195 21-10	7.337 5E-09

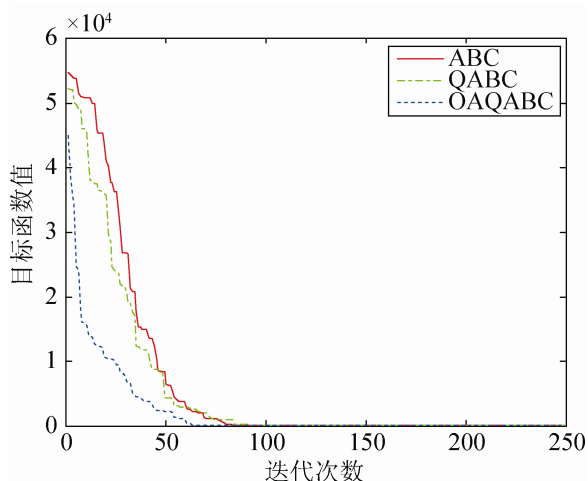


图 1 Sphere 函数进化曲线

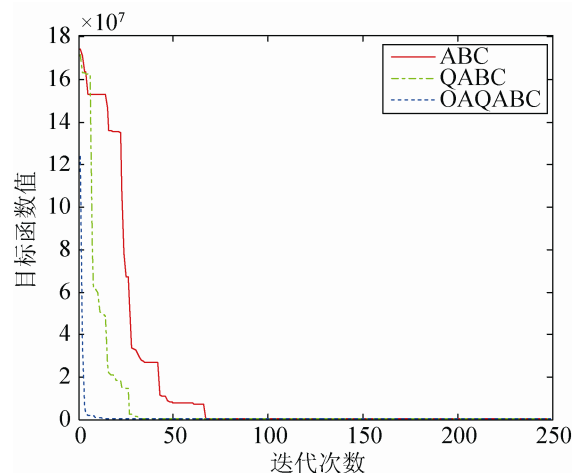


图 3 Rosenbrock 函数进化曲线

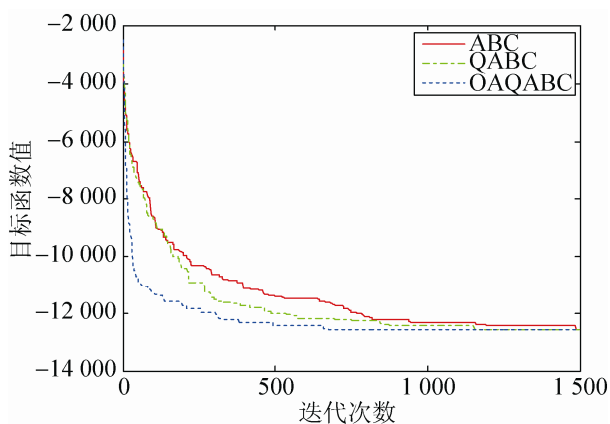


图 2 Schwefel 函数进化曲线

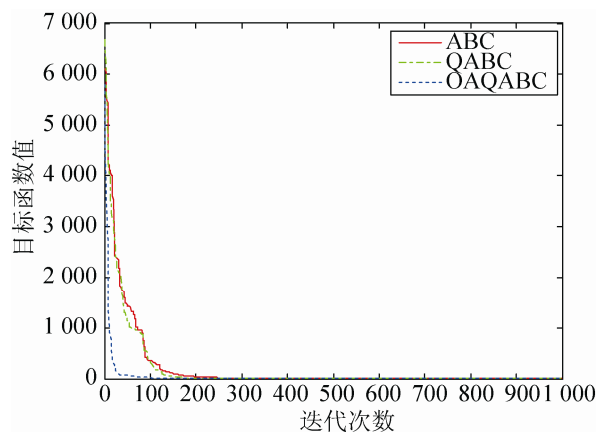


图 4 rastrigin 函数进化曲线

<http://www.china-simulation.com>

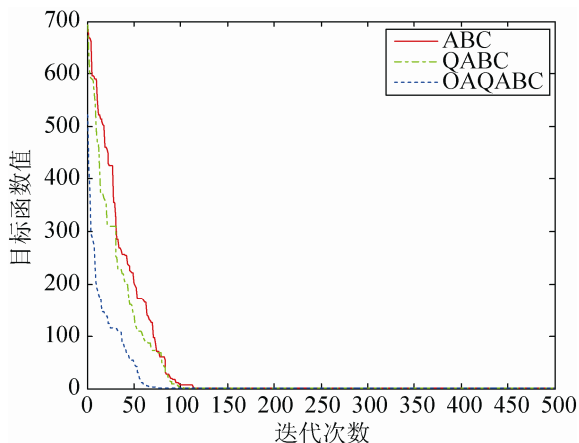


图 5 Griewank 函数进化曲线

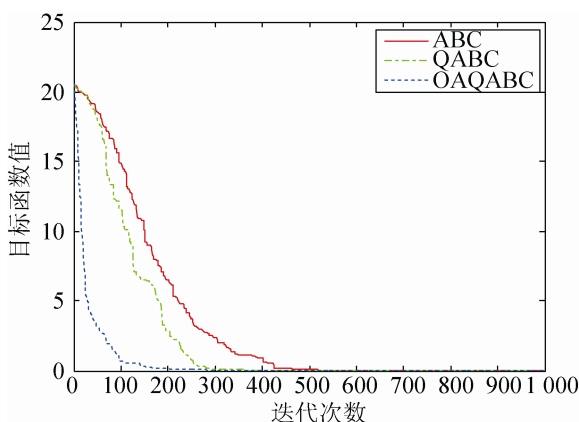


图 6 Ackley 函数进化曲线

PSO 算法是近几年比较热门的算法，而 CS 算

法是近一段时间新兴的群智能优化算法。为了更好的验证 OAQABC 算法在群智能算法中的优化性能，特此在仿真实验中测试粒子群算法^[15](Particle Swarm Optimization, 简称 PSO)和布谷鸟搜索算法^[16](Cuckoo Search, 简称 CS)的优化性能。

PSO 算法的种群大小设置为 20；惯性权重下限设置为 0.7；惯性权重上限设置为 1；速度下限设置为 0；速度上限设置为 2。详细参数设置如表 9 所示。

CS 算法的鸟巢的数量设置为 30；发现蛋的概率设置为 0.25；函数维数设置均为 30。详细参数设置如表 10 所示。

每个测试函数运行 20 次，得出 PSO 和 CS 算法优化结果，包括平均值、标准差、最优值和最差值。PSO 和 CS 算法对于函数优化结果如表 11 所示。

表 9 PSO 算法参数设置

算法	种群数	惯性权重	惯性权重	速度	速度
		下限	上限	下限	上限
PSO	20	0.7	1	0	2

表 10 CS 算法参数设置

算法	鸟巢数量	发现蛋概率值	函数维数
CS	30	0.7	30

表 11 PSO 算法和 CS 算法对于函数优化问题的结果比较

函数名称	PSO				CS			
	平均值	标准差	最优值	最差值	平均值	标准差	最优值	最差值
Sphere	1.567 8E-08	1.127 9E-08	7.576 2E-09	3.754 1E-08	1.995 0E-08	1.801 9E-08	5.558 8E-09	5.242 2E-08
Griewank	0.940 1	0.257 4	0.557 9	1.224 1	2.712 7E-03	1.942 1E-03	4.889 0E-04	5.484 8E-03
Rastrigin	9.915 3	0.402 6	9.497 7	10.481 6	63.232 8	3.335 8	59.413 4	69.071 3
Rosenbrock	1.975 1E+03	4.046 1E+02	1.345 7E+03	2.458 5E+03	5.540 5	0.892 2	4.034 1	6.729 1

3.2 实验数据分析

Sphere 函数是比较简单的单峰函数，比较容易搜索到最优值。由表 3 可知，在对 Sphere 函数的仿真中，OAQABC 算法的解的精度最高，ABC 算法次之，而 QABC 算法精度最差。在算法的稳定方面，明显 OAQABC 算法更为稳定。由图 1 可以

看出 QABC 算法和 OAQABC 算法的收敛速度明显优于 ABC 算法，而 OAQABC 算法的收敛速度也明显优于 QABC 算法。

Schwefel 是比较复杂的单峰函数。由表 4 可知，3 种算法均可搜索到最优值。但是从由图 2 可知，OAQABC 算法的收敛速度明显优于另外两种算法，可以更快的找到最优值。

Rosenbrock 函数是很难找到最优值的多峰函数。由表 5 可知, 相对于前两种测试函数, 3 种算法的收敛精度没有之前那么高。原因就是因为在 Rosenbrock 函数的最优值非常难以搜寻很容易陷入局部最优。从表中的数据可以看出 OAQABC 算法的收敛精度依然最好, 且最稳定。由图 3 可知, OAQABC 算法的收敛速度最快, 远远高于其他两种蜂群算法。

表 6~8 所反映的是关于多模态函数 Rastrigin、Ackley, Griewank 的优化问题仿真结果。以上函数均是复杂的非线性全局优化函数, 这对于测试算法的全局搜索性非常好。从表 6~8 可以看出 OAQABC 算法对与函数优化的精度均高于 ABC 算法和 QABC 算法, 特别是对于 Griewank 函数和 Ackley 函数的优化结果表明, OAQABC 算法的搜索精度较 QABC 算法要精确很多。同时, 图 4~6 也表现出了 OAQABC 算法明显高于其他两种蜂群算法的搜索速度。所以对于 QABC 算法所存在的后期容易陷入局部最优, 收敛速度减慢, 甚至不如 ABC 算法的问题, OAQABC 算法已经有了很好的改进, 不仅如此, OAQABC 算法的收敛速度在 QABC 算法的基础上又再一次得到提高。

表 11 所反映的是 PSO 算法和 CS 算法对于不同测试函数的仿真结果, 可以看出 PSO 算法和 CS 算法的搜索性能, 特别是搜索精度要明显低于 OAQABC 算法, 甚至低于 ABC 算法和 QABC 算法。对于 Rastrigin 函数和 Rosenbrock 函数, PSO 算法和 CS 算法均表现出了很差的优化性, 最终也并没有搜索到理论最优值。

4 结论

本文在 QABC 算法的基础上, 提出一种 OAQABC 算法。该算法具有以下特点: 1) 引入反向学习优化引领蜂搜索阶段, 使算法能保持更快的收敛速度; 2) 提出自适应步长 d , 使得 QABC 算法在收敛速度快的同时能更好的提高收敛精度; 3)

通过仿真实验可以看出 OAQABC 算法相对 ABC 算法和 QABC 算法有更好的收敛精度和更快的收敛速度; 4) 对于 QABC 算法所存在的后期容易陷入局部最优, 收敛速度减慢, 甚至不如 ABC 算法的问题, OAQABC 算法已经有了很好的改进, 不仅如此, OAQABC 算法的收敛速度在 QABC 算法的基础上又再一次得到提高; 5) 无论是对于简单的单峰函数、复杂的单峰函数、多峰函数还是多模态函数, OAQABC 算法的搜索性能均高于 ABC 算法和 QABC 算法; 6) 和其他群智能算法的函数优化结果的比较也能看出以 OAQABC 算法为代表, 3 种蜂群算法的收敛精度和优化性能要远远优于 PSO 算法和 CS 算法。

可以看出 OAQABC 算法具有较快的收敛速度和较高的收敛精度, 在不同的函数优化中都显现出优势, OAQABC 算法的优化性能已经高于 ABC 算法、QABC 算法、PSO 算法和 CS 算法。

参考文献:

- [1] Karaboga D, Basturk B. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm [J]. *Journal of Global Optimization* (S0925-5001), 2007, 39(3): 459-471.
- [2] Karaboga D, Basturk B. On the performance of artificial bee colony (ABC) algorithm [J]. *Applied Soft Computing* (S1568-4946), 2008, 8(1): 687-697.
- [3] Cui Zhe, Gu Xingsheng. An improved discrete artificial bee colony algorithm to minimize the makespan on hybrid flow shop problems [J]. *Neurocomputing* (S0925-2312), 2015, 148(2015): 248-259.
- [4] 周文越, 吕飞鹏, 廖小君. 基于人工蜂群算法的分布式电源规划 [J]. *电力系统及其自动化学报*, 2014, 26(5): 39-43. (Zhou Wenyue, Lu Feipeng, Liao Xiaojun. Optimal Planning of Distributed Generation Based on Artificial Bee Colony Algorithm [J]. *Proceedings of the CSU-EPSC*, 2014, 26(5): 39-43.)
- [5] 马文强, 唐秋华, 张超勇, 等. 基于离散人工蜂群算法的炼钢连铸调度优化方法 [J]. *计算机集成制造系统*, 2014, 20(3): 586-594. (Ma Wenqiang, Tang Qiuhua, Zhang Chaoyong. Steel-making continuous-casting scheduling method based on DABC algorithm [J]. *Computer Integrated Manufacturing Systems*, 2014, 20(3): 586-594.)

(下转第 2700 页)

<http://www.china-simulation.com>

• 2691 •