基于向量整体扰动的快速收敛人工蜂群算法*

向万里[1,2]，马寿峰[2]
（1. 兰州交通大学 交通运输学院，兰州 730070；2. 天津大学 系统工程研究所，天津 300072）

摘　要：针对标准人工蜂群算法收敛速度慢和易陷入早熟收敛等问题，提出一种快速收敛人工蜂群算法。首先借助反向学习理论初始化种群来提高初始解的分布质量，并在雇佣蜂和跟随蜂阶段引入向量整体扰动搜索方程加快局部搜索；然后为了跳出局部最优，采用一种随机更新搜索策略来增加蜂群多样性以平衡全局探索和局部利用能力；最后通过八个标准测试函数的仿真实验，发现所提出的算法与几个改进的人工蜂群算法相比，具有更快的收敛速度且获得了更高的求解精度，验证了算法的优越性。

关键词：人工蜂群算法；向量整体扰动；反向学习；随机更新搜索策略
中图分类号：TP18　　**文献标志码**：A　　**文章编号**：1001-3695(2013)05-1329-05
doi:10.3969/j.issn.1001-3695.2013.05.012

Fast convergent artificial bee colony algorithm based on full vector perturbation

XIANG Wan-li[1,2], MA Shou-feng[2]
(1. *School of Traffic & Transportation*, *Lanzhou Jiaotong University*, *Lanzhou* 730070, *China*; 2. *Institute of Systems Engineering*, *Tianjin University*, *Tianjin* 300072, *China*)

Abstract: To the defect of low searching speed and premature convergence frequently appeared in the standard artificial bee colony (ABC) algorithm, this paper proposed a fast convergent artificial bee colony algorithm based on full vector perturbation. Firstly, it employed opposition-based learning to initialize the population so as to improve the distributive quality of initial solutions. Secondly, it introduced search equation based on full vector perturbation to enhance local search during the first two phases of ABC. And then, in order to get out of the local minima and maintain the population diversity, it proposed random update search strategy for balancing between the ability of exploration and the one of exploitation. Lastly, experimental results tested on 8 benchmark functions show that the proposed algorithm reaches extremely faster convergence speed as well as better computational precision than other improved variants of ABC, which indicates that its performance is enhanced conspicuously.

Key words: artificial bee colony algorithm; full vector perturbation; opposition-based learning; random update search strategy

0　引言

人工蜂群(artificial bee colony, ABC)算法是由土耳其开塞利大学教授 Karaboga[1] 于 2005 年提出的一种群体智能优化算法。ABC 通过模拟自然界中蜂群采蜜行为来实现最优解的随机搜索。目前关于 ABC 算法的研究还处于不断探索改进的阶段，但得益于其算法结构简单、控制参数少、无须梯度信息、易于实现及全局寻优能力强等优点，已受到越来越多的学者关注，并已在函数优化[2~10]、系统辨识[11]、组合优化[12,13]、聚类分析[14]、生产调度[15]等领域得到了广泛的应用，尤其是在单模态、多模态、高维等复杂函数优化方面展现出了优于粒子群算法、遗传算法、差分进化算法等的卓越性能[7,9,10]。与其他群体智能优化算法一样，标准 ABC 也存在陷入局部最优而早熟收敛的可能。为此，众多学者进行了诸多改进，以克服其早熟收敛的特性。文献[3,4]通过引入混沌映射分别提出了带混沌局部搜索的混沌人工蜂群算法；Gao 等人[5]采用混沌映射和反向学习理论初始化 ABC 种群，然后引入差分变异搜索机制和一个平衡选择两种搜索机制(差分变异搜索机制和原 ABC 搜索机制)的概率以提高算法的全局进化性能；暴励等人[6]使用差分进化与蜂群算法相结合提出了一种新的双种群差分蜂群算法，并获得了极好的搜索性能；Akay 等人[8]基于 ABC 位置搜索更新方程引入一个扰动概率参数 MR(modify rate)和一个自适应尺度因子 ASF(adaptive scaling factor)，提出了一种改进的人工蜂群算法；文献[9,10]基于适应度最好的个体构建个体搜索方程，分别提出了 Gbest-ABC 和 best-so-far ABC。以上各种改进都不同程度地改进了算法的性能，取得了较好的优化效果。然而上面所提出的算法在对许多标准测试函数取得较高收敛速度和求解精度的同时，对诸如 Rosenbrock 这样复杂的非线性函数[16]所表现出的收敛效果却不太理想。

为进一步提高 ABC 对各类型函数的收敛速度和求解精度，本文提出一种收敛速度更快、求解精度更高的人工蜂群算法——基于向量整体扰动的人工蜂群算法(vector perturbation-based artificial bee colony, VABC)。首先，VABC 不像标准 ABC 中每次只对个体中的某一随机选中的分量进行扰动，而是采取所有分量都向最优个体学习以提高学习效果和收敛速度；不同

收稿日期：2012-09-05；**修回日期**：2012-10-29　**基金项目**：国家自然科学基金资助项目(70971094)；天津市科技支撑计划基金重点资助项目(08ZCKFSF01000)；高等学校博士学科点专项科研基金资助项目(20090032110033)；兰州交通大学青年科学基金资助项目(2012029)

作者简介：向万里(1978-)，男(土家族)，湖南永顺人，讲师，博士研究生，主要研究方向为进化算法及其应用(xiangwl@tju.edu.cn)；马寿峰(1965-)，男(回族)，天津人，教授，博导，博士，主要研究方向为交通系统工程、人工智能.

于 best-so-far ABC 只在跟随蜂阶段进行向量整体扰动，VABC 在雇佣蜂阶段也对解向量的所有分量进行扰动，从而进一步加快收敛速度；此外，为了平衡局部利用和全局探索的能力，在每次进化迭代的跟随蜂阶段，都选择个体适应度差的个体进行向量整体扰动，这样可使较好个体不被破坏，达到保持较好个体种群的多样性，不会像轮盘赌选择机制那样使得更多较好个体向最优个体进化而陷入局部最优；最后，为了进一步平衡向量整体扰动所带来的种群快速趋于最优个体的不利特性，提出了随迭代次数增加随机替换掉种群中一部分较差个体以维护种群多样性的随机更新策略。

1 标准人工蜂群算法

标准 ABC 算法是模拟蜜蜂采蜜行为，并且通过蜜蜂个体相互协作而达到群体采蜜行为涌现的一种仿生智能优化算法。在 ABC 中，蜂群分为雇佣蜂（employed bee）、跟随蜂（onlooker bee）和侦察蜂（scout bee）三类。其中，蜂群中一半为雇佣蜂，个体数目为 SN（对应于解的数目），另一半为跟随蜂（特殊情形下，蜂群中的一个个体会转变角色为侦察蜂）。此外，在采蜜的过程中，当雇佣蜂的食物源位置（对应一个解）经过一个预先给定的限定次数 limit 之后仍未得到更新，其中一个雇佣蜂将丢弃相应的食物源，变成侦察蜂，并以随机的形式寻找一个新的食物源。这样，ABC 算法可大致由以下几个步骤组成：a）种群初始化；b）雇佣蜂阶段；c）适应度与选择概率计算；d）跟随蜂阶段；e）侦察蜂阶段。在 ABC 随机搜索迭代的过程中，以 MCN 表示最大迭代次数，iteration 表示迭代变量，trial_i 记录第 i 个雇佣蜂对应的食物源位置连续未被更新的次数。

在阶段 a），通过式（1）随机产生雇佣蜂食物源位置 X_i（$X_i=(x_{i1},x_{i2},\cdots,x_{iD})$）。

$$x_{ij}=x_{ij}^{\min}+\mathrm{rand}(0,1)\times(x_{ij}^{\max}-x_{ij}^{\min}) \tag{1}$$

其中：$i=1,2,\cdots,SN$；$j=1,2,\cdots,D$；$x_{ij}^{\min}$ 表示第 i 个解的第 j 个分量所能赋予的最小取值；$x_{ij}^{\max}$ 表示第 i 个解的第 j 个分量所能赋予的最大取值；rand(0,1) 表示随机产生一个位于[0,1]之间的随机数。

在阶段 b），对每一个雇佣蜂目前正在开采的食物源位置以式（2）进行一次局部利用，以更新食物源位置。在采用式（2）获得新位置 V_i（$V_i=(v_{i1},v_{i2},\cdots,v_{iD})$）后，通过式（3）计算其适应度值，并与 X_i 的适应度值进行比较。如果 V_i 的适应度值优于 X_i 的适应度值，用新位置 V_i 替换 X_i，且将参数 trial_i 清 0；否则，不作替换，令 $\mathrm{trial}_i=\mathrm{trial}_i+1$。

$$v_{ij}=x_{ij}+\varphi_{ij}(x_{ij}-x_{kj}) \tag{2}$$

其中：x_{ij} 对应于第 i 个解的第 j 个分量；v_{ij} 表示第 i 个雇佣蜂或者跟随蜂在 x_{ij} 基础上探索或开采出的新食物源位置的第 j 个分量位置，即对原食物源位置分量 x_{ij} 进行扰动后的新值；$k\in[1..SN]\setminus\{i\}$，$j\in\{1,2,\cdots,D\}$ 分别是随机选择的雇佣蜂个体及对应的食物源位置的第 j 个分量；$\varphi_{ij}\in[-1,1]$ 是一个随机数，用于控制 x_{ij} 的邻域半径。

在阶段 c），ABC 以式（3）的形式计算每个食物源的适应度，以式（4）的形式计算选择概率 p_i。

$$\mathrm{fit}_i=\begin{cases}1/(1+f(X_i)) & \text{if } f(X_i)\geqslant 0\\ 1+|f(X_i)| & \text{otherwise}\end{cases} \tag{3}$$

其中：$f(X_i)$ 表示第 i 个雇佣蜂对应的解 X_i 的目标函数值，fit_i 表示相应的适应度值。

$$p_i=\frac{\mathrm{fit}_i}{\sum_{j=1}^{SN}\mathrm{fit}_j} \tag{4}$$

其中：SN 表示食物源个数或者解的个数。

在阶段 d），根据式（4）中的选择概率 p_i，对每一个跟随蜂随机选择一个雇佣蜂的食物源，并以式（2）进行食物源位置更新；然后与阶段 c）一样，以贪婪选择机制在旧个体与新个体之间进行锦标赛选择。

在阶段 e），选择 trial_i 阈值最大的雇佣蜂 i，如果该雇佣蜂 i 的阈值 $\mathrm{trial}_i>\mathrm{limit}$，该雇佣蜂丢弃原食物源，变成侦察蜂，以式（1）随机产生一个新食物源位置替换被丢弃的食物源位置。

2 基于向量整体扰动的人工蜂群算法

2.1 基于反向学习的种群初始化

标准 ABC 算法采取随机形式产生初始种群，这种形式不能保证初始个体较均匀地分布在解空间内，从而在一定程度上可降低算法的求解效率。为提高初始群体的质量，本文采用基于反向学习的形式产生初始群体[6]。即首先随机产生 $\lfloor SN/2\rfloor$ 个雇佣蜂对应的食物源位置 X_i（或解），同时以式（5）的形式再产生相应数目的食物源位置 $X_{i+\lfloor SN/2\rfloor}$。不失一般性，可取 SN 为偶数。

$$x_{i+\lfloor SN/2\rfloor,j}=x_{ij}^{\min}+(x_{ij}^{\max}-x_{ij}) \tag{5}$$

其中：$i=1,2,\cdots,\lfloor SN/2\rfloor$；$j=1,2,\cdots,D$。

2.2 基于向量整体扰动的搜索方程

标准 ABC 算法中交叉变异操作以式（2）的形式实现，只对个体的单个分量实施了扰动，对原个体的扰动较小，这样原个体从其他个体获得的知识较少，从而降低了 ABC 算法的收敛速度。为了提高 ABC 收敛速度，改进其局部利用的能力，受具有较快收敛性能的 Gbest-guided ABC[9] 和 best-so-far ABC[10] 的启发，本文采用式（6）搜索新的解。该搜索方程实现了对个体所有分量的扰动，使得原个体获得较强的学习进化能力，从而实现所有被选择的个体迅速向最优个体聚集以加快收敛速度。而且，为了获得更快的收敛速度，在阶段 b）和 d）都以式（6）进行向量整体扰动。

$$v_{id}=x_{ij}+(x_{ij}-x_{bj})\varphi_{ij} \tag{6}$$

其中：$d=1,2,\cdots,D$；$j\in\{1,2,\cdots,D\}$ 是一个随机数；b 表示当前种群中花蜜最好的食物源；φ_{ij} 是[-1,1]内的一个随机数，用于控制搜索更新领域半径的方向和大小。

2.3 新的选择策略

轮盘赌选择机制是选择好的个体进行进一步开采利用，即较好的个体迅速向最好的个体聚集。这易导致较好个体的群体多样性急剧下降，从而使算法陷入早熟收敛。基于轮盘赌的反向选择机制是选择较差的个体进行进一步开采利用，即使较差个体向较好群体聚集，这种选择机制使得较好群体的多样性不遭破坏，从而使得人工蜂群算法具有较强的进化能力。从系统最优的角度而言，后一种选择机制有利于保持种群的多样性。

2.4 随机更新搜索策略

由于基于向量整体扰动的搜索方程具有较强的局部开采利用能力，根据“没有免费午餐的理论[18]”可知，高效的收敛速度势必快速降低种群的多样性，使得算法陷入局部最优。为

此在 ABC 的基础上增加一个新的随机更新搜索策略[17],即随着 VABC 迭代搜索次数 iteration 的增加,当 iteration > $\lfloor r_1 \cdot MCN \rfloor$时,其中比例系数 $r_1 \in [0,1]$,按适应度值对雇佣蜂群体进行排序,对适应度值最低的$\lfloor r_2 \cdot SN \rfloor$(比例系数 $r_2 \in [0,1]$)个个体按式(1)进行随机替换更新,以使种群在后期能够继续保持一定的种群多样,起到平衡 VABC 全局探索与局部开采的能力,进而使得算法可以跳出局部最优的陷阱。

2.5　VABC 算法的具体步骤

a)基于反向学习的种群初始化。设置 iteration = 0;设置最大迭代次数 MCN、控制参数 limit、雇佣蜂个数 SN 和函数维数 D 的值;并对所有的 $\text{trial}_i (i = 1,2,\cdots,SN)$ 置 0。

b) iteration = iteration + 1。

c)雇佣蜂阶段。

(a) $i = 0$;

(b) $i = i + 1$,对每一个雇佣蜂对应的食物源位置 X_i,根据式(6)产生一个中间新位置 V_i;

(c) if $f(V_i) < f(X_i)$ then $X_i = V_i$ and $\text{trial}_i = 0$

　　else $\text{trial}_i = \text{trial}_i + 1$;

(d) if $i < SN$ then 转本阶段(b),否则转 d)。

d)计算雇佣蜂个体的适应度值以及选择概率。

e)跟随蜂阶段。

(a) $t = 0$; $i = 1$;

(b)对每一个跟随蜂,依据概率 p_i 选择一个雇佣蜂对应的食物源位置 X_i,即

if rand(0,1) < p_i then 按式(6)产生一个中间新位置 V and $t = t + 1$;否则转本阶段(d);

(c) if $f(V) < f(X_i)$ then $X_i = V$ and $\text{trial}_i = 0$;

　　else $\text{trial}_i = \text{trial}_i + 1$;

(d) $i = i + 1$, if $i == (SN + 1)$ then $i = 1$;

(e) if $t < SN$ then 转本阶段(b),otherwise 转 f)。

f)新的随机更新搜索策略。

if iteration ≥ $\lfloor r_1 \cdot MCN \rfloor$ then 对雇佣蜂群体按适应度值升序排序,选择适应度值低的前$\lfloor r_2 \cdot SN \rfloor$个个体按式(1)随机产生食物源位置并更新相应雇佣蜂食物源位置。

g)侦察蜂阶段:

(a) $\eta = \arg \max_{i \in [1..SN]} (\text{trial}_i)$;

(b) if $\text{trial}_\eta \geq \text{limit}$ then 利用式(1)随机产生一个食物源位置,并替换相应雇佣蜂丢弃的食物源位置 X_η,同时令 $\text{trial}_\eta = 0$。

h)记录到目前为止最好的个体适应度值及其相应信息。

i) if iteration < MCN then 转 b);否则,算法终止。

3　数值仿真实验与结果分析

3.1　标准测试函数

为了验证 VABC 的有效性,将本文提出的 VABC 算法与标准 ABC 进行了比较,同时又与文献[6,10]分别提出的双种群差分进化蜂群算法 BDABC 以及 best-so-far ABC 进行了性能对比分析。仿真实验中选择了广泛使用的八个标准测试函数[6,10]。表 1 列出了这些函数的表达式、搜索空间及理论极值,并根据函数性质分为具有单一极小点(单模态)和多个局部极小点(多模态)两大类。其中,Sphere、Quartic 函数为单模态函数,且 Quartic 函数为带噪声的单模态函数;其余函数均为复杂多模态函数(Rastrigin、Rosenbrock、Griewank、Ackley、Schaffer、Schwefel's Problem 2.26)。多模态函数 Rosenbrock 的全局最优点位于一个平滑、狭长的抛物线型山谷内,加之 Rosenbrock 函数仅为优化算法提供了少量信息,使得算法很难辨别搜索方向,找到全局最小点的机会较小。最后,借助 MATLAB 2009a 平台编程进行仿真实验,在一定的迭代次数下,通过对测试函数独立运行测试 30 次,记录考察 30 次的均值、方差、平均收敛代数等信息来评估算法的收敛性能。

表 1　标准测试函数

函数名称	函数表达式	搜索空间	理论极值
Sphere	$f_1(X) = \sum_{i=1}^{D} x_i^2$	$[-100,100]^D$	0
Griewank	$f_2(X) = \frac{1}{4000}\sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D}\cos(\frac{x_i}{\sqrt{i}}) + 1$	$[-600,600]^D$	0
Rastrigin	$f_3(X) = \sum_{i=1}^{D}[x_i^2 - 10\cos(2\pi x_i) + 10]$	$[-5.12,5.12]^D$	0
Rosenbrock	$f_4(X) = \sum_{i=1}^{D-1}[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-30,30]^D$	0
Ackley	$f_5(X) = -20\exp(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}) - \exp(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)) + 20 + e$	$[-32,32]^D$	0
Schaffer	$f_6(X) = 0.5 + \frac{\sin^2(\sqrt{x_1^2 + x_2^2}) - 0.5}{(1 + 0.001(x_1^2 + x_2^2))^2}$	$[-100,100]^2$	0
Quartic	$f_7(X) = \sum_{i=1}^{D} i x_i^4 + \text{random}[0,1)$	$[-1.28,1.28]^D$	0
Schwefel's Problem 2.26	$f_8(x) = -\sum_{i=1}^{D} x_i \sin(\sqrt{\lvert x_i \rvert})$	$[-500,500]^D$	$-418.9829 \times D$

3.2　仿真实验 1

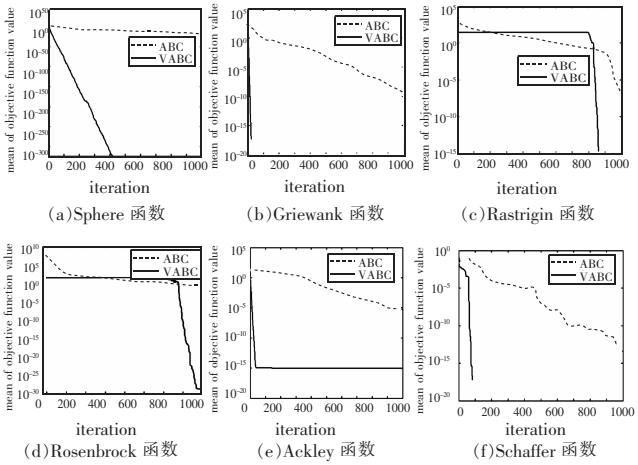
3.2.1　实验参数设置

为了与目前收敛性能效果非常好的 best-so-far ABC 进行性能对比分析,选取了与文献[10]同样的标准测试函数,即 f_1(Sphere)、f_2(Griewank)、f_3(Rastrigin)、f_4(Rosenbrock)、f_5(Ackley)以及 f_6(Schaffer)。考虑到比较的公平性,选择了与文献[10]一样的参数,即雇佣蜂个数 SN 为 100,除 Schaffer 函数维数 D 为 2 以外,其余函数维数 D 均为 30,测试运行最大代数 MCN 为1 000,控制参数 limit 设置为 100。此外,经过仿真实验,随机替换搜索策略阶段中迭代次数的比例系数 r_1 设置为 4/5,即当迭代次数超过总迭代次数的 4/5 时开始执行随机替换搜索策略;选择替换群体个数的比例系数 r_2 设置为 1/10,即雇佣蜂中适应度值最差的 1/10 个体被当做侦察蜂执行了随机更新搜索策略。在 MATLAB 2009a 平台上独立运行 30 次,采集 30 次运行的均值、方差、平均收敛代数,具体数据如表 2 所示。ABC 和 VABC 对各函数的进化过程对比曲线如图 1 所示(图中的横坐标表示进化代数,纵坐标表示平均目标函数值)。

3.2.2 仿真实验 1 结果分析

a)由表 2 和图 1(a)可知,对于单模态函数 Sphere 而言,改进的 VABC 算法收敛速度和求解精度远优于标准 ABC 算法,在求解精度方面略胜于 best-so-far ABC,而收敛速度快近一半多,且达到了理论极值点 0,方差也为 0,说明 VABC 算法对 Sphere 而言是健壮的,也说明基于向量整体扰动的搜索方程具备快速收敛的能力。

b)对十分复杂的 Rosenbrock 函数而言,很多优化算法对此函数的求解精度都较低[3,5],从图 1(d)可以看出,标准 ABC 对此函数的求解精度在 10^0 附近(远离理论极值 0),而 VABC 的求解精度亦远优于 ABC。此外,由于向量整体扰动搜索方程的快速收敛效果使得 VABC 长时间陷入 Rosenbrock 函数的平滑、狭长的抛物线型山谷中,从而使 VABC 进化收敛曲线有一长段近乎没有进化的水平线,这与前面分析的 Rosenbrock 函数特征相吻合,随后在达到总进化迭代次数 4/5 时,即随机更新搜索比例因子 $r_1=4/5$ 时执行随机替换搜索策略,以改善蜂群的多样性,使得 VABC 的全局探索能力和局部开发能力得到较好平衡,这时 VABC 对 Rosenbrock 函数的进化曲线在收敛性能方面迅速提升,求解精度达到了 10^{-29} 附近,表明引入的随机更新比例因子起到了进一步平衡探索与开采能力的作用。同时从表 2 可以看出,其收敛性能也优于 best-so-far ABC。另外,对其他多模态函数而言,由图 1(b)(c)(e)(f)可知,VABC 的进化收敛效果都远好于 ABC 算法;而且从表 2 也可以看出对多模态函数 Griewank、Rastrigin、Schaffer 而言,VABC 和 best-so-far ABC 都取得了理论最优值 0,而 VABC 收敛速度更甚一筹;对 Ackley 函数而言,VABC 在取得较高求解精度的同时收敛性能稍逊于 best-so-far ABC。



(a)Sphere 函数 (b)Griewank 函数 (c)Rastrigin 函数

(d)Rosenbrock 函数 (e)Ackley 函数 (f)Schaffer 函数

图 1 各函数进化进程曲线(Schaffer 函数维数 D=2,其余函数维数 D=30)

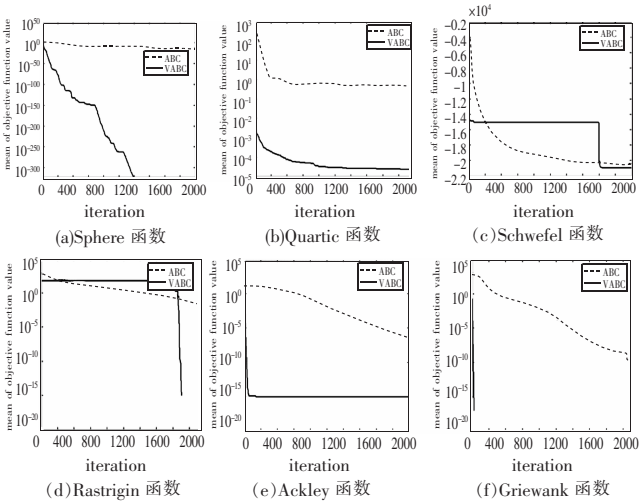
表 2 VABC 与 best-so-far ABC 的性能比较

函数	维数 D	算法	平均值	方差	平均收敛代数
Sphere	30	best-so-far ABC	5.6413e-302	3.967e-301	1 000
		VABC	0	0	421
Griewank	30	best-so-far ABC	0	0	845
		VABC	0	0	22
Rastrigin	30	best-so-far ABC	0	0	863
		VABC	0	0	862
Rosenbrock	30	best-so-far ABC	3.604e-15	3.585e-14	1 000
		VABC	1.925e-29	4.993e-29	979
Ackley	30	best-so-far ABC	0	0	109
		VABC	8.88178e-16	0	144
Schaffer	2	best-so-far ABC	0	0	947
		VABC	0	0	84

3.3 仿真实验 2

3.3.1 实验参数设置

为了进一步分析 VABC 算法对各类型函数及高维复杂函数的普适性,将 VABC 与文献[6]中的双种群差分蜂群算法 BDABC 进行了全面对比分析,选取与文献[6]一致的标准测试函数,即 f_1(Sphere)、f_2(Griewank)、f_3(Rastrigin)、f_5(Ackley)、f_7(Quartic)、f_8(Schwefel's Problem 2.26)共六个标准测试函数,同时选择了同文献[6]一样的参数,即雇佣蜂个数 SN 为 50,函数维数 D 分为 50 维和 100 维两种情形。对维数 D 为 50 维和 100 维的情形,测试运行最大代数 MCN 分别设置为2 000 和5 000,VABC 和 ABC 中控制参数 limit 设置为 100,比例系数 r_1、r_2 的设置与仿真实验 1 相同。在 MATLAB 2009a 平台上独立运行 30 次,采集 30 次运行的均值、方差、平均收敛代数,具体信息如表 3 所示,ABC 和 VABC 对各函数在 50 维情形下的进化过程曲线如图 2 所示(图中的横坐标表示进化代数,纵坐标表示平均目标函数值)。



(a)Sphere 函数 (b)Quartic 函数 (c)Schwefel 函数

(d)Rastrigin 函数 (e)Ackley 函数 (f)Griewank 函数

图 2 各函数进化进程曲线(各函数维数 D=50,SN=50)

3.3.2 仿真实验 2 结果分析

a)从图 2 和表 3 可以看出,对高维(50 维和 100 维)单模态函数和多模态函数而言,在相同参数设置下,本文提出的 VABC 算法明显优于文献[6]提出的双种群差分蜂群算法 BDABC 及标准 ABC 算法,且对 50 维和 100 维的 Sphere、Schwefel's Problem 2.26、Rastrigin 以及 Griewank 函数,VABC 均获得了理论极值,且方差为 0,说明改进的 VABC 较好地平衡了全局探索和局部开发,有效地提高了算法性能。对于带随机噪声的 Quartic 函数,极难获得理论极值 0 的情形下,VABC 也达到了 10^{-5} 附近的精度,好于 BDABC 所得的精度 10^{-1};同时从图 2(b)也可以看出,VABC 的求解精度和收敛速度都要好于标准 ABC;对于多模态函数 Ackley,在 50 维和 100 维的情形下,VABC 的求解精度都好于 BDABC。

3.4 仿真实验结果分析

通过以上分析可知,VABC 在各种类型和各种维度的函数情形下求解精度和收敛速度都比标准 ABC 显著提高,且随着维数的增长,VABC 在收敛速度和求解精度方面也保持了较好的稳健性;除了对 30 维情形的 Ackley 函数 VABC 逊色于 best-so-far ABC 以外,对其他几个广泛使用的函数优化问题而言,VABC 都优于 best-so-far ABC;同时,在与文献[6]提出的双种群差分蜂群算法 BDABC 对比分析中,VABC 收敛性能都好于

BDABC。因此,本文提出的 VABC 具备很快的收敛速度、很高的求解精度及很强的鲁棒性。

4　结束语

针对 ABC 算法长于全局探索而拙于局部利用的特点,为提高标准 ABC 收敛速度慢和求解精度不高的缺陷,提出一种基于向量整体扰动的搜索方程,且同时应用于 ABC 算法的雇佣蜂阶段和跟随蜂阶段,从而加快 ABC 的局部利用,起到提高收敛速度和求解精度的目的;与此同时,为了平衡算法的全局探索和局部利用能力,引入了新的随机更新搜索策略以增加种群的多样性,并经过仿真实验给出了随机更新策略中涉及的比例系数 r_1、r_2 的取值。这样,VABC 算法在提高算法求解精度和收敛速度的同时,保持了种群多样性,较好地平衡了算法的全局探索和局部利用能力。最后,经过八个标准测试函数的仿真测试,实验结果表明,本文提出的 VABC 算法在求解精度、收敛速度及鲁棒性方面都比标准的 ABC、best-so-far ABC 和双种群差分蜂群算法 BDABC 均有较大的改善,取得了令人满意的效果。此外,对于算法收敛性的理论分析证明将是一个非常有意义和挑战性的工作,值得进一步深入研究。

表 3　VABC 与 BDABC[6] 的性能比较

函数	维数 D	算法	平均值	方差	最差值	最优值
Sphere	50	BDABC	4.477788e－24	6.233437e－24	3.454583e－23	4.096191e－25
		VABC	0	0	0	0
	100	BDABC	1.255630e－33	1.300959e－33	5.119839e－33	1.880861e－34
		VABC	0	0	0	0
Quartic	50	BDABC	1.226310e－01	3.462654e－02	1.917625e－01	6.800217e－02
		VABC	2.232396e－05	1.556787e－05	7.613628e－05	1.412300e－06
	100	BDABC	2.500413e－01	5.214447e－02	4.017494e－01	1.822192e－01
		VABC	1.123319e－05	9.321279e－06	3.746699e－05	5.193098e－07
Schwefel	50	BDABC	－2.093728e＋04	3.613201e＋01	－2.083071e＋04	－2.094914e＋04
		VABC	－2.094914e＋04	0	－2.094914e＋04	－2.094914e＋04
	100	BDABC	－4.160215e＋04	1.512783e＋02	－4.130610＋04	－4.189829e＋04
		VABC	－4.189829e＋04	0	－4.189829e＋04	－4.189829e＋04
Rastrigin	50	BDABC	3.197442e－15	8.427937e－15	4.618528e－14	0
		VABC	0	0	0	0
	100	BDABC	1.369275e－16	2.837024e－16	1.443290e－15	0
		VABC	0	0	0	0
Ackley	50	BDABC	3.264174e－12	1.580295e－12	7.777778e－12	6.794565e－13
		VABC	8.881784e－16	0	8.881784e－16	8.881784e－16
	100	BDABC	3.408829e－13	9.451118e－14	5.870859e－13	2.353673e－13
		VABC	8.881784e－16	0	8.881784e－16	8.881784e－16
Griewank	50	BDABC	3.700743e－18	2.026981e－17	1.110223e－16	0
		VABC	0	0	0	0
	100	BDABC	7.216450e－17	1.261999e－16	4.440829e－16	0
		VABC	0	0	0	0

参考文献:

[1] KARABOGA D. An idea based on honey bee swarm for numerical optimization[R]. Kayseri:Erciyes University,2005.

[2] KARABOGA D,BASTURK B. Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems[C]//Proc of the 12th International Fuzzy Systems Association World Congress on Foundations of Fuzzy Logic and Soft Computing. 2007:789-798.

[3] 罗钧,李研. 具有混沌搜索策略的蜂群优化算法[J]. 控制与决策,2010,25(12):1913-1916.

[4] ALATAS B. Chaotic bee colony algorithms for global numerical optimization [J]. Expert System with Applications,2010,37(8):5682-5687.

[5] GAO Wei-feng,LIU San-yang. A modified artificial bee colony algorithm [J]. Computers & Operations Research,2012,39(3):687-697.

[6] 暴励,曾建潮. 一种双种群差分蜂群算法[J]. 控制理论与应用,2011,28(2):266-272.

[7] KARABOGA D,AKAY B. A comparative study of artificial bee colony algorithm[J]. Applied Mathematics and Computation,2009,241(1):108-132.

[8] AKAY B,KARABOGA D. A modified artificial bee colony algorithm for real-parameter optimization [J]. Information Sciences,2012,192(1):120-142.

[9] ZHU Guo-pu,KWONG S. Gbest-guided artificial bee colony algorithm for numerical function optimization[J]. Applied Mathematics and Computation,2010,217(7):3166-3173.

[10] BANHARSAKUN A, ACHALAKUL T, SIRRINAOVAKUL B. The best-so-far selection in artificial bee colony algorithm[J]. Applied Soft Computing,2011,11(2):2888-2901.

[11] 刘路,王太勇. 基于人工蜂群算法的支持向量机优化[J]. 天津大学学报,2011,44(9):803-809.

[12] 胡中华,赵敏. 基于人工蜂群算法的 TSP 仿真[J]. 北京理工大学学报,2009,29(11):978-982.

[13] 刘勇,马良. 最优证券投资组合的蜂群算法[J]. 上海理工大学学报,2011,33(1):44-46.

[14] KARABOGA D,OZTRUK C. A novel clustering approach: artificial bee colony (ABC) algorithm [J]. Applied Soft Computing,2011,11(1):652-657.

[15] 李修琳,鲁建厦,柴国钟,等. 混合蜂群算法求解柔性作业车间调度问题[J]. 计算机集成制造系统,2011,17(7):1495-1500.

[16] SHANG Yun-wei,QIU Yu-huang. A note on the extended Rosenbrock function[J]. Evolutionary Computation,2006,14(1):119-126.

[17] LIU Bo,WANG Ling,JIN Yi-hui,*et al*. Improved particle swarm optimization combined with chaos [J]. Chaos, Solitons & Fractals,2005,25(5):1261-1271.

[18] WOLPERT D H,MACREADY W G. No free lunch theorems for optimization[J]. IEEE Trans on Evolutionary Computation,1997,1(1):67-82.