

自扰动人工蜂群算法

周树亮 冯冬青 陈雪美

(郑州大学电气工程学院 郑州 450001)

摘要 人工蜂群(Artificial Bee Colony, ABC)算法是一种模仿蜂群寻找蜜源的新型算法,因具有参数简单、灵活性强等优点而被广泛用于解决工程问题。但该算法在早熟、收敛速度慢和个体越界等缺点。为此,提出一种自扰动人工蜂群算法(Novel Artificial Bee Algorithm with Adaptive Disturbance, IGABC)。该算法采用轴对称策略处理蜂群中的越界个体,提高了算法的搜索效率。通过改进全局搜索方程的结构,同时加入带阈值的线性递增策略,提出一种全新的自适应搜索方程。自适应搜索方程提高了算法的收敛精度并加快了速度。为了获得更好的全局最优解,提出一种自扰动方法对全局最优解进行扰动。选取18个基准测试函数以及近4年提出的6个改进ABC算法进行对比实验,结果表明,该算法在收敛速度和精度上均有较大的优势,尤其在处理Rosenbrock等很难寻优的复杂函数时,收敛精度提高了16个数量级。

关键词 改进算法, 自扰动, 带阈值的线性递增策略, 轴对称策略, 自适应, Rosenbrock

中图分类号 TP301.6 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2017.07.042

Novel ABC Algorithm with Adaptive Disturbance

ZHOU Shu-liang FENG Dong-qing CHEN Xue-mei

(School of Electrical Engineering, Zhengzhou University, Zhengzhou 450001, China)

Abstract As a new type of algorithm, artificial bee colony simulates the bee behaviors to find food. Since its simple parameters and flexibility, ABC is widely used to solve engineering problems. But the premature convergence and cross-border are disadvantages of ABC. To solve these problems, a novel ABC algorithm with adaptive disturbance(IGABC) was proposed in this paper. This improved algorithm adopted symmetry axis strategy to deal with the cross-border individuals, so the search efficiency is improved. A novel global self-adaptive search equation was proposed in this paper. The new search equation improves the structure of original global search equation, and adds linear increasing strategy with threshold. The search method for onlooker bees and employed bees improves the convergence precision and speed. IGABC algorithm designs a novel method on the base of global adaptive disturbance. The simulation results on 18 benchmark functions show that IGABC algorithm enhances the exploitation capacity, and the convergence speed and accuracy have made great progress, contrasting with other six improved ABC algorithms, which were proposed in the last two years. Especially when the test function is Rosenbrock, which is very difficult to find optimum solution, the convergence precision is increased by 16 orders of magnitude.

Keywords Improved algorithm, Adaptive disturbance, Linear increasing strategy with threshold, Symmetry axis, Self-adaptive, Rosenbrock

1 引言

人工蜂群算法(ABC)^[1]是继粒子群算法^[2]、遗传算法^[3]和蚁群算法^[4]后由土耳其学者Kataboga于2005年提出的一种新的群体智能算法。ABC算法因具有原理简单、控制参数少、灵活性强、适应度高、鲁棒性强等特点^[5],被广泛用于解决工程问题。

同时,ABC算法存在容易早熟、收敛速度慢等缺点,究其

原因在于搜索方式精于探索、疏于开发。

针对上述问题,研究者提出了很多改进方法。Zhu等^[6]借鉴PSO算法中的搜索方程,提出一种Gbest-guided ABC(GABC)算法,此算法在最优值的引导下,收敛速度明显加快,但是仍旧容易陷入局部最优。Jadhav等^[7]在PSO算法的启发下,提出一种改进的蜂群算法。Wang等^[8]受PSO算法和DE算法的共同启发,提出了两种不同的搜索方程,此IABC算法在收敛速度和精度上都有不错的表现,但是两种

到稿日期:2016-06-05 返修日期:2016-08-28 本文受国家自然科学基金(61473266),河南省重点科技攻关项目(152102210036)资助。

周树亮(1990-),男,硕士生,CCF会员,主要研究方向为嵌入式系统、智能算法,E-mail:845598986@qq.com;冯冬青(1958-),男,博士,教授,主要研究方向为智能控制理论与应用、图像处理、模式识别;陈雪美(1990-),女,硕士生,主要研究方向为智能算法和图像处理。

搜索方程依次搜索无疑加大了运算难度。Gao^[9]针对 ABC 算法的不足,提出通过产生一个候选解来提高算法的搜索能力,该方法简称为 CABC。Amira^[10]将量子计算和 ABC 算法结合,提出了量子人工蜂群(QABC)算法,提高了算法的多样性和计算能力。Guo^[11]将历史最优解加入 ABC 算法的搜索方程中,以提高算法的探索能力,但是种群多样性被限制,其依旧容易陷入局部最优。罗钧等人^[12]提出基于分段搜索策略的改进蜂群算法(SABC),即将搜索空间分段后再进行搜索。SABC 的搜索效率、收敛速度都有很大的改善,但是开发能力不足,收敛精度基本没有提高。Zhang 等^[13]在 DE 算法的启发下提出改进蜂群算法(NABC),改进后的算法提高了收敛精度。汪继文等人^[14]通过改进变异算子,提出了 ABC/current-to-best 算法,该算法提高了收敛速度,但是容易陷入局部最优。Li 等^[16]在原有算法中加入记忆机制,提出了 ABC-CM 算法。

由于有很大的随机性,因此在算法迭代过程中,很多个体会出现越界。经查阅文献发现,针对 ABC 算法中个体越界的问题,学者尚未提出合适的解决方案,往往都是将其放在搜索边界上,这样往往会丢失之前的历史信息,并且对算法收敛速度的影响很大。实验证明,算法运行过程中,5%~51%的个体会出现越界。本文提出一种轴对称策略来处理越界的个体,将种群中的每个个体牢牢锁定在搜索范围内,同时保留了一定的历史信息。

本文改进了雇佣蜂原有的搜索策略,设计了一个不同于文献^[6]的全局搜索方程。为了平衡算法不同时期的开发能力和探索能力,进一步改进雇佣蜂的搜索策略,在其中又加入带阈值的线性递增策略。为了获得更好的全局最优解,本文提出一个全新的全局最优解更新式(见式(7))。

仿真实验证明,IGABC 算法在收敛精度和速度上都有很大的改善,算法的开发能力也得到了很大的提高。通过对比其他 6 种改进的 ABC 算法,可以发现 IGABC 算法的开发能力非常出众,并且善于解决复杂函数的优化问题,例如 Ronsenbrock 和 Schaffer 函数。

2 标准蜂群算法

在 ABC 算法中,根据分工不同将蜜蜂分为雇佣蜂、跟随蜂、侦查蜂。种群大小为 SN,算法根据式(1)进行初始化。

$$x_{ij} = lb_j + \text{rand}(0,1)(ub_j - lb_j) \quad (1)$$

其中, $j=1, \dots, D; i=1, \dots, SN$, SN 为蜜源的个数; D 为个体向量维数; lb_j 为第 j 维下界; ub_j 为第 j 维上界。

雇佣蜂在当前依附的蜜源附近进行搜索以寻找更好的蜜源,如果发现比当前更优秀的蜜源,则更新当前蜜源。雇佣蜂根据式(2)邻域搜索新的蜜源。

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \quad (2)$$

其中, $j=1, 2, \dots, D; k=1, 2, \dots, D; k$ 随机生成且 $k \neq i$; ϕ_{ij} 是在 $[-1, 1]$ 范围内的随机数。

跟随蜂按照轮盘赌方式选择蜜源,进而转化为雇佣蜂,并在蜜源邻域搜索新的蜜源。轮盘赌保证了优秀的蜜源更容易

被选中。轮盘赌选择概率如式(3)所示。

$$P_i = \frac{\text{fit}(x_i)}{\sum_{i=1}^D \text{fit}(x_i)} \quad (3)$$

其中, $\text{fit}(x_i)$ 是蜜源 x_i 的适应度值; P_i 是蜜源 x_i 被选择依附的概率。

雇佣蜂和跟随蜂在找到新蜜源后,进行适应度比较,留下适应度高的蜜源。适应度的计算公式如式(4)所示。

$$\text{fit}(x_i) = \begin{cases} 1/(1+f(x_i)), & f(x_i) \geq 0 \\ 1+|f(x_i)|, & \text{else} \end{cases} \quad (4)$$

其中, $f(x_i)$ 是蜜源 x_i 对应的目标函数的函数值。

蜜源不更新次数 failTime 超过阈值 limit 时,雇佣蜂就会转化为侦查蜂进行全局搜索,并按照式(1)产生新解。

3 自扰动的人工蜂群算法

3.1 边界改进

算法都有很大的随机性,因此 ABC 算法中每一个蜂群个体都有可能飞出边界。在算法迭代过程中,种群个体的分布情况如图 1 所示,其中实线框是界定的搜索空间。

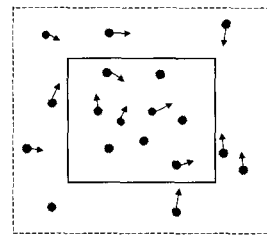


图 1 个体运动越界

为了得到 ABC 算法种群个体越界的具体数目,本文选取了 10 个测试函数(解位于搜索空间的中心)进行实验(测试函数的具体描述见表 2)。实验设置种群大小为 20,维数分别为 50 和 30,最大迭代次数为 2000,搜索范围为 $(-30, 30)$,运行次数为 15。将越界个体数目的平均值作为实验结果,如表 1 所列。

表 1 越界个体的数量

维数	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}
30	655	620	373	165	155	668	192	2059	646	669
50	218	237	553	204	225	289	335	2022	385	429

根据参数设置,不难得出种群个体更新位置的次数共为 4000(忽略侦查蜂的更新次数),越界次数最多的测试函数是 f_8 。超过 2000 次。不失一般性,在算法的运行过程中,大约有 5%~51%的个体出现越界。当前所有学者的做法都是暴力地将个体放置在搜索边界上,这样不仅破坏了蜂群个体原先带有的社会信息,还减缓了算法的收敛速度。

针对这个问题,本文提出一种轴对称的策略来处理越界的个体,即以边界线为对称轴,将越界的个体放置到搜索空间内与当前位置对称的位置。轴对称公式如式(5)所示。

$$v_{ij} = \begin{cases} 2ub_j - x_{ij}, & x_{ij} > ub_j \\ 2lb_j - x_{ij}, & x_{ij} < lb_j \end{cases} \quad (5)$$

选取 4 个测试函数进行实验,以直观观测采用轴对称边

界改进策略的效果。设定最大迭代次数为 2000,种群大小为 20,维数为 30,运行 10 次。10 次全局最优解的平均值如图 2 所示,其中 ABCMB 为边界改进后的算法。

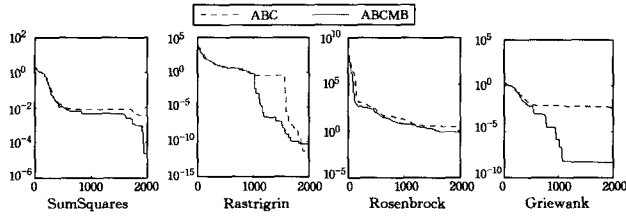


图 2 ABCMB 算法的收敛曲线

3.2 基于自扰动的全局自适应搜索方程

虽然改进后的 ABCMB 算法在收敛精度和速度上有所提高,但是核心的搜索方程并没有改变,算法的开发能力没有从根本上得到改善。针对这个问题,文献[6]借鉴 PSO 算法中的搜索方程,提出了最优引导搜索方程,如式(6)所示:

$$v_{ij} = x_{ij} + r_{ij}(x_{ij} - x_{ki}) + \phi_{ij}(x_{kj} - x_{ij}) \quad (6)$$

其中, x_{kj} 代表当前全局最优解; x_{ij} 代表当前解; x_{ki} 是当前不同于 x_{ij} 的一个随机解; r_{ij} 和 ϕ_{ij} 是 $(-1, 1)$ 之间的随机数。改进后的搜索方程在一定程度上提高了算法的开发能力,但是没有考虑到算法进化前期需要很快的收敛速度。因此,本文在该搜索方程的基础上,删繁就简,加入带阈值的线性递增策略,提出了自适应搜索方程,如式(7)所示。

$$v_{ij} = x_{kj} + r_{ij} * \left(\frac{iter}{maxiter} + c \right) (x_{kj} - x_{ij}) \quad (7)$$

其中, r_{ij} 是一个 $(-1, 1)$ 之间的随机数; $iter$ 是当前迭代次数; $maxiter$ 是最大迭代次数; c 是一个 $(0, 1)$ 范围内的常数,根据不同问题进行不同设置,若测试函数为 Rosenbrock,则 $c = 0.1$ 。

根据式(7)可以知道,种群中每个个体都在朝全局最优解移动。但是如果纯粹按照这样的思路来改进算法,由于算法太过依赖于全局最优解,因此很有可能陷入局部最优而无法跳出。如果全局最优解质量差,会导致整个种群陷入局部最优解。相反,质量好的全局最优解可以加快收敛速度,提高搜索效率。为了获得更好的全局最优解,本文提出一种全局自扰动的方法对全局最优解进行扰动,如式(8)所示。

$$X_g = X_g + (X_r - X_g) * rand * c \quad (8)$$

其中, X_g 是全局最优解(向量); c 是一个大于零的整数。通过种群内部自扰动的方法,增加了种群的多样性,全局最优解在种群内部不规则运动,从而解决了算法容易陷入局部最优的问题。

IGABC 算法采用轴对称策略来减少无效搜索。根据式(7)可知,在 IGABC 算法的迭代进化过程中,全局搜索方程过于依赖最优解,因此最优解的质量决定了算法的性能。IGABC 算法采用自扰动的方法来获得更好的全局最优解,即一个优秀的“领导者”可以带领团队获得更大的成功,从而加快了算法的收敛速度并提高了精度。

4 算法分析及步骤

针对种群个体越界的问题,本文利用轴对称策略对 ABC

算法进行边界改进,提高了算法的搜索效率。IGABC 算法改进了之前学者所采用的搜索方程(6),采用基于自扰动的全局自适应搜索方程,进一步加快了算法的收敛速度。IGABC 算法引进带阈值的线性递增策略,从而平衡了算法的探索能力和开发能力。从搜索方程中不难看出,IGABC 算法对全局最优解的依赖特别严重,只有保证有好的全局最优解,才能保证算法的高效。为了获得更好的“领导者”,本文提出一种全局自扰动的方法对“领导者”进行一次扰动。在这种扰动下,全局最优解在种群内部不规则运动,减小了陷入局部最优解的可能。具体步骤如下:

设置算法的各个参数;

初始化种群规模为 SN 的蜂群,计算每个雇佣蜂对应蜜源位置的适应度值,并记录全局最优解;

While 算法终止条件不满足

for 雇佣蜂根据式(7)更新蜜源位置,并计算新蜜源的适应度值;

采用轴对称策略处理越界个体;

采用贪婪机制选择较好的蜜源;

if 蜜源位置得到更新,则 failTime=0;

else failTime=failTime+1;

end if

end for;

根据更新后的蜜源的适应度值,按照式(3)计算选择概率 P_i ;

for 蜂群中每个跟随蜂

if $rand(0, 1) < P_i$

根据式(7)对蜜源进行更新;

采用轴对称策略处理越界个体;

采用贪婪机制选择较好的蜜源;

if 蜜源位置得到更新,则 failTime=0;

else failTime=failTime+1;

end if

end for

for 种群中每一雇佣蜂

采用根据式(9)对全局最优解进行更新;

轴对称策略处理越界个体;

采用贪婪机制选择较好的蜜源作为新的全局最优解;

end for

end while(满足迭代结束条件)

输出最优解及最优值。

5 实验仿真

为了验证 IGABC 算法的性能,将其与标准 ABC 算法以及近两年在国际重要期刊上发表的算法在 18 个基准测试函数上进行比较,对比算法包括 GABC1 算法^[6]、CABC 算法^[9]、NABC 算法^[13]、WGABC 算法^[16]、ABCSGQ 算法^[17] 和 GABC2 算法^[18]。为了充分检验算法的性能,选取的基准测试函数包括单峰函数、多峰函数、移位单峰函数和移位多峰函数。18 个基准测试函数(包括单峰函数、多峰函数、移位单峰函数和移位多峰函数)的数学表达式、搜索范围和理论最优值如表 2 所列。

表2 基准测试函数

函数	数学描述	搜索范围	极值
Sphere	$f_1(x) = \sum_{i=1}^D x_i^2$	(-100,100)	$f_1(\vec{0})=0$
SumSquare	$f_2(x) = \sum_{i=1}^D i x_i^2$	(-30,30)	$f_2(\vec{0})=0$
Rosenbrock	$f_3(x) = \sum_{i=1}^{D-1} (100(x_{i+1}^2 - x_i^2)^2 + (x_i - 1)^2)$	(-30,30)	$f_3(\vec{1})=0$
QuadricNoise	$f_4(x) = \sum_{i=1}^D i x_i^4 + \text{rand}(0,1)$	(-30,30)	$f_4(\vec{0})=0$
Ackley	$f_5(x) = -20 \exp(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}) - \exp(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)) + 20 + e$	(-30,30)	$f_5(\vec{0})=0$
Step	$f_6(x) = \sum_{i=1}^D (\lfloor x_i + 0.5 \rfloor)^2$	(-100,100)	$f_6(\vec{0})=0$
HyperEllipsoid	$f_7(x) = \sum_{i=1}^D 5i x_i^2$	(-30,30)	$f_7(\vec{x})=0$ $x(i)=5i$
Rastrigin	$f_8(x) = \sum_{i=1}^D \{x_i^2 - 10 \cos(2\pi x_i) + 10\}$	(-30,30)	$f_8(\vec{0})=0$
Zakharov	$f_9(x) = \sum_{i=1}^D x_i^2 + (\sum_{i=1}^D \frac{i x_i^2}{2})^2 + (\sum_{i=1}^D \frac{i x_i^2}{2})^4$	(-30,30)	$f_9(\vec{0})=0$
Alpine	$f_{10}(x) = \sum_{i=1}^D x_i * \sin(x_i) + 0.1 * x_i$	(-10,10)	$f_{10}(\vec{0})=0$
Schewel	$f_{11}(x) = \sum_{i=1}^D x_i + \prod_{i=1}^D x_i $	(-30,30)	$f_{11}(\vec{0})=0$
SumDifferent	$f_{12}(x) = \sum_{i=1}^D x_i ^{i+1}$	(-30,30)	$f_{12}(\vec{0})=0$
Schaffer	$f_{13}(x) = 0.5 + \frac{\sin^2 \sqrt{x_1^2 + x_2^2} - 0.5}{(1 + 0.001(x_1^2 + x_2^2))^2}$	(-30,30)	$f_{13}(\vec{0})=0$
GoldsteinPrice	$f_{14}(x) = (1 + (x_1 + x_2 + 1)^2 * (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)) * (30 + (2x_1 - 3x_2)^2 * (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2))$	(-30,30)	$f_{14}(0, -1)=3$
Griewank	$f_{15}(x) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos(\frac{x_i}{\sqrt{i}}) + 1$	(-600,600)	$f_{15}(\vec{0})=0$
Shifted Sphere	$f_{16}(x) = \sum_{i=1}^D x_i^2 + 390$	(-100,100)	$f_{16}(\vec{0})=390$
Shifted Rosenbrock	$f_{17}(x) = \sum_{i=1}^{D-1} (100(x_{i+1}^2 - x_i^2)^2 + (x_i - 1)^2) + 390$	(-30,30)	$f_{17}(\vec{1})=390$
Shifted Griewank	$f_{18}(x) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos(\frac{x_i}{\sqrt{i}}) + 400$	(-600,600)	$f_{18}(\vec{0})=390$

从表2可以看出,Sphere和Step函数是单峰函数,其定义域仅有一个极值点,一般用来测试算法的收敛速度;Ackley,Rastrigin和Griewank函数都是非线性多峰函数,其定义域存在多个局部极值点,一般用来测试函数全局寻优能力和收敛精度;Schaffer,Rosenbrock和GoldsteinPrice都是病态的复杂优化函数,很难找到最优解,通常用来测试算法的收敛精度和开发能力。Shifted Sphere,Shifted Rosenbrock和Shifted

Griewank都是移位函数。实验分为两组,通过Matlab编程实现。两组实验的种群规模设置为20,最大循环次数为2000,维数分别设置为30和50。将每个算法运行30次的平均值MEAN和方差STD作为实验结果。两组实验结果如表3和表4所列。GoldsteinPrice和Schaffer是2维的基准测试函数,实验结果如表5所列。部分基准测试函数的收敛曲线如图3、图4所示。

表3 8种算法的测试结果对比(D=30)

	f_1		f_2		f_3		f_4	
	MEAN	STD	MEAN	STD	MEAN	STD	MEAN	STD
ABC	1.05e-15	3.54e-16	1.02e-15	4.67e-16	2.48e+00	4.37e+00	1.97e-02	4.49e-02
GABC1	6.46e-16	1.26e-16	6.60e-16	1.36e-16	9.70e+00	1.38e+01	3.83e-02	1.73e-02
CABC	4.35e-03	1.80e-03	4.24e-03	8.20e-03	5.71e+02	1.02e+03	2.36e+01	8.74e+01
NABC	3.55e-07	1.77e-06	2.63e-04	1.16e-03	9.56e+01	1.71e+02	8.41e-02	4.97e-02
WGABC	5.15e-16	9.99e-17	5.19e-16	9.88e-17	3.20e+01	1.88e+01	4.14e-02	2.74e-02
ABCSGQ	3.84e-11	2.07e-10	4.75e-08	2.16e-07	1.34e+01	2.35e+01	3.17e-02	1.29e-02
GABC2	5.87e-16	1.26e-16	6.00e-16	1.01e-16	9.44e+00	1.47e+01	3.19e-02	1.54e-02
IGABC	2.74e-16	6.51e-17	7.83e-16	1.01e-15	7.78e-16	9.16e-16	1.15e-03	1.25e-03
	f_5		f_6		f_7		f_8	
	MEAN	STD	MEAN	STD	MEAN	STD	MEAN	STD
ABC	1.97e-12	2.80e-12	3.33e-02	1.83e-01	9.99e-16	2.66e-16	3.34e-02	1.82e-01
GABC1	4.57e-14	4.90e-15	0.00e+00	0.00e+00	7.15e-16	1.28e-16	9.26e-05	2.47e-04
CABC	2.22e+00	1.07e+00	0.00e+00	0.00e+00	3.60e-02	1.36e-01	9.95e+00	4.25e+00
NABC	9.71e-01	7.12e-01	0.00e+00	0.00e+00	4.02e-03	1.94e-02	4.80e+00	3.67e+00
WGABC	4.33e-14	5.90e-15	0.00e+00	0.00e+00	5.40e-16	9.74e-17	4.36e-14	3.56e-14
ABCSGQ	1.61e-04	5.93e-04	0.00e+00	0.00e+00	4.93e-07	1.75e-06	1.62e-09	8.07e-09
GABC2	4.55e-14	7.50e-15	0.00e+00	0.00e+00	6.13e-16	9.44e-17	1.68e-05	7.93e-05
IGABC	3.11e-14	3.75e-15	0.00e+00	0.00e+00	2.91e-16	7.09e-17	0.00e+00	0.00e+00

(续表)

	f_9		f_{10}		f_{11}		f_{12}	
	MEAN	STD	MEAN	STD	MEAN	STD	MEAN	STD
ABC	9.44e-16	2.31e-16	5.82e-05	8.17e-05	1.96e-12	2.28e-12	1.93e-05	2.49e-05
GABC1	6.50e-16	1.15e-16	1.71e-05	4.58e-05	1.64e-15	2.38e-16	3.80e-06	7.35e-06
CABC	3.00e-03	1.45e-02	8.73e-02	1.34e-02	5.54e-02	7.28e-02	2.86e-15	1.35e-14
NABC	5.54e-07	1.77e-06	2.09e-02	5.31e-02	6.11e-03	1.03e-02	8.80e-12	2.72e-11
WGABC	5.51e-16	1.24e-16	4.81e-05	8.57e-05	1.46e-15	2.11e-16	1.05e-16	8.49e-17
ABCSGQ	8.49e-08	3.15e-07	4.97e-05	1.00e-04	1.79e-03	2.19e-03	1.13e-05	3.43e-05
GABC2	6.24e-16	1.50e-16	1.87e-05	6.61e-05	1.63e-15	2.14e-16	8.84e-14	3.37e-13
IGABC	2.30e-16	4.36e-17	3.98e-07	1.40e-06	7.38e-16	1.12e-16	2.39e-17	1.14e-17
	f_{15}		f_{16}		f_{17}		f_{18}	
	MEAN	STD	MEAN	STD	MEAN	STD	MEAN	STD
ABC	2.56e-05	6.95e-05	3.90e+02	1.65e-13	3.93e+02	3.02e+00	3.90e+02	1.12e-02
GABC1	3.60e-08	1.87e-07	3.90e+02	1.17e-13	3.99e+02	1.41e+01	3.90e+02	5.88e-03
CABC	4.43e-02	4.04e-02	3.90e+02	6.72e-04	8.75e+02	8.24e+02	3.90e+02	1.72e-01
NABC	1.01e-02	2.65e-02	3.90e+02	3.67e-04	5.31e+02	4.14e+02	3.90e+02	1.77e-01
WGABC	1.13e-09	5.97e-09	3.90e+02	8.11e-14	4.25e+02	2.07e+01	3.90e+02	7.67e-02
ABCSGQ	2.46e-05	1.17e-04	3.90e+02	8.03e-10	3.95e+02	1.33e+01	3.90e+02	1.32e-02
GABC2	1.48e-10	4.35e-10	3.90e+02	1.19e-13	3.93e+02	4.35E+00	3.90e+02	5.54e-03
IGABC	0.00e+00	0.00e+00	3.90e+02	2.11e-14	3.90e+02	4.74e-10	3.90e+02	3.66e-14

表 4 8 种算法的测试结果对比(D=50)

	f_1		f_2		f_3		f_4	
	MEAN	STD	MEAN	STD	MEAN	STD	MEAN	STD
ABC	5.97e-12	1.57e-11	6.88e-09	2.24e-08	7.62e+00	7.24e+00	5.76e-01	1.13e-01
GABC1	1.56e-15	3.17e-16	1.39e-15	2.48e-16	1.10e+01	1.43e+01	1.59e-01	4.56e-01
CABC	2.91e-05	7.34e-05	2.13e+00	3.29e+00	5.49e+02	6.20e+02	3.08e+00	7.49e+00
NABC	3.96e-02	1.16e-01	1.09e+00	5.36e+00	1.11e+02	1.73e+02	1.88e-01	9.91e-02
WGABC	1.26e-15	1.75e-16	1.34e-15	2.26e-16	6.01e+01	2.50e+01	3.99e-03	1.83e-03
ABCSGQ	3.05e-06	8.62e-06	1.76e-06	3.78e-06	2.67e+01	3.36e+01	1.08e-01	3.07e-02
GABC2	1.48e-15	2.58e-16	1.55e-15	4.07e-16	2.22e+01	2.93e+01	1.61e-01	3.17e-02
IGABC	6.43e-16	1.09e-16	6.61e-16	1.40e-16	5.30e-11	2.69e-10	9.30e-03	1.51e-02
	f_5		f_6		f_7		f_8	
	MEAN	STD	MEAN	STD	MEAN	STD	MEAN	STD
ABC	5.44e-06	4.84e-06	1.10e+00	7.59e-01	2.64e-08	4.07e-08	1.63e+00	1.43e+00
GABC1	3.12e-10	1.96e-10	0.00e+00	0.00e+00	1.43e-15	2.53e-16	5.51e-01	9.10e-01
CABC	2.68e+00	1.27e+00	0.00e+00	0.00e+00	5.40e+00	1.08e+01	3.48e+01	1.18e+01
NABC	1.66e+00	4.46e-01	0.00e+00	0.00e+00	1.25e-03	3.14e-02	2.28e+01	1.14e+01
WGABC	1.10e-12	6.53e-13	0.00e+00	0.00e+00	1.32e-15	3.83e-16	9.49e-13	2.16e-12
ABCSGQ	2.90e-02	9.86e-02	0.00e+00	0.00e+00	1.29e-07	2.94e-07	6.53e+00	4.11e+00
GABC2	3.27e-10	2.03e-10	0.00e+00	0.00e+00	1.38e-15	1.97e-16	6.14e-01	7.99e-01
IGABC	6.73e-14	8.47e-15	0.00e+00	0.00e+00	8.26e-16	8.39e-16	0.00e+00	0.00e+00
	f_9		f_{10}		f_{11}		f_{12}	
	MEAN	STD	MEAN	STD	MEAN	STD	MEAN	STD
ABC	1.32e-08	1.64e-08	5.71e-02	3.86e-02	1.19e-12	6.09e-13	7.69e+00	2.04e+01
GABC1	1.43e-15	2.91e-16	1.05e-02	1.96e-02	1.63e-15	1.73e-16	2.69e-01	6.07e-01
CABC	2.52e+00	1.08e+01	1.26e+00	9.88e-01	1.56e-01	3.44e-01	3.47e-02	1.17e-01
NABC	8.52e-07	2.49e-06	2.42e-01	4.84e-01	1.43e-02	2.76e-02	4.98e-02	9.05e-02
WGABC	1.43e-15	3.91e-16	5.18e-04	2.20e-04	1.49e-15	2.10e-16	9.90e-11	3.51e-10
ABCSGQ	1.21e-09	2.73e-09	5.10e-02	1.77e-02	1.09e-03	1.42e-03	5.38e-02	8.86e-02
GABC2	1.44e-15	2.55e-16	2.45e-02	4.86e-02	1.63e-15	2.26e-16	5.51e-05	1.49e-04
IGABC	2.72e-16	6.99e-17	3.55e-05	6.60e-05	7.47e-16	1.19e-16	2.67e-17	1.30e-17
	f_{15}		f_{16}		f_{17}		f_{18}	
	MEAN	STD	MEAN	STD	MEAN	STD	MEAN	STD
ABC	3.4e-08	1.73e-07	3.90e+02	9.14e-13	4.03e+02	1.82e+01	3.90e+02	7.71e-03
GABC1	1.36e-11	4.57e-11	3.90e+02	2.08e-13	4.24e+02	3.91e+01	3.90e+02	6.24e-03
CABC	6.62e-02	8.51e-02	3.91E+02	2.62e+00	9.91e+04	5.37e+05	3.91e+02	4.63e-01
NABC	1.19e-02	5.24e-02	3.90e+02	1.48e-03	4.50e+02	9.89e+01	3.90e+02	2.80e-01
WGABC	1.13e-09	5.97e-09	3.90e+02	2.2e-13	4.57e+02	2.84e+01	3.90e+02	8.82e-03
ABCSGQ	7.27e-06	3.46e-05	3.90e+02	2.43e-04	4.21e+02	3.04e+01	3.90e+02	3.63e-02
GABC2	1.76e-09	9.01e-09	3.90e+02	2.11e-13	4.23e+02	3.96e+01	3.90e+02	3.04e-03
IGABC	0.00e+00	0.00e+00	3.90e+02	1.18e-13	3.90e+02	8.40e-06	3.90e+02	1.73e-13

表 5 f_{13} 和 f_{14} 的测试结果

		ABC	GABC1	CABC	NABC	WGABC	ABCSGQ	GABC2	IGABC
f_{13}	MEAN	1.45e-01	1.45e-01	2.75e-01	1.41e-01	1.02e-02	3.88e-02	1.33e-01	1.64e-03
	STD	5.65e-02	9.96e-02	1.15e-01	1.07e-01	2.42e-03	3.15e-02	9.22e-03	3.57e-03
f_{14}	MEAN	3.42e+01	6.85e+00	3.00e+00	3.07e+01	3.00e+00	3.13e+00	3.20e+00	3.00e+00
	STD	3.69e+01	7.15e+00	9.20e-04	3.43e+01	5.64e-04	3.40e-01	4.90e-01	2.56e-12

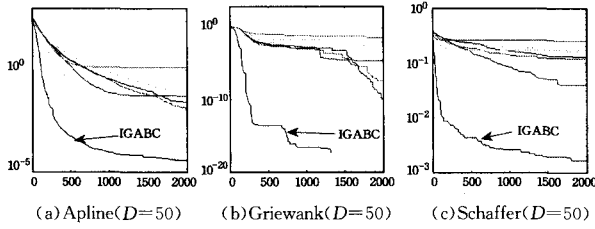


图 3 不同基准测试函数的收敛曲线

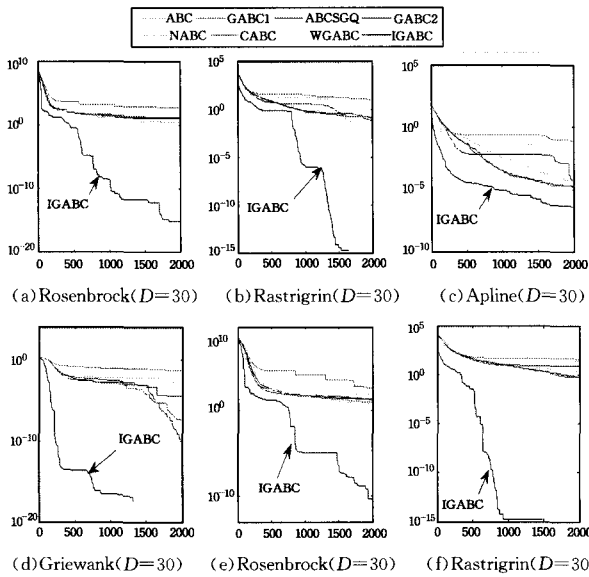


图 4 不同基准测试函数的收敛曲线

从表 2 和表 3 可以看出,相对于标准 ABC 算法,IGABC 算法对 18 种基准测试函数的优化效果都有不同程度的提高;相对于其他 6 种改进的 ABC 算法,IGABC 算法对于大多数基准测试函数的优化效果又有进一步提高,尤其对于 Griewank, Step 和 Rastrigrin 基准测试函数,无论是 30 维还是 50 维,IGABC 算法都可以在较少的迭代次数下收敛到理论最优值。本算法虽然对于其他测试函数没有收敛到理论最优值,但是收敛速度和精度都有很大的提高,尤其是对于寻优难度大的基准测试函数 Rosenbrock, GoldsteinPrice 和 Schaffer, IGABC 算法的收敛精度要明显高于其他 7 种算法。IGABC 算法采用了自适应搜索方程,并且加入了全局最优解的引导,大大加快了算法的收敛速度;利用全局个体对全局最优解进行扰动,增大了种群的多样性,算法的开发能力也得到加强。在 18 个被测函数中, Ronsenbrock 函数是一个单峰病态螺旋型非二次函数,在其定义域内的最优解 $(x_1, x_2, \dots, x_D) = (1, 1, \dots, 1)_D$ 旁边有一条狭长深谷,搜索其边缘时往往会发生振荡,很有可能陷入局部最优解。Ronsenbrock 虽然是一个单峰函数,但是常常被归为多模态复杂函数,因此 Ronsenbrock 函数常常用于评价算法的开发能力、收敛精度和搜索性能。从图 4 中 IGABC 算法对 Rosenbrock 函数的收敛曲线可以看

出,相对于其他 6 种改进算法,IGABC 算法的开发能力是非常显著的。

为了进一步验证 IGABC 算法解决多模复杂优化问题的寻优效果,选取了 6 种 2016 年提出的改进 ABC 算法进行对比。数据都是来自相应的文献,对比结果如表 6 所列。

表 6 Rosenbrock 函数的收敛结果

算法	维数	搜索范围	均值	方差
IGABC	30	$[-30, 30]$	7.78e-16	9.16e-16
IGABC	50	$[-30, 30]$	5.30e-11	2.69e-10
ABC ^[19]	30	$[-5.12, 5.12]$	6.74e-02	1.39e-01
GBABC ^[20]	30	$[-30, 30]$	3.63e+00	3.40e+00
GBABC ^[20]	50	$[-30, 30]$	4.92e+01	1.86e+01
EcABC ^[21]	30	$[-30, 30]$	3.42e-01	-
AC-ABC ^[22]	30	$[-30, 30]$	2.28e+01	8.91e+00
NABC ^[23]	30	$[-30, 30]$	2.26e-02	3.61e-02
LFABC ^[24]	10	$[-30, 30]$	2.53e-01	7.64e-01

Ronsenbrock 是一个病态的螺旋型函数,很难进行全局极小化,因此常被用来评价算法的寻优能力和搜索性能。从表 6 可知,无论是 30 维还是 50 维,IGABC 算法的收敛精度都远远超过其他 6 种改进的 ABC 算法,因此 IGABC 算法解决复杂函数 Ronsenbrock 的效果是非常显著的。

Sphere 是一个单峰函数,易于收敛。Sphere 一般用来测试算法的收敛速度。从图 5 可知,IGABC 算法可以在较少的迭代次数下找到最优解。与其他算法相比,IGABC 算法的收敛速度有了很大的改善。由于测试函数 Rosenbrock, Rastrigrin 和 Griewank 的自身理论最优解很难找到,因此一般情况下这些函数都用于评价算法的寻优能力。根据图 3、图 4 可以发现,IGABC 算法的寻优能力得到了很大的提高。表 7 记录了 5 种算法收敛到同一精度所需要的迭代次数。由表 7 可知,IGABC 算法对所有的测试函数都可以在较少的迭代次数下收敛到指定的精度,从而进一步反映了 IGABC 算法收敛速度快的特性。

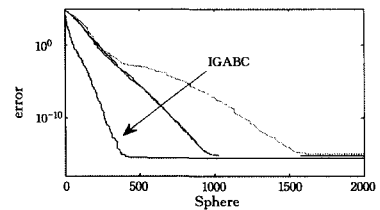


图 5 Sphere 函数的收敛曲线

表 7 算法的迭代次数

收敛精度	测试函数	f_6	f_1	f_2	f_3	f_4	f_5
		0	10^{-10}	10^{-10}	10^1	10^2	10^{-12}
ABC		1256	1145	132	1026	1856	-
GABC1		711	693	128	1746	1312	310
WGABC		704	724	79	1979	1126	111
GABC2		697	675	148	1902	1335	315
IGABC		266	299	21	345	452	74

结束语 对于种群中个体越界问题,本文采用轴对称策略将种群中每个个体都约束在搜索范围内,减少了无效搜索,提高了搜索效率。标准 ABC 算法由于具有精于探索、疏于开发的特性,因此存在早熟收敛和收敛速度慢等缺点。为此,本文提出基于自扰动的全局自适应搜索方程,并在搜索方程中加入全局最优解的引导,这样不仅加快了算法的收敛速度,而且使得算法的开发能力也得到了提高。为了平衡算法在不同时期的开发能力和探索能力,在搜索方程中加入了带阈值的线性递增策略。通过对 18 个基准测试函数寻优及与其他 6 种改进 ABC 算法进行对比,表明本文提出的改进算法 IGA-BC 在收敛速度和精度上都有很大的提高。相对于其他 6 种改进的 ABC 算法,IGABC 算法在处理 Rosenbrock 和 Schaffer 等这些很难寻优的复杂函数时,收敛精度提高了 16 个数量级。

参 考 文 献

- [1] KARABOGA D. An idea based on honey bee swarm for numerical optimization[R]. Turkey;Erciyes University,2005.
- [2] KENNEDY J,EBERHART R. Particle swarm optimization [C]// IEEE Int Conf on Neural Networks, Perth,1995:1942-1949.
- [3] TANG K S,MAN K F,KWONG S, et al. Genetic algorithms and their application [J]. IEEE Signal Processing Magazine, 1996,13(6):22-37.
- [4] DORIGO M,STUTZLE T. Ant colony optimization [M]. Cambridge,MA MIT Press,2004.
- [5] KARABOGA D,BASTURK B. On the performance of artificial bee colony algorithm[J]. Applied Soft Computing, 2008,8(1): 687-697.
- [6] ZHU G P,KWONG S. Gbest-guided artificial bee colony algorithm for numerical function optimization [J]. Applied Mathematics and Computation,2010,217(7):3166-3173.
- [7] JADHAV H T,ROY R. Gbest guided artificial bee colony algorithm for environmental/economic dispatch considering wind power[J]. Expert Systems with Applications, 2013,40(16): 6385-6399.
- [8] WANG Z,KONG X Y. An Improved Artificial Bee Colony Algorithm for Global Optimization [J]. Information Technology Journal,2013,12(24):8362-8369.
- [9] 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 Transactions on Cybernetics, 2013,43(3): 1011-1024.
- [10] AMIRA B,AMER D,SALIM C. A quantum-inspired artificial bee colony algorithm for numerical optimization[C]// Proceedings of International Symposium on Programming and Systems. Algiers,2013:81-88.
- [11] GUO P,CHENG W,LIANG J. Global artificial bee colony search algorithm for numerical function optimization[C]// Proceedings of 2011 Seventh International Conference on Natural Computation. Shanghai,China,2011:1280-1283.
- [12] LUO J,XIAO X H,FU L, et al. Modified artificial bee colony algorithm based on segmental-search strategy[J]. Control and Decision,2012,27(9):1402-1410. (in Chinese)
罗钧,肖向海,付丽,等. 基于分段搜索策略的改进蜂群算法[J]. 控制与决策,2012,27(9):1402-1410.
- [13] ZHANG S,LIU S Y. A Novel Artificial Bee Colony Algorithm for Function Optimization [J]. Mathematical Problems in Engineering,2015,2015:1-10.
- [14] WANG J W,YANG D,QIU J F, et al. Improved artificial bee colony algorithm for solving nonlinear equations[J]. Journal of Anhui University(Natural Science Edition),2014,38(3):16-23. (in Chinese)
汪继文,杨丹,邱剑锋,等. 改进人工蜂群算法求解非线性方程组 [J]. 安徽大学学报(自然科学版),2014,38(3):16-23.
- [15] LI X N,YANG G F. Artificial bee colony algorithm with memory[J]. Applied Soft Computing, 2016,41(1):362-372.
- [16] ZHANG Y Y,ZENG P,WANG Y, et al. Linear Weighted Gbest-guided Artificial Bee Colony Algorithm [C]//2012 5th International Symposium on Computational Intelligence and Design (ISCID). Hangzhou,China,2012:155-159.
- [17] ZHAO H,LI M D,WENG X W. Improved artificial bee colony algorithm with self-adaptive global best-guided quick searching strategy[J]. Control and Decision,2014,29(11):2041-2047. (in Chinese)
赵辉,李牧东,翁兴伟. 具有自适应全局最优引导快速搜索策略的人工蜂群算法[J]. 控制与决策,2014,29(11):2041-2047.
- [18] ROY R,JADHAV H T. Optimal power flow solution of power system incorporating stochastic wind power using Gbest guided artificial bee colony algorithm[J]. International Journal of Electrical Power and Energy Systems,2015,64(1):562-578.
- [19] ZHOU X Y,WU Z J,WANG H, et al. Gaussian bare-bones artificial bee colony algorithm[J]. Soft Computing, 2016,20(3): 907-924.
- [20] SHARMA K,GUPTA P C,SHARMA H. Fully informed artificial bee colony algorithm[J]. Journal of Experimental & Theoretical Artificial Intelligence,2016,28(1/2):403-416.
- [21] SHIMPI J,BANSA S,CHAND J, et al. Escalated convergent artificial bee colony [J]. Journal of Experimental & Theoretical Artificial Intelligence,2016,28(1/2):181-200.
- [22] LV L,HAN L Z,FAN T H, et al. Artificial bee colony algorithm with accelerating convergence[J]. International Journal of Wireless and Mobile Computing,2016,10(1):76-82.
- [23] WANG K. A new artificial bee colony by improving the search of onlooker bees[J]. International Journal of Wireless and Mobile Computing,2016,10(1):62-67.
- [24] SHARMA H,BANSAL B J C,ARYA K V, et al. Lévy flight artificial bee colony algorithm[J]. International Journal of Systems Science,2016,47(11):2652-2670.