

基于改进邻域搜索策略的人工蜂群算法

魏锋涛[†], 岳明娟, 郑建明

(西安理工大学 机械与精密仪器工程学院, 西安 710048)

摘要: 针对人工蜂群算法存在易陷入局部最优、收敛速度慢的缺陷, 提出一种改进邻域搜索策略的人工蜂群算法. 首先, 将混沌思想和反向学习方法引入初始种群, 设计混沌反向解初始化策略, 以增大种群多样性, 增强跳出局部最优的能力; 然后, 在跟随蜂阶段根据更新前个体最优位置引入量子行为模拟人工蜂群获取最优解, 通过交叉率设计更新前个体最优位置, 并利用势阱模型的控制参数提高平衡探索与开发的能力, 对观察蜂邻域搜索策略进行改进, 以提高算法的收敛速度和精度; 最后, 将改进人工蜂群算法与粒子群算法、蚁群算法以及其他改进人工蜂群算法进行比较, 利用12个标准测试函数进行仿真分析. 结果表明, 改进算法不仅提高了收敛速度和精度, 而且在高维函数优化方面具有一定的优势.

关键词: 人工蜂群算法; 混沌反向解初始化策略; 邻域搜索改进策略; 改进算法; 函数优化; 仿真分析

中图分类号: TP301

文献标志码: A

Artificial bee colony algorithm based on improved neighborhood search strategy

WEI Feng-tao[†], YUE Ming-juan, ZHENG Jian-ming

(College of Mechanical and Precision Instrument Engineering, Xi'an University of Technology, Xi'an 710048, China)

Abstract: As to overcome the drawback of easily falling into local optimum and slow convergence rate of the conventional artificial bee colony algorithm, this paper proposes an artificial bee colony algorithm based on the improved neighborhood search strategy. Firstly, in order to enhance the diversity of population and prevent local optimum, a kind of chaotic anti-base initialization mechanism is designed according to the chaotic thoughts and opposed-based learning method. Then, in the following stage of following bee stage, the quantum behavior is introduced to simulate the optimal solution of the artificial bee according to the optimal position of the former individual, the optimal position of the former individual is designed with crossover, and the control parameters of the well model is used to improve the balance exploration and development capability, a strategy of neighborhood search improvement strategy in observation is designed, to improve the convergence accuracy of the algorithm. Finally, the proposed algorithm is compared with the particle swarm optimization algorithm, ant colony algorithms, and other improved artificial colony algorithm, and simulation analysis is made on 12 standard test functions, the results show that the proposed algorithm not only improves the convergence speed and accuracy, but also has certain advantages in terms of high-dimensional function optimization.

Keywords: artificial bee colony algorithm; chaotic anti-base initialization strategy; neighborhood search improvement strategy; improved algorithm; function optimization; simulation analysis

0 引言

人工蜂群 (Artificial bee colony, ABC) 算法是 Karaboba 在 2005 年模仿蜜蜂群体觅食行为的一种智能算法, 具有互学习、结构简单、易于实现以及控制参数小等优点^[1-2], 已广泛应用于许多工程领域^[3-7]. 针

对人工蜂群算法存在收敛速度慢、易陷入局部最优等不足, 许多学者对其进行了改进研究. 文献 [8] 通过设计新的选择和交叉策略, 同时将反向学习引入变异策略中, 并应用于侦查蜂阶段, 实验结果加快了算法寻找最优解的速度. 文献 [9-10] 将差分进化算法

收稿日期: 2017-11-08; 修回日期: 2018-01-18.

基金项目: 国家自然科学基金项目 (51575443, 51475365); 陕西省自然科学基金基础研究计划项目 (2017JM5088); 陕西省教育厅科学研究计划项目 (15JK1521); 西安理工大学博士启动基金项目 (102-451115002).

责任编委: 林崇.

作者简介: 魏锋涛 (1976—), 男, 副教授, 博士, 从事现代优化设计理论与方法、复杂产品多目标多学科设计优化等研究; 郑建明 (1968—), 男, 教授, 博士, 从事优化设计理论及工程应用等研究.

[†]通讯作者. E-mail: weifengtao@xaut.edu.cn.

融入人工蜂群算法中,以平衡算法的探索 and 开发能力. Kiran 等^[11]集成多种搜索策略,对不同的优化问题采用不同的搜索策略.文献[12]在人工蜂群的基础上引入时间延迟参数和排序参数,利用分阶混沌系统解决参数估计的多维化问题,利用时滞分阶混沌系统的两个典型执行系统验证了算法的可行性.文献[13-14]将人工蜂群与Kmeans数据聚类方法相结合,即将人工蜂群的初始种群应用到Kmeans算法的初始阶段,结果表明了改进算法对数据聚类应用的有效性.王志刚等^[15]提出多搜索策略协同进化的人工蜂群算法,动态调整搜索的维数以提高搜索效率,并结合人工蜂群算法不同搜索策略的特点,使其协同进化,实验结果具有较好的收敛速度和精度.Li等^[16]将生物地理优化算法BBO的开发能力与人工蜂群算法的探索能力有效结合,产生较优的候选解,加快了算法的收敛速度.李国亮等^[17]设计了分阶段雇佣蜂搜索策略,降低了算法陷入局部极值的概率,并定义逃逸半径,更好地获得早熟个体跳出局部极值,适合高维复杂函数求解.

以上改进算法均从不同角度增强了ABC算法的寻优能力,性能得到很大提高,并得到广泛的应用,但如何更好地平衡全局探索和局部开发能力仍需进一步研究.鉴于此,本文提出一种基于改进邻域搜索策略的人工蜂群算法(Artificial bee colony algorithm based on improved neighborhood search strategy, ISABC),该算法为了扩大种群多样性,在初始化阶段,采用了混沌反向解初始化策略.同时,在跟随蜂阶段引入邻域搜索改进策略,采用量子行为对更新前个体最优解进行局部搜索,以保证算法的收敛精度和速度.通过对标准测试函数进行实验仿真,并与ABC算法、粒子群算法、蚁群算法及其他改进人工蜂群算法在不同维数下进行比较,以验证ISABC算法的有效性.

1 人工蜂群算法

人工蜂群算法是源于蜜蜂群体觅食行为的智能算法,蜜蜂通过引领蜂、跟随蜂和侦查蜂进行交流觅食.引领蜂和跟随蜂主要负责食物源(蜜源)的开采过程,这两种蜂的数量相同,每个雇佣蜂对应一个蜜源,侦查蜂负责探索新的食物源.

对于求解函数优化问题,蜜源的位置代表函数的可行解,蜜蜂所带的花蜜量代表了所求函数的适应度值,函数求解最优值的过程即是蜜蜂找寻花蜜量最大的过程.求解目标函数 $f(x)$ 时,随机初始化SN(引领蜂和观察蜂的数量)个解 $x_i (i = 1, 2, \dots, SN)$,每个

解 x_i 是一个 D 维向量,该算法的工作过程如下.

Step 1: 在当前解空间随机初始化种群,并计算函数的适应度值.

Step 2: 引领蜂对初始蜜源采用下式搜索附近新的食物源进行评价,并通过贪婪选择进行食物源更新:

$$v_{ij} = x_{ij} + \varphi_{ij}(x_{ij} - x_{kj}). \quad (1)$$

其中: φ_{ij} 为 $[-1, 1]$ 之间的随机数; $i, k \in 1, 2, \dots, SN$ 且 $i \neq k, j \in 1, 2, \dots, D$.

Step 3: 跟随蜂根据式(2)的轮盘赌概率选择某一食物源,并通过式(1)进行邻域搜索更新,有

$$p_i = \frac{\text{fit}_i}{\sum_{i=1}^{\text{SN}} \text{fit}_i}. \quad (2)$$

Step 4: 侦查蜂阶段,放弃长期未更新的食物源,并由跟随蜂转变为侦查蜂,通过式(3)在附近进行随机搜索新的食物源,有

$$x_i = \text{lb} + \text{rand}(1, d) \times (\text{ub} - \text{lb}). \quad (3)$$

其中:ub和lb分别为解的上、下界;rand(1, d)表示产生一个1行 d 列的矩阵, $d \in (1, 2, \dots, D)$.

Step 5: 终止条件判断,若满足,则输出最优解,否则返回Step 2.

2 基于改进邻域搜索策略的人工蜂群算法

人工蜂群算法在解决优化问题时,采用随机初始化方案,其随机性使得蜜蜂不能充分搜索蜜源(解)的空间,易陷入局部最优.同时,跟随蜂采用随机邻域搜索策略,其随机性和盲目性决定了探索和开发能力的不平衡,降低了算法的收敛速度和精度.针对以上不足,将混沌反向解初始化策略和邻域搜索改进策略引入ABC算法中,以提高算法的收敛速度和精度.

2.1 混沌反向解初始化策略

为了扩大种群的多样性,增强算法跳出局部最优的能力,将混沌思想和反向学习方法引入种群初始化过程,提出一种混沌反向解初始化策略.该策略通过概率思想将混乱下的正向解和反向解进行组合,将可能的较优解提取出来,扩大解的空间.

1) 混沌思想.

混乱的本质是随机的、不可预测的,也有规律性的因素,一个混乱的空间就是一个离散时间的动态系统,其数学描述为

$$x_{k+1} = f(x_k), 0 < x_k < 1, k = 0, 1, \dots \quad (4)$$

运行在混乱的状态下,混乱的顺序 $\{x_k : k = 0, 1, \dots\}$ 可以作为随机数的顺序.

本文引入具有遍历性的混沌映射,将初始化变量转化为混沌变量,采用的正弦迭代公式为

$$\text{ch}_{k+1} = \sin(\pi \text{ch}_k), \quad (5)$$

其中 $\text{ch}_k \in (0, 1), k = 0, 1, \dots, m, k$ 为迭代计数器, m 为最大的混沌迭代次数.

将正弦迭代公式(5)引入种群初始化过程中,采用混沌变量后的种群变量转换公式如下所示:

$$x_i^j = x_{\min}^j + \text{ch}_k^j (x_{\max}^j - x_{\min}^j), \quad (6)$$

其中 x_{\max}^j 和 x_{\min}^j 分别为初始化的上界和下界.

2) 反向学习方法.

在概率学中,通常一个解有 50% 的几率不如自身的反向解优秀,故在建立初始种群时检查其反向解可增大种群多样性以加快收敛速度.

在算法中,设粒子 i 的当前位置为 x_i , 反向位置为 \tilde{x}_i . 通过反向学习方法对初始种群变量取相反数转换为反向解种群,粒子 i 在第 j 维的位置为 x_{ij} , 则反向位置可定义为

$$\tilde{x}_{ij} = x_{\min}^j + x_{\max}^j - x_{ij}. \quad (7)$$

将以上混沌反向解初始化策略应用于种群初始化过程,具体步骤如下.

Step 1: 首先随机初始化两个种群 A, B (每个种群个数为 NP 个).

Step 2: 将 A 种群按照式(6)转化为混沌种群 A' , 同时将 B 种群按照式(7)转化为相反变量产生反向解种群 B' .

Step 3: 根据 A' 与 B' 适应度值的大小选择 NP 个较优个体作为初始种群.

2.2 邻域搜索改进策略

跟随蜂阶段,通过蜜蜂借鉴自身位置和历史最优位置作为更新前个体的最优位置,并在更新前个体最优位置邻域内引入量子策略展开搜索,以加快算法的收敛速度和精度.

在量子空间中^[18],粒子的速度和位置不能同时获得,因此借鉴波函数 $\psi(x, t)$ 来描述粒子的状态.波函数的物理意义是:波函数模的平方是粒子在空间某点出现的概率密度,即

$$dx dy dz = Q dx dy dz, \quad (8)$$

其中 Q 为概率密度函数,且满足归一化条件.在量子空间中粒子运动的动力学方程式为

$$i\hbar \frac{\partial}{\partial t} \psi(X, t) = \left(-\frac{\hbar^2}{2m} \nabla^2 + V(X) \right) \psi(X, t). \quad (9)$$

其中: i 为虚数单位, \hbar 为普朗克常数, ∇ 为拉普拉斯算子, m 为粒子质量, $V(X)$ 为粒子所在的势场.

将量子行为引入人工蜂群算法中,根据 δ 势阱的量子行为模拟人工蜂群求解最优值.求解最优值即适应度最小值,需获得蜜蜂的准确位置,故根据蒙特卡罗方法又称逆变换法,得到粒子相对收敛点 P (即准确位置)的一维 δ 势阱的随机运动方程如下所示:

$$X = P \pm \frac{L}{2} \ln(1/\tau). \quad (10)$$

其中: $L = 1/\beta = \hbar^2/m\gamma$ 为 δ 势阱的步长, \hbar 为普朗克常数, m 为粒子的质量, τ 为 $(0, 1)$ 之间均匀分布随机数.应用到 N 维空间下,单个蜜蜂位置的收敛性可归为一维空间蜜蜂的收敛性.

通常人工蜂群系统最终会收敛到某一点.将整个蜂群看作量子系统,蜂群中每个蜜蜂都具有量子行为.定义量子系统中 δ 势阱的步长 L 为蜂群中当前全局历史最优位置与第 i 个食物源位置之间的距离 L' , 有

$$L' = 2\alpha(m_{\text{best}} - x_i). \quad (11)$$

蜂群中蜜蜂相对 A_i 点的 N 维 δ 势阱的随机运动方程,即为蜜蜂的位置更新方程,有

$$V_i = A_i + \alpha \times (m_{\text{best}} - x_i) \times \ln(1/\tau), \quad (12)$$

其中 A_i 为当前个体最优值.蜜蜂记录自己位置并借鉴历史最优位置作为当前个体最优位置,即通过交叉率选取权重比帮助蜜蜂在自身和社会位置基础上搜索最佳位置.当前个体最优位置如下所示:

$$A_i = Y \times x_i + (1 - Y) \times \text{Queen}_i. \quad (13)$$

其中: Queen 为历史最优解的位置;交叉率 $Y = \exp\left(-\frac{\text{iter}}{\text{max Cycle}}\right)$, 是通过选取父母的某些基因产生新的个体,即产生具有上代特征质量的解, iter 是当前迭代次数, max Cycle 是最大迭代次数. α 为控制参数,对 D 维的每一个变量 α_j 均有 $\alpha_j = (-1) \times \omega_j \times \varphi_j$, 整个种群的历史最优解的平均值

$$m_{\text{best}} = \frac{1}{\text{iter}} \sum_{i=1}^{\text{iter}} x_{i\text{best}},$$

$x_{i\text{best}}$ 是指比 i 更优的历史最优解, ω_j 是 $[0, 1]$ 的随机整数, φ_j 是通过混沌序列产生的 chaotic 计算,有

$$\text{chaotic}1_j = 4 \times r_{2,j} \times (1 - r_{2,j}), \quad (14)$$

$$\text{chaotic}2_j = 4 \times r_{3,j} \times (1 - r_{3,j}), \quad (15)$$

$$\varphi_j = \left| 1 - \frac{\text{chaotic}1_j}{\text{chaotic}2_j} \right|, \quad (16)$$

其中 $r_{2,j}$ 和 $r_{3,j}$ 均为 $[0, 1]$ 之间的随机数.

在迭代初期, Y 较大有利于扩大搜索范围,增加种群的多样性;在迭代后期, Y 较小有利于收敛到全

局最优解.跟随蜂在跟随自身最优位置的同时,以全局的最优位置为导向,加快蜜蜂向着最优解的方向前进,提高算法的收敛速度.

2.3 ISABC的实现流程

ISABC算法具体流程如图1所示.

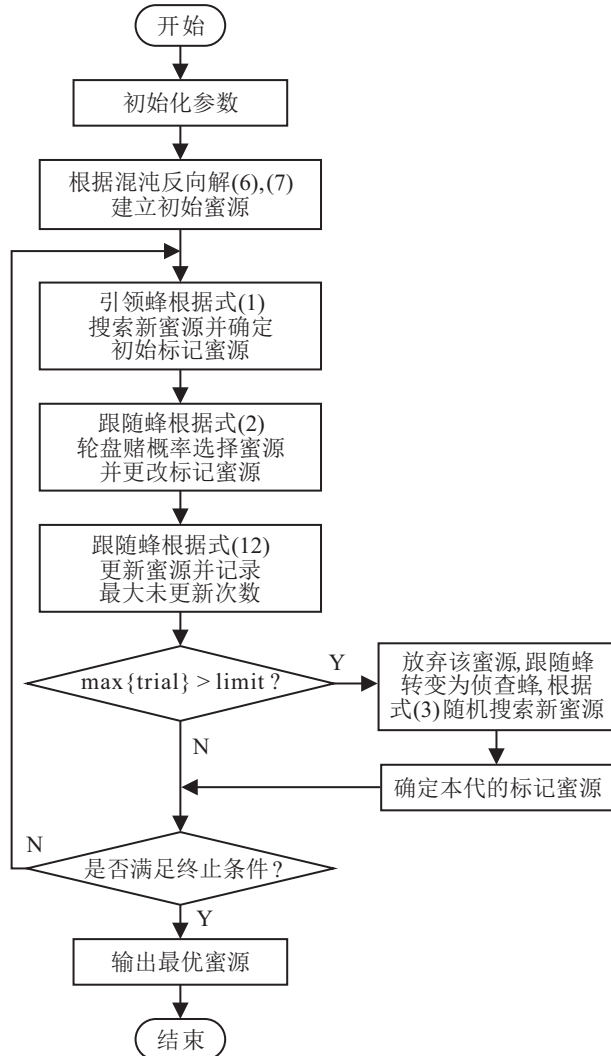


图1 ISABC算法流程

2.4 ISABC算法的复杂度分析

ABC算法的时间复杂度为 $O(\text{Maxcycle} \times \text{NP})$, ISABC算法相比于ABC算法,将随机初始化改进为混沌反向解初始化,由初始化步骤可知,其时间复杂度为 $O(\text{Maxcycle} \times \text{NP} \times D + \text{NP} \times D)$.其中:Maxcycle为最大迭代次数, NP为种群个数, D 为粒子维数.同时,跟随蜂阶段提出邻域搜索改进策略, ISABC算法只是将ABC算法的更新公式改变,故其时间复杂度与ABC算法一致.可见, ISABC改进算法的时间复杂度为 $O(\text{Maxcycle} \times \text{NP} \times D)$.

对于ISABC算法的空间复杂度,混沌反向解初始化需要NP个个体,个体的维数为 D ,故其空间复杂度为 $O(\text{NP} \times D)$,与ABC算法一致.

3 实验仿真与分析

3.1 实验测试函数

为了测试ISABC算法的优化性能,采用12个标准测试函数,并将ISABC与ABC、IABC^[19]、GABC^[20]、SABC^[21]、qABC^[22]算法、粒子群算法(PSO)和蚁群算法(ACO)进行比较.12个测试函数的具体信息如表1所示.

表1 基本测试函数

编号	函数名	是否多峰	搜索范围	最优值
f_1	Quartic	否	$[-1.28, 1.28]$	0
f_2	Rastrigin	是	$[-5.12, 5.12]$	0
f_3	Sphere	否	$[-100, 100]$	0
f_4	Alpine	是	$[-10, 10]$	0
f_5	Griewank	是	$[-600, 600]$	0
f_6	Schewefel2.22	否	$[-10, 10]$	0
f_7	Schaffer	否	$[-10, 10]$	0
f_8	Sumpower	是	$[-10, 10]$	0
f_9	Sumsquares	否	$[-10, 10]$	0
f_{10}	Ackley	是	$[-32, 32]$	0
f_{11}	Michalewicz	是	$[0, \pi]$	-0.8013
f_{12}	Levy	是	$[-10, 10]$	0

3.2 实验平台及参数设定

实验设备为一般PC机, Pentium(R)Dual-Core CPU 2.93 GHz, 2.00 GB内存,比较算法基于Matlab 2014a实现.采用文献[19]的第2组实验参数,具体设置如下:蜜蜂种群数 $\text{NP} = 40$,最大未更新次数 $\text{Limit} = \text{NP} \times D$.实验测试分别对函数在50维和100维下进行仿真,最大迭代次数为5000.比较算法的其余参数均参照原文献设定.对每个测试函数分别独立运行30次,取其最优值(Best)、平均值(Mean)、方差(Std)和运行时间(Time(s))进行比较,函数50维下测试结果比较如表2所示.

由表2可以看出,对于所有函数, ISABC与ABC、PSO和ACO比较,其收敛精度和稳定性均优于3种算法,特别是 f_2 、 f_5 和 f_7 函数, ISABC算法的收敛精度远超过3种算法达到理论最优值;对于函数 f_4 、 f_8 、 f_{10} 和 f_{11} , ISABC算法的优化性能都远超过3种算法;对于函数 f_{12} , ISABC算法的最优值和平均值更是显著;对于所有函数, ISABC算法在运行时间上相比3种基本算法相差无几.可见, ISABC算法的优化性能明显高于3种算法.

ISABC与其他改进算法比较:对于函数 f_1 、 $f_3 \sim f_{11}$, ISABC算法的收敛精度和稳定性均优于其他改进算法;对于函数 f_2 、 f_5 和 f_7 , ISABC和IABC均达到理论最优值,且具有很强的稳定性;特别是对于函数 f_4 、 f_8 和 f_{11} , ISABC算法的收敛精度相对其他改进算

法有很大提高;对于函数 f_{12} , ISABC 最优值和平均值均远超过其他改进算法;对于所有函数, ISABC 算法

在时间效率上均优于其他改进算法. 故 ISABC 算法的优化性能明显优于其他改进算法.

表2 ISABC 算法测试结果比较 ($D = 50$)

函数		ABC	PSO	ACO	IABC	GABC	SABC	qABC	ISABC
f_1	Best	6.07e-16	1.26e-09	6.21e-05	2.28E-16	3.02E-16	2.20E-16	2.77e-16	5.68e-18
	Mean	1.30e-15	4.11e-08	8.14e-05	2.98e-16	5.70e-16	2.94e-16	3.77e-16	7.91e-18
	Std	2.72e-31	3.84e-26	3.78e-30	1.21e-33	2.64e-32	1.28e-33	7.269e-33	2.227e-34
	Time	12.54	17.5	13.52	48.69	36.40	33.64	29.15	18.16
f_2	Best	0.00e+00	2.01e+00	8.77e+01	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
	Mean	6.62e-11	6.54e+01	1.46e+01	0.00e+00	4.35e-12	3.55e-16	1.78e-16	0.00e+00
	Std	3.32e-20	2.21e-26	5.33e-19	0.00e+00	4.53e-23	5.61e-31	3.15e-31	0.00e+00
	Time	10.56	18.62	16.25	32.61	24.58	21.72	31.62	14.65
f_3	Best	1.54e-14	2.69e-02	9.56e-02	4.66e-16	6.57e-15	4.50e-16	8.73e-16	2.38e-18
	Mean	8.13e-13	7.21e-02	8.95e-01	6.82e-16	8.26e-15	7.78e-16	9.94e-16	3.26e-17
	Std	1.11e-28	6.35e-30	6.32e-28	1.04e-32	1.09e-23	4.54e-32	1.88e-32	3.55e-32
	Time	10.57	20.71	19.34	37.85	29.49	26.41	19.54	13.12
f_4	Best	1.68e-05	3.33e-03	6.34e-04	1.53e-15	1.02e-08	5.02e-09	1.17e-07	3.28e-16
	Mean	5.02e-05	5.96e-02	5.14e-03	7.66e-12	4.15e-08	1.43e-06	2.51e-06	4.05e-16
	Std	1.08e-09	8.69e-21	5.88e-28	1.74e-18	1.44e-11	1.80e-11	6.31e-09	1.09e-32
	Time	18.55	23.64	25.81	46.28	35.21	29.58	27.33	22.18
f_5	Best	7.22e-15	5.66e-04	7.33e-03	0.00e+00	9.99e-16	0.00e+00	2.22e-16	0.00e+00
	Mean	3.21e-11	6.74e-04	7.64e-03	1.11e-17	1.22e-14	9.77e-14	5.00e-15	0.00e+00
	Std	8.67e-21	4.12e-35	7.11e-32	1.23e-33	1.04e-27	9.43e-26	5.63e-29	0.00e+00
	Time	13.48	15.44	16.28	39.25	26.88	37.21	29.45	17.34
f_6	Best	9.47e-13	2.65e-01	8.46e-10	9.97e-16	3.33e-14	4.56e-14	4.17e-11	5.14e-16
	Mean	2.22e-12	8.21e-01	7.14e-09	1.43e-15	8.76e-14	1.63e-13	1.43e-10	5.98e-16
	Std	1.45e-24	1.33e-29	9.56e-28	4.75e-32	3.15e-27	1.88e-26	6.62e-21	9.22e-33
	Time	15.58	14.54	15.36	34.28	29.65	28.99	29.48	17.43
f_7	Best	1.66e-07	3.31e-05	4.15e-04	5.55e-17	1.88e-15	1.48e-11	2.31e-14	0.00e+00
	Mean	3.17e-05	3.55e-03	8.11e-04	2.15e-08	3.53e-07	2.81e-06	3.58e-12	0.00e+00
	Std	4.06e-13	7.41e-19	6.49e-18	2.37e-09	1.07e-15	1.84e-11	8.24e-11	0.00e+00
	Time	12.71	15.0	13.6	36.84	29.11	34.21	19.62	16.54
f_8	Best	9.35e-11	7.21e-10	5.41e-08	7.14e-18	2.87e-17	2.67e-17	2.39e-17	1.17e-19
	Mean	4.23e-09	9.54e-09	2.11e-07	1.84e-17	4.33e-17	4.65e-17	3.99e-17	1.46e-19
	Std	8.59e-18	6.54e-30	3.54e-32	5.63e-35	5.92e-35	2.29e-34	1.35e-34	6.36e-36
	Time	16.34	25.19	13.25	49.26	19.64	28.15	21.33	18.64
f_9	Best	2.48e-15	5.11e-02	8.21e-01	5.18e-16	6.65e-16	5.20e-16	7.64e-16	1.72e-18
	Mean	4.72e-15	9.11e-01	9.51e-01	5.72e-16	7.59e-16	7.16e-16	9.54e-16	5.25e-18
	Std	3.87e-30	3.54e-21	8.45e-19	4.13e-33	4.20e-33	2.91e-32	1.27e-32	3.25e-35
	Time	12.59	13.54	14.65	31.25	28.64	29.33	24.31	15.55
f_{10}	Best	1.35	5.74e+00	6.54e-02	8.35e-13	3.26e-14	3.23e-15	9.16e-15	7.66e-16
	Mean	2.36	8.11e+00	6.87e-01	1.82e-12	5.22e-13	9.12e-14	8.84e-14	4.35e-15
	Std	5.81e-01	3.54e-02	4.51e-26	2.67e-28	4.55e-27	3.85e-27	4.46e-28	6.31e-29
	Time	17.36	18.5	14.38	37.91	28.36	29.14	28.74	18.52
f_{11}	Best	-46.64	-13.56	-23.59	-49.61	-35.73	-35.44	-33.86	-17.54
	Mean	-47.34	-12.62	-23.92	-49.61	-37.12	-35.70	-34.32	-18.67
	Std	2.55e-01	1.45e-01	4.36e-02	5.08e-03	19.28	8.56e-03	1.42e-01	1.41e-29
	Time	10.43	15.62	12.25	39.22	23.15	24.59	21.44	12.28
f_{12}	Best	3.08e-12	5.12e-05	3.88e-02	4.97e-16	1.17e-15	1.64e-15	1.64e-15	1.35e-31
	Mean	3.44e-11	4.22e-05	9.61e-01	5.37e-16	1.81e-14	6.27e-14	5.27e-14	5.26e-25
	Std	6.46e-22	6.33e-24	9.61e-20	2.44e-33	6.01e-28	1.15e-26	6.25e-27	5.68e-22
	Time	14.66	16.45	15.31	29.95	27.66	18.35	24.82	16.54

为了说明ISABC算法在收敛速度上也有很大优势,选取其中6个测试函数,并与ABC、PSO、ACO及

改进算法在100维下进行比较,测试结果比较如图2所示。

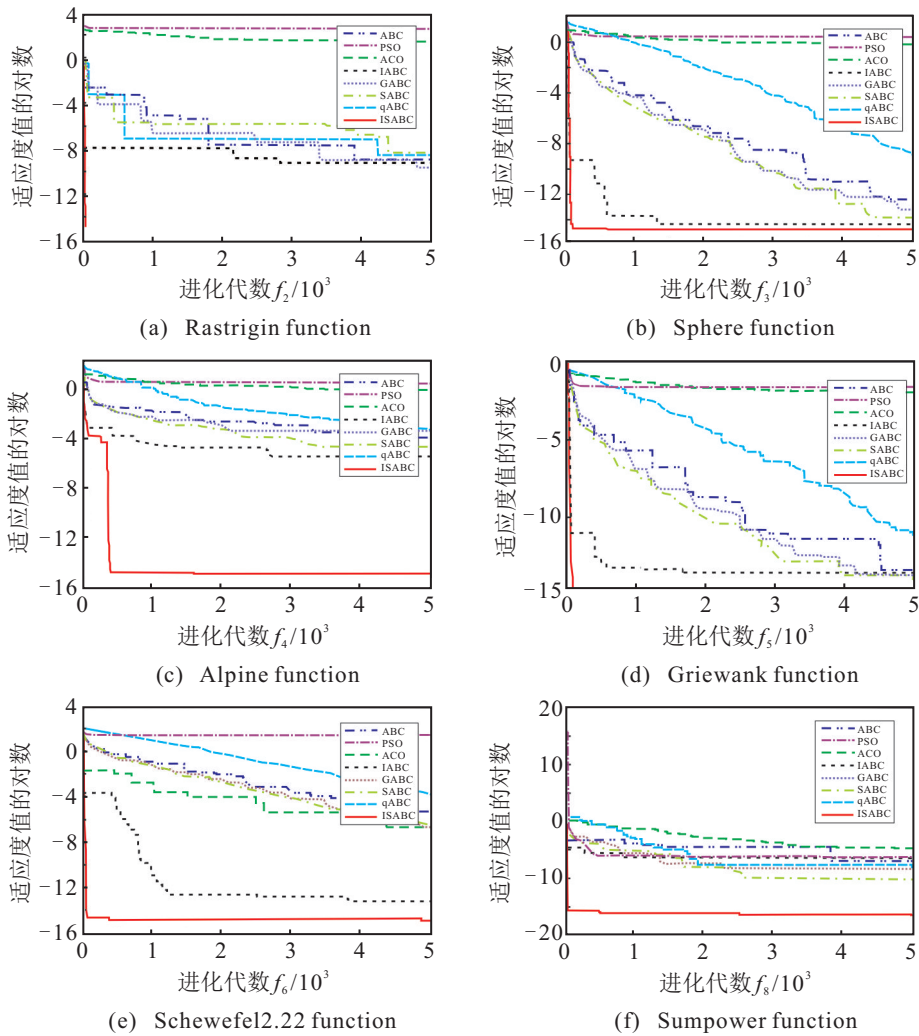


图2 ISABC算法测试函数优化迭代曲线比较 ($D = 100$)

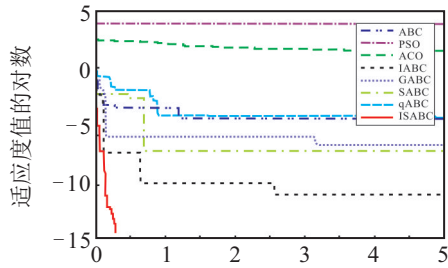
表3 高维函数下ISABC算法优化性能对比 ($D = 400$ 和 $D = 800$)

函数	D	ABC	PSO	ACO	IABC	GABC	SABC	qABC	ISABC	
f_2	400	Best	2.46e-05	3.55e-04	6.54e-04	3.73e-14	3.91e-07	3.71e-08	6.63e-08	0.00e+00
		Mean	1.76e-04	9.58e-03	8.21e-03	1.22e-14	1.31e-05	5.58e-06	5.59e-06	0.00e+00
		Std	4.93e-08	6.33e-33	7.21e-35	5.31e-35	3.73e-10	6.74e-11	6.19e-11	0.00e+00
		Time	26.59	29.48	28.66	68.51	49.22	51.49	38.69	29.48
	800	Best	5.83e-05	3.94e-03	5.11e-03	6.35e-14	1.02e-08	2.45e-09	4.52e-07	0.00e+00
		Mean	6.36e-04	3.71e-02	8.77e-02	8.16e-14	1.19e-05	9.99e-06	9.17e-06	0.00e+00
		Std	5.25e-07	6.99e-34	9.14e-30	4.22e-34	3.89e-10	7.32e-11	3.08e-10	0.00e+00
		Time	49.36	33.58	38.64	84.65	58.52	73.56	47.31	32.14
f_3	400	Best	9.46e-05	1.66e-05	3.97e-06	8.69e-13	1.65e-05	3.44e-06	1.46e+00	6.58e-15
		Mean	5.10e-04	4.23e-04	5.88e-05	5.21e-12	4.09e-04	4.24e-05	1.78e+00	8.87e-15
		Std	2.04e-07	9.51e-29	5.44e-30	3.55e-29	1.41e-19	1.05e-09	4.39e-02	6.35e-34
		Time	20.94	23.66	18.58	69.81	37.39	39.48	34.16	21.52
	800	Best	5.62e-03	6.44e-04	7.14e-05	7.66e-12	1.27e-03	5.44e-03	6.78e+01	1.68e-14
		Mean	1.78e-02	8.77e-03	8.11e-04	5.33e-11	1.41e-03	6.51e-03	8.66e+01	1.78e-14
		Std	8.57e-05	5.14e-29	1.66e-28	5.62e-28	1.05e-08	8.12e-07	5.77e+00	7.58e-31
		Time	21.55	25.62	20.29	95.36	44.68	41.22	37.98	23.33

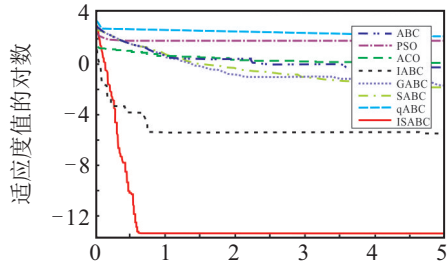
由图2可以看出,对于6个函数,相比于其他算法,ISABC算法收敛到一定精度所需的迭代次数最少,且收敛精度最高.从50维变到100维时,对于函数 f_2 和 f_5 ,ISABC算法依然能收敛到理论最优,且在迭代早期便已收敛;对于函数 f_3 和 f_6 ,ISABC相对于IABC算法在收敛精度上差异不大,但ISABC算法在迭代早期即可找到与IABC算法同等精度的值.相比其他算法,ISABC算法随着维数增加,收敛精度和收敛速度较为稳定.

3.3 高维函数下的算法性能仿真分析

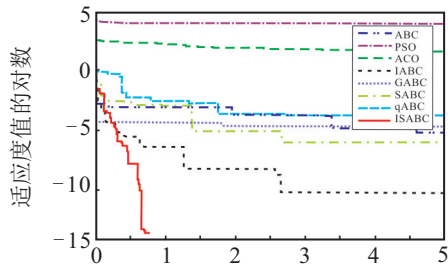
为了更清晰地说明ISABC算法在更高维数下的



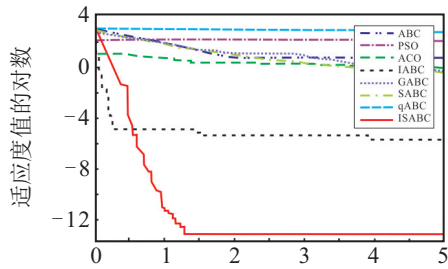
(a) Rastrigin function ($D = 1000$)



(b) Sphere function ($D = 1000$)



(c) Rastrigin function ($D = 2000$)



(d) Sphere function ($D = 2000$)

图3 高维函数下ISABC算法收敛曲线比较

优化性能,将ISABC算法与其他对比算法在不同高维数下进行仿真分析,由于篇幅有限,采用函数 f_2 和 f_3 进行仿真.表3为算法在 $D = 400$ 和 $D = 800$ 下的优化性能对比,图3为算法在 $D = 1000$ 和 $D = 2000$ 下的收敛迭代曲线.

由表3可以看出,在时间效率上,ISABC算法整体优于其他对比算法,两个函数在不同的高维环境下,ISABC算法的收敛精度均远超过其他算法,尤其对于函数 f_2 ,高维下的ISABC算法依旧收敛到理论最优.同时,由图3可以看出:ISABC算法的收敛速度和收敛精度远优于其他算法,对于函数 f_2 ,在1000维或2000维下,ISABC算法的收敛精度均在迭代早期便收敛到理论最优.ISABC算法在高维函数优化上具有明显优势.

4 结论

本文研究了一种邻域搜索改进策略的人工蜂群算法,通过在ABC初始化阶段引入混沌反对解策略,增大种群多样性,加强跳出局部最优的能力.在观察蜂阶段,提供当前个体最优位置,同时当前蜜源领域内采用量子策略更新搜索,使观察蜂高效率的搜索解的质量.通过对12个测试函数进行实验仿真分析,表明ISABC改善了ABC易陷入局部最优的缺陷,加快了算法收敛到全局最优解的速度,且具有一定的稳定性,在优化性能和效率上相比粒子群算法、蚁群算法及其他ABC改进算法均有很好的改善.

参考文献(References)

- [1] Karaboba D. An idea based on honey bee swarm for numerical optimization[R]. Kayseri: Faculty of Engineering, Erciyes University, 2005.
- [2] Dervis Karaboga, Bahriye Akay. A comparative study of artificial bee colony algorithm[J]. Applied Mathematics and Computation, 2009, 214(1): 108-132.
- [3] Shi Y J, Pun C M, Hu H D, et al. An improved artificial bee colony and its application[J]. Knowledge-Based Systems, 2016, 107(5): 14-31.
- [4] 王永琦, 吴飞, 孙建华. 求解连续空间优化问题的改进蜂群算法[J]. 计算机应用研究, 2017, 35(3): 1008-1013.
(Wang Y Q, Wu F, Sun J H. Improved swarm optimization algorithm for continuous spatial optimization problems[J]. Application Research of Computer, 2017, 35(3): 1008-1013.)
- [5] Cai S H, Long W, Jiao J J. Hybridizing artificial bee colony with biogeography-based optimization for constrained mechanical design problems[J]. Springer,

- 2015, 22(6): 2250-2259.
- [6] Xu H D, Jiang M Y, Xu K. Archimedean copula estimation of distribution algorithm based on artificial bee colony algorithm[J]. IEEE Systems Engineering and Electronics, 2015, 26(2): 388-396.
- [7] Huang L P, Zhang B, Yuan X, et al. Solving service selection problem based on a novel multi-objective artificial bees colony algorithm[J]. Shanghai Jiaotong University, 2017, 22(4): 474-480.
- [8] 毕晓君, 王艳娇. 加速收敛的人工蜂群算法[J]. 系统工程与电子技术, 2011, 33(12): 2755-2761.
(Bi X J, Wang Y Q. Artificial swarm algorithm for accelerating convergence[J]. Systems Engineering and Electronics Technology, 2011, 33(12): 2755-2761.)
- [9] Shimipi S J, Ritu T, Harism S, et al. Hybrid artificial bee colony algorithm with differential evolution[J]. Applied Soft Computing, 2017, 58(5): 11-24.
- [10] Weifeng G, Lingling H, Jue W, et al. Enhanced artificial bee colony algorithm through differential evolution[J]. Applied Soft Computing, 2016, 48(6): 137-150.
- [11] Kiran M S, Hakli H, Gunguz M. Artificial bee colony algorithm with variable search strategy for continuous optimization[J]. Information Sciences, 2015, 300(1): 140-157.
- [12] Gu W J, Yu Y G, Hu W. Artificial bee colony algorithm-based parameter estimation of fractional-order chaotic system with time delay[J]. IEEE/CAA J of Automatica Sinica, 2017, 4(1): 107-113.
- [13] Ajit K, Dharmender K, Jarial S K. A novel hybrid K -means and artificial bee colony algorithm approach for data clustering[J]. Decision Science Letters, 2018, 7(1): 65-76.
- [14] Changsheng Z, Dantong O Y, Jiayu N. An artificial bee colony approach for clustering[J]. Expert Systems with Applications, 2010, 37(7): 4761-4767.
- [15] 王志刚, 尚旭东, 夏慧明, 等. 多搜索策略协同进化的人工蜂群算法[J]. 控制与决策, 2018, 33(2): 235-241.
(Wang Z G, Shang X D, Xia H M, et al. Artificial bee colony algorithm with multiple-search strategies cooperative evolutionary[J]. Control and Decision, 2018, 33(2): 235-241.)
- [16] Li M D, Zhao H, Wang X W, et al. Artificial bee colony algorithm with comprehensive search mechanism for numerical optimization[J]. IEEE Systems Engineering and Electronics, 2015, 26(3): 603-617.
- [17] 李国亮, 魏振华, 徐蕾. 分阶段搜索的改进人工蜂群算法[J]. 计算机应用, 2015, 35(4): 1057-1061.
(Li G L, Wei Z H, Xu L. The improved artificial swarm algorithm for the phased search[J]. Computer Application, 2015, 35(4): 1057-1061.)
- [18] 王建, 丁学明, 董新燕. 基于量子策略的布谷鸟搜索算法研究[J]. 电子科技, 2015, 28(12): 40-44.
(Wang J, Ding X M, Dong X Y. Research of cuckoo search algorithm based on quantum strategy[J]. Electronic Science and Technology, 2015, 28(12): 40-44.)
- [19] 王冰. 基于局部最优解的改进人工蜂群算法[J]. 计算机应用研究, 2014, 31(4): 1023-1026.
(Wang B. Improved artificial swarm algorithm based on local optimal solution[J]. Computer Application Research, 2014, 31(4): 1023-1026.)
- [20] Guopu Z, Sam K. Gbest-guided artificial bee colony algorithm for numerical function optimization[J]. Applied Mathematics and Computation, 2010, 217(7): 3166-3173.
- [21] 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): 1402-1406.
(Luo J, Xiao X H, Fu L, et al. Improved swarm algorithm based on segmented search strategy[J]. Control and Decision, 2012, 27(9): 1402-1406.)
- [22] Dervis K, Beyza G. A quick artificial bee colony (qABC) algorithm and its performance on optimization problems[J]. Applied Soft Computing, 2014, 23(5): 227-238.

(责任编辑: 郑晓蕾)