

加强局部搜索能力的人工蜂群算法

刘琨^a, 封硕^b

(长安大学 a.理学院;b.工程机械学院,西安 710061)

摘要:针对人工蜂群算法在求解过程中存在收敛速度慢、易陷入局部最优解等缺点,提出了基于加强局部搜索策略的人工蜂群算法(ABC Based On Enhancing Local Search Ability, LSABC).一方面,在雇佣蜂搜索阶段,利用两种不同的搜索公式得到两组解,并将适应度最佳者作为候选解,增加解的多样性;同时,在搜索公式中加入个体的双重认知能力平衡算法的勘探和开发能力.另一方面,在侦察蜂搜索阶段,采用禁忌搜索策略,将局部极值存入禁忌表中,帮助算法跳脱局部最优解,达到避免算法早熟的同时加快算法收敛速度的目的.由于 LSABC 算法的改进与粒子群算法相似,为验证 LSABC 算法的寻优性能,针对 8 个经典基准函数,选取标准 ABC 算法、PSO 算法、EABC 算法、RLPSO 算法及 LSABC 算法分别进行对比测试.计算实验结果表明,LSABC 算法在求解精度和收敛速度方面明显提高,易于跳脱局部最优解.

关键词:人工蜂群算法;局部搜索能力;权重因子;禁忌搜索策略

中图分类号:TP18

文献标志码:A

人工蜂群算法(Artificial Bee Colony, ABC)是模仿蜜蜂群体采蜜行为提出来的一种启发式智能优化算法,与遗传算法(Genetic Algorithm, GA)^[1]、蚁群算法(Ant Colony Optimization, ACO)^[2]和粒子群算法(Particle Swarm Optimization, PSO)^[3]相比,由于 ABC 算法具有参数少、精度高和结构简单易实现等优点,近几年来受到广泛关注,并被广泛应用到人工神经网络^[4]、特征选择^[5]、数据挖掘^[6]、生产调度^[7]、信息处理^[8]和聚类分析^[9]等领域.

与其他智能优化算法相似,在解决复杂优化函数时,ABC 算法存在收敛速度较慢、易陷入局部最优解等缺点.许多学者对其进行改进,主要尝试在提高算法的收敛速度、增加种群的多样性及改进算法的寻优机制等方面.文献[10]受粒子群算法的影响,在搜索公式中加入最优解指导雇佣蜂的局部搜索,加快算法的收敛速度;文献[11]将最优解的适应度值作为更新的误差矫正乘数,加快算法的收敛速度;文献[12]引入排斥因子和时变权重因子,平衡算法的勘探和开发能力;文献[13]引入更多的搜索信息指导搜索策略,增强解的多样性;文献[14]通过控制扰动频率和扰动幅度,平衡算法的勘探和开发能力;文献[15]提出了惯性权重和加速度系数,修正算法的搜索过程,加快算法的收敛速度.

以上研究在很大程度上提升了算法的寻优效率,但仍存在不足之处:首先,种群的搜索机制仅利用一个搜索公式随机选择新解,导致解的多样性不足,不利于算法求得潜在最优解.其次,搜索过程中仅利用先前个体的信息产生候选解,这种机制利于算法的勘探能力,不利于算法的开发能力.最后,以上改进对帮助算法跳脱局部最优解的能力没有得到根本的改善,导致算法优化多峰函数时容易陷入局部最优解.为解决这些问题,本研究在雇佣蜂搜索阶段中利用两个不同的搜索方程得到两组解,比较适应度值最优的个体作为雇佣蜂的候选解,增加解的多样性.同时,在搜索策略中加入个体双重认知能力和权重因子,平衡算法的勘探和开发能力,加快人工蜂群算法的收敛速度.最后,在侦察蜂搜索阶段中加入禁忌搜索策略,帮助算法跳脱局部最优

收稿日期:2020-03-12;**修回日期:**2020-05-08.

基金项目:国家自然科学基金(11971075);重点科研平台水平提升项目(300102258510).

作者简介:刘琨(1996—),女,山东德州人,长安大学硕士研究生,研究方向为计算数学、智能优化算法, E-mail: chdliukun@163.com.

通信作者:封硕(1987—),男,陕西西安人,博士,研究方向为人工智能、机器人, E-mail: shuo_feng@yeah.net.

解,从而达到避免算法早熟、加快算法收敛速度的目的。

1 改进的人工蜂群算法

基于种群的智能优化算法应用于复杂的优化函数时,需要具备快速收敛、高多样性和易跳脱局部最优解等特点.为使ABC算法更好地解决复杂的优化问题,本研究对局部搜索策略进行了改进,并在侦察蜂阶段加入禁忌搜索策略,帮助算法跳脱局部最优解,避免算法早熟。

1.1 基于改进的局部搜索策略

标准ABC算法的局部搜索策略是将旧解向(或远离)另一个随机个体移动而生成新的候选解,该随机个体是优解和差解的概率相同,所以新的候选解并不一定比先前个体质量更好.另一方面, R 是 $[-1,1]$ 均匀分布的随机数.因此,ABC算法的局部搜索策略善于勘探,但不善于开发.在实际运算中,这两者是相互矛盾的,想要实现良好的优化性能,必须保证勘探能力和开发能力得到平衡。

为解决该问题,本研究在搜索过程中加入个体的自我认知能力和社会认知能力(自我认知能力指个体对其自身经验的认知,代表算法的勘探能力;社会认知能力指个体向种群中其他已有经验的个体学习,代表算法的开发能力).与此同时,加入权重因子 w 平衡勘探与开发能力,提出了如下新的搜索公式:

$$v_{ij} = wx_{ij} + c_1 R(x_{\text{best}}^j - x_{ij}) + c_2 R(x_{kj} - x_{ij}), \quad (1)$$

$$w = w_{\max} - \frac{i}{M_c} (w_{\max} - w_{\min}), \quad (2)$$

(1)式中, w 为权重因子; c_1 和 c_2 为学习因子,取值为2时可以获得较好的实验结果; R 是 $[-1,1]$ 的随机数; x_{best}^j 表示当前种群中最优解; $k \in \{1,2,\dots,SN\}$, $j \in \{1,2,\dots,D\}$, k 和 j 都是随机选取的,且 $k \neq i$;(2)式中, w_{\max} 和 w_{\min} 分别为权重因子的最大值和最小值,通常 $w_{\max} = 1.0$, $w_{\min} = 0.5$ 可以获得较好的实验结果; i 表示当前迭代次数; M_c 表示最大的迭代次数。

算法初期阶段应侧重于勘探能力,此时权重因子取较大值;随着迭代次数的增加,算法后期阶段应侧重于开发能力,此时权重因子取较小值;(1)式的第2项为社会认知能力,通过最优解指导邻域搜索行为,以增强算法的开发能力;(1)式的第3项为自我认知能力,通过个体向外或向内延伸拓宽个体的搜索方向,以增强算法的勘探能力。

文献[13]提出了改进的EABC算法(Enhancing artificial bee colony algorithm),为提高算法的性能,对搜索公式加入更多的搜索信息,便于较快地得到一个更优的候选解.本研究为增加解的多样性,将其作为本研究算法的第2个搜索公式:

$$v_{ij} = x_{ij} + \alpha(x_{\text{best}}^j - x_{r_1j}) + \beta(x_{r_2j} - x_{r_2j}), \quad (3)$$

(3)式中, x_{best}^j 表示当前种群中最优解; $r_1, r_2 \in \{1,2,\dots,SN\}$, r_1 和 r_2 为随机个体且 $r_1, r_2 \neq i$; α 是 $[0,A]$ 均匀分布的随机数, A 为任意正整数; $\beta = N(0.3,0.3) \times \text{rand}(0,1)$, $N(0.3,0.3)$ 表示均值和方差都为0.3的随机数。

1.2 禁忌搜索策略

为改善算法易陷入局部最优解这一缺点,可尝试的途径有:爬山法、进化计算、变结构邻域搜索、禁忌搜索(Tabu Search, TS)策略[16]等.其中,禁忌搜索具有算理清晰、易于编程实现和避免迂回搜索等优点,是一个用来跳脱局部最优解的搜索方法.其主要思想是标记已经得到的局部最优解,并在以后的迭代中避开已标记的局部最优解。

禁忌搜索算法流程。

步骤1 禁忌列表置空;

步骤2 判断算法是否满足停止原则;

步骤3 若算法满足停止原则,将局部最优解存入禁忌表中,作为禁忌对象;否则,将非禁忌对象对应的最优解作为当前算法的最优解;

本研究在侦察蜂阶段加入禁忌搜索策略:食物源的收益率没有通过预先确定的循环次数(Limit)得到更

新,即雇佣蜂陷入局部最优解时,该处食物源被放弃,侦察蜂寻找新解,帮助被束缚的个体跳脱局部最优解;与此同时,采用禁忌表记录局部最优解的位置,侦察蜂产生的新解与禁忌表中的元素比对.若新解存在于禁忌表中,则需要再次进行局部搜索得到新解;否则,算法继续.通过加入禁忌搜索策略,帮助算法跳脱局部最优解,避免算法早熟.

1.3 LSABC 算法的步骤

综上所述,本研究提出的 LSABC 算法主要步骤如下.

步骤 1 在 D 维空间中随机生成 SN 个初始解;

步骤 2 雇佣蜂在初始位置附近利用两种不同的搜索公式(1)、(3)进行局部邻域搜索;

步骤 3 计算解的目标函数,并计算解的适应度值,贪婪选择适应度值较优的食物源;

步骤 4 全部的雇佣蜂完成一次局部搜索后,观察蜂根据适应度值按照概率选择食物源,并在该食物源附近进行邻域搜索,寻找新的食物源,贪婪选择较优的食物源;

步骤 5 如果食物源的收益率没有通过预先确定的循环次数(Limit)得到改善时,即算法陷入局部最优解时,使用禁忌表存储其位置.该处食物源被放弃,与之对应的雇佣蜂转化为侦察蜂.侦察蜂随机产生新解并与禁忌表中的元素比对,若新解存在于禁忌表中,重新产生新解;

步骤 6 若迭代次数小于最大迭代次数,转至步骤 2;否则,输出最优解,算法结束.

2 仿真实验及结果分析

为测试所提算法的性能,选用表 1 给出的基准函数^[17]进行测试.其中: $f_1 \sim f_3$ 为单峰函数,搜索空间内仅存在一个全局最小值,用于检验算法收敛速度和精度; f_4 为 Rosenbrock 函数,当维数为 2 或者 3 时,该函数为单峰函数,随着维数的增加,该函数可能有多个极小值点; $f_5 \sim f_8$ 为多峰函数,其局部最优解会随着问题维数的增加呈指数增长,用于检验算法的整体寻优能力和算法跳脱局部最优值的能力.

表 1 测试函数表达式、搜索空间和理论最优值

Tab. 1 Test function expressions, search spaces, and theoretical optimal values

函数表达式	搜索空间	理论最优值
$f_1 = \sum_{i=1}^n x_i^2$	$(-100, 100)^n$	0
$f_2 = \sum_{i=1}^n ix_i^2$	$(-10, 10)^n$	0
$f_3 = \sum_{i=1}^n (x_i + 0.5)^2$	$(-100, 100)^n$	0
$f_4 = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$(-5, 10)^n$	0
$f_5 = \sum_{i=1}^n [x_i^2 - 100\cos(2\pi x_i) + 10]$	$(-5.12, 5.12)^n$	0
$f_6 = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1$	$(-600, 600)^n$	0
$f_7 = -20\exp\left(-0.2\sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$	$(-32, 32)^n$	0
$f_8 = \sum_{i=1}^n x_i \sin(x_i) + 0.1x_i $	$(-10, 10)^n$	0

由于所提算法的改进与粒子群算法相似,为验证所提算法的寻优性能.针对表 1 所列的 8 个函数,选取赖兆林^[18]等人提出的 RLPSO 算法(Reverse Learning Particle Swarm Optimization),与标准 ABC 算法、PSO 算法、EABC 算法与 LSABC 算法分别进行对比测试.算法参数设置如下:6 种算法的种群规模(SN)为 100,阈值(Limit)为 300,最大迭代次数为 200,维数(D)分别为 50 和 200.其余参数设置为:PSO 算法中惯性权重因子 $\omega = 1.0$, c_1 和 c_2 取值为 2.0;RLPSO 算法中惯性权重因子 $\omega = 1.0$, c_1 和 c_2 取值为 2.0,方向选择概率 $\mu = 0.6$.

2.1 实验结果

统计 ABC 算法、PSO 算法、EABC 算法、RLPSO 算法与 LSABC 算法这 5 种算法的最优值、最差值、平均值和标准差,实验结果如表 2~5 所示.其中,平均值反映了算法的求解精度;标准差反映了算法的稳定性.

表 2 各算法在 50 维情况下对单峰函数测试的实验结果

Tab. 2 Experimental results of unimodal functions tested by each algorithm in 50 dimensions

函数	算法	最优值	最差值	平均值	标准差
f_1	ABC 算法	3.86E+03	6.86E+04	5.73 E+04	2.81E+03
	PSO 算法	4.71E-07	7.58E-06	8.55E-06	8.35E-07
	EABC 算法	4.83E+00	5.98E+01	9.75E+00	3.03E+00
	RLPSO 算法	7.53 E-91	5.75E-90	5.11E-91	2.18E-91
	LSABC 算法	3.64E-116	4.26E-115	7.83E-115	4.18E-116
f_2	ABC 算法	7.54E+00	5.08E+01	2.18E+01	3.44E+00
	PSO 算法	3.55E+05	8.04E+05	5.57E+05	4.41E+05
	EABC 算法	1.41E+02	5.41E+02	3.04E+02	2.83E+02
	RLPSO 算法	8.15E-90	5.05E-89	6.42E-89	9.06E-90
	LSABC 算法	2.57E-102	4.41E-101	5.06E-101	8.59E-102
f_3	ABC 算法	2.83E+01	5.75E+02	1.48E+02	7.51E+00
	PSO 算法	5.74E+02	3.95E+04	6.27E+03	8.91E+02
	EABC 算法	6.75E-0.2	4.25E+02	6.44E+01	1.05E+01
	RLPSO 算法	8.77E-0.6	8.54E+01	6.05E+00	2.05E+00
	LSABC 算法	6.12E-0.6	4.56E+00	2.21E+00	1.01E+00

表 3 各算法在 200 维情况下对单峰函数测试的实验结果

Tab. 3 Experimental results of unimodal functions tested by each algorithm in 200 dimensions

函数	算法	最优值	最差值	平均值	标准差
f_1	ABC 算法	4.29E+04	2.19E+05	8.43E+04	4.85E+03
	PSO 算法	3.86E+01	6.86E+01	5.03E+01	5.14E+00
	EABC 算法	6.54E+00	1.47E+01	8.71E+00	3.73E+00
	RLPSO 算法	6.68 E-90	5.51E-89	1.69E-90	9.21E-91
	LSABC 算法	5.15E-103	4.06E-102	2.28E-102	7.25E-103
f_2	ABC 算法	5.36E+03	6.33E+03	3.61E+03	3.39E+02
	PSO 算法	4.33E+06	3.95E+92	1.06E+92	6.14E+91
	EABC 算法	6.07E+01	3.78E+02	1.42E+02	4.33E+01
	RLPSO 算法	7.30E-89	2.19E-88	6.36E-88	6.06E-89
	LSABC 算法	3.62E-115	2.06E-114	5.09E-114	8.82E-115
f_3	ABC 算法	4.33E+01	6.08E+02	3.39E+02	7.54E+01
	PSO 算法	3.35E+02	6.15E+03	6.92E+02	4.13E+02
	EABC 算法	4.36E-0.2	8.19E+01	7.84E+00	9.33E+00
	RLPSO 算法	5.11E-0.6	5.06E+01	9.24E+00	1.15 E+00
	LSABC 算法	4.69E-0.6	2.39E+01	5.12E+00	1.57 E+00

表 4 各算法在 50 维情况下对多峰函数测试的实验结果

Tab. 4 Experimental results of multi-modal functions tested by each algorithm in 50 dimensions

函数	算法	最优值	最差值	平均值	标准差
f_4	ABC 算法	4.99E+05	2.37E+06	1.62E+06	8.02E+05
	PSO 算法	2.79E+03	8.18E+03	4.63E+03	6.05E+03
	EABC 算法	5.43E+01	3.84E+02	6.08E+01	2.91E+01
	RLPSO 算法	7.66E+00	2.69E+01	8.61E+00	5.95E+00
	LSABC 算法	3.64E-02	3.35E-01	7.16E-02	5.79E-02
f_5	ABC 算法	7.65E+01	5.71E+02	3.45E+02	6.24E+03
	PSO 算法	4.16E+02	7.09E+03	3.17E+06	1.42E+04
	EABC 算法	6.54E+01	3.81E+02	7.26E+04	2.94E+03
	RLPSO 算法	0	0	0	0
	LSABC 算法	0	0	0	0
f_6	ABC 算法	2.39E+00	6.44E+00	5.94E+00	3.08E+00
	PSO 算法	4.64E+00	2.06E+01	8.39E+00	8.71E+00
	EABC 算法	6.03E-03	1.75E-02	5.52E-02	6.38E-03
	RLPSO 算法	0	0	0	0
	LSABC 算法	0	0	0	0
f_7	ABC 算法	4.34E-02	1.64E-01	3.05E-01	5.44E-02
	PSO 算法	8.08E-01	4.32E+00	2.05E+00	6.64E-01
	EABC 算法	1.33E-03	3.41E-02	8.55E-02	1.02E-03
	RLPSO 算法	6.51E-18	1.34E-17	5.64E-17	6.81E-17
	LSABC 算法	6.04E-18	8.26E-17	3.94E-17	2.06E-18
f_8	ABC 算法	6.65E-01	2.96E+00	5.26E-01	7.25E-01
	PSO 算法	5.24E+02	7.14E+03	3.95E+03	8.11E+02
	EABC 算法	3.64E-02	6.12E-01	8.14E-01	4.15E-02
	RLPSO 算法	9.58E-46	5.73E-45	6.15E-45	8.31E-45
	LSABC 算法	8.26E-57	2.46E-56	4.59E-56	7.24E-57

2.2 实验结果分析

2.2.1 同维数不同算法的实验结果分析

不同算法在同一维数下对单峰函数实验结果(表 2)的数据表明,ABC 算法的稳定性差,收敛精度不高.通过分析表中 f_1 和 f_2 两个函数的实验结果可以得出,LSABC 算法的最优值、最差值、平均值和标准差的精度与 ABC 算法、PSO 算法、EABC 算法以及 RLPSO 算法的精度相比都有明显提高.因此说明,在收敛速度和精度方面,LSABC 算法优于其他算法,可以有效提高算法的局部搜索能力;分析表 2 中 f_3 函数的实验结果可以得出,LSABC 算法的最优值、最差值、平均值和标准差的精度与 ABC 算法、PSO 算法、EABC 算法相比都有提高但并不明显,与 RLPSO 算法相差不大.综上所述,在收敛速度和精度方面,LSABC 算法优于 ABC 算法、PSO 算法与 EABC 算法,可以提高算法的局部搜索能力.

不同算法在同一维数下对多峰函数实验结果(表 4)的数据表明,ABC 算法易陷入局部最优值、整体寻优能力差.分析表 3 中 f_4 的实验结果可以得出,LSABC 算法的最优值、最差值、平均值和标准差的精度与 ABC 算法、PSO 算法、EABC 算法以及 RLPSO 算法相比都有提高但并不明显.因此说明,在整体寻优及跳脱局部最优解方面,LSABC 算法优于另外 4 种算法;分析表 4 中 f_5 、 f_6 和 f_8 这 3 个函数的实验结果可以得出,LSABC 算法的最优值、最差值、平均值和标准差的精度与 ABC 算法、PSO 算法、EABC 算法以及 RLPSO 算

法相比都有明显提高.因此说明,在整体寻优及跳脱局部最优解方面,LSABC 算法明显优于另外 4 种算法;通过分析表 5 中 f_7 函数的实验结果可以得出,LSABC 算法的最优值、最差值、平均值和标准差的精度与 ABC 算法、PSO 算法、EABC 算法的精度相比都有明显的提高,与 RLPSO 算法相差不大.因此说明,在算法的整体寻优及跳脱局部最优解方面,LSABC 算法明显优于 ABC 算法、PSO 算法和 EABC 算法;综上所述,在算法整体寻优能力及跳脱局部最优解方面,LSABC 算法优于 ABC 算法,可以避免算法早熟.

表 5 各算法在 200 维情况下对多峰函数测试的实验结果

Tab. 5 Experimental results of multi-modal functions tested by each algorithm in 200 dimensions

函数	算法	最优值	最差值	平均值	标准差
f_4	ABC 算法	5.03E+02	6.27E+03	2.64E+03	4.22E+02
	PSO 算法	3.05E+01	6.58E+02	4.10E+02	6.43E+01
	EABC 算法	4.88E-01	5.48E+00	3.18E+00	2.24E-01
	RLPSO 算法	5.05E+00	6.41E+01	3.35E+01	4.65E+01
	LSABC 算法	5.06E-02	6.37E-01	8.28E-01	6.27E-02
f_5	ABC 算法	2.51E+03	8.05E+04	6.67E+04	4.20E+03
	PSO 算法	6.65E+04	5.74E+05	3.85E+05	4.61E+04
	EABC 算法	6.02E+03	4.28E+04	8.16E+03	5.04E+03
	RLPSO 算法	0	0	0	0
	LSABC 算法	0	0	0	0
f_6	ABC 算法	4.72E+00	5.47E+01	3.75E+01	5.06E+00
	PSO 算法	6.07E+00	6.25E+01	2.33E+01	5.54E+00
	EABC 算法	8.42E-01	4.54E+00	5.77E+00	8.75E+00
	RLPSO 算法	0	0	0	0
	LSABC 算法	0	0	0	0
f_7	ABC 算法	6.75E+00	4.25E+01	2.10E+01	3.04E+00
	PSO 算法	6.89E+01	5.45E+02	2.42E+02	5.64E+01
	EABC 算法	5.14E+00	4.92E+01	1.61E+01	6.22E+00
	RLPSO 算法	4.26E-15	3.32E-14	6.88E-14	8.68E-15
	LSABC 算法	3.66E-15	2.54E-14	4.64E-14	3.45E-15
f_8	ABC 算法	7.06E+01	6.22E+02	4.25E+02	4.04E+01
	PSO 算法	4.36E+03	6.05E+04	2.62E+04	7.45E+03
	EABC 算法	2.75E+00	5.22E+01	4.61E+01	5.25E+00
	RLPSO 算法	5.09E-45	6.48E-44	7.21E-44	3.55E-45
	LSABC 算法	5.44E-55	4.60E-54	8.47E-54	6.06E-55

2.2.2 同维数不同算法的实验结果分析

算法分别在 50 维和 200 维情况下,单峰函数实验结果表(表 2 和表 3)和多峰函数实验结果表(表 4 和表 5)中的数据表明:当维度为 100 时,LSABC 算法在统计的 8 个函数中, f_1, f_2, f_5, f_6 和 f_8 这 5 个函数的寻优指标值明显优于 ABC 算法、PSO 算法、EABC 算法和 RLPSO 算法的寻优指标值;当函数维数降到 50 维时,LSABC 算法的优势更为明显.因此说明,当函数的维数越高时,各算法的优势比低维数时各算法的优势要弱.

2.3 不同算法的性能分析

为了更直观地反映 LSABC 算法的性能,针对表 1 所列的 8 个函数,分别采用 ABC 算法、PSO 算法、

EABC 算法、RLPSO 算法和 LSABC 算法进行测试.图 1 至图 8 展示了维数为 200 时,5 种算法对测试函数的收敛曲线(为便于发现曲线间的差距,将各函数的适应度值取对数).

根据图 1 和图 2 的单峰函数进化曲线可以得出,在收敛速度和求解精度方面,LSABC 算法明显优于 ABC 算法、PSO 算法、EABC 算法和 RLPSO 算法.图 3 为不连续的步长函数,从图 3 中可以发现:LSABC 算法与 RLPSO 算法在收敛速度和求解精度方面相差不大,优于 ABC 算法、PSO 算法、EABC 算法.故 LSABC 算法加快了算法的收敛速度,提高了求解精度.

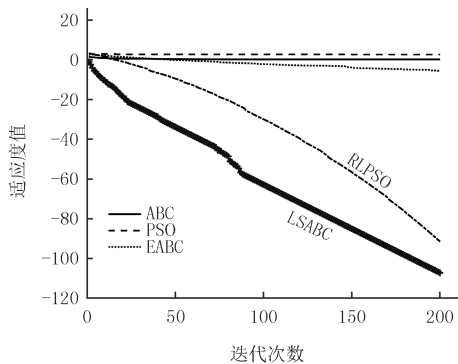


图1 函数1的进化曲线

Fig.1 The evolution curve of function 1

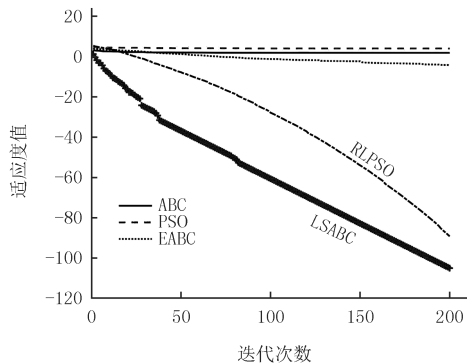


图2 函数2的进化曲线

Fig.2 The evolution curve of function 2

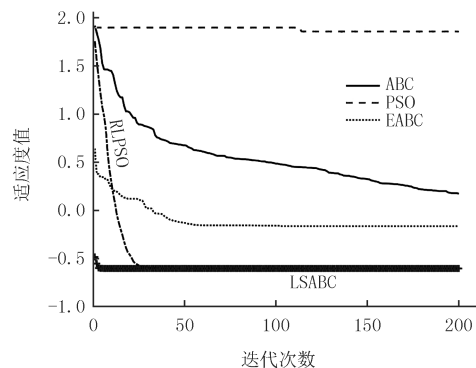


图3 函数3的进化曲线

Fig.3 The evolution curve of function 3

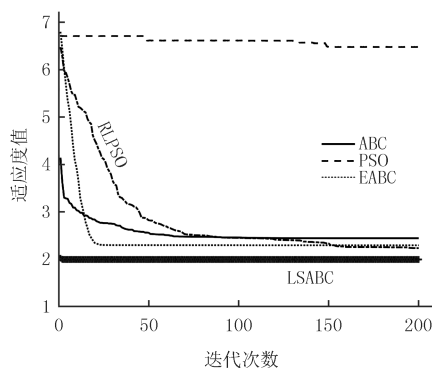


图4 函数4的进化曲线

Fig.4 The evolution curve of function 4

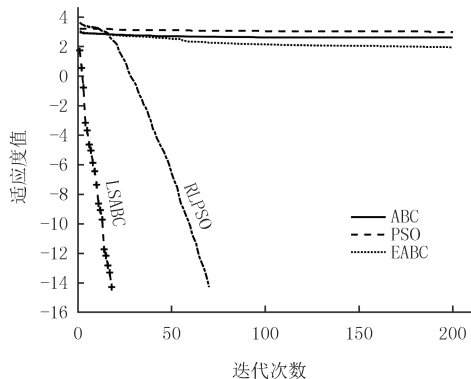


图5 函数5的进化曲线

Fig.5 The evolution curve of function 5

图 4 为 Rosenbrock 函数的收敛曲线,从图 4 中可以发现:算法的迭代早期阶段,进化曲线出现重叠交叉现象,曲线间的差别小;但随着迭代次数的增加,算法间的差距明显.产生该现象的原因是:Rosenbrock 函数是一个病态的螺旋型函数,算法的初期阶段侧重于勘探能力,导致算法陷入局部最优解.随着迭代次数的增

加,算法后期阶段侧重于开发能力,最优解的质量提高,对局部搜索行为指导作用明显.此外,从图4中还可以看出,禁忌搜索能力在种群的第150代时帮助算法跳脱局部最优解.故LSABC算法在跳脱局部最优解方面优于其他算法.

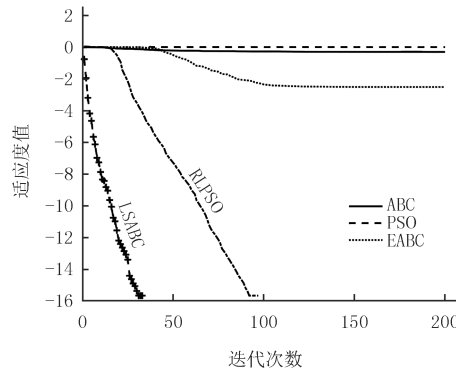


图6 函数6的进化曲线

Fig.6 The evolution curve of function 6

根据图5、图6和图8的函数进化曲线可以得出,利用LSABC算法在收敛速度、精度和全局优化能力方面明显优于其他算法,LSABC算法收敛速度快且易于跳出局部最优解.从函数进化曲线图中看出,LSABC算法求解多峰函数时,禁忌搜索能力分别在种群的第40代时帮助算法跳脱局部最优解.故LSABC算法在跳脱局部最优解方面明显优于ABC算法.

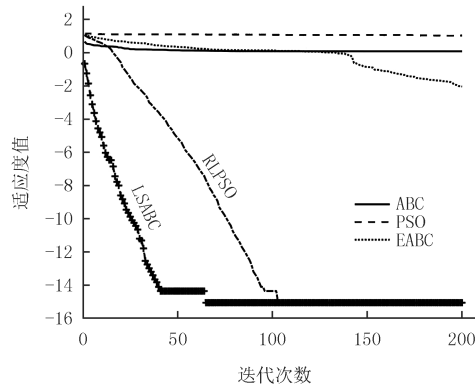


图7 函数7的进化曲线

Fig.7 The evolution curve of function 7

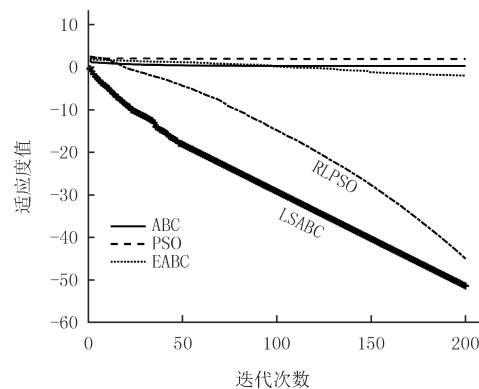


图8 函数8的进化曲线

Fig.8 The evolution curve of function 8

3 结 论

在解决复杂多维优化问题时,标准 ABC 算法在收敛速度、求解精度、全局优化能力和陷入局部最优解等方面仍然存在不足,通过对雇佣蜂阶段局部搜索能力和侦查蜂阶段的改进,得到以下结论。

(1)在雇佣蜂的搜索阶段,利用两个不同的搜索公式得到两组新解,并将适应度值最佳的个体作为候选解,增加了了解的多样性,使算法具有更好的指导雇佣蜂的搜索能力。

(2)在雇佣蜂阶段的局部搜索策略中加入个体双重认知能力和权重因子,平衡了算法的勘探和开发能力,从而加快了人工蜂群算法的收敛速度。

(3)在侦查蜂的搜索阶段,采用禁忌搜索策略,将局部极值放入禁忌表中,帮助算法跳脱局部最优解,有效减少了无效迭代的次数,避免算法的早熟。

实验结果表明,本文所提算法具有较强的全局收敛能力,保持了 ABC 算法运算的特点,显著优化了算法性能,有效减少了无效迭代的次数,达到避免算法早熟、加快算法收敛速度的目的。

参 考 文 献

- [1] TANG K S.Genetic algorithms and their applications[J].IEEE Signal Processing Magazine,1996,13(6):30-37.
- [2] DORIGO M.Ant colony optimization theory:A survey[J].Theoretical Computer Science,2005,344(2):243-278.
- [3] WANG Y.Crystal Structure Prediction via Particle Swarm Optimization[J].Physics,2010,82(9):7174-7182.
- [4] 杨咪.基于人工蜂群算法的 BP 双隐层神经网络水质模型[J].环境监测管理与技术,2018,30(1):21-26.
YANG M.Water Quality model of BP double hidden layer neural network based on artificial bee colony algorithm[J].Environmental Monitoring Management and Technology,2018,30(1):21-26.
- [5] 巢秀琴.人工蜂群算法优化的特征选择方法[J].计算机科学与探索,2019,13(2):124-133.
CHAO X Q.Feature selection optimized by artificial bee colony algorithm.[J].Computer Science and Exploration,2019,13(2):124-133.
- [6] 梁显丽.基于人工蜂群优化的多段支持度数据挖掘仿真[J].计算机仿真,2019(7):273-276.
LIANG X L.Multi-segment support data mining simulation based on artificial swarm optimization[J].computer simulation,2019(7):273-276.
- [7] 刘倩雯.人工蜂群算法及其在调度问题中的应用研究[D].北京:北京交通大学,2014.
LIU Q W.Research on artificial bee colony algorithm and its application in scheduling problems[D].Beijing:Beijing Jiaotong University,2014.
- [8] 刘蓓蕾.人工蜂群算法理论及其在信息处理中的应用研究[D].济南:山东大学,2016.
LIU B L.Research on artificial bee colony algorithm theories and its application in Information processing[D].Jinan:Shandong University,2016.
- [9] 胡本木.人工蜂群算法的改进及其在聚类分析中的应用研究[D].南京:南京师范大学,2014.
HU B M.Improvement of artificial bee colony algorithm and its application in cluster analysis[D].Nanjing:Nanjing Normal University,2014.
- [10] ZHU G.Gbest-guided artificial bee colony algorithm for numerical function optimization[J].Applied Mathematics and Computation,2010,217(7):3166-3173.
- [11] BANHARNSAKUN A.The best-so-far selection in Artificial Bee Colony algorithm[J].Applied Soft Computing,2011,11(2):2888-2901.
- [12] AKBARI R.A novel bee swarm optimization algorithm for numerical function optimization[J].Communications in Nonlinear Science and Numerical Simulation,2010,15(10):3142-3155.
- [13] GAO W F.Enhancing artificial bee colony algorithm using more information-based search equations[J].Information Sciences,2014,270:112-133.
- [14] KARANOGA D.A modified Artificial Bee Colony algorithm for real-parameter optimization[J].Information Sciences,2012,192(10):120-142.
- [15] LI G.Development and investigation of efficient artificial bee colony algorithm for numerical function optimization[J].Applied Soft Computing,2013,12(1):320-332.
- [16] GLOVER F.A Users Guide to Tabu Search[J].Annals of Operations Research,1993,41(1):1-28.
- [17] GAO W.Improved artificial bee colony algorithm for global optimization[J].Information Processing Letters,2011,111(17):871-882.
- [18] 赖兆林.基于逆向学习行为粒子群算法的云计算大规模任务调度[J].华东理工大学学报(自然科学版),2020,46(2):259-268.
LAI Z L.A reverse learning behavior Particle Swarm Optimization for large-scale task scheduling in cloud computing environment[J].

An Improved artificial bee colony algorithm for enhancing local search ability

Liu kun^a, Feng Shuo^b

(a. School of Science; b. School of Engineering Machinery, Chang'an University, Xi'an 710061, China)

Abstract: Aiming at the shortcomings of artificial bee colony algorithm in solving process, such as slow convergence rate and easy to fall into local optimal solution, this paper proposes an artificial bee colony algorithm based on enhanced local search ability(LSABC). On the one hand, in the employed bees search stage, two different search formulas are used to obtain two sets of solutions, and the best fitness value is used as a candidate solution to increase the diversity of the solutions. At the same time, the search formula is added with individual self-cognition ability(exploration ability)and social cognition ability(exploitation ability)to balance the exploration and exploitation ability of the algorithm, so as to speed up the convergence of the artificial bee colony algorithm. On the other hand, in the scout bees search stage, the Tabu search strategy is adopted and the local optimal solution is stored in the taboo table to help the algorithm to jump off the local optimal solution, so as to avoid the premature algorithm and accelerate the convergence speed of the algorithm. Because the improvement of LSABC algorithm is similar to particle swarm optimization algorithm, in order to verify the optimization performance of LSABC algorithm, the standard ABC algorithm, PSO algorithm, EABC algorithm, RLPSO algorithm and LSABC algorithm are selected for comparative test for eight classic benchmark functions. The experimental results show that LSABC algorithm can improve the accuracy and convergence speed, and is easy to jump out of the local optimal solution.

Keywords: artificial bee colony algorithm; local search capability; weight factor; Tabu search strategy

[责任编辑 陈留院 赵晓华]