

多搜索策略协同进化的人工蜂群算法

王志刚[†], 尚旭东, 夏慧明, 丁 华

(南京师范大学泰州学院 数学科学与应用学院, 江苏 泰州 225300)

摘 要: 针对人工蜂群算法存在的计算精度不高、收敛速度较慢的缺点, 提出一种多搜索策略协同进化的人工蜂群算法. 所提出的算法在引领蜂和跟随蜂进行邻域搜索时, 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, 以平衡算法的局部搜索能力和全局搜索能力. 14 个基准函数的仿真实验结果表明, 所提出的算法能有效改善寻优性能, 增强摆脱局部最优的能力. 与其他一些改进的人工蜂群算法相比, 具有较快的收敛速度和较高的求解精度.

关键词: 人工蜂群算法; 搜索策略; 协同进化

中图分类号: TP301.6 **文献标志码:** A

Artificial bee colony algorithm with multi-search strategy cooperative evolutionary

WANG Zhi-gang[†], SHANG Xu-dong, XIA Hui-ming, DING Hua

(School of Mathematics, Nanjing Normal University Taizhou College, Taizhou 225300, China)

Abstract: An artificial bee colony(ABC) algorithm with multi-search strategy cooperative evolutionary is presented in order to overcome the drawbacks of low computational accuracy and slow convergence of the artificial bee colony algorithm. In this algorithm, the dimensions of the search are dynamically adjusted when the employed bees and the onlooker bees search around the neighborhood to improve the search efficiency. The characteristics of different search strategies are combined to cooperative evolution so as to improve the local search ability and the global search ability. Experiments are conducted on a set of 14 benchmark functions, and the results demonstrate that the proposed algorithm can improve optimizing performance and avoid getting struck at local optima effectively. Compared with several other ABC-based algorithms, it has fast convergence and high accuracy.

Keywords: artificial bee colony algorithm; search strategy; cooperative evolution

0 引 言

人工蜂群^[1](ABC)算法是近年来由 Karaboga 提出的一种智能优化算法. 该算法控制参数少、易于实现, 在求解函数优化问题方面表现突出, 目前已成功应用于求解诸如神经网络^[2]、最小生成树^[3]、聚类分析^[4]和车辆路径问题^[5]等实际优化问题.

在 ABC 算法中, 引领蜂和跟随蜂都在食物源附近作一维邻域搜索, 这使得算法全局搜索能力较强, 但局部搜索能力不足、搜索效率较低. 针对算法存在的不足, 学者们提出了不同的改进方案^[6-15]. 例如, 文献[6]采用邻域搜索机制来改进人工蜂群算法的解搜索方程, 从当前食物源的环形邻域拓扑结构中选择

较优的邻居食物源进行开采; 文献[7]通过全局引导搜索策略改进了原有的搜索策略, 提高了算法的求解精度; 文献[8]集成了不同特征的搜索策略, 根据优化问题的类型选择相应的搜索策略; 文献[9]利用混沌序列搜索具有较好的随机性和遍历性的特点, 提出了一种具有混沌搜索策略的蜂群优化算法; 文献[10]采用“分段搜索”的方式更新了食物源, 简化了食物源的选择方式, 提高了搜寻过程的寻优效率; 文献[11]利用符号函数对不同的搜索策略进行了融合, 在进化过程中发挥不同搜索策略的优势; 文献[12]受差分进化算法的启发, 提出了 ABC/best 算法; 文献[13]采用当前全局最优解代替随机选取的邻域个体, 并依据当前

收稿日期: 2016-12-15; 修回日期: 2017-02-21.

基金项目: 国家自然科学基金项目(11601234); 江苏省自然科学基金项目(BK20160571); 江苏省青蓝工程基金项目.

作者简介: 王志刚(1978—), 男, 副教授, 硕士, 从事组合优化与智能优化算法等研究; 尚旭东(1982—), 男, 副教授, 博士, 从事智能算法、变分法等研究.

[†]通讯作者. E-mail: wzg19.scut@163.com

全局最优解的适应度来调整邻域搜索步长;文献[14]利用全局最优解、惯性权重和加速系数来提高算法的收敛性能,并提出了具有预测和选择能力的混合ABC算法;文献[15]受差分进化算法启发,在全局最优引导人工蜂群算法的基础上,提出了一种受启发的人工蜂群算法。

本文在前人研究工作的基础上,提出一种多搜索策略协同进化的人工蜂群算法(ABCMSSE)。通过动态调整引领蜂和跟随蜂的搜索维数来提高搜索效率,并结合不同搜索策略的特点,采用协同进化的方式平衡算法的局部搜索能力和全局搜索能力。仿真实验的结果表明,所提出的算法能较快发现最优解,优化性能有较为明显的提高。

1 人工蜂群算法

ABC算法模拟了蜂群个体在觅食过程中的分工以及个体间的信息共享机制,蜂群按照分工的不同可以分为引领蜂、跟随蜂和侦察蜂。引领蜂在食物源附近进行邻域搜索,并将食物源的信息反馈给跟随蜂,跟随蜂根据收到的信息,挑选较优的食物源继续开采。若经过一段时间开采后,某个食物源质量没有进一步提高,则对应的引领蜂就转变为侦察蜂,并重新选择新食物源进行开采。求解优化问题时,每个食物源对应优化问题的一个解,目标函数的适应值决定食物源的优劣,适应值越高,食物源越优。 $x_i = (x_{i1}, x_{i2}, \dots, x_{iD})$ 表示第*i*个食物源($i = 1, 2, \dots, SN$),解的个数(SN)等于蜂群中引领蜂或跟随蜂的个数,*D*表示搜索空间维数。

引领蜂在食物源附近进行一维邻域搜索,生成新食物源,搜索策略如下:

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

其中: v_{ij} 是新生成的食物源; $k \in \{1, 2, \dots, SN\}$ 和 $j \in \{1, 2, \dots, D\}$ 是随机选取的,但 $k \neq i$; φ_{ij} 是 $[-1, 1]$ 上均匀分布的随机数。

引领蜂在食物源更新后会通过贪婪选择策略确定食物源,并向跟随蜂传递食物源信息,跟随蜂则根据传递的信息通过如下概率来选择食物源:

$$P_i = \frac{\text{fit}_i}{\sum_{n=1}^{SN} \text{fit}_n}. \quad (2)$$

其中: P_i 是第*i*个食物源的选择概率, fit_i 是第*i*个食物源的适应值。 fit_i 计算公式为

$$\text{fit}_i = \begin{cases} \frac{1}{1 + f_i}, & f_i \geq 0; \\ 1 + |f_i|, & f_i < 0. \end{cases} \quad (3)$$

其中 f_i 是第*i*个食物源的目标函数值。可见,食物源

的适应值越高,它被跟随蜂选中的概率就越大。选择完食物源之后,跟随蜂在食物源附近根据式(1)进行搜索,并通过贪婪选择策略确定食物源。

若某个食物源经过 $limit$ 次评估后质量仍未提高,则该食物源处的引领蜂转变为侦察蜂,并按式(4)在搜索空间内重新生成新食物源

$$x_{ij} = L_j + \text{rand}(U_j - L_j). \quad (4)$$

其中: x_{ij} 是新食物源的第*j*维分量, rand 是 $(0, 1)$ 之间均匀分布的随机数, L_j 、 U_j 分别是第*j*维分量的下界和上界。

2 多搜索策略协同进化的人工蜂群算法

在式(1)中,引领蜂和跟随蜂只在 x_{ij} 附近作一维邻域搜索,这种方式容易导致算法的搜索效率不高。为提高搜索效率,文献[16]引入参数MR来改变搜索的维数,在搜索时,对每一维均作一次判断,即先产生一个 $[0, 1]$ 之间的随机数 R_{ij} ,然后将 R_{ij} 与MR进行比较。若 $R_{ij} < MR$,则对应的这一维就进行邻域搜索,否则不进行。可见,MR越大,引领蜂和跟随蜂进行邻域搜索的维数就越多,越有利于局部搜索和提高收敛速度;MR越小,越有利于全局搜索。本文也采用这种方式来提高算法的搜索效率,不同之处在于对MR进化过程中作动态调整,使其由小到大变化,以便算法在进化初期有较好的全局搜索能力,在进化后期有较好的局部搜索能力。MR的取法如下:

$$MR = \frac{0.3FE}{\text{Max FEs}} + 0.1. \quad (5)$$

其中:FE是评价次数,Max FEs是目标函数的最大评价次数。

从式(1)还可以看出,引领蜂和跟随蜂仅在 x_{ij} 附近产生新食物源,这种方式使得算法局部搜索能力不足、种群多样性较差,导致算法在求解优化问题时存在收敛速度慢、求解精度低的问题。为此,专家们提出了许多改进的搜索策略,其中常见的搜索策略有

$$v_{ij} = x_{rj} + \varphi_{ij}(x_{rj} - x_{kj}), \quad (6)$$

$$v_{ij} = x_{rj} + \varphi_{ij}(x_{ij} - x_{kj}), \quad (7)$$

$$v_{ij} = x_{\text{best},j} + \varphi_{ij}(x_{rj} - x_{kj}), \quad (8)$$

$$v_{ij} = x_{\text{best},j} + \varphi_{ij}(x_{ij} - x_{kj}). \quad (9)$$

其中: $r, k \in \{1, 2, \dots, SN\}$, $r \neq k \neq i$; $j \in \{1, 2, \dots, D\}$; x_{best} 是种群的全局最优个体。式(6)^[17]和(7)^[17]在随机选择的个体附近产生新食物源,它们的全局搜索能力较强,可较好地保持种群多样性,但收敛速度较慢;式(8)^[18]和(9)^[19]中新食物源在全局最优个体附近产生,它们的局部搜索能力较强,但容易陷入局部最优。

这些搜索策略与ABC算法所采用的搜索策略既有各自的优势,也有各自的不足,不能笼统地说哪种

搜索策略绝对地好或差.同时,各搜索策略既存在结构和进化方式上的共性特征,又具有性能差异的特点,这种共性与差异使得它们可以协同进化.

借鉴政府部门为抑制物价非正常上涨往往同时采用临时价格干预、出台补贴政策、制定相关法规以及加强监管力度等多种措施和方法的社会经验^[20],提出一种多搜索策略协同进化的人工蜂群算法.在算法中,同时使用ABC算法所采用的搜索策略和上述4种改进的搜索策略.在每次邻域搜索时,总是依次从5种搜索策略中选取一种用于引领蜂和跟随蜂生成新食物源,因此新食物源必然是在5种搜索策略共同作用下产生的,即将式(1)中的 v_{ij} 定义如下:

$$v_{ij} = \begin{cases} x_{ij} + \varphi_{ij}(x_{ij} - x_{kj}), & M = 1; \\ x_{rj} + \varphi_{ij}(x_{rj} - x_{kj}), & M = 2; \\ x_{rj} + \varphi_{ij}(x_{ij} - x_{kj}), & M = 3; \\ x_{best,j} + \varphi_{ij}(x_{rj} - x_{kj}), & M = 4; \\ x_{best,j} + \varphi_{ij}(x_{ij} - x_{kj}), & M = 5. \end{cases} \quad (10)$$

其中: $M = i(\text{mod}5) + 1, M \in \{1, 2, 3, 4, 5\}$.

新算法的具体步骤如下.

- Step 1: 设置相关参数,生成初始种群;
- Step 2: 计算引领蜂对应食物源的目标函数值;
- Step 3: 对于每个引领蜂,根据式(5)和(10)生成新

食物源,计算其目标函数值,并通过贪婪选择的方式选取较优的食物源;

Step 4: 计算更新后食物源的目标函数值,并按式(2)计算选择概率 P_i ;

Step 5: 跟随蜂按概率 P_i 选择食物源,根据式(5)和(10)生成新食物源,计算其目标函数值,并通过贪婪选择的方式选取较优的食物源;

Step 6: 若引领蜂对应的食物源连续 $limit$ 次没有改进,则对应的引领蜂变为侦察蜂,根据式(4)产生新食物源;

Step 7: 记录全局最优值,并跳转至Step 3,直至算法满足结束条件.

3 仿真实验

为分析ABCMSSCE算法性能,使用14个经典测试函数对ABC算法、全局最优引导的人工蜂群(GABC)算法^[7]与ABCMSSCE算法进行比较.实验时, $SN = 20, limit = SN \times D$,目标函数的最大评价次数 $MaxFEs = 5000 \times D$.最优值、最差值、平均值和标准差分别为3种算法独立实验30次的结果.14个测试函数如表1所示.在函数类型中,U表示单峰函数,M表示多峰函数,S表示可分函数,N表示不可分函数.3种算法在 $D = 60$ 和 $D = 100$ 时的仿真结果如表2和表3所示.

表1 测试函数

名称	表达式	类型	搜索范围	最优值
Sphere	$f_1(x) = \sum_{i=1}^D x_i^2$	US	[-100,100]	0
Schwefel 2.22	$f_2(x) = \sum_{i=1}^D x_i + \prod_{i=1}^D x_i $	UN	[-10,10]	0
Schwefel 2.21	$f_3(x) = \max_i \{ x_i , 1 \leq i \leq D\}$	UN	[-100,100]	0
Step	$f_4(x) = \sum_{i=1}^D (x_i + 0.5)^2$	US	[-100,100]	0
Quartic	$f_5(x) = \sum_{i=1}^D ix_i^4 + \text{random}[0, 1)$	US	[-1.28,1.28]	0
SumSquares	$f_6(x) = \sum_{i=1}^D ix_i^2$	US	[-10,10]	0
Rastrigin	$f_7(x) = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10)$	MS	[-5.12,5.12]	0
Griewank	$f_8(x) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	MN	[-600,600]	0
Ackley	$f_9(x) = 20 + e - 20 \exp\left\{-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}\right\} - \exp\left\{\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i)\right\}$	MN	[-32,32]	0
Penalized1	$f_{10}(x) = \frac{\pi}{D} \left\{ 10 \sin^2(\pi y_1) + \sum_{i=1}^{D-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_D - 1)^2 \right\} + \sum_{i=1}^D u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{1}{4}(x_i + 1), u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a; \\ 0, & -a \leq x_i \leq a; \\ k(-x_i - a)^m, & x_i < -a. \end{cases}$	MN	[-50,50]	0

续表

名称	表达式	类型	搜索范围	最优值
Penalized2	$f_{11}(x) = \frac{1}{10} \left\{ \sin^2(3\pi x_1) + \sum_{i=1}^{D-1} (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})] + (x_D - 1)^2 [1 + \sin^2(2\pi x_D)] \right\} + \sum_{i=1}^D u(x_i, 5, 100, 4)$	MN	[-50,50]	0
Schaffer	$f_{12}(x) = 0.5 + \frac{\sin^2 \left(\sqrt{\sum_{i=1}^D x_i^2} \right) - 0.5}{\left(1 + 0.001 \sum_{i=1}^D x_i^2 \right)^2}$	MN	[-100,100]	0
Shifted sphere	$f_{13}(x) = \sum_{i=1}^D z_i^2, z = x - o$	US	[-100,100]	0
Shifted ackley	$f_{14}(x) = 20 + e - 20 \exp \left\{ -0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D z_i^2} \right\} - \exp \left\{ \frac{1}{D} \sum_{i=1}^D \cos(2\pi z_i) \right\}, z = x - o$	MN	[-32,32]	0

表2 ABC、GABC与ABCMSSCE在 D = 60时的优化性能比较

函数	算法	最优值	最差值	平均值	标准差
f ₁	ABC	9.65e-16	1.44e-15	1.27e-15	1.30e-16
	GABC	7.15e-16	1.16e-15	9.86e-16	1.02e-16
	ABCMSSCE	2.32e-114	2.36e-110	2.98e-111	6.38e-111
f ₂	ABC	2.29e-15	3.39e-15	2.82e-15	2.73e-16
	GABC	2.26e-15	2.95e-15	2.56e-15	1.77e-16
	ABCMSSCE	1.40e-73	6.97e-72	1.14e-72	1.37e-72
f ₃	ABC	4.73e+00	1.61e+01	1.09e+01	2.68e+00
	GABC	2.59e+00	4.72e+00	3.45e+00	4.83e-01
	ABCMSSCE	5.75e-03	2.96e-02	1.53e-02	6.73e-03
f ₄	ABC	0	0	0	0
	GABC	0	0	0	0
	ABCMSSCE	0	0	0	0
f ₅	ABC	5.41e-02	1.30e-01	9.48e-02	2.02e-02
	GABC	3.22e-02	7.10e-02	5.12e-02	1.11e-02
	ABCMSSCE	3.86e-03	1.35e-02	9.25e-03	2.60e-03
f ₆	ABC	9.75e-16	1.41e-15	1.17e-15	1.31e-16
	GABC	7.44e-16	1.20e-15	1.01e-15	1.04e-16
	ABCMSSCE	6.21e-116	1.76e-111	1.95e-112	3.58e-112
f ₇	ABC	0	0	0	0
	GABC	0	0	0	0
	ABCMSSCE	0	6.96e+00	2.99e+00	1.83e+00
f ₈	ABC	0	6.46e-11	2.16e-12	1.16e-11
	GABC	0	1.11e-16	5.92e-17	5.54e-17
	ABCMSSCE	0	0	0	0
f ₉	ABC	6.45e-14	9.30e-14	7.95e-14	6.70e-15
	GABC	6.10e-14	8.23e-14	7.02e-14	4.55e-15
	ABCMSSCE	7.69e-15	1.48e-14	1.14e-14	1.12e-15
f ₁₀	ABC	9.65e-16	1.43e-15	1.19e-15	1.12e-16
	GABC	6.91e-16	1.20e-15	9.86e-16	1.05e-16
	ABCMSSCE	7.85e-33	7.85e-33	7.85e-33	1.37e-48
f ₁₁	ABC	7.32e-16	1.43e-15	1.21e-15	1.66e-16
	GABC	7.50e-16	1.18e-15	9.55e-16	1.27e-16
	ABCMSSCE	1.35e-32	1.35e-32	1.35e-32	5.47e-48
f ₁₂	ABC	4.42e-01	4.89e-01	4.76e-01	1.09e-02
	GABC	4.15e-01	4.80e-01	4.62e-01	1.47e-02
	ABCMSSCE	7.82e-02	1.27e-01	1.03e-01	2.44e-02
f ₁₃	ABC	1.13e-15	1.44e-15	1.30e-15	1.11e-16
	GABC	6.23e-16	1.17e-15	9.40e-16	1.41e-16
	ABCMSSCE	0	0	0	0
f ₁₄	ABC	6.81e-14	9.30e-14	8.16e-14	6.18e-15
	GABC	5.74e-14	8.23e-14	7.14e-14	5.26e-15
	ABCMSSCE	7.69e-15	1.48e-14	1.09e-14	1.41e-15

表3 ABC、GABC与ABCMSSCE在 D = 100时的优化性能比较

函数	算法	最优值	最差值	平均值	标准差
f ₁	ABC	2.02e-15	2.55e-15	2.28e-15	1.70e-16
	GABC	1.59e-15	2.08e-15	1.80e-15	1.51e-16
	ABCMSSCE	1.73e-108	1.84e-105	3.36e-106	4.99e-106
f ₂	ABC	4.33e-15	5.39e-15	4.96e-15	2.27e-16
	GABC	3.62e-15	4.96e-15	4.47e-15	2.89e-16
	ABCMSSCE	1.43e-72	1.94e-70	2.98e-71	3.77e-71
f ₃	ABC	2.39e+01	3.40e+01	2.85e+01	2.50e+00
	GABC	1.10e+01	1.64e+01	1.36e+01	1.22e+00
	ABCMSSCE	1.06e+00	3.87e+00	2.16e+00	6.35e-01
f ₄	ABC	0	0	0	0
	GABC	0	0	0	0
	ABCMSSCE	0	0	0	0
f ₅	ABC	1.32e-01	2.19e-01	1.66e-01	2.07e-02
	GABC	7.21e-02	9.74e-02	8.56e-02	6.30e-03
	ABCMSSCE	1.13e-02	2.26e-02	1.59e-02	2.78e-03
f ₆	ABC	1.87e-15	2.75e-15	2.27e-15	2.09e-16
	GABC	1.21e-15	2.09e-15	1.78e-15	2.10e-16
	ABCMSSCE	2.20e-109	2.10e-105	3.96e-106	5.98e-106
f ₇	ABC	0	0	0	0
	GABC	0	0	0	0
	ABCMSSCE	1.09e+01	3.48e+01	2.20e+01	5.57e+00
f ₈	ABC	0	9.99e-16	3.77e-16	4.13e-16
	GABC	0	4.44e-16	1.04e-16	1.03e-16
	ABCMSSCE	0	0	0	0
f ₉	ABC	1.32e-13	1.64e-13	1.45e-13	7.79e-15
	GABC	1.11e-13	1.43e-13	1.28e-13	6.23e-15
	ABCMSSCE	1.48e-14	1.84e-14	1.55e-14	1.42e-15
f ₁₀	ABC	1.85e-15	2.51e-15	2.24e-15	1.86e-16
	GABC	1.44e-15	2.09e-15	1.80e-15	1.78e-16
	ABCMSSCE	4.71e-33	4.71e-33	4.71e-33	2.74e-48
f ₁₁	ABC	1.89e-15	2.54e-15	2.25e-15	2.18e-16
	GABC	1.38e-15	2.09e-15	1.78e-15	2.17e-16
	ABCMSSCE	1.35e-32	1.35e-32	1.35e-32	5.47e-48
f ₁₂	ABC	4.93e-01	4.98e-01	4.97e-01	1.32e-03
	GABC	4.90e-01	4.98e-01	4.96e-01	1.44e-03
	ABCMSSCE	1.27e-01	2.73e-01	1.68e-01	3.72e-02
f ₁₃	ABC	1.88e-15	2.69e-15	2.24e-15	1.76e-16
	GABC	1.44e-15	2.09e-15	1.77e-15	1.60e-16
	ABCMSSCE	0	0	0	0
f ₁₄	ABC	1.18e-13	1.57e-13	1.40e-13	9.13e-15
	GABC	1.21e-13	1.43e-13	1.30e-13	6.39e-15
	ABCMSSCE	1.48e-14	1.84e-14	1.68e-14	1.76e-15

由表2可看出:单峰函数仿真实验中,ABC算法、GABC算法和ABCMSSE算法都可以求出 $f_4(x)$ 的理论最优值;对于 $f_{13}(x)$,ABCMSSE算法求解结果优于ABC算法和GABC算法,可以求得理论最优值;而对于其他单峰函数,3种算法都未能求得理论最优值,但ABCMSSE算法在求解精度方面和稳定性方面明显优于ABC算法和GABC算法。

对于多峰函数,ABCMSSE算法在 $f_7(x)$ 上的求解结果不是很理想,未能找到理论最优值.事实上,引入参数MR来改变搜索的维数这种方式类似于差分进化算法中的交叉操作,而差分进化算法本身对函数 $f_7(x)$ 的求解效果同样不是很理想.而只进行一维搜索的ABC算法和GABC算法对 $f_7(x)$ 却求解效果不错,说明这个函数对搜索维数的改变比较敏感,在维数较高的情形下出现了不收敛的情况.对于 $f_8(x)$,ABC算法、GABC算法都没有求得其理论最优值,而ABCMSSE算法却可以求得理论最优值;对于 $f_9(x) \sim f_{12}(x)$ 和 $f_{14}(x)$,ABCMSSE算法的求解结果也明显优于ABC算法和GABC算法。

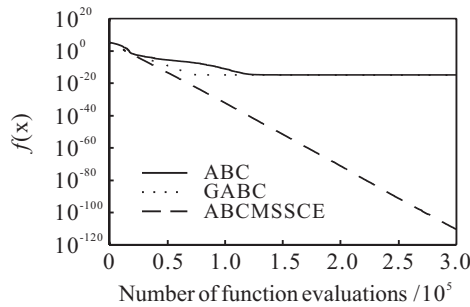


图1 Sphere函数的收敛曲线

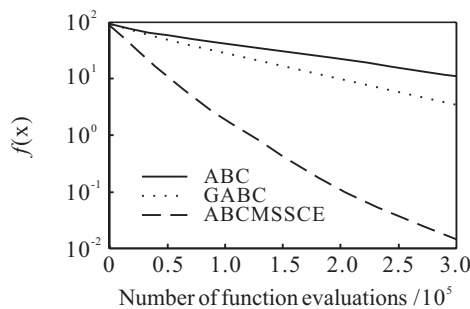


图2 Schwetel 2.21函数的收敛曲线

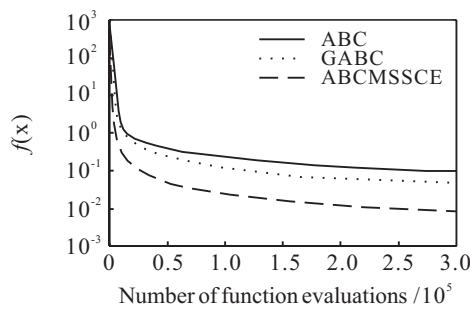


图3 Quartic函数的收敛曲线

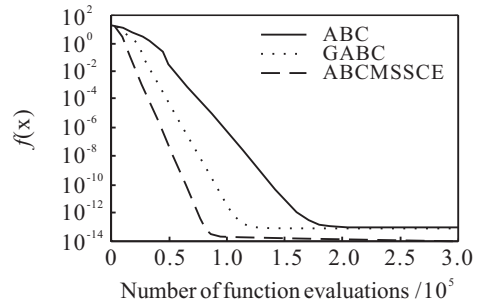


图4 Ackley函数的收敛曲线

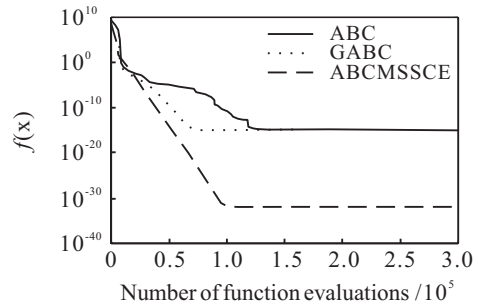


图5 Penalized2函数的收敛曲线

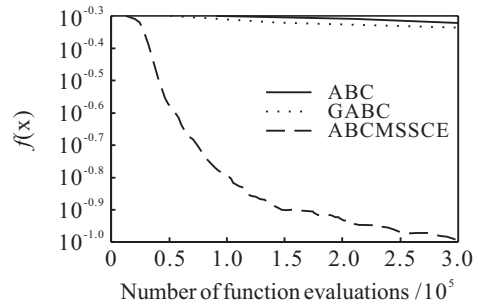


图6 Schaffer函数的收敛曲线

图1~图6给出了部分测试函数在 $D = 60$ 时的收敛曲线.从图中可以看出,对于单峰函数Sphere和Schwefel2.21,ABCMSSE算法的收敛速度很快,几乎是以线性递减的方式收敛到最优解,对于Quartic函数,ABCMSSE算法虽不能以线性递减方式收敛,但收敛速度依然比ABC算法和GABC算法快,且能收敛到较高精度的解.多峰函数Ackley、Penalized2和Schaffer均为复杂的非线性全局优化问题,算法在求解他们时很容易陷入局部最优,由于ABCMSSE算法对搜索维数的动态调整 and 不同搜索策略的协同进化,使算法可以尽可能跳出局部最优,提高了算法的求解精度。

由表3可以看出:在 $D = 100$ 时,ABCMSSE算法对 $f_7(x)$ 的求解结果依旧不很理想;对于其他函数,ABCMSSE算法都能取得较好的优化结果,求解精度均优于ABC算法和GABC算法。

为了进一步展示ABCMSSE算法的先进性,将其与ABCBest1^[12]、ABCBest2^[12]、ABCVSS^[8]等较新的算法在 $D = 60$ 和 $D = 100$ 时进行仿真实验,实验

表4 $D = 60$ 时的优化性能比较

函数	ABCBest1		ABCBest2		ABCVSS		ABCMSSCE	
	平均值	标准差	平均值	标准差	平均值	标准差	平均值	标准差
f_1	3.92e-44	2.64e-44	4.82e-33	2.59e-33	1.09e-83	5.01e-83	2.98e-111	6.38e-111
f_2	8.48e-24	2.31e-24	1.58e-17	3.32e-18	1.47e-45	7.00e-45	1.14e-72	1.37e-72
f_3	2.10e+01	1.68e+00	2.40e+01	2.16e+00	1.68e+00	4.05e-01	1.53e-02	6.73e-03
f_4	0	0	0	0	0	0	0	0
f_5	6.11e-02	8.89e-03	6.79e-02	9.38e-03	4.35e-02	7.69e-03	9.25e-03	2.60e-03
f_6	2.06e-44	1.83e-44	9.10e-34	3.87e-34	8.17e-86	4.47e-85	1.95e-112	3.58e-112
f_7	0	0	0	0	0	0	2.99e+00	1.83e+00
f_8	0	0	3.96e-09	2.04e-08	0	0	0	0
f_9	6.93e-14	5.00e-15	7.47e-14	4.12e-15	5.93e-14	7.65e-15	1.14e-14	1.12e-15
f_{10}	7.85e-33	2.78e-48	7.85e-33	2.78e-48	7.85e-33	2.78e-48	7.85e-33	1.37e-48
f_{11}	1.35e-32	5.57e-48	1.35e-32	5.57e-48	1.35e-32	5.57e-48	1.35e-32	5.47e-48
f_{12}	4.61e-01	1.15e-02	4.68e-01	9.17e-03	4.72e-01	1.42e-02	1.03e-01	2.44e-02
f_{13}	0	0	0	0	0	0	0	0
f_{14}	6.90e-14	4.82e-15	7.39e-14	3.54e-15	5.97e-14	8.52e-15	1.09e-14	1.41e-15

表5 $D = 100$ 时的优化性能比较

函数	ABCBest1		ABCBest2		ABCVSS		ABCMSSCE	
	平均值	标准差	平均值	标准差	平均值	标准差	平均值	标准差
f_1	1.54e-42	8.93e-43	5.09e-32	2.03e-32	1.01e-83	5.52e-83	3.36e-106	4.99e-106
f_2	6.27e-23	1.09e-23	7.21e-17	1.36e-17	3.94e-40	2.12e-39	2.98e-71	3.77e-71
f_3	4.72e+01	2.29e+00	5.06e+01	2.67e+00	7.91e+00	1.32e+00	2.16e+00	6.35e-01
f_4	0	0	0	0	0	0	0	0
f_5	1.30e-01	1.12e-02	1.42e-01	1.58e-02	7.87e-02	1.16e-02	1.59e-02	2.78e-03
f_6	8.94e-43	7.34e-43	2.15e-32	1.24e-32	5.31e-85	2.68e-84	3.96e-106	5.98e-106
f_7	0	0	0	0	0	0	2.20e+01	5.57e+00
f_8	0	0	2.94e-10	1.16e-09	0	0	0	0
f_9	1.27e-13	7.15e-15	1.39e-13	5.75e-15	1.11e-13	9.93e-15	1.55e-14	1.42e-15
f_{10}	4.71e-33	1.39e-48	4.71e-33	1.39e-48	4.71e-33	1.39e-48	4.71e-33	2.74e-48
f_{11}	1.35e-32	5.57e-48	2.18e-32	6.62e-33	1.35e-32	5.57e-48	1.35e-32	5.47e-48
f_{12}	4.96e-01	8.28e-04	4.97e-01	6.58e-04	4.98e-01	8.15e-04	1.68e-01	3.72e-02
f_{13}	0	0	0	0	0	0	0	0
f_{14}	1.26e-13	8.48e-15	1.38e-13	6.90e-15	1.15e-13	1.08e-14	1.68e-14	1.76e-15

时目标函数的最大评价次数均为 $\text{Max FEs} = 5000 \times D$ 。算法的实验结果如表4和表5所示。

由表4和表5可以看出:4种算法都能求得 $f_4(x)$ 和 $f_{13}(x)$ 的理论最优值;对于 $f_7(x)$, ABCMSSCE 求解结果最差,出现了不收敛的现象,而其余3种算法都可以求得该函数的理论最优值;对于其他函数, ABCMSSCE 算法在求解的精度和稳定性上都优于 ABCBest1 和 ABCBest2;除了在 $f_{10}(x)$ 和 $f_{11}(x)$ 上与 ABCVSS 算法求解结果相差无几外, ABCMSSCE 算法在其他函数上的求解结果也都明显优于 ABCVSS 算法。

从上述实验结果可以看出, ABCMSSCE 算法通过动态调整搜索维数和协同进化的方式能够提高算法的搜索效率,克服 ABC 算法收敛速度慢和易陷入局部最优的缺陷。

4 结论

本文针对 ABC 算法传统搜索策略存在的不足,提出了一种多搜索策略协同进化的人工蜂群算法。所提出的算法为提高搜索效率,引入参数动态调整引领蜂和跟随蜂的搜索维数,并结合不同搜索策略的特点,采用协同进化的方式平衡算法的局部搜索能力和全局搜索能力。基准测试函数的仿真实验结果表明,所提出的算法寻优效果好,求解精度高。当然,本文算法也存在不足之处,参数 MR 的选取更多是建立在人工经验的基础上,且算法在每次迭代时,都需要产生一个随机数与参数 MR 进行比较,这在一定程度上增加了算法的计算时间,因此对参数 MR 的选取进行研究以及将算法应用于约束优化和图像处理等方面将是下一步的工作。

参考文献(References)

- [1] Karaboga D, Basturk B. A powerful and efficient algorithm for numerical function optimization: artificial bee colony(ABC) algorithm[J]. *J of Global Optimization*, 2007, 39(3): 459-471.
- [2] Yeh W, Hsieh T. Artificial bee colony algorithm-neural networks for S-system models of biochemical networks approximation[J]. *Neural Computing and Applications*, 2012, 21(2): 365-375.
- [3] Singh A. An artificial bee colony algorithm for the leaf-constrained minimum spanning tree problem[J]. *Applied Soft Computing*, 2009, 9(2): 625-631.
- [4] Karaboga D. A novel clustering approach: Artificial bee colony(ABC) algorithm[J]. *Applied Soft Computing*, 2011, 11(1): 652-657.
- [5] Szeto W Y, Wu Y, Ho S C. An artificial bee colony algorithms for the capacitated vehicle routing problem[J]. *European J of Operational Research*, 2011, 215(1): 126-135.
- [6] 周新宇, 吴志健, 邓长寿, 等. 一种邻域搜索的人工蜂群算法[J]. *中南大学学报: 自然科学版*, 2015, 46(2): 534-546.
(Zhou X Y, Wu Z J, Deng C S, et al. Neighborhood search-based artificial bee colony algorithm[J]. *J of Central South University: Science and Technology*, 2015, 46(2): 534-546.)
- [7] 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.
- [8] Kiran M S, Hakli H, Gunduz M, et al. Artificial bee colony algorithm with variable search strategy for continuous optimization[J]. *Information Sciences*, 2015, 300(1): 140-157.
- [9] 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.
(Luo J, Li Y. Artificial bee colony algorithm with chaotic-search strategy[J]. *Control and Decision*, 2010, 25(12): 1913-1916.)
- [10] 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): 1402-1405.
(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): 1402-1405.)
- [11] 王志刚, 王明刚. 基于符号函数的多搜索策略人工蜂群算法[J]. *控制与决策*, 2016, 31(11): 2037-2044.
(Wang Z G, Wang M G. Multi-search strategy of artificial bee colony algorithm based on symbolic function[J]. *Control and Decision*, 2016, 31(11): 2037-2044.)
- [12] Gao W F, Liu S Y, Huang L L. A global best artificial bee colony algorithm for global optimization[J]. *J of Computational and Applied Mathematics*, 2012, 236(11): 2741-2753.
- [13] Banharnsakun A, Achalakul T, Sirinaovakul B. The best-so-far selection in artificial bee colony algorithm[J]. *Applied Soft Computing*, 2011, 11(2): 2888-2901.
- [14] Li G Q, Niu P F, Xiao X J. Development and investigation of efficient artificial bee colony algorithm for numerical function optimization[J]. *Applied Soft Computing*, 2012, 12(1): 320-332.
- [15] 高卫峰, 刘三阳, 黄玲玲. 受启发的人工蜂群算法在全局优化问题中的应用[J]. *电子学报*, 2012, 40(12): 2396-2403.
(Gao W F, Liu S Y, Huang L L. Inspired artificial bee colony algorithm for global optimization problems[J]. *Acta Electronica Sinica*, 2012, 40(12): 2396-2403.)
- [16] Akay B, Karaboga D. A modified artificial bee colony algorithm for real-parameter optimization[J]. *Information Sciences*, 2012, 192(6): 120-142.
- [17] 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 Trans on Cybernetics*, 2013, 43(3): 1011-1024.
- [18] Gao W F, Liu S Y. A modified artificial bee colony algorithm[J]. *Computer & Operations Research*, 2012, 39(3): 687-697.
- [19] Gao W F, Liu S Y. Improved artificial bee colony algorithm for global optimization[J]. *Information Processing Letters*, 2011, 111(17): 687-697.
- [20] 贺毅朝, 王熙熙, 刘坤起, 等. 差分演化的收敛性分析与算法改进[J]. *软件学报*, 2010, 21(5): 875-885.
(He Y C, Wang X X, Liu K Q, et al. Convergent analysis and algorithmic improvement of differential evolution[J]. *J of Software*, 2010, 21(5): 875-885.)

(责任编辑: 闫妍)