

混合人工蜂群算法

高卫峰, 刘三阳, 姜 飞, 张建科

(西安电子科技大学应用数学系, 陕西 西安 710071)

摘 要: 针对人工蜂群算法收敛速度慢、容易出现“早熟”的缺点, 提出了一种混合的人工蜂群算法 (hybrid artificial bee colony, HABC)。在人工蜂群算法的迭代中引入淘汰规则和新的搜索策略, 以提高算法的收敛速度; 同时, 为了维护群体的多样性, 对种群中的个体采用差分进化。通过对一个调频 (frequency-modulated, FM) 合成器参数优化问题测试, 表明该算法能够有效地克服“早熟”现象, 提高了全局寻优的能力。将其应用于线性系统逼近问题, 仿真实验表明该算法是快速有效的。

关键词: 人工蜂群; 差分进化; 线性系统逼近; 参数优化

中图分类号: TP 301

文献标志码: A

DOI: 10.3969/j.issn.1001-506X.2011.05.41

Hybrid artificial bee colony algorithm

GAO Wei-feng, LIU San-yang, JIANG Fei, ZHANG Jian-ke

(Department of Applied Mathematics, Xidian University, Xi'an 710071, China)

Abstract: In order to overcome prematurity and low searching speed of the artificial bee colony (ABC), a hybrid artificial bee colony (HABC) algorithm is proposed. An eliminative rule and the new search strategy is introduced into the iteration of ABC to improve the convergence rate. Then, to maintain the population diversity, differential evolution (DE) simulates evolution and all individuals are taken into account in each generation. One experiment of parameter optimization of frequency-modulated (FM) synthesis indicates that the proposed algorithm can avoid prematurity effectively, and the algorithm possesses better ability in finding global optimum than compared algorithms. The proposed algorithm can be used to solve linear system approximation problems, and results show that the algorithm is fast and effective, and greatly outperforms other algorithms.

Keywords: artificial bee colony; differential evolution; approximation of linear system; parameter optimization

0 引 言

人工蜂群 (artificial bee colony, ABC) 算法和差分进化 (differential evolution, DE) 算法都是基于群体智能理论的优化算法。ABC 算法是文献[1]于 2005 年提出的一种群集智能随机优化算法, 模拟蜜蜂群的智能采蜜行为。蜜蜂根据各自的分工进行不同的活动, 并实现蜂群信息的共享和交流, 从而找到问题的最优解。它已经在函数优化^[2-4]、组合优化^[5]和工程领域^[6-9]得到了大量的应用, 但其存在着收敛速度慢、容易出现“早熟”的缺点。文献[10-11]将混沌思想引入 ABC 算法中, 利用混沌运动的随机性、遍历性等特点提高算法的全局搜索能力。文献[12]提出 DE 算法, 其原理是从当前种群中提取差分信息并指导进一步搜索, 不需要借助问题的特征信息, 具有较强的全局收敛能力和鲁棒性。DE 算法在维护群体的多样性及搜索能力方面功能较强, 但 DE 算法也存在不足, 如对于求解复杂优化问题时存在局部收敛, 且后期收敛速度较慢, 表现不够稳健等问题^[13]。

本文在改进 ABC 算法的基础上, 综合 DE 算法, 提出一种混合人工蜂群 (hybrid artificial bee colony, HABC) 算法。参数优化和典型的线性系统逼近工程优化问题的实验都表明, HABC 算法是非常有效的, 具有较高的实用价值和应用潜力。

1 改进的 ABC 算法

ABC 算法^[1-4]是基于种群的并通过迭代来完成寻优过程的一种随机搜索算法。在 ABC 算法中, 已知当前记忆位置 X_i , 通过式(1)产生新的候选位置 V_i 。

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

式中, k 和 j 是随机选择的下标, 并且满足 $k \in \{1, 2, \dots, SN\}$, $j \in \{1, 2, \dots, D\}$, $k \neq i$; Φ_{ij} 为 $[-1, 1]$ 之间的随机数。式(1)称为人工蜂群算法的搜索策略。

然而, 新的候选位置 V_i 与记忆位置 X_i 相比, 只有一个分量有差异, 其余的都相同。换句话说, 搜索策略(1)只能进行一维搜索, 因此其搜索能力十分有限。为此, 本文给出一个改进的 ABC 算法, 其搜索策略为

收稿日期: 2010-03-29; 修回日期: 2011-01-12。

基金项目: 国家自然科学基金(60974082); 中央高校基本科研业务费专项资金(K50510700004)资助课题

作者简介: 高卫峰(1985-), 男, 博士研究生, 主要研究方向为最优化理论与方法及进化算法。E-mail: gaoweifeng2004@126.com

$$v_{ij_z} = x_{ij_z} + \phi_{ij_z}(x_{ij_z} - x_{kj_z}), z = 1, 2, \dots, m \quad (2)$$

式中, $1 \leq m \leq D$ 。相对于搜索策略(1), 策略(2)扩大了搜索范围。以二维问题为例, 搜索策略(1)只能在直线上产生新的候选位置, 而策略(2)可以在平面内产生新的候选解。

为了进一步提高 ABC 算法的收敛速度, 当最优适应值经过有限的循环次数 L 之后, 并且下降量小于阈值 ϵ 时, 进行杂交和变异操作, 其表达式为

$$u_k = \lambda_k x_{ik} + (1 - \lambda_k) x_{best,k} \quad (3)$$

$$Z = X_{best} + N(0, 1) \quad (4)$$

式中, $\lambda_k \in [-1, 1]$ 为均匀分布的随机数, $k = 1, 2, \dots, D$; X_{best} 为当前最优位置; X_i 为不同于 X_{best} 的随机选取的个体。

然后, 用新产生的个体 Z 中较好者替换 X_i 。本文将上述操作称为淘汰规则。

2 HABC 算法

2.1 DE 算法

DE 算法^[12]是一种随机的并行直接搜索算法, 其基本思想是: 从某一随机产生的初始群体开始, 按照一定的操作规则不断迭代计算, 并根据每一个体的适应度值, 保留优良个体, 淘汰劣质个体, 引导搜索过程向最优解逼近。整个运行过程中保持群体的规模不变, 有 3 种运算贯穿着整个算法的执行过程, 即变异、交叉和选择。分别描述为

(1) 变异。对于个体 X_i , 生成的变异个体为

$$U = X_{r1} + F(X_{r2} - X_{r3}) \quad (5)$$

式中, X_{r1}, X_{r2}, X_{r3} 为从进化群体中随机选取的互不相同的 3 个个体; F 为缩放比例因子, 用于控制差向量的影响大小。

(2) 交叉。为增加群体多样性, 交叉操作被引入 DE 算法, 为

$$v(i) = \begin{cases} u(i), & p \leq CR \\ x(i), & p > CR \end{cases} \quad (6)$$

其中, $i = 1, 2, \dots, D$, D 为解空间维数; $CR \in [0, 1]$ 为杂交参数; p 为 $[0, 1]$ 之间的随机数。

(3) 选择。在 DE 算法中, 选择操作采取贪婪策略, 即只有当产生的子代个体优于父代个体时才被保留, 否则父代个体将保留至下一代。

2.2 HABC 算法的基本思想及实现步骤

基于 ABC 算法在寻优过程中有收敛速度慢、容易出现“早熟”的缺点, 可能使算法限于局部最优, 严重影响算法的性能。针对这一问题, 本文在改进 ABC 算法的基础上, 综合 DE 算法, 提出了 HABC 算法。算法的主要思想是: 在 ABC 算法的循环过程中, 首先构造一个新的搜索策略, 该策略扩大了个体的搜索范围, 提高了局部搜索的能力, 从而加快了算法的收敛速度; 然后, 为了进一步提高算法的收敛能力, 引入淘汰规则, 当满足淘汰规则的时候, 随机选择一个与最优个体不同的个体, 通过与最优个体进行算术杂交或最优个体的变异来更新自己, 以加快适应值向最优值更新; 最后, 对种群的个体进行差分操作, 来维护群体的多样性。虽然这种混合机制是简单的, 没有引入复杂的操作。但从第 3 节的仿真实验可以看出, 混合算法成功地融合了两种算法的优点, 克服了他们的不足之处, 算法的性能得到很大的提高。

混合人工蜂群算法的基本步骤如下:

步骤 1 设置种群规模 SN 、阈值 ϵ 、 L 、 m 及 ABC 算法和 DE 算法的相关初始参数, 随机生成 SN 个解构成初始蜜源位置。

步骤 2 采蜜蜂按照式(2)搜索一个新蜜源, 并计算该位置的适应度, 如果新位置优于原来的位置, 则用新位置替换原位置。

步骤 3 观察蜂根据蜂源的花蜜量依概率选择一个蜜源位置, 并根据式(2)产生一个新位置, 评价该位置, 如果新位置优于观察蜂选择的位置, 则用新蜜源替换所选蜜源。

步骤 4 若存在放弃的蜜源, 该处的采蜜蜂变为侦察蜂, 用随机生成的蜜源替换掉。

步骤 5 若满足淘汰规则, 随机选择一个个体, 按照式(3)和式(4)产生两个新的蜂源, 并评价他们的适应度, 用贪婪选择机制进行更新。

步骤 6 种群中的所有个体进行差分变异、交叉、选择操作, 更新蜂源的位置。记录下当前的最优位置及适应值。

3 仿真实验

3.1 HABC 算法用于参数优化

所要求解的调频(frequency-modulated, FM)合成器参数优化问题来自文献[14]。这个问题所要确定的 FM 合成器的 6 个参数为 $X = \{a_1, \omega_1, a_2, \omega_2, a_3, \omega_3\}$, 由式(7)给出, 用式(7)估计式(8)。

$$\begin{aligned} y(t) &= a_1 \sin(\omega_1 t\theta + a_2 \sin(\omega_2 t\theta + a_3 \sin(\omega_3 t\theta))) \quad (7) \\ y_0(t) &= 1.0 \sin(5.0t\theta - 1.5 \sin(4.8t\theta + 2.8 \sin(4.9t\theta))) \quad (8) \end{aligned}$$

式中, $\theta = 2\pi/100$, 搜索区间为 $[-6.4, 6.35]^6$ 。

优化的目标是最小化由式(9)给出的误差平方和。这是一个复杂的高维多峰函数优化问题, 它的最优值为 0。

$$f(X) = \sum_{t=0}^{100} (y(t) - y_0(t))^2 \quad (9)$$

在仿真实验中, HABC 算法的参数设置如下为 $SN=50$, $m=4$, $L=15$, $\epsilon=0.01$, $CR=0.9$, $F=0.5$ 。算法的终止条件为达到最优解 0 或函数的评价次数大于 50 000。对 DE 算法、ABC 算法及 HABC 算法随机独立运行 30 次, 计算结果如表 1 所示。测试结果表明 HABC 算法的鲁棒性、收敛精度和速度要优于基本 ABC 算法和 DE 算法, 能有效提高基本 ABC 算法的全局搜索性能和基本 DE 算法的搜索速度。

表 1 不同算法对 FM 合成器参数优化结果比较

	平均值	最差值	方差	平均函数评价次数
DE 算法	0.712 489	11.207 19	2.669 26	45 193.3
ABC 算法	10.655 91	17.801 46	5.854 802	50 000.0
HABC 算法	1.52e-029	8.18e-029	1.49e-029	28 027.8

为了进一步比较本文所提出的 HABC 算法的性能, 将本文所得结果与文献[14]给出 6 个算法的结果进行比较, 见表 2。文献[14]的终止条件是函数评价次数大于 1 000 000, 相比而言本文的终止条件是很苛刻的。表 2 进一步表明 HABC 算法在搜索速度和收敛精度上是其他算法所无可比拟的。图 1 给出了 3 种算法收敛性能比较。

表2 HABC与文献[14]结果比较

	BFOA 算法	HPSO-TVAC 算法	EA 算法	BSO 算法	ABFOA1 算法	ABFOA2 算法	HABC 算法
均值	2.748 49	0.765 35	0.015 4	0.759 32	0.003 65	0.004 51	1.52e-29
方差	0.831 4	0.115 4	0.002 64	0.273 5	0.000 851	0.001 63	1.49e-29

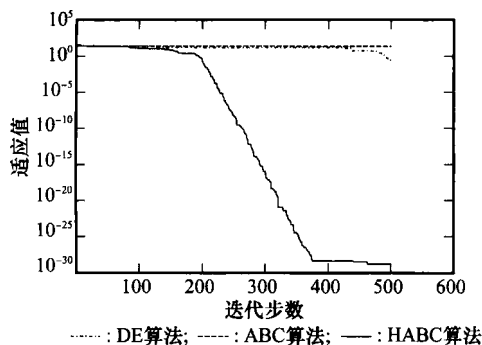


图1 DE算法、ABC算法和HABC算法收敛性能比较

3.2 HABC 算法用于线性系统逼近

本节应用 HABC 算法求解线性系统逼近问题来验证实际应用的有效性。所求解的稳定线性系统逼近问题来自文献[15-18],需要确定的参数为 $a_{2,1}$ 、 $k_{2,p}$ 、 $\tau_{2,z}$ 、 $\tau_{2,p}$,每个参数都位于 $[0, +\infty)$ 上。实验中,分别在固定区域以及文

献[15]采用的动态扩展区域运行算法。本实验采用 $SN=10$ 、 $m=2$ 、 $F=0.9$,其余参数与第3.1节相同。算法的终止条件为函数的评价次数大于19 500,每一类实验独立运行算法30次。

首先,用固定搜索区间的 HABC 算法同时搜索参数 $a_{2,1}$ 、 $k_{2,p}$ 、 $\tau_{2,z}$ 、 $\tau_{2,p}$,分别令搜索区间为 $[0, 10]^4$ 、 $[0, 50]^4$ 、 $[0, 100]^4$ 、 $[0, 150]^4$ 进行测试。然后采用搜索区间动态扩展的 HABC 算法来搜索参数 $a_{2,1}$ 、 $k_{2,p}$ 、 $\tau_{2,z}$ 、 $\tau_{2,p}$,其中初始搜索区间为 $[0, 0.1]^4$,扩展周期为10,区间扩展参数为2。HABC 算法与其他算法的最优结果比较如表3所示。

表3中 J 表示最优解的值。对比表3中 HABC 算法、DE 算法和 AIRA 算法所得最优逼近模型可以看出无论在固定区域还是动态扩展区域搜索, HABC 算法都能找到好于其他算法的逼近模型。同时, HABC 算法性能受搜索区间扩大的影响明显弱于 DE 算法,甚至在一定范围内,搜索区间的增大,不会使 HABC 算法的搜索结果变差。并且, HABC 算法得到的逼近误差也明显小于 MGA 算法的逼近误差。此外,应用 HABC 算法求解非稳定线性系统逼近问题亦取得了较好效果,限于篇幅,这里不加赘述。

表3 不同算法逼近稳定线性系统的最优结果比较

搜索区间	算法	逼近模型	J
$[0, 10]^4$	DE 算法	$H_2(s) = \frac{0.026\ 611\ 4(s+6.106\ 468\ 21)e^{0.489\ 442\ 2s}}{s^2+1.651\ 052\ 64s+0.257\ 571\ 66}$	$1.430\ 8 \times 10^{-4}$
	AIRA 算法	$H_2(s) = \frac{0.021\ 430\ 231(s+6.130\ 164\ 886)e^{-0.454\ 259\ 829s}}{s^2+1.226\ 172\ 374s+0.208\ 228\ 229}$	$4.488\ 9 \times 10^{-5}$
	HABC 算法	$H_2(s) = \frac{0.013\ 448\ 685(s+10.000\ 000\ 000)e^{-0.408\ 534\ 200s}}{s^2+1.252\ 364\ 442s+0.213\ 167\ 229}$	1.6757×10^{-5}
$[0, 50]^4$	DE 算法	$H_2(s) = \frac{0.004\ 955\ 62(s+35.050\ 531\ 3)e^{-0.390\ 779\ 37s}}{s^2+1.760\ 686\ 2s+0.275\ 317\ 59}$	$6.256\ 8 \times 10^{-5}$
	AIRA 算法	$H_2(s) = \frac{0.006\ 473\ 125(s+20.835\ 707\ 811)e^{-0.360\ 175\ 108s}}{s^2+1.254\ 970\ 252s+0.213\ 777\ 917}$	$8.226\ 9 \times 10^{-6}$
	HABC 算法	$H_2(s) = \frac{0.002\ 759\ 328(s+49.322\ 898\ 58)e^{-0.337\ 144\ 378s}}{s^2+1.265\ 774\ 608s+0.215\ 721\ 113}$	$5.878\ 2 \times 10^{-6}$
$[0, 100]^4$	DE 算法	$H_2(s) = \frac{0.004\ 158\ 86(s+65.911\ 472\ 8)e^{-0.590\ 709\ 7s}}{s^2+2.714\ 033\ 3s+0.434\ 486\ 90}$	$1.002\ 1 \times 10^{-3}$
	AIRA 算法	$H_2(s) = \frac{0.003\ 374\ 488(s+39.802\ 784\ 056)e^{-0.332\ 274\ 27s}}{s^2+1.250\ 285\ 692s+0.212\ 893\ 270}$	8.5147×10^{-6}
	HABC 算法	$H_2(s) = \frac{0.001\ 361\ 766(s+100.000\ 000\ 0)e^{-0.327\ 238\ 814s}}{s^2+1.266\ 412\ 97s+0.215\ 845\ 673}$	$5.655\ 6 \times 10^{-6}$
$[0, 150]^4$	DE 算法	$H_2(s) = \frac{0.002\ 553\ 8(s+68.170\ 640\ 5)e^{-0.378\ 792\ 9s}}{s^2+1.256\ 812\ 898s+0.214\ 089\ 609}$	$6.112\ 9 \times 10^{-5}$
	AIRA 算法	$H_2(s) = \frac{0.006\ 163\ 363(s+21.914\ 785\ 100)e^{-0.357\ 316\ 587s}}{s^2+1.256\ 812\ 898s+0.214\ 089\ 609}$	$7.736\ 9 \times 10^{-6}$
	HABC 算法	$H_2(s) = \frac{0.001\ 623\ 456(s+83.871\ 515\ 25)e^{-0.329\ 105\ 811s}}{s^2+1.266\ 286\ 406s+0.215\ 821\ 991}$	$5.684\ 5 \times 10^{-6}$
动态扩展区间	MGA 算法	$H_2(s) = \frac{0.017\ 252\ 0(s+7.766\ 802\ 4)e^{-0.434\ 611\ 9s}}{s^2+1.248\ 371\ 7s+0.212\ 384\ 2}$	2.7180×10^{-5}
	AIRA 算法	$H_2(s) = \frac{0.004\ 717\ 881(s+28.822\ 747\ 965)e^{-0.351\ 0422\ 12s}}{s^2+1.264\ 931\ 127s+0.215\ 537\ 542}$	6.5526×10^{-6}
	HABC 算法	$H_2(s) = \frac{0.001\ 867\ 753(s+72.897\ 771\ 87)e^{-0.330\ 850\ 999s}}{s^2+1.266\ 227\ 867s+0.215\ 811\ 394}$	$5.716\ 4 \times 10^{-6}$

4 结束语

ABC 算法是一种新颖的群集智能优化算法,具有操作简单、易于实现及控制参数少等特点。本文针对 ABC 算法收敛速度慢、容易出现“早熟”的缺点,在迭代中引入淘汰规则和新的搜索策略,以提高算法的收敛速度;同时成功地将 DE 算法和 ABC 算法融合,提出了 HABC 算法。参数优化和典型的线性系统逼近工程优化问题的实验都表明,混合算法的性能得到很大的提高。进一步的研究可使算法适应于非线性动态系统逼近、组合优化、多目标优化等问题,使得算法体现出更多的应用价值。

参考文献:

- [1] Karaboga D. An idea based on honey bee swarm for numerical optimization[R]. Kayseri, Turkey: Erciyes University, 2005.
- [2] Karaboga D, Basturk B. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm[J]. *Journal of Global Optimization*, 2007, 39(3): 459 - 471.
- [3] Karaboga D, Basturk B. A comparative study of artificial bee colony algorithm[J]. *Applied Mathematics and Computation*, 2009, 214(1): 108 - 132.
- [4] Karaboga D, Basturk B. On the performance of artificial bee colony (ABC) algorithm[J]. *Applied Soft Computing*, 2010, 8(1): 687 - 697.
- [5] Singh A. An artificial bee colony algorithm for the leaf-constrained minimum spanning tree problem[J]. *Applied Soft Computing*, 2009, 9(2): 625 - 631.
- [6] Kang F, Li J J, Xu Q. Structural inverse analysis by hybrid simplex artificial bee colony algorithms[J]. *Computers and Structures*, 2009, 87(13): 861 - 870.
- [7] Karaboga N. A new design method based on artificial bee colony algorithm for digital IIR filters[J]. *Journal of the Franklin Institute*, 2009, 346(4): 328 - 348.
- [8] Xu C F, Duan H B. Artificial bee colony (ABC) optimized edge potential function (EPF) approach to target recognition for low-altitude aircraft[J]. *Pattern Recognition Letters*, 2010, 31(13): 1759 - 1772.
- [9] Karaboga D, Ozturk C. A novel clustering approach: artificial bee colony (ABC) algorithm[J]. *Applied Soft Computing*, 2011, 11(1): 652 - 657.
- [10] Alatas B. Chaotic bee colony algorithms for global numerical optimization[J]. *Expert Systems with Applications*, 2010, 37(8): 535 - 541.
- [11] Xu C F, Duan H B, Liu F. Chaotic artificial bee colony approach to uninhabited combat air vehicle (UCAV) path planning[J]. *Aerospace Science and Technology*, 2010, 14(8): 535 - 541.
- [12] Storn R P K. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces[J]. *Journal of Global Optimization*, 1997, 11(4): 341 - 359.
- [13] 孟红云, 张小华, 刘三阳. 用于约束多目标优化问题的双群体差分进化算法[J]. 计算机学报, 2008, 31(2): 228 - 235. (Meng H Y, Zhang X H, Liu S Y. A differential evolution based on double population for constrained multiobjective optimization problem [J]. *Chinese Journal of Computers*, 2008, 31(2): 228 - 235.)
- [14] Dasgupta S, Das S, Abraham A, et al. A daptive computational chemotaxis in bacterial foraging optimization: an analysis[J]. *IEEE Trans. on Evolutionary Computation*, 2009, 13(4): 919 - 941.
- [15] Cheng S L, Huang C Y. Optimal approximation of linear systems by a differential evolution algorithm[J]. *IEEE Trans. on Systems, Man, and Cybernetics-Part A*, 2001, 31(6): 698 - 707.
- [16] Gong M G, Du H F, Jiao L C. Optimal approximation of linear systems by artificial immune response[J]. *Science in China: Series F Information Science*, 2006, 49(1): 63 - 79.
- [17] 张俊岭, 梁昌勇, 杨善林. 具有轮盘反转算子的多 Agent 算法用于线性系统逼近[J]. 控制理论与应用, 2009, 26(1): 39 - 45. (Zhang J L, Liang C Y, Yang S L. Effective multi-agent algorithm with roulette inversion operator for approximating linear systems[J]. *Control Theory & Applications*, 2009, 26(1): 39 - 45.)
- [18] 钟伟才, 刘静, 焦李成. 多智能体遗传算法用于线性系统逼近[J]. 自动化学报, 2004, 30(6): 933 - 938. (Zhong W C, Liu J, Jiao L C. Optimal approximation of linear systems by multi-agent genetic algorithm[J]. *Acta Automatica Sinica*, 2004, 30(6): 933 - 938.)