

基于余弦控制因子和迭代局部搜索的蝙蝠优化算法



郑浩¹ 于俊洋^{1,2} 魏上斐¹

1 河南大学软件学院 河南 开封 475004

2 河南省智能数据工程研究中心 河南 开封 475004

(15516558060@163.com)

摘要 针对蝙蝠算法寻优精度低、易陷入局部极值、求解不稳定的问题,提出了一种基于余弦控制因子和迭代局部搜索策略的蝙蝠的算法。首先在蝙蝠速度公式中加入由余弦因子控制的非线性惯性权重,来动态调节算法全局搜索与局部搜索的平衡,提高算法寻优精度和稳定性。其次,在每轮迭代结束时引入迭代局部搜索策略,扰动局部最优解获得中间状态,并重新搜索上述中间状态得到全局最优解,使算法快速跳出局部最优解,找到全局理论最优。最后与其他算法在12个复杂基准函数上进行仿真实验。结果表明,改进后的算法较好地解决了蝙蝠算法寻优精度不高、易陷入局部极值和求解不稳定的问题。

关键词:蝙蝠算法;余弦控制因子;迭代局部搜索策略;扰动

中图分类号 TP301.6

Bat Optimization Algorithm Based on Cosine Control Factor and Iterative Local Search

ZHENG Hao¹, YU Jun-yang^{1,2} and WEI Shang-fei¹

1 School of Software, Henan University, Kaifeng, Henan 475004, China

2 Henan Intelligent Data Engineering Research Center, Kaifeng, Henan 475004, China

Abstract To solve the problem that bat algorithm is easy to fall into local optimal solution when solving high-dimensional complex problems, an improved bat algorithm is proposed in this paper. Firstly, the nonlinear inertia weight controlled by cosine factor is added to the bat velocity formula to dynamically adjust the balance between global search and local search, so as to improve the accuracy and stability of the algorithm. Secondly, at the end of each iteration, the concept of iterated local search is introduced to perturb the local optimal solution to obtain the intermediate state, and then re-search the intermediate state to get the global optimal solution, which can make it jump out of the local optimal solution quickly. Finally, the simulation results on 12 complex benchmark functions with other algorithms show that the improved algorithm solves the problems of low precision, easy to fall into local extremum and unstable solution.

Keywords Bat algorithm, Cosine control factor, Iterative local search, Disturbance

1 引言

近年来,随着优化问题的规模越来越大,复杂度越来越高,传统的数值优化方法难以在一定时间内给出合理解,而群智能优化算法却能在给定的时间内解决大规模的、复杂的问题。经典的智能算法有模拟达尔文进化论提出的遗传算法(Genetic Algorithm, GA)^[1];模拟飞鸟集群觅食提出的粒子群算法(Particle Swarm optimization, PSO)^[2-3];模拟蚂蚁觅食时释放信息素寻找最短路径提出的蚁群算法(Ant Colony Optimization, ACO)^[4];模拟布谷鸟寄生生育雏行为提出的布谷鸟搜索算法(Cuckoo Search, CS)^[5];模拟灰狼种群捕食行为提出的灰狼优化算法(Grey Wolf Optimizer, GWO)^[6];模拟鲸鱼狩猎提出的鲸鱼优化算法(Whale Optimization Algorithm, WOA)^[7]等等。这些算法已成功应用于机器学习、模式识别、神经网络、控制系统优化、电力、机械设计、机器人、信

号处理、生物信息等多个领域^[8-11]。

蝙蝠算法^[12]是Yang于2010年提出的一种基于蝙蝠通过回声定位进行探索和捕食的行为演变而来的仿生群智能优化算法。蝙蝠算法具有收敛速度快、算法参数少、鲁棒性好等特点,但求解高维函数时也存在寻优精度低、易陷入局部极值、求解不稳定的问题。对此,国内外研究人员做出了各种相应的改进,如文献^[13]提出了一种动态调整惯性权重的自适应蝙蝠算法,在提高算法的收敛速度的同时还提高了种群跳出局部最优值的能力。文献^[14]引入了模拟退火的思想,并对种群中的某些个体进行高斯扰动,在收敛速度和精度方面提升较好。文献^[15]引入了一种基于混沌映射的改进算法,然后提出了一种基于孤岛模型的多种群并行的蝙蝠优化算法,有效地提高了收敛速度和寻优精度。文献^[16]分别使用了3种不同的方法增强了蝙蝠算法的局部和全局搜索特性,实验效果良好。这些改进在相应领域均不同程度地提高了蝙

本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。

基金项目:国家自然科学基金(61672209)

This work was supported by the National Natural Science Foundation of China (61672209).

通信作者:于俊洋(jyyu@henu.edu.cn)

蝙蝠算法的寻优性能,但蝙蝠算法仍有进一步改进的空间。

针对蝙蝠算法在求解高维函数时寻优精度低、易陷入局部极值、求解不稳定的问题,本文提出了一种基于余弦控制因子和迭代局部搜索的蝙蝠算法(Cosine Iterated Local Search Bat Algorithm, CILSBA)。基于余弦控制的惯性权重提高了算法的寻优精度和收敛速度,引入的迭代局部搜索策略提高了算法跳出局部最优解的能力。实验仿真结果表明,改进后的算法具有较快的收敛速度和跳出局部最优解得能力,并且在高维也有较好的寻优精度。

2 蝙蝠算法

蝙蝠在捕食期间会发出声波,当声波遇到猎物或者障碍物时会返回,蝙蝠检测到返回的声波就可以识别到猎物的大小、距离、速度、方向等,从而安全飞行和捕食,蝙蝠算法是根据蝙蝠捕食猎物的这种机制而提出的优化算法。算法具体流程如下:

1)在 d 维的空间中初始化蝙蝠种群个数 n ,最大迭代次数为 N ,蝙蝠的位置为 x_i ,蝙蝠的速度为 v_i ,蝙蝠发出的响度为 A_i^t ,脉冲发射速率为 r_i , t 为当前迭代次数。

2)根据适应度函数 $f(x)$,求出每只蝙蝠的适应度函数值,并且记录最优值的情况下蝙蝠的位置为 x_* 。

3)分别按照式(1)~式(3)更新每个蝙蝠的飞行速度 v_i 和位置 x_i :

$$f_i = f_{\min} + (f_{\max} - f_{\min})\beta \quad (1)$$

$$v_i^t = v_i^{t-1} + (x_i^t - x_*)f_i \quad (2)$$

$$x_i^t = x_i^{t-1} + v_i^t \quad (3)$$

其中, f_{\max} 和 f_{\min} 分别表示蝙蝠发出频率的最大值和最小值, f_i 表示第 i 只蝙蝠发出的频率, β 是在 $[0,1]$ 之间的均匀随机向量, v_i 和 x_i 分别表示第 i 只蝙蝠的速度和位置。

4)在局部搜索部分,局部位置更新公式为:

$$x_{\text{new}} = x_{\text{old}} + \epsilon A^t \quad (4)$$

其中, ϵ 是 $[-1,1]$ 之间的一个随机数, A^t 是第 t 次迭代时所有蝙蝠的平均响度。

5)随着不断的迭代,蝙蝠离目标越来越近,响度逐渐减小的同时脉冲发射率逐渐增加。分别按照式(5)、式(6)更新响度 A_i^t 和脉冲发射率 r_i :

$$A_i^{t+1} = \alpha A_i^t \quad (5)$$

$$r_i^{t+1} = r_i^0 [1 - \exp(-\gamma t)] \quad (6)$$

其中, α 是 $[-1,1]$ 之间的一个随机数, γ 是一个大于 0 的随机数。

6)更新当前蝙蝠位置最优值 x_* 。

7)如果当前迭代次数 t 小于 N ,则返回第三步;如果当前迭代次数 t 等于 N ,则输出全局最优解并且结束蝙蝠算法。

3 基于余弦控制因子和迭代局部搜索的蝙蝠算法 (CILSBA)

3.1 余弦控制因子

从式(2)可以看出,蝙蝠算法的更新速度的系数为 1,这在一定程度上降低了蝙蝠飞行的灵活性,同时减少了种群的多样性,没有考虑蝙蝠前期搜索积累的经验,而且大大降低了蝙蝠飞行的机动性,导致了算法缺乏机动性,从而引起算法全局搜索和局部搜索的不平衡。针对此问题,本文提出了新的

余弦控制因子函数,表达式为:

$$\omega = \cos \frac{\pi \cdot t}{2 \cdot N} \quad (7)$$

速度更新公式变为:

$$v_i^t = \omega * v_i^{t-1} + (x_i^t - x_*)f_i \quad (8)$$

其中, t 为当前迭代次数, N 为最大迭代次数, $\omega \in [0,1]$ 。本文称 ω 为余弦控制因子,前期时, ω 的值偏大,算法搜索范围较大,有利于跳出局部极值。随着迭代次数的增加, ω 逐渐变小,此时蝙蝠进行精细搜索,这是蝙蝠前期搜索经验对蝙蝠后期搜索的帮助和支持的作用,从而可以提高算法的寻优精度与稳定性。

3.2 迭代局部搜索

迭代局部搜索算法(Iterated Local Search, ILS)是一种简单有效的元启发式算法,其工作思想是:首先扰动局部最优解获得中间状态,然后重新搜索上述中间状态得到全局最优解。该算法能够快速跳出局部最优解,具体步骤如下:

1)对当前蝙蝠位置最优值 x_* 进行扰动,得到中间状态 x_{**} ,更新公式如下:

$$x_{**} = x_* * \text{rand}() \quad (9)$$

为排除 0 这个特殊值的干扰,本文 $\text{rand}()$ 的取值区间为 $[0.001,1]$ 。

2)根据中间状态 x_{**} 获得当前局部最优解 $f(x_{**})$ 。

3)如果 $f(x_{**}) < f(x_*)$,则 $x_* = x_{**}$, $f(x_*) = f(x_{**})$ 。

4)输出当前最优解并且结束 ILS 算法。

针对蝙蝠算法易陷入局部最优解的弊端,在每次迭代结束之前加入 ILS 算法的思想,提高了跳出局部最优解的能力,也加快了算法的收敛速度。

3.3 CILSBA 算法流程

1)在 d 维的空间中初始化蝙蝠种群个数 n ,最大迭代次数为 N ,蝙蝠的位置为 x_i ,蝙蝠的速度为 v_i ,蝙蝠发出的响度为 A_i^t ,脉冲发射速率为 r_i , t 为当前迭代次数。

2)根据适应度函数 $f(x)$,求出每只蝙蝠的适应度函数值,并且记录最优值的情况下蝙蝠的位置为 x_* 。

3)分别按照式(1)、式(8)、式(3)更新每个蝙蝠的飞行速度 v_i 和位置 x_i 。

4)按照式(4)进行局部位置更新。

5)分别按照式(5)、式(6)更新响度 A_i^t 和脉冲发射率 r_i 。

6)根据式(9)得到中间状态 x_{**} ,从而得到当前局部最优解 $f(x_{**})$ 。

7)如果 $f(x_{**}) < f(x_*)$,则 $x_* = x_{**}$, $f(x_*) = f(x_{**})$ 。

8)如果当前迭代次数 t 小于 N ,则返回第三步,如果当前迭代次数 t 等于 N ,则输出全局最优解并且结束蝙蝠算法。

4 仿真实验及分析

为了验证本文提出的 CILSBA 算法的寻优性能,本文选取了 12 个标准测试函数进行仿真实验测试,并且与基本蝙蝠算法(Bat Algorithm, BA)和一种基于动态自适应权重和柯西变异的蝙蝠优化算法(Bat Optimization Algorithm, BOA)^[17]进行对比测试。算法所在实验平台为 Windows、64 bit 系统、8G 内存,采用 MATLAB R2016a 进行仿真实验。

4.1 标准测试函数

本文选取的 12 个标准测试函数,具体如表 1 所列。

由表 2 可以看出,在函数 $f_1(x), f_3(x), f_4(x), f_6(x), f_7(x)$ 中,BA,BOA 算法求解精度随着维度的增加而降低,但是 CILSBA 算法在多个高维度下均可以寻得最优解。在函数 $f_{10}(x), f_{12}(x)$ 中,BA,BOA 算法在少数维度下亦可寻得最优解,但是 CILSBA 在 10 维、30 维、50 维均可寻的最优解。对于每个测试函数,本文改进算法 CILSBA 的最差解、最优解和平均值的精度均明显高于基本的 BA 和 BOA 算法,特别是在高维的寻优精度上比其他两种算法有较好的提升。这充分说明了本文改进算法是有效的。

4.4 收敛速度分析

算法的收敛速度能够直观地显示出算法跳出局部最优解的能力,是评估算法性能的重要标准,图 1—图 12 为 3 种算法仿真实验的收敛曲线。

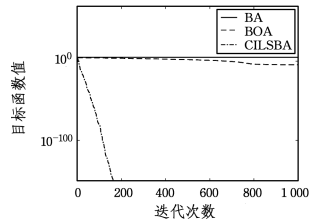


图 1 $f_1(x)$ 的收敛曲线

Fig. 1 Convergence curve of $f_1(x)$

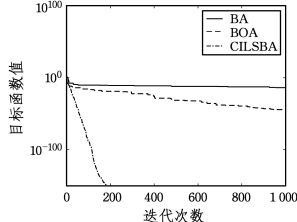


图 2 $f_2(x)$ 的收敛曲线

Fig. 2 Convergence curve of $f_2(x)$

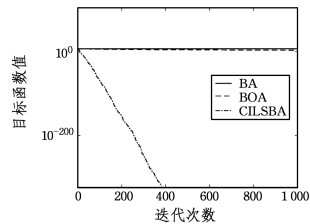


图 3 $f_3(x)$ 的收敛曲线

Fig. 3 Convergence curve of $f_3(x)$

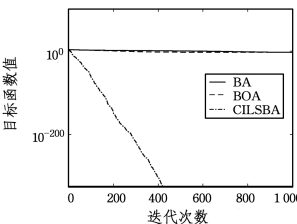


图 4 $f_4(x)$ 的收敛曲线

Fig. 4 Convergence curve of $f_4(x)$

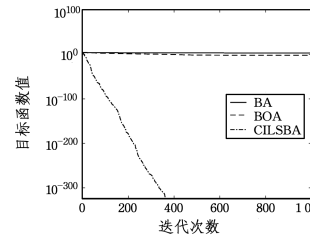


图 5 $f_5(x)$ 的收敛曲线

Fig. 5 Convergence curve of $f_5(x)$

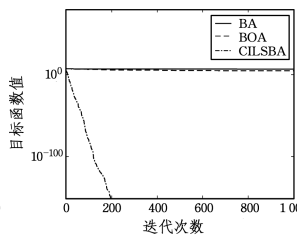


图 6 $f_6(x)$ 的收敛曲线

Fig. 6 Convergence curve of $f_6(x)$

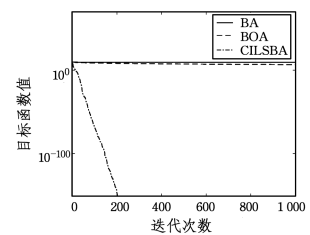


图 7 $f_7(x)$ 的收敛曲线

Fig. 7 Convergence curve of $f_7(x)$

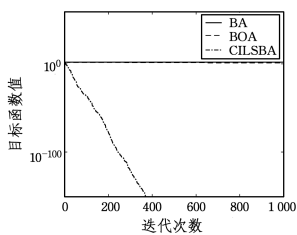


图 8 $f_8(x)$ 的收敛曲线

Fig. 8 Convergence curve of $f_8(x)$

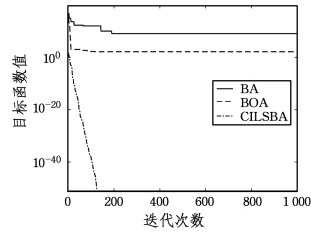


图 9 $f_9(x)$ 的收敛曲线

Fig. 9 Convergence curve of $f_9(x)$

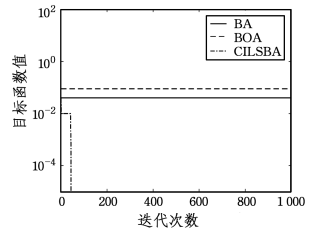


图 10 $f_{10}(x)$ 的收敛曲线

Fig. 10 Convergence curve of $f_{10}(x)$

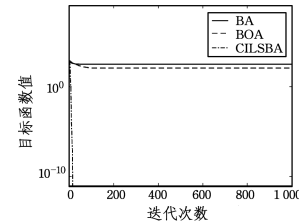


图 11 $f_{11}(x)$ 的收敛曲线

Fig. 11 Convergence curve of $f_{11}(x)$

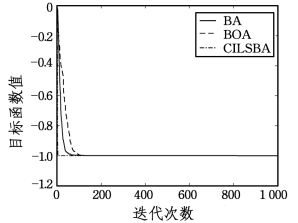


图 12 $f_{12}(x)$ 的收敛曲线

Fig. 12 Convergence curve of $f_{12}(x)$

由图 1、图 2、图 5、图 9、图 11 可见,BOA 算法在寻优精度上优于 BA 算法,CILSBA 则明显优于 BA 和 BOA 算法,由图 10—图 12 可见,CILSBA 算法的收敛曲线近乎垂直,说明该算法具有良好的跳出局部最优解的能力。综上所述,可以看出,BOA 算法相比基本蝙蝠算法虽然有了改进,寻优精度和速度有一定的提高,但是在高维空间仍然有改进空间。本文提出的 CILSBA 算法在保持寻优精度和收敛速度的前提下,同时在低维空间和高维空间都能够较快地搜索到全局最优解。

结束语 本文针对法蝙蝠算法在求解高维函数寻优精度低、易陷入局部最优解和求解不稳定的问题,提出了一种的基于余弦控制因子和迭代局部搜索的蝙蝠算法。该算法基于余弦控制的惯性权重提高了算法的寻优精度和收敛速度,积累前期搜索经验提升后期搜索效果,同时引入迭代局部搜索的概念,提高了收敛速度。仿真实验表明,本文所提算法 CILSBA 有较好的寻优精度和收敛速度,适合用于求解高维函数优化问题。下一步的研究方向是运用本文提出的蝙蝠优化算法在实际工程中的应用。

参考文献

- [1] GOLDBERG D E. Genetic algorithm in search, optimization and machine learning [M]. Boston: Addison-Wesley Longman Publishing Co. Inc, 1989.
- [2] EBERHART R, KENNEDY J. A new optimizer using particle swarm theory [C]// Proceedings of the Sixth International Symposium on Micro Machine and Human Science. 1995: 39-43.
- [3] KENNEDY J, EBERHART R. Particle swarm optimization [C]// Proceedings of IEEE International Conference on Neural Networks. 1995: 1942-1948.
- [4] DORIGO M, MANIEZZO V, COLONNI A. Ant system: optimization by a colony of cooperating agents [C] // IEEE Transactions on Systems Man & Cybernetics. 1996: 29-41.
- [5] YANG X S, DEB S. Cuckoo search via Levy flights [C] // Proceedings of World Congress on Nature & Biologically Inspired

- Computing, India; IEEE Publications, 2009; 210-214.
- [6] MIRJALILI S, MIRJALILI S M, LEWIS A. Grey Wolf Optimizer [J]. *Advances in Engineering Software*, 2014, 69: 46-61.
- [7] MIRJALILI S, LEWIS A. The whale optimization algorithm [J]. *Advances in Engineering Software*, 2016(95): 51-67.
- [8] FU J C, LU Q S. Fault sections location of distribution based on bat algorithm [J]. *Power System Protection and Control*, 2015, 43(16): 100-105.
- [9] SHENG X H, YE C M. Application of Bat Algorithm to Permutation Flow-Shop Scheduling Problem [J]. *Industrial Engineering Journal*, 2013, 16(1): 119-124.
- [10] FAN L, WEI Z N, LI H J, et al. Short-term wind speed interval prediction based on VMD and BA-RVM algorithm [J]. *Electric Power Automation Equipment*, 2017, 37(1): 93-100.
- [11] GUPTA D, ARORA J, AGRAWAL U, et al. Optimized Binary Bat algorithm for classification of white blood cells [J]. *Measurement*, 2019, 143: 180-190.
- [12] YANG X S. A New meta heuristic Bat-Inspired Algorithm [M]// *Nature Inspired Cooperative Strategies for Optimization (NIS-CO 2010)*. Berlin Heidelberg; Springer-Verlag, 2010: 65-74.
- [13] PEI Y H, LIU J S, LI Y. Adaptive Bat Algorithm with Dynamically Adjusting Inertia Weight [J]. *Computer Science*, 2017, 44(6): 240-244.
- [14] HE X S, DING W J, YANG X S. Bat algorithm based on simulated annealing and Gaussian perturbations [J]. *Application Research of Computers*, 2014, 31(2): 392-397.
- [15] GUO S S, WANG J S, MA X X, et al. Improved Bat Algorithm Based on Multipopulation Strategy of Island Model for Solving Global Function Optimization Problem [J]. *Computational Intelligence and Neuroscience*, 2019.
- [16] YILMAZ S, KÜÇÜKSİLLE E U. A new modification approach on bat algorithm for solving optimization problems [J]. *Applied Soft Computing Journal*, 2015, 28: 259-275.
- [17] ZHAO Q J, LI J, YU J Y, et al. Bat Optimization Algorithm Based on Dynamically Adaptive Weight and Cauchy Mutation [J]. *Computer Science*, 2019, 46(S1): 89-92.



ZHENG Hao, born in 1996, postgraduate. His main research interests include Intelligent algorithm.



YU Jun-yang, born in 1982, Ph.D, professor, is a member of China Computer Federation. His main research interests include cloud computing and big data.

(上接第 45 页)

- [24] QIN C, SUN Y, DONG Y. A New Method for Identifying Essential Proteins Based on Network Topology Properties and Protein Complexes [J]. *PLoS One*, 2016, 11(8): e0161042.
- [25] ZHANG X, XIAO W, HU X. Predicting essential proteins by integrating orthology, gene expressions, and PPI networks [J]. *PLoS One*, 2018, 13(4): e0195410.
- [26] LI M, LU Y, NIU Z, et al. United Complex Centrality for Identification of Essential Proteins from PPI Networks [J]. *IEEE/ACM Trans Comput Biol Bioinform*, 2017, 14(2): 370-380.
- [27] LEI X, YANG X. A new method for predicting essential proteins based on participation degree in protein complex and subgraph density [J]. *PLoS One*, 2018, 13(6): e0198998.
- [28] LI M, LI W, WU F X, et al. Identifying essential proteins based on sub-network partition and prioritization by integrating sub-cellular localization information [J]. *Journal of Theoretical Biology*, 2018, 447: 65-73.
- [29] LEI X, YANG X, FUJITA H. Random walk based method to identify essential proteins by integrating network topology and biological characteristics [J]. *Knowledge-Based Systems*, 2019, 167: 53-67.
- [30] BURT R S. *Structural Holes: The Social Structure of Competition* [M]. Harvard University Press, 2009.
- [31] IOANNIS X, LUKASZ S, JOYCE D X. DIP, the database of interacting proteins; a research tool for studying cellular networks of protein interactions [J]. *Nucleic Acids Research*, 2002, 30(1): 303-305.
- [32] KROGAN N J, CAGNEY G, YU H, et al. Global landscape of protein complexes in the yeast *Saccharomyces cerevisiae* [J]. *Nature*, 2006, 440(7084): 637-643.
- [33] MEWES H W, DIETMANN S, FRISHMAN D, et al. MIPS: analysis and annotation of genome information in 2007 [J]. *Nucleic Acids Res*, 2008, 36: 196-201.
- [34] MICHAEL C J, CAROLINE A, CATHERINE B. SGD: *saccharomyces genome database* [J]. *Nucleic Acids Research*, 1998, 26(1): 73-79.
- [35] ZHANG R, LIN Y. DEG 5.0, a database of essential genes in both prokaryotes and eukaryotes [J]. *Nucleic Acids Res*, 2009, 37: 455-458.
- [36] DE MATTEIS G, GRAUDENZI A, ANTONIOTTI M. A review of spatial computational models for multi-cellular systems, with regard to intestinal crypts and colorectal cancer development [J]. *J Math Biol*, 2013, 66(7): 1409-1462.



YANG Zhuang, born in 1992, MS. His main research interests include algorithms and complexity theory, and computational biology.



LIU Pei-qiang, born in 1970, Ph.D, professor, is a member of China Computer Federation. His main research interests include algorithms and complexity theory, and computational biology.