基于随机扰动的自适应布谷鸟算法

叶亚荣,贺兴时,张 超
(西安工程大学 理学院,陕西 西安 710048)

摘 要: 布谷鸟搜索算法(CS)是模仿布谷鸟的繁殖行为所建的一种元启发式算法。这是一种新兴启发算法,通过模拟某些种属布谷鸟的寄生育雏崽来有效地求解最优化问题。针对该算法计算精度不高,收敛速度慢,容易陷入局部最优等缺陷,提出了一种基于自适应步长随机扰动的布谷鸟搜索算法(ASCS)。在增加鸟窝位置变化活力的基础上,对鸟窝位置之间的距离引入自适应的调整步长因子,可以防止算法在运行过程中陷入局部最优。同时为了更大程度地提高鸟窝的计算精度与搜索速度,在寻找最优鸟窝的时候增加一个扰动因子,提高了算法的收敛速度。通过 7 个测试函数进行仿真实验,结果证明了该算法的可行性,其性能显著优于原始的布谷鸟算法。

关键词: 布谷鸟算法; 自适应步长; 鸟窝位置; 随机扰动
中图分类号: TP301.6 **文献标识码:** A **文章编号:** 1673-629X(2019)05-0077-04
doi: 10.3969/j.issn.1673-629X.2019.05.17

An Adaptive Cuckoo Algorithm Based on Random Disturbance

YE Ya-rong, HE Xing-shi, ZHANG Chao
(School of Science, Xi'an Polytechnic University, Xi'an 710048, China)

Abstract: The cuckoo search (CS) is a kind of meta-heuristic algorithm constructed by imitating the breeding behavior of cuckoos. This is an emerging heuristic algorithm that effectively solves optimization problems by simulating the breeding of some species of cuckoos. Aiming at the shortcomings of this algorithm, such as low computational accuracy, slow convergence speed and easy to be trapped in local optimum, we propose a cuckoo search based on adaptive step size random perturbation (ASCS). On the basis of increasing the vitality of the bird nest position, an adaptive adjustment step factor is introduced to the distance between the bird nest positions to prevent the algorithm from falling into a local optimum during the operation. At the same time, in order to improve the calculation accuracy and search speed of the bird nest to a greater extent, an interference factor is added to find the optimal bird nest, which improves the convergence speed of the algorithm. The simulations are performed by seven test functions. The experiment shows that the ASCS is feasible and its performance is significantly better than that of CS..

Key words: cuckoo algorithm; adaptive step size; nest location; random disturbance

0 引 言

元启发式算法[1]自提出以来越来越被人们所熟知,其中包括遗传算法[2](genetic algorithm)、蚁群算法[3](ant colony optimization)、粒子群算法[4](particle swarm optimization)、萤火虫算法[5](fire fly algorithm)等。诸多研究已经证实了元启发式算法在实际应用中的可行性。

布谷鸟搜索算法是由英国剑桥大学杨新社和 Suash Deb 提出的,该算法模拟了布谷鸟寻找鸟窝放置新的鸟蛋行为,并有效结合了其他鸟类和果蝇的 Levy 飞行行为。这种算法不仅简单易行,参数较少,而且性能优于粒子群和遗传算法等启发式算法[6]。

布谷鸟算法与其他智能算法一样都存在易陷入局部最优且收敛速度慢等缺陷。对此,王凡等提出了一种基于高斯扰动的布谷鸟搜索算法(GCS)[7],该算法与原有的布谷鸟算法相比能更好地跳出局部最优;王李进等提出了一种逐维改进的布谷鸟搜索算法[8],但是存在求解全局最优解不够快的问题;薛益鸽等提出一种基于动态分组与高斯扰动的布谷鸟算法[9],该算法求解速度快,精度高,但是易陷入局部最优。在算法

收稿日期: 2018-06-22 **修回日期:** 2018-10-25 **网络出版时间:** 2019-03-05
基金项目: 陕西省软科学研究计划项目(2014KRM2801);西安市教育科技重大招标项目(2015ZB-ZY04);陕西省教育专项科研计划项目(16JK1326)
作者简介: 叶亚荣(1992-),女,硕士研究生,研究方向为大数据分析与处理;贺兴时,教授,硕导,研究方向为智能计算、概率论与数理统计。
网络出版地址: http://kns.cnki.net/kcms/detail/61.1450.TP.20190305.1618.002.html

运行过程中,初始化鸟群的质量起着至关重要的作用,好的初始种群能加快种群的收敛速度;宋庆庆等提出基于混沌序列的布谷鸟算法[10];郑洪清等提出了一种自适应步长调整布谷鸟搜索算法[11];陈华等提出了基于 logistic 模型的自适应布谷鸟算法[12];李荣雨等提出一种基于梯度的自适应快速布谷鸟算法[13]。

受上述算法启发,针对布谷鸟算法的缺陷,文中提出了一种基于随机扰动的自适应布谷鸟算法。

1 CS 算法

布谷鸟搜索算法(cuckoo search,SC)是模拟布谷鸟寻找鸟窝放置鸟蛋的行为,并结合 Levy 飞行更新鸟窝位置,完成了每代的更新,如果更新后的位置优于当前位置,则对鸟窝的位置进行更新,否则保留当前位置。由于布谷鸟算法简单、高效、搜索路径优等优点,已在背包优化[14]、机构优化[15]、工程优化[16]等问题中成功应用。

布谷鸟算法是在三个理想状态下进行概述。

规则 1:每个布谷鸟每次只产一个蛋,并随机选择鸟窝放置它。

规则 2:最好的鸟窝(解)会被保留在下一代。

规则 3:可用的巢主鸟窝为 n,巢主鸟能够发现外来鸟的概率是 p_a,其中 $p_a \in [0,1]$。在这种情况下,巢主鸟可将该鸟蛋丢弃,或者干脆抛弃这个鸟窝,在新的位置建立一个全新的鸟窝。

布谷鸟所选取的孵化鸟蛋的鸟窝,其实就是寻找搜索过程中的可行解,实质是当产生更好的鸟窝时,就代替上一代的鸟窝。基于布谷鸟育崽的习惯,布谷鸟下一代鸟窝的更新用 levy 飞行[5]进行更新,公式如下:

$$X_i^{(t+1)} = X_i^{(t)} + \alpha \oplus L(s,\lambda) \tag{1}$$

其中

$$L(s,\lambda) = \frac{\lambda\Gamma(\lambda)\sin(\pi\lambda/2)}{\pi}\frac{1}{s^{1+\lambda}}, s \gg s_0 > 0, \quad 1 < \lambda \leqslant 3 \tag{2}$$

其中,$\alpha > 0$ 为步长因子,它与问题的利害关系的程度相关联,大多数情况下取 $\alpha = o(L/10)$,L 为问题利害关系的特征范围。

2 自适应步长的随机扰动布谷鸟算法

2.1 自适应鸟窝位置更新

为了加快布谷鸟算法的收敛速度,在原始的布谷鸟算法中定义步长因子:

$$d_i = \frac{\| n_i - n_{\text{best}} \|}{d_{\max}} \tag{3}$$

其中,n_i 表示第 i 代鸟窝的位置;n_{best} 表示此代的最佳状态的鸟窝位置;$d_{\max}$ 表示最好位置与其他所有鸟窝的距离的最大值。

在式 3 的基础上,引入了调解鸟窝位置的自适应调整步长策略:

$$\text{stepsize}_i = \text{step}_{\min} + (\text{step}_{\max} - \text{step}_{\min})\, d_i \tag{4}$$

其中,$\text{step}_{\max}$ 和 $\text{step}_{\min}$ 分别表示步长的最大值和最小值。

位置更新公式如下:

$$s = s + \alpha \oplus \text{stepsize}_i \oplus \varepsilon \tag{5}$$

其中,ε 是与 p_t 同阶的随机矩阵;α 为常数,$\oplus$ 表示点对点乘法,通过多次实验,α 取 0.05 来控制鸟窝的活动范围。

2.2 随机扰动

在 CS 算法的第 t 次迭代后得到最佳的鸟窝位置 $x_i^{(t)}$,$i = 1,2,\cdots,n$,如果下一代的鸟窝不优于当前鸟窝,则不让 $x_i^{(t+1)}$ 直接进入下一次迭代,而是继续进行扰动,使 $x_i^{(t+1)}$ 得到进一步搜索。首先记 $x_i^{(t+1)}$,$i = 1,2,\cdots,n$ 组成的矩阵为:$p_{t+1} = [x_1^{(t+1)}, x_2^{(t+1)}, \cdots, x_n^{(t+1)}]^{\mathrm{T}}$,设 $x_i^{(t+1)}$ 为 d 维向量,则 p_{t+1} 为 $d \times n$ 阶矩阵,GCS 算法的具体操作是给 p_{t+1} 加随机扰动[11-13]。

即

$$p'_{t+1} = p_{t+1} + a \oplus \varepsilon \tag{6}$$

其中,ε 是与 p_{t+1} 同阶的随机矩阵,$\oplus$ 表示点对点乘法。由于 ε 的随机取值范围较大,容易使鸟窝位置的活动范围偏离过大,经过多次实验取 $a = 0.05$ 来控制 ε 的搜索范围。

2.3 改进的布谷鸟算法(ASGCS)流程

1.初始化种群:n host nest $X_i(i = 1,2,\cdots,n)$;

2.计算适应值:$F_i(i = 1,2,\cdots,n)$;

3.While(t <最大迭代次数);

4.通过 Levy flight 与自适应步长来更新鸟窝 X_i;

5.计算更新鸟窝的适应值 F_i;

6.从 n 个鸟窝中随机选取一个鸟窝(记为 j);

7. if($F_i < F_j$)

8.用新的解替换 j;

9.Else 对新的鸟窝进行随机扰动(记为 k);

10.End if

11.通过与发现概率 pa 进行比较,抛弃较差的解;

12.保留结果最优的解;

13.end

3 实验与结果分析

在参数相同的前提下,通过 7 个测试函数对 ASGCS 与 CS 算法性能进行比较,以 MATLAB2016 和 Window7 作为测试平台进行实验。对于所有的测试

函数,种群的个数设置为 $n = 8$,最大迭代次数为 500,　　所示。
搜索精度为 10^5,扰动因子 $\alpha = 0.05$。测试函数如表 1

表 1　测试函数

函数	表达式	搜索空间
Crownedcross	$f_1(x) = 0.0001 \times \left\lvert \sin(\sin x_1) \times \sin x_2 \times e^{\left\lvert 100 - \frac{\sqrt{x_{(1)}^2 + x_{(2)}^2}}{\pi} + 1 \right\rvert^{0.1}} \right\rvert$	[-10,10]
Matyas	$f_2(x) = 0.26 \times (x_1^2 + x_2^2) - 0.48 x_1 x_2$	[-10,10]
Sphere	$f_i(x) = \sum_{i=1}^{n} x_i^2$	[-100,100]
Multigaussian	$f_4(x) = \sum_{i=1}^{D} a_{(i)} \times e^{\left[x_{(i)} - b_{(i)}\right)^2 + \left(x_{(i)} - c_{(i)}\right)^2\right] / d_{(i)}^2}$	[-2,2]
Salomon	$f_5(x) = -1\cos 2\pi \lvert x \rvert + 0.1 \lvert x \rvert + 1$	[-100,100]
Threehumpcamel	$f_6(x) = \sum_{i=1}^{D} \left[D - \sum_{j=1}^{D} \cos x_j + i(1 - \cos x_i - \sin x_i)\right]^2$	[-5,5]
Zakh	$f_7(x) = x_1^2 + x_2^2 + (0.5 \times x_1 + x_2)^2 + (0.5 \times x_1 + x_2)^4$	[-10,10]

在所有参数都一致的条件下,由图 1 可以看出,在 Crownedcross 函数下,原始的布谷鸟算法的初始值小于改进以后的 ASCS 算法的初始值,在后期收敛速度明显优于原始的 CS 算法,迭代次数也明显少于原始的 CS 算法。图 2、图 3 在 Matyas、Sphere 函数下,无论是初始值还是收敛速度以及迭代次数,改进后的 ASCS 算法都优于原始的 CS 算法。由图 4 可以看出,在 Multigaussian 函数下虽然进化过程中收敛速度比较缓慢,但是改进后的 ASCS 算法比原始的 CS 算法能更快收敛。由于篇幅限制,Threehumpcamel、Zakh、Salomon 函数没有进行展示,但是从收敛速度和迭代次数方面看,改进后的布谷鸟算法更好。故改进的 ASCS 算法性能优于原始的 CS 算法。

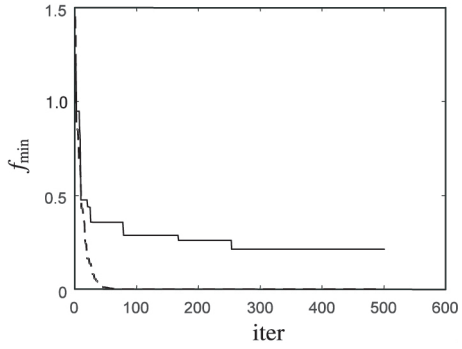

图 1　Crownedcross 函数的寻优曲线

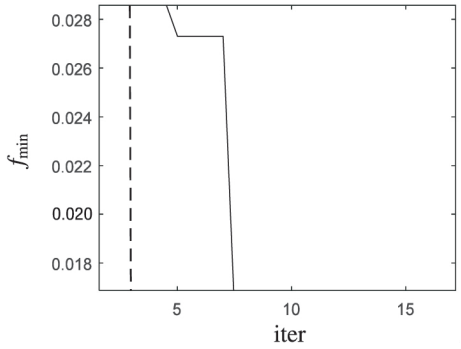

图 2　Matyas 函数的寻优曲线

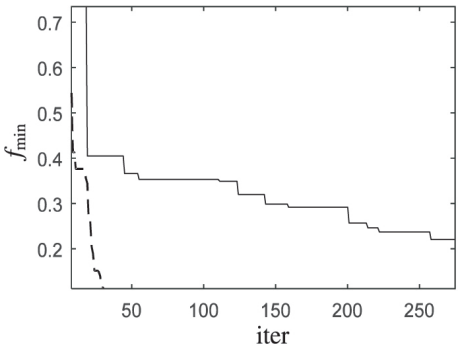

图 3　Sphere 函数的寻优曲线

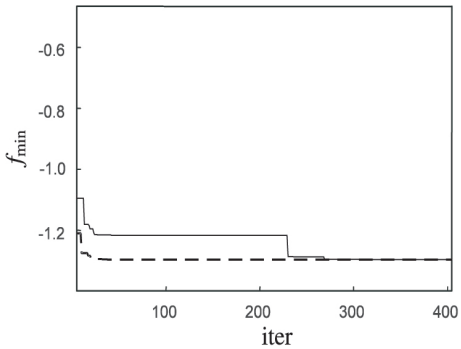

图 4　Multigaussian 函数的寻优曲线

表 2 统计了两种算法在 7 个标准的测试函数下计算的平均最优值。可以看出,Crownedcross 函数在 ASCS 的平均最优值比 CS 算法提高了 4 个点,而 ASCS 的最大值也比原始的 CS 值大,最小值也优于原

始的 CS 算法。对于 Matyas 和 Sphere 函数的最大值与最小值，ASCS 都优于原始的 CS 算法，ASCS 算法的平均最优值也优于原始的 CS 算法。对于 Salomon 函数，ASCS 算法的最大、最小值，平均值都为 0，但是优于原始的 CS 算法。Zakh、Threehumpcamel 函数在最大、最小值以及平均最优值提高了几十个点，虽然 Multigaussian 函数在 ASCS 算法下最优值比原始 CA 算法只提高了 0.000 1，但是综上所述，ASCS 在性能上优于 CS 算法。

表 2 ASCS 与 CS 的性能比较

测试函数	算法	最大值	最小值	平均最优值
Crownedcross	CS	0.284 8	0.118 6	0.169 3
	ASCS	1.000×10^{-4}	1.000×10^{-4}	1.0×10^{-4}
Matyas	CS	$6.161\ 5 \times 10^{-25}$	$1.843\ 9 \times 10^{-29}$	$9.133\ 7 \times 10^{-26}$
	ASCS	$2.812\ 9 \times 10^{-149}$	$4.593\ 2 \times 10^{-180}$	$1.266\ 6 \times 10^{-155}$
Sphere	CS	$1.464\ 1 \times 10^{-10}$	$6.214\ 4 \times 10^{-11}$	$3.889\ 7 \times 10^{-10}$
	ASCS	$2.049\ 6 \times 10^{-59}$	$2.591\ 9 \times 10^{-168}$	$2.273\ 6 \times 10^{-80}$
Multigaussian	CS	−1.297 1	−1.297 0	−1.297 1
	ASCS	−1.296 8	−1.296 9	−1.298 5
Salomon	CS	$1.382\ 2 \times 10^{-4}$	$2.451\ 8 \times 10^{-6}$	$2.887\ 9 \times 10^{-5}$
	ASCS	0	0	0
Threehumpcamel	CS	$5.512\ 1 \times 10^{-23}$	$7.105\ 4 \times 10^{-27}$	$1.177\ 8 \times 10^{-24}$
	ASCS	$5.591\ 4 \times 10^{-115}$	$7.299\ 0 \times 10^{-183}$	$6.373\ 8 \times 10^{-135}$
Zakh	CS	$1.136\ 9 \times 10^{-24}$	$3.188\ 9 \times 10^{-28}$	$2.151\ 8 \times 10^{-26}$
	ASCS	$1.035\ 1 \times 10^{-126}$	$2.175\ 9 \times 10^{-169}$	$1.214\ 1 \times 10^{-151}$

4 结束语

ASCS 算法在 Levy 寻找鸟窝的过程中添加自适应步长因子，可以提高原始算法的搜索能力，之后在寻找最优鸟窝时添加了扰动因子，两种思想的结合充分提高了鸟窝位置的多样性，很大程度上提高了算法的性能。仿真实验结果证明了该算法的可行性。

参考文献：

[1] YANG Xinshe, DEB S. Cuckoo search via Lévy flights [C] //Proceedings of world congress on nature & biologically inspired computing. Coimbatore, India: IEEE, 2009: 210-214.

[2] 席裕庚，柴天佑，恽为民. 遗传算法综述 [J]. 控制理论与应用，1996，13(6)：697-708.

[3] 张纪会，高齐圣，徐心和. 自适应蚁群算法 [J]. 控制理论与应用，2000，17(1)：1-3.

[4] 赵乃刚，邓景顺. 粒子群优化算法综述 [J]. 科技创新导报，2015，12(26)：216-217.

[5] 王吉权，王福林. 萤火虫算法的改进分析及应用 [J]. 计算机应用，2014，34(9)：2552-2556.

[6] YANG Xinshe. Nature inspired metaheuristicalgorithms [M]. United Kinggom: LUNIVER Press, 2010: 105-108.

[7] 王 凡，贺兴时，王 燕. 基于高斯扰动的布谷鸟搜索算法 [J]. 西安工程大学学报，2011，25(4)：566-569.

[8] 王李进，尹义龙，钟一文. 逐维改进的布谷鸟搜索算法 [J]. 软件学报，2013，24(11)：2687-2698.

[9] 薛益鸽，邓辉文. 基于动态分组与高斯扰动的改进布谷鸟算法 [J]. 重庆师范大学学报：自然科学版，2018，35(2)：108-113.

[10] 宋庆庆，贺兴时，郭 旭，等. 基于混沌序列的布谷鸟算法的改进 [J]. 纺织高校基础科学学报，2017，30(3)：423-428.

[11] 郑洪清，周永权. 一种自适应步长布谷鸟搜索算法 [J]. 计算机工程与应用，2013，49(10)：68-71.

[12] 陈 华，张艺丹. 基于 logistic 模型的自适应布谷鸟算法 [J]. 计算机工程与应用，2015，51(20)：31-35.

[13] 李荣雨，刘 洋. 基于梯度的自适应快速布谷鸟搜索算法 [J]. 运筹学学报，2016，20(3)：45-56.

[14] LAYBE A. A novel quantum inspired cuckoo search for Knapsackproblem [J]. International Journal of Bio-Inspired Computation, 2011, 3(5): 297-305.

[15] CANDOMI A H, YANG XINSHE, ALAVI A H. Cukoo search algorithm: a metaheuristic approach to solve structural optimization problems [J]. Engineer with Computers, 2013, 29(2): 245.

[16] YANG Xinshe, Deb S. Engineering optimisation by cuckoo search [J]. International Journal of Mathematical Modelling and Numerical Optimisation, 2010, 1(4): 330-343.