

引用格式:王涛. 非线性权重和柯西变异的蝗虫算法[J]. 微电子学与计算机, 2020, 37(5): 82-86. [WANG T. Grasshopper optimization algorithm with nonlinear weight and cauchy mutation[J]. Microelectronics & Computer, 2020, 37(5): 82-86.]

非线性权重和柯西变异的蝗虫算法

王 涛^{1,2}

(1 中国科学院 国家天文台南京天文光学技术研究所, 江苏 南京 210042;

2 中国科学院 天文光学技术重点实验室(南京天文光学技术研究所), 江苏 南京 210042)

摘 要: 针对蝗虫优化算法收敛速度慢, 易陷入局部最优等问题, 提出一种基于非线性权重和柯西变异的蝗虫优化算法. 首先使用佳点集初始化种群, 均匀种群分布. 然后将线性权重改进为非线性权重, 提高勘探和开发能力. 同时对最优个体采用柯西变异, 增加跳出局部最优能力. 最后通过 6 个基准函数进行测试, 实验表明改进后蝗虫算法收敛速度快、精度高.

关键词: 蝗虫优化算法; 佳点集; 非线性权重; 柯西变异; 勘探; 开发

中图分类号: TP391

文献标识码: A

文章编号: 1000-7180(2020)05-0082-05

Grasshopper optimization algorithm with nonlinear weight and cauchy mutation

WANG Tao^{1,2}

(1 National Astronomical Observatories/Nanjing Institute of Astronomical Optics & Technology, Chinese Academy of Sciences, Nanjing 210042, China; 2 CAS Key Laboratory of Astronomical Optics & Technology, Nanjing Institute of Astronomical Optics & Technology, Chinese Academy of Sciences, Nanjing 210042, China)

Abstract: To overcome the problems of slow convergence, falling into local optima of the standard grasshopper optimization algorithm, a nonlinear weight and cauchy mutation in grasshopper optimization algorithm is proposed. Firstly, the good point set is applied to initial population and uniform population distribution. Then the linearly weight is improved to a nonlinear weight to enhance the exploration and exploitation capabilities of the grasshopper optimization algorithm. At the same time, the cauchy mutation is added to the best grasshopper to increase the ability to jump out of local optima. Finally, six benchmark functions are selected for testing. Experiments show that the improved algorithm has fast convergence and high precision.

Key words: grasshopper optimization algorithm; good point set; nonlinear weight; cauchy mutation; exploration; exploitation

1 引言

群智能算法模仿自然生物觅食行为, 构建相应迭代公式, 通过不断迭代更新种群来求解最优化问题. 近年来不少群智能算法: 如鲸鱼优化算法^[1] (Whale Optimization Algorithm, WOA)、蝗虫优化

算法^[2-9] (Grasshopper Optimization Algorithm, GOA) 等算法及其改进算法不断提出.

为提高蝗虫算法收敛速度并增强跳出局部最优解能力, 本文提出一种基于非线性权重和柯西变异的蝗虫优化算法. 首先通过佳点集初始化种群, 使初始种群均匀分布在解空间中, 增加初始搜索范围. 然

后根据深度学习中 Sigmoid 激活函数,提出一种改进后的非线性惯性权重,提高算法收敛速度和精度.接着对最优蝗虫个体进行柯西变异,避免算法陷入局部最优解.最后通过基准函数来验证所提出的改进后算法的有效性.

2 基本蝗虫优化算法

蝗虫个体位置 X_i 由种群间社交作用 S_i 、重力作用 G_i 以及风力作用 A_i 三者共同影响,则第 i 个蝗虫位置的数学模型如下:

$$X_i = r_1 S_i + r_2 G_i + r_3 A_i \quad (1)$$

$$S_i = \sum_{\substack{j=l \\ j \neq i}}^N s(d_{ij}) \hat{d}_{ij} \quad (2)$$

$$s(r) = f e^{\frac{-r}{l}} - e^{-r} \quad (3)$$

$$G_i = -g e_g \quad (4)$$

$$A_i = u e_w \quad (5)$$

式中, r_1, r_2 和 r_3 是 $(0,1)$ 上的随机数, $d_{ij} = |x_j - x_i|$ 表示第 i 个蝗虫与第 j 个蝗虫间的欧式距离, $\hat{d}_{ij} = \frac{x_j - x_i}{d_{ij}}$ 表示第 i 个蝗虫到第 j 个蝗虫的单位向量, $s(r)$ 表征种群社交影响的强弱和范围, f 表示吸引力强度, l 表示吸引力范围, g 是引力常数, e_g 表示指向地心的单位向量, u 是一个常数漂移量, e_w 表示风向的单位向量.

参数 f 和 l 的不同组合影响着舒适区,吸引区和排斥区的范围,通常取 $f = 0.5$ 和 $l = 1.5$. 为克服在远距离时 $s(r)$ 为 0 的问题,需要将 d_{ij} 映射到 $[1,4]$.

幼年蝗虫在地面觅食,因而在不考虑重力作用 G_i 和风力作用 A_i 作用下,仅依靠种群间社交作用 S_i 更新蝗虫位置,算法可以收敛到舒适区,但不会收敛到指定点.因此针对优化问题,将蝗虫优化算法更新为:

$$X_i^d = c \left(\sum_{\substack{j=l \\ j \neq i}}^N c \frac{ub_d - lb_d}{2} s(|x_j^d - x_i^d|) \frac{x_j^d - x_i^d}{d_{ij}} \right) + T_d^A \quad (6)$$

$$c = c_{\max} - l \frac{c_{\max} - c_{\min}}{L} \quad (7)$$

式中, ub_d 和 lb_d 是解空间的第 d 维的上限和下限, c 是从最大值 $c_{\max}$ 线性递减到最小值 $c_{\min}$ 的惯性权重, l 是当前迭代次数, L 是最大迭代次数, T_d^A 是当前最优个体.

3 改进蝗虫优化算法

3.1 佳点集初始化种群

基本蝗虫算法采用随机初始化的方式来生成初始种群,并在初始种群中确定最优蝗虫个体,进而迭代更新蝗虫位置,因此种群初始的优劣会影响到后期算法寻优效率.初始种群应当尽量均匀分布在解空间中,使算法在迭代初期在全局进行勘探.随机初始化方式并不能保证初始种群的均匀性和多样性.由于佳点集^[10]取点的偏差小于随机取点的偏差,所以采用佳点集来初始化种群,可以保证初始种群的多样性.

若在 n 维欧式空间中的单位立方体中,存在某一点 $r(r_1, r_2, \dots, r_i, \dots, r_n)$, 其第 i 维上数值如下:

$$r_i = 2 \cos\left(\frac{2\pi i}{p}\right) \quad (8)$$

式中, $1 \leq i \leq n$ 且 p 是满足 $p \geq 2n+3$ 的最小素数,则称这样的点是佳点.由佳点构成的点集称为佳点集.

本文结合佳点集和解空间各维度的上下限来初始化蝗虫种群,将引入佳点集初始化种群的蝗虫算法定义为 G-GOA.

3.2 非线性惯性权重

在基本蝗虫算法中式(6)中外部系数 c 起到全局勘探和局部开发的平衡作用.由公式(7)可知惯性权重 c 在迭代中随着迭代次数增加而线性递减.可见基本蝗虫算法在初期勘探和后期开发中都是采取同样程度的递减策略,而无法满足算法在不同时期的实际要求:在算法前期要求惯性权重 c 值比较大且下降较缓慢,让算法有能力(权重大)和时间(下降缓)进行全局勘探,寻找到全局最优解的粗略范围;在算法后期,即勘探到全局最优解的粗略范围后的,要求惯性权重 c 较小且下降迅速,以便算法快速收敛到目前局部最优值.

Sigmoid 函数是深度学习中一种常见的激活函数,其定义如下:

$$S(x) = \frac{1}{1 + e^{-x}} \quad (9)$$

式中, $x \in \mathbb{R}$.

Sigmoid 函数的导数曲线类似于标准正态分布曲线:导数值以原点 0 为中心向 x 轴正向和负向减小.在 $(-\infty, -5)$ 和 $(5, +\infty)$ 上导数接近于 0,即 Sigmoid 函数值变化过小,不利于算法收敛,故舍弃该部分.选取 $[-5, 5]$ 部分作为待改进部分.改进后

的非线性惯性权重定义如下：

$$c = \frac{1}{1 + 1.5e^{\frac{10l}{L}-5}} - 0.1r \tag{10}$$

式中, l 是当前迭代次数, L 是最大迭代次数, r 是 $(0,1)$ 上的随机数, $0.1r$ 可以消除后期导数变化过缓的影响使算法后期迅速收敛, 系数 1.5 为多次实验后取得, 可以平衡勘探和开发能力并加快收敛速度. 系数 5 使 e 的指数部分的可行域落在 $[-5,5]$.

改进后的非线性惯性权重前期权重较大且变化缓慢, 中间寻优过程下降迅速, 后期权重小且变化迅速, 利于快速收敛. 本文将采用佳点集和非线性惯性权重的蝗虫算法定义为 WG-GOA.

3.3 柯西变异

针对基本蝗虫算法易陷入局部最优的问题, 在迭代中添加柯西变异操作, 变异后的蝗虫可以增加种群多样性, 有助于算法跳出局部最优, 寻找到新的最优解.

柯西分布^[11] 是一个连续型概率分布. 标准柯西分布概率密度函数如下：

$$f(x;0,1) = \frac{1}{\pi(1+x^2)} \tag{11}$$

与正态分布相比, 柯西分布在 0 处峰值较小, 从

峰值到 0 值下降趋势较平缓, 即变异范围更均匀, 有利于算法跳出局部最优.

基本蝗虫算法根据当前自身与种群间其他个体间的距离和最优个体的位置来更新下一步自身位置. 根据柯西累积分布函数式(12), 对最优个体构建柯西变异公式, 如下式(13):

$$F(x) = \frac{1}{\pi} \arctan(x) + \frac{1}{2} \tag{12}$$

$$\hat{T}_d = \hat{T}_d(1 + \tan(\pi(r - 0.5))) \tag{13}$$

式中, $\hat{T}_d$ 是当前最优个体位置, $\hat{T}_d$ 是经过柯西变异后的最优蝗虫位置, r 是 $(0,1)$ 上的随机数.

本文将采用佳点集和非线性惯性权重以及柯西变异的蝗虫算法定义为 CWG-GOA.

4 实验与结果分析

为验证本文改进后 CWG-GOA 算法的有效性, 在文献[2]中选择了 6 个基准函数进行仿真测试实验. 如表 1 显示了实验所用的 6 个基准函数的信息. 表 1 中函数 $F_1(x)$ 到 $F_3(x)$ 为单峰基准函数, 用来测试算法的局部开发能力, 函数 $F_4(x)$ 到 $F_6(x)$ 为多峰基准函数, 用来测试算法的全局探勘能力.

表 1 基准函数

基准函数	维度	范围	最小值
$F_1(x) = \sum_{i=1}^n x_i^2$	30	$[-100,100]$	0
$F_2(x) = \sum_{i=1}^i (\sum_{j=1}^i x_j)^2$	30	$[-100,100]$	0
$F_3(x) = \max_i \{ x_i , 1 \leq i \leq n \}$	30	$[-100,100]$	0
$F_4(x) = \sum_{i=1}^n [x_i^2 - 10\cos(2\pi x_i) + 10]$	30	$[-5.12,5.12]$	0
$F_6(x) = -20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2}) - \exp(\frac{1}{n}\sum_{i=1}^n \cos(2\pi x_i)) + 20 + e$	30	$[-32,32]$	0
$F_7(x) = \frac{\pi}{n} \{ 10\sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10\sin^2(\pi y_{i+1})] + (y_n - 1)^2 \} \dots$ $+ \sum_{i=1}^n u(x_i, 10, 100, 4) + \sum_{i=1}^n u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{x_i + 1}{4}$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	30	$[-50,50]$	0

本文将基本蝗虫算法 GOA 以及改进后的 G-GOA、WG-GOA 和 CWG-GOA 进行对比实验. 实验将初始化蝗虫数目设为 30 , 最大迭代次数为 500 , 基本蝗虫算法 GOA 中设 $c_{\max}=1$ 且 $c_{\min}=0.000\ 04$. 实验中四种算法独立运行 30 次, 采用平均值和标准差对算法结果进行评价, 实验结果如表 2 所示.

由表 2 可以看出无论是在单峰基准函数还是在多峰基准函数上, 本文提出的 CWG-GOA 的均值都优于其他算法, 大部分标准差小于其他算法. 同时 CWG-GOA 在单峰基准函数 $F_1(x)$ 实现的精度优于其他基准函数. 在多峰基准函数 $F_4(x)$ 上, 基本 GOA 求解精度很差, 而本文改进后的 CWG-GOA 将求解精度提高了一个数量级. G-GOA 的均值优

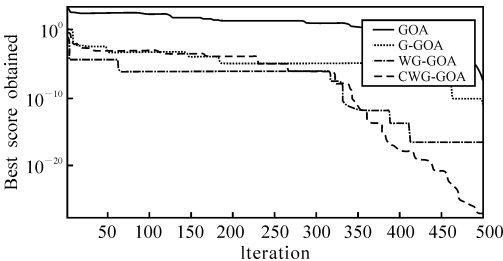
于基本 GOA,表明了佳点集初始化种群的有效性. 在除多峰基准函数 $F_6(x)$ 外, WG-GOA 的均值都低于 G-GOA,验证了非线性惯性权重 的有效性. 而

CWG-GOA 的均值都优于其他三种算法,说明了柯西变异的有效性. 综上所述本文提出的 CWG-GOA 提高了基本蝗虫算法的求解精度.

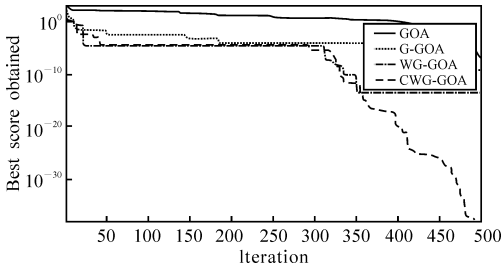
表 2 各算法测试结果

基准函数	GOA		G-GOA		WG-GOA		CWG-GOA	
	均值	标准差	均值	标准差	均值	标准差	均值	标准差
F_1	2.060 4e-08	1.992 9e-08	7.554 5e-13	4.039 e-28	3.773 3e-15	1.125 9e-14	6.070 6e-29	3.248 5e-28
F_2	0.000 293 98	0.001 347 8	8.31e-12	6.462 3e-27	7.446 2e-14	2.879 4e-13	2.393 8e-28	8.298 7e-28
F_3	0.001 927 6	0.009 409	3.887e-07	1.588 2e-22	1.222 6e-08	1.963 8e-08	9.232 1e-17	2.200 3e-16
F_4	9.830 2	5.797 1	1.691 6	0	1.335 7	0.193 23	0.961 79	0.178 6
F_5	0.756 04	0.961 85	1.183 5e-05	0	1.750 6e-06	2.585 2e-06	2.873 7e-07	6.589 5e-07
F_6	0.004 706 6	0.011 503	6.114 2e-09	3.308 7e-24	1.198e-08	5.325e-08	1.912 2e-09	7.215 7e-09

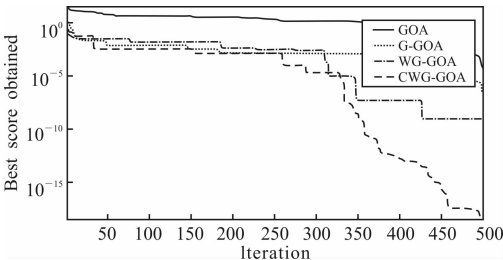
为验证 CWG-GOA 收敛速度,本文展示了各算法收敛曲线,如图 1 所示.



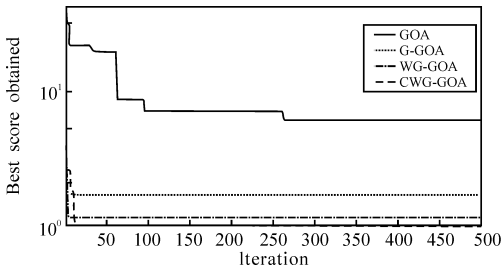
(a) F_1 函数的收敛曲线



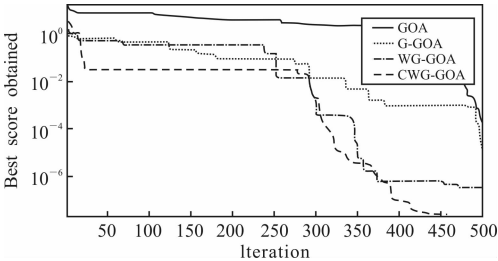
(b) F_2 函数的收敛曲线



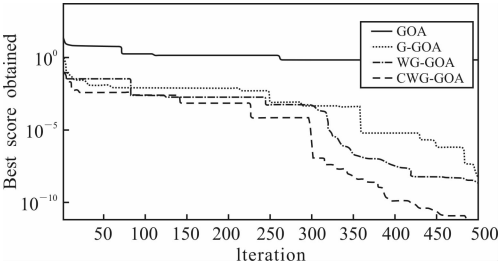
(c) F_3 函数的收敛曲线



(d) F_4 函数的收敛曲线



(e) F_5 函数的收敛曲线



(f) F_6 函数的收敛曲线

图 1 各算法在不同基准函数上的收敛曲线

由收敛曲线可以看出,迭代前期改进后的三种算法收敛速度基本持平,但优于基本蝗虫算法. 迭代后期,提出的 CWG-GOA 算法收敛速度加快,且能够跳出局部最优值,收敛精度提高了.

5 结束语

本文在标准蝗虫算法上采用佳点集初始化种群,增加了初始种群的均匀性,有利于更好地全局勘探. 将原始线性权重改进为非线性惯性权重,增强了算法勘探和开发能力. 柯西变异使算法易跳出局部最优解. 最终将提出的 CWG-GOA 算法和其他算法在 6 个基准函数上实验测试,测试结果表明所提出的 CWG-GOA 算法收敛精度高且收敛速度快.

参考文献:

[1] 王涛. Ryad Chellali. 非线性权重和收敛因子的鲸鱼算

- 法[J]. 微电子学与计算机, 2019, 36(1): 11-15.
- [2] SAREMI S, MIRJALILI S, LEWIS A. Grasshopper Optimisation Algorithm: Theory and application[J]. *Advances in Engineering Software*, 2017(105): 30-47.
- [3] MIRJALILI S Z, MIRJALILI S, SAREMI S, et al. Grasshopper optimization algorithm for multi-objective optimization problems[J]. *Applied Intelligence*, 2018, 48(4): 805-820.
- [4] ALJARAH I, ALA'M A Z, FARIS H, et al. Simultaneous feature selection and support vector machine optimization using the grasshopper optimization algorithm[J]. *Cognitive Computation*, 2018, 10(3): 478-495.
- [5] HEIDARI A A, FARIS H, ALJARAH I, et al. An efficient hybrid multilayer perceptron neural network with grasshopper optimization[J]. *Soft Computing*, 2019, 23(17): 7941-7958.
- [6] LUO J, CHEN H, XU Y, et al. An improved grasshopper optimization algorithm with application to financial stress prediction[J]. *Applied Mathematical Modelling*, 2018(64): 654-668.
- [7] 李洋州, 顾磊. 一种基于曲线自适应和模拟退火的蝗虫优化算法[J/OL]. *计算机应用研究*. <https://doi.org/10.19734/j.issn.1001-3695.2018.07.0580>.
- [8] ARORA S, ANAND P. Chaotic grasshopper optimization algorithm for global optimization[J]. *Neural Computing and Applications*, 2018: 1-21. DOI: 10.1007/s00521-018-3343-2.
- [9] EWEES A A, ELAZIZ M A, Houssein E H. Improved grasshopper optimization algorithm using opposition-based learning[J]. *Expert Systems with Applications*, 2018(112): 156-172.
- [10] 孙博文, 韦素媛. 基于自适应调整策略灰狼算法的 DV-Hop 定位算法[J]. *计算机科学*, 2019, 46(5): 7-82.
- [11] 郭振洲, 王平, 马云峰, 等. 基于自适应权重和柯西变异的鲸鱼优化算法[J]. *微电子学与计算机*, 2017, 34(9): 20-25.

作者简介:

王 涛 男, (1992-), 硕士, 助理研究员. 研究方向为图像处理、数据处理、深度学习. E-mail: 781036789@qq.com.