

文章编号: 2095-2163(2020)08-0051-07

中图分类号: TP301.6

文献标志码: A

精英反向学习带扰动因子的混沌蚁狮算法

王茜^{1,2}, 何庆^{1,2}, 林杰^{1,2}, 杨荣莹^{1,2}

(1 贵州大学 大数据与信息工程学院, 贵阳 550025; 2 贵州大学 贵州省公共大数据重点实验室, 贵阳 550025)

摘要: 针对蚁狮算法易陷入局部最优、收敛速度慢的缺点, 本文提出了基于精英反向学习带扰动因子的混沌蚁狮算法。该算法首先通过对蚂蚁的随机游走公式引入扰动因子, 有效提高了寻优精度, 避免算法陷入局部最优, 有效平衡了全局最优搜索与局部最优搜索; 对蚁狮的精英个体进行反向学习生成精英反向解, 增加了种群多样性; 通过比较当前最优解与精英反向解, 选出优异个体作为新的全局最优解; 最后在蚂蚁受轮盘赌选择蚁狮、精英蚁狮影响的随机游走公式中引入 Logistic 混沌映射, 有效提高了算法的全局搜索能力和寻优精度。通过 7 个经典测试函数实验对比, 就目前改进的蚁狮算法而言, 本文提出的算法能有效提高寻优的精度与收敛速度、高效解决函数优化问题。

关键词: 蚁狮算法; 扰动因子; 反向学习; 混沌映射; 函数优化

Chaos ant lion optimizer

based on elite opposition-based learning with perturbation factor

WANG Qian^{1,2}, HE Qing^{1,2}, LIN Jie^{1,2}, YANG Rongying^{1,2}

(1 College of Big Data & Information Engineering, Guizhou University, Guiyang 550025, China;

2 Guizhou Provincial Key Laboratory of Public Big Data, Guiyang 550025, China)

[Abstract] Aiming at the shortcomings of ant lion algorithm which is easy to fall into local optimum and slow convergence, this paper proposes a chaotic ant lion optimizer based on elite opposition-based learning with perturbation factor. First, the perturbation factor is introduced into the ant's random walk formula, effectively improving the precision of optimization, avoiding the algorithm falling into local optimum, balancing the global optimal search and the local optimal search. Later the algorithm generates the elite inverse solution by opposition-based learning the elite individual of the ant lion, which increases the diversity of the population and compares the current optimal solution with the elite to solve the superior individual as the new global optimal solution. Finally, the logistic chaotic map is introduced in the random walk formula of ant influenced by elite ant lion and roulette selected ant lion to improve the global search ability and optimization precision of the algorithm effectively. Through the comparison of seven classical test functions the algorithm proposed in this paper can effectively improve the precision and convergence speed of optimization, and can effectively solve the problem of function optimization.

[Key words] ant lion optimizer; perturbation factor; opposition-based learning; chaotic mapping; function optimization

0 引言

蚁狮算法 (Ant Lion Optimizer, ALO) 是由澳大利亚学者 Seyedali Mirjalili 于 2015 年提出的新型仿生群智能优化算法。该算法通过模拟蚂蚁、蚁狮在自然界的捕食行为关系而提出, 有着比遗传算法 (Genetic Algorithms, GA)、蝙蝠算法 (Bat Algorithm, BA)、花朵授粉算法 (Flower Pollination Algorithm, FPA) 等 7 个算法更好的寻优精度与收敛速度^[1]。群智能算法作为由自然生物启发而提出的新型算法, 已经得到越来越多的关注, 特别是早期典型的模

拟鸟群觅食过程提出的粒子群算法 (Particle Swarm Optimization, PSO)^[2]、根据蚂蚁在寻找食物过程中的路径行为提出的蚁群算法 (Ant Colony Optimization, ACO)^[3]、以及后来涌现出的萤火虫算法 (Firefly Algorithm, FA)^[4]、布谷鸟算法 (Cuckoo Search, CS)^[5] 和鲸鱼算法 (Whale Optimization Algorithm, WOA)^[6] 等新的群智能算法。近年来, 越来越多的群智能算法因其操作简单、参数设置少和易于实现的优点被提出, 并在工程领域得到广泛的应用, 蚁狮算法已被应用在优化 SVM 参数、电力系

基金项目: 贵州省科技计划项目重大专项项目 (黔科合重大专项字 [2018] 3002, 黔科合重大专项字 [2016] 3022); 贵州省公共大数据重点实验室开放课题 (2017BDKFJJ004); 贵州省教育厅青年科技人才成长项目 (黔科合 KY 字 [2016] 124)。

作者简介: 王茜 (1995-), 女, 硕士研究生, 主要研究方向: 优化算法、机器学习; 何庆 (1982-), 男, 博士, 副教授, 主要研究方向: 数据挖掘、大数据应用、人工智能; 林杰 (1995-), 女, 硕士研究生, 主要研究方向: 优化算法、机器学习、图像处理; 杨荣莹 (1995-), 女, 硕士研究生, 主要研究方向: 优化算法、NLP、机器学习。

通讯作者: 何庆 Email: qhe@ gzu.edu.cn

收稿日期: 2020-06-04

统、航迹规划、无线传感器^[7-10]等领域。

文献[11]利用改进蚁狮算法,有效的解决了无人机三维航迹规划的问题,但所使用的算法经过标准测试函数测试并为寻到最优值,其收敛精度还有待改进;文献[12]利用具有动态随机搜索和寻优半径连续收缩机制的改进蚁狮算法,虽然优化了模糊前件的结构参数,但是对于算法本身的实验结果未寻到最优值;文献[13]提出基于柯西变异的蚁狮算法,虽然提高了算法的收敛精度,但是收敛速度没有得到很好的改进;文献[14]提出了具有自适应边界与最优引导的莱维飞行改进算法,改进效果只与原始算法作对比且寻到的最优值不是理论最优值,收敛速度还有待提高。

本文针对蚁狮算法在寻优过程中易陷入局部最优、收敛速度慢的缺点,提出了一种基于精英反向学习扰动因子的混沌蚁狮算法(Chaos Ant Lion Optimizer Based on Elite Opposition-based Learning with Perturbation Factor, EOPCALO)。首先在蚂蚁的随机游走公式中引入扰动因子,让蚂蚁的活动范围发生由大到小的动态变化,在前期有效提高了全局寻优范围,后期在局部范围继续动态搜索,避免函数陷入局部最优,有效平衡了全局最优搜索与局部最优搜索;随后对蚁狮的精英个体进行反向学习生成精英反向解,通过比较当前最优解与精英反向解,选出优异个体作为新的全局最优解,用做修缮下一代新陷阱位置;最后在蚂蚁受轮盘赌选择蚁狮、精英蚁狮影响的随机游走公式中引入 Logistic 混沌映射,增加种群多样性,减小因轮盘赌选择蚁狮因为适应度不高而影响收敛效果的问题,有效提高了算法的全局搜索能力和寻优精度。通过7个测试函数的实验,与其它改进算法对比,有效证明了本文改进算法 EOPCALO 具有更好地寻优精度与更快地寻优速度。

1 原始蚁狮优化算法(ALO)

1.1 ALO 基本原理

蚁狮在沙土中利用下颚挖出不同大小的圆锥形坑,用来诱捕蚂蚁的。当蚂蚁进入陷阱后,蚁狮为防止蚂蚁逃走则向外刨出沙土使蚂蚁滑落到陷阱底部,随后拉入土壤里开始进食。

在实际问题中,蚂蚁代表候补最优解,它围绕着蚁狮陷阱进行随机游走,探索解空间的最优解,当蚂蚁搜索到比其围绕的蚁狮所在位置更优秀的解时,蚁狮将“吃掉”蚂蚁,通过不断的捕食蚂蚁,即取代蚂蚁所在的位置,从而实现解的更新。

ALO 算法的主要步骤以及数学表达公式如下:

步骤1 蚂蚁的随机游走

$$X_i = [0, \dots, \text{cumsum}(2r(t_n) - 1)], \quad (1)$$

$$r(t) = \begin{cases} 1, & \text{if rand} > 0.5; \\ 0, & \text{if rand} \leq 0.5. \end{cases} \quad (2)$$

其中, cumsum 为计算累积和; n 为最大迭代次数; t 为当前迭代次数, $r(t)$ 是一个随机函数; rand 是 $[0, 1]$ 均匀分布生成的随机数。

为了保证蚂蚁在搜索空间内随机游走,对式(1)进行标准化处理:

$$X_i^t = \frac{(X_i^t - a_i) \times (b_i - c_i)}{d_i - a_i} + c_i. \quad (3)$$

其中, a_i, b_i 分别为第 i 个变量的最小值和最大值; c_i^t, d_i^t 分别是第 t 次迭代第 i 个变量的最小值和最大值。蚂蚁的寻优过程中,用矩阵分别保存位置信息、适应度函数值;蚁狮隐藏在搜索空间的某个位置的信息与适应度函数值也存放于不同矩阵中。

步骤2 蚂蚁靠近蚁狮陷阱

用如下公式来表示蚂蚁游走的区域受到了蚁狮位置的影响:

$$c_i^t = \text{AntLion}_j^t + c^t, \quad (4)$$

$$d_i^t = \text{AntLion}_j^t + d^t. \quad (5)$$

其中, AntLion_j^t 是第 t 次迭代时,第 j 只蚁狮的位置; c_i^t, d_i^t 分别表示所有变量对于第 i 只蚂蚁的最小、最大值; c^t, d^t 分别表示第 t 次迭代时,所有变量的最小、最大值。

步骤3 蚂蚁滑落到陷阱底部

当蚂蚁掉入陷阱时,为防止蚂蚁逃走,蚁狮会继续向陷阱外刨出沙土,使得蚂蚁能掉入穴底,该过程可以看作是蚂蚁绕蚁狮随机游走的半径在逐渐减小,模拟了蚂蚁在陷阱内的滑动过程,公式为:

$$c^t = \frac{c^t}{I}, \quad (6)$$

$$d^t = \frac{d^t}{I}, \quad (7)$$

$$I = 10^{\omega} \frac{t}{T} \begin{cases} \omega = 2, & t > 0.1T; \\ \omega = 3, & t > 0.5T; \\ \omega = 4, & t > 0.75T; \\ \omega = 5, & t > 0.9T; \\ \omega = 6, & t > 0.95T. \end{cases} \quad (8)$$

其中, c^t, d^t 分别表示第 t 次迭代时,所有变量的最小、最大值; t 为当前迭代次数; T 为最大迭代次数; ω 是由当前迭代次数 t 和最大迭代次数 T 所决定的常数,可以调整开发的精度。

步骤 4 蚁狮重筑陷阱

蚂蚁掉入陷阱底部被蚁狮吃掉后,若蚂蚁种群中出现了适应值高于蚁狮的个体,则将蚂蚁的位置更新到蚁狮上,并重筑下一代的陷阱以增加捕捉猎物的机会:

$$AntLion_j^t = Ant_i^t, \text{if } f(Ant_i^t) > f(AntLion_j^t). \quad (9)$$

其中, $AntLion_j^t$ 是第 t 次迭代中第 j 只蚁狮的位置; Ant_i^t 表示第 t 次迭代中第 i 只蚂蚁的位置; f 为适应度函数; t 为当前迭代次数。

步骤 5 精英化

每次迭代完成后,将适应度最好的蚁狮作为精英蚁狮,该蚁狮能在迭代过程中影响所有蚂蚁的活动,蚂蚁的随机游走受轮盘赌选择蚁狮和精英蚁狮的影响:

$$Ant_i^t = \frac{R_A^t + R_E^t}{2}. \quad (10)$$

Ant_i^t 是第 t 次迭代中第 i 只蚂蚁的位置; R_A^t 是第 t 次迭代中围绕轮盘赌选择蚁狮的随机游走; R_E^t 是第 t 次迭代中围绕精英蚁狮的随机游走。

2 EOPCALO

2.1 加入扰动因子 q 的蚂蚁随机游走公式

在蚂蚁随机游走公式中添加扰动因子,增大了蚂蚁随机游走初期的游走范围,加强全局勘探能力,增加了搜索到全局最优值的概率。在算法后期勘探范围由大到小过渡,增强局部搜索能力,扰动因子 q 使得算法在全局搜索和局部搜索中有了一个好的平衡,并且有效的避免了陷入局部最优的问题,从而得到最优解。扰动因子 q 定义如下:

$$q = \left[\left(-\alpha \times \frac{t}{T} \right)^5 + 1 \right] \times \beta. \quad (11)$$

其中, t 是当前迭代次数; T 为最大迭代次数。通过大量实验证明, $\alpha = 10, \beta = 0.15$ 时寻优效果最好,改进后的蚂蚁随机游走公式为:

$$X_i^t = \left[\frac{(X_i^t - a_i) \times (b_i - c_i)}{d_i^t - a_i} + c_i \right] \times q. \quad (12)$$

2.2 Logistic 混沌映射

由于蚁狮种群的适应度在算法迭代过程中仍然存在着相对较差的个体,而通过轮盘赌可能会选择到适应度较差的蚁狮。若蚂蚁围绕该蚁狮进行游走会影响算法整体的寻优效率,将 Logistic 混沌映射与蚂蚁受轮盘赌选择蚁狮与精英蚁狮影响的随机游走公式相结合,使蚂蚁能够进行混沌搜索,提高蚂蚁的寻优效率,增大算法跳出局部最优解的能力。

Logistic 混沌映射如下:

$$x_n \in \text{rand}(1), \quad (13)$$

$$x_{n+1} = 4 \times x_n (1 - x_n). \quad (14)$$

其中, x_n 是 0 到 1 中的一个随机数,在公式 (10) 中加入混沌因子,有效提高了算法的寻优效率,并将本次计算的蚂蚁位置作为下一代的初始位置,改进后的公式为:

$$Ant_i^t = \left(\frac{R_A^t + R_E^t}{2} \right) \times x_{n+1}. \quad (15)$$

2.3 精英蚁狮的反向学习

反向学习 (opposition-based learning, OBL) 由 Tizhoosh^[15] 教授于 2005 年提出,主要是求问题可行解的反向解,并对当前可行解与反向解进行评估,从中选择较优的解作为下一代个体。反向学习能有效提高群体多样性,从中选择较好的解作为当前最优解,大大提高了算法的效率。本文合理利用精英蚁狮个体的有效信息,通过对精英个体的反向学习产生精英反向解,增加了种群多样性,加快了算法的收敛速度。

定义 1 反向数 Opposition Number^[16]: 若 $x \in [a, b]$ 且 $x \in R$, 则其反向数 x^* 为:

$$x^* = a + b - x. \quad (16)$$

定义 2 反向点 Opposition Point^[16]: 若 $p = (x_1, x_2, \dots, x_N)$ 之间的任意 N 维空间中的点为 $x \in [x_{\min}, x_{\max}]$, $x_i \in [a_i, b_i]$, 则它反向学习的解如下所示^[17]:

$$\begin{cases} P^* = (x_1^*, x_2^*, \dots, x_D^*), \\ x_i^* = x_{\min} + x_{\max} - x_i. \end{cases} \quad (17)$$

定义 3 精英反向学习 (elite opposition-based learning, EOBL)^[17]。

本文中蚂蚁与蚁狮种群的整体适应度中的精英个体赋值给精英蚁狮,定义为 $DF_i = (DF_1, DF_2, \dots, DF_T)$ 。其中 $DF_i \in [a_i, b_i]$, $[a_i, b_i]$ 为蚂蚁与蚁狮按适应度排序后的最小最大值。则对应的精英反向解 $DF_i^* = (DF_1^*, DF_2^*, \dots, DF_T^*)$ 定义为:

$$DF_i^* = a_i + b_i - DF_i. \quad (18)$$

将式 (9) 中最优的适应度与精英蚁狮、精英反向解作比较,选出适应度最好的作为当前最优值。

2.4 EOPCALO 算法流程

改进的蚁狮算法 EOPCALO 流程如下:

(1) 参数设置,设置好种群规模、维数、迭代次数与搜索空间上下限等参数。

(2)在搜索空间范围内随机初始化蚂蚁与蚁狮的位置并计算它们的适应度。

(3)组合蚂蚁与蚁狮适应度,整体降序排列后,选取适应度最好的值赋给精英蚁狮。

(4)通过轮盘赌方式选择蚁狮,利用公式(4)~(8)更新 c_i^t 与 d_i^t 的值,通过式(1)、(2)、(12)更新蚂蚁随机游走值,完成蚂蚁掉落陷阱底部被蚁狮捕捉的过程。

(5)蚂蚁被蚁狮捕捉后,利用式(9)对比被捕蚂蚁与精英蚁狮的适应度。将当前适应度最好的值更新到精英蚁狮上,重新调整蚁狮位置,并更新最优值。

(6)通过式(18)对精英蚁狮进行精英反向学习,对比精英反向解与精英蚁狮的适应度值,择优选择为当前最好的精英蚁狮。更新精英蚁狮位置信息与适应度值。再利用式(15)更新蚂蚁的位置。

(7)判断该结果否达到了算法的最大迭代次数,若达到了理论最优适应度,则获得最佳的蚁狮位置,并输出适应度值,否则转至步骤3)。

改进算法流程如图1所示。

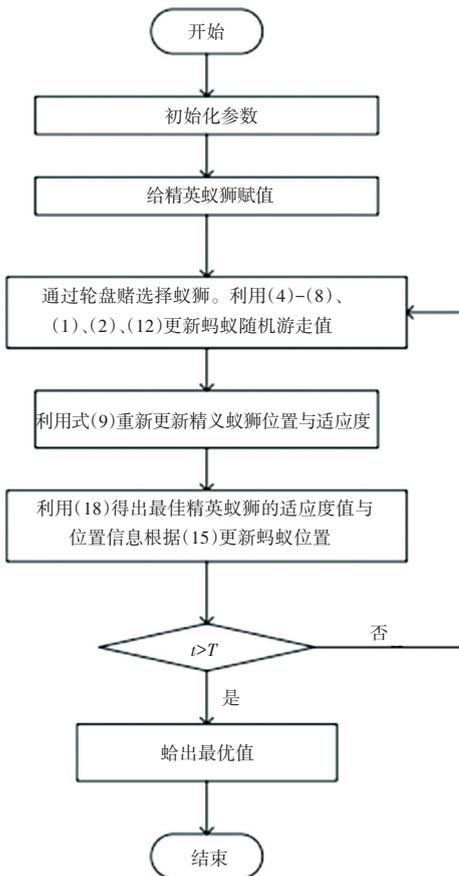


图1 EOPCALO流程图

Fig. 1 The flow chart of EOPCALO

3 实验仿真与结果分析

3.1 测试函数与参数设置

本次实验选取7个具有不同寻优特征的经典标准测试函数,来测试本文改进算法EOPCALO的寻优能力。标准测试函数见表1。其中 $F_1 \sim F_4$ 为单峰函数, $F_5 \sim F_7$ 为带有许多局部极值的多峰函数,求解难度较高。在本次实验中运用多峰函数测试改进算法在高维条件下的寻优速度与寻优精度。选择原始蚁狮算法(ALO)、鲸鱼算法(WOA)、改进的蚁狮算法(IALO)^[11]、自适应边界与最优引导的莱维飞行的蚁狮算法(ABLALO)^[14]、基于反向的拉普拉斯蚁狮算法(OB-L-ALO)^[18]、具有Levy变异和精英自适应竞争机制的蚁狮算法(LEALO)^[19]共6个算法,与本文改进的EOPCALO算法在参数设置统一的条件下进行测试。通过最优值(Best)、平均值(Mean)与标准差(Std)3个指标,比较改进算法EOPCALO与ALO、WOA、IALO、ABLALO、OB-L-ALO、LEALO算法在30维条件下的寻优能力;选取ALO、WOA、IALO、EOPCALO这4个算法在100维上进行对比实验。实验结果表明,不管在低维还是高维条件下都具有更好的寻优精度、更快的寻优速度与良好的稳定性。

算法在MATLAB 2014b上进行实验,运行环境为64位Window7操作系统。为保证实验的公平性,算法中种群规模都设为 $N = 40$,最大迭代次数 $\text{Max_iteration} = 500$ 、 $\alpha = 10$ 、 $\beta = 0.15$,对每个函数进行30次独立实验,为减少实验中的随机性,分别记录最优值(Best)、平均值(Mean)与标准差(Std)3个参数,整体评价各算法的性能。其中,平均值反映出算法的寻优精度,标准差反映出算法的鲁棒性与稳定性。

3.2 实验结果与对比分析

由表2可以看出,在30维、500次迭代的情况下,本文提出的基于精英反向学习带扰动因子的混沌蚁狮算法,有着比其它算法更好的寻优性能,体现了算法良好的鲁棒性。

对于 $F_1 \sim F_2$,EOPCALO寻优成功率高达100%。虽然OB-L-ALO也寻到最优值,但是从平均值与标准差来看,EOPCALO具有更好的稳定性。本文算法整体性能相对于其它算法最好,OB-L-ALO次之,WOA与ABLALO也有着不错的寻优效果,其次是LEALO,最后是IALO与ALO。

表 1 标准测试函数
Tab. 1 Standard test function

函数名	表达式	维度	范围	最优值
Sphere	$F_1(x) = \sum_{i=1}^D x_i^2$	30,100	[100,100]	0
Schwefel 2.22	$F_2(x) = \sum_{i=1}^D x_i + \prod_{i=1}^D x_i $	30,100	[10,10]	0
Rotated hyper-ellipsoid	$F_3(x) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2$	30,100	[100,100]	0
Schwefel 2.21	$F_4(x) = \max_i \{ x_i , 1 \leq i \leq D \}$	30,100	[-100,100]	0
Rastrigin	$F_5(x) = \sum_{i=1}^D [x_i^2 - 10\cos(2\pi x_i) + 10]$	30,100	[-5.12,5.12]	0
Ackley	$F_6(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$	30,100	[-32,32]	0
Griewank	$F_7(x) = \frac{1}{4000}\sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30,100	[-600,600]	0

对于 $F_3 \sim F_4$, EOPCALO 寻优成功率依然是 100%, 其次是 OB-L-ALO, 与 $F_1 \sim F_2$ 相似, 虽然寻到了理论最优值, 但是其平均值与标准差不够稳定; IALO 与 ALO 对于 F_3 的寻优效果稍优于 WOA, 但是在 F_4 上 WOA 略优于 IALO、ALO, 整体对比来看 EOPCALO 寻优能力依然最强。

对于 F_5 , EOPCALO、OB-L-ALO、ABLALO、WOA 都寻到了理论最优值, 并且 ABLALO 有着与 EOPCALO 相同的平均值与标准差值, 体现了两算法对于 F_5 具有良好的寻优能力, WOA 与 OB-L-ALO 次之, IALO 与 ALO 最差。

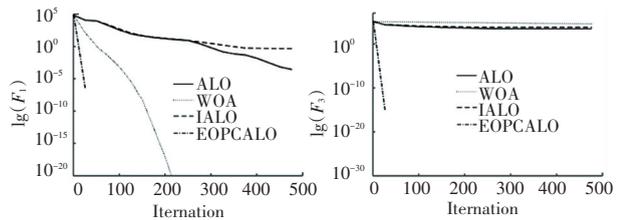
对于 F_6 , 所测算法均陷入了局部最优值, EOPCALO、OB-L-ALO、WOA 所寻得的最优值相同。通过平均值对比发现, EOPCALO 即使陷入局部最优, 平均值也优于其它算法。

对于 F_7 , EOPCALO、OB-L-ALO、ABLALO、WOA 都寻到了理论最优值。其中 EOPCALO 寻优成功率为 100%, ABLALO 平均值稍优于 OA-L-ALO, 其次是 WOA, IALO 与 ALO 寻优效果最差。

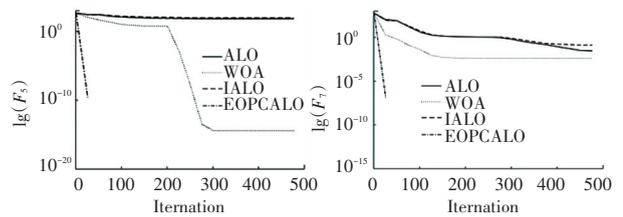
总体来看, 不管对于单峰函数还是多峰函数, EOPCALO 都具有良好的寻优精度与较快的寻优速度, 而且具有良好的稳定性。

图 2 给出了 30 维条件下 ALO、WOA、IALO、EOPCALO 4 个算法在函数 $F_1、F_3、F_5、F_7$ 中 500 次迭代的收敛曲线图, 从图中看出 EOPCALO 收敛速

度最快, 能在最短的时间内寻到了理论最优值。ALO、IALO 在图中都陷入了局部最优, WOA 相对于 ALO、IALO 来说寻优精度更好, 但也容易陷入局部最优。总体来说 EOPCALO 对于测试函数都表现出了最好的寻优精度、寻优速度与稳定性, 有效的增强了算法的搜索性能。



(a) F_1 收敛曲线 (a) F_1 convergence curve
(b) F_3 收敛曲线 (b) F_3 convergence curve



(c) F_5 收敛曲线 (c) F_5 convergence curve
(d) F_7 收敛曲线 (d) F_7 convergence curve

图 2 ALO/WOA/IALO/EOPCALO 收敛曲线 (30 dim)
Fig. 2 ALO/WOA/IALO/EOPCALO convergence curve (30 dim)

3.3 高维度下算法性能测试

由表3可以看出,对于测试函数 $F_1 \sim F_7$ (除 F_6), EOPCALO在100维、500次迭代的情况下都寻到了理论最优值,且平均值与方差都稳定于最优值; WOA寻到了 F_5, F_7 的理论最优值,寻优性能其次; IALO与ALO寻优效果最差。

表2、表3对比可看出, EOPCALO不管是在30维还是100维条件下,都具有良好且稳定的寻优性能。WOA在100维条件下对于函数 $F_1, F_2, F_3, F_4, F_6, F_7$ 的寻优精度与30维条件下的寻优精度值相对持平,准确寻到了 F_5 的理论最优值,且平均值、标准差比30维条件下的值更好; 100维条件下的IALO、ALO对于函数 $F_1 \sim F_7$ 的寻优精度都低于30维条件下的寻优精度值,说明算法在高维条件下的寻优能力很差。总体看来 EOPCALO不管是在30维还是200维的高维条件下,依然具有良好的收敛精度与鲁棒性。

图3给出了100维条件下ALO、WOA、IALO、EOPCALO4个算法在函数 F_1, F_3, F_5, F_7 中500次迭代的收敛图。从图中可以看出, EOPCALO在高维条件下仍然具有良好的寻优能力与较好的鲁棒性,都在最短的时间内寻到了最优值, IALO与ALO再次陷入局部最优, WOA的寻优性能比IALO和ALO好,但是除了 F_5 ,对于其它测试函数最后也未能寻得最优值。

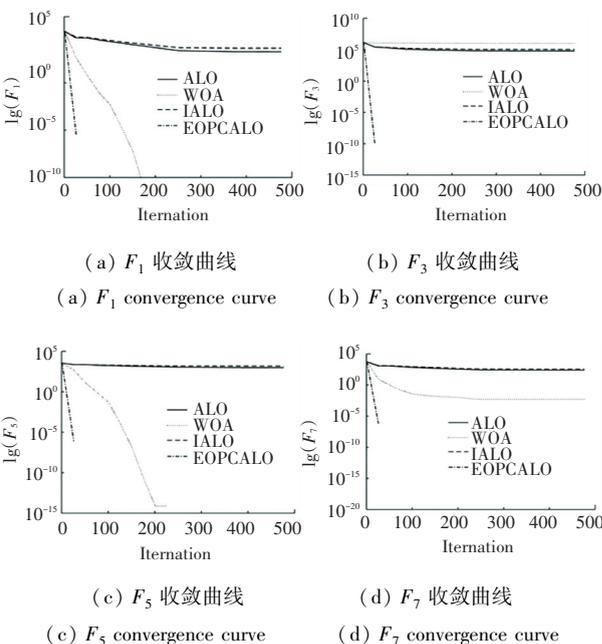


图3 ALO/WOA/IALO/EOPCALO收敛曲线(100 dim)

Fig. 3 ALO/WOA/IALO/EOPCALO convergence curve (100 dim)

是100维寻优能力与收敛曲线趋势都稳定,都在最快的时间里寻得了最优值; WOA对于高维也具有较好的适应性,收敛趋势差别不大; IALO、ALO在高维情况下更快的陷入了局部最优且寻优精度更差。总体来看, EOPCALO在高维条件下所表现出的稳定寻优能力,能够有效解决高维复杂的函数优化问题。

表2 函数测试实验结果(30 dim)

Tab. 2 Function test experimental results (30 dim)

函数	算法	30 dim		
		Best	Mean	Std
F_1	ALO	7.11E-05	2.55E-04	1.57E-04
	WOA	2.90E-91	9.33E-79	4.43E-73
	IALO	1.96E-01	4.85E-01	3.05E-01
	ABLALO ^[14]	4.45E-18	1.91E-17	7.40E-18
	OB-L-ALO ^[18]	0.00E+00	1.01E-09	4.63E-09
	LEALO ^[19]	1.87E-11	5.89E-10	5.64E-10
	EOPCALO	0.00E+00	0.00E+00	0.00E+00
F_2	ALO	1.80E+00	5.33E+01	4.58E+01
	WOA	7.45E-59	1.20E-52	3.29E-52
	IALO	1.49E+01	1.01E+02	1.64E+02
	ABLALO ^[14]	4.10E-17	6.97E-17	1.41E-17
	OB-L-ALO ^[18]	0.00E+00	5.98E-05	4.26E-05
	LEALO ^[19]	5.45E-08	1.70E-05	2.78E-05
	EOPCALO	0.00E+00	0.00E+00	0.00E+00
F_3	ALO	4.81E+02	2.83E+03	1.18E+03
	WOA	1.70E+04	3.71E+04	1.16E+03
	IALO	2.24E+03	6.01E+03	2.54E+03
	OB-L-ALO ^[18]	0.00E+00	1.16E-29	5.95E-29
	EOPCALO	0.00E+00	0.00E+00	0.00E+00
F_4	ALO	8.01E+00	1.47E+01	2.93E+00
	WOA	7.80E-03	4.30E+01	3.04E+01
	IALO	8.99E+00	1.54E+01	4.30E+00
	OB-L-ALO ^[18]	0.00E+00	1.09E-05	2.29E-05
	EOPCALO	0.00E+00	0.00E+00	0.00E+00
F_5	ALO	2.98E+01	7.31E+01	2.31E+01
	WOA	0.00E+00	3.79E-15	1.42E-14
	IALO	4.37E+01	9.14E+01	1.97E+01
	ABLALO ^[14]	0.00E+00	0.00E+00	0.00E+00
	OB-L-ALO ^[18]	0.00E+00	3.01E-09	4.82E-09
EOPCALO	0.00E+00	0.00E+00	0.00E+00	
F_6	ALO	0.93E+00	2.85E+00	1.49E+00
	WOA	8.88E-16	4.09E-15	2.16E-15
	IALO	2.66E+00	5.07E+00	2.55E+00
	OB-L-ALO ^[18]	8.88E-16	1.54E-05	8.10E-06
EOPCALO	8.88E-16	8.88E-16	0.00E+00	
F_7	ALO	2.30E-03	3.00E-02	1.37E-02
	WOA	0.00E+00	3.89E-03	2.16E-02
	IALO	8.00E-02	1.34E-01	3.42E-02
	ABLALO ^[14]	0.00E+00	1.51E-15	7.39E-15
	OB-L-ALO ^[18]	0.00E+00	6.32E-09	1.27E-08
	LEALO ^[19]	4.77E-09	9.30E-03	2.09E-02
	EOPCALO	0.00E+00	0.00E+00	0.00E+00

结合图2、图3来看, EOPCALO不管是30维还

表 3 函数测试实验结果(100 dim)

Tab. 3 Function test experimental results(100 dim)

函数	算法	100 dim		
		Best	Mean	Std
F_1	ALO	5.74E+02	1.93E+03	9.44E+02
	WOA	1.91E-88	1.00E-78	4.38E-78
	IALO	1.98E+03	4.64E+03	2.02E+03
	EOPCALO	0.00E+00	0.00E+00	0.00E+00
F_2	ALO	5.32E+01	2.66E+16	1.46E+17
	WOA	3.43E-57	9.25E-51	4.84E-50
	IALO	2.86E+02	5.80E+33	3.17E+34
	EOPCALO	0.00E+00	0.00E+00	0.00E+00
F_3	ALO	4.36E+04	6.50E+04	1.53E+04
	WOA	5.38E+05	9.87E+05	2.38E+05
	IALO	4.58E+04	1.10E+05	3.62E+04
	EOPCALO	0.00E+00	0.00E+00	0.00E+00
F_4	ALO	2.36E+01	3.06E+01	3.83E+00
	WOA	2.71E+00	6.42E+01	2.85E+01
	IALO	2.63E+01	3.33E+01	4.23E+00
	EOPCALO	0.00E+00	0.00E+00	0.00E+00
F_5	ALO	8.09E+02	9.74E+02	1.03E+02
	WOA	0.00E+00	0.00E+00	0.00E+00
	IALO	1.24E+03	1.47E+03	1.20E+02
	EOPCALO	0.00E+00	0.00E+00	0.00E+00
F_6	ALO	1.30E+01	1.45E+01	7.36E-01
	WOA	8.88E-16	4.68E-15	2.46E-15
	IALO	1.25E+01	1.48E+01	7.98E-01
	EOPCALO	8.88E-16	8.88E-16	0.00E+00
F_7	ALO	1.31E+02	2.42E+02	6.20E+01
	WOA	0.00E+00	4.80E-03	2.63E-02
	IALO	2.46E+02	3.54E+02	8.52E+01
	EOPCALO	0.00E+00	0.00E+00	0.00E+00

4 结束语

本文提出的基于精英反向学习的带扰动因子的混沌蚁狮算法(EOPCALO),通过对蚂蚁的随机游走公式加入动态变化的扰动因子,使得蚂蚁能在前期扩大搜索范围,避免算法过早陷入局部最优,后期对局部的搜索能更好地寻到最优值,加快收敛速度;再利用精英蚁狮的特性对其求反向解,增大了种群多样性,提高了算法的收敛精度;最后对随机游走受轮盘赌选择蚁狮与精英蚁狮影响的蚂蚁加入 Logistic

混沌,提高了算法的收敛精度。最后通过与 6 个不同的算法在同一条件下进行实验得出,EOPCALO 算法在解的精度、收敛速度以及鲁棒性上都表现出的良好的性能。

参考文献

- [1] MIRJALILI S. The ant lion optimizer [J]. Advances in Engineering Software, 2015, 83 (1): 80-98.
- [2] KENNEDY J, EBERHART R C. Particle swarm optimization [C]// Proc of IEEE ICNN. Piscataway: IEEE Press, 1995: 1942-1948.
- [3] COLORNI A, Dorigo Mand Maniezzo V. Distributed optimization by ant colonies[A] . In-Proc.of 1st European Conf . Artificial Life [C]//Pans, France; Elsevier, 1991; 134-142.
- [4] YANG X S. Nature-inspired metaheuristic algorithm [M].Luniver Press,2008; 81-96.
- [5] YANG X S, DEB S. Cuckoo Search via Levy flights [C]// Proc of World Congress on Nature & Biologically Inspired Computing. Piscataway: IEEE Publications, 2009: 210-214.
- [6] MIRJALILI S, LEWIS A. The whale optimization algorithm [J]. Advances in Engineering Software, 2016, 95: 51-67.
- [7] 赵世杰,高雷阜,于冬梅,徒君.带混沌侦查机制的蚁狮优化算法优化 SVM 参数[J].计算机科学与探索,2016,10(5):722-731.
- [8] 何忠华.基于蚁狮优化 GM(1,1)模型的北京市电力需求预测[J].现代工业经济和信息化,2017,7(22):57-60,83.
- [9] 赵克新,黄长强,王渊,周瑞.基于混沌蚁狮算法的无人机航迹规划[J].飞行力学,2018,36(1):93-96.
- [10]徐钦帅,何庆,魏康园.改进蚁狮算法的无线传感器网络覆盖优化[J].传感技术学报,2019,32(2):266-275.
- [11]黄长强,赵克新.基于改进蚁狮算法的无人机三维航迹规划[J].电子与信息学报,2018,40(7):1532-1538.
- [12]赵小国,刘丁,景坤雷.融合改进蚁狮算法和 T-S 模糊模型的噪声非线性系统辨识[J].控制与决策,2019,34(4):759-766.
- [13]于建芳,刘升,韩斐斐,等.基于柯西变异的蚁狮优化算法[J].微电子学与计算机,2019,36(6):45-49,54.
- [14]王若安,周越文,韩博,李剑峰,等.具有自适应边界与最优引导的莱维飞行蚁狮优化算法[J].微电子学与计算机,2018,35(9):20-25,31.
- [15]TIZHOOSH H R. Opposition-based learning: a new scheme for machine intelligence [C]// Proc of Computational Intelligence for Modelling. Vienna, Austria: Computer Society, 2005, 1: 695-701.
- [16] RAHNAMAYAN S, TIZHOOSH H R, SALAMA M M. Opposition-based differential evolutions [J]. .IEEE Transactions on Evolutionary Computation, 2008, 12(1):64-79.
- [17]AHANDANI M A, ALAVI-RAD H. Opposition-based learning in shuffled frog leaping: an application for parameter identification [J]. Information Sciences, 2015, 291(291): 19-42.
- [18]Shail Kumar Dinkar, Kusum Deep. Opposition based Laplacian Ant Lion Optimizer[J]. Journal of Computational Science,2017, 23.
- [19]景坤雷,赵小国,张新雨,等.具有 Levy 变异和精英自适应竞争机制的蚁狮优化算法[J].智能系统学报,2018,13(2):236-242.