

融合莱维飞行与黄金正弦的蚁狮优化算法^{*}

于建芳¹, 刘升¹, 王俊杰², 鲁晓艺¹

(1. 上海工程技术大学 管理学院, 上海 201620; 2. 东华大学 信息科学与技术学院, 上海 201620)

摘要: 针对蚁狮优化算法较易陷入局部最优停滞, 收敛精度低以及收敛速度较慢等问题, 将莱维飞行机制和黄金正弦算法融合到蚁狮优化算法中, 提出了融合莱维飞行与黄金正弦的蚁狮优化算法(LGSALO)。该算法利用 Levy 飞行的变异机制对寻优过程中位置更新方式进行变异操作, 可以改善种群多样性, 使得算法跳出局部最优, 提高全局寻优能力, 并在一定程度上避免了算法的过早收敛; 同时引入黄金正弦算法改进精英蚁狮的寻优方式, 协调算法的全局探索与局部开发能力; 实验仿真结果表明该改进算法的寻优性能良好, 开发能力强。

关键词: 蚁狮优化算法; 黄金正弦; 莱维飞行; 函数优化

中图分类号: TP301.6 doi: 10.19734/j.issn.1001-3695.2018.11.0959

Antlion optimization algorithm integrating with Levy flight and golden sine

Yu Jianfang¹, Liu Sheng¹, Wang Junjie², Lu Xiaoyi¹

(1. College of Management, Shanghai University of Engineering Sciences, Shanghai 201620, China; 2. College of Information Sciences & Technology, Donghua University, Shanghai 201620, China)

Abstract: To solve the problem that the ant lion optimization algorithm is prone to local optimal stagnation, low convergence accuracy and slow convergence speed, this paper proposed the antlion optimization algorithm integrating with levy flight and golden sinusoids(LGSALO). This algorithm used the variation mechanism of Levy flight to carry out mutation operation on the position update mode in the optimization process, which could improve the population diversity, make the algorithm jump out of the local optimum, improve the global optimization ability, and avoid the premature convergence of the algorithm to some extent. At the same time, the algorithm introduced the golden sine algorithm to improve the optimization mode of the elite antlion and coordinate the ability of global exploration and local development. The experimental simulation results show that the improved algorithm has good performance of optimization and strong development ability.

Key words: antlion optimization algorithm; gold sine; Levy flight; function optimization

0 引言

蚁狮优化算法(ALO)^[1]是一种新的启发式搜索算法, 由澳大利亚教授 Mirjalili 于 2015 年提出。该算法的仿真原理是模仿蚁狮幼虫捕食蚂蚁的行为实现对实际的目标问题的求解。蚁狮算法是众多群智能优化算法的一个比较新的算法, 具有原理简单, 改动参数少, 易于实现等特点。

近年来, 元启发式算法越来越多的被应用于实际工程设计优化问题中求解最优解等问题。在国外, 蚁狮算法一经提出就被应用于杆系结构优化^[2]、在复杂网络中进行社区挖掘^[3]、无用功的配电问题^[4]、对线性的离散滤波器进行高效的建模^[5]以及无人机航线规划的问题^[6]等领域。在国内, 张振兴等人^[7]利用混沌算子优化蚁狮算法, 能够快速准确的得到一条满足实验要求的航迹, 但是对于函数测试的对比算法只有原蚁狮算法 ALO, 对比结果缺乏说服力。赵世杰等人^[8]引入了混沌算子机制优化 SVM 参数, 提出了基于混沌侦察机制的蚁狮优化算法(CIALO), 提高了种群的总体适应值, 但未改善算法的收敛速度。赵小国等人^[9]利用融合改进的蚁狮优化算法实现了系统辨识中噪声影响的有效抑制, 但是对于基准函数测试的结果并未达到理论最优值, 需要进一步的改进。景坤雷等人^[10]提出采用 Levy 变异和精英竞争机制的蚁狮优化算法(LEALO), 并应用于硅单晶热场温度模型的参数

辨识, 取得了很好的优化效果, 但是测试结果并未达到理论最优, 还有很大改进空间。

针对以上蚁狮算法存在的容易陷入局部最优和收敛精度低的缺陷和实际应用中对于精度的要求, 提出了融合莱维飞行和黄金正弦的蚁狮优化算法, 对 12 个标准函数进行了测试, 莱维飞行机制中短距离步长与长距离跳跃的结合使得蚁狮个体受局部极值点约束力下降, 能够快速跳出局部最优, 大大提高了全局搜索能力; 利用黄金正弦算法改变精英蚁狮的位置更新方式, 提高蚁狮的寻优能力和收敛速度, 对于蚁狮算法应用于实际工程中具有很重要的现实意义。

1 基本蚁狮优化算法

蚁狮优化算法是根据蚁狮幼虫诱捕蚂蚁的捕猎机制原理而提出的一种智能算法。蚁狮狩猎时实施的 5 个主要步骤, 即蚂蚁的随机游走, 建造陷阱, 诱捕蚂蚁进入陷阱, 捕捉猎物, 以及重新建造陷阱。狩猎方式如下:

蚁狮幼虫会在沙地上挖出不同大小锥形的坑。在挖出陷阱后, 藏在陷阱中心的下方等待蚂蚁被困在坑里, 当猎物被抓住时马上就会被拉到土壤里吃掉。在吃完猎物后, 蚁狮会把剩下的食物扔到坑外, 然后再去重新构建陷阱, 寻找下一个猎物。

蚁狮优化算法模仿了蚁狮捕猎蚂蚁的聪明行为。构建模

收稿日期: 2018-11-16; 修回日期: 2019-01-12 基金项目: 国家自然科学基金资助项目(61075115, 61673258)

作者简介: 于建芳(1992-), 女, 河南周口人, 硕士研究生, 主要研究方向为智能计算、项目调度与优化; 刘升(1966-), 男(通信作者), 湖北黄石人, 教授, 博士, 主要研究方向为智能计算、机器学习(ls6601@sina.com); 王俊杰(1992-), 男, 河南周口人, 硕士, 主要研究方向为智能计算, 电力系统; 鲁晓艺(1994-), 女, 湖北襄阳人, 硕士, 主要研究方向为智能算法、智能审计。

型寻求最优解时, 本文有以下 7 个假设条件:

- a) 蚂蚁以不同的随机游走步长在蚁狮的搜索空间中移动, 其游走行为受陷阱范围的限制。
- b) 随机游走适用于蚂蚁的所有维度。
- c) 蚁狮可以与它们的适应度成正比(适应性越高, 锥形陷阱半径越大, 越容易捕获蚂蚁)。
- d) 每一个蚂蚁都可以在每一次迭代中被一个精英蚁狮捕获。
- e) 随机游走的范围可以自适应地减少, 以模拟蚂蚁滑向坑底的行为。
- f) 如果蚂蚁比蚁狮的适应值更高, 意味着它被蚁狮捕获。
- g) 蚁狮在捕猎之后会把自己重新定位到捕获的猎物的位置, 并在寻找下一个猎物的过程中, 建立一个适应值更高的陷阱来改善它的捕猎机制。

蚂蚁在寻找食物时的随机游走公式如下:

$$X(t) = [0, \text{cumsum}(2r(t_1) - 1), \text{cumsum}(2r(t_2) - 1), \dots, \text{cumsum}(2r(t_n) - 1)] \quad (1)$$

其中: cumsum 为蚂蚁游走步长计算数组累加和, n 是迭代的最大数量, t 为目前的迭代次数, $r(t)$ 是一个随机函数:

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

其中 rand 是在 [0,1] 内均匀分布的随机数。

在算法优化过程中, 蚁狮和蚂蚁的位置被保存在位置矩阵里并被利用求得各自的适应值, 把求得的适应值保存在适应值函数矩阵里, 记录最优值。

蚂蚁在优化的每一步中都以随机游走的方式更新它们的位置。蚂蚁的位置更新公式如下:

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

其中: x_i^t 为第 i 只蚂蚁在第 t 次迭代后的位置, a_i 和 b_i 分别为第 i 维变量随意游走的最小值和最大值, c_i^t 和 d_i^t 分别为第 i 个变量第 t 次迭代的最小值和最大值。

随机游走的蚂蚁受到了蚁狮陷阱范围大小的影响。对这个假设进行了数学建模如下:

$$c_i^t = AL_i + c^t \quad (4)$$

$$d_i^t = AL_i + d^t \quad (5)$$

其中: c^t 是在第 t 次迭代中所有变量的最小值, d^t 是在包括 t 次迭代中所有变量的最大值, AL_i 显示了在第 t 次迭代中选择的第 i 个蚁狮的位置。

方程式(4)和(5)表明, 蚂蚁随机地在一个被选定的蚁狮周围的向量 c 和 d 所定义的超球体中行走。图 1 说明了这种随机游走行为的概念模型。

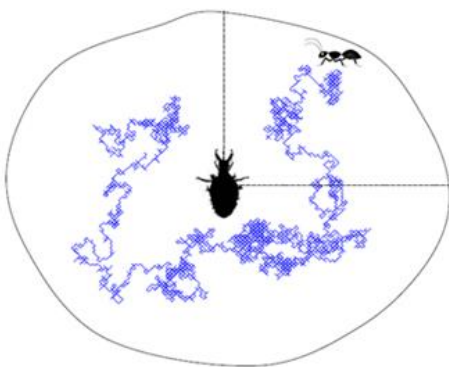


图 1 蚂蚁在蚁狮陷阱里的随机游走轨迹图

Fig. 1 A random pattern of ants in an ant-lion trap

对于蚁狮的捕猎机制, 它们能够建立与适应性成比例的陷阱, 适应性越大, 陷阱的范围就越大。蚂蚁在陷阱范围内

随机移动。然而, 当蚁群发现一只蚂蚁落在陷阱里时, 它们就会集体向陷阱的中心发射沙子。这种行为可以帮助滑下被困在陷阱的蚂蚁逃跑。而这种行为会使陷阱的范围半径自适应的变小。与此同时, 处在陷阱中心的蚁狮发现有蚂蚁落在陷阱里时, 就会从陷阱中心向陷阱边缘投射沙子, 以使蚂蚁向陷阱中心滑落。有如下陷阱范围变化的方程式:

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

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

其中: $I=10^{w(t/T)}$ 是蚁狮构建陷阱的范围大小, w 是一个常数, T 表示最大的迭代次数。

狩猎的最后阶段是当蚁狮抓到蚂蚁后, 吃掉它的身体。然后, 蚁狮会将它的位置更新到被猎杀的蚂蚁的最新位置, 以增加捕获新猎物的机会。蚁狮的位置更新方程式如下:

$$AL_i = Ant_i^t, \text{ if } f(Ant_i^t) > f(AL_i) \quad (8)$$

其中: Ant_i^t 表示第 t 次迭代的第 i 个蚂蚁的位置。

在每次迭代中都采用轮盘赌的搜索方法将获得的最好的蚁狮适应值保存为精英蚁狮。由于精英是适应值最强者, 所以它能够在迭代过程中影响所有蚂蚁的游走。因此, 假设每只蚂蚁随机地在轮盘赌精英蚁狮范围里随机游走, 公式为

$$Ant_i^t = \frac{R_A + R_E}{2} \quad (9)$$

其中: R_A 是在第 t 次迭代中由轮盘赌选择的蚂蚁的随机游走的步长, R_E 是在第 t 次迭代中随机游走于精英蚁狮圈的蚂蚁的步长。

ALO 算法步骤如下:

- a) 设置参数。最大迭代次数 T 、蚂蚁和蚁狮的数目 $NumB$ 、 $NumY$ 、适应度函数维数 dim 以及变量范围 u 和 l 。
- b) 随机初始化蚁狮和蚂蚁的位置, 计算各相应的适应度值并且记录最优值, 把初始结果中适应度值最大的蚁狮选为精英蚁狮个体 $Eantlion$, $I=1$ 。
- c) 通过轮盘赌搜索方式对蚁狮进行优化选择, 利用式(4)和(5)更新 c_i^t 和 d_i^t 的值, 实现蚂蚁在被选出精英蚁狮的周围随机游走, 利用式(1)~(8)使蚂蚁慢慢不断地靠近蚁狮, 之后根据式(9)式更新蚂蚁的位置。
- d) 蚂蚁被捕获之后, 根据式(8)与当前位置最好的精英蚁狮对比, 重新调整最佳精英蚁狮的位置, 更新最优值。如果蚁狮的位置超出了最远边界 u 或者 l , 则按照最远边界处理, $I=I+1$ 。
- e) 判断该结果否达到了算法的最大迭代次数, 若达到了最大精度, 则获得最佳的蚁狮位置, 否则就需要转至步骤 c)。

2 融合莱维飞行与黄金正弦的蚁狮优化算法

2.1 Levy 变异机制

Levy 飞行是法国数学家保罗·列维提出来的, 在此基础上学者们发现大多数生物群体的捕食狩猎的方式均可用 Levy 飞行模式进行描述^[11]。经过研究形成了一种 Levy 飞行觅食假说: Levy 飞行可以提高觅食的效率 and 准确度, 且自然适应性更强。

Levy 飞行机制的长期短距离游走可以提高种群的多样性, 偶尔长距离跳跃具有的方向多变性特点保证了种群对附近区域详细搜索, 具有的突变性特点对探索大空间范围的问题有着很大的优势。短距离与长距离两种方式的结合对求解域实现了充分寻优, 很大程度上提高算法的全局搜索能力。搜索范围内莱维飞行 50 步运动轨迹图^[12]如图 2 所示, 充分验证了 Levy 飞行短距离结合偶尔长距离跳跃的特征, 对求解域实现了充分的探索。



图 2 搜索范围内的 Levy 飞行 50 步轨迹图

Fig. 2 A 50-step flight path map of Levy in the search area

Levy 飞行的概率密度函数服从 Levy 分布, 其函数如下:

$$P_{\alpha,\beta}(x) = \frac{1}{\pi} \int_0^{\infty} \exp(-rq^{\alpha}) \cos(qx) dq \quad (10)$$

其中: $0 < \alpha \leq 2$ 控制 Levy 分布图的峰的锐度, $r > 0$ 控制分布图形的跨度。当 $\alpha = 2$ 时, Levy 分布就是高斯分布, $\alpha = 1$ 时, Levy 分布就是柯西分布。对于非整数的 α 取值, 通过 Levy 分布的概率密度函数分析较为困难, 需要采用模拟方法来产生 Levy 分布的随机步长。其位置更新公式如下:

$$x_i^{t+1} = x_i^t + \alpha \oplus Levy(\lambda) \quad (11)$$

其中: α 为随机步长, \oplus 为点乘, Levy 为符合 Levy 分布的随机搜索路径且符合以下约束:

$$Levy(\lambda) \sim \frac{\varphi * u}{|v|^{\lambda/2}} \quad (12)$$

其中: u, v 服从标准正态分布, $\lambda = 1.5$,

$$\varphi = \left[\frac{\Gamma(1+\lambda) * \sin(\pi * \lambda / 2)}{\Gamma[(1+\lambda)/2] * \lambda * 2^{(\lambda-1)/2}} \right]^{1/\lambda}$$

2.2 黄金正弦算法

黄金正弦算法(golden sine algorithm, Golden-SA)是 Tanyildizi 于 2017 年提出的新型元启发式算法^[13], 该算法的设计灵感来源于数学中的正弦函数, 该算法利用数学中的正弦函数进行计算迭代寻优, 其优点是收敛速度快、鲁棒性好、易于实现、调节的参数和运算符少。

Gold-SA 根据正弦函数与单位圆的关系, 可以历遍正弦函数上的所有值即寻遍单位圆上所有的点, 同时在其位置更新过程中引入黄金分割数缩小解决方案的空间, 以便扫描可能只产生良好结果的区域, 很大提高了搜索速度, 且使“搜索”和“开发”达到良好地平衡。

Golden-SA 算法的核心是位置更新过程, 随机产生 s 个个体的位置, 并用 $X_i^t = (X_{i1}, X_{i2}, X_{i3}, \dots, X_{id})^T$, 其中 X_i^t 表示 d 维个体空间中第 T 次迭代中第 i ($i=1, 2, \dots, s$) 个个体的空间位置, $P_i^t = (P_{i1}, P_{i2}, P_{i3}, \dots, P_{id})^T$ 表示第 T 代个体 i 的最优位置, 在 $T+1$ 次迭代中, 第 i ($i=1, 2, \dots, np$) 个个体的位置更新公式如下:

$$X_i^{T+1} = X_i^T * |\sin(R1)| + R2 * \sin(R1) * |x_1 * P_i^T - x_2 * X_i^T| \quad (13)$$

其中: $R1$ 和 $R2$ 是随机数, $R1$ 决定下一次迭代中个体的移动距离, $R1 \in [0, 2\pi]$, $R2$ 决定下一次迭代的位置更新方向, $R2 \in [0, \pi]$; x_1 和 x_2 是通过黄金分割得到的系数, 可以缩小搜索空间

引领个体趋近最优值, 黄金分割数 $\tau = (\sqrt{5}-1)/2$,

$$x_1 = -\pi + (1-\tau) * 2\pi, \quad x_2 = -\pi + \tau * 2\pi$$

2.3 融合莱维飞行与黄金正弦的蚁狮优化算法(LGSALO)

针对蚁狮算法存在的容易陷入局部最优和收敛精度低的缺陷以及实际工程应用中对于精度的要求, 提出了融合莱维飞行与黄金正弦的蚁狮优化算法(LGSALO)。采用莱维飞行机制对蚂蚁的位置更新处进行变异, 利用式(11)代替式(3)对蚂蚁位置进行更新, 由于 Levy 飞行短距离结合偶尔长距离跳跃的特性使蚂蚁的更新位置更加多样性, 由于蚁狮的最终位置由被吃掉的蚂蚁位置来决定, 从而改善蚁狮整个群体的多样性, 能够快速跳出局部最优, 使得搜索范围更加全面, 种群的总体寻优效率更高。

LGSALO 算法中在精英蚁狮的位置更新处引入黄金正弦算法的黄金分割数, 根据式(13)确定其更新位置, 由于 Gold-SA 函数与单位圆的关系, 可以历遍正弦函数上的所有值即寻遍单位圆上所有的点, 使寻优区域更加全面, 同时通过参数 $R1$ 和 $R2$ 的随机选择控制位置更新距离和方向, 可以逐步缩小搜索空间, 快速引领蚁狮个体趋近最优值, 从而减少算法的寻优时间, 提高算法的寻优速度和精度, 获得理想的寻优结果。

LGSALO 算法的基本步骤归结为

a) 执行原 ALO 算法的步骤 a) ~d);

b) 在进入循环迭代后, 当蚁狮群体中相邻两次迭代的几乎没有差别的最优值都作为历史最优值被记录下来时, 认为算法陷入局部最优停滞, 此时需要进行莱维飞行变异操作;

c) 先将最优蚁狮个体的适应值和个数复制到原先种群的规模, 然后根据式(11)进行莱维飞行变异, 进一步更新蚁狮群体的位置, 并更新最优值。

d) 在精英蚁狮的位置更新处引入黄金分割数, 根据式(13)对蚁狮位置进行更新, 再次记录最优值;

e) 重复步骤 a)~d), 直到达到最大迭代次数时终止操作。

融合莱维飞行与黄金正弦的蚁狮优化算法(LGSALO)的操作流程如图 3 所示。

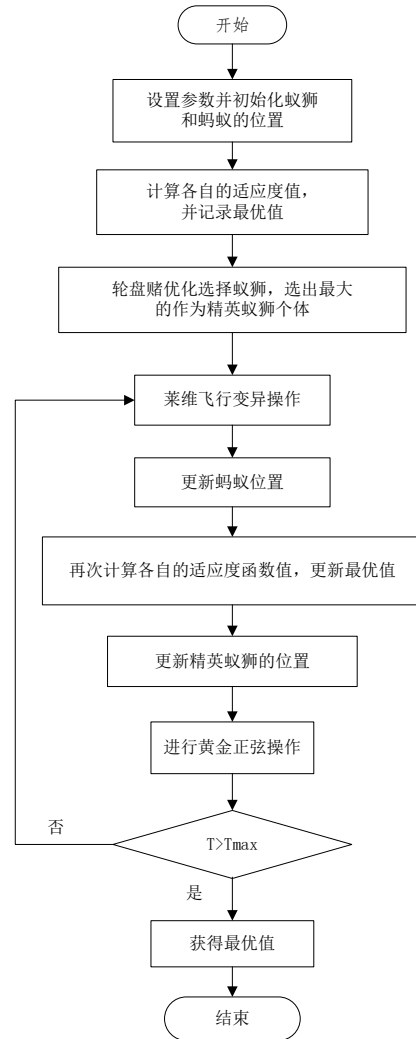


图 3 改进蚁狮算法流程图

Fig. 3 Improved ant lion algorithm flow chart

3 实验仿真与结果分析

3.1 参数设置

该算法所在的实验平台为 Window 7 64 位系统、32 GB 内存。采用 MATLAB 2017b 进行仿真实验。

本文选取了基本的蚁狮算法(ALO)、具有 Levy 变异和精英自适应竞争机制的蚁狮优化算法(LEALO)、双重反馈机制的蚁狮算法(DFALO)^[14]、花授粉算法(flower pollination algorithm ,FPA)^[15]、粒子群算法(Particle Swarm Optimization , PSO)^[16] 和具有自适应边界与最优引导的莱维飞行蚁狮优化算法(ABLALO)^[17]与融合莱维飞行与黄金正弦的蚁狮优化算法(LGSALO)函数进行优化结果对比, 其中 ALO、LGSALO、FPA、PSO 四种算法的共有参数统一设置, 利用 MATLAB 对四种算法每个函数分别进行 30 次的独立运行仿真实验, 以避免结果偶然性带来的偏差。种群规模统一设置为 30, 迭代次数设置为 500, 以保证实验的公平性与合理性。

3.2 测试函数

实验使用的测试函数及其具体信息如表 1 所示。

表 1 测试函数及其具体信息

函数	名称	维数	范围	最优值
$F_1(x) = \sum_{i=1}^n x_i^2$	Sphere	30	[-100,100]	0
$F_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	Schwefel	30	[-10,10]	0
$F_3(x) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2$	Rotated hyper-ellipsoid	30	[-100,100]	0
$F_4(x) = \max_i \{ x_i , 1 \leq x \leq n \}$		30	[-100,100]	0
$F_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	Rosenbrock	30	[-30,30]	0
$F_6(x) = \sum_{i=1}^n ([x_i + 0.5])^2$		30	[-100,100]	0
$F_7(x) = \sum_{i=1}^n i x_i^4 + \text{random}[0,1]$	Sum squares	30	[-1.28,1.28]	0
$F_8(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	Rastrigin	30	[-5.12,5.12]	0
$F_9(x) = -20 \exp \left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \right) - \exp \left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i) \right) + 20 + e$	Ackley	30	[-32,32]	0
$F_{10}(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos \left(\frac{x_i}{\sqrt{i}} \right) + 1$	Griewank	30	[-600,600]	0
$F_{11} = 0.1 \left\{ \begin{aligned} & \sin^2(3\pi x_i) + \sum_{i=1}^n (x_i - 1)^2 \\ & [1 + \sin^2(3\pi x_i + 1)] + \\ & (x_i - 1)^2 [1 + \sin^2(2\pi x_i)] \\ & + \sum_{i=1}^n u(x_i, 5, 100, 4) \end{aligned} \right\}$		30	[-50,50]	0
$F_{12} :$ $f_1, f_2, f_3, \dots, f_{10} =$ Griewank's Function $[\delta_1, \delta_2, \delta_3, \dots, \delta_{10}] = [1, 1, 1, \dots, 1]$ $[\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_{10}] =$ $[5/100, 5/100, 5/100, \dots, 5/100]$		10	[-5,5]	0

3.3 函数实验结果与分析

选取 ALO、FPA、PSO 以及相应的 DFALO、LEALO、ABLALO 与 LGSALO 进行函数优化的结果对比, (其中 ALO、FPA、PSO 和本研究算法均为测试函数仿真 30 次实验结果, LEALO、DFALO、ABLALO 为参考其他文献数据结果所得) 以验证 LGSALO 的性能。为减少实验中随机性的影响, 分别记录其最优值、平均值和标准方差, 实验结果如表 2 所示。

表 2 函数测试实验结果

函数	算法	最优值	平均值	标准方差
F_1	LGSALO	0	0	0
	ALO	1.36E-09	4.15E-09	1.80E-09
	LEALO ^[10]	1.87E-11	-	5.64 E-10
	ABLALO ^[17]	4.45E-18	1.91E-17	7.40E-18
	FPA	2.0258	3.9615	1.0449
F_2	LGSALO	0	0	0
	ALO	1.26E-05	0.392192	0.831436
	LEALO ^[10]	5.45E-08	-	2.78E-05
	ABLALO ^[17]	4.10E-17	6.97E-17	1.41E-17
	FPA	16.4433	31.4963	5.8462
F_3	LGSALO	0	0	0
	ALO	1.67E-05	4.86 E-03	1.63E-02
	FPA	2.13E+02	3.56E+02	1.71E+02
	PSO	2.12E-90	4.55E-71	4.38E-70
	F_4	LGSALO	0	0
ALO		6.03E-05	5.96E-04	5.36 E-04
FPA		1.31E+01	1.59E+01	2.16E+00
PSO		3.86E-51	5.03E-49	8.48E-36
F_5		LGSALO	3.5E-05	2.79E-02
	ALO	1.12E-02	1.47 E+02	3.20 E+02
	ABLALO ^[17]	2.75E-04	6.13E-03	6.10E-03
	FPA	3.16E+03	3.81E+04	2.61E+04
	PSO	1.22E-03	6.28 E+00	2.07E+00
F_6	LGSALO	1.4E-05	1.68E-04	1.27E-04
	ALO	2.88 E-03	1.49E-02	1.07E-02
	FPA	4.01E-02	1.54E-01	9.96E-02
	PSO	1.50E-22	7.43E-20	4.21E-21
	F_7	LGSALO	0	0
ALO		4.9748	20.9273	9.4335
DFALO ^[14]		8.17E-07	1.21E-04	9.15E-05
FPA		93.5748	136.0570	19.4422
PSO		1.7907	2.1706	1.2262
F_8	LGSALO	0	0	0
	ALO	4.43E-02	0.2198	0.1116
	DFALO ^[14]	0	0.547 23	2.447 3
	FPA	9.7327	16.5867	3.9004
	PSO	2.0000	10.4000	4.5400
F_9	LGSALO	8.88E-16	8.88E-16	0
	ALO	10.5557	13.8402	2.7962
	FPA	11.3736	13.0578	2.3947
	PSO	3.25E-09	3.73E-01	6.93E-01
	F_{10}	LGSALO	0	0
ALO		9.0956	15.6366	3.7852
LEALO ^[10]		4.77E-09	-	0.0209
ABLALO ^[17]		0	1.51E-15	7.39E-15
FPA		43.0990	49.4258	5.6458
F_{11}	LGSALO	1.48E-02	9.13E-02	4.04E-02
	LGSALO	1.89E-03	2.61E-03	0.32452
	ALO	5.71E-02	0.19429	0.47399
	FPA	2.56E+00	3.45E+01	1.34E+00
	PSO	2.56 E-22	3.29E-18	1.38 E-22
F_{12}	LGSALO	4.02E-04	1.36E-03	3.57E-02
	ALO	1.06E-02	2.26 E-02	0.16834
	FPA	3.84E+02	3.65E+02	1.84E+02
	PSO	4.48E-03	9.89E-03	4.54E-03

根据实验结果可以看出, 对于所有的测试函数, 本文提出的改进蚁狮优化算法的最优值、平均值、标准方差全部明显优于 ALO、FPA, 部分函数仅次于 PSO。

对于函数 F_1 、 F_2 、 F_3 、 F_4 、 F_7 、 F_8 、 F_{10} , 改进算法 LGSALO 得

到了理论上的最优值, 寻优成功率达到了 100%, 说明其寻优性能良好且鲁棒性强, 而且性能明显优于原算法以及其他两类算法 PSO 和 FPA, PSO 算法一直都是寻优能力很强的; 对于 F_1 , LGSALO 的每一次迭代均搜索到了最优值, PSO 次之, 但是 ALO 的平均最优值优于 PSO, 并且优于 FPA; 对于 $F_2, F_3, F_4, F_7, F_{10}$, LGSALO 搜索到了最优值, PSO 次之, ALO 的性能优于 FPA; 对于函数 F_8 , LGSALO 最优, ALO 优于另外两种算法, 取得了不错的优化效果; 对于函数 F_5, F_9, F_{12} , LGSALO 虽然没有达到理论最优值, 但是与其他算法相比是最接近于最优理论值的, 最优值达到了 $8.88E-16$ 这一级别, 且整体性能好, 最优值 PSO 次之, ALO 优于 FPA; 对于函数 F_6, F_{11} , LGSALO 的最优值仅次于寻有能力很强的 PSO 算法, 测试结果在很大程度上优于其他两种对比函数: 其平均值和标准方差明显优于其他算法很多, 说明其性能良好且稳定性好, 有相对较好的满意值, 寻优效果良好, ALO 次之, FPA 表现最差。

对于文献[10]的函数 F_1, F_2, F_{10} , 文献[14]的函数 F_4, F_3 和文献[17] 的函数 F_1, F_2, F_3, F_{10} 与 LGSALO 算法的相同函数结果进行比较, LGSALO 算法的仿真结果几乎全部达到了理论最优值, LEALO、DFALO、ABLALO 三类改进算法并没有达到最优理论值, LGSALO 的函数仿真结果明显优于这三类改进蚁狮算法。对比结果证明 LGSALO 算法的寻优精度更高, 寻优性能更加稳定。

总体来说, 改进的算法 LGSALO 的收敛精度高, 具有很好的鲁棒性, 寻优性能可以一直保持, 相对更加稳定, 83.3% 达到了算法对比最优, 58.3% 达到了理论最优, 开发性强, 测试结果方面更具有优势, 寻优性能更好。

为了更加简明直观地对比 LGSALO 算法与 ALO、FPA、PSO 四种算法的寻优性能, 图 4 给出了 12 个测试函数在实验时 30 次独立测试的其中一次的收敛曲线。

从图中可以看出, LGSALO 算法可以在最短的时间内找到最优值或者接近于理论最优值, 且部分收敛曲线最接近于数轴, 这说明了 LGSALO 的收敛速度快, 收敛精度更高, 具有很强的开发性。 $F_1, F_2, F_3, F_4, F_7, F_8, F_{10}$ 七个函数曲线全部在迭代次数在 50 左右的时候收敛曲线就已经达到了理论最优且一直保持, 收敛性能表现最好, 证明了 LGSALO 寻优性能很高, 其收敛速度和收敛精度相比其他函数都有很大的提高, 很好的弥补了基本 ALO 算法的不足; F_5, F_9, F_{12} 三个函数在迭代最后是最接近于理论值最优值的, 搜寻结果较为理想, 相比其他算法寻优精度更高, 具有很高的稳定性; 对于 F_6, F_{11} 两个函数, LGSALO 相比 ALO 和 FPA 算法更接近最优值, 仅次于 PSO 算法, 稳定性良好。这些图像更加简明直观的证明 LGSALO 在改进之后不仅提高了收敛精度和收敛速度, 并且有效提升了全局搜索能力以及局部的开发能力, 且在函数优化方面表现良好。

基于测试函数的适应度值的比较结果表 2 和迭代收敛曲线图 4 的分析, 表明了所提出的融合莱维飞行与黄金正弦的蚁狮优化算法(LGSALO)具有良好的寻优性能, 在函数优化方面是有效可靠的。

4 结束语

本文提出了一种融合莱维飞行与黄金正弦的蚁狮优化算法(LGSALO)。该算法融合了莱维飞行和黄金正弦算法大大提高了全局搜索能力, 能够在很大程度上改善蚁狮优化算法的收敛精度低和易陷入局部最优的缺陷。通过 12 个测试函数的测试, 仿真实验结果证明, LGSALO 算法的寻优性能表现良好, 开发能力强。

本文的不足: 首先该实验只使用了 12 个测试函数, 但是

其他的标准测试函数和工程问题应该在将来进行测试, 以验证和扩展所提出的算法; 其次如何使改进算法表现出更好的寻优性能是下一步的研究工作; 最后就是进一步将算法应用到更广泛的领域。

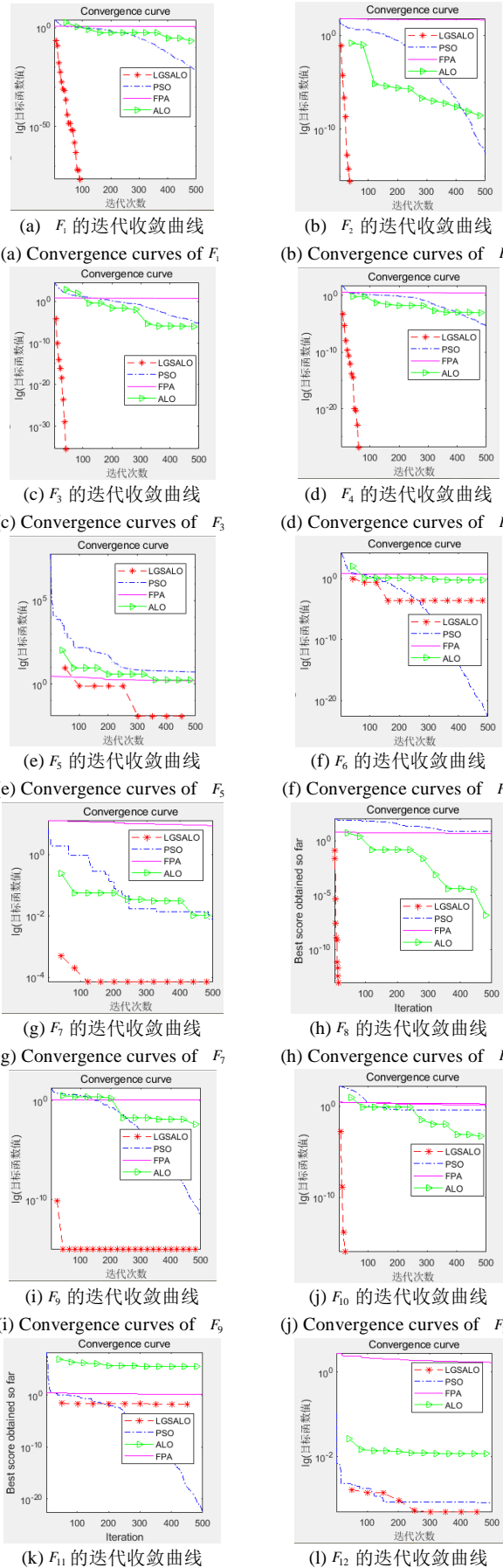


图 4 测试函数的迭代收敛曲线
Fig. 4 Iterative convergence curve of the test function

参考文献:

- [1] Mirjalili S. The ant lion optimizer [J]. *Advances in Engineering Software*, 2015, 83 (1): 80-98.
- [2] Talatahari S. Optimum design of skeletal structures using ant lion optimizer [J]. *International Journal of Optimization in Civil Engineering*, 2016, 6 (1): 13-25.
- [3] Kaur M, Mahajan A. Community detection in complex networks: a novel approach based on ant lion optimizer [C]//*Proc of the 6th International Conference on Soft Computing for Problem Solving*. 2017: 22-34.
- [4] Mouassa S, Bouktir T, Salhi A. Ant lion optimizer for solving optimal reactive power dispatch problem in power systems [J]. *Engineering Science and Technology: an International Journal*, 2017, 20 (3): 885-895.
- [5] Nair S S, Rana K P S, Kumar Ve, t al. Efficient modeling of linear discrete filters using antlion optimizer [J]. *Circuits Systems and Signal Processing*, 2017, 36 (4): 1535-1568.
- [6] Yao P, Wang H. Dynamic adaptive ant lion optimizer applied to route planning for unmanned aerial vehicle [J]. *Soft Computing*, 2017, 21 (18): 5475-5488.
- [7] 张振兴, 杨任农, 房育寰, 等. 自适应 Tent 混沌搜索的蚁狮优化算法 [J]. *哈尔滨工业大学学报*, 2018, 50 (5): 152-159. (Zhang Zhenxing, Yang Rennong, Fang Yuhuan, *et al.* Ant lion optimization algorithm with adaptive tent chaotic search [J]. *Journal of Harbin Institute of Technology*, 2018, 50 (5): 152-159.)
- [8] 赵世杰, 高雷阜, 于冬梅, 等. 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. (Zhao Shijie, Gao Leifu, Yu Dongmei, *et al.* Ant lion optimization algorithm with chaotic detection mechanism to optimize SVM parameters [J]. *Computer Science and Exploration*, 2016, 10 (5): 722-731.)
- [9] 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/OL]. *控制与决策*, [2018-10-11]. <https://doi.org/10.13195/j.kzyjc.2017.1282>. (Zhao Xiaoguo, Liu Ding, Jing Kunlei. Noise nonlinear system identification based on fusion improved ant-lion algorithm and T-S fuzzy model [J/OL]. *Control and Decision*, [2018-10-11]. <https://doi.org/10.13195/j.kzyjc.2017.1282>.)
- [10] 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. (Jing Kunlei, Zhao Xiaoguo, Zhang Xinyu, *et al.* Ant lion optimization algorithm with levy variation and elite adaptive competition mechanism [J]. *Journal of Intelligent Systems*, 2018, 13 (2): 236-242.)
- [11] Mehrdad G, Zahra Z, Yazdan A. Computer Simulation Study of the Levy Flight Process [J]. *Physica A*, 2009, (388): 1509-1514.
- [12] Yang X S. Cuckoo search and firefly algorithm: overview and analysis [M]. Cham: Springer International Publishing, 2014, 516: 1-26.
- [13] Tanyildizi E, Demir G. Golden Sine Algorithm: A Novel Math-Inspired Algorithm [J]. *Advances in Electrical & Computer Engineering*, 2017, 17 (2): 71-78.
- [14] 吴伟民, 张晶晶, 林志毅, 等. 双重反馈机制的蚁狮算法 [J]. *计算机工程与应用*, 2017, 53 (12): 31-35+75. (Wu Weimin, Zhang Jingjing, Lin Zhiyi, *et al.* Ant lion algorithm with dual feedback mechanism [J]. *Computer Engineering and Application*, 2017, 53 (12): 31-35+75.)
- [15] Yang X S. Flower pollination algorithm for global optimization [C]//*Proc of International Conference on Unconventional Computation and Natural Computation*. Springer-Verlag, 2012, 7445: 240-249.
- [16] Kennedy J, Eberhart R. Particle swarm optimization [C]//*Proc of IEEE Conference on Neural Networks*. 1995: 1942-1948.
- [17] 王若安, 周越文, 韩博, 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. (Wang Ruohan, Zhou Yuewen, Han Bo, *et al.* Levy flying ant lion optimization algorithm with adaptive boundary and optimal guidance [J]. *Microelectronics and Computer*, 2018, 35 (9): 20-25+31.)