



计算机工程与应用
Computer Engineering and Applications
ISSN 1002-8331, CN 11-2127/TP

《计算机工程与应用》网络首发论文

题目: 透镜成像反向学习的精英池侏儒猫鼬优化算法
作者: 贾鹤鸣, 陈丽珍, 力尚龙, 刘庆鑫, 吴迪, 卢程浩
网络首发日期: 2022-10-20
引用格式: 贾鹤鸣, 陈丽珍, 力尚龙, 刘庆鑫, 吴迪, 卢程浩. 透镜成像反向学习的精英池侏儒猫鼬优化算法[J/OL]. 计算机工程与应用.
<https://kns.cnki.net/kcms/detail/11.2127.TP.20221019.1554.014.html>



网络首发: 在编辑部工作流程中, 稿件从录用到出版要经历录用定稿、排版定稿、整期汇编定稿等阶段。录用定稿指内容已经确定, 且通过同行评议、主编终审同意刊用的稿件。排版定稿指录用定稿按照期刊特定版式(包括网络呈现版式)排版后的稿件, 可暂不确定出版年、卷、期和页码。整期汇编定稿指出版年、卷、期、页码均已确定的印刷或数字出版的整期汇编稿件。录用定稿网络首发稿件内容必须符合《出版管理条例》和《期刊出版管理规定》的有关规定; 学术研究成果具有创新性、科学性和先进性, 符合编辑部对刊文的录用要求, 不存在学术不端行为及其他侵权行为; 稿件内容应基本符合国家有关书刊编辑、出版的技术标准, 正确使用和统一规范语言文字、符号、数字、外文字母、法定计量单位及地图标注等。为确保录用定稿网络首发的严肃性, 录用定稿一经发布, 不得修改论文题目、作者、机构名称和学术内容, 只可基于编辑规范进行少量文字的修改。

出版确认: 纸质期刊编辑部通过与《中国学术期刊(光盘版)》电子杂志社有限公司签约, 在《中国学术期刊(网络版)》出版传播平台上创办与纸质期刊内容一致的网络版, 以单篇或整期出版形式, 在印刷出版之前刊发论文的录用定稿、排版定稿、整期汇编定稿。因为《中国学术期刊(网络版)》是国家新闻出版广电总局批准的网络连续型出版物(ISSN 2096-4188, CN 11-6037/Z), 所以签约期刊的网络版上网络首发论文视为正式出版。

透镜成像反向学习的精英池侏儒猫鼬优化算法

贾鹤鸣¹, 陈丽珍¹, 力尚龙¹, 刘庆鑫², 吴迪³, 卢程浩¹

1. 三明学院信息工程学院, 福建 三明 365004

2. 海南大学计算机科学与技术学院, 海口 570228

3. 三明学院教育与音乐学院, 福建 三明 365004

摘要: 侏儒猫鼬优化算法 (Dwarf Mongoose Optimization, DMO) 是新提出的一种元启发式算法, 该算法具有较强的全局探索能力和稳定性, 但由于原始算法中仅依靠雌性首领带领整个猫鼬种群进行搜索, 会产生收敛速度较慢、易陷入局部最优以及探索阶段与开发阶段之间的平衡较差等问题。针对上述问题, 提出一种融合透镜成像反向学习的精英池侏儒猫鼬优化算法 (Improved Dwarf Mongoose Optimization, IDMO), 首先采用透镜成像反向学习策略, 避免算法在迭代过程中陷入局部最优, 增强算法的探索能力; 然后在阿尔法组觅食时引入精英池策略, 提高了算法的收敛精度, 进一步增强算法探索能力。通过基准测试函数进行实验, 表明 IDMO 算法具有良好的寻优性能和鲁棒性, 且算法收敛速度得到显著提升。最后通过对汽车碰撞优化问题的求解, 进一步验证了 IDMO 算法具有良好的适用性和有效性。

关键词: 侏儒猫鼬优化算法; 元启发式算法; 透镜成像反向学习策略; 精英池策略

文献标志码: A **中图分类号:** TP301.6 **doi:** 10.3778/j.issn.1002-8331.2208-0291

Optimization algorithm of elite pool dwarf mongoose based on lens imaging reverse learning

JIA Heming¹, CHEN Lizhen¹, LI Shanglong¹, LIU Qingxin², WU Di³, LU Chenghao¹

1. Department of Information Engineering, Sanming University, Sanming, Fujian 365004, China

2. School of Computer Science and Technology, Hainan University, Haikou 570228, China

3. College of Enducation and Music, Sanming University, Sanming, Fujian 365004, China

Abstract: Dwarf Mongoose Optimization (DMO) is a newly proposed meta heuristic algorithm. The algorithm has strong global exploration ability and stability. However, due to the fact that only female leader is used in the original algorithm to lead the whole mongoose population to search, there will be some problems, such as slow convergence speed, easy to fall into local optimization and poor balance between exploration stage and exploitation stage. To solve the above problems, this paper proposes an improved Dwarf Mongoose Optimization (IDMO). Firstly, the lens imaging reverse learning strategy is adopted to avoid the algorithm falling into local optimization in the iterative process and enhance the exploration ability of the algorithm; then the elite pool strategy is introduced into the Alpha Group foraging, which improves the convergence accuracy of the algorithm and further enhances the exploration ability of the algorithm. Experiments with benchmark function show that IDMO has good optimization performance

基金项目: 福建省自然科学基金(2021J011128)。

作者简介: 贾鹤鸣(1983—),男,博士,教授,CCF 会员,主要研究领域为群体智能优化算法及工程应用,E-mail: jiaheminglucky99@126.com; 陈丽珍(2003—),女,本科生,研究方向为群体智能优化算法及工程应用;力尚龙(2000—),男,本科生,研究方向为群体智能优化算法及工程应用;刘庆鑫(1997—),男,硕士研究生,研究方向为群体智能优化算法;吴迪(1984—),女,博士,副教授,研究方向为群体智能优化算法及应用;卢程浩(2002—),男,本科生,研究方向为群体智能优化算法及工程应用。

and robustness, and the convergence speed of the algorithm is significantly improved. Finally, by solving the car crash worth iness optimization problem, it is further verified that the IDMO algorithm has good applicability and effectiveness.

Key words: dwarf mongoose optimization algorithm; meta heuristic algorithm; lens imaging reverse learning strategy; elite pool strategy

随着科技的不断发展,工程优化问题变得越发复杂,亟需寻找更为有效的求解算法。因此许多学者受自然界中生物的习性和事物的自然规律启发提出了元启发式算法,并将其应用于工程优化问题求解。近年来提出的元启发式算法主要有算术优化算法 (Arithmetic Optimization Algorithm, AOA)^[1]、正余弦优化算法 (Sine Cosine Algorithm, SCA)^[2]、灰狼优化算法 (Grey Wolf Optimization, GWO)^[3]、樽海鞘群优化算法 (Salp Swarm Algorithm, SSA)^[4]、蛇优化算法 (Snake Optimizer, SO)^[5]、鲟鱼优化算法 (Remora Optimization Algorithm, ROA)^[6]、黏菌优化算法 (Slime Mould Algorithm, SMA)^[7]和粒子群优化算法 (Particle Swarm Optimization, PSO)^[8]等。元启发式算法因具有稳定性强且易于实现等优点,常被人们应用于工程设计问题求解。

侏儒猫鼬优化算法 (Dwarf Mongoose Optimization Algorithm, DMO) 是由 Jeffrey O. Agushaka 等于 2022 年提出的一种群体智能优化算法^[9]。其灵感来源于侏儒猫鼬的群体觅食行为,但 DMO 算法依然存在收敛速度慢及易陷入局部最优的缺点。透镜成像反向学习策略是基于透镜成像原理提出的一种改进策略^[10],能够增强算法跳出局部最优的能力,从而提高算法的寻优性能。周鹏等提出一种改进的平衡优化器算法,通过透镜成像反向学习策略来提高算法的搜索能力,降低算法在迭代后期陷入局部最优的概率^[11]。刘庆鑫等提出了一种多策略融合的鲟鱼优化算法,利用布朗运动提高算法的探索能力,再引入透镜成像反向学习策略,进一步提高算法跳出局部最优解的能力,使算法的探索能力和开发能力更加平衡^[12]。

为了提高 DMO 算法的收敛速度,增强其探索能力,本文提出了融合透镜成像反向学习的精英池侏儒猫鼬优化算法。首先,采用透镜成像反向学习策略,降低算法陷入局部最优的概率,增强算法全局探索的能力;然后,在阿尔法组成员觅食的时候引入精英池策略,避免了算法迭代后期个体向食物源聚集的问题,进一步降低陷入局部最优的概率。基于 13 个标准测试函数的实验结果表明, IDMO 算法具有更好的寻优性能和鲁棒性。最后,通过对汽车碰撞优化问题的求解,

验证 IDMO 算法对工程问题同样具有较高的适用性。

1 侏儒猫鼬优化算法

DMO 算法是模拟侏儒猫鼬半游牧式生活的一种元启发式算法。侏儒猫鼬通常生活在一个母系社会的家族群体中,主要有觅食、侦察和保姆三种社会职能。侏儒猫鼬以集体觅食和侦察而闻名,由雌性首领引导种群进行食物源的搜寻。一旦满足保姆交换条件,即当阿尔法组未能寻找到合适的食物时,将交换阿尔法组和保姆组的成员,且阿尔法组同时进行觅食和寻找睡眠丘。

1.1 阿尔法组

1.1.1 雌性首领的产生

雌性首领在阿尔法组中产生,阿尔法组中每个雌性个体成为首领的概率为 α , 计算公式如下:

$$\alpha = \frac{fit_i}{\sum_{i=1}^N fit_i} \quad (1)$$

其中 fit_i 是第 i 个个体的适应度, N 是侏儒猫鼬种群中个体的总数。阿尔法组的个体数量为 n' , bs 为保姆的数量。

1.1.2 阿尔法组成员觅食

阿尔法组成员将共行并进行觅食,食物源的候选位置由式 (2) 给出:

$$X_{i+1} = X_i + phi \times peep \times (X_i - X_{rand}) \quad (2)$$

其中 X_{i+1} 是找到的食物源新位置, X_i 为雌性首领的当前位置, phi 是均匀分布在 $[-1,1]$ 之间的随机数, 本文 $peep$ 选取为 2, X_{rand} 是阿尔法组中的随机个体。

1.2 保姆交换条件

保姆交换条件是用于重置阿尔法组和保姆组中的猫鼬个体。当阿尔法组成员未能搜寻到合适的食物时,认为阿尔法组成员能力不足,将交换阿尔法组和保姆组的成员。交换条件满足后,阿尔法组将同时进行觅食和寻找睡眠丘,计算公式如下:

$$X_b = lb + rand * (ub - lb) \quad (3)$$

其中 X_b 为交换后个体的新位置, ub 和 lb 分别为搜索空间的上界和下界, $rand$ 是 0 到 1 之间的随机数。

保姆交换后的觅食行为由公式(2)实现。睡眠丘是猫鼬休息的场所,而猫鼬不会回到之前的睡眠丘,这种生活模式能够避免搜索区域被过度开发的问题。新搜寻到的睡眠丘的数学模型如下:

$$X_{sm} = \begin{cases} X_i - CF * phi * rand * [X_i - \bar{M}] & \text{if } \varphi_{i+1} > \varphi_i \\ X_i + CF * phi * rand * [X_i - \bar{M}] & \text{else} \end{cases} \quad (4)$$

其中 X_{sm} 为新的睡眠丘的位置, \bar{M} 是决定猫鼬移动到新睡眠丘的方向向量, φ 是睡眠丘的平均值, 计算公式如下:

$$\bar{M} = \sum_{i=1}^N \frac{X_i \times sm_i}{X_i} \quad (5)$$

$$\varphi = \frac{\sum_{i=1}^N sm_i}{n} \quad (6)$$

其中 sm_i 代表睡眠丘值:

$$sm_i = \frac{fit_{i+1} - fit_i}{\max\{|fit_{i+1}|, |fit_i|\}} \quad (7)$$

CF 表示猫鼬种群移动能力的参数, 它会随着迭代次数线性递减, 计算公式如下:

$$CF = \left(1 - \frac{t}{T}\right)^{\left(\frac{2t}{T}\right)} \quad (8)$$

其中 t 为当前迭代次数, T 为最大迭代次数。

2 改进的侏儒猫鼬优化算法

2.1 基于透镜成像原理的反向学习策略

反向学习是通过计算当前位置的反向解, 从而扩大搜索范围的一种改进策略。在群智能优化算法中采用反向学习策略, 可以在一定程度上提升算法的寻优能力; 但通过反向学习求得的反向解是固定的, 若个体已经陷入局部最优, 且其反向解劣于当前解, 则反向学习策略无法使该个体跳出局部最优。而透镜成像反向学习可以有效解决上述问题。

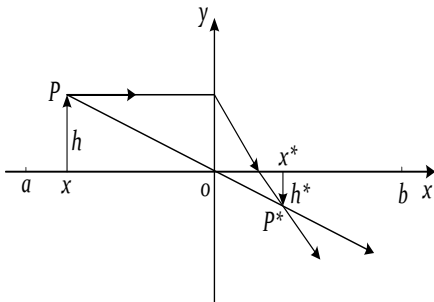


图1 透镜成像反向学习策略示意图

Fig.1 Opposition learning strategy based on lens image

透镜成像反向学习策略如图1所示, 以二维空间

为例, $[a, b]$ 为解的搜索范围, y 轴表示凸透镜。假设有一物体 P , 高度为 h , 在 x 轴上的投影为 x ; 该物体通过凸透镜成像在凸透镜的另一侧呈现出一个倒立的实像 P^* , 高度为 h^* , 在 x 轴上的投影为 x^* 。由凸透镜成像原理可得:

$$\frac{(a+b)/2 - x}{x^* - (a+b)/2} = \frac{h}{h^*} \quad (9)$$

令 $k = h/h^*$, 则式(9)可改写为:

$$x^* = \frac{a+b}{2} + \frac{a+b}{2k} - \frac{x}{k} \quad (10)$$

式(10)为凸透镜成像反向学习策略的反向解的求解公式, 当 $k=1$ 时, 式(10)可化简为:

$$x^* = a+b-x \quad (11)$$

该式即为反向学习的求解公式。

由上述可知, 反向学习就是特殊的透镜成像反向学习, 采用反向学习得到的是固定的反向解。而通过调整 k 的大小, 可以在透镜反向学习中获得动态变化的反向解, 进一步提升算法的寻优能力。本文中采用的 k 值计算公式如下:

$$k = (1 + (\frac{3 * t}{T})^{0.5})^8 \quad (12)$$

2.2 精英池策略

精英池策略是用于提高算法探索能力的一种选择策略^[13]。其将当前最优的三个个体和种群中适应度值前一半个体的加权平均位置储存在精英池中, 种群中前一半的个体在进行位置更新时都会随机选择精英池中的一个个体作为食物源; 该策略有效避免了种群因只有一个食物源而陷入局部最优的问题, 提高了算法的探索能力。

原始 DMO 算法具有较强的全局探索能力, 但由于仅确定一个食物源会导致猫鼬种群出现向一个食物源聚集的问题, 导致算法易陷入局部最优, 所以本文采用上述策略来解决该问题。选择当前适应度值最优的三个个体和阿尔法组全部成员的加权平均位置存储在精英池中。三个最优个体能够帮助阿尔法组进行探索, 加权平均位置则代表整个优势种群的进化趋势, 有利于算法进行开发。随后将随机选择精英池中的一个个体进行觅食, 从而增强了原始算法的探索能力。精英池策略如图2所示。

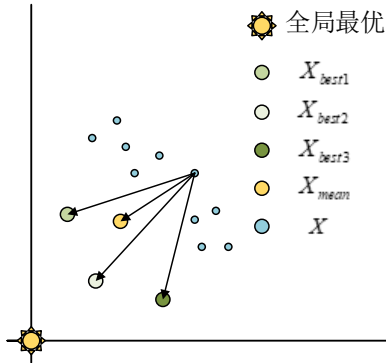


图2 精英池策略示意图

Fig.2 Schematic diagram of elite pool strategy

图中, X_{best1} 、 X_{best2} 和 X_{best3} 分别为当前种群适应度值最优的前三个个体, X_{mean} 是阿尔法全部成员的加权平均位置, 其计算公式如下:

$$X_{mean} = \sum_{i=1}^{n'/2} \omega_i \times X_t^i \quad (13)$$

其中 X_t^i 是第 i 个阿尔法成员在第 t 代的位置, ω_i 是阿尔法组中按适应度值降序排列的权重系数, 其计算公式如下:

$$\omega_i = \frac{\ln(n'/2 + 0.5) - \ln(i)}{\sum_{i=1}^{n'/2} (\ln(n'/2 + 0.5) - \ln(i))} \quad (14)$$

调整后的食物源的更新公式如下:

$$X_{i+1} = X_{elite} + \text{phi} \times \text{peep} \times (X_i - X_{rand}) \quad (15)$$

其中 X_{elite} 是精英池策略选择的个体位置。

2.3 IDMO 的伪代码

融合透镜成像反向学习的精英池侏儒猫鼬优化算法的伪代码如下所示:

IDMO 伪代码

```

Begin
  初始化算法参数
  初始化猫鼬种群
  For  $t = 1 : T$ 
    计算猫鼬的适应度
    使用透镜成像反向学习策略根据公式 (11)
    计算个体成为首领的概率  $\alpha$  根据公式 (1)
    基于精英池策略的觅食
    If 满足保姆交换条件
      初始化个体根据公式 (3) 并计算适应度
      基于精英池策略的觅食
      阿尔法组寻找新睡眠丘根据公式 (4)
      计算评估睡眠丘值根据公式 (7)
      计算睡眠丘值的平均值根据公式 (6)
    End
    更新目前最优解
  End
End

```

2.4 IDMO 的流程图

融合透镜成像反向学习的精英池侏儒猫鼬优化算法实现流程如图 3 所示:

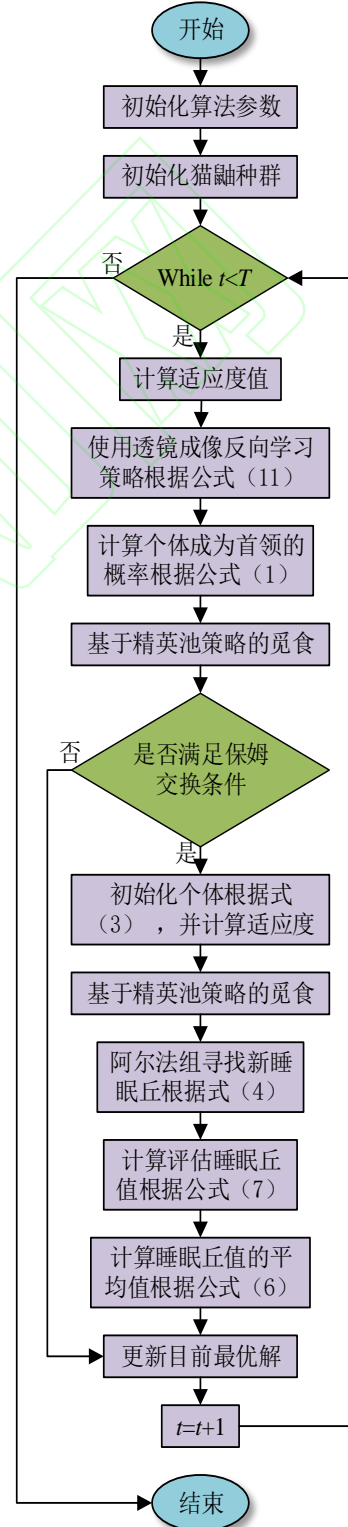


图3 IDMO 算法流程

Fig.3 Flowchart of IDMO

2.5 IDMO 的时间复杂度分析

时间复杂度直接反映算法的运算效率,间接反映算法的收敛速度,是体现算法性能的重要因素。

在 DMO 算法中,假设种群规模为 N ,个体维度为 n ,初始化算法参数所需时间为 η_0 ,每一维上产生均匀分布随机数的时间为 η_1 ,则初始化种群阶段的时间复杂度为

$$T_1 = O(\eta_0 + Nn\eta_1) = O(n)$$

进入迭代后,迭代次数为 T ,计算每个猫鼬个体适应度值的时间为 $f(n)$,根据公式(1)计算阿尔法组中雌性个体成为首领的概率所需时间为 η_2 ,则此阶段的时间复杂度为

$$T_2 = O(Nf(n) + \eta_2) = O(n + f(n))$$

在阿尔法组成员觅食阶段,阿尔法组的个体数量为 n' ,在公式(2)中, ϕ 为均匀分布的随机数,每个个体的每一维都各不相同,其生成时间均与 η_1 一致,选取 $peep$ 的时间为 η_3 且由公式(2)进行每一维食物源位置更新的时间为 η_4 。则该阶段的时间复杂度为

$$T_3 = O(\eta_3 + nn'(\eta_1 + \eta_1 + \eta_4)) = O(n)$$

满足保姆交换条件后,在公式(3)中, $rand$ 是随机数且每个个体的每一维都各不相同,其生成时间均与 η_1 一致。公式(3)初始化所需时间为 η_5 。在公式(4)中,计算 \bar{M} 、 ϕ 和 sm_i 所需时间分别为 η_6 、 η_7 、 η_8 ,计算 ϕ 和 $rand$ 所需时间与上述一致。则该阶段的时间复杂度为

$$T_4 = O(Nn(\eta_1 + \eta_1 + \eta_1 + \eta_6 + \eta_7 + \eta_8) + \eta_5) = O(n)$$

综上所述,DMO 的总时间复杂度为

$$T_D = T_1 + T(T_2 + T_3 + T_4) = O(n + f(n))$$

在改进算法 IDMO 中,算法的种群规模、个体维度、初始化参数时间以及每一维上产生均匀分布随机数的时间均与 DMO 算法一致,所以初始化种群阶段

的时间复杂度为

$$T'_1 = T_1 = O(\eta_0 + Nn\eta_1) = O(n)$$

进入迭代后,迭代次数为 T ,计算每个猫鼬个体适应度值的时间和计算阿尔法组中雌性个体成为首领的概率所需的时间均与 DMO 算法一致,则此阶段的时间复杂度为

$$T'_2 = T_2 = O(Nf(n) + \eta_2) = O(n + f(n))$$

设由公式(12)产生 k 的时间为 t_1 ,由公式(14)产生的阿尔法组中按适应度值降序排列的权重系数 ω_i 的时间为 t_2 ,由公式(13)产生的 X_{mean} 的时间为 t_3 ,由公式(15)产生的 X_{i+1} 的时间为 t_4 。则该阶段的时间复杂度为

$$T'_3 = O(t_1 + nn'(t_2 + t_4) + t_3) = O(n)$$

综上所述,IDMO 算法的总时间复杂度为

$$T_I = T'_1 + T(T'_2 + T'_3) = O(n + f(n))$$

由此可知,与 DMO 算法相比,本文 IDMO 算法并未增加时间复杂度,执行效率没有下降。

3 实验仿真与分析

本次实验的实验操作系统为 Windows11 系统,11th Gen Intel(R) Core(TM) i7-11700 2.50GHz,16.00GB 内存,实验仿真过程在 Matlab 2021a 中实现。为了验证 IDMO 算法的性能,本文选取了 13 个标准测试函数对 IDMO 算法的性能进行测试^[14]。表 1 为 13 个标准测试函数的基本信息。其中,F1~F5 是单峰函数,只有一个全局最优解,对算法的开发能力进行检验。F6~F10 是多峰函数,有一个全局最优解和多个局部最优解,可对算法的全局搜索能力进行检测。F11~F13 是固定维度多峰函数,对算法探索和开发能力之间的平衡性进行验证。通过上述三种不同类型的测试函数,充分验证算法的寻优性能。

表 1 标准测试函数
Table 1 Benchmark functions

Fun	dim	Range	f_{min}
$F_1(x) = \sum_{i=1}^n x_i^2$	30/100/500	[-100,100]	0
$F_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	30/100/500	[-10,10]	0
$F_3(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	30/100/500	[-100,100]	0
$F_4(x) = \max\{ x_i , 1 \leq i \leq n\}$	30/100/500	[-100,100]	0
$F_5(x) = \sum_{i=1}^n (x_i + 5)^2$	30/100/500	[-100,100]	0
$F_6(x) = \sum_{i=1}^n [x_i^2 - 10\cos(2\pi x_i) + 10]$	30/100/500	[-5.12,5.12]	0
$F_7(x) = -20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2} - \exp(\frac{1}{n}\sum_{i=1}^n \cos(2\pi x_i))) + 20 + e$	30/100/500	[-32,32]	0
$F_8(x) = \frac{1}{400}\sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1$	30/100/500	[-600,600]	0
$F_9(x) = \frac{\pi}{n}\{10\sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10\sin^2(\pi y_{i+1})] + (y_n - 1)^2\}$ $+ \sum_{i=1}^n u(x_i, 10, 100, 4)$, where $y_i = 1 + \frac{x_i + 1}{4}$, $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	30/100/500	[-50,50]	0
$F_{10}(x) = 0.1(\sin^2(3\pi x_1) + \sum_{i=1}^n (x_i - 1)^2 [1 + \sin^2(3\pi x_i + 1)])$ $+ (x_n - 1)^2 [1 + \sin^2(2\pi x_n)] + \sum_{i=1}^n u(x_i, 5, 100, 4)$	30/100/500	[-50,50]	0
$F_{11}(x) = (\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6})^{-1}$	2	[-65,65]	1
$F_{12}(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right]^2$	4	[-5,5]	0.00030
$F_{13}(x) = -\sum_{i=1}^5 [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0,10]	-10.1532

3.1 对比算法与参数设置

为充分验证本文提出的 IDMO 算法的性能,选取了 7 种算法进行对比,分别是,DMO^[9]、AOA^[1]、SCA^[2]、GWO^[3]、SSA^[4]、SO^[5]和 ROA^[6],这些算法已被证实具有良好的寻优性能。为了更准确地验证 IDMO 算法与对比算法的优劣性,统一设定种群规模 $N = 30$,维度 $D = 30$,最大迭代次数 500 次,各算法独立运行 30 次。选取平均值、标准差与 Wilcoxon 秩和检验作为评价标

准。平均值与标准差越小,算法的性能越好。表 2 为各对比算法的参数设置。

表 2 算法参数设置
Table 2 Algorithm Parameter Setting

算法	参数设置
IDMO	$nBabysitter = 3, peep = 2$
	$k = (1 + (\frac{3 * t}{T})^{0.5})^8$
DMO	$nBabysitter = 3, peep = 2$
AOA	$c1 = 2, c2 = 6, c3 = 2$

	$c4 = 0.5, u = 0.9, l = 0.1$
SCA	$r1 \in [0, 2], r2 \in [0, 2\pi],$ $r3 \in [0, 2], r4 \in [0, 1]$
GWO	$\alpha \in [0, 2]$
SSA	$r1 \in [0, 2], c1, c2 \in [0, 1],$ $m \in [0.2, 1]$
SO	$c1 = 0.5, c2 = 0.05, c3 = 2$
ROA	$c = 0.1$

3.2 求解精度分析

IDMO 算法及其对比算法的最优适应度值、平均适应度值和适应度值标准差如表 3 所示。Best 为最优适应度值，Mean 为平均适应度值，Std 为适应度值标准差。函数 F1-F5 为单峰函数；在求解函数 F1 时，IDMO 算法得到了理论最优值；在求解函数 F2 和 F4 时，IDMO 算法均得到了理论最优值，明显优于 DMO 和其他对比算法；在求解函数 F3 时，IDMO 算法得到了理论最优值，ROA 虽得到了最优适应度值和适应度值标准差的理论最优值，但其稳定性较差；在求解函数 F5 时，IDMO 算法得到了最优适应度值的理论最优值，SSA 虽得到了最佳的平均适应度值和适应度值标

准差，但其最优适应度值明显较差，表明 IDMO 算法跳出局部最优的能力较强；由此可知，引入透镜成像反向学习和精英池策略使得 IDMO 算法的局部开发能力显著提升，算法收敛精度有所提高。函数 F6~F10 为多峰函数；在求解函数 F6 和 F8 时，IDMO 算法和 ROA 均得到了理论最优值，SO 则得到了最优适应度值的理论最优值，但其稳定性较差；在求解函数 F7、F9 和 F10 时，IDMO 算法均得到了最优值，明显优于 DMO 和其余对比算法；由此可知，IDMO 算法的探索能力优于其他对比算法。函数 F11~F13 为固定维度多峰函数；在求解函数 F11 时，IDMO 算法虽未取得最优的适应度值标准差，但其最优适应度值和平均适应度值都是最优的；在求解函数 F12 和 F13 时，IDMO 算法均得到了最佳的最优适应度值、平均适应度值和适应度值标准差，GWO 和 ROA 均得到最佳的最优适应度值，但平均适应度值和适应度值标准差略差于 IDMO 算法；这验证了 IDMO 算法的收敛精度明显更高，且搜索能力和开发能力的平衡明显更优。

表 3 各算法标准函数测试结果

Table 3 Test results of benchmark functions of each algorithm

函数	统计值	IDMO	DMO	AOA	SCA	GWO	SSA	SO	ROA
F1	Best	0.00E+00	8.01E-04	8.71E-07	2.77E-03	9.44E-30	2.89E-08	2.02E-99	0.00E+00
	Mean	0.00E+00	1.57E-03	4.80E-06	1.07E+01	1.25E-27	1.70E-07	4.66E-94	0.00E+00
	Std	0.00E+00	4.38E-04	1.84E-06	1.03E+01	1.87E-27	2.51E-07	1.31E-93	0.00E+00
F2	Best	0.00E+00	3.31E-03	2.06E-09	1.59E-04	2.36E-17	2.84E-01	2.06E-45	1.30E-184
	Mean	0.00E+00	5.00E-03	1.63E-03	2.44E-02	9.49E-17	2.28E+00	3.72E-43	2.99E-162
	Std	0.00E+00	1.02E-03	1.68E-03	2.73E-02	4.98E-17	1.96E+00	8.25E-43	1.09E-161
F3	Best	0.00E+00	9.24E+01	9.60E-07	1.00E+03	1.89E-08	1.96E+02	3.74E-65	0.00E+00
	Mean	0.00E+00	2.68E+02	9.36E-04	9.49E+03	2.57E-06	1.73E+03	9.72E-57	1.18E-278
	Std	0.00E+00	1.23E+02	6.35E-04	5.32E+03	4.70E-06	1.03E+03	3.87E-56	0.00E+00
F4	Best	0.00E+00	2.88E+00	8.26E-04	1.87E+01	9.08E-08	5.41E+00	1.59E-43	6.06E-180
	Mean	0.00E+00	1.40E+01	1.82E-02	3.61E+01	8.63E-07	1.13E+01	2.54E-40	7.39E-161
	Std	0.00E+00	6.68E+00	1.38E-02	9.71E+00	8.12E-07	3.31E+00	4.70E-40	2.92E-160
F5	Best	0.00E+00	3.66E-03	2.72E+00	4.49E+00	5.36E-05	2.84E-08	3.15E-02	7.38E-03
	Mean	1.16E-06	7.34E-03	3.03E+00	1.42E+01	6.50E-01	1.75E-07	8.24E-01	6.91E-02

	Std	4.59E-06	2.09E-03	2.06E-01	1.53E+01	3.82E-01	2.81E-07	6.64E-01	6.05E-02
	Best	0.00E+00	9.93E+01	3.94E-11	1.66E-01	5.68E-14	2.49E+01	0.00E+00	0.00E+00
F6	Mean	0.00E+00	1.37E+02	1.77E-06	4.37E+01	2.38E+00	5.67E+01	5.90E+00	0.00E+00
	Std	0.00E+00	1.38E+01	1.57E-06	3.18E+01	3.39E+00	1.84E+01	1.16E+01	0.00E+00
	Best	8.88E-16	9.93E-03	1.49E-05	1.48E-01	7.55E-14	1.65E+00	4.44E-15	8.88E-16
F7	Mean	8.88E-16	1.61E-02	4.28E-04	1.58E+01	1.01E-13	2.72E+00	6.08E-02	8.88E-16
	Std	0.00E+00	3.17E-03	1.80E-04	7.95E+00	1.79E-14	5.38E-01	3.33E-01	0.00E+00
	Best	0.00E+00	4.15E-04	9.85E-06	9.49E-02	0.00E+00	1.08E-03	0.00E+00	0.00E+00
F8	Mean	0.00E+00	3.58E-03	3.54E-04	9.11E-01	3.45E-03	1.61E-02	6.97E-02	0.00E+00
	Std	0.00E+00	3.95E-03	1.80E-03	3.88E-01	7.14E-03	1.05E-02	1.73E-01	0.00E+00
	Best	1.57E-32	7.70E-02	6.61E-01	1.03E+00	1.63E-02	3.42E+00	5.71E-05	1.49E-03
F9	Mean	1.63E-32	8.34E-01	7.42E-01	8.20E+03	4.79E-02	8.26E+00	1.12E-01	1.10E-02
	Std	1.86E-33	4.12E-01	3.25E-02	2.49E+04	2.44E-02	3.84E+00	2.11E-01	5.74E-03
	Best	1.35E-32	3.92E-06	2.77E+00	2.36E+00	2.04E-01	1.60E-02	1.33E-02	8.48E-02
F10	Mean	1.35E-32	1.57E-03	2.96E+00	2.93E+05	6.43E-01	2.03E+01	3.23E-01	3.08E-01
	Std	5.57E-48	2.20E-03	3.54E-02	1.05E+06	2.70E-01	1.55E+01	5.29E-01	2.17E-01
	Best	9.98E-01	9.98E-01	9.98E-01	9.98E-01	9.98E-01	9.98E-01	9.98E-01	9.98E-01
F11	Mean	9.98E-01	9.98E-01	1.11E+01	1.92E+00	5.44E+00	1.23E+00	1.17E+00	4.51E+00
	Std	5.83E-17	0.00E+00	3.29E+00	1.91E+00	4.24E+00	7.65E-01	9.00E-01	4.94E+00
	Best	3.08E-04	3.27E-04	3.11E-04	5.08E-04	3.08E-04	4.15E-04	3.09E-04	3.08E-04
F12	Mean	3.13E-04	5.36E-04	1.01E-02	1.06E-03	3.87E-03	2.84E-03	1.17E-03	4.79E-04
	Std	3.20E-06	1.14E-04	2.48E-02	3.80E-04	7.51E-03	5.95E-03	3.63E-03	2.61E-04
	Best	-1.02E+01	-1.02E+01	-1.02E+01	-6.50E+00	-1.02E+01	-1.02E+01	-1.02E+01	-1.02E+01
F13	Mean	-1.02E+01	-1.02E+01	-7.88E+00	-1.84E+00	-9.14E+00	-7.06E+00	-1.01E+01	-1.01E+01
	Std	4.90E-08	4.16E-04	2.67E+00	1.75E+00	2.06E+00	3.46E+00	3.32E-01	3.60E-02

3.3 收敛曲线分析

为了更直观地反映各算法的收敛速度和跳出局部最优的能力, 本文给出部分测试函数的收敛曲线如图4所示。对于函数 F1、F2、F3 和 F4, DMO、AOA、SCA、GWO、SSA 和 SO 的收敛精度都较低; ROA 在函数 F1 和 F3 中, 收敛精度较高, 但收敛速度过慢, 在函数 F2 和 F4 中, 收敛精度较低; IDMO 算法在迭代一开始就快速收敛, 且找到了理论最优值。对于函数 F5, IDMO 算法在迭代初期快速收敛, 收敛精度明显更高。单峰函数的收敛曲线证明了 IDMO 算法引入透镜成像反向学习策略和精英池策略有效提高了收敛

能力和全局探索能力。对于函数 F7、F8、F9 和 F10, IDMO 算法在迭代初期快速收敛, 且在函数 F8 中得到理论最优值, AOA 在函数 F7、F9 和 F10 中长期陷入局部最优。多峰函数 F7、F8、F9 和 F10 体现了 IDMO 算法具有卓越的探索能力。对于固定维度多峰函数 F12 和 F13, IDMO 算法在迭代前期快速收敛, 寻优时间均少于其它算法; 在函数 F13 中, AOA、SCA、GWO、SSA、SO、RAO 均多次陷入局部最优, DMO 算法的收敛速度略逊于 IDMO 算法; 体现了 IDMO 算法探索和开发之间平衡性的优越。

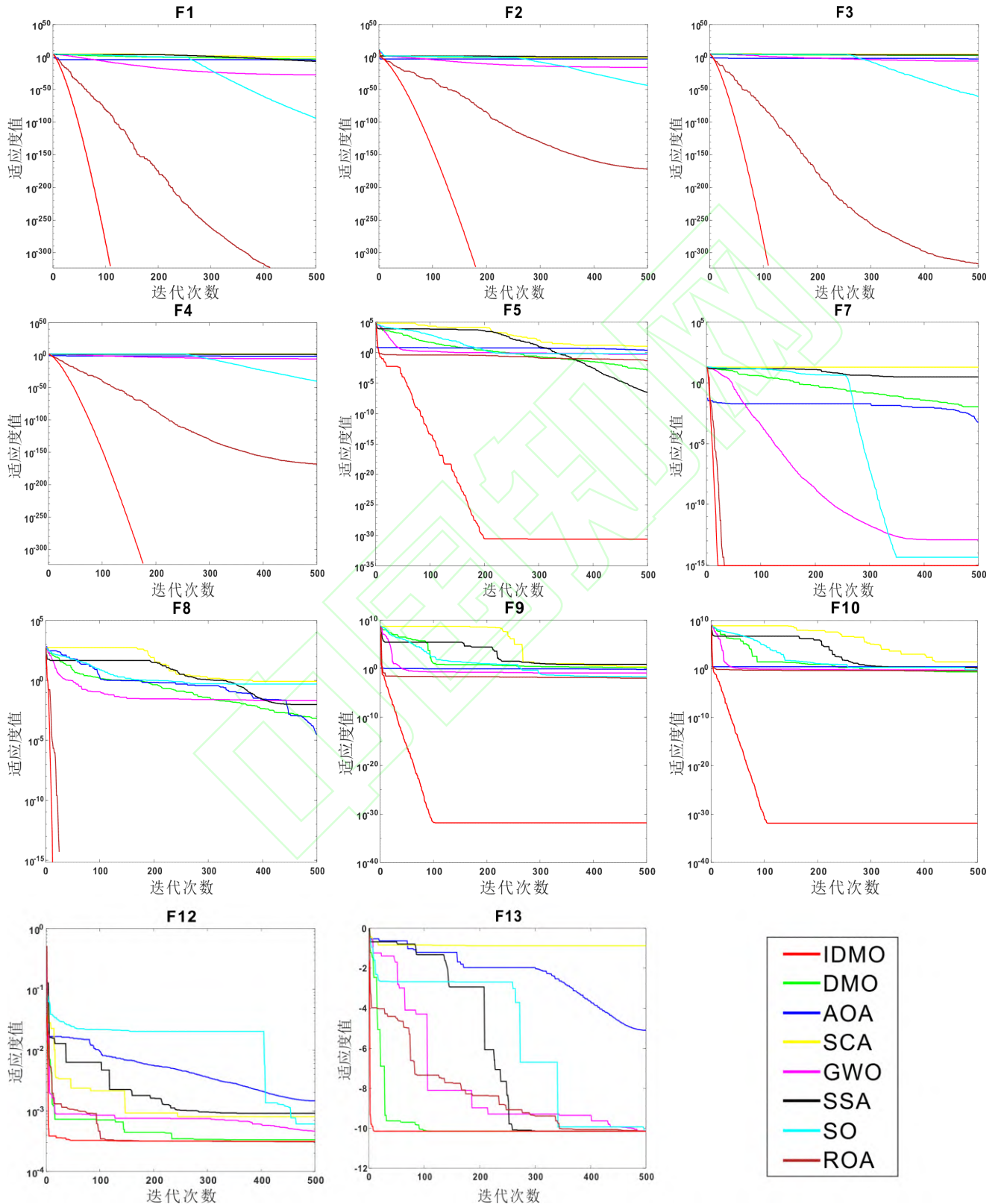


图 4 部分测试函数的收敛曲线
Fig.4 Convergence curve of partial test function

3.4 Wilcoxon 秩和检验

为了充分验证算法的鲁棒性,本文采用 Wilcoxon 秩和检验来验证各算法整体结果的显著性差别。Wilcoxon 秩和检验在 5% 的显著性水平下进行,当 $p < 5\%$ 时,表明两种对比算法存在显著差异,反之表明两种算法的寻优性能差异不大。因此,本文将 8 种算法作为样本,各算法独立求解 15 次,设定种群个数 N

$= 30$, 维度 $D = 30$, 测试 13 个标准测试函数判断 IDMO 算法所得结果与 7 个对比算法所得结果的显著性区别。Wilcoxon 统计检验 p 值结果如表 4 所示。由表 4 可知,大部分 p 值均小于 5%,说明 IDMO 算法与其余对比算法之间存在显著差异。

表 4 各算法 Wilcoxon 秩和检验结果
Table 4 Wilcoxon rank sum test results of each algorithm

函数	DMO	AOA	SCA	GWO	SSA	SO	ROA
F1	6.10E-05	6.10E-05	6.10E-05	6.10E-05	6.10E-05	6.10E-05	1.00E+00
F2	6.10E-05	6.10E-05	6.10E-05	6.10E-05	6.10E-05	6.10E-05	6.10E-05
F3	6.10E-05	6.10E-05	6.10E-05	6.10E-05	6.10E-05	6.10E-05	6.10E-05
F4	6.10E-05	6.10E-05	6.10E-05	6.10E-05	6.10E-05	6.10E-05	6.10E-05
F5	6.10E-05	6.10E-05	6.10E-05	6.10E-05	2.62E-03	6.10E-05	6.10E-05
F6	6.10E-05	6.10E-05	6.10E-05	6.10E-05	6.10E-05	1.95E-03	1.00E+00
F7	6.10E-05	6.10E-05	6.10E-05	6.10E-05	6.10E-05	6.10E-05	1.00E+00
F8	6.10E-05	6.10E-05	6.10E-05	3.13E-02	6.10E-05	1.56E-02	1.00E+00
F9	6.10E-05	6.10E-05	6.10E-05	6.10E-05	6.10E-05	6.10E-05	6.10E-05
F10	6.10E-05	6.10E-05	6.10E-05	6.10E-05	6.10E-05	6.10E-05	6.10E-05
F11	1.00E+00	6.10E-05	6.10E-05	6.10E-05	1.95E-03	3.91E-03	6.10E-05
F12	6.10E-05	3.05E-04	6.10E-05	2.62E-03	6.10E-05	6.10E-05	8.36E-03
F13	6.10E-05	6.10E-05	6.10E-05	6.10E-05	8.36E-03	8.36E-03	6.10E-05
+/-/-	12/1/0	13/0/0	13/0/0	13/0/0	13/0/0	13/0/0	9/4/0

通过综合分析各算法标准函数测试结果、部分测试函数的收敛曲线和各算法 Wilcoxon 秩和检验结果,可得出如下结论: IDMO 算法的局部和全局能力均显著提升,且优于原始 DMO 和 SSA 等对比优化算法,具有最佳的收敛速度、收敛精度以及稳定性。

4 汽车碰撞优化问题求解

汽车碰撞优化问题的设计目的是实现汽车轻量化,即在保证汽车安全性能的前提下,最大程度地降低汽车的整备质量^[15]。该问题包含了 11 个决策变量和 10 个约束条件,其中决策变量分别是 B 柱内部厚度、B 柱钢筋厚度、地板内侧厚度、横梁厚度、门梁厚度、门带线钢筋厚度、车顶纵梁厚度、B 柱内部材料、地板内侧材料、障碍高度、障碍撞击位置,约束条件分别是腹部负荷、上部粘性标准、中部粘性标准、下部粘性标准、上部肋骨偏转、中部肋骨偏转、下部肋骨偏转、耻骨联合力、B 柱中间点速度和 B 柱前门速度。

其数学模型如下:

目标函数:

$$f(\vec{x}) = \text{Weight},$$

约束条件:

$$g_1(\vec{x}) = F_a \leq 1 \text{ kN},$$

$$g_2(\vec{x}) = V \times Cu \leq 0.32 \text{ m/s},$$

$$g_3(\vec{x}) = V \times Cm \leq 0.32 \text{ m/s},$$

$$g_4(\vec{x}) = V \times Cl \leq 0.32 \text{ m/s},$$

$$g_5(\vec{x}) = \Delta_{ur} \leq 32 \text{ mm},$$

$$g_6(\vec{x}) = \Delta_{mr} \leq 32 \text{ mm},$$

$$g_7(\vec{x}) = \Delta_{lr} \leq 32 \text{ mm},$$

$$g_8(\vec{x}) = F_p \leq 4 \text{ kN},$$

$$g_9(\vec{x}) = V_{MBP} \leq 9.9 \text{ mm/ms},$$

$$g_{10}(\vec{x}) = V_{FD} \leq 15.7 \text{ mm/ms},$$

变量范围:

$$0.5 \leq x_1 - x_7 \leq 1.5, \quad x_8, x_9 \in (0.192, 0.345),$$

$$-30 \leq x_{10}, x_{11} \leq 30,$$

表5 汽车碰撞优化问题实验结果

Table 5 Experimental results of car crashworthiness optimization

算法	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	质量
IDMO	0.5	1.2319	0.5	1.2049	0.5	1.3517	0.51226	0.345	0.345	0.98992	-0.25418	23.2569
DMO	0.50806	1.2537	0.5	1.192	0.5	1.1038	0.50112	0.345	0.19963	1.444	1.6146	23.36
AOA	0.5	1.2345	0.87684	1.0283	0.5493	0.68626	0.5	0.34218	0.345	0.029599	-0.00024411	25.2582
SCA	0.51377	1.2521	0.5	1.2179	0.55909	1.5	0.53628	0.30441	0.19902	0.89322	-0.31884	23.682
GWO	0.50028	1.2459	0.50007	1.181	0.51821	1.1234	0.50158	0.345	0.31486	3.3122	3.1374	23.2599
SSA	0.5	1.2537	0.5	1.1715	0.67359	0.8231	0.5	0.32553	0.2355	2.7273	3.9311	23.5439
SO	0.5	1.2491	0.5	1.1896	0.5	0.97655	0.5	0.345	0.345	4.1885	9.1979	23.2773
ROA	0.881	1.1758	0.5	1.1263	0.88344	0.69298	0.5	0.29962	0.192	9.3283	5.9881	25.0862

IDMO 算法与对比算法求解汽车碰撞优化问题的实验结果如表 5 所示。从表中可直观看出, IDMO 算法得到的最小质量为 23.2569, 该结果不仅优于原始 DMO 算法, 且小于其余的对比优化算法, 说明 IDMO 算法在求解该类工程问题时具有良好的工程实用性。

5 总结

本文为提高 DMO 算法的收敛速度和寻优性能, 提出了一种融合透镜成像反向学习的精英池侏儒猫鼬优化算法 (IDMO)。使用 13 个标准测试函数及 Wilcoxon 秩和检验对 IDMO 算法进行仿真试验, 实验结果表明, IDMO 算法具有良好的寻优性能和鲁棒性, 再通过将其应用于汽车碰撞优化问题实例, 验证了 IDMO 算法在工程方面的适用性和可行性。在后续的研究中, 考虑将算法应用在其他工程问题求解中, 更深入地验证改进算法的性能。

参考文献:

- [1] ABUALIGAH L, DIABAT A, MIRJALILI S, et al. The arithmetic optimization algorithm[J]. Computer methods in applied mechanics and engineering, 2021, 376: 1136-1149.
- [2] MIRJALILI S. SCA: a sine cosine algorithm for solving optimization problems[J]. Knowledge-Based Systems, 2016, 96: 1-14.
- [3] MIRJALILI S, MIRJALILI S M, LEWIS A. Grey wolf optimizer[J]. Advances in Engineering Software, 2014, 69: 46-61.
- [4] MIRJALILI S, GANDOMI A H, MIRJALILI S Z, et al. Salp swarm algorithm: a bio-inspired optimizer for engineering design problems[J]. Advances in Engineering Software, 2017, 114: 163-191.
- [5] HASHIM F, HUSSINI A. Snake optimizer: a novel meta-heuristic optimization algorithm[J]. Knowledge-Based Systems, 2022, 242: 108-320.
- [6] JIA H M, PENG X X, LANG C B. Remora optimization algorithm[J]. Expert Systems With Applications, 2021, 185(9): 15665.
- [7] LI S M, CHEN H L, WANG M J, et al. Slime mould algorithm: a new method for stochastic optimization[J]. Future Generation Computer Systems, 2020, 111: 300-323.
- [8] KENNEDY J, EBERHART R. Particle swarm optimization[J]. Proceedings of IEEE International Conference on Neural Networks, 1995, 4: 1942-1948.
- [9] AGUSHAKA J O, EZUGWU A E, ABUALIGAH L. Dwarf mongoose optimization algorithm[J]. Computer methods in applied mechanics and engineering, 2022, 391(2022): 114570.
- [10] 陈功, 曾国辉, 黄勃, 刘瑾. 融合互利共生和透镜成像学习的 HHO 优化算法[J]. 计算机工程与应用, 2022, 58(10): 76-86.
- CHEN G, ZENG G H, HUANG B, LIU J. Harris hawk optimization algorithm combining mutualism and lens imaging learning[J]. Computer Engineering And Applications, 2022, 58(10): 76-86.
- [11] 周鹏, 董朝轶, 陈晓艳, 赵肖懿, 王启来. 基于 Tent 混沌和透镜成像学习策略的平衡优化器算法[J/OL]. 控制与决策 (2022-04-03) [2022-04-25]. <https://doi.org/10.13195/j.kzyjc.2021.1537>.
- ZHOU P, DONG C Y, CHEN X Y, ZHAO X Y, WANG Q L. An equilibrium optimizer algorithm based on a tent chaos and lens imaging learning strategy[J/OL]. Control and Decision(2022-04-03)[2022-04-25]. <https://doi.org/10.13195/j.kzyjc.2021.1537>.

- 13195/j. kzyjc.2021.1537.
- [12] LIU Q X, LI N, JIA H M, et al. Modified remora optimization algorithm for global optimization and multilevel thresholding image segmentation[J]. Mathematics, 2022, 10: 1014.
- [13] 汤安迪,韩统,徐登武,等. 使用高斯分布估计策略的改进樽海鞘群算法[J].系统工程与电子技术, 2022, 44(07): 2229-2240.
- TANG A D, HAN T, XU D W, et al. An improved salp swarm algorithm using Gaussian distribution estimation strategy[J]. Systems Engineering and Electronics, 2022, 44(07): 2229-2240.
- [14] 贾鹤鸣,刘宇翔,刘庆鑫,王爽,郑荣. 融合随机反向学习的黏菌与算术混合优化算法[J].计算机科学与探索, 2022, 16(05): 1182-1192.
- JIA H M, LIU Y X, LIU Q X, WANG S, ZHENG R. Hybrid algorithm of slime mould algorithm and arithmetic optimization algorithm based on random opposition-based learning[J]. Journal of Frontiers of Computer Science and Technology, 2022, 16(05): 1182-1192.
- [15] GU L, YANG R J, THO C H, MAKOWSK M, FARUQUE O, LI Y. Optimization and robustness for crashworthiness of side impact[J]. Vehicle Design, 2001, 26(4): 348-360.