



控制与决策

Control and Decision

ISSN 1001-0920, CN 21-1124/TP

## 《控制与决策》网络首发论文

题目: 混合改进策略的黑猩猩优化算法及其机械应用  
作者: 何庆, 罗仕杭  
DOI: 10.13195/j.kzyjc.2021.1108  
收稿日期: 2021-06-26  
网络首发日期: 2021-12-03  
引用格式: 何庆, 罗仕杭. 混合改进策略的黑猩猩优化算法及其机械应用[J/OL]. 控制与决策. <https://doi.org/10.13195/j.kzyjc.2021.1108>



**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。

## 混合改进策略的黑猩猩优化算法及其机械应用

何庆<sup>†</sup>, 罗仕杭

(贵州大学 大数据与信息工程学院, 贵州 贵阳 550025)

**摘要:** 针对黑猩猩优化算法存在易陷入局部最优、收敛速度慢、寻优精度低等缺陷, 提出了混合改进策略的黑猩猩优化算法(SLWChOA)。首先, 利用Sobol序列初始化种群, 增加种群的随机性和多样性, 为算法全局寻优奠定基础; 其次, 引入基于凸透镜成像的反向学习策略, 将其应用到当前最优个体上产生新的个体, 提高算法的收敛精度和速度; 同时, 将水波动态自适应因子添加到攻击者位置更新处, 增强算法跳出局部最优的能力; 最后, 通过10个基准测试函数、Wilcoxon秩和检验以及部分CEC2014函数进行仿真实验来评价改进算法的寻优性能, 实验结果表明所提算法在寻优精度、收敛速度和鲁棒性上均较对比算法有较大提升。另外, 通过一个机械优化设计实验进行测试分析, 进一步验证SLWChOA的可行性和适用性。

**关键词:** 黑猩猩优化算法; Sobol序列; 凸透镜成像; 水波动态自适应因子; 机械优化设计

中图分类号: TP301

文献标志码: A

DOI: 10.13195/j.kzyjc.2021.1108

开放科学(资源服务)标识码(OSID):



## Chimp optimization algorithm based on hybrid improvement strategy and its mechanical application

HE Qing<sup>†</sup>, LUO Shi-hang

(School of Big Data and Information Engineering, Guizhou University, Guiyang Guizhou 550025, China)

**Abstract:** Chimp optimization algorithms based on convex lens imaging strategy is proposed overcome the drawbacks of easily trapping into local optimum, slow convergence speed and low optimization precision. Firstly, the population was initialized by the Sobol sequence, which increase the randomness and diversity of the population, and lay the foundation for the global optimization of the algorithm. Then, introduce opposition-based learning strategy based on convex lens imaging applying it to the current optimal individual to generate new individuals, and improve the convergence accuracy and speed of the algorithm. At the same time, the water wave dynamic adaptive factor is added to the attacker's location update to enhance the ability of the algorithm to escape from the local optimum. Finally, The Simulation experiments are conducted on the 10 benchmark functions, Wilcoxon rank sum test and some part of CEC2014 functions to evaluate the optimization performance of the improved algorithm. The experimental results show that the proposed algorithm has more significant improvement in optimization accuracy, convergence speed and robustness than the comparison algorithm. In addition, three mechanical optimization design experiments are conducted to test and analyze the feasibility and applicability of the improved algorithm.

**Keywords:** Chimp optimization algorithm; Sobol sequence; convex lens imaging; water wave dynamic adaptive factor; mechanical optimization design

## 0 引言

在工程应用和科学研究中存在着大量高维、非线性、目标函数不可导的复杂优化问题, 元启发式算法具有参数少、易操作、不需要梯度信息等优点, 可以很好地在合理的时间且高度复杂的约束下找到这些问题的最优解。因此, 近年来元启发式算法在生产调度<sup>[1]</sup>、速度预测<sup>[2]</sup>、故障诊断<sup>[3]</sup>等领域得到

了广泛地研究和应用。

黑猩猩优化算法(Chimp Optimization Algorithm)是由M. Khishe等<sup>[4]</sup>于2020年提出的一种新型元启发式优化算法。它源于对自然界中黑猩猩个体智力、性动机及捕食行为的模拟, 通过驱赶、追逐、攻击等过程, 构建一种有效的优化方案。虽然ChOA具有原理简单、易操作、需调节参数少等特点, 但

收稿日期: 2021-06-26; 修回日期: 2021-10-27.

基金项目: 国家自然科学基金项目(62166006); 贵州省科技计划项目重大专项项目(黔科合重大专项字[2021] 335, 黔科合重大专项字[2018] 3002); 贵州省科学技术厅(黔科合基础-ZK[2021] 一般335).

<sup>†</sup>通讯作者. E-mail: qhe@gzu.edu.cn.

是ChOA与其他群体智能优化算法相似,存在求解精度低、收敛速度慢和易陷入局部极值等不足。为改善上述缺陷,宁杰琼等<sup>[5]</sup>针对BOA收敛精度低的问题,在个体位置更新处添加自适应余切权重系数,使算法收敛速度和精度得到提高;龙文等<sup>[6]</sup>提出一种新的参数 $C$ 策略,达到平衡GWO的探索 and 开发能力的效果;Kaur等<sup>[7]</sup>将正余弦函数引入ChOA个体位置更新处,提高算法的收敛性能;He等<sup>[8]</sup>将多项式变异用于ChOA种群初始化,提高种群的多样性和初始阶段解的质量。

虽然上述文献中的改进策略在一定程度上提高算法收敛精度和速度,但算法仍存在易陷入局部极值、“早熟”收敛等问题。为此,本文提出混合改进策略的黑猩猩优化算法(SLWChOA)。首先,采用Sobol序列产生均匀分布的种群,增加初始种群的多样性和随机性和提高求解效率;其次,引入基于凸透镜成像原理的反向学习策略,将产生的解与当前最优解结合生成更优解,增强群体之间信息交流和学习,提高算法的寻优性能;最后,在攻击者个体位置更新处加入水波动态自适应因子,扩大群体的寻优范围,降低算法陷入局部最优的概率。通过对10个基准测试函数及其Wilcoxon秩和检验和部分CEC2014函数进行仿真实验及1个实际工程问题验证SLWChOA算法的有效性和可行性。

## 1 黑猩猩优化算法

ChOA的灵感来源于自然界中黑猩猩的狩猎行为而衍生出的一种优化算法,黑猩猩根据分工采取不同的行动寻找猎物。标准的ChOA算法将黑猩猩群体分为四种类型:攻击者、障碍者、驱赶者和追逐者,其中攻击者是种群的领导者,其他三类黑猩猩协助狩猎,社会地位依次下降。黑猩猩驱逐和追逐猎物的数学模型如式(1)和(2)所示:

$$d = |C \cdot X_{prey}(t) - m \cdot X_{chimp}(t)| \quad (1)$$

$$X_{chimp}(t+1) = X_{prey}(t) - a \cdot d \quad (2)$$

式中:  $t$ 表示当前迭代次数;  $X_{prey}$ 为猎物位置向量;  $X_{chimp}$ 为当前黑猩猩位置向量;  $a, m, C$ 为系数向量,计算公式如式(3)~(5)所示:

$$a = 2f \cdot r_1 - f \quad (3)$$

$$m = Chaotic\_value \quad (4)$$

$$C = 2 \cdot r_2 \quad (5)$$

式中:  $r_1$ 和 $r_2$ 分别为[0,1]间的随机向量;  $f$ 为收敛因子,其值随迭代次数增加从2.5非线性减小到0;  $a$ 是

决定黑猩猩与猎物距离的随机向量,其值为 $[-f, f]$ 之间的随机数,  $m$ 为混沌映射矢量,代表黑猩猩在狩猎过程中性动机的影响;  $C$ 为黑猩猩驱逐和追赶猎物的控制系数,其值为[0,2]之间的随机数。

种群初始化后,依次选出四个最优解作为攻击者、障碍者、驱赶者和追逐者的位置,种群中的其他黑猩猩位置围绕以下四种黑猩猩的位置进行更新,其数学模型描述如式(6)~(8)所示:

$$X_1 = X_{attacker} - a_1 \cdot |C_1 \cdot X_{attacker} - m_1 \cdot X| \quad (6)$$

$$X_2 = X_{barrier} - a_2 \cdot |C_2 \cdot X_{barrier} - m_2 \cdot X| \quad (7a)$$

$$X_3 = X_{chaser} - a_3 \cdot |C_3 \cdot X_{chaser} - m_3 \cdot X| \quad (7b)$$

$$X_4 = X_{driver} - a_4 \cdot |C_4 \cdot X_{driver} - m_4 \cdot X| \quad (7c)$$

$$X(t+1) = (X_1 + X_2 + X_3 + X_4)/4 \quad (8)$$

式中:  $X$ 表示当前黑猩猩的位置向量;  $X_{attacker}$ 表示攻击者的位置向量;  $X_{barrier}$ 表示障碍者的位置向量;  $X_{chaser}$ 表示追逐者的位置向量;  $X_{driver}$ 表示驱赶者的位置向量;  $X(t+1)$ 表示当前黑猩猩更新后的位置向量。

## 2 改进的黑猩猩优化算法

首先,在标准ChOA中,种群初始化采用随机分布的方式,这种方式造成种群多样性差,导致个体前期搜索存在一定的盲目性,使算法收敛速度慢;其次,标准ChOA通过对种群历史前四个最优位置的加权记忆,实现了个体在搜索空间快速寻优,但这一做法并未考虑到每个黑猩猩个体自身的搜索经验,导致算法寻优精度低;最后,标准ChOA的种群围绕攻击者、障碍者、驱赶者和追逐者进行位置更新,当这四个个体陷入局部空间极值时,种群会随之陷入局部最优,导致算法出现停滞搜索现象。

综上所述,本文针对上述标准ChOA原理的缺陷,引入对应的策略进行改进。首先,本文通过Sobol序列初始化种群,使种群分布更加均匀,降低前期个体搜索的盲目性,从而提高算法的收敛速度;其次,引用基于凸透镜成像的反向学习策略,通过对算法中当前最优个体进行凸透镜成像反向学习,将各维度的值映射到搜索空间中得到反向解,引导群体向全局最优解附近靠拢,提高算法的寻优精度;最后,在种群的引导者“攻击者”位置更新处添加水波动态自适应因子,利用水波的动态变化的方向和大小不确定性帮助攻击者引导种群在更广阔的范围搜索,增加算法寻优位置的多样性,避免趋同性,降低算法陷入局部最优的概率。具体策略介绍如下:

## 2.1 Sobol序列初始化种群

初始解在解空间的分布情况很大程度上影响元启发式算法的收敛速度和寻优精度。均匀分布的种群初始解可以有效提高算法的搜索效率。标准ChOA采用随机方式初始化种群,这种方法遍历性低且种群分布不均匀。本文采用Sobol序列初始化种群, Sobol序列是一种低差异序列,使用确定性拟随机数序列代替伪随机数序列的方法,将尽可能均匀的点填充至多维超立方体。因此, Sobol序列在解决概率问题时,计算效率更高,采样点覆盖率更广<sup>[9]</sup>。

设全局解的取值范围为 $[ub, lb]$ ,由Sobol序列产生的第 $i$ 个随机数为 $S_i \subseteq [0, 1]$ ,则种群初始位置表示如式(9)所示:

$$X_n = lb + S_i \cdot (ub - lb) \quad (9)$$

假设搜索空间为二维,上下界分别为0和1,种群规模为100,对比随机初始化种群空间分布和Sobol序列初始化种群空间分布如图1所示。

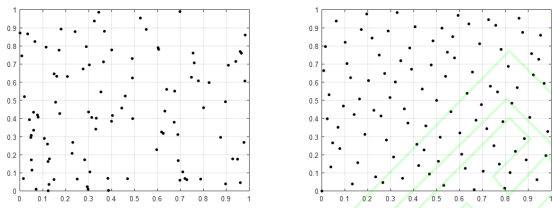


图1 不同方法产生的初始化种群分布图

如图1可知,通过Sobol序列产生的初始种群分布更加均匀,每个坐标区间都有个体分布其中,对解空间的覆盖率更广。

## 2.2 基于凸透镜成像的反向学习策略

由式(6)和(7)可知,种群中其他黑猩猩个体通过决策层个体(攻击者、障碍者、驱赶者和追逐者)的引导向最优解区域收敛。然而,当攻击者、障碍者、驱赶者和追逐者搜索到更好的猎物时,整个种群会聚集在这四类黑猩猩周围的搜索区域,导致种群多样性减少,从而使算法出现“早熟”收敛,寻优精度降低。为解决这一问题,本文提出一种基于凸透镜成像原理的反向学习策略,将其产生的反向个体与当前最优个体结合生成新的个体,具体数学模型如下所示:

黑猩猩在空间中寻找反向点的过程可以近似为凸透镜成像的过程。假设在一维空间中,在坐标轴区间 $[lb, ub]$ 上有一个高度为 $h$ 的个体 $P$ ,其在 $X$ 轴的投影为 $X$ ( $X$ 为全局最优个体),基点位置 $O$ (本文取基点位置为 $(lb + ub)/2$ )上放置焦距为 $F$ 的凸透镜,个

体 $P$ 通过凸透镜得到一个高度为 $h^*$ 的倒像 $P^*$ ,此时通过透镜成像在 $X$ 轴上产生第一个反向个体 $X^*$ ,成像原理如图2所示。

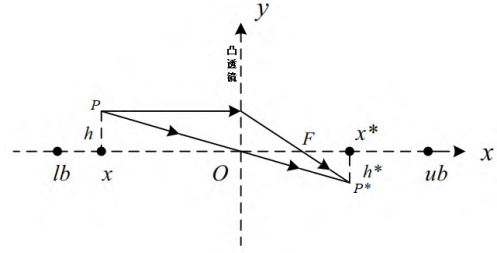


图2 基于凸透镜成像反向学习示意图

图2中,全局最优个体 $X$ 以 $O$ 为基点找到其对应的反向个体 $X^*$ ,则根据凸透镜成像原理可以得出:

$$\frac{(ub + lb)/2 - X}{X^* - (ub + lb)/2} = \frac{h}{h^*} \quad (10)$$

设 $h/h^* = \eta$ ,表示伸缩因子与物和像之间的对应比例关系,其中 $\eta$ 称为伸缩因子。对公式(10)进行变化即可得到反向点 $X^*$ 的计算公式:

$$X^* = \frac{ub + lb}{2} + \frac{ub + lb}{2 \cdot \eta} - \frac{X}{\eta} \quad (11)$$

在SLWChOA搜索解的过程中,伸缩因子 $\eta$ 作为微观调控因子加强算法的局部开发能力,在基本透镜成像反向学习策略中,伸缩因子一般作为固定值使用,不利于算法在解空间进行全方位的探索。本文提出一种基于非线性动态递减的伸缩因子策略,在算法迭代前期可以获得较大值,使算法可以在不同维度区域进行更大范围的搜索,提高群体的多样性;在算法迭代后期获得较小值,在最优个体附近进行精细搜索,提高局部寻优能力。非线性动态伸缩因子计算公式如式(12)所示:

$$\eta = \eta_{\max} - (\eta_{\max} - \eta_{\min}) \cdot (t/t_{\max})^2 \quad (12)$$

式中: $\eta_{\max}$ 和 $\eta_{\min}$ 分别为最大伸缩因子和最小伸缩因子; $t_{\max}$ 为最大迭代次数。

将式(11)推广至 $D$ 维搜索空间可得:

$$X_j^* = \frac{ub_j + lb_j}{2} + \frac{ub_j + lb_j}{2 \cdot \eta} - \frac{X_j}{\eta} \quad (13)$$

式中: $X_j$ 和 $X_j^*$ 分别为 $X$ 和 $X^*$ 的第 $j$ 维向量, $ub_j$ 和 $lb_j$ 分别为决策变量的第 $j$ 为向量。通过对算法中全局最优个体进行凸透镜成像反向学习,将各维度的值映射到空间中得到反向解,不仅避免各维度之间的干扰,而且扩大算法的搜索范围。虽然凸透镜的反向学习策略极大程度地提高算法的求解精度,在一定程度上帮助种群跳出局部最优,但是无法直接判断产生新的反向个体是否优于原始个体。因此,采用贪婪机制比较新旧个体适应度值,进而筛选出最优个体,通过这种方式不断获得更优解,



提高算法的寻优能力。其中贪婪机制的数学模型描述如式(14)所示:

$$X_{\text{new}}(t) = \begin{cases} X^* & f(X) \geq f(X^*) \\ X & f(X) < f(X^*) \end{cases} \quad (14)$$

### 2.3 水波动态自适应因子

在标准ChOA中,攻击者是种群的领导者,对于种群最终位置更新有着至关重要的作用。在全局探索过程中,攻击者引导种群探寻更广阔的搜索区域,一旦攻击者陷入小范围最优解时,种群将受其的影响陷入局部极值空间,导致算法出现停止搜索现象。因此,本文在攻击者位置更新公式中加入水波动态自适应因子,其迭代1000次的值如图3所示。利用水波的动态变化的不确定性使得攻击者能在更广泛的区域搜索,降低其他个体跟随的盲目性,增强群体间的信息交流和学习,保持种群多样性,有效避免了趋同性,进而提高算法跳出局部最优的能力。

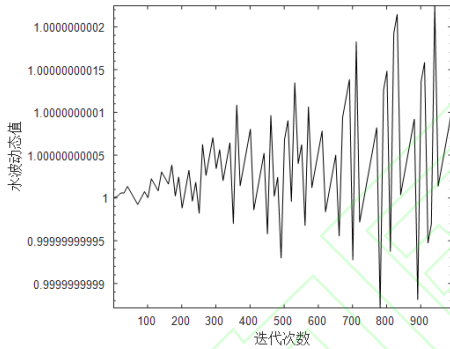


图3 迭代1000次水波动态值分布

水波动态自适应因子的数学模型如式(15)所示:

$$\lambda = 1 - \sin\left(\frac{\pi \cdot t}{2 \cdot t_{\max}} + 2 \cdot \pi\right) \quad (15)$$

ChOA引入水波动态自适应因子后攻击者位置更新公式为:

$$X_1 = \lambda \cdot X_{\text{attacker}} - a_1 \cdot |C_1 \cdot X_{\text{attacker}} - m_1 \cdot X| \quad (16)$$

### 2.4 SLWChOA实现步骤

综合上述改进方法,本文所提SLWChOA实现步骤如下所示:

step 1: 初始化算法相关参数: 种群规模 $N$ 、空间维度 $dim$ 、种群的可搜索空间 $[ub, lb]$ 、最大迭代次数 $t_{\max}$ ;

step 2: 按照公式(9)利用Sobol序列初始化种群;

step 3: 计算每个黑猩猩个体的适应度值,并选择适应度最小的前四个个体位置,分别记录为 $X_{\text{attacker}}$ 、 $X_{\text{barrier}}$ 、 $X_{\text{chaser}}$ 和 $X_{\text{driver}}$ ;

step 4: 按照公式(3)、(4)、(5)计算参数 $a$ 、 $m$ 、 $C$ 的值;

step 5: 按照公式(13)对 $X_{\text{attacker}}$ 进行凸透镜成像反向学习,产生反向解 $X_{\text{attacker}}^*$ ,根据式(14)贪婪机制选择适应度值低的解;

step 6: 按照公式(7a)、(7b)、(7c)和(16)更新 $X_{\text{barrier}}$ 、 $X_{\text{chaser}}$ 、 $X_{\text{driver}}$ 和 $X_{\text{attacker}}$ ,进一步根据公式(8)更新黑猩猩种群位置;

step 7: 判断是否满足迭代终止条件,满足则输出全局最优黑猩猩位置 $X_{\text{attacker}}$ ,否则进入step 3继续执行。

### 2.5 改进算法时间复杂度分析

时间复杂度间接反映算法的收敛速度。在ChOA算法中,假设初始化参数(种群规模为 $N$ 、搜索空间维度为 $n$ 、系数 $a$ 、 $m$ 、 $C$ 等参数)所需时间为 $\alpha_1$ ,每一维按式(8)更新种群其他黑猩猩个体位置所需时间为 $\alpha_2$ ,求解目标适应度函数时间为 $f(n)$ ,则标ChOA的时间复杂度为:

$$T_1(n) = O(\alpha_1 + N(n\alpha_2 + f(n))) = O(n + f(n)) \quad (17)$$

在SLWChOA算法中,初始化参数所需时间与标准ChOA保持一致,采用Sobol序列初始化种群所需时间为 $\alpha_3$ ,在算法循环阶段,设执行凸透镜成像反向学习策略所需时间为 $\alpha_4$ ,执行贪婪机制时间为 $\alpha_5$ ,每一维按照式(16)更新攻击者位置所需时间为 $\alpha_6$ ,则SLWChOA的时间复杂度为:

$$T_2(n) = O(\alpha_1 + \alpha_5 + N(n\alpha_3 + \alpha_4 + n\alpha_6 + f(n))) = O(n + f(n)) \quad (18)$$

本文所提SLWChOA和基本ChOA时间复杂度一致:

$$T_1(n) = T_2(n) = O(n + f(n)) \quad (19)$$

综上所述,本文针对ChOA缺陷所提改进策略没有增加时间复杂度。

## 3 仿真实验与结果分析

### 3.1 实验设计与测试函数

表1 基准测试函数

编号	名称	维度	定义域	理论值
F1	Sphere	30/200/500	[-100,100]	0
F2	Schwefel.2.21	30/200/500	[-100,100]	0
F3	Rosenbrock	30/200/500	[-30,30]	0
F4	Step	30/200/500	[-100,100]	0
F5	Quartic	30/200/500	[-1.28,1.28]	0
F6	Rastrigin	30/200/500	[-5.12,5.12]	0
F7	Ackley	30/200/500	[-32,32]	0
F8	Griewank	30/200/500	[-600,600]	0
F9	Foxholes	2	[-65.56,65.56]	1
F10	Goldstein-Price	2	[-2,2]	3

本文基于Intel(R) Core(TM) i7-i7-6500U CPU、2.50GHz主频、8GB内存以及Windows 10(64位)的操作系统对所提出的算法进行仿真实验。编程软件为MATLAB R2018a。表1为10个基准测试函数的基本信息, 其中5个单峰函数F1~F5, 3个非线性多峰函数F6~F8和2个固定维度的多峰函数F9~F10。表2为每个对比算法的参数设置。

表2 参数设置表

算法	主要参数
ChOA	$r_1 \in [0, 1], r_2 \in [0, 1], f \in (0, 2.5), u \in [0, 1]$
SChOA	$r_1 \in [0, 1], r_2 \in [0, 1], f \in (0, 2.5), u \in [0, 1]$
LChOA	$r_1 \in [0, 1], r_2 \in [0, 1], f \in (0, 2.5), u \in [0, 1]$ $\eta_{\max} = 10, \eta_{\min} = 1$
WChOA	$r_1 \in [0, 1], r_2 \in [0, 1], f \in (0, 2.5), u \in [0, 1]$
SLWChOA	$r_1 \in [0, 1], r_2 \in [0, 1], f \in (0, 2.5), u \in [0, 1]$ $\eta_{\max} = 10, \eta_{\min} = 1$

### 3.2 不同改进策略对算法性能影响分析

为充分验证SLWChOA的有效性和优越性, 将基本ChOA与本文加入Sobol序列初始化的黑猩猩算法(SChOA)、加入凸透镜成像反向学习策略的黑猩猩算法(LChOA)、加入水波动态自适应因子的黑猩猩算法(WChOA)在10个具有不同寻优特征的基准测试函数上进行仿真实验对比。算法通用条件设置为相同, 种群大小 $N=30$ , 空间维度 $dim=30$ , 最大迭代次数 $t_{max}=1000$ 。表3通过最优值、最差值、平均值和标准差四个性能指标来评估每个算法的寻优性能。五种算法对10个基准测试函数的求解结果如表3所示:

表3通过最优值和平均值来反映算法的寻优性能和收敛精度的能力。首先, 从最优值看, SLWChOA 对F1、F2、F6和F8求解时, 获得理论值。对于函数F3~F5, 其形状类似于抛物面, 存在大量局部最优值, 同时F7是具有山谷状的多峰函数, 其全局最优值位于山低端比较难寻, 因此所提改进算法与其他对比算法均未寻到理论值, 但是SChOA、LChOA和WChOA的寻优精度和稳定性优于标准ChOA, 说明不同改进策略对算法的寻优性能有不同程度的提高。其次, 从平均值看, 仅采用Sobol序列初始化策略(SChOA)对ChOA性能的改进有限, 而LChOA算法在函数F1、F2、F6和F8均能寻到理论值, 这是因为凸透镜成像的反向学习策略产生将反向解应用到当前最优解上, 其引导种群向全局最优区域靠拢, 提高了算法的寻优精度和收敛速度。在固定维度的多峰函数F9上, SLWChOA寻优精度达到最高, 对于函数F10, SChOA、LChOA、WChOA与SLWChOA均能寻到理论值, 且SLWChOA的标准差表现排名

第一, SChOA、WChOA、LChOA的标准差表现分别位居第二至第四, 均优于标准ChOA, 表明SLWChOA稳定性和鲁棒性更好。综上所述, 所提改进策略对标准ChOA具有明显的优化效果, 融合三种策略的SLWChOA无论在寻优精度还是稳定性上均表现出一定的优势。

表3 基准测试函数求解结果对比

函数	算法	最优值	最差值	平均值	标准差
F1	ChOA	1.52E-23	1.09E-12	6.18E-13	2.07E-13
	SChOA	2.81E-34	3.96E-13	2.09E-14	8.07E-14
	LChOA	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	WChOA	7.44E-95	4.69E-89	2.91E-90	9.04E-90
	SLWChOA	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>
F2	ChOA	1.52E-06	2.13E-02	2.44E-03	4.52E-03
	SChOA	2.52E-07	1.03E-02	1.65E-04	2.46E-04
	LChOA	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	WChOA	8.56E-41	3.25E-37	3.36E-38	7.57E-38
	SLWChOA	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>
F3	ChOA	2.81E+01	2.90E+01	2.88E+01	2.24E-01
	SChOA	2.87E+01	2.90E+01	2.89E+01	8.24E-02
	LChOA	2.72E+01	2.90E+01	2.78E+01	3.65E-01
	WChOA	2.72E+01	2.90E+01	2.80E+01	4.19E-01
	SLWChOA	<b>2.63E+01</b>	<b>2.88E+01</b>	<b>2.68E+01</b>	<b>3.06E-01</b>
F4	ChOA	2.38E+00	3.77E+00	3.20E+00	3.59E-01
	SChOA	2.04E+00	3.80E+00	3.10E+00	3.54E-01
	LChOA	1.56E+00	2.81E+00	2.33E+00	3.37E-01
	WChOA	2.54E+00	3.66E+00	2.67E+00	3.22E-01
	SLWChOA	<b>1.04E+00</b>	<b>2.50E+00</b>	<b>1.37E+00</b>	<b>2.47E-01</b>
F5	ChOA	5.19E-05	3.85E-03	7.76E-04	7.50E-04
	SChOA	9.89E-05	3.38E-03	1.05E-04	9.09E-04
	LChOA	9.86E-06	5.84E-04	2.10E-04	1.44E-04
	WChOA	6.93E-06	5.90E-04	1.01E-04	1.11E-04
	SLWChOA	<b>1.74E-05</b>	<b>6.37E-04</b>	<b>1.71E-05</b>	<b>1.44E-05</b>
F6	ChOA	4.89E-12	3.30E+01	3.85E+00	6.23E+00
	SChOA	4.88E-11	5.51E+00	1.69E+00	1.83E+00
	LChOA	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	WChOA	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	SLWChOA	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>
F7	ChOA	2.00E+01	2.00E+01	2.00E+01	1.36E-03
	SChOA	3.72E-07	2.00E+01	1.85E+00	5.02E+00
	LChOA	8.88E-16	8.88E-16	8.88E-16	0.00E+00
	WChOA	2.00E+00	2.00E+00	2.00E+00	1.47E-03
	SLWChOA	<b>8.88E-16</b>	<b>8.88E-16</b>	<b>8.88E-16</b>	<b>0.00E+00</b>
F8	ChOA	2.22E-16	6.41E-02	1.53E-02	1.87E-02
	SChOA	0.00E+00	5.09E-02	8.02E-03	1.38E-03
	LChOA	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	WChOA	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	SLWChOA	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>
F9	ChOA	9.98E-01	9.98E-01	9.98E-01	1.83E-05
	SChOA	9.98E-01	9.98E-01	9.98E-01	1.23E-05
	LChOA	9.98E-01	9.98E-01	9.98E-01	4.95E-05
	WChOA	9.98E-01	9.98E-01	9.98E-01	2.55E-05
	SLWChOA	<b>9.99E-01</b>	<b>9.98E-01</b>	<b>9.99E-01</b>	<b>2.31E-05</b>
F10	ChOA	3.00E+00	3.00E+00	3.00E+00	9.38E-05
	SChOA	3.00E+00	3.00E+00	3.00E+00	9.30E-05
	LChOA	3.00E+00	3.00E+00	3.00E+00	6.30E-05
	WChOA	3.00E+00	3.00E+00	3.00E+00	7.99E-05
	SLWChOA	<b>3.00E+00</b>	<b>3.00E+00</b>	<b>3.00E+00</b>	<b>5.62E-05</b>

表4 不同维度下各算法对单峰函数与多峰函数的求解结果比较

函数	算法	30dim		200dim		500dim	
		平均值	标准差	平均值	标准差	平均值	标准差
F1	PSO	1.20E-04	1.81E-04	3.78E-06	3.35E-06	5.91E+03	4.79E+02
	SChOA <sup>[7]</sup>	1.57E+03	5.68E+03	—	—	—	—
	EChOA <sup>[8]</sup>	1.22E-36	6.33E-36	—	—	—	—
	MSWOA	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	LILGWO	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	SLWChOA	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>
F2	PSO	1.14E+00	2.39E-01	9.46E+01	4.12E+00	2.79E+01	1.65E+00
	SChOA <sup>[7]</sup>	1.22E+01	1.27E+00	—	—	—	—
	EChOA <sup>[8]</sup>	2.57E-16	8.96E-16	—	—	—	—
	MSWOA	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	LILGWO	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	SLWChOA	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>
F3	PSO	7.43E+01	4.66E+01	1.98E+02	2.83E-01	3.67E+03	3.26E+06
	SChOA <sup>[7]</sup>	5.95E+05	1.00E+07	—	—	—	—
	EChOA <sup>[8]</sup>	<b>1.04E+01</b>	<b>8.21E-01</b>	—	—	—	—
	MSWOA	2.86E+01	5.71E-01	1.98E+02	6.36E-01	8.80E+02	4.67E+00
	LILGWO	2.88E+01	8.82E-01	1.98E+02	5.37E-01	8.01E+02	1.70E+00
	SLWChOA	2.70E+01	3.17E-01	<b>1.46E+02</b>	<b>5.78E-14</b>	<b>4.02E+02</b>	<b>2.89E-14</b>
F4	PSO	<b>2.51E-04</b>	<b>7.77E-04</b>	6.10E+04	2.26E+03	6.01E+04	2.52E+03
	SChOA <sup>[7]</sup>	1.01E+03	5.62E+03	—	—	—	—
	EChOA <sup>[8]</sup>	1.10E+00	4.33E-01	—	—	—	—
	MSWOA	2.64E+00	3.47E-01	3.54E+01	1.94E+00	8.89E+01	3.57E+00
	LILGWO	2.66E+00	5.00E-01	3.33E+01	1.17E+00	1.01E+02	1.74E+00
	SLWChOA	1.63E+00	2.58E-01	<b>3.14E+01</b>	<b>1.45E-14</b>	<b>8.00E+01</b>	<b>2.89E-14</b>
F5	PSO	1.90E-01	7.80E-02	6.08E+04	2.20E+03	6.03E+04	1.72E+03
	SChOA <sup>[7]</sup>	1.44E+00	5.77E+00	—	—	—	—
	EChOA <sup>[8]</sup>	<b>1.09E-06</b>	<b>7.99E-07</b>	—	—	—	—
	MSWOA	5.99E-01	3.72E-01	1.20E+00	0.90E+00	6.69E-01	3.32E-01
	LILGWO	3.39E-04	3.46E-04	5.66E-04	6.54E-04	3.86E-04	2.86E-04
	SLWChOA	1.73E-05	1.21E-05	<b>2.87E-05</b>	<b>2.18E-05</b>	<b>3.30E-05</b>	<b>2.82E-05</b>
F6	PSO	5.53E+01	1.13E+01	6.02E+04	2.10E+03	5.92E+04	3.31E+03
	SChOA <sup>[7]</sup>	4.40E+01	7.77E+01	—	—	—	—
	EChOA <sup>[8]</sup>	1.41E-01	6.05E-01	—	—	—	—
	MSWOA	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	LILGWO	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	SLWChOA	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>
F7	PSO	4.06E-01	6.37E-01	5.97E+04	2.82E+03	5.94E+04	2.08E+03
	SChOA <sup>[7]</sup>	1.89E+00	1.71E+00	—	—	—	—
	EChOA <sup>[8]</sup>	1.94E-14	3.63E-15	—	—	—	—
	MSWOA	8.88E-16	0.00E+00	8.88E-16	0.00E+00	8.88E-16	0.00E+00
	LILGWO	8.88E-16	0.00E+00	8.88E-16	0.00E+00	8.88E-16	0.00E+00
	SLWChOA	<b>8.88E-16</b>	<b>0.00E+00</b>	<b>8.88E-16</b>	<b>0.00E+00</b>	<b>8.88E-16</b>	<b>0.00E+00</b>
F8	PSO	8.00E-03	8.32E-03	6.03E+04	3.06E+03	6.01E+04	2.13E+03
	SChOA <sup>[7]</sup>	2.09E+01	8.30E+01	—	—	—	—
	EChOA <sup>[8]</sup>	3.10E-03	7.33E-03	—	—	—	—
	MSWOA	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	LILGWO	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	SLWChOA	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>

### 3.3 SLWChOA收敛性分析

为了反映SLWChOA的动态收敛特性, 图4(a)~(d)给出了4个基准测试函数的平均收敛曲线图。

由图4可知, LChOA和WChOA求解单峰函数时F1、F2和多峰函数F6时, 其寻优精度均比标准ChOA分别至少提高200个数量级和30个数量级, 这归功于凸透镜成像反向学习策略产生的更优解

引导种群稳定寻到全局最优解, 提高算法的收敛精度和速度, 同时在水波动态自适应因子的协助下, “攻击者”个体带领种群向更大的猎物搜索范围寻找最优解, 保持种群多样性的同时帮助算法跳出局部最优。SLWChOA求解函数F5时陷入局部空间极值且难以跳出, 同时其他对比算法也出现停滞寻优, 但SLWChOA的平均收敛曲线均位于4种算法



平均收敛曲线下方,达到特定精度所需的迭代次数最少。具体来说,SLWChOA在迭代前期的搜索性能和迭代末期的开拓性能都优于其他对比算法,在相同的迭代次数下具有更高的求解精度和更快的收敛速度,表明SLWChOA在保证开拓能力的同时也能充分保证探索能力,不失群体多样性和寻优稳定性。

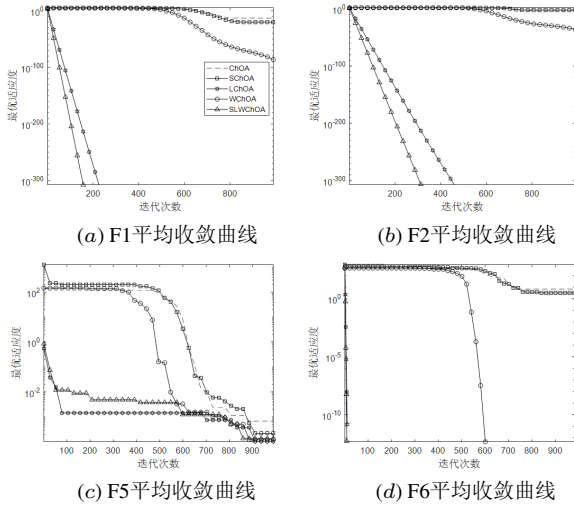


图4 不同策略算法平均收敛曲线

综上,表3的实验结果与图4的平均曲线验证了本文所提改进算法的有效性。虽然在某些函数上收敛精度差距不明显,但收敛速度远高于其他比较算法。SLWChOA的综合寻优能力比其他对比算法更强,稳定性更高。

### 3.4 SLWChOA与群智能算法及改进算法对比

为了更进一步验证其有效性,将SLWChOA与其他改进的蝴蝶优化算法(MSWOA, Mixed strategy to improve Butterfly Optimization Algorithm)<sup>[5]</sup>、改进的灰狼优化算法(LILGWO, Grey Wolf Optimizer Algorithm Based on Lens Imaging Learning Strategy)<sup>[6]</sup>、最新改进黑猩猩优化算法(SChOA, Sine-cosine Chimp Optimization Algorithm<sup>[7]</sup>、EChOA, An Enhanced Chimp Optimization Algorithm<sup>[8]</sup>)以及经典的粒子群算法(Particle Swarm Optimization, PSO)<sup>[10]</sup>进行比较,引用文献[7, 8]实验数据,并复现文献[5, 6, 10]的实验。为了体现比较的公平性,采用与文献相同的实验参数设置(种群规模30,最大迭代次数500),表4和表5列出了每个算法对不同类型的基准测试函数的优化结果,其中单峰函数和多峰函数搜索维度分别设置为30/200/500,每个算法独立运行30次,记录其平均值和标准差(“—”为文献缺失数据)。

表4实验结果表明,SLWChOA在函数F1、F2、F6和F8上均寻到了理论最优值。在其它函数中表现不好的EChOA在F3和F5上表现发生了变化,获得

最高寻优精度,而SLWChOA的表现位居第二。在函数F5和F7中,大量局部极小点分布在其解空间中,算法对全局最优值的寻找有一定的困难,然而SLWChOA相较于其他改进算法具有更高的收敛精度和稳定性。当维度从30维上升到200维再上升到500维时,算法对求解精度和鲁棒性均有不同程度下降,这是因为随着维度增加,算法搜索范围不断扩大,寻优过程需要更多计算,但是相较于5种对比算法,SLWChOA求解精度在大部分函数上仍最高。因此,SLWChOA在求解低维和高维问题时具有极强的寻优性能和鲁棒性,进一步说明了SLWChOA在求解复杂函数优化问题时具有显著的竞争优势。

表5 各算法求解固定维度多峰函数的结果比较(dim=2)

函数	算法	平均值	标准差
F9	PSO	2.61E+00	2.04E+00
	SChOA <sup>[7]</sup>	1.33E+00	1.04E+01
	EChOA <sup>[8]</sup>	—	—
	MSWOA	1.79E+00	1.90E+00
	LILGWO	7.60E+00	4.73E+00
	SLWChOA	<b>9.99E-01</b>	<b>2.67E-02</b>
F10	PSO	3.00E+00	1.47E-05
	SChOA <sup>[7]</sup>	3.00E+00	1.92E-01s
	EChOA <sup>[8]</sup>	—	—
	MSWOA	3.00E+00	2.48E-05
	LILGWO	3.00E+00	2.67E-05
	SLWChOA	<b>3.00E+00</b>	<b>1.43E-05</b>

由表5可知,SLWChOA在2个固定维度的多峰函数上寻优精度都达到了最高。在函数F9上,SLWChOA获得最高寻优精度且标准差最低;PSO、LILGWO和SLWChOA在函数F10上收敛到理论最优值,但SLWChOA的标准差最小,说明SLWChOA具有更好的寻优稳定性。

### 3.5 改进算法与其他算法在不同维度上比较

本节基于10个基准测试函数,使用平均绝对误差(Mean Absolute Error, MAE)<sup>[11]</sup>来评估各算法性能差异的真实情况,维数选取30~510维,呈等差数列递增120维变化。通过MAE对算法进行排序,更进一步地验证算法的性能指标,避免因最优值近似而引起算法趋同性。实验参数设置种群规模30,最大迭代次数500,各算法独立运行30次。表6为基准测试函数的MAE排序结果,MAE的数学模型如式(20)所示:

$$MAE = \sum_{i=1}^{N_f} |\vartheta_i - \delta_i| / N_f \quad (20)$$

式中:  $\vartheta_j$ 表示算法最优值的平均值,  $\delta_i$ 表示每个基准函数的理论最优值,  $N_f$ 表示所选用基准函数的个数。



表 6 MAE算法排名

算法	30dim		150dim		270dim		390dim		510dim	
	MAE	排名	MAE	排名	MAE	排名	MAE	排名	MAE	排名
PSO	13.295	5	3.574E+04	4	2.449E+05	4	3.735E+05	4	3.297E+06	4
SChOA	5.97E+04	6	—	—	—	—	—	—	—	—
EChOA	<b>1.455</b>	<b>1</b>	—	—	—	—	—	—	—	—
MSWOA	3.262	3	17.453	2	31.674	2	35.015	2	99.859	2
LILGWO	3.806	4	17.910	3	32.340	3	35.830	3	101.560	3
SLWChOA	2.862	2	<b>13.170</b>	<b>1</b>	<b>22.120</b>	<b>1</b>	<b>32.220</b>	<b>1</b>	<b>53.750</b>	<b>1</b>

由表6可知，在30维中，EChOA的MAE最小，而SLWChOA的MAE表现位列第二。随着维度的不断增加，SLWChOA的MAE排名保持第一，说明SLWChOA的寻优性能不会随着复杂高维多极值的维度增加而陷入“维数灾难”，相较于其他对比算法具有更显著优势，表明所提算法的有效性。

3.6 改进算法在高维上的性能分析

本节选取与3.5节相同变化的维度，限于篇幅，

选用4个不同类型的函数(单峰函数F1、F5和多峰函数F6、F7)，将算法优化函数的平均值及其平均变化率作为评价指标，测试本文算法SLWChOA相较于ChOA独立运行30次的性能随函数维度的增加而变化的情况，实验参数设置种群规模30，最大迭代次数500，结果如表7所示。平均变化率的数学模型如式(21)所示：

$$\text{平均变化率} = \left( \sum \frac{\text{后一次维度的平均值} - \text{前一次维度的平均值}}{\text{前一次维度的平均值}} \right) / 4 \tag{21}$$

表 7 SLWChOA与ChOA在不同维度上优化函数平均值比较

函数	算法	维度					平均变化率(%)
		30	150	270	390	510	
F1	ChOA	7.71E-06	8.08E+00	9.54E+01	3.34E+02	7.73E+02	2.61E+05
	SLWChOA	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0</b>
F5	ChOA	1.47E-03	5.91E-02	3.30E-01	1.19E+00	3.11E+00	1.23E+01
	SLWChOA	<b>1.71E-05</b>	<b>2.76E-05</b>	<b>2.89E-05</b>	<b>3.42E-05</b>	<b>3.53E-05</b>	<b>2.19E-01</b>
F6	ChOA	1.42E+00	5.38E+01	1.41E+02	2.21E+02	3.33E+02	9.90E+00
	SLWChOA	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0</b>
F7	ChOA	1.98E+01	2.00E+01	2.00E+01	2.01E+01	2.01E+01	3.77E-03
	SLWChOA	<b>8.88E-16</b>	<b>8.88E-16</b>	<b>8.88E-16</b>	<b>8.88E-16</b>	<b>8.88E-16</b>	<b>0</b>

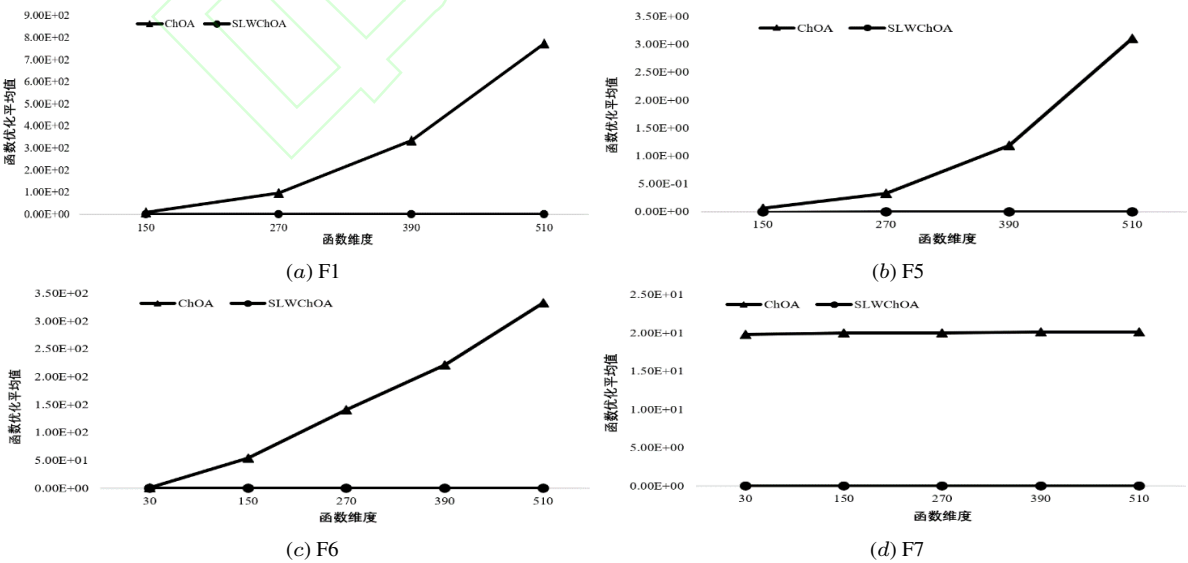


图 5 SLWChOA与ChOA优化函数平均值随函数维度变化曲线

由表7可知，随着维度的增加，SLWChOA的平均值基本保持不变。具体来说，对于函数F1、F6，

SLWChOA的平均值均能保持为0，然而ChOA随着维度的递增，平均值呈现不断增加趋势。在被大

量局部极小值包围的函数F5上, SLWChOA的平均值基本保持不变, 而ChOA的平均值变化幅度略大于SLWChOA。在函数F1、F6和F7上, SLWChOA的平均变化率均为0, 远低于ChOA的平均变化率。

图5是两种算法在不同维度下的平均值变化曲线图。从图5可以观察到, 在函数F1、F5和F6上, ChOA的平均值随维度递增而大幅度增加, 而SLWChOA的平均值保持不变。这说明维度的不断增加, 对SLWChOA的寻优能力影响较小, 相比于ChOA具有更突出的寻优性能, 进一步佐证本文所提算法的优越性。

### 3.7 Wilcoxon秩和检验

为了体现改进算法的有效性, 在显著性水平 $P=5\%$ 、维度为30下, 本文采用Wilcoxon秩和检验验证SLWChOA是否在统计上与标准ChOA与SChOA、LChOA、WChOA及新提出的元启发式算法AOA<sup>[12]</sup>和经典算法PSO存在明显差异。符号“+”, “-”, “=”分别表示SLWChOA性能优于、劣于和相当于对比算法, “NaN”表示不适用, 即无法进行显著性判断。结果如表8所示, 在10个基准测试函数中, 大部分 $P$ 值都小于5%, 总体上SLWChOA的性能与其他6种算法在统计上具有显著性差异, 从而表明SLWChOA比其他算法拥有更好的有效性。

表8 Wilcoxon秩和检验结果

函数	ChOA $P_1$	SChOA $P_2$	LChOA $P_3$	WChOA $P_4$	PSO $P_5$	AOA $P_6$
F1	3.31E-20	3.31E-20	3.31E-20	3.31E-20	1.84E-10	1.84E-10
F2	3.31E-20	3.31E-20	3.31E-20	3.31E-20	1.84E-10	1.84E-10
F3	3.30E-02	3.40E-02	3.87E-02	4.29E-02	4.04E-02	2.50E-05
F4	4.92E-02	4.39E-02	9.00E-02	1.54E-02	1.44E-02	6.32E-12
F5	3.36E-02	9.97E-02	1.63E-02	8.93E-02	7.07E-18	7.05E-07
F6	NaN	NaN	NaN	NaN	1.84E-10	NaN
F7	3.30E-20	3.08E-20	2.98E-20	3.31E-20	1.84E-10	2.70E-09
F8	NaN	NaN	NaN	NaN	1.84E-10	NaN
F9	1.81E-02	2.46E-02	3.87E-02	4.29E-02	1.95E-03	2.50E-05
F10	1.17E-02	1.77E-03	5.08E-03	1.99E-02	5.60E-07	2.35E-11
+/-	8/2/0	7/2/1	7/2/1	7/2/1	10/0/0	8/2/0

### 3.8 SLWChOA求解CEC2014测试函数

为了更好地验证SLWChOA的鲁棒性, 本文将SLWChOA与基本ChOA、PSO算法、MSBOA、LILGWO算法及L-SHADE算法<sup>[13]</sup>优化部分具有复杂特征的CEC2014测试函数, 其中包含单峰(CEC03)、多峰(CEC5、CEC6、CEC16)、混合(CEC19)和复合(CEC23)类型的函数, 选取的部分函数具体信息如表9所示。为保证对比的公平性, 最大迭代次数设置为1000, 每个算法分别独立运行30次, PSO算法和L-SHADE算法的数据来源于文献[14]。结果如表10所示:

表9 CEC2014测试函数(部分)

函数	维度	特征	定义域	最优值
CEC03	30	UN	[-100,100]	300
CEC5	30	MF	[-100,100]	500
CEC6	30	MF	[-100,100]	600
CEC16	30	MF	[-100,100]	1600
CEC19	30	HF	[-100,100]	1900
CEC23	30	CF	[-100,100]	2300

由表10可知, L-SHADE在单峰函数CEC03上表现出色, SLWChOA收敛精度低于L-SHADE, 因

为SLWChOA需要进行更多的参数计算, 造成收敛精度有所下降, 而SLWChOA的寻优性能略优于标准ChOA。在多峰(CEC5、CEC6、CEC16)和混合函数(CEC19)上, SLWChOA的寻优精度更加接近理论值且比ChOA稳定。在复合函数上, SLWChOA的收敛精度同LILGWO算法、MSBOA一致, 远高于比其他3个对比算法, 进一步证明了SLWChOA具有较好的寻优鲁棒性。

## 4 机械优化设计问题

优化问题作为工程设计与应用领域中经常出现的数值约束问题, 传统的机械方法难以解决非线性甚至高维的数值优化问题。区别于传统方法, 本文将所提SLWChOA用于优化伸张/压缩弹簧机械设计问题, 进一步验证改进算法的可行性和适用性。

表 10 CEC2014测试函数优化结果对比

算法	指标	CEC03	CEC5	CEC6	CEC16	CEC19	CEC23
ChOA	平均值	8.5262E+04	5.2104E+02	6.3955E+02	1.6132E+03	2.2677E+03	2.7530E+03
	标准差	4.5779E+03	5.4374E-02	2.2655E+00	2.2278E-01	1.1736E+02	6.1584E+01
PSO <sup>[14]</sup>	平均值	4.87E+01	2.00E+01	1.00E+01	1.13E+01	7.76E+00	3.16E+02
	标准差	6.61E+01	2.79E-02	2.25E+00	7.05E-01	1.87E+00	3.52E-01
LILGWO	平均值	4.7897E+04	5.2105E+02	<b>6.1994E+02</b>	1.6129E+03	1.9859E+03	<b>2.5000E+03</b>
	标准差	1.0698E+04	4.7332E-02	<b>2.7672E+00</b>	5.4023E-01	3.0825E+01	<b>0.0000E+00</b>
MSBOA	平均值	4.4620E+04	5.2173E+02	6.3435E+02	1.6126E+03	<b>1.9438E+03</b>	<b>2.5000E+03</b>
	标准差	8.0652E+03	9.5294E-02	2.9016E+00	3.8264E-01	<b>1.4244E+01</b>	<b>0.0000E+00</b>
L-SHADE <sup>[14]</sup>	平均值	0.00E+00	2.01E+01	1.67E-02	8.48E+00	3.121E+03	3.15E+02
	标准差	0.00E+00	1.70E-02	9.17E-02	2.97E-01	7.2E-01	1.14E-13
SLWChOA	平均值	7.9854E+04	<b>5.2103E+02</b>	6.3744E+02	<b>1.6120E+03</b>	2.1508E+03	<b>2.5000E+03</b>
	标准差	7.7506E+03	<b>5.5306E-02</b>	2.3690E+00	<b>3.0113E-01</b>	1.1479E+02	<b>0.0000E+00</b>

#### 4.1 伸张/压缩弹簧设计优化设计案例

伸张/压缩弹簧设计问题的优化目标是减少弹簧的重量,其结构示意图如图6所示。约束条件包括受到最小偏差( $g_1$ )、剪切应力( $g_2$ )、冲击频率( $g_3$ )、外径限制( $g_4$ )以及决策变量包括线径 $d$ 、平均线圈直径 $D$ 及有效线圈数 $P$ ,  $f(x)$ 为最小化弹簧重量。伸张/压缩弹簧设计的数学模型描述如下所示:

目标函数:

$$\text{设 } x = [x_1 x_2 x_3] = [d D N]$$

$$\min f(x) = x_1^2 x_2 (2 + x_3) \quad (22)$$

约束条件:

$$\begin{cases} g_1(x) = 1 - \frac{x_2^3 x_3}{71785 x_1^4} \leq 0 \\ g_2(x) = \frac{4x_2^2 - x_1 x_2}{12566(x_2 x_1^3 - x_1^4)} + \frac{1}{5108 x_1^2} - 1 \leq 0 \\ g_3(x) = 1 - \frac{140.45 x_1}{x_2^2 x_3} \leq 0 \\ g_4(x) = \frac{x_1 + x_2}{1.5} - 1 \leq 0 \end{cases} \quad (23)$$

其中:  $0.05 \leq x_1 \leq 2, 0.25 \leq x_2 \leq 1.3, 2 \leq x_3 \leq 15$

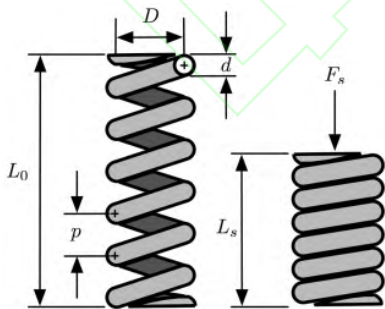


图 6 伸张/压缩弹簧结构示意图

#### 4.2 测试结果与分析

本文将所提SLWChOA与黑猩猩优化算法(ChOA)、粒子群算法PSO、正余弦算法(Sine Cosine Algorithm, SCA)<sup>[15]</sup>、哈里斯鹰优化算法(Harris Hawks Optimization, HHO)<sup>[16]</sup>、均衡器优化算法(equilibrium Optimization Algorithm, EO)<sup>[17]</sup>、斑鬣狗优化算法(Spotted Hyena Optimization, SHO)<sup>[18]</sup>、

灰狼优化算法(Grey Wolf Optimization, GWO)<sup>[19]</sup>、多元宇宙优化算法(Multi-Verse Optimization, MVO)<sup>[20]</sup>、麻雀搜索算法(Sparrow Search Algorithm, SSA)<sup>[21]</sup>进行实验比较,其中对比算法的数据来源于文献[12, 22]。实验选取种群规模为30,最大迭代次数为1000,每个算法独立运行30次并取平均值。表11分别是各算法伸张/压缩弹簧设计的优化结果。

表 11 伸张/压缩弹簧设计问题中各算法的最优解

算法	$d$	$D$	$N$	$f(x)$
SCA	0.0500	0.3168	14.1109	0.0132
HHO[12]	0.0562	0.4754	6.6670	0.0130
EO[12]	0.0512	0.3445	12.0455	0.0127
SHO[22]	0.0511	0.03437	12.0955	0.0127
GWO[22]	0.0500	0.3415	12.0734	0.0127
MVO[22]	0.0500	0.3159	14.2262	0.0128
PSO[22]	0.0500	0.3104	15.0000	0.0131
SSA	0.0551	0.4451	7.5125	0.0132
ChOA	0.0500	0.3146	14.4315	0.0134
SLWChOA	<b>0.0612</b>	<b>0.6260</b>	<b>4.1894</b>	<b>0.0127</b>

如表11所示,SLWChOA与其他算法处理约束函数值,SLWChOA获得的函数最优解为 $[x_1, x_2, x_3] = [0.0612, 0.06260, 4.1894]$ ,最优值 $f(x)=0.0127$ ,表明SLWChOA在减少弹簧的重量上获得最佳解决方案。进一步佐证SLWChOA在实际应用中的可行性和适用性。

## 5 结论

为了改善ChOA的寻优性能,首先,本文采用Sobol序列使初始化种群分布均匀,增加种群的多样性;其次,对种群在局部阶段聚集程度进行分析,引入基于凸透镜成像的反向学习策略生成动态的反向解,有效地提高算法的搜索精度和收敛速度;同时,在“攻击者”个体位置更新处加入水波动态自适应因子,引导种群向全局最优的狩猎搜索范围进行全方位搜索,保持种群寻优多样性的同



时有效避免了趋同性, 降低算法跳出局部最优的概率。通过对10个基准测试函数及其Wilcoxon检验及部分CEC2014函数等实验验证了改进算法具有更高的寻优性能和更强的鲁棒性。最后, 对机械设计问题进行优化测试分析, 验证了SLWChOA在实际工程问题中的适用性, 为解决复杂的工程优化问题提供一条新途径。

#### 参考文献(References)

- [1] Sooncharoen S, Pongcharoen P, Hicks C. Grey Wolf production scheduling for the capital goods industry[J]. *Applied Soft Computing*, 2020, 94: 106480.
- [2] Yan X, Liu Y, Jia M. A fault diagnosis approach for rolling bearing integrated SGMD, IMSDE and Multiclass relevance vector machine[J]. *Sensors*, 2020, 20(15): 4352.
- [3] Gai J, Shen J, Hu Y, et al. An integrated method based on hybrid grey wolf optimizer improved variational mode decomposition and deep neural network for fault diagnosis of rolling bearing[J]. *Measurement*, 2020, 162: 107901.
- [4] Khishe M, Mosavi M R. Chimp optimization algorithm[J]. *Expert systems with applications*, 2020, 149: 113338.
- [5] 宁杰琼,何庆.混合策略改进的蝴蝶优化算法[J].*计算机应用研究*,2021,38(6):1718-1723+1738.  
(Ning J Q, He Q. Mixed strategy to improve butterfly optimization algorithm[J]. *Application Research of Computers*, 2021, 38(6): 1718-1723+1738).
- [6] 龙文,伍铁斌,唐明珠,等.基于透镜成像学习策略的灰狼优化算法[J].*自动化学报*,2020, 46(10): 2148-2164.  
(Long W, Wu T B, Tang M Z, et al. Grey Wolf Optimizer Algorithm Based on Lens Imaging Learning Strategy[J]. *Acta Automatica Sinica*, 2020, 46(10): 2148-2164.)
- [7] Kaur M, Kaur R, Singh N, et al. SChOA: a newly fusion of sine and cosine with chimp optimization algorithm for HLS of datapaths in digital filters and engineering applications[J]. *Engineering with Computers*, 2021: 1-29.
- [8] Jia H, Sun K, Zhang W, et al. An enhanced chimp optimization algorithm for continuous optimization domains[J]. *Complex & Intelligent Systems*, 2021: 1-18.
- [9] Bratley P, Fox B L. Implementing sobols quasirandom sequence generator (algorithm 659)[J]. *ACM Transactions on Mathematical Software*, 2003, 29(1): 49-57.
- [10] Kennedy J, Eberhart R. Particle swarm optimization [C]//*Proceedings of ICNN'95-international conference on neural networks*. IEEE, 1995, 4: 1942-1948.
- [11] Nabil E. A modified flower pollination algorithm for global optimization[J]. *Expert Systems with Applications*, 2016, 57: 192-203.
- [12] Fatma A. Hashim; Kashif Hussain, Essam H. Houssein, et al. Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems[J]. *Applied Intelligence*, 2021, 51(3): 1531-1551.
- [13] Tanabe R, Fukunaga A S. Improving the search performance of SHADE using linear population size reduction[C]//2014 IEEE Congress on Evolutionary Computation (CEC). Beijing, China: IEEE, 2014:1658-1665.
- [14] 徐桂萍.基于维度学习策略的粒子群算法的研究与应用[D].吉林大学, 2019.  
(Xu Gui-ping. Study on particle swarm optimization based with dimensional learning strategy[D]. Jilin university: 2019.)
- [15] Mirjalili S. SCA: a sine cosine algorithm for solving optimization problems[J]. *Knowledge-Based Systems*, 2016, 96.
- [16] Heidari A A, Mirjalili S, Faris H, et al. Harris hawks optimization: Algorithm and applications[J]. *Future Generation Computer Systems*, 2019, 97: 849-872.
- [17] Faramarzi A, Heidarinejad M, Stephens B, et al. Equilibrium optimizer: A novel optimization algorithm[J]. *Knowledge-Based Systems*, 2020, 191: 105190.
- [18] Dhiman G, Kumar V. Spotted hyena optimizer for solving complex and non-linear constrained engineering problems[M]//*Harmony search and nature inspired optimization algorithms*. Springer, Singapore, 2019: 857-867.
- [19] Mirjalili S, Mirjalili S M, Lewis A. Grey wolf optimizer[J]. *Advances in engineering software*, 2014, 69: 46-61.
- [20] Mirjalili S, Mirjalili S M, Hatamlou A. Multi-verse optimizer: a nature-inspired algorithm for global optimization[J]. *Neural Computing and Applications*, 2016, 27(2): 495-513.
- [21] Xue J, Shen B. A novel swarm intelligence optimization approach: sparrow search algorithm[J]. *Systems Science & Control Engineering*, 2020, 8(1): 22-34.
- [22] Kaur S, Awasthi L K, Sangal A L, et al. Tunicate Swarm Algorithm: A new bio-inspired based metaheuristic paradigm for global optimization[J]. *Engineering Applications of Artificial Intelligence*, 2020, 90: 103541.

#### 作者简介

何庆(1982—), 男, 博士, 副教授, 从事大数据应用、进化计算等研究, E-mail: qhe@gzu.edu.cn;

罗仕杭(1998—), 男, 研究生, 从事群智能优化算法、神经网络等研究, E-mail: 1065678104@qq.com.