

基于非线性自适应比例因子的雪豹优化算法

崔铭悦^{1,2}, 莫愿斌^{1,2*}, 王子豪^{1,2}, 胡飓风^{1,2}

(1. 广西民族大学 人工智能学院, 广西 南宁 530006;
2. 广西混杂计算与集成电路设计分析重点实验室, 广西 南宁 530006)

摘要:针对雪豹优化算法在求解复杂优化问题时,存在全局勘探能力不足、寻优精度低等问题,提出一种改进的雪豹优化算法。首先,基于分段 Logistic 混沌映射初始化从而提高初始种群多样性;其次,引入非线性比例因子用于平衡算法的全局勘探能力和局部开发能力;然后,提出了一种差分变异策略,在第一次种群更新位置后,使用5个随机个体提高全局搜索能力和算法收敛能力,在第二次种群更新位置后,使用3个随机个体保证在求解过程的中后期也具有一定的全局勘探能力,尽可能避免陷入局部最优。通过在 IEEE CEC2022 基准函数测试集上测试,并与其他算法进行比较,结果表明所提出的算法在种群质量、求解精度以及算法稳定性上均有较大提升。最后将所提出的算法应用于工程优化,计算结果进一步证实了算法的强优化能力。

关键词:雪豹优化算法;混沌映射;非线性自适应比例因子;差分进化算子;约束优化问题

中图分类号: TP18

文献标识码: A

文章编号: 1673-629X(2024)04-0212-09

doi: 10.20165/j.cnki.ISSN1673-629X.2024.0032

Snow Leopard Optimization Algorithm for Nonlinear Adaptive Scaling Factor

CUI Ming-yue^{1,2}, MO Yuan-bin^{1,2*}, WANG Zi-hao^{1,2}, HU Ju-feng^{1,2}

(1. School of Artificial Intelligence, Guangxi Minzu University, Nanning 530006, China;
2. Guangxi Key Laboratory of Hybrid Computation & IC Design Analysis, Nanning 530006, China)

Abstract: An improved snow leopard optimization algorithm is proposed to solve complex optimization problems, such as insufficient global exploration ability and low optimization accuracy. Firstly, the initial population diversity is improved based on piecewise Logistic chaotic mapping initialization. Secondly, a nonlinear scaling factor is introduced to balance the global exploration ability and local development ability of the algorithm. Then, a differential variation strategy is proposed. After the first population update position, five random individuals are used to improve the global search ability and the convergence ability of the algorithm. After the second population update position, three random individuals are used to ensure the global exploration ability in the middle and late period of the solution process, so as to avoid falling into the local optimal as much as possible. The proposed algorithm is tested on IEEE CEC2022 benchmark function test set and compared with other algorithms. The results show that the proposed algorithm has great improvement in population quality, solving accuracy and algorithm stability. Finally, the proposed algorithm is applied to engineering optimization, and the calculation results further confirm its strong optimization ability.

Key words: snow leopard optimization algorithm; chaotic mapping; nonlinear adaptive scaling factor; differential evolution operator; constraint optimization problems

0 引言

在科学研究和工业实践中,各种复杂函数优化问题是众多学者研究的一个热点。考虑到这些问题的局限性和技术需求,优化目标是找到可接受的最佳解^[1]。

群智能算法凭借其原理简单、寻优速度快、精度高等优点,已广泛被应用于该领域当中。粒子群算法^[2](Particle Swarm Optimization, PSO)是最为经典的群智能优化算法,在此之后研究学者受自然界生物群体的

收稿日期: 2023-05-28

修回日期: 2023-09-29

基金项目: 国家自然科学基金资助项目(2146008); 广西自然科学基金资助项目(2019GXNSFAA185017); 广西民族大学科研资助项目(2021MDKJ004)

作者简介: 崔铭悦(1997-),女,CCF会员(P1842G),硕士研究生,研究方向为智能信息控制;通信作者:莫愿斌(1969-),男,教授,硕士,研究方向为智能信息控制。

社会行为规律启发,提出了更多的群体智能优化算法^[3~5]来解决优化问题。例如,遗传算法^[6](Genetic Algorithm, GA)、灰狼算法^[7](Gray Wolf Optimization, GWO)、鲸鱼优化算法^[8](Whale Optimization Algorithm, WOA)、樽海鞘群算法^[9](Slap Swarm Algorithm, SSA)等。然而,面对各种复杂问题时,传统的优化算法存在操作不灵活、求解精度不够高等不足。因此,研究学者对群智能优化算法进行了改进,如莱维飞行搜索、混合黄金正弦、自适应参数调整等。武泽权等人^[10]提出一种改进的鲸鱼优化算法,运用反向学习初始种群,结合自适应权重以及差分变异策略优化鲸鱼算法,以提升算法的优化性能;Jia Yinghui 等人^[11]将粒子群算法和乌鸦搜索算法相结合,为个体在另一个随机个体的引领下探索未知区域提供了可能;刘云平等^[12]提出一种控制参数非线性自适应调整策略改进灰狼算法,用来平衡算法局部和全局搜索能力;谢良波等人^[13]提出一种自适应蝙蝠算法,利用混沌反向映射初始化种群,引入自适应因子更新种群位置;董红斌等人^[14]提出一种动态调整权重的粒子群算法,采用不同适应值的指数函数改变惯性权重,使算法可以跳出局部最优。

雪豹优化算法(Snow Leopard Optimization Algorithm, SLOA)是由 Petr Coufal 等^[15]于2021年提出的新的启发式优化算法,该算法受自然界雪豹群体独有的习性和自然行为演化成随机优化过程。SLOA最主要的优点是不依赖于群体中的最佳成员,从而提高算法的搜索能力,但也依旧存在收敛速度慢、求解精度低以及易陷入局部最优的缺点。因此,为了更好地解决上述不足,该文设计了一个新的算法(Improved Snow Leopard Optimization Algorithm, ISLOA),把分段 Logistic 混沌映射初始化与自适应比例因子改进相结合,同时将差分进化算子引入 SLOA,增强算法全局勘探与局部开发的平衡能力。将 ISLOA 算法在部分 CEC2022 测试函数集上进行仿真实验,结果表明算法在收敛速度和求解精度上都获得了显著的提升;并将 ISLOA 算法应用于压缩弹簧设计问题和压力容器设计问题^[16],证明了 ISLOA 的有效性和工程适用性。

1 雪豹优化算法

SLOA 是模拟自然界雪豹群体的捕猎行为和社会生活,重点关注旅行路线、狩猎、繁殖和死亡四个阶段衍生出的一种群体智能优化算法。在 SLOA 中,搜索空间中的每只雪豹都可以为问题的目标函数计算一个值。假设搜索个体空间维度为 d ,雪豹种群规模为 N ,用 $X_i = (x_{i,1}, x_{i,2}, \dots, x_{i,d})$, $i = 1, 2, \dots, N$ 表示第 i 只雪豹在第 d 维空间中的位置。用式 1~3 进行数学

建模:

$$x_{i,d}^{p1} = x_{i,d} + r \times (x_{k,d} - l \times x_{i,d}) \times \text{sign}(F_i - F_k) \\ k \in 1, 2, \dots, N, \quad d = 1, 2, \dots, m \quad (1)$$

$$X_1 = \begin{cases} X_i^{p1}, & F_i^{p1} < F_i \\ X_i, & \text{else} \end{cases} \quad (2)$$

$$l = \text{round}(1 + r) \quad (3)$$

其中, $x_{i,d}^{p1}$ 是第 i 只雪豹更新后的新值, r 是 $[0, 1]$ 区间上的随机数, X_i^{p1} 是更新的雪豹位置, F_i^{p1} 是目标函数值, X_1 是第一阶段更新后雪豹的位置。

根据观察,雪豹发现猎物后会先行走一段路程,然后在行走距离猎物约 37.5% 处开始跑向猎物,最后完成猎杀。式 4 使用一个名为 P 的参数来模拟雪豹如何向猎物移动:

$$x_{i,d}^{p2} = x_{i,d} + r \times ((x_{j,d} - x_{i,d}) \times P + (x_{j,d} - 2 \times x_{i,d}) \times (1 - P)) \times \text{sign}(F_1 - F_p) \quad (4)$$

其中, $x_{i,d}^{p2}$ 为雪豹位置, $x_{j,d}$ 为当前猎物位置, r_1 为 $[0, 1]$ 之间的随机数, F_p 是猎物位置的目标函数值, P 为常数 0.375。

式 5 模拟了雪豹攻击猎物后的新位置:

$$X_2 = \begin{cases} X_i^{p2}, & F_i^{p2} < F \\ X_i, & \text{else} \end{cases} \quad (5)$$

其中, X_2 是第二阶段更新后雪豹的位置, F_i^{p2} 是目标函数值,如果目标函数在新位置的值比之前的位置更好,那么更新位置,反之则不变。

基于雪豹自然繁殖行为,成员中两只雪豹交配,假设幼崽会出生,根据式 6 进行数学建模:

$$C_l = \frac{X_l + X_{N-l+1}}{2}, \quad l = 1, 2, \dots, \frac{N}{2} \quad (6)$$

其中, C_l 是两只雪豹交配而生的第 l 只幼崽。

SLOA 的四个阶段:第一阶段,模拟雪豹锯齿形运动有效精确地扫描搜索空间并获得局部最优解;第二阶段,雪豹捕猎过程以两种不同的前进速度接近猎物;第三阶段,雪豹中两个成员组合产生一个新成员作为随机解;第四阶段,消除雪豹中的弱势群体成员。

2 改进的雪豹优化算法

2.1 基于分段 Logistic 混沌的种群初始化

混沌映射^[17]具有非线性、随机性、遍历性等特点,可以在一定范围内按自身的性质不重复地遍历所有状态。在智能群体算法优化中利用混沌运动的这些特性可以生成初始种群并增加其多样性。在 SLOA 的种群个体位置更新时,由于一定程度上的随机性致使种群位置不好,缺乏多样性,易陷入局部最优。分段 Logistic 混沌映射^[18-19]产生序列具有较好的非线性、初值敏感性、增加种群多样性等性质,虽然其结构简

单,但却能产生分布较为均匀且复杂的混沌动力学特性的优点,并且在生成混沌序列时无需额外进行扰动运算,确保算法搜索速率快以及具有更好的安全性。因此,该文借助分段 Logistic 混沌映射优化 SLOA,即搜寻种群最优解的问题。具体表达式如下:

$$x_{n+1} = \begin{cases} 4 \times \mu \times x_n \times (0.5 \times x_n), & 0 \leq x_n < 0.5 \\ 1 - 4 \times \mu \times (x_n - 0.5) \times (1 - x_n), & 0.5 \leq x_n \leq 1 \end{cases} \quad (7)$$

其中, x_0 的取值范围: $0 < x_0 < 1$; 参数 μ 的取值范围: $3.569\,946\cdots \leq \mu \leq 4$, 当 $\mu = 4$ 时,几乎所有的初始值都会出现混沌特性。

分段 Logistic 映射的种群初始化步骤如下(雪豹种群个体数量 N , 求解维度 d , 最大迭代次数 T):

(1) 随机生成一个 d 维向量 $X_i = (x_{i,1}, x_{i,2}, \cdots, x_{i,d})$, 其每个分量在 $(0,1)$ 之间, 根据式 7 产生 N 个向量 $y_i = f(x_i(j))$, $i = 1, 2, \cdots, N$; $j = 1, 2, \cdots, d$;

(2) 根据 $\vec{y}_i = \vec{lb}(j) + \vec{x}_i(j) \times (\vec{ub}(j) - \vec{lb}(j))$, 将 x_i 的第 j 个分量映射到区间 $[\vec{lb}(j), \vec{ub}(j)]$ 内;

(3) 将这 N 个向量作为种群的初始位置。

2.2 非线性自适应比例因子

在 SLOA 中,模拟雪豹在狩猎过程中的自然行为使用方程式 1~3 进行数学建模,已在第 1 节中详细阐述。式 4 中使用了一个名为 P 的固定参数来模拟雪豹如何向猎物移动的行为。但是考虑到当雪豹过于接近猎物时,行走与奔跑距离比例应发生相应的改变,以防止猎物发现并逃跑。为此,该文在 SLOA 的基础上对参数 P 进行改进,提出了一种新的非线性自适应比例因子。其中 P 的值随着迭代的进行不断衰减,在迭代初期,一个较大的 P 可以引导算法在搜索空间找到更多的候选解,从而提升算法的全局搜索能力;迭代中期,比例因子 P 有一个较快的衰减,使算法可以迅速收敛;迭代后期,其衰减程度逐渐减缓,从而提升算法的局部开发能力。

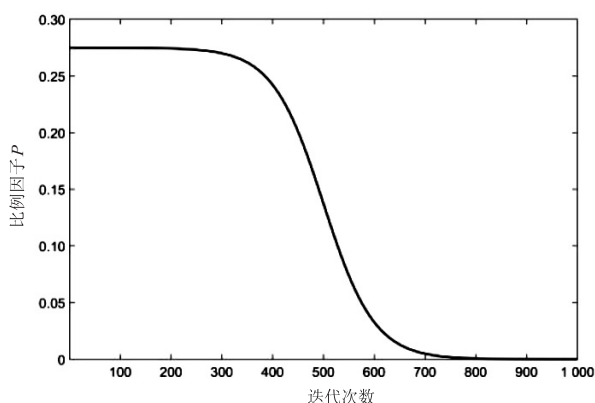


图 1 Sigmoid 函数

Sigmoid 函数^[20]在线性和非线性之间能够取得较好的平衡,因此该文选取 $[-L_{\max}, L_{\max}]$ 之间的 Sigmoid 函数(如图 1 所示)对 P 进行自适应调整,公式如下:

$$P_t = P_{\min} + \frac{P_{\max} - P_{\min}}{1 + e^{-\frac{t - T_{\max}}{2L_{\max} - T_{\max}}}} \quad (8)$$

其中, P_{\max} 和 P_{\min} 为比例因子 P 的最大值和最小值,取值分别为 0.275 和 0, T_{\max} 和 t 分别为最大迭代次数和当前迭代次数。

2.3 差分变异策略

SLOA 作为一种新型群体智能优化算法,虽然具有一定的全局勘探能力,但随着迭代次数的增加种群多样性逐渐减少,致使算法无法跳出局部最优,收敛精度不高。因此,该文提出了一种新的差分变异策略。

在第一阶段使用 5 个随机个体,提高算法的全局搜索能力和收敛能力。公式如下:

$$V_{i,T} = X_{r_1,T} + \text{rand}(X_{r_2,T} - X_{r_3,T}) + \text{rand}(X_{r_4,T} - X_{r_5,T}) \quad (9)$$

其中, $X_{r_1,T}, X_{r_2,T}, X_{r_3,T}, X_{r_4,T}, X_{r_5,T}$ 为 T 代种群中 5 个不同的随机个体,变异因子 rand 是 $[0,1]$ 之间的随机数, $V_{i,T}$ 为当前产生的变异个体。

将候选个体 X_1 与变异个体 $V_{i,T}$ 进行适应度值的比较,并且选择适应度较好的个体作为目标个体 $X_{i,T+1}$,其具体表达式如下所示:

$$X_{i,T+1} = \begin{cases} X_1, & \text{fit}(X_1) < \text{fit}(V_{i,T}) \\ V_{i,T}, & \text{else} \end{cases} \quad (10)$$

其中, fit 是目标函数的适应度值。

在第二阶段使用 3 个不同的随机个体,保证在求解过程的中后期算法也具有一定的全局勘探能力,促使个体离开当前搜索区域,从而避免陷入局部最优,公式如下:

$$V_{i,T} = X_{r_1,T} + \text{rand}(X_{r_2,T} - X_{r_3,T}) \quad (11)$$

将候选个体 X_2 与变异个体 $V_{i,T}$ 进行适应度值比较,选择适应度较好的个体作为目标个体 $X_{i,T+1}$,具体表达式如下:

$$X_{i,T+1} = \begin{cases} X_2, & \text{fit}(X_2) < \text{fit}(V_{i,T}) \\ V_{i,T}, & \text{else} \end{cases} \quad (12)$$

2.4 ISLOA 实现流程

算法 1: ISLOA。

输入: 雪豹种群数量 N , 求解维度 d 。

输出: 通过 SLOA 获得的最优解。

a) Evaluate the objective function

For $i = 1 : N$

For $d = 1 : m$

According Equation (1) and Equation (3) to calculate.

End

According Equation (2) and Equation(9) to update

End// 第一次更新种群位置
 b) For $i = 1; N$
 For $d = 1; m$
 According Equation (4) and Equation (8) to calculate .
 End
 According Equation (5) and Equation (11) to update
 End// 第二次更新种群位置
 c) For $i = 1; 0.5N$
 Generate cub according Equation (6).
 End// 更新种群,数量增加到 $1.5N$
 d) Update the population to N // 更新种群,数量减少到 N
 Save the best quasi-optimal solution obtained by SLOA so far.
 return the best solution and best value.

3 算法复杂度分析

SLOA 主要由四个部分(初始化种群、位置更新、种群繁殖和死亡)组成。假设雪豹种群数量为 N ,最大迭代次数为 T ,求解维度为 d ,则 SLOA 初始化阶段时间复杂度为 $O(N \times d)$,求其使用度需花费 $O(N \times O_{\text{obj}})$;种群进行两次位置更新的时间复杂度均为 $O(N \times O_{\text{obj}} \times d)$;繁殖阶段为 $O(\frac{N}{2} \times d)$,死亡阶段为 $O(N \log N)$ 。综上省略低次位求得 SLOA 总的时间复杂度为 $O((O_{\text{obj}} \times d + \log N)N \times T)$ 。

同理,提出的 ISLOA 在 SLOA 的基础上增加了两次差分进化突变的时间复杂度为 $O(N \times d)$,因此 ISLOA 总的时间复杂度为 $O((O_{\text{obj}} \times d + \log N)N \times T)$,并没有在 SLOA 的基础上增加其时间复杂度。

4 实验仿真与结果分析

在实验过程中选用 MATLAB 编程语言,实验环境均为 Intel(R) Core(TM) i9-10980HK CPU @ 2.40 GHz,所有算法均使用 MATLAB R2021a 进行测试。

4.1 数据集

为了验证所提出 ISLOA 处理优化问题的寻优能力,采用了 IEEE CEC2022 基准函数(其源代码链接:<https://github.com/P-N-Suganthan/2022-SO-BO>),测试函数信息如表 1 所示。

表 1 基准测试函数

Type	Number	Name	Optimal solution
Unimodal	F1	Zakharov Function	300
	F2	Rosenbrock's Function	400
Basic	F3	Schaffer's F7 Function	600
	F4	Rastrigin's Function	800
	F5	Levy Function	900

续表 1

Type	Number	Name	Optimal solution
Mixed	F6	Hybrid Function 1	1 800
	F7	Hybrid Function 2	2 000
	F8	Hybrid Function 3	2 200
Compound	F9	Composition Function 1	2 300
	F10	Composition Function 2	2 400
	F11	Composition Function 3	2 600
	F12	Composition Function 4	2 700
Define domain: $[-100, 100]$			

4.2 评价指标和共同参数设置

在所有实验中,6 种算法分别独立运行 30 次,每次种群个数为 80,最大迭代次数为 1 000,取 30 次实验中每个基准函数最佳适应度值的平均值(Mean)和标准差(Std)作为性能评价标准。将 ISLOA 与 PSO、GA、WOA、GWO、SLOA 进行对比测试,比较算法的详细信息如表 2 所示。

表 2 参数设置

算法名称	参数指标
PSO	$\omega = 0.1, c_1 = 2, c_2 = 2$
GA	Crossover = 0.8, Mutation = 0.02
WOA	$au = [2, 0], b = 1$
GWO	$a = [2, 0]$
SLOA	$P = 0.375$
ISLOA	$P_{\max} = 0.275, P_{\min} = 0$

4.3 实验结果与分析

实验数据结果如表 3 所示。从中可以看出,算法在求解函数 F1, F2, F3, F5, F6 时均能够收敛到全局最优解。对于函数 F4, F7 ~ 12 在 ISLOA 中虽然没能收敛到理论的最优值,但其获得的值最接近理论最优解。其中,除了函数 F3, F8, F11 以外,所提出的算法平均最优适应度值和标准差均排在第一,但在这三个函数上 ISLOA 也能求出最接近理论最优值。综上可知,所提算法在一定程度上具有较强的求解优化问题的能力。

为了直观体现出 ISLOA 与其他算法在收敛速度与求解精度的不同,在绘图时将每个最佳适应度值与所求解的基准函数全局最优值做差以进行比较,见图 2。从图 2 中可以看出,对于函数 F1, F2, F4 ~ 7, F11 的收敛速度和求解精度都明显优于其他对比算法,证实了 ISLOA 改进的有效性。值得一提的是函数 F5,只需迭代 30 次左右就可以收敛到目标函数的理论最优值,而其他算法虽然收敛速度也不差,但是会陷入局部最优解则无法跳出,从而影响算法求解精度。为了更直

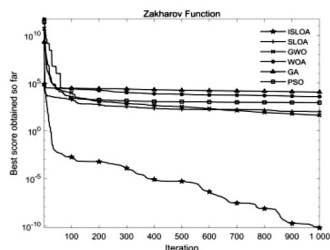
观看出算法的求解精度,对函数 F3, F7 ~ 10, F12 的迭代曲线进行局部放大,由图 2 可以看出函数 F3 最终在局部收敛的函数值上没有 GA 算法的值更精,但是除了函数 F3 以外所提算法在求解精度上均比其他算法

更高。

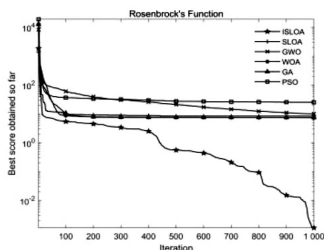
进一步说明了改进之后算法在单维、多维、高维等函数的测试中均体现了自身的优势,不仅加快了收敛速度,也提高了求解精度。

表 3 6 种优化算法寻优结果比较

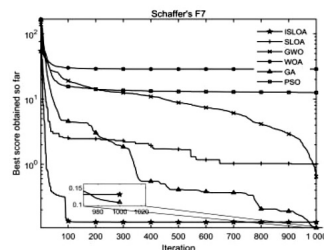
		ISLOA	SLOA	GWO	WOA	GA	PSO
F1	Mean	3.00E+02	3.90E+02	3.80E+02	5.19E+03	1.36E+04	1.19E+03
	Std	1.30E-06	4.07E+01	2.22E+02	6.33E+03	7.42E+03	5.17E+01
F2	Mean	4.00E+02	4.11E+02	4.15E+02	4.19E+02	4.17E+02	4.34E+02
	Std	7.28E-01	1.35E+01	1.90E+01	2.62E+01	2.81E+01	1.86E+01
F3	Mean	6.00E+02	6.01E+02	6.02E+02	6.34E+02	6.00E+02	6.14E+02
	Std	3.59E-02	5.02E-01	3.23E+00	1.45E+01	1.45E-02	2.98E+00
F4	Mean	8.05E+02	8.23E+02	8.14E+02	8.40E+02	8.23E+02	8.44E+02
	Std	2.34E+00	1.74E+01	6.26E+00	1.84E+01	1.09E+01	2.61E+00
F5	Mean	9.00E+02	9.03E+02	9.06E+02	1.57E+03	1.03E+03	9.74E+02
	Std	4.16E-01	3.06E+00	1.07E+01	3.50E+02	1.29E+02	6.61E+00
F6	Mean	1.80E+03	2.58E+03	5.03E+03	3.88E+03	3.02E+03	4.31E+06
	Std	7.04E+00	1.05E+03	2.22E+03	2.13E+03	1.27E+03	7.97E+05
F7	Mean	2.01E+03	2.02E+03	2.03E+03	2.08E+03	2.02E+03	2.06E+03
	Std	9.22E+00	7.88E+00	9.39E+00	3.92E+01	9.24E+00	4.51E+00
F8	Mean	2.21E+03	2.22E+03	2.22E+03	2.24E+03	2.22E+03	2.24E+03
	Std	9.53E+00	5.98E+00	7.67E+00	1.18E+01	3.54E+00	2.21E+01
F9	Mean	2.49E+03	2.49E+03	2.53E+03	2.54E+03	2.53E+03	2.54E+03
	Std	1.93E-05	3.16E+00	7.83E+00	3.63E+01	5.25E+00	6.86E-01
F10	Mean	2.50E+03	2.53E+03	2.54E+03	2.57E+03	2.53E+03	2.57E+03
	Std	1.93E+01	4.92E+01	5.55E+01	7.02E+01	5.08E+01	7.60E+01
F11	Mean	2.65E+03	2.72E+03	2.72E+03	2.77E+03	2.68E+03	2.84E+03
	Std	1.02E+02	8.51E+01	1.75E+02	2.05E+02	1.12E+02	1.66E+02
F12	Mean	2.86E+03	2.90E+03	2.86E+03	2.92E+03	2.87E+03	2.87E+03
	Std	9.97E-01	8.96E+00	2.05E+00	4.51E+01	3.41E+00	1.28E+00



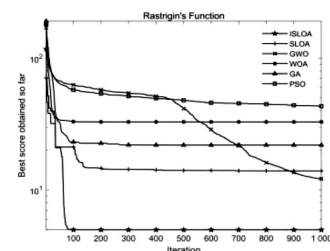
(a)迭代曲线



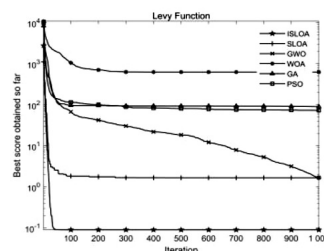
(b)迭代曲线



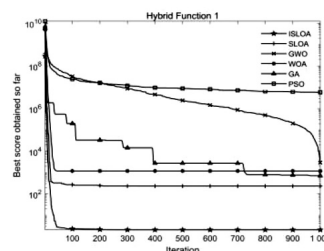
(c)迭代曲线



(d)迭代曲线



(e)迭代曲线



(f)迭代曲线

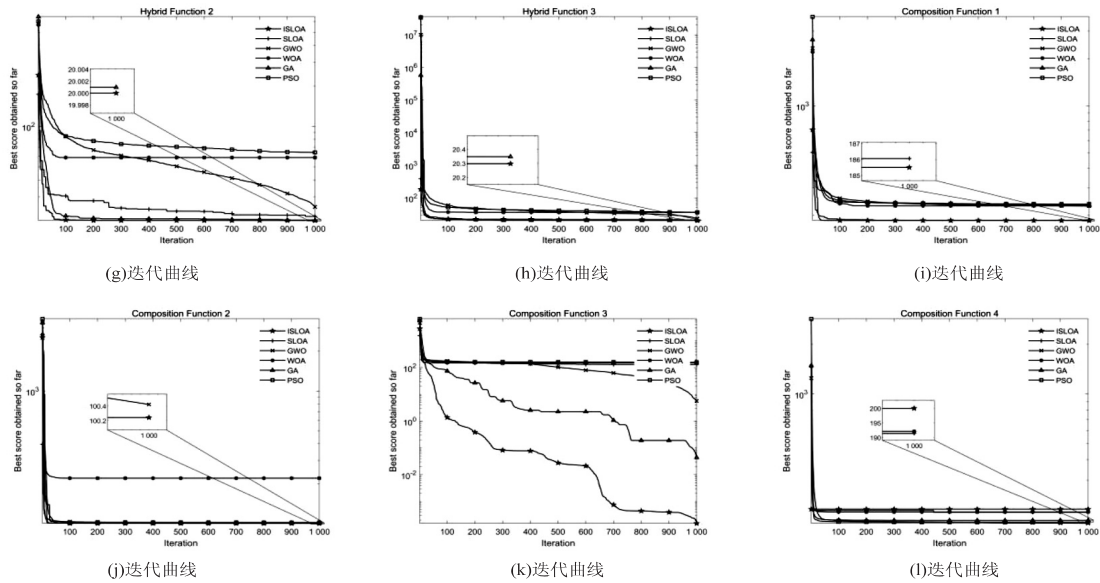


图2 基准函数平均收敛曲线

图3以箱线图的形式展示了提出的ISLOA和五种优化算法的性能比较。该图直观地展示了提出的ISLOA在优化函数F1到F9的功能方面的优越性。

为了更加直观地看出算法的求解能力,根据每个

算法的优化结果进行排序(优先考虑算法的平均最优适应度值,如果出现一样的值则根据其标准差进行排序)绘制出图4的雷达图。算法的排序越低,越靠近中心,其性能越好。

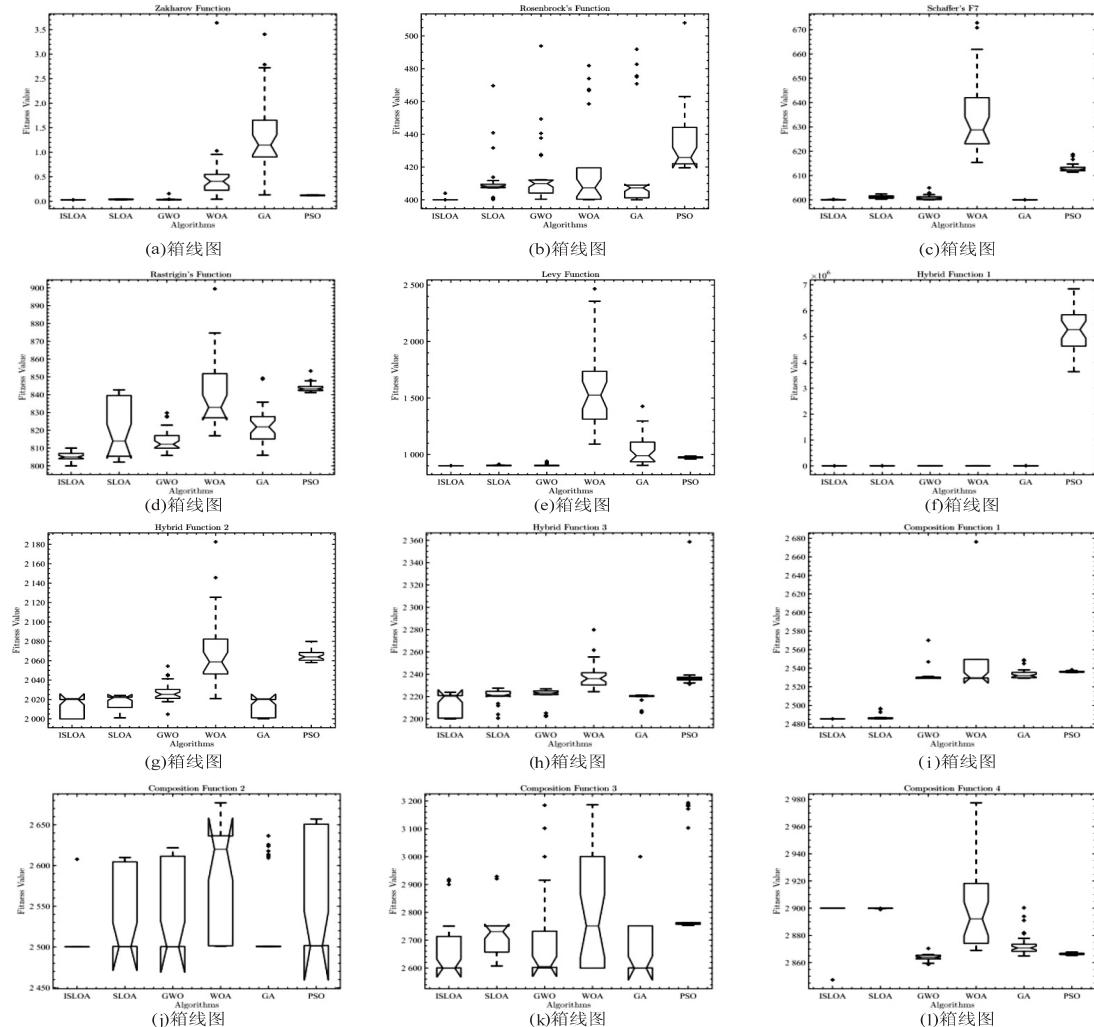


图3 基准函数箱线图

从图 4 中可以看出,在 12 个测试函数排名中 ISLOA 均超过了 SLOA,说明在基本 SLOA 上进行的改进可以有效提升算法的求解精度。除了函数 F3 上,所提出的算法被 GA 超越,其余在 11 个测试函数中均排在第一,进一步说明 ISLOA 具有收敛精度高、鲁棒性好等特点,可以作为一种新的处理约束优化问题的工具。

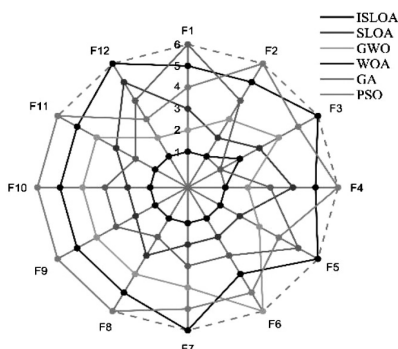


图 4 雷达图

表 4 Wilcoxon 统计检验结果

	SLOA	GWO	WOA	GA	PSO
F1	6.02E-07+	6.02E-07+	6.02E-07+	6.02E-07+	6.02E-07+
F2	1.08E-06+	6.02E-07+	6.02E-07+	1.75E-06+	6.02E-07+
F3	6.02E-07+	6.02E-07+	6.02E-07+	1.98E-04+	6.02E-07+
F4	2.34E-05+	9.82E-07+	6.02E-07+	6.02E-07+	6.02E-07+
F5	6.64E-07+	1.81E-05+	6.02E-07+	6.02E-07+	6.02E-07+
F6	6.02E-07+	6.02E-07+	6.02E-07+	6.02E-07+	6.02E-07+
F7	2.15E-05+	1.31E-06+	6.02E-07+	2.53E-02+	6.02E-07+
F8	9.32E-04+	4.91E-05+	6.02E-07+	7.99E-03+	8.08E-07+
F9	6.02E-07+	6.02E-07+	6.01E-07+	6.02E-07+	6.02E-07+
F10	9.90E-06+	1.19E-06+	2.54E-06+	6.02E-07+	3.06E-06+
F11	3.90E-03+	4.28E-02+	2.02E-03+	1.65E-02+	8.16E-04+
F12	8.08E-07+	3.64E-01-	6.02E-07+	6.02E-07+	8.30E-06+

5 工程应用

为了证明 ISLOA 在工程应用领域中的实用性和有效性,选用压缩弹簧设计问题和压力容器设计问题两个经典的工程应用优化案例进行实验分析,实验重复独立运行 30 次,每次种群个数为 80,最大迭代次数为 1 000,记录其平均值,并与其他算法进行对比。

5.1 压缩弹簧设计问题

该问题的目的是使压缩弹簧的重量最小化,模型如图 5 所示。

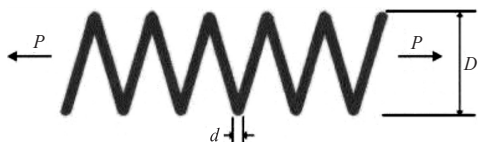


图 5 压缩弹簧设计问题

它有三个设计变量:导线直径 $d = x_1 \in [0.05,$

4.4 统计检验

由最佳解决方案的平均值和标准差作为统计指标展示出目标函数优化的结果,可以提供有用且有价值的信息。然而,一个算法在解决优化问题时性能表现好其实有可能是存在巧合的,即使在进行了 30 次独立实验之后也是如此。因此,该文使用了 Wilcoxon 秩和检验^[21]。为了对比分析所提出的 ISLOA 和其他五种优化算法的优化结果,分析判断 ISLOA 获得的结果是否与其他算法在统计上具有显著差异。检验结果如表 4 所示。其中, p 值表示给定的优化算法是否具有统计意义。如果 $p < 0.05$,则相应的优化算法具有统计显著性。表中符号“+”表示 ISLOA 相对其他算法具有显著性差异,符号“-”表示 ISLOA 相对其他算法无显著性差异。基于对表 4 中测试结果的分析可以得出结论,除了 ISLOA 与 GWO 求解函数 F12 时无明显差异以外,ISLOA 在其余所有函数测试中均有显著优势。

2],线圈或绕组的直径 $D = x_2 \in [0.25, 1.3]$,有效线圈的数量 $N = x_3 \in [2, 15]$ 。假设 $k = [k_1, k_2, k_3] = [d, D, N]$,其目标函数模型可见公式 13:

$$\text{Minf}(K) = (k_3 + 2)k_2k_1^2 \quad (13)$$

其中约束条件为:

$$g_1(K) = 1 - \frac{k_2^3k_1}{71\,785k_1^4} \leq 0$$

$$g_2(K) = \frac{4k_2^2 - k_1k_2}{12\,566(k_1^3k_2 - k_1^4)} + \frac{1}{5\,108k_1^2} - 1 \leq 0$$

$$g_3(K) = 1 - \frac{140.45k_1}{k_2^2k_3} \leq 0$$

$$g_4(K) = \frac{k_1 + k_2}{1.5} - 1 \leq 0$$

分别利用该文提出的算法和 SLOA 求解压缩弹簧优化问题,并与算法 PSO^[22]、WOA^[9]、HHO^[23]、

RO^[24]、MFO^[25]的最优结果进行比较。为保证数据的准确性,相关算法数据均来自原文献。对比结果如表5所示。

表5 压缩弹簧设计的比较结果

	x_1	x_2	x_3	best
ISLOA	0.051 69	0.356 72	11.288 97	0.012 665
SLOA	0.050 86	0.337 07	12.541 30	0.012 678
HHO	0.051 97	0.363 66	10.892 75	0.012 666
WOA	0.051 20	0.345 21	12.004 03	0.012 676
RO	0.051 37	0.349 09	11.762 79	0.012 678
PSO	0.051 72	0.357 64	11.244 54	0.012 674
MFO	0.051 99	0.314 60	10.868 42	0.012 666

表5将ISLOA与其他优化算法的最佳结果进行了比较。由表可以看出所提出的算法可以精确求得其压缩弹簧的最小质量,证明了ISLOA在压缩弹簧设计问题上的优越性。

5.2 压力容器设计问题

该问题的主要目的是在非线性约束条件下,使得压力容器设计的成本最小化,模型如图6所示。

它有四个设计变量:筒体厚度为 $T_h = x_1 \in [0, 99]$,半球体厚度为 $T_s = x_2 \in [0, 99]$,内部半径为 $R =$

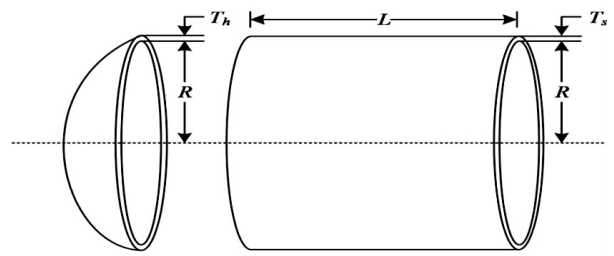


图6 压力容器设计问题

$x_3 \in [10, 200]$,筒体长度为 $L = x_4 \in [10, 200]$ 。假设 $k = [k_1, k_2, k_3, k_4] = [T_h, T_s, R, L]$,其目标函数模型可见公式14:

$$\min f(x) = 0.622 4 k_1 k_3 k_4 + 1.778 1 k_2 k_3^2 + 3.166 1 k_1^2 k_4 + 19.84 k_1^2 k_3 \quad (14)$$

其中约束条件为:

$$g_1(K) = -k_1 + 0.019 3 k_3 \leq 0$$

$$g_2(K) = -k_3 + 0.009 54 k_3 \leq 0$$

$$g_3(K) = -\pi k_3^2 k_4 - \frac{4}{3} \pi k_3^3 + 1 296 000 \leq 0$$

$$g_4(K) = k_4 - 240 \leq 0$$

该文利用ISLOA和SLOA求解PVD问题,并与算法ACO^[26]、GA^[27]、WOA^[9]、GSA^[28]、MVO^[29]的求解结果进行比较。对比结果如表6所示。

表6 压力容器设计的比较结果

	x_1	x_2	x_3	x_4	best
ISLOA	0.778 2	0.384 6	40.319 60	200.000 0	5 885.332 7
SLOA	0.838 0	0.424 5	43.390 70	161.419 0	6 037.274 0
ACO	0.812 5	0.437 5	42.098 35	176.637 7	6 059.725 8
GA	0.812 5	0.437 5	42.097 39	176.654 0	6 059.946 3
WOA	0.812 5	0.437 5	42.098 26	176.638 9	6 059.741 0
GSA	1.125	0.625	55.988 65	84.454 20	8 538.835 9
MVO	0.812 5	0.437 5	42.090 73	176.738 6	6 060.806 6

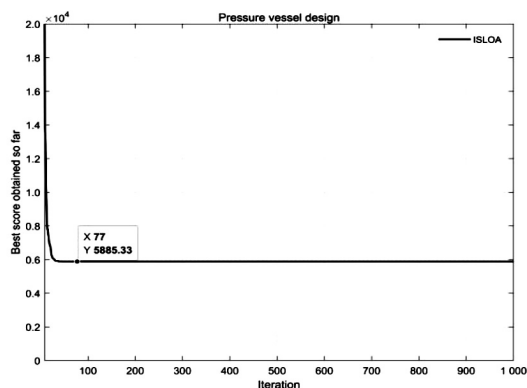


图7 压力容器收敛图

表5将ISLOA与其他优化算法的最佳结果进行了比较分析。在PVD问题寻优结果中,所提出的算法结果最优,其中SLOA相比近年来提出的其他算法也

有所提升。此外,通过图7可以看出ISLOA的收敛曲线在迭代第77次时就可以收敛到理论最优值,说明ISLOA具有较好的收敛性能。同时,对比SLOA,ISLOA在PVD问题总造价节约方面最优解提升了2.38%,且优势较为明显。

通过测试以上两个著名的工程设计问题,可以看出ISLOA在求解工程优化设计问题时可以找到其最优解,充分验证了提出的算法在处理不同复杂程度工程设计问题具有良好的应用前景,可以提供最佳解决方案。

6 结束语

为了改善SLOA后期全局探索能力不足、求解精度低以及易陷入局部最优等问题,该文采用分段

Logistic 混沌初始化,提高初始种群多样性;引入非线性比例因子更新种群位置,提高算法的全局勘探能力和局部开发能力;提出一种差分进化算子,在两次种群更新位置后使用不同的更新策略,提高算法收敛能力以及防止算法陷入局部最优。仿真实验表明,ISLOA 与其他多种优化算法比较在种群质量、寻优精度以及算法鲁棒性上都有较大提升。通过 Wilcoxon 秩和检验,可以得出提出的算法在寻优性能上具有更优的显著性差异。最后选择应用两种工程设计问题的实验测试表明,ISLOA 可用于解决实际工程问题并具有较好的工程应用潜力。但是,和 SLOA 一样,该文提出的算法也具有一定的自身局限性,在后续的研究中将在时间复杂度上做进一步改进,并将所提出的算法应用到更广泛的实际问题当中,提高算法的应用价值。

参考文献:

- [1] 郑慧杰,刘弘,郑向伟. 基于模拟退火的差分变异群搜索优化算法[J]. 计算机工程,2012,38(17):178-181.
- [2] 郝琨,邓晁硕,赵璐. 基于区域搜索粒子群算法的机器人路径规划[J]. 电子测量与仪器学报,2022,36(12):126-135.
- [3] WANG Z,MO Y,CUI M,et al. An improved golden jackal optimization for multilevel thresholding image segmentation[J]. PLOS ONE,2023,18(5):e0285211.
- [4] 莫愿斌,刘付永,张宇楠. 带高斯变异的人工萤火虫优化算法[J]. 计算机应用研究,2013,30(1):121-123.
- [5] 汪其. 改进型人工鱼群算法的物流配送应用研究[J]. 电脑知识与技术,2022,18(33):13-16.
- [6] KALYANMOY D,SAMIR A,AMRIT P,et al. A fast and elitist multiobjective genetic algorithm: NSGA-II[J]. IEEE Trans. Evolutionary Computation,2002,6(2):182-197.
- [7] 陈敏,陈晔,牛兴龙. 求解全局优化问题的多策略改进灰狼算法[J]. 国外电子测量技术,2022,41(11):22-29.
- [8] SEYEDALI M,ANDREW L. The whale optimization algorithm[J]. Advances in Engineering Software,2016,95:51-67.
- [9] 黄小根,钟尚勤. 一种多策略驱动的改进樽海鞘群算法[J]. 计算机仿真,2022,39(1):308-311.
- [10] 武泽权,牟永敏. 一种改进的鲸鱼优化算法[J]. 计算机应用研究,2020,37(12):3618-3621.
- [11] JIA Y H,QIU J,MA Z Z,et al. A novel crow swarm optimization algorithm (CSO) coupling particle swarm optimization (PSO) and crow search algorithm (CSA)[J]. Computational Intelligence and Neuroscience,2021,2021:1-14.
- [12] 刘云平,朱慧如,方卫华. 改进灰狼算法的无人机路径规划[J]. 电光与控制,2023,30(7):1-7.
- [13] 谢良波,李宇洋,王勇. 基于自适应蝙蝠算法的室内 RFID 定位算法[J]. 通信学报,2022,43(8):90-99.
- [14] 董红斌,李冬锦,张小平. 一种动态调整惯性权重的粒子群优化算法[J]. 计算机科学,2018,45(2):98-102.
- [15] COUFAL P,ŠTĚPÁN H,MARIE H,et al. Snow leopard optimization algorithm: a new nature-based optimization algorithm for solving optimization problems[J]. Mathematics,2021,9(21):2832.
- [16] 刘睿,莫愿斌. 一种改进的麻雀搜索算法[J]. 计算机技术与发展,2022,32(3):21-26.
- [17] 冯艳红,刘建芹,贺毅朝. 基于混沌理论的动态种群萤火虫算法[J]. 计算机应用,2013,33(3):796-799.
- [18] 陈虹,赵菊芳,郭鹏飞. 基于混沌映射的分块循环 DNA 图像加密算法[J]. 计算机应用研究,2022,39(6):1865-1871.
- [19] SAMAA Y,ABEER T,AHMED S,et al. A secure and efficient cryptography system based on chaotic maps for securing data image in fog computing[J]. International Journal of Computer Network and Information Security,2023,13(1):64.
- [20] 田东平,赵天绪. 基于 Sigmoid 惯性权值的自适应粒子群优化算法[J]. 计算机应用,2008,28(12):3058-3061.
- [21] JOAQUÍN D,SALVADOR G,MOLINA D,et al. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms[J]. Swarm and Evolutionary Computation,2011,1(1):3-18.
- [22] 郁畅. 改进粒子群算法在能耗预测中的应用[J]. 计算机时代,2023(7):42-45.
- [23] ALI A H,SEYEDALI M,HOSSAM F. Harris hawks optimization: algorithm and applications[J]. Future Generation Computer Systems,2019,97:849-872.
- [24] KAVEH A,KHAYATAZAD M. A new meta-heuristic method: ray optimization[J]. Computers and Structures,2012,112:283-294.
- [25] SEYEDALI M. Moth-flame optimization algorithm: a novel nature inspired heuristic paradigm[J]. Knowledge-Based Systems,2015,89:228-249.
- [26] YU Yongbin,DENG Quanxin,REN Liyong,et al. Memristor crossbar array based ACO for image edge detection[J]. Neural Processing Letters,2020,51(2):1-15.
- [27] COELLO C A C. Use of a self-adaptive penalty approach for engineering optimization problems[J]. Computers in Industry,2000,41(2):113-127.
- [28] ESMAT R,HOSSEIN N,SAEID S. GSA: a gravitational search algorithm[J]. Information Sciences,2009,179(13):2232-2248.
- [29] 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.