

混合策略改进型秃鹰搜索算法

曹 慧, 秦江涛

上海理工大学 管理学院, 上海 200093

摘 要:目的 针对秃鹰搜索算法(Bald Eagle Search, BES)在函数优化时存在寻优精度低, 易陷入局部最优等问题, 提出一种混合策略改进型秃鹰搜索算法(Hybrid Strategy Improved Bald Eagle Search, HSIBES);方法 首先利用 Logistic 映射策略初始化种群, 使种群分布更加均匀, 其次在搜索空间阶段引入莱维飞行, 控制步长, 改善收敛效果并跳出局部最优, 最后在搜寻空间食物中使用自适应惯性权重, 提高收敛速度与精度, 平衡算法的局部与全局搜索能力;结果 将 HSIBES 算法与其他五种基准算法以及其他学者改进的算法进行对比, 通过在 9 个测试函数上进行仿真实验, 并进行 Wilcoxon 秩和检验验证 HSIBES 算法的性能, 发现 HSIBES 的结果优于其他对比算法, 与其他对比算法之间具有显著性差异;结论 实验结果表明:HSIBES 算法的寻优精度, 收敛速度以及稳定性都更好, 算法的性能更具优越性。

关键词:秃鹰搜索算法; Logistic 映射; 莱维飞行; 自适应惯性权重

中图分类号: TP301 **文献标识码:** A **doi:** 10.16055/j.issn.1672-058X.2023.0006.010

Hybrid Strategy Improved Bald Eagle Search Algorithm

CAO Hui, QIN Jiangtao

Business School, University of Shanghai for Science & Technology, Shanghai 200093, China

Abstract: Objective Aiming at the problems of low optimization accuracy and easy fall into local optimization of the Bald Eagle Search (BES) in function optimization, a Hybrid Strategy Improved Bald Eagle Search (HSIBES) was proposed.

Methods First, the Logistic mapping strategy was used to initialize the population to make the population distribution more uniform. Secondly, Levy flight was introduced in the search space stage, the step size was controlled, the convergence effect was improved and the local optimal was jumped. Finally, the adaptive inertial weight was used in the search for space food to improve the convergence speed and accuracy and to balance the local and global search capabilities of the algorithm. **Results** The HSIBES algorithm was compared with other five benchmark algorithms and other improved algorithms by other scholars, and the performance of the HSIBES algorithm was verified by simulation experiments on 9 test functions and Wilcoxon rank sum test, and it was found that the results of HSIBES were better than other comparison algorithms, and there was a significant difference between them and other comparison algorithms. **Conclusion** The experimental results show that the HSIBES algorithm has better optimization accuracy, convergence speed, and stability, and the performance of the algorithm is superior.

Keywords: bald eagle search; Logistic mapping; Levy flight; adaptive inertia weight

1 引 言

随着科学技术的进步, 需要解决的优化问题越来

越多, 如函数极值、聚类问题等, 大多都具有复杂、多维、非线性等特点, 用常规的数学方法难以解决。因此

收稿日期: 2022-10-27 修回日期: 2022-12-02 文章编号: 1672-058X(2023)06-0074-09

基金项目: 国家自然科学基金资助项目(72174121)。

作者简介: 曹慧(1997—), 女, 江苏徐州人, 硕士研究生, 从事信息管理与电子商务研究。Email: 202671092@st.usst.edu.cn.

引用格式: 曹慧, 秦江涛. 混合策略改进型秃鹰搜索算法[J]. 重庆工商大学学报(自然科学版), 2023, 40(6): 74—82.

CAO Hui, QIN Jiangtao. Hybrid strategy improved bald eagle search algorithm[J]. Journal of Chongqing Technology and Business University (Natural Science Edition), 2023, 40(6): 74—82.

一些学者根据大自然中种群活动或自然规律的启发提出多种元启发式优化算法^[1],这类算法通过模拟生物行为或物理现象,建立不同的数学模型以解决优化问题。常见的元启发式算法有粒子群优化算法(PSO)^[2],遗传算法(GA)^[3],灰狼优化算法(GWO)^[4],鲸鱼优化算法(WOA)^[5],蚁狮优化算法(ALO)^[6],麻雀搜索算法(SSA)^[7]等。这些元启发式算法具备方法简便,参数少,易于实现等优势,可以解决不同类型的优化问题,目前多被用于特征选择、路径规划等相关领域^[8-12]。此外,一些学者对这类算法也进行了相应的改进,以进一步增强算法的性能。但按照NFL(No Free Lunch)^[13]理论,不存在一种可以独立处理全部优化问题的元启发式算法,因此应不断探索新算法并对算法进行改进。

秃鹰搜索算法(Bald Eagle Search, BES)在2020年由Alsattar等^[14]受秃鹰的搜索和狩猎行为的启发而提出的元启发式算法。目前,国内外一些学者已将秃鹰搜索算法应用于实际工程问题的优化,如文献[15]将BES用于支持向量机(SVM),构建了BES-SVM预测模型,BES算法可以对SVM算法进行有效的优化,提高了SVM的预测精度。文献[16]将其应用于光伏参数估计,结果证明所使用的BES算法能够获得更好的光伏参数结果,但学者忽略了BES算法本身所存在的易陷入局部最优等问题。文献[17]将其用来解决水下无线传感器网络(UWSN)的高能耗等问题,根据实验结果,BES算法在UWSN中表现的性能具有极大的优越性。大部分文献将BES算法用于实际问题的优化中,并取得了较好的结果,但是并未考虑BES算法本身所存在的问题。而BES作为新型元启发式算法,类似于其他传统算法,具有收敛速度慢,寻优精度不高等问题,因此文献[18]提出了基于莱维飞行和模拟退火策略的秃鹰搜索算法(IBES),使用莱维飞行扩大群体的搜索范围,模拟退火策略增强原算法在局部邻域内求解精度,算法的性能得到一定程度的改善,但该算法侧重于跳出局部最优解,收敛速度和精度还有一定的提高空间。

上述改进策略侧重于算法跳出局部最优,在提升算法局部搜索上具有一定的改善,但是对于算法的收敛速度以及平衡局部和全局搜索方面还具有很大的改进空间。为了弥补这些不足,本文将提出一种混合

策略改进型秃鹰搜索算法(HSIBES),此算法利用Logistic映射策略初始化种群,使其分布更加均匀,有助于算法在全局范围内搜索。考虑莱维飞行具有长短步长交替搜索的特点,使用莱维飞行进行搜索空间中步长的控制,扩大搜寻区域,提高算法跳出局部极值点的能力。最后使用自适应惯性权重,协调秃鹰搜索算法在局部以及全局范围内的寻优能力,提高算法寻优的精度和速度。将HSIBES算法在9个基准测试函数上进行仿真实验,并且进行Wilcoxon秩和检验。实验结果说明本文提出的HSIBES算法具有更快的收敛速度和更高的收敛精度,可以更好地平衡全局和局部搜索能力。

2 秃鹰搜索算法

秃鹰种群主要分布于北美地区,它们具有视力敏锐,观察能力优秀的特点。在进行捕食食物时,秃鹰种群会先根据食物的浓度来确定搜寻空间,之后飞向所确定的区域;接着在所选择的搜寻空间中搜寻食物;最终秃鹰根据食物所在的位置,慢慢改变飞行高度,加速向下飞行,直至成功获取猎物。

Alsattar等根据秃鹰捕获食物的活动建立了秃鹰搜索算法(BES)数学模型,算法可以总结为3个阶段,分别是选取搜寻空间、搜寻空间食物以及俯冲捕获食物。

2.1 选取搜寻空间

秃鹰选取搜寻区域,根据区域内食物的数量选择最优搜索位置,易于搜寻食物,该行为用数学模型表示如下:

$$P_{i,new} = P_{best} + \alpha \times r \times (P_{mean} - P_i) \quad (1)$$

式(1)中: α 是控制秃鹰位置改变的因子,取值在1.5和2之间; r 是0和1之间的随机数; $P_{i,new}$ 为秃鹰的更新位置; P_{best} 是秃鹰种群搜寻选择的最优位置; P_{mean} 是当前种群的平均位置; P_i 表示种群中第*i*个个体的位置。

2.2 搜寻空间食物(探索)

在此搜索阶段,秃鹰种群在确定的搜寻空间中搜寻食物,并在空间中以螺旋状飞行,加速对猎物的追捕,寻求最优向下飞行捕获食物的位置。秃鹰种群以螺旋状搜寻食物的飞行轨迹可用以下数学模型进行表示:

$$\theta(i) = a \times \pi \times \text{rand}() \quad (2)$$

$$r(i) = \theta(i) + R \times \text{rand}() \quad (3)$$

$$xr(i) = r(i) \times \sin(\theta(i)) \quad (4)$$

$$yr(i) = r(i) \times \cos(\theta(i)) \quad (5)$$

$$x(i) = xr(i) / \max(|xr|) \quad (6)$$

$$y(i) = yr(i) / \max(|yr|) \quad (7)$$

其中; $\theta(i)$ 表示螺旋方程的极角, $r(i)$ 表示螺旋方程的极径; a 表示控制螺旋轨迹的因子介于 5 至 10 之间, R 用来确定搜索周期数,取值在 0.5 至 2 之间, $x(i)$ 与 $y(i)$ 为极坐标方程中个体所处的位置,取值范围均在-1 到 1 之间。秃鹰位置更新如下:

$$P_{i,\text{new}} = P_i + x(i) \times (P_i - P_{\text{mean}}) + y(i) \times (P_i - P_{i+1}) \quad (8)$$

2.3 俯冲捕获猎物

秃鹰从所选择的搜寻区域的最优位置加速飞向目标食物,所有个体也会飞向最优位置去捕获食物,飞行轨迹仍然使用极坐标数学模型进行描述,公式如下:

$$\theta(i) = a \times \pi \times \text{rand}() \quad (9)$$

$$r(i) = \theta(i) \quad (10)$$

$$xr(i) = r(i) \times \sinh(\theta(i)) \quad (11)$$

$$yr(i) = r(i) \times \cosh(\theta(i)) \quad (12)$$

$$x1(i) = xr(i) / \max(|xr|) \quad (13)$$

$$y1(i) = yr(i) / \max(|yr|) \quad (14)$$

加速飞向目标过程中秃鹰的位置更新公式为

$$\begin{cases} \delta_x = x1(i) \times (P_i - c_1 \times P_{\text{mean}}) \\ \delta_y = y1(i) \times (P_i - c_2 \times P_{\text{best}}) \end{cases} \quad (15)$$

$$P_{i,\text{new}} = \text{rand} \times P_{\text{best}} + \delta_x + \delta_y \quad (16)$$

其中; c_1 和 c_2 分别表示秃鹰向最佳位置与中心位置的运动强度,取值区间均为[1,2]。

3 混合策略改进型秃鹰搜索算法

标准的秃鹰搜索算法与其他标准元启发算法相比有良好的收敛速度和收敛精度,但与其他基准元启发式算法相同也存在寻优精度低,易陷入局部最优的缺陷,为了提升收敛速度和精度,提出了一种混合策略改进型秃鹰搜索算法,并设计了 3 种改进策略,分别是: Logistic 混沌映射,进行种群初始化,增加种群多样性; 莱维飞行,控制步长,扩大搜索范围,跳出局部最优解; 自适应惯性权重,改善群体之间的信息交流,平衡局部和全局搜索能力。

3.1 Logistic 混沌映射

秃鹰搜索算法采用随机法初始化种群,使得秃鹰个体在搜索空间内分布不均匀,从而导致算法在解空间的遍历性低,降低收敛速度和求解精度。将 Logistic 混沌映射^[19]引入 BES 算法,增加种群的多样性,提高

算法的收敛速度并提升算法在全局范围内的寻优能力,其数学模型如下所示:

$$y_{t+1} = \mu y_t (1 - y_t) \quad (17)$$

式(17)中, y_t 表示第 t 次迭代产生的混沌变量,取值范围为[0,1], μ 为控制参数,取值范围为[0,4]。当 μ 的取值为 4 时,变量会遍历整个搜索空间。

将获得的混沌序列 y_t 通过下式逆映射到搜寻空间中,得到初始化种群 P_t :

$$P_t = L_t + (U_t - L_t) y_t \quad (18)$$

其中, U_t 和 L_t 分别为搜索空间的上界和下界。

3.2 莱维飞行

莱维飞行^[20]是一个随机漫步的过程,由法国数学家莱维提出,莱维飞行大步长与小步长随机交替,可有效跳出局部最优。由于秃鹰搜索算法在选择搜索空间阶段的搜索步长是一个定值,容易陷入局部最优。而引入莱维飞行能够控制步长,扩大搜索范围,使算法有机会跳出局部最优解。根据莱维飞行,改进后的选择搜索空间阶段位置更新函数如下:

$$P_{i,\text{new}} = P_{\text{best}} + \alpha * r(P_{\text{mean}} - P_i) \times \text{Levy} \quad (19)$$

Levy 符合莱维分布,满足 $\text{Levy}(\lambda) \sim u \cdot t^{-\lambda} (1 < \lambda \leq 3)$ 。由于莱维飞行的复杂性,通常使用 Mantegna 算法对其进行模拟^[21],步长 S 的计算式为

$$S = \frac{u}{|v|^{1/\beta}} \quad (20)$$

式(20)中, u, v 均遵循正态分布:

$$u \sim N(0, \sigma_u^2), v \sim N(0, \sigma_v^2) \quad (21)$$

$$\begin{cases} \sigma_u = \left\{ \frac{\tau(1+\beta) \cdot \sin(\frac{\pi\beta}{2})}{\tau(\frac{1+\beta}{2}) \cdot \beta \cdot 2^{(\frac{\beta-1}{2})}} \right\}^{1/\beta} \\ \sigma_v = 1 \end{cases} \quad (22)$$

其中, τ 为 Gamma 函数,参数 $\beta = 1.5$ 。

3.3 自适应惯性权重

在搜寻空间食物中只依照当前秃鹰种群的信息来更新所处位置,而未考虑其余迭代中出现的位置信息,会导致在搜索更新位置中,位置更新不准确,并在一定程度上限制了算法的搜索效率。因此,将引入自适应惯性权重,在权重值较大的情况下,算法的在全局范围内的搜寻能力比较强,在权值较小的情况下,算法后期在局部范围内的寻优能力较强。因此,加入自适应惯性权重,可以有效改善群体之间的信息交流,平衡局部

和全局搜索能力,提高算法寻优的精度和速度^[22]。

自适应权重公式如式(23)所示:

$$\omega = \sin((\pi \times p) / (2 \times M) + \pi) + 1 \quad (23)$$

其中, p 为当前迭代次数, M 为最大迭代次数。

将式(23)代入式(8),得到改进的秃鹰位置更新函数:

$$P_{i,new} = P_i + \omega \times x(i) \times (P_i - P_{mean}) + \omega \times y(i) \times (P_i - P_{i+1}) \quad (24)$$

3.4 算法流程

step1:初始化秃鹰算法种群规模,空间维度等参数,使用 Logistic 混沌映射进行秃鹰种群初始化;

step2:计算适应度值,获得最优个体;

step3:秃鹰选择搜索空间,利用式(19)更新位置;

step4:秃鹰在搜索空间搜索猎物,利用式(24)更新位置;

step5:秃鹰俯冲利用式(16),更新位置;

step6:当符合结束条件时,输出最优结果,否则继续进行从 step2 到 step6 的流程。

3.5 时间复杂度分析

假设种群规模为 N ,空间维度为 D ,则 BES 算法参数初始化的时间复杂度为 $O(1)$,计算函数适应度为 $O(N)$,迭代过程中种群复杂为 $O(ND)$,BES 算法总的时间复杂度为

$$O(1) + O(N) + O(ND) = O(ND) \quad (25)$$

在 HSIBES 算法中,随机初始化替换为 Logistic 混沌初始化时间复杂度为 $O(ND)$,计算适应度为 $O(N)$,引入莱维飞行和自适应惯性权重进行位置更新所对应的时间复杂度均为 $O(ND)$,则 HSIBES 算法总的时间复杂度为

$$O(ND) + O(N) + O(ND) + O(ND) = O(ND) \quad (26)$$

基于上述分析,HSIBES 算法和 BES 算法相比,时间复杂度并未增加。

4 实验结果分析

实验测试结果均在 Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz,64 位 Windows10 操作系统和 MATLAB R2018b 上实现。

为了验证混合策略改进型秃鹰搜索算法的有效性,将从以下几个部分进行实验验证:

(1) 将 HSIBES 与基本秃鹰搜索算法(BES)^[14]、粒子群优化算法(PSO)^[2]、鲸鱼优化算法(WOA)^[5]、蚁

狮算法(ALO)^[6]、灰狼优化算法(GWO)^[4]这5个基本元启发式算法进行对比,验证混合策略改进型秃鹰搜索算法的寻优能力和鲁棒性。

(2) 将 HSIBES 与文献[18]中改进的秃鹰搜索算法进行对比,验证本文改进算法具有一定的竞争力。

(3) 通过 Wilcoxon 秩和检验进行差异性检验,验证 HSIBES 算法与其他对比算法的差异性。

为了检验提出的改进的秃鹰搜索算法的鲁棒性以及有效性,从文献[23]中选择了9个具备不同特征的基准测试函数进行实验测试,具体函数表达式如表1所示。这9个用于实验的测试函数主要分为单峰函数以及多峰函数,其中 $f_1(x) - f_6(x)$ 是单峰函数, $f_7(x) - f_9(x)$ 是多峰函数,算法的局部搜索能力与收敛速度可用 $f_1(x) - f_6(x)$ 测试,算法在全局范围内的寻优能力以及跳出局部极值点的能力可用 $f_7(x) - f_9(x)$ 测试。

表1 基本测试函数

Table 1 Basic test functions

函数表达式	搜索区间	最优值
$f_1(x) = \sum_{i=1}^n x_i^2$	$[-100, 100]$	0
$f_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	$[-10, 10]$	0
$f_3(x) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2$	$[-100, 100]$	0
$f_4(x) = \max\{ x_i , 1 \leq i \leq n\}$	$[-100, 100]$	0
$f_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-30, 30]$	0
$f_6(x) = \sum_{i=1}^n ix_i^4 + \text{random}[0, 1)$	$[-1.28, 1.28]$	0
$f_7(x) = \sum_{i=1}^n [x_i^2 - 10\cos(2\pi x_i) + 10]$	$[-5.12, 5.12]$	0
$f_8(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$	$[-32, 32]$	0
$f_9(x) = \frac{1}{4000}\sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\frac{x_i}{\sqrt{i}} + 1$	$[-600, 600]$	0

4.1 与其他基准算法进行对比

将提出的 HSIBES 算法与 BES^[14]、PSO^[2]、WOA^[5]、ALO^[6]、GWO^[4]算法在9个基准测试函数上进行对比实验,为保证实验的公平性,种群规模均设为30,最大迭

代次数为 500,维度设为 30 维,各个算法其他参数设置与相应参考文献一致,实验将记录每种算法在测试函数上独立运行 30 次的平均值、方差和平均耗时以便进行对比,结果如表 2 所示。

表 2 函数测试实验结果
Table 2 Results of function test experiment

测试函数	统计量	HSIBES	BES	GWO	ALO	PSO	WOA
$f_1(x)$	平均值	7.85E-226	2.68E-39	8.58E-28	1.95E-03	3.96E-04	4.63E-76
	标准差	0.00E+00	1.01E-38	1.07E-27	2.80E-03	6.67E-04	1.60E-75
	耗时/s	0.262 0	1.934 4	0.084 2	6.744 0	0.052 3	0.074 0
$f_2(x)$	平均值	5.32E-115	9.16E-26	8.42E-17	4.46E+01	7.03E+00	2.09E-50
	标准差	1.76E-114	2.44E-25	6.41E-17	4.36E+01	7.80E+00	9.50E-50
	耗时/s	0.273 7	2.134 8	0.086 4	6.796 6	0.051 5	0.084 6
$f_3(x)$	平均值	3.95E-218	8.37E-03	1.34E-05	4.31E+03	7.92E+01	4.61E+04
	标准差	0.00E+00	4.49E-02	3.76E-05	1.93E+03	3.17E+01	1.26E+04
	耗时/s	0.607 8	2.374 5	0.197 0	6.829 8	0.166 9	0.241 7
$f_4(x)$	平均值	6.82E-117	2.38E-01	8.99E-07	1.84E+01	1.10E+00	5.90E+01
	标准差	3.30E-116	3.91E-01	1.25E-06	4.43E+00	1.91E-01	2.59E+01
	耗时/s	0.273 9	1.881 7	0.095 0	6.828 1	0.048 6	0.061 1
$f_5(x)$	平均值	2.90E+01	2.22E+01	2.72E+01	3.43E+02	3.09E+03	2.81E+01
	标准差	2.17E-02	9.70E-01	6.61E-01	4.71E+02	1.61E+04	4.73E-01
	耗时/s	0.303 4	1.650 7	0.097 6	8.280 8	0.077 8	0.094 0
$f_6(x)$	平均值	1.39E-04	2.63E-03	1.70E-03	2.55E-01	2.99E+00	5.38E-03
	标准差	1.85E-04	2.66E-03	1.16E-03	1.04E-01	4.85E+00	4.88E-03
	耗时/s	0.599 5	3.317 4	0.196 3	10.099 5	0.153 1	0.213 9
$f_7(x)$	平均值	0.00E+00	3.12E+01	2.20E+00	8.39E+01	9.60E+01	3.79E-15
	标准差	0.00E+00	5.02E+01	3.80E+00	2.52E+01	2.77E+01	1.42E-14
	耗时/s	0.386 5	2.482 0	0.139 0	11.137 9	0.087 9	0.124 0
$f_8(x)$	平均值	8.88E-16	1.12E-01	1.04E-13	4.41E+00	2.73E-01	4.44E-15
	标准差	0.00E+00	4.22E-01	1.87E-14	2.80E+00	4.92E-01	2.25E-15
	耗时/s	0.440 8	4.496 1	0.138 7	10.109 1	0.086 9	0.112 4
$f_9(x)$	平均值	0.00E+00	1.00E-05	5.84E-03	5.91E-02	8.38E-03	1.37E-02
	标准差	0.00E+00	3.03E-05	8.90E-03	2.17E-02	9.31E-03	4.17E-02
	耗时/s	0.390 1	3.264 8	0.125 9	7.975 9	0.085 4	0.160 1

从均值来看,提出的 HSIBES 算法在测试函数 $f_7(x)$ 和 $f_9(x)$ 的均值达到了理论最优值,说明 30 次独立运行中得到的结果精度较高,稳定性较好。HSIBES 算法在测试函数 $f_5(x)$ 的均值结果不是算法结果里最优的,劣于其他算法,但是差异不是非常显著,求解精度在 1 个数量级以内,在可以接受的范围内。在测试函数 $f_1(x)$ — $f_4(x)$ 、 $f_6(x)$ 、 $f_8(x)$ 上 HSIBES 算法明显优

于其他对比算法,尤其在函数 $f_1(x)$ 、 $f_3(x)$ 和 $f_4(x)$ 上,HSIBES 算法相较于 BES、GWO、ALO、PSO、WOA 均提升了 100~200 个数量级左右。在 $f_2(x)$ 、 $f_6(x)$ 和 $f_8(x)$ 函数上也提升了多个数量级,说明混合策略改进型秃鹰搜索算法求解精度较好。从标准差来看 HSIBES 算法测试函数上的标准差均是所有结果里最优的,说明提出的算法的鲁棒性较好。从平均耗时来看,PSO 算

法整体耗时最短,HSIBES 算法相对于标准 BES 的平均耗时要小,说明引进的改进策略并未降低原算法的执行效率。经过上述分析,HSIBES 算法在平均值标准差以及耗时方面整体来说要优于对比算法,具有较好的收敛速度和收敛精度,稳定性也较好。

4.2 与其他改进算法的对比

为了突出提出的混合策略改进型秃鹰搜索算法相比于其他学者改进的秃鹰搜索算法的竞争优势,选取文献[18]改进的秃鹰搜索算法 IBES 进行对比,和上文设置统一的参数条件: $c_1=c_2=\alpha=2, a=10, R=1.5$,种群规模均设为 30,最大迭代次数为 500,维度设为 30 维,在上文的测试函数及运行环境上进行测试,独立运行 30 次,对所得结果求平均值与标准差,与本文提出的 HSIBES 进行对比分析,结果如表 3 所示。

表 3 函数测试结果

Table 3 Function test results

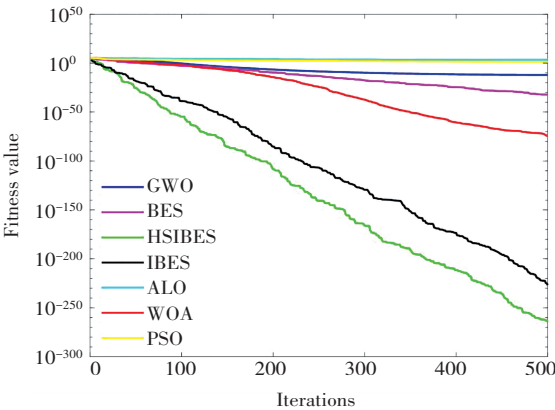
函 数	HSIBES			IBES		
	平均值	标准差	耗时/s	平均值	标准差	耗时/s
$f_1(x)$	7.85E-226	0.00E+00	0.262 0	1.99E-206	0.00E+00	1.186 8
$f_2(x)$	5.32E-115	1.76E-114	0.273 7	3.94E-105	2.12E-104	1.288 3
$f_3(x)$	3.95E-218	0.00E+00	0.607 8	1.15E-191	0.00E+00	5.777 8
$f_4(x)$	6.82E-117	3.30E-116	0.273 9	7.59E-108	4.01E-107	1.314 1
$f_5(x)$	2.90E+01	2.17E-02	0.303 4	5.57E-04	9.72E-04	2.145 1
$f_6(x)$	1.39E-04	1.85E-04	0.599 5	1.32E-03	2.55E-03	3.380 8
$f_7(x)$	0.00E+00	0.00E+00	0.386 5	0.00E+00	0.00E+00	2.486 2
$f_8(x)$	8.88E-16	0.00E+00	0.440 8	8.88E-16	0.00E+00	2.738 6
$f_9(x)$	0.00E+00	0.00E+00	0.390 1	0.00E+00	0.00E+00	2.720 1

对于函数 $f_1(x)$ 和 $f_3(x)$,HSIBES 算法的平均寻优精度高于 IBES 算法,高出 20 个数量级左右,两者的标准差相同,说明两种算法在函数 $f_1(x)$ 和 $f_3(x)$ 的稳定性相当,在测试函数 $f_2(x)$ 、 $f_4(x)$ 和 $f_6(x)$ 上,HSIBES 算法的均值和标准差均高于 IBES 算法,在测试函数 $f_5(x)$ 上,HSIBES 算法的寻优精度和稳定性不及 IBES,在函数 $f_7(x)$ — $f_9(x)$ 上,二者的均值和标准差的求解结果相同。从平均值和标准差来看,HSIBES 算法在 55%以上的测试函数上优于 IBES 算法,33%测试函数和 IBES 算法结果相同。从平均耗时的结果来看,HSIBES 算法在 9 个函数上的耗时都比 IBES 算法的耗时短。综上,和对比算法 IBES 相比,HSIBES 算法具有一定的竞争优势。

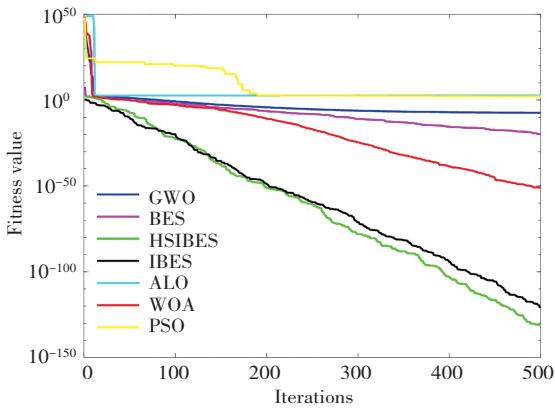
4.3 算法收敛曲线对比分析

为了更加直观地反映 HSIBES 算法的性能,给出了算法在函数上运行 30 次中其中一次的收敛图,如图 1 所示。和其他基本算法相比,HSIBES 算法在单峰函数 $f_1(x)$ — $f_4(x)$ 、 $f_6(x)$ 的收敛速度和收敛精度明显高于其他对比算法,虽然在 $f_5(x)$ 函数上的精度略低于其他算法,但是其收敛速度明显优于其他对比算法,在多峰函数 $f_7(x)$ — $f_9(x)$ 上,HSIBES 算法可以快速地收敛并且跳出局部最优,说明混合策略改进型秃鹰搜索算法 HSIBES 可以提高基本算法的收敛速度,跳出局部极值点,进行全局范围的寻优。

和 IBES 对比算法相比,HSIBES 算法除了在函数 $f_5(x)$ 上表现不佳,在其他函数上都有较好的寻优精度和收敛速度。在函数 $f_1(x)$ — $f_4(x)$ 上,HSIBES 算法和 IBES 算法相比精度显著提升,在函数 $f_6(x)$ 上 HSIBES 算法出现了多次拐点,但停滞次数相对较少,并且在后期跳出局部最优解,在函数 $f_7(x)$ — $f_9(x)$ 上,HSIBES 算法可在较短的时间内和其他改进算法达到相同的寻优值,综上所述本文提出的 HSIBES 算法具有较好的收敛速度和寻优精度,能更好地平衡算法局部和全局搜索能力。



(a) $f_1(x)$ 收敛曲线



(b) $f_2(x)$ 收敛曲线

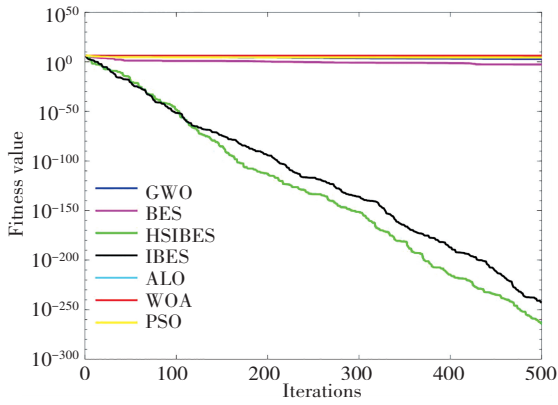
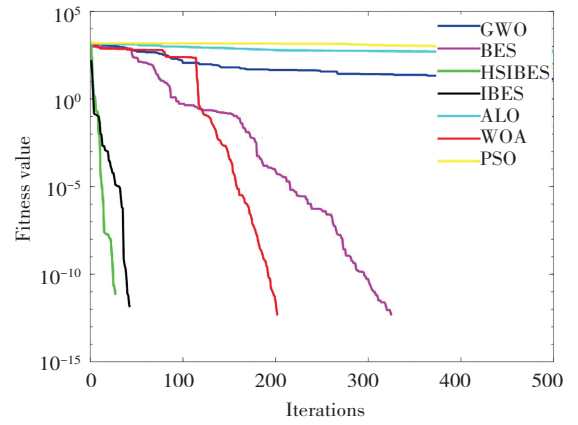
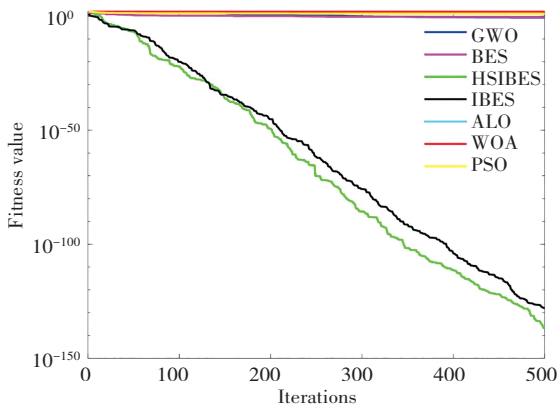
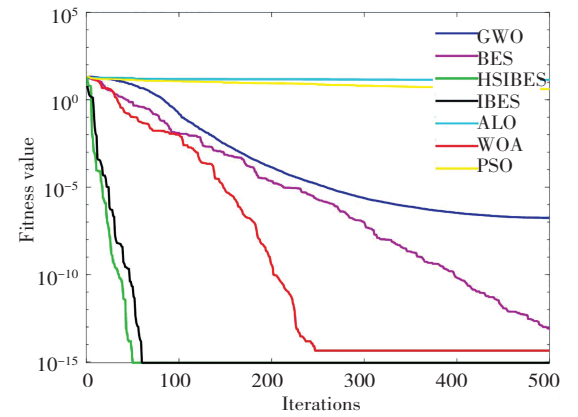
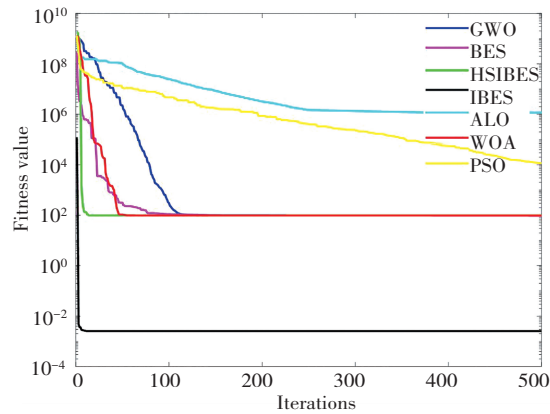
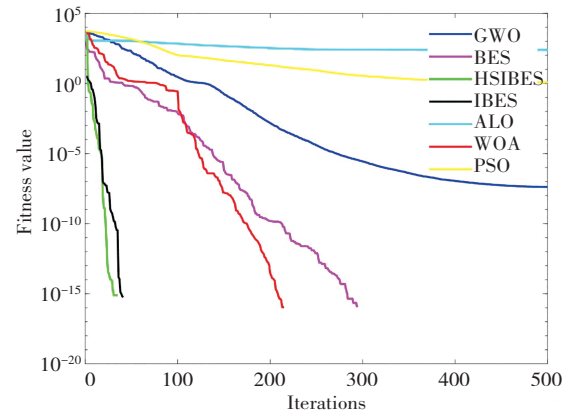
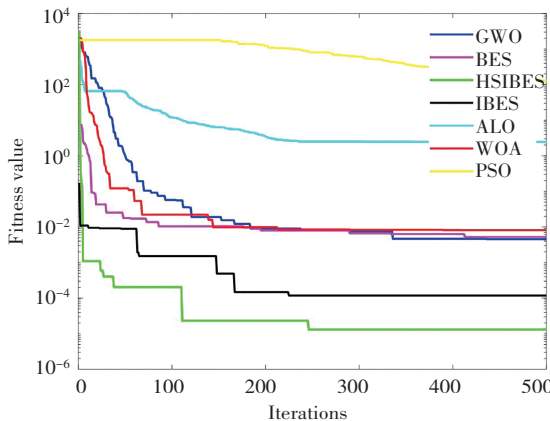
(c) $f_3(x)$ 收敛曲线(g) $f_7(x)$ 收敛曲线(d) $f_4(x)$ 收敛曲线(h) $f_8(x)$ 收敛曲线(e) $f_5(x)$ 收敛曲线(i) $f_9(x)$ 收敛曲线(f) $f_6(x)$ 收敛曲线

图 1 各测试函数下的收敛曲线

Fig. 1 Convergence curves under each test function

4.4 Wilcoxon 秩和检验

运用 Wilcoxon 秩和检验^[24]的方法来检验提出的 HSIBES 算法与其他算法的显著性差别。将获得的实验结果在 5% 的显著性水平下进行统计检验,在 p 大于 0.05 的情况下,说明两种算法性能相差不大,否则两种比较算法的性能具有显著性差异。为了判断 HSIBES 算法与其他算法的显著性区别,将以上算法在函数上

独立运行 30 次的结果作为样本,进行实验验证。得到的 Wilcoxon 秩和检验的 p 值结果如表 4 所示,由于 HSIBES 算法不能和本身进行比较,所以在这里不再列

出 HSIBES 算法的 p 值,当实验样本数据一样时,说明两个对比算法性能相当,此时数据无效,在下表中使用 NaN 表示。

表 4 Wilcoxon 秩和检验 p 值
Table 4 Wilcoxon rank and test p -value

测试函数	HSIBES/BES	HSIBES/GWO	HSIBES/ALO	HSIBES/PSO	HSIBES/WOA	HSIBES/IBES
$f_1(x)$	3. 02E-11	3. 02E-11	3. 02E-11	3. 02E-11	3. 02E-11	2. 84E-04
$f_2(x)$	3. 02E-11	3. 02E-11	3. 02E-11	3. 02E-11	3. 02E-11	8. 68E-03
$f_3(x)$	3. 02E-11	3. 02E-11	3. 02E-11	3. 02E-11	3. 02E-11	2. 43E-05
$f_4(x)$	3. 02E-11	3. 02E-11	3. 02E-11	3. 02E-11	3. 02E-11	3. 26E-01
$f_5(x)$	3. 02E-11	3. 02E-11	1. 11E-06	3. 26E-07	3. 02E-11	3. 02E-11
$f_6(x)$	6. 69E-11	1. 78E-10	3. 02E-11	3. 02E-11	1. 17E-09	1. 84E-02
$f_7(x)$	2. 21E-06	1. 17E-12	1. 21E-12	1. 21E-12	1. 61E-01	NaN
$f_8(x)$	4. 16E-14	1. 10E-12	1. 21E-12	1. 21E-12	9. 84E-10	NaN
$f_9(x)$	4. 19E-02	6. 61E-04	1. 21E-12	1. 21E-12	2. 16E-02	NaN

由表 4 可知,根据对比算法在测试函数 $f_7(x)$ — $f_9(x)$ 的检验结果来看,HSIBES 算法与 IBES 算法性能相当,在函数 $f_4(x)$ 上,二者性能显著性差异不明显,在函数 $f_7(x)$ 上,HSIBES 算法与 WOA 算法显著性差异不明显,除此之外,其余 p 值均小于 5%,说明 HSIBES 算法和其他对比算法之间具有显著性差异,总体看来,本文提出的 HSIBES 算法的优越性在统计上是显著的,与其他对比算法具有显著性差异。

5 结束语

根据 BES 算法存在易陷入局部最优,收敛精度低的问题,提出了混合策略改进的秃鹰搜索算法(HSIBES),利用 Logistic 映射策略初始化种群,使种群分布更加均匀,其次引入莱维飞行,其长短步长交替搜索,控制搜索步长,有利于提高解的质量。最后在探索阶段使用自适应惯性权重,提高了算法寻优的精度和速度,平衡了局部和全局探索能力。最后在 9 个基准测试函数上进行了仿真实验,并与 BES、PSO、WOA、ALO、GWO 以及其他学者改进的 IBES 算法进行对比实验,分析得出了提出的 HSIBES 算法收敛速度、收敛精度以及鲁棒性都表现较好,并通过 Wilcoxon 秩和检验

验证了 HSIBES 算法与其他算法的显著性差异。在后续的研究中,将会把 HSIBES 算法用于实际工程问题中,如神经网络的优化,图像分割等。

参考文献(References):

[1] 霍星, 张飞, 邵堃, 等. 改进的元启发式优化算法及其在图像分割中的应用[J]. 软件学报, 2021, 32(11): 452—4467.
HUO Xing, ZHANG Fei, SHAO Kun, et al. Improved meta-heuristic optimization algorithm and its application in image segmentation[J]. Journal of Software, 2021, 32(11): 3452—3467.

[2] WANG Q X, CHEN S L, LUO X. An adaptive latent factor model via particle swarm optimization[J]. Neurocomputing, 2019, 369(C): 176—184.

[3] MENG S, ZHANG C C. Composition of web services of multi-population adaptive genetic algorithm based on cosine improvement[J]. Journal of Computer and Communications, 2021, 9(6): 109—119.

[4] ZHANG X Q, ZHANG Y Y, MING Z F. Improved dynamic grey wolf optimizer[J]. Frontiers of Information Technology & Electronic Engineering, 2021, 22(6): 877—890.

[5] GUO W Y, LIU T, DAI F, et al. An improved whale optimization algorithm for feature Selection [J]. CMC-

- Computers, Materials & Continua, 2020, 62(1): 337—354.
- [6] MIRJALILI S. The ant lion optimizer [J]. Advances in Engineering Software, 2015, 83(C): 80—98.
- [7] XUE J K, SHEN B. A novel swarm intelligence optimization approach: sparrow search algorithm [J]. System Science & Control Engineering, 2020, 8(1): 22—34.
- [8] 赵转哲, 叶国文, 张宇, 等. 基于改进的二进制蚁狮算法的特征选择模型及应用[J]. 计算机集成制造系统, 2021, 27(7): 1898—1908.
- ZHAO Zhuan-zhe, YE Guo-wen, ZHANG Yu, et al. Feature selection model based on improved binary ant lion optimizer and its application [J]. Computer Integrated Manufacturing Systems, 2021, 27(7): 1898—1908.
- [9] WANG S K, JIA H M, PENG X X. Modified salp swarm algorithm based multilevel thresholding for color image segmentation[J]. Mathematical Biosciences and Engineering: MBE, 2019, 17(1): 700—724.
- [10] LANG C B, JIA H M. Kapur's entropy for color image segmentation based on a hybrid whale optimization algorithm[J]. Entropy, 2019, 21(3): 318—346.
- [11] YUAN X, YUAN X W, WANG X H. Path planning for mobile robot based on improved bat algorithm[J]. Sensors, 2021, 21(13): 4389.
- [12] SUN P, SHAN R. Predictive control with velocity observer for cushion robot based on PSO for path planning[J]. Journal of Systems Science and Complexity, 2020, 33(4): 988—1011.
- [13] WOLPERT D H, MACREADY W G. No free lunch theorems for optimization [J]. IEEE Transactions on Evolutionary Computation, 1997, 1(1): 67—82.
- [14] ALSATTAR H A, ZAIDAN A A, ZAIDAN B B. Novel meta-heuristic bald eagle search optimisation algorithm [J]. Artificial Intelligence Review, 2020, 53(3): 2237—2264.
- [15] HUANG Y, JIANG H, WANG W F, et al. Soil moisture content prediction model for tea plantations based on SVM optimised by the bald eagle search algorithm[J]. Cognitive Computation and Systems, 2021, 3(4): 351—360.
- [16] NICAIRE N F, STEVE P N, SALOME N E, et al. Parameter estimation of the photovoltaic system using bald eagle search (BES) algorithm [J]. International Journal of Photoenergy, 2021(12): 47—56.
- [17] KAPILESWAR N, KUMAR P P. Energy efficient routing in IOT based UWSN using bald eagle search algorithm [J]. Transactions on Emerging Telecommunications Technologies, 2021, 33(1): 102—109.
- [18] 贾鹤鸣, 姜子超, 李瑶. 基于改进秃鹰搜索算法的同步优化特征选择[J]. 控制与决策, 2022, 37(2): 445—454.
- JIA He-ming, JIANG Zi-chao, LI Yao. Simultaneous feature selection optimization based on improved bald eagle search algorithm[J]. Control and Decision, 2022, 37(2): 445—454.
- [19] 吴忠强, 王国勇, 谢宗奎, 等. 基于 IALO 算法的蓄电池参数辨识[J]. 计量学报, 2021, 42(9): 1206—1213.
- WU Zhong-qiang, WANG Guo-yong, XIE Zong-kui, et al. Parameter identification of battery based on IALO algorithm[J]. Acta Metrologica Sinica, 2021, 42(9): 1206—1213.
- [20] DALLEL N, DIAB M. Optimisation of multi-objective problems using an efficient levy flight grasshopper algorithm [J]. International Journal of High Performance Systems Architecture, 2022, 11(1): 26—35.
- [21] 张严, 秦亮曦. 基于 Levy 飞行策略的改进樽海鞘群算法[J]. 计算机科学 2020, 47(7): 154-160.
- ZHANG Yan, QIN Liang-xi. Improved salp swarm algorithm based on levy flight strategy[J]. Computer Science, 2020, 47(7): 154—160.
- [22] PRAKASH M, CHINNAMUTHU P. Techno-economic modeling of integrated renewable energy system using adaptive inertia weight based PSO [J]. Journal of Scientific and Industrial Research, 2020, 79(7): 647—652.
- [23] 王波, 王浩, 杜晓昕, 等. 基于亚群和差分进化的混合蜻蜓优化算法[J/OL]. 计算机应用, 2022(11): 1—13. <http://kns.cnki.net/kcms/detail/51.1307.TP.20220920.1114.014.html>.
- WANG Bo, WANG Hao, DU Xiao-xin, et al. Hybrid dragonfly algorithm based on subpopulation and differential evolution [J/OL]. Journal of Computer Applications, 2022(11): 1—13. <http://kns.cnki.net/kcms/detail/51.1307.TP.20220920.1114.014.html>.
- [24] WILCOXON F. Individual comparisons by ranking methods[J]. Biometrics Bulletin, 1945, 1(6): 80—83.