

# 基于混合策略的鹈鹕优化算法

刘魏岩, 齐迹\*, 梁红, 林钰川

(齐齐哈尔大学通信与电子工程学院, 黑龙江 齐齐哈尔 161006)

**摘要:** 为了提高鹈鹕优化算法的优化性能与稳定性, 提出一种基于混合策略的鹈鹕优化算法。在鹈鹕算法备选解的生成机制中使用偏好权重策略, 引导备选解进行多样性的探索。采用随机搜索策略更新部分目标函数值较差的备选解位置, 使得备选解能够跳出局部最优限制。使用大范围的自适应搜索半径策略, 增大各备选解发现更优解的可能。在18个测试函数上结合对比算法开展寻优实验, 分析寻优结果并进行 Wilcoxon 秩和检验, 验证了改进后的鹈鹕优化算法具有更好的寻优性能与稳定性。

**关键词:** 鹈鹕优化算法; 混合策略; 偏好权重; 函数优化

**中图分类号:** TP301 **文献标志码:** A

**引用格式:** 刘魏岩, 齐迹, 梁红. 基于混合策略的鹈鹕优化算法[J]. 山东大学学报(理学版), 2025, 60(9): 52-61.

## A pelican optimization algorithm based on hybrid strategy

LIU Weiyan, QI Ji\*, LIANG Hong, LIN Yuchuan

(College of Communication and Electronic Engineering, Qiqihar University, Qiqihar 161006, Heilongjiang, China)

**Abstract:** In order to improve the optimization performance and stability of the pelican optimization algorithm, a hybrid strategy-based pelican optimization algorithm is proposed. In the generation mechanism of candidate solutions for the pelican algorithm, the preference weight strategy is employed to guide the candidate solutions towards exploring diversity. A random search strategy is used to update partial alternative solution positions, which helps these candidate solutions with poor objective function values escape local optima. A large-range adaptive search radius strategy has been applied to enhance the algorithm, increasing the probability for each candidate solution to discover better solutions. Combined with comparative algorithms, optimization experiments are conducted on 18 test functions. The optimization results are analyzed and Wilcoxon rank sum tests are performed to verify that this improved Pelican optimization algorithm has better optimization performance and stability.

**Key words:** pelican optimization algorithm; hybrid strategy; preference weight; function optimization

## 0 引言

智能优化算法具有寻优迅速且鲁棒性强的特点, 能够很好的解决优化问题, 目前被广泛应用于图像处理<sup>[1-2]</sup>、电力系统<sup>[3-4]</sup>、信号检测<sup>[5-6]</sup>、深度学习<sup>[7-8]</sup>、教学管理<sup>[9-10]</sup>等领域。文献[11]通过精英反向策略提升初代种群质量, 并使用混沌映射避免鲸鱼优化算法存在的局部最优问题。文献[12]在麻雀搜索算法基础上引入莱维飞行机制, 平衡算法的全局寻优与局部寻优能力, 避免算法陷入局部最优。文献[13]在鲸鱼优化算法中加入自适应惯性权重和反向学习策略, 提高了算法种群的多样性和收敛精度, 引入的参数非线性衰减策略增强了鲸鱼优化算法在高维问题上的优化性能。文献[14]用透镜成像反向学习策略和精英池策略解

收稿日期: 2024-01-22; 网络出版时间: 2024-12-26 11:40:14

基金项目: 河北省自然科学基金资助项目(F2023107002); 黑龙江省省属本科高校基本科研业务费资助项目(135509226); 黑龙江省农业多维传感器信息感知工程技术研究中心开放课题资助项目(DWCGQKF202103); 黑龙江省高等教育教学改革研究资助项目(SJGY20210952)

第一作者: 刘魏岩(1989—), 男, 讲师, 硕士, 研究方向为智能优化算法。E-mail: l\_weiyan@163.com

\* 通信作者: 齐迹(1979—), 女, 教授, 博士, 研究方向为网络化系统控制与智能优化。E-mail: qi\_ji\_1979@126.com

决了侏儒猫鼬优化算法收敛速度较慢、易陷入局部最优的问题,并平衡了全局搜索与局部搜索性能。文献[15]将广泛学习、精英学习和协调学习等策略应用于灰狼优化算法,改善了算法的平衡能力以及收敛速度。文献[16]引入透镜成像学习和混沌映射提升了平衡优化器算法的寻优性能。文献[17]使用自适应高斯变异策略使哈里斯鹰优化算法能够跳出局部最优。文献[18]在每次迭代中对单个备选解的位置进行差分变异,增加了海鸥优化算法种群的多样性。文献[19]使用较优备选解引导群体中其他备选解开展搜索,保持了种群的多样性,可使阿奎拉优化算法跳出局部最优。文献[20]在教与学优化算法基础上使用混沌搜索和余弦函数分阶段扰动,增加了备选解的多样性。

鹈鹕优化算法<sup>[21]</sup>(pelican optimization algorithm, POA)通过模拟鹈鹕的行为与策略,对待优化问题进行优化。为了提高鹈鹕优化算法的寻优精度和稳定性,本文提出一种基于混合策略鹈鹕优化算法(hybrid strategy pelican optimization algorithm, HSPOA)。HSPOA使用偏好权重策略生成猎物位置引导备选解移动,增强备选解移动方向的多样性;采用随机搜索策略使得备选解有机会跳出局部最优;通过自适应调整的搜索半径扩大了各备选解的搜索范围。实验验证表明,HSPOA在寻优精度、收敛速度和稳定性方面具有更好的性能。

## 1 鹈鹕优化算法

在鹈鹕优化算法中,种群内每只鹈鹕的位置代表解空间内备选解的位置,鹈鹕的行为包括2个阶段:负责全局搜索的位置移动行为,即迭代时种群内的全体备选解随机向猎物移动的行为( $S_1$ );负责局部搜索的狩猎区域捕猎行为,即备选解对各自位置周围区域的探索行为( $S_2$ )。

设POA备选解位置为 $\mathbf{X}$ ,若种群个数为 $N$ 、待优化问题维数为 $D$ ,则第 $i$ 只鹈鹕在解空间第 $j$ 维的初代位置 $X_{i,j}$ 由式(1)产生:

$$X_{i,j} = l_j + k(u_j - l_j), \quad (1)$$

其中, $k \in (0, 1)$ ,  $l_j$ 、 $u_j$ 是解空间第 $j$ 维的下界和上界, $i = 1, 2, \dots, N$ ,  $j = 1, 2, \dots, D$ 。

### 1.1 位置移动行为

在位置移动行为阶段,算法通过模拟鹈鹕的位置移动行为进行全局搜索。确定初代备选解位置后,随机选择备选解在解空间的位置作为猎物 $\mathbf{P}$ ,种群内其他备选解按式(2)移动到新的位置

$$\mathbf{X}_i^{S_1} = \begin{cases} \mathbf{X}_i + k(\mathbf{P} - I\mathbf{X}_i), & F_p < F_i, \\ \mathbf{X}_i + k(\mathbf{X}_i - \mathbf{P}), & \text{其他}, \end{cases} \quad (2)$$

其中, $F_p$ 、 $F_i$ 为 $\mathbf{P}$ 与 $\mathbf{X}_i$ 的目标函数值, $I$ 是取值为1或2的随机数,用于控制备选解的位移。

按式(3)确定是否接受 $\mathbf{X}_i^{S_1}$ :

$$\mathbf{X}_i = \begin{cases} \mathbf{X}_i^{S_1}, & F_i^{S_1} < F_i, \\ \mathbf{X}_i, & \text{其他}, \end{cases} \quad (3)$$

即新位置 $\mathbf{X}_i^{S_1}$ 的目标函数值 $F_i^{S_1}$ 更优则接受新位置,否则保持位置不变。

### 1.2 区域捕猎行为

在区域捕猎行为阶段,算法通过模拟鹈鹕的捕食行为进行局部搜索,捕猎的区域在各备选解的周围。第 $i$ 只鹈鹕在解空间第 $j$ 维的位置即各备选解按式(4)移动到新的位置

$$X_{i,j}^{S_2} = X_{i,j} + R \left( 1 - \frac{t}{T} \right) (2k - 1) X_{i,j}, \quad (4)$$

其中, $R = 0.2$ ,  $t$ 为当前迭代次数, $T$ 为所设定的最大迭代次数。随着迭代的进行, $R \left( 1 - \frac{t}{T} \right)$ 将从0.2逐渐减小,各备选解局部搜索的范围将逐渐缩小。

按式(5)确定是否接受 $\mathbf{X}_i^{S_2}$ :

$$\mathbf{X}_i = \begin{cases} \mathbf{X}_i^{S_2}, & F_i^{S_2} < F_i, \\ \mathbf{X}_i, & \text{其他}, \end{cases} \quad (5)$$

即新位置  $X_i^{S_2}$  的目标函数值  $F_i^{S_2}$  更优则接受新位置,否则保持位置不变。

历次迭代均按  $X_i^{S_1}$  和  $X_i^{S_2}$  更新种群备选解的位置,直至算法迭代完毕并输出最优解。

## 2 鹈鹕优化算法的改进

### 2.1 偏好权重

为增强算法位置移动行为阶段产生备选解的多样性,提高算法的寻优性能,HSPOA 在每次迭代开始后,针对全体备选解,使用基于备选解目标函数值的偏好权重策略计算猎物  $P$  的位置,以此引导各备选解历次迭代中的位置更新。

从  $N$  个备选解中选择目标函数值较好的  $G$  个组成  $X_{Best} = [X_{best_1}; X_{best_2}; \dots; X_{best_G}]$ 。  $X_{Best}$  对应的目标函数值为  $F_{Best} = [F_{best_1}, F_{best_2}, \dots, F_{best_G}]^T$ ,使用式(6)对  $F_{Best}$  内的数值进行处理

$$\lambda_g = \frac{F_{Best}^{mean} - F_{best_g}}{F_{Best}^{max} - F_{Best}^{min}}, \quad (6)$$

其中,  $g=1, 2, \dots, G$ ,  $F_{Best}^{mean}$ 、 $F_{Best}^{max}$ 、 $F_{Best}^{min}$  分别为  $F_{Best}$  的均值、最大值和最小值。

猎物  $P$  的位置表示为

$$P = \lambda_1 X_{best_1} + \lambda_2 X_{best_2} + \dots + \lambda_G X_{best_G} \quad (7)$$

若式(6)中  $F_{Best}^{max} - F_{Best}^{min} = 0$ ,则  $P$  仍按原方式选择;若式(7)生成  $P$  的位置第  $j$  维超出解空间上下界,则使用式(1)对第  $j$  维位置进行替换。

### 2.2 随机搜索

在算法历次迭代寻优过程中会有部分备选解的目标函数值较差。为了避免备选解陷入局部最优,参考人工鱼(artificial fish, AF)优化算法中的随机行为,模拟鹈鹕追逐逃跑的鱼类,在 HSPOA 引入随机搜索策略。

设种群内目标函数值较差的  $W$  个备选解位置为  $X_{Worst} = [X_{worst_1}; X_{worst_2}; \dots; X_{worst_W}]$ ,其对应的目标函数值为  $F_{Worst} = [F_{worst_1}, F_{worst_2}, \dots, F_{worst_W}]^T$ ,使用式(8)进行随机搜索

$$X_{Worst}^{new} = X_{Worst} + k(X_{median} - X_{Worst}), \quad (8)$$

其中,  $X_{Worst}^{new}$  为  $X_{Worst}$  更新后的备选解位置,  $X_{median}$  为全体备选解位置的中位数,  $k = [k_{11}, k_{21}, \dots, k_{w1}]^T$ ,第  $q$  行第 1 列的元素  $k_{q1} \in (0, 1)$ 。

计算  $X_{Worst}^{new}$  的目标函数值  $F_{Worst}^{new}$ ,按式(9)确定是否替换相应备选解:

$$X_{worst_w} = \begin{cases} X_{worst_w}^{new}, & F_{worst_w}^{new} < F_{worst_w} \\ X_{worst_w} + k(X_{median} - X_{worst_w}), & \text{其他,} \end{cases} \quad (9)$$

其中  $w=1, 2, \dots, W$ 。

### 2.3 自适应搜索半径

原始鹈鹕优化算法在区域捕猎行为阶段中,使用  $R\left(1 - \frac{t}{T}\right)$  控制局部搜索的范围。HSPOA 采用式(10)

替代  $R\left(1 - \frac{t}{T}\right)$ ,扩大鹈鹕优化算法区域捕猎行为阶段的搜索范围,

$$L = 2(e^{-\frac{t}{T}} - e^{-1}). \quad (10)$$

图 1 比较了 HSPOA、POA 在区域捕猎行为阶段中搜索范围与迭代次数的关系,迭代前期 HSPOA 的搜索范围更大,且随着迭代进行,  $L$  也会逐渐降低至 0。

引入式(10)后,将式(4)替换为

$$X_{i,j}^{S_2} = X_{i,j} + L(2k-1)X_{i,j}. \quad (11)$$

若式(11)产生的备选解  $X_i^{S_2}$  的第  $j$  维超出解空间上下界,则使用式(1)进行修正。

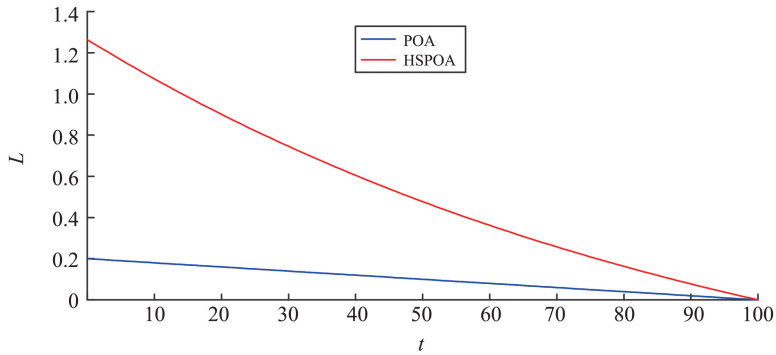


图 1 搜索范围比较  
Fig.1 Comparison results for the search scope

### 2.4 HSPOA 步骤

- (1) 初始化算法参数:设置算法最大迭代次数  $T$ 、备选解总数  $N$ 、待优化问题维数  $D$ 、 $W$  和  $G$  值。使用式 (1) 产生初代备选解并计算对应的目标函数值,准备开始迭代寻优。
- (2) 第  $t$  次迭代时,对全体备选解使用式 (6)、(7) 计算猎物  $P$  的位置。对各备选解使用式 (2)、(3) 完成位置移动行为阶段的位置更新。
- (3) 使用式 (11)、(5) 完成区域捕猎行为阶段的位置更新。
- (4) 选取种群内目标函数值较差的  $W$  个备选解位置,使用式 (8)、式 (9) 进行随机搜索。
- (5) 第  $t$  次迭代完成,更新历次迭代寻找到的最优解位置,判断  $t$  是否达到最大迭代次数  $T$ 。若达到则输出最优解位置及对应的目标函数值,否则返回步骤 2。

HSPOA 的流程图见图 2。

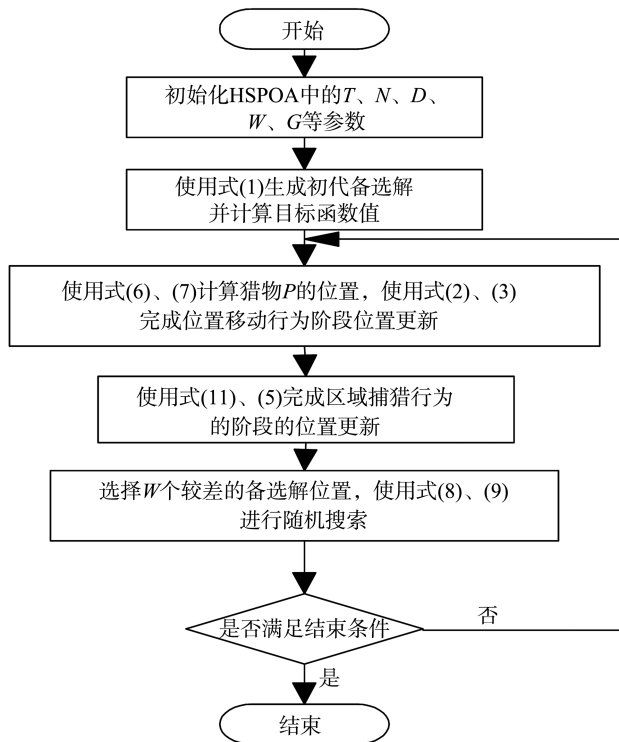


图 2 HSPOA 流程图  
Fig.2 Flow chart of HSPOA

## 3 测试函数优化结果与分析

为了验证 HSPOA 性能,选取 18 个最小值均为 0 的测试函数进行对比实验,测试函数如表 1 所示。本文

选择 AF 优化算法、生物地理优化算法 (biogeography-based optimization, BBO)、文化算法 (cultural algorithm, CA)、白骨顶鸡优化算法 (coot optimization algorithm) 和原始 POA 作为对比算法。表 1 中的测试函数  $f_1$ — $f_8$  为单峰测试函数, 用于检测优化算法的寻优精度,  $f_9$ — $f_{18}$  为多峰测试函数, 用于检测优化算法跳出局部最优的能力。从寻优精度、收敛速度和稳定性等方面分析这 18 个测试函数的寻优结果。

表 1 18 个测试函数  
Table 1 18 test functions

函数式	维数	取值范围	单峰/多峰
$f_1(x) = \max( x_i )$	30	$[-100, 100]$	单峰
$f_2(x) = \sum_{i=1}^n (x_i)^2$	30	$[-100, 100]$	单峰
$f_3(x) = \sum_{i=1}^n (ix_i)^2$	30	$[-10, 10]$	单峰
$f_4(x) = 0.26(x_1^2 + x_2^2) - 0.48x_1x_2$	2	$[-10, 10]$	单峰
$f_5(x) = \sum_{i=1}^n x_i^2 + \left(\sum_{i=1}^n 0.5ix_i\right)^2 + \left(\sum_{i=1}^n 0.5ix_i\right)^4$	10	$[-5, 10]$	单峰
$f_6(x) = \sum_{i=1}^{n/k} (x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2 + (x_{4i-2} - x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^4$	24	$[-4, 5]$	单峰
$f_7(x) = \sum_{i=1}^n  x_i  + \prod_{i=1}^n  x_i $	30	$[-10, 10]$	单峰
$f_8(x) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j\right)^2$	30	$[-10, 10]$	单峰
$f_9(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7$	2	$[-100, 100]$	多峰
$f_{10}(x) = \sum_{i=1}^n [x_i^2 - 10\cos(2\pi x_i) + 10]$	30	$[-5.12, 5.12]$	多峰
$f_{11}(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1)(4\pi x_2) + 0.3$	2	$[-100, 100]$	多峰
$f_{12}(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1 + 4\pi x_2) + 0.3$	2	$[-100, 100]$	多峰
$f_{13}(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	$[-600, 600]$	多峰
$f_{14}(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2}\right) + 20 - \exp\left(\frac{1}{n}\sum_{i=1}^n \cos(2\pi x_i)\right) + e$	30	$[-32, 32]$	多峰
$f_{15}(x) = x_1^2 + x_2^2 + 25(\sin^2(x_1) + \sin^2(x_2))$	2	$[-5, 5]$	多峰
$f_{16}(x) = 2x_1^2 - 1.05x_1^4 + \frac{x_1^6}{6} + x_1x_2 + x_2^2$	2	$[-5, 5]$	多峰
$f_{17}(x) = \sum_{i=1}^n  x_i \sin(x_i) + 0.1x_i $	30	$[-10, 10]$	多峰
$f_{18}(x) = \sum_{i=1}^n k_i  x_i ^i$ , 其中 $k_i \in (0, 1)$	30	$[-5, 5]$	多峰

为了统一测试条件, 设置所有算法备选解个数  $N=30$ ,  $T=100$ 。AF 算法中参数  $V=10$ ,  $T=20$ ,  $S=3$ ,  $\delta=0.2$ ; BBO 算法中参数  $K=0.2$ ,  $\alpha=0.9$ ,  $M=0.1$ ; CA 算法中  $A=0.35$ ,  $\alpha=0.3$ ,  $\beta=0.5$ , HSPOA 式(8)中  $W=3$ ,  $G=3$ 。将各算法单独运行 30 次获得的寻优结果的平均值、最优值、最差值和标准差分别记录于表 2—5 中。其中, 平均值、最优值和最差值描述了算法的精度, 值越小说明算法的寻优精度越高; 标准差反映算法的稳定性, 值越小则说明算法寻优的稳定性越高。

观察表 2—5 所记录数据, HSPOA 在平均值和标准差的结果上, 除在函数  $f_9$ 、 $f_{11}$  和  $f_{12}$  上与 POA 接近外, 其余均为最优。这说明 HSPOA 采用混合策略后, 能够在保证寻优精度提高的同时, 有效提升算法寻优的稳定性。

表 2 18 个测试函数的寻优结果的平均值  
Table 2 Average value of the optimization results for 18 test functions

函数	AF	BBO	CA	COOT	POA	HSPOA
$f_1$	32.610 9	22.105 1	27.458 8	0.012 1	$3.853 7 \times 10^{-9}$	$2.411 1 \times 10^{-40}$
$f_2$	$2.520 1 \times 10^3$	19.201 4	$2.443 2 \times 10^3$	$9.730 7 \times 10^{-4}$	$6.783 6 \times 10^{-16}$	$4.772 2 \times 10^{-79}$
$f_3$	40.687 8	3.610 5	362.076 0	$5.626 4 \times 10^{-4}$	$3.801 0 \times 10^{-17}$	$2.608 8 \times 10^{-79}$
$f_4$	$1.216 9 \times 10^{-6}$	$6.4161 \times 10^{-4}$	$3.885 2 \times 10^{-5}$	$3.862 0 \times 10^{-14}$	$9.092 4 \times 10^{-31}$	$1.775 4 \times 10^{-95}$
$f_5$	0.150 8	2.686 9	2.406 6	$7.324 4 \times 10^{-6}$	$1.443 0 \times 10^{-19}$	$2.632 0 \times 10^{-80}$
$f_6$	6.768 9	5.999 6	83.336 6	$5.824 8 \times 10^{-6}$	$1.273 1 \times 10^{-18}$	$6.602 6 \times 10^{-80}$
$f_7$	9.088 6	1.573 1	58.213 6	0.003 2	$1.374 8 \times 10^{-8}$	$1.880 7 \times 10^{-40}$
$f_8$	9.757 7	115.834 5	182.646 7	$3.358 9 \times 10^{-5}$	$3.183 0 \times 10^{-17}$	$1.113 5 \times 10^{-78}$
$f_9$	0.001 4	$3.523 9 \times 10^{-4}$	0.418 4	$4.110 2 \times 10^{-11}$	0	0
$f_{10}$	101.587 6	57.955 9	283.662 3	10.520 9	$2.279 7 \times 10^{-14}$	0
$f_{11}$	0.001 4	0.072 8	0.274 7	$1.179 8 \times 10^{-10}$	0	0
$f_{12}$	$6.769 0 \times 10^{-4}$	0.031 2	0.177 9	$2.575 4 \times 10^{-10}$	0	0
$f_{13}$	474.255 9	1.167 8	22.793 1	0.014 2	$9.547 9 \times 10^{-16}$	0
$f_{14}$	4.966 5	2.216 5	10.567 5	0.003 3	$1.615 6 \times 10^{-9}$	$1.746 8 \times 10^{-15}$
$f_{15}$	0.001 2	$6.620 2 \times 10^{-8}$	0.002 4	$1.073 6 \times 10^{-14}$	$1.404 1 \times 10^{-25}$	$1.016 4 \times 10^{-95}$
$f_{16}$	$1.239 3 \times 10^{-4}$	0.049 8	$1.653 8 \times 10^{-4}$	$4.351 5 \times 10^{-15}$	$3.097 8 \times 10^{-27}$	$8.391 0 \times 10^{-98}$
$f_{17}$	5.472 6	0.806 1	36.250 6	0.141 3	$7.019 7 \times 10^{-10}$	$2.154 5 \times 10^{-41}$
$f_{18}$	0.050 3	53.066 5	$4.248 3 \times 10^3$	$1.008 4 \times 10^{-4}$	$1.059 2 \times 10^{-9}$	$1.223 3 \times 10^{-18}$

表 3 18 个测试函数的寻优结果的最优值  
Table 3 Optimal value of the optimization results for 18 test functions

函数	AF	BBO	CA	COOT	POA	HSPOA
$f_1$	13.567 9	12.719 5	21.835 7	$9.304 7 \times 10^{-6}$	$8.718 9 \times 10^{-12}$	$4.368 7 \times 10^{-42}$
$f_2$	47.709 6	12.999 1	$1.691 2 \times 10^3$	$5.480 6 \times 10^{-14}$	$2.508 3 \times 10^{-21}$	$9.571 3 \times 10^{-83}$
$f_3$	19.153 8	2.179 4	232.478 1	$3.278 4 \times 10^{-10}$	$1.184 3 \times 10^{-23}$	$4.962 2 \times 10^{-84}$
$f_4$	$1.256 5 \times 10^{-9}$	$3.067 3 \times 10^{-8}$	$3.108 4 \times 10^{-6}$	$7.968 3 \times 10^{-20}$	$9.964 2 \times 10^{-40}$	$1.298 8 \times 10^{-101}$
$f_5$	0.036 1	0.074 8	1.082 8	$5.468 7 \times 10^{-17}$	$1.280 3 \times 10^{-26}$	$3.178 1 \times 10^{-87}$
$f_6$	2.951 3	0.493 3	52.127 3	$4.296 7 \times 10^{-13}$	$3.694 5 \times 10^{-25}$	$1.002 4 \times 10^{-87}$
$f_7$	5.813 2	1.102 6	20.543 0	$5.899 4 \times 10^{-7}$	$2.087 3 \times 10^{-12}$	$2.205 2 \times 10^{-42}$
$f_8$	4.637 2	37.612 5	105.659 5	$1.023 4 \times 10^{-13}$	$1.017 4 \times 10^{-25}$	$8.569 4 \times 10^{-85}$
$f_9$	$1.442 8 \times 10^{-4}$	$6.883 4 \times 10^{-15}$	0.002 1	0	0	0
$f_{10}$	58.986 7	31.121 1	230.919 6	$1.017 7 \times 10^{-8}$	0	0
$f_{11}$	$7.769 0 \times 10^{-5}$	$1.585 3 \times 10^{-11}$	0.051 3	0	0	0
$f_{12}$	$6.952 5 \times 10^{-5}$	$1.270 5 \times 10^{-6}$	0.025 1	0	0	0
$f_{13}$	170.723 5	1.099 5	15.814 2	$3.151 6 \times 10^{-10}$	0	0
$f_{14}$	3.696 4	1.603 2	8.785 0	$3.089 1 \times 10^{-6}$	$3.094 9 \times 10^{-12}$	$4.440 9 \times 10^{-16}$
$f_{15}$	$7.050 3 \times 10^{-5}$	$4.380 1 \times 10^{-14}$	$4.097 4 \times 10^{-7}$	$4.426 2 \times 10^{-23}$	$8.355 2 \times 10^{-35}$	$5.382 5 \times 10^{-102}$
$f_{16}$	$1.750 8 \times 10^{-6}$	$4.106 7 \times 10^{-16}$	$5.219 1 \times 10^{-6}$	$2.209 4 \times 10^{-24}$	$4.315 2 \times 10^{-36}$	$3.442 0 \times 10^{-102}$
$f_{17}$	1.635 9	0.291 0	19.809 5	$3.783 7 \times 10^{-6}$	$1.817 8 \times 10^{-13}$	$9.436 7 \times 10^{-44}$
$f_{18}$	0.001 7	0.005 5	3.188 3	$6.003 7 \times 10^{-9}$	$5.105 6 \times 10^{-13}$	$7.185 7 \times 10^{-31}$

表 4 18 个测试函数的寻优结果的最差值  
Table 4 Worst value of the optimization results for 18 test functions

函数	AF	BBO	CA	COOT	POA	HSPOA
$f_1$	50.863 7	31.384 0	51.543 8	0.160 1	$3.262 3 \times 10^{-8}$	$1.335 7 \times 10^{-39}$
$f_2$	$1.262 3 \times 10^4$	29.077 6	$3.235 4 \times 10^3$	0.020 1	$1.665 4 \times 10^{-14}$	$3.616 6 \times 10^{-78}$
$f_3$	70.054 9	7.297 2	484.587 8	0.016 5	$4.784 4 \times 10^{-16}$	$6.611 9 \times 10^{-78}$
$f_4$	$4.229 0 \times 10^{-6}$	0.004 6	$1.114 7 \times 10^{-4}$	$5.487 0 \times 10^{-13}$	$2.607 8 \times 10^{-29}$	$3.934 4 \times 10^{-94}$
$f_5$	0.322 8	11.262 2	3.823 8	$2.013 5 \times 10^{-4}$	$1.552 5 \times 10^{-18}$	$6.408 3 \times 10^{-79}$
$f_6$	15.537 3	21.676 8	230.595 0	$6.717 4 \times 10^{-5}$	$2.270 2 \times 10^{-17}$	$1.116 7 \times 10^{-78}$

表 4(续)

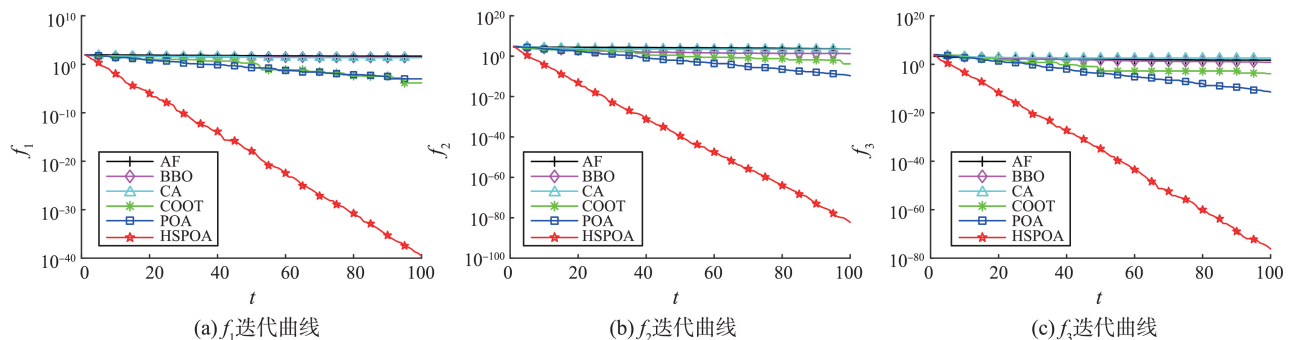
函数	AF	BBO	CA	COOT	POA	HSPOA
$f_7$	13.049 2	4.784 9	355.943 4	0.045 1	$3.250 5 \times 10^{-7}$	$1.908 7 \times 10^{-39}$
$f_8$	18.369 4	224.480 0	286.167 7	$9.508 5 \times 10^{-4}$	$7.102 4 \times 10^{-16}$	$2.416 0 \times 10^{-77}$
$f_9$	0.005 6	0.004 9	0.785 9	$6.100 5 \times 10^{-10}$	0	0
$f_{10}$	136.740 4	103.041 0	385.453 9	165.988 8	$4.707 3 \times 10^{-13}$	0
$f_{11}$	0.006 9	0.218 3	0.565 0	$3.245 6 \times 10^{-9}$	0	0
$f_{12}$	0.001 9	0.146 6	0.405 9	$4.584 5 \times 10^{-9}$	0	0
$f_{13}$	596.342 9	1.231 2	26.709 0	0.269 6	$1.754 2 \times 10^{-14}$	0
$f_{14}$	7.198 9	2.986 6	11.668 3	0.052 5	$1.539 5 \times 10^{-8}$	$3.996 8 \times 10^{-15}$
$f_{15}$	0.005 6	$1.2174 \times 10^{-6}$	0.005 8	$1.469 8 \times 10^{-13}$	$3.625 0 \times 10^{-24}$	$1.569 3 \times 10^{-94}$
$f_{16}$	$6.572 2 \times 10^{-4}$	0.298 6	$5.609 9 \times 10^{-4}$	$1.177 3 \times 10^{-13}$	$9.285 7 \times 10^{-26}$	$1.884 2 \times 10^{-96}$
$f_{17}$	11.455 6	2.134 3	53.174 3	1.459 6	$1.365 5 \times 10^{-8}$	$6.118 8 \times 10^{-41}$
$f_{18}$	0.405 0	993.977 5	$1.205 7 \times 10^5$	$5.972 4 \times 10^{-4}$	$1.117 2 \times 10^{-8}$	$3.656 4 \times 10^{-17}$

表 5 18 个测试函数的寻优结果的标准差

Table 5 The standard deviation of the optimization results for 18 test functions

函数	AF	BBO	CA	COOT	POA	HSPOA
$f_1$	12.904 5	4.342 3	5.940 7	0.033 5	$7.324 3 \times 10^{-9}$	$3.514 8 \times 10^{-40}$
$f_2$	$3.294 1 \times 10^3$	3.463 7	445.460 0	0.003 7	$3.032 0 \times 10^{-15}$	$9.478 6 \times 10^{-79}$
$f_3$	12.403 0	1.236 4	53.749 2	0.003 0	$1.124 4 \times 10^{-16}$	$1.205 5 \times 10^{-78}$
$f_4$	$1.114 2 \times 10^{-6}$	0.001 2	$3.523 9 \times 10^{-5}$	$1.272 2 \times 10^{-13}$	$4.754 9 \times 10^{-30}$	$7.1487 \times 10^{-95}$
$f_5$	0.067 3	2.937 7	0.728 5	$3.668 9 \times 10^{-5}$	$3.961 3 \times 10^{-19}$	$1.185 3 \times 10^{-79}$
$f_6$	2.785 0	5.633 8	34.464 2	$1.530 3 \times 10^{-5}$	$4.339 8 \times 10^{-18}$	$2.350 3 \times 10^{-79}$
$f_7$	1.872 5	0.684 3	72.753 3	0.008 2	$5.912 9 \times 10^{-8}$	$3.695 5 \times 10^{-40}$
$f_8$	3.367 8	42.719 9	49.645 9	$1.733 3 \times 10^{-4}$	$1.293 9 \times 10^{-16}$	$4.649 5 \times 10^{-78}$
$f_9$	0.001 3	0.001 2	0.224 1	$1.243 0 \times 10^{-10}$	0	0
$f_{10}$	19.532 4	15.420 8	33.778 1	34.489 8	$8.773 0 \times 10^{-14}$	0
$f_{11}$	0.001 5	0.104 7	0.138 4	$5.918 5 \times 10^{-10}$	0	0
$f_{12}$	$4.730 9 \times 10^{-4}$	0.040 2	0.094 7	$8.734 7 \times 10^{-10}$	0	0
$f_{13}$	116.662 1	0.033 7	3.095 9	0.054 9	$3.372 7 \times 10^{-15}$	0
$f_{14}$	0.899 3	0.299 2	0.546 4	0.009 8	$3.423 4 \times 10^{-9}$	$1.741 3 \times 10^{-15}$
$f_{15}$	0.001 1	$2.294 9 \times 10^{-7}$	0.001 9	$3.352 8 \times 10^{-14}$	$6.622 8 \times 10^{-25}$	$3.200 3 \times 10^{-95}$
$f_{16}$	$1.196 9 \times 10^{-4}$	0.113 2	$1.386 7 \times 10^{-4}$	$2.146 7 \times 10^{-14}$	$1.695 3 \times 10^{-26}$	$3.443 3 \times 10^{-97}$
$f_{17}$	2.233 4	0.416 1	9.490 7	0.294 3	$2.517 3 \times 10^{-9}$	$2.092 8 \times 10^{-41}$
$f_{18}$	0.079 5	181.563 7	$2.197 5 \times 10^4$	$1.612 4 \times 10^{-4}$	$2.495 9 \times 10^{-9}$	$6.674 9 \times 10^{-18}$

为了更加直观地对比 HSPOA 与其他算法在收敛速度与寻优精度方面的优势,图 3 给出了 6 种优化算法优化在 18 个测试函数上的寻优迭代曲线。横轴为迭代次数,纵轴为算法历次迭代搜寻到的最优目标函数值。相较于其他对比算法,HSPOA 目标函数值曲线随着迭代次数的增加迅速下降,说明寻优性能得到较大提升。虽然在表 2—5 中函数  $f_9$ 、 $f_{11}$  和  $f_{12}$  的 POA 与 HSPOA 寻优结果一致,但是在图 3(i)、(k) 和 (l) 中,HSPOA 的迭代曲线收敛速度明显快于 POA,说明 HSPOA 的寻优效率更高,能更快找到最优解。



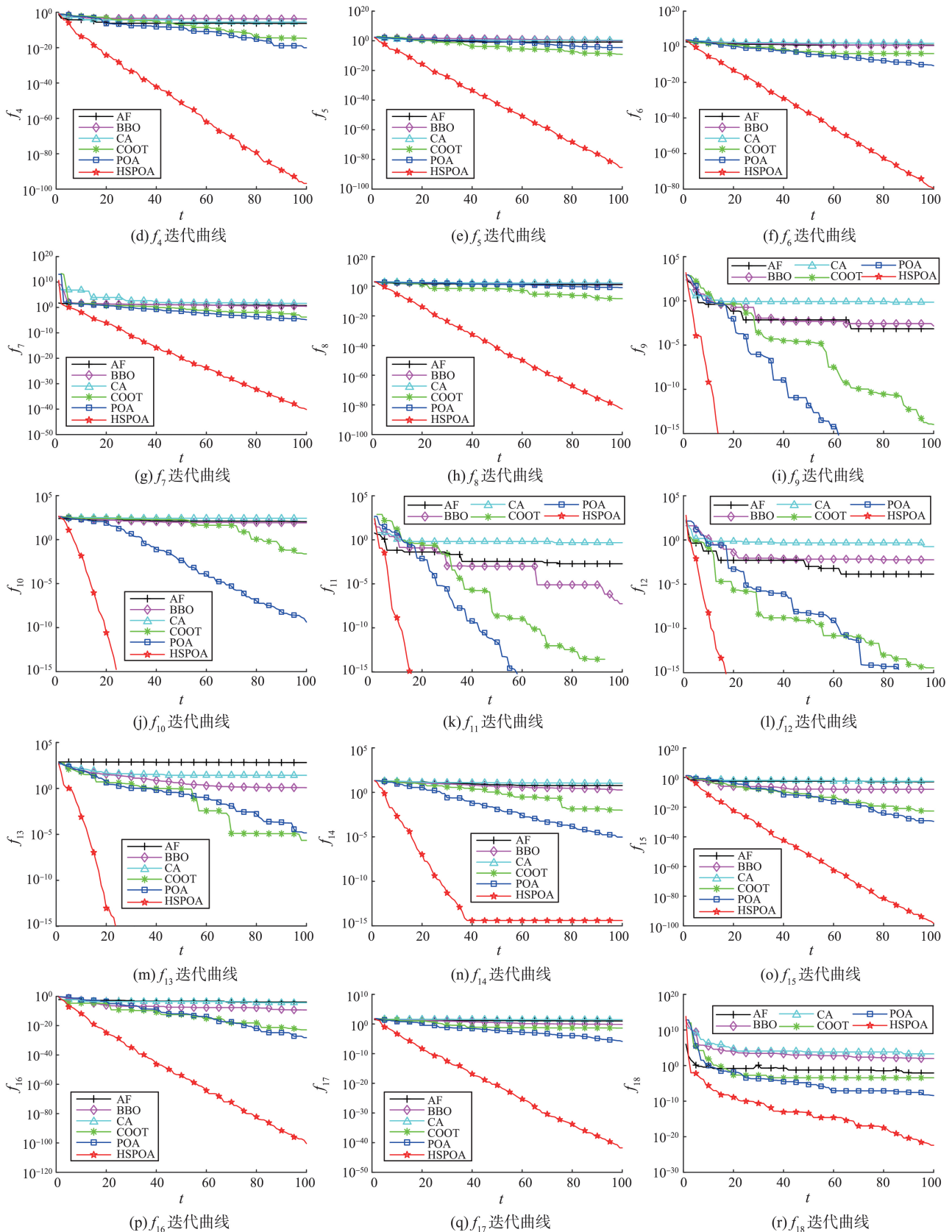


图3 18个测试函数的寻优迭代曲线

Fig.3 Optimization iterative curves for 18 test functions

利用 Wilcoxon 秩和检验分析 HSPOA 与其他对比算法在 30 次寻优结果上是否存在显著差异。若  $p < 0.05$ ,则表示两组数据的分布存在显著差异,即 HSPOA 性能提升显著;若  $p > 0.05$ ,则两组数据没有表现出显著差异,即 HSPOA 优化结果与其他算法相比提升不明显;若两种对比算法优化结果接近,可能会导致检验

结果无法计算。18个测试函数的 Wilcoxon 秩和检验结果记入表6。表中 HSPOA 优化结果的秩和检验在大多数测试函数上要优于对比算法,其中函数  $f_9$ 、 $f_{11}$  和  $f_{12}$  的30次寻优结果与 POA 接近,但结合图3迭代寻优曲线,HSPOA 的寻优效率更高,在能够在相同迭代次数下更快地发现最优解。函数  $f_{10}$  和  $f_{13}$  优化结果的差异程度与 POA 相比具有一定显著性,但弱于其他测试函数优化结果,结合表2和表5中  $f_{10}$  和  $f_{13}$  寻优结果平均值和标准差分析可得,HSPOA 的寻优精度和稳定性更好。

表6 18个测试函数的 Wilcoxon 秩和检验结果  
Table 6 Results of Wilcoxon rank sum test for 18 test functions

函数	AF	BBO	CA	COOT	POA
$f_1$	$3.019 9 \times 10^{-11}$	$3.019 9 \times 10^{-11}$	$3.019 9 \times 10^{-11}$	$3.019 9 \times 10^{-11}$	$3.019 9 \times 10^{-11}$
$f_2$	$3.019 9 \times 10^{-11}$	$3.019 9 \times 10^{-11}$	$3.019 9 \times 10^{-11}$	$3.019 9 \times 10^{-11}$	$3.019 9 \times 10^{-11}$
$f_3$	$3.019 9 \times 10^{-11}$	$3.019 9 \times 10^{-11}$	$3.019 9 \times 10^{-11}$	$3.019 9 \times 10^{-11}$	$3.019 9 \times 10^{-11}$
$f_4$	$3.019 9 \times 10^{-11}$	$3.019 9 \times 10^{-11}$	$3.019 9 \times 10^{-11}$	$3.019 9 \times 10^{-11}$	$3.019 9 \times 10^{-11}$
$f_5$	$3.019 9 \times 10^{-11}$	$3.019 9 \times 10^{-11}$	$3.019 9 \times 10^{-11}$	$3.019 9 \times 10^{-11}$	$3.019 9 \times 10^{-11}$
$f_6$	$3.019 9 \times 10^{-11}$	$3.019 9 \times 10^{-11}$	$3.019 9 \times 10^{-11}$	$3.019 9 \times 10^{-11}$	$3.019 9 \times 10^{-11}$
$f_7$	$3.019 9 \times 10^{-11}$	$3.019 9 \times 10^{-11}$	$3.019 9 \times 10^{-11}$	$3.019 9 \times 10^{-11}$	$3.019 9 \times 10^{-11}$
$f_8$	$3.019 9 \times 10^{-11}$	$3.019 9 \times 10^{-11}$	$3.019 9 \times 10^{-11}$	$3.019 9 \times 10^{-11}$	$3.019 9 \times 10^{-11}$
$f_9$	$1.211 8 \times 10^{-12}$	$1.211 8 \times 10^{-12}$	$1.211 8 \times 10^{-12}$	$1.701 6 \times 10^{-8}$	—
$f_{10}$	$1.211 8 \times 10^{-12}$	$1.211 8 \times 10^{-12}$	$1.211 8 \times 10^{-12}$	$1.211 8 \times 10^{-12}$	0.021 577
$f_{11}$	$1.211 8 \times 10^{-12}$	$1.211 8 \times 10^{-12}$	$1.211 8 \times 10^{-12}$	$5.772 0 \times 10^{-11}$	—
$f_{12}$	$1.211 8 \times 10^{-12}$	$1.211 8 \times 10^{-12}$	$1.211 8 \times 10^{-12}$	$1.656 0 \times 10^{-11}$	—
$f_{13}$	$1.211 8 \times 10^{-12}$	$1.211 8 \times 10^{-12}$	$1.211 8 \times 10^{-12}$	$1.211 8 \times 10^{-12}$	0.005 584
$f_{14}$	$1.245 5 \times 10^{-11}$	$1.245 5 \times 10^{-11}$	$1.245 5 \times 10^{-11}$	$1.245 5 \times 10^{-11}$	$1.245 5 \times 10^{-11}$
$f_{15}$	$3.019 9 \times 10^{-11}$	$3.019 9 \times 10^{-11}$	$3.019 9 \times 10^{-11}$	$3.019 9 \times 10^{-11}$	$3.019 9 \times 10^{-11}$
$f_{16}$	$3.019 9 \times 10^{-11}$	$3.019 9 \times 10^{-11}$	$3.019 9 \times 10^{-11}$	$3.019 9 \times 10^{-11}$	$3.019 9 \times 10^{-11}$
$f_{17}$	$3.019 9 \times 10^{-11}$	$3.019 9 \times 10^{-11}$	$3.019 9 \times 10^{-11}$	$3.019 9 \times 10^{-11}$	$3.019 9 \times 10^{-11}$
$f_{18}$	$3.019 9 \times 10^{-11}$	$3.019 9 \times 10^{-11}$	$3.019 9 \times 10^{-11}$	$3.019 9 \times 10^{-11}$	$3.019 9 \times 10^{-11}$

注:“—”表示该项无法计算。

表2—6与图3表明 HSPOA 优于对比算法。HSPOA 由偏好权重策略生成猎物位置并引导备选解移动,提升了备选解位置更新的多样性,增强了算法的全局寻优性能。而随着迭代,自适应减小的搜索半径策略扩大了 HSPOA 的搜索范围,在一定程度上平衡了全局寻优与局部寻优。HSPOA 采用的随机搜索策略,使得算法备选解能够有效跳出局部最优限制,有更多机会找到最优解。综上分析,HSPOA 引入混合策略后,能够在扩大鹈鹕算法备选解搜索范围的同时更加有效地进行寻优,进而提高算法的寻优精度。

## 4 结论

本文在原始鹈鹕优化算法基础上进行改进,提出了基于混合策略的鹈鹕优化算法。改进算法使用偏好权重计算位置移动行为阶段的猎物位置,以此引导备选解移动路径的同时,提高备选解移动方向多样性。采用随机搜索策略更新寻优较差的备选解,帮助算法跳出局部最优。采用搜索范围更大的自适应搜索半径策略,使算法在区域捕猎行阶段能更加广泛地探索解空间。对18个测试函数进行对照实验,通过分析实验结果验证了基于混合策略的鹈鹕优化算法的有效性。

参考文献:

- [1] ZHANG Xinming, WANG Doudou, CHEN Haiyan. Improved biogeography-based optimization algorithm and its application to clustering optimization and medical image segmentation[J]. IEEE Access, 2019, 7:28810-28825.
- [2] WANG Zhaoxia, PEN Haibo, YANG Ting, et al. Structure-priority image restoration through genetic algorithm optimization [J]. IEEE Access, 2020, 8:90698-90708.
- [3] XIONG Xin, HU Xi, GUO Huan. A hybrid optimized grey seasonal variation index model improved by whale optimization algorithm for forecasting the residential electricity consumption[J]. Energy, 2021, 234:121127.
- [4] GUERRAICHE K, DEKHICI L, CHATELET E, et al. Multi-objective electrical power system design optimization using a

- modified bat algorithm[J]. *Energies*, 2021, 14:3956.
- [5] 张金珂,张建功. 基于改进粒子群优化算法的信号检测及故障诊断[J]. *山东大学学报(理学版)*, 2023, 58(5):63-75, 83.  
ZHANG Jinke, ZHANG Jiangang. Signal detection and fault diagnosis based on improved particle swarm optimization algorithm[J]. *Journal of Shandong University (Natural Science)*, 2023, 58(5):63-75, 83.
- [6] 行鸿彦,韩杰,刘刚. 混沌变步长萤火虫优化的随机共振微弱信号检测[J]. *探测与控制学报*, 2019, 41(1):64-70.  
XING Hongyan, HAN Jie, LIU Gang. Chaotic variable step glowworm swarm optimization stochastic resonance for weak signal detection[J]. *Journal of Detection & Control*, 2019, 41(1):64-70.
- [7] CHALLAB J M, MARDUKHI F. Ant colony optimization-rain optimization algorithm based on hybrid deep learning for diagnosis of lung involvement in coronavirus patients[J]. *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, 2023, 47:887-902.
- [8] RERE LMR, FANANY M I, ARYMURTHY A M. Simulated annealing algorithm for deep learning[J]. *Procedia Computer Science*, 2015, 72:137-144.
- [9] 纪延峰. 改进的萤火虫算法及其在多约束环境下的排课问题研究[D]. 南昌:江西财经大学, 2022:44-55.  
JI Yanfeng. Research on improved firefly algorithm and its course problem in multi-constraint environment[D]. Nanchang: Jiangxi University of Finance and Economics, 2022:44-55.
- [10] 刘明. 智能算法在实验教学排课中的应用[J]. *实验技术与管理*, 2021, 38(7):244-247.  
LIU Ming. Application of intelligent algorithm in experimental course arrangement[J]. *Experimental Technology and Management*, 2021, 38(7):244-247.
- [11] 李安东,刘升. 混合策略改进鲸鱼优化算法[J]. *计算机应用研究*, 2022, 39(5):1415-1421.  
LI Andong, LIU Sheng. Multi-strategy improved whale optimization algorithm[J]. *Application Research of Computers*, 2022, 39(5):1415-1421.
- [12] 李建伟,于广滨. 改进麻雀搜索算法的轮毂减速器优化设计[J]. *哈尔滨理工大学学报*, 2022, 27(5):56-63.  
LI Jianwei, YU Guangbin. Optimization design of hub reducer based on improved sparrow search algorithm[J]. *Journal of Harbin University of Science and Technology*, 2022, 27(5):56-63.
- [13] 耿召里,李目,曹淑睿,等. 基于混合反向学习策略的鲸鱼优化算法[J]. *计算机工程与科学*, 2022, 44(2):355-363.  
GENG Zhaoli, LI Mu, CAO Shurui, et al. A whale optimization algorithm based on hybrid reverse learning strategy[J]. *Computer Engineering & Science*, 2022, 44(2):355-363.
- [14] 贾鹤鸣,陈丽珍,力尚龙,等. 透镜成像反向学习的精英池侏儒猫鼬优化算法[J]. *计算机工程与应用*, 2023, 59(24):131-139.  
JIA Heming, CHEN Lizhen, LI Shanglong, et al. Optimization algorithm of elite pool dwarf mongoose based on lens imaging reverse learning[J]. *Computer Engineering and Applications*, 2023, 59(24):131-139.
- [15] 秦宏伍,王立铮,傅渝,等. 基于多策略结合的灰狼优化算法及应用[J]. *山东大学学报(理学版)*, 2024, 59(3):51-60.  
QIN Hongwu, WANG Lizheng, FU Yu, et al. Grey wolf optimization algorithm based on multi-strategy combination and its application[J]. *Journal of Shandong University (Natural Science)*, 2024, 59(3):51-60.
- [16] 周鹏,董朝轶,陈晓艳,等. 基于 Tent 混沌和透镜成像学习策略的平衡优化器算法[J]. *控制与决策*, 2023, 38(6):1569-1576.  
ZHOU Peng, DONG Chaoyi, CHEN Xiaoyan, et al. An equilibrium optimizer algorithm based on a tent chaos and lens imaging learning strategy[J]. *Control and Decision*, 2023, 38(6):1569-1576.
- [17] 闫晓斌,方洋旺,彭维仕. 基于自适应高斯变异的多目标哈里斯鹰优化算法[J]. *北京航空航天大学学报*, 2024, 50(8):2636-2645.  
YAN Xiaobin, FANG Yangwang, PENG Weishi. Multi-objective Harris Hawk optimization algorithm based on adaptive gaussian mutation[J]. *Journal of Beijing University of Aeronautics and Astronautics*, 2024, 50(8):2636-2645.
- [18] 李大海,熊文清,王振东. 融合多策略的增强海鸥优化算法[J]. *计算机应用研究*, 2023, 40(3):717-724.  
LI Dahai, XIONG Wenqing, WANG Zhendong. Enhancing seagull optimization algorithm by applying multiple strategies[J]. *Application Research of Computers*, 2023, 40(3):717-724.
- [19] 付小鹏,王勇,冯爱武. 采用混合搜索策略的阿奎拉优化算法[J]. *计算机应用研究*, 2022, 39(10):3026-3032.  
FU Xiaopeng, WANG Yong, FENG Aiwu. Aquila optimization algorithm using hybrid search strategies[J]. *Application Research of Computers*, 2022, 39(10):3026-3032.
- [20] 王筱薇,范勤勤,王维莉. 基于基因水平多样性的微种群教与学优化算法[J]. *计算机应用研究*, 2021, 38(4):1097-1101.  
WANG Xiaowei, FAN Qinqin, WANG Weili. Micro-population teaching-learning-based optimization based on gene level diversity[J]. *Application Research of Computers*, 2021, 38(4):1097-1101.
- [21] TROJOVSKÝ P, DEGHANI M. Pelican optimization algorithm: a novel nature-inspired algorithm for engineering applications[J]. *Sensors*, 2022, 22:855.