

基于阶段自适应与差分进化的 中华穿山甲优化算法

宋甜甜, 张佳晨, 郑步京, 魏亦董, 孔维宾*

盐城工学院信息工程学院, 江苏 盐城

收稿日期: 2026年4月16日; 录用日期: 2026年5月21日; 发布日期: 2026年5月29日

摘要

针对原始中华穿山甲优化算法在中后期收敛速度慢、局部开发能力不足和求解精度不高等问题, 本文提出一种融合阶段自适应搜索、差分进化增强与局部精修机制的改进中华穿山甲优化算法。该算法通过混沌对立初始化提高初始种群质量, 通过阶段自适应机制协调前期探索与后期开发, 并在中后期引入差分进化增强与局部精修策略以提高收敛速度和求解精度。本文采用平均值、标准差和Wilcoxon秩和检验进行定量分析。结果表明, 改进算法在整体性能上优于原始算法, 具有较好的数值优化能力和应用价值。

关键词

中华穿山甲优化算法, 阶段自适应搜索, 差分进化, 局部精修

Chinese Pangolin Optimizer Algorithm Based on Stage Adaptation and Differential Evolution

Tiantian Song, Jiachen Zhang, Bujing Zheng, Yidong Wei, Weibin Kong*

School of Information Engineering, Yancheng Institute of Technology, Yancheng Jiangsu

Received: April 16, 2026; accepted: May 21, 2026; published: May 29, 2026

Abstract

To address the shortcomings of the original Chinese Pangolin Optimizer in the middle and late stages, including slow convergence, insufficient local exploitation, and limited solution accuracy, an

*通讯作者。

文章引用: 宋甜甜, 张佳晨, 郑步京, 魏亦董, 孔维宾. 基于阶段自适应与差分进化的中华穿山甲优化算法[J]. 人工智能与机器人研究, 2026, 15(3): 968-977. DOI: 10.12677/airr.2026.153088

improved Chinese Pangolin Optimizer is proposed in this paper. The proposed method employs chaotic opposition-based initialization to improve the initial population quality, uses a stage-adaptive mechanism to balance early exploration and late exploitation, and introduces differential evolution enhancement together with local refinement to accelerate convergence and improve accuracy. The mean value, standard deviation, and Wilcoxon rank-sum test are used for quantitative analysis. Experimental results demonstrate that the proposed algorithm outperforms the original algorithm in overall performance and shows good potential for numerical optimization applications.

Keywords

Chinese Pangolin Optimizer, Stage-Adaptive Search, Differential Evolution, Local Refinement

Copyright © 2026 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

群智能优化算法由于具有全局搜索能力强、模型依赖性低和实现简单等特点,近年来在复杂优化问题求解中得到了广泛关注[1]。作为元启发式优化方法的重要组成部分,该类算法在图像处理、参数优化和工程建模等问题中表现出较好的适应能力[2]。例如,群智能算法已被用于PID参数优化[3]、无人艇路径规划[4]以及模糊神经网络设计等工程问题[5]。然而,随着优化问题复杂度的不断提高,如何在保持前期全局探索能力的同时提高中后期收敛速度与求解精度,仍然是群智能优化算法研究中的关键问题[6]。

中华穿山甲优化算法(Chinese Pangolin Optimizer, CPO)是一种新近提出的仿生优化方法。该算法通过模拟中华穿山甲在诱捕和捕食过程中的行为特征完成种群更新,在连续优化问题中表现出一定的搜索能力[7]。然而,从算法机理与实验表现来看,原始CPO仍然存在以下不足:前期搜索与后期开发之间的切换不够灵活;中后期对优质区域的强化能力不足;部分复杂问题的收敛精度和稳定性仍有进一步提升空间。

针对上述问题,本文提出一种改进中华穿山甲优化算法(Improved Chinese Pangolin Optimizer, ICPO)。该方法引入混沌对立初始化、阶段自适应搜索、差分进化增强、自适应种群缩减以及局部精修机制[8]。为验证所提方法的有效性,本文从单峰、多峰及固定维函数中选取了8个具有代表性的基准函数,结果表明所提ICPO在不同类型测试问题上均表现出良好的优化能力,尤其在中后期收敛效率和复杂问题求解性能方面表现出更明显的优势。

2. 穿山甲优化算法

穿山甲优化算法(Chinese Pangolin Optimizer, CPO)是一种基于中华穿山甲诱捕与捕食行为构建的群智能优化方法。其搜索过程主要由气味浓度因子、能量波动因子及Lévy随机步长共同驱动。原始气味浓度模型可表示为:

$$M(t) = \frac{Q}{\pi u \sigma_y(t) \sigma_z(t)} \exp\left(-\frac{H^2}{2\sigma_z^2(t)}\right) \quad (1)$$

其中, $M(t)$ 为第 t 次迭代时的气味浓度; Q 为气味释放强度; u 为扩散相关控制参数; $\sigma_y(t)$ 与 $\sigma_z(t)$ 分别表示在不同方向上的扩散尺度; H 为垂直扩散高度参数。该式用于刻画搜索过程中个体感知强度随扩

散条件变化的动态特征。

进一步对其归一化，可得到气味浓度因子：

$$C_m(t) = \text{rescale}(M(t)) \quad (2)$$

其中， $\text{rescale}(\cdot)$ 表示归一化操作，用于将气味浓度映射到统一数值范围，从而便于后续搜索行为控制。

与此同时，为控制搜索强度随迭代逐步减弱，原始 CPO 采用快速衰减因子：

$$C_1(t) = 2 - \frac{2t}{T_{\max}} \quad (3)$$

其中， t 为当前迭代次数， T_{\max} 为最大迭代次数。由式(3)可知，CPO 通过线性递减机制实现由前期全局探索向后期局部开发的过渡。

总体而言，CPO 具有较好的全局搜索能力，但其线性衰减搜索机制在复杂优化问题上往往难以兼顾前期充分探索与中后期高效开发，容易导致中期停滞和后期收敛精度不足。基于此，本文对其初始化方式、阶段搜索策略以及中后期开发机制进行改进。

3. 改进中华穿山甲优化算法

3.1. 混沌对立初始化策略

为了提高初始种群的分布均匀性与个体质量，本文采用 Tent 混沌映射生成初始序列：

$$x_{k+1} = \begin{cases} 2x_k, & x_k < 0.5 \\ 2(1-x_k), & x_k \geq 0.5 \end{cases} \quad (4)$$

其中， x_k 为第 k 个混沌变量， x_{k+1} 为下一时刻混沌变量。

利用混沌序列的遍历性和均匀性，可将其映射到搜索空间中，从而得到初始个体位置：

$$X_i = lb + X_i^{\text{chaos}} \cdot (ub - lb) \quad (5)$$

其中， X_i^{chaos} 表示由混沌序列生成的归一化变量， lb 和 ub 分别表示搜索空间下界与上界。该映射用于将混沌变量转换为实际搜索空间中的候选解。

在此基础上，为进一步提升初始解质量，本文引入对立学习思想，构造对立个体：

$$X_i^{\text{op}} = lb + ub - X_i \quad (6)$$

其中， X_i^{op} 为个体 X_i 的对立解。

并采用贪婪策略保留更优个体：

$$X_i^0 = \begin{cases} X_i^{\text{op}}, & f(X_i^{\text{op}}) < f(X_i) \\ X_i, & \text{otherwise} \end{cases} \quad (7)$$

其中， X_i^0 表示最终保留下来的初始化个体。若对立个体适应度更优，则保留对立个体；否则保留原个体。该贪婪选择有助于提高初始种群质量。

3.2. 阶段自适应搜索机制

为了更好地协调不同阶段的搜索行为，本文定义迭代阶段比例为：

$$\phi = \frac{t}{T_{\max}} \quad (8)$$

为了在不同阶段采用不同搜索强度，本文构造分段控制因子：

$$C_1(\phi) = \begin{cases} 2(1-\phi)^{1.20}, & \phi \leq \phi_e \\ 2(1-\phi)^{1.55}, & \phi_e < \phi \leq \phi_m \\ 2(1-\phi)^{1.90}, & \phi > \phi_m \end{cases} \quad (9)$$

其中, ϕ_e 表示前期结束阈值, ϕ_m 表示中期结束阈值。与原始线性递减因子相比, 式(9)中的分段控制方式能够在前期维持较强的搜索步长, 在中期加快向优势区域过渡, 在后期进一步提高局部开发强度。

在此基础上, 为了减少中后期冗余搜索, 本文进一步引入自适应种群缩减机制:

$$NP(t) = \begin{cases} N_0, & \phi \leq \phi_e \\ \text{round} \left[N_{\min} + (N_0 - N_{\min})(1-\phi)^{1.35} \right], & \phi > \phi_e \end{cases} \quad (10)$$

其中, N_0 为初始种群规模, N_{\min} 为最小种群规模。该机制使得算法在前期保留较大种群规模以保障搜索广度, 而在中后期逐步缩减种群规模, 从而提升收敛效率。

3.3. 差分进化增强机制

在中后期, 为提高算法的收敛速度与搜索效率, 本文对停滞个体和劣质个体引入 current-to-pbest/1 变异策略:

$$V_i = X_i + F_i(X_{pbest} - X_i) + F_i(X_{r1} - X_{r2}) \quad (11)$$

其中, V_i 为第 i 个个体对应的变异向量, X_{pbest} 为从精英个体集合中随机选取的优质个体, X_{r1} 和 X_{r2} 分别从当前种群或外部档案中随机选取, F_i 为变异因子。

随后采用二项交叉方式生成试验个体:

$$U_{i,j} = \begin{cases} V_{i,j}, & \text{rand}_j \leq CR_i \text{ or } j = j_{rand} \\ X_{i,j}, & \text{otherwise} \end{cases} \quad (12)$$

其中, CR_i 为交叉概率, rand_j 为 $[0, 1]$ 区间内均匀随机数, j_{rand} 为保证至少一维发生交叉的随机维度。通过式(11)和式(12)可见, 该机制能够在中后期为种群提供更强的定向开发能力, 从而加快收敛速度。

为避免人工设定固定参数带来的局限性, 本文基于历史成功参数均值对 F_i 和 CR_i 进行自适应采样, 其形式分别为:

$$F_i = M_F(r) + 0.10 \tan[\pi(\text{rand} - 0.5)] \quad (13)$$

$$CR_i = M_{CR}(r) + 0.10 \text{randn} \quad (14)$$

其中, $M_F(r)$ 和 $M_{CR}(r)$ 表示基于历史成功参数信息构造的参数均值, randn 表示标准正态随机数。该采样方式能够在一定程度上提高参数设置的自适应性, 从而增强算法对不同优化问题的适应能力。

3.4. 精英引导收缩与局部精修机制

为进一步提高中后期对最优区域的聚集能力, 本文构造精英引导收缩更新方式:

$$X_i^{elite} = X_i + \eta(X_{best} - X_i) + \gamma(X_{mean}^{elite} - X_i) + \sigma \cdot \text{randn} \quad (15)$$

其中, X_{best} 为当前最优个体, X_{mean}^{elite} 为精英均值, η 、 γ 和 σ 分别表示最优引导系数、精英引导系数和扰动强度。由式(15)可知, 个体在向当前最优解靠近的同时, 也受到精英群体中心的共同引导, 从而提高种群整体向高质量区域收缩的效率, 并减少单个最优个体造成的过早偏置。

在搜索后期，为进一步提高最终求解精度，本文在全局最优个体附近构造局部邻域候选解。其典型形式表示为：

$$X_1^{loc} = X_{best} + \sigma_1 (ub - lb) randn \quad (16)$$

$$X_2^{loc} = X_{best} + 0.5\sigma_1 (X_{mean}^{elite} - X_{best}) + Levy(D, 0.01) \odot (ub - lb) \quad (17)$$

$$X_3^{loc} = \frac{X_{best} + X_{mean}^{elite}}{2} + 0.2\sigma_2 (ub - lb) \odot randn \quad (18)$$

其中， X_1^{loc} 、 X_2^{loc} 、 X_3^{loc} 分别为三种局部邻域候选解， σ_1 和 σ_2 为局部扰动强度， D 为问题维数， $Levy(D, 0.01)$ 表示 D 维 Lévy 扰动向量，其中 0.01 为步长尺度参数， \odot 表示逐元素乘积。

三种局部候选解分别从随机邻域扰动、精英引导扰动和中心均值扰动三个角度增强后期精修能力，从而提高算法对最优区域的细粒度搜索能力。

3.5. 贪婪选择与边界处理

在候选解生成之后，本文采用贪婪选择策略进行更新：

$$X_i^{t+1} = \begin{cases} X_i^{new}, & f(X_i^{new}) < f(X_i^t) \\ X_i^t, & \text{otherwise} \end{cases} \quad (19)$$

其中， X_i^t 表示第 t 次迭代时第 i 个个体的位置， X_i^{new} 表示当前迭代中生成的候选解， X_i^{t+1} 表示更新后的个体位置， $f(\cdot)$ 为目标函数。该策略能够保证种群中保留的个体质量不下降，从而维持算法整体收敛过程的稳定改进。

若候选解超出边界，为保证种群始终在可行域内演化，本文采用边界投影方式进行修复，其表达式为：

$$X_{i,j} = \min(\max(X_{i,j}, lb_j), ub_j) \quad (20)$$

其中， $X_{i,j}$ 表示第 i 个个体在第 j 维上的分量， lb_j 和 ub_j 分别表示第 j 维变量的下界和上界。该处理方式能够将越界分量直接投影回可行区间，在保证解可行性的同时避免因复杂修复机制引入额外干扰。通过贪婪选择与边界投影的结合，算法能够在保证搜索有效性的基础上提高更新过程的稳定性与实现简洁性。

4. 仿真结果及分析

4.1. 基准函数

为评估所提改进中华穿山甲优化算法(ICPO)的优化性能，本文选取8个典型基准函数进行数值实验，并与中华穿山甲优化算法(CPO)进行对比分析，以验证改进策略的有效性。其中，F1、F2为单峰函数，主要用于考察算法的局部开发能力与收敛精度；F3、F4为多峰函数，主要用于检验算法的全局搜索能力及跳出局部最优的能力；F5~F8为固定维函数，用于进一步验证算法在复杂低维非线性优化问题上的适应能力。各基准函数的名称、类型、维数及变量范围见表1。

4.2. 参数设置与结果分析

实验中，种群规模设为 $N=50$ ，最大迭代次数设为 $T=500$ 。对于可扩展测试函数，其维数按照对应函数定义设置；对于固定维测试函数，其维数取各基准函数的原始定义值。每个基准函数均独立运行30次，并采用相同随机种子机制进行对照实验，以减弱随机初始化对实验结果的影响。同时，为进一步检验两种算法性能差异的统计显著性，本文采用Wilcoxon秩和检验对CPO和ICPO的独立运行结果进行显著性分析，显著性水平设为0.05。CPO与ICPO在8个基准函数上的实验结果如表2所示。

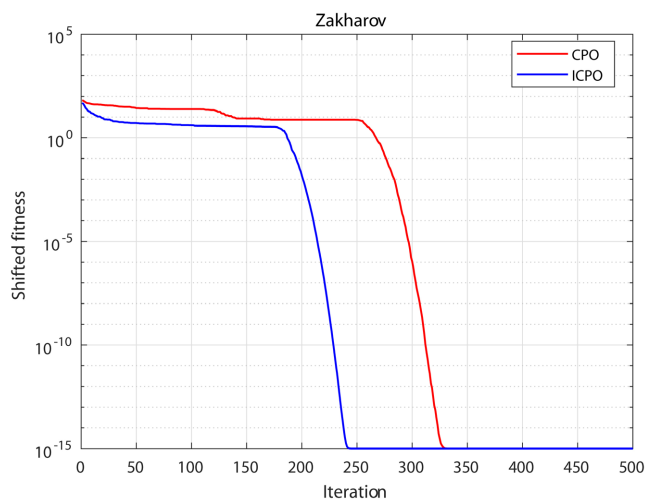
Table 1. Benchmark functions**表 1.** 基准函数

函数	名称	类型	维数	取值范围
F1	Zakharov	单峰	30	[-5, 10]
F2	Shifted Sphere	单峰	30	[-100, 100]
F3	Schwefel	多峰	30	[-500, 500]
F4	CEC-Style F7	多峰	15	[-100, 100]
F5	Shekel's Foxholes	固定维	2	[-65.54, 65.54]
F6	Goldstein-Price	固定维	2	[-2, 2]
F7	Six-Hump Camel	固定维	2	[-5, 5]
F8	Hartman 6	固定维	6	[0, 1]

Table 2. Experimental comparison results of CPO and ICPO on eight benchmark functions**表 2.** CPO 与 ICPO 在 8 个基准函数上的实验结果比较

函数	CPO 均值	CPO 标准差	ICPO 均值	ICPO 标准差	p 值	优胜算法
F1	3.6486E-114	1.9648E-113	0	0	1.1444E-12	ICPO
F2	0.2971	0.2685	1.3275E-28	4.8462E-28	2.8719E-11	ICPO
F3	4.0499E+03	1.5264E+03	3.2953E+03	1.1890E+03	0.045945	ICPO
F4	8.1388E+04	1.4915E+05	3.4270E+03	224.53	2.8719E-11	ICPO
F5	9.1144	4.5115	2.7045	2.7541	4.3499E-08	ICPO
F6	3.0002	2.8113E-04	3.0000	1.2871E-15	3.8898E-12	ICPO
F7	-1.0316	7.3831E-07	-1.0316	6.1481E-16	8.4106E-12	ICPO
F8	-3.2711	0.0665	-3.2747	0.058399	8.3504E-04	ICPO

为直观比较 CPO 与 ICPO 在不同类型优化问题上的收敛行为, 图 1~8 分别给出了两种算法在 8 个基准函数上的收敛曲线。

**Figure 1.** Convergence curves on F1**图 1.** F1 函数收敛曲线

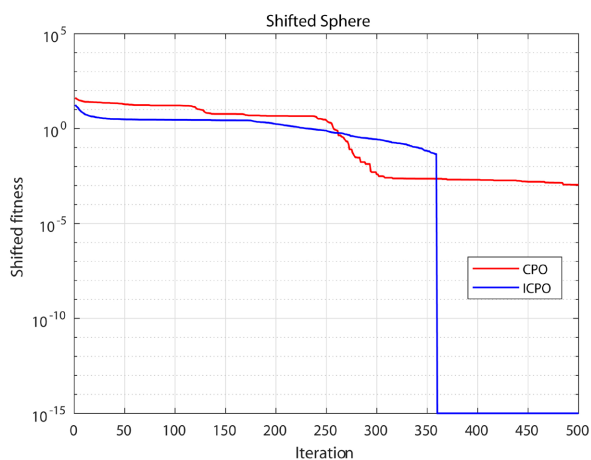


Figure 2. Convergence curves on F2
图 2. F2 函数收敛曲线

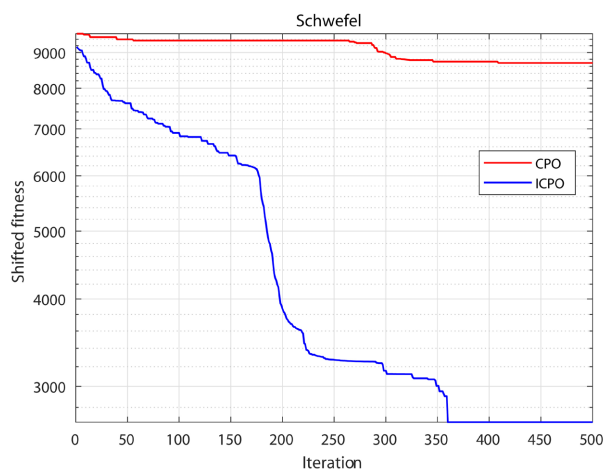


Figure 3. Convergence curves on F3
图 3. F3 函数收敛曲线

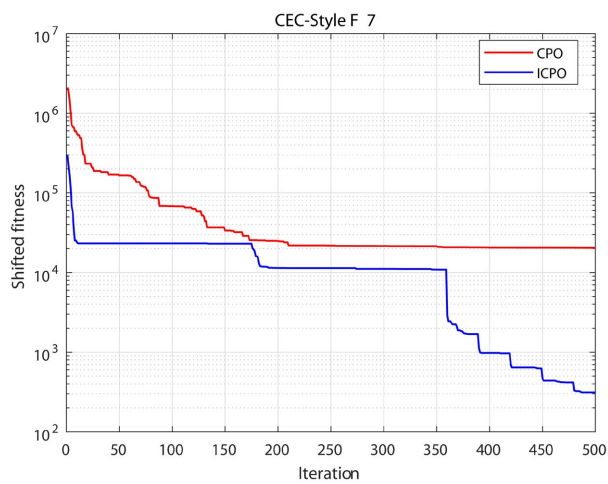


Figure 4. Convergence curves on F4
图 4. F4 函数收敛曲线

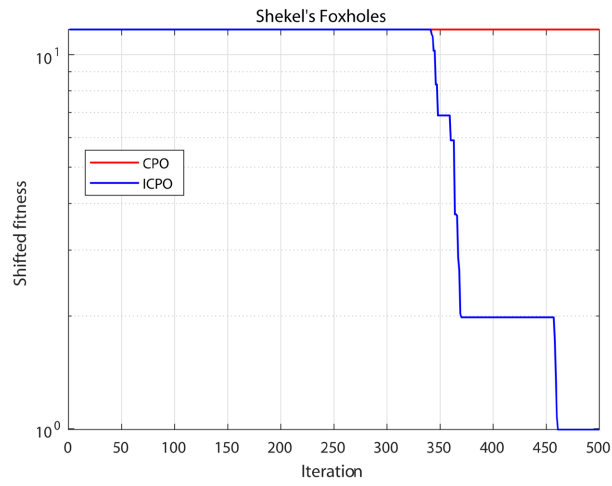


Figure 5. Convergence curves on F5
图 5. F5 函数收敛曲线

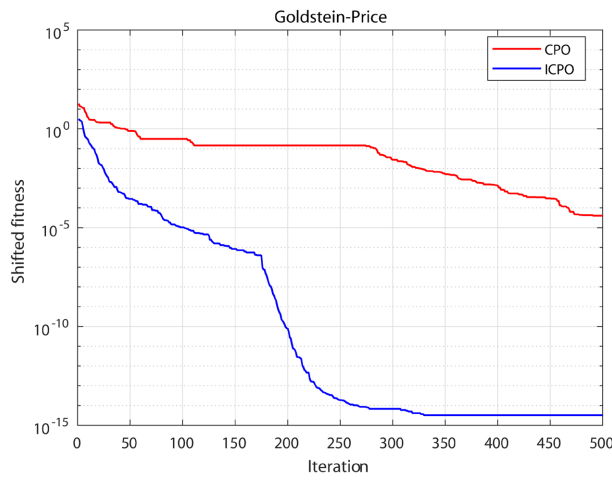


Figure 6. Convergence curves on F6
图 6. F6 函数收敛曲线

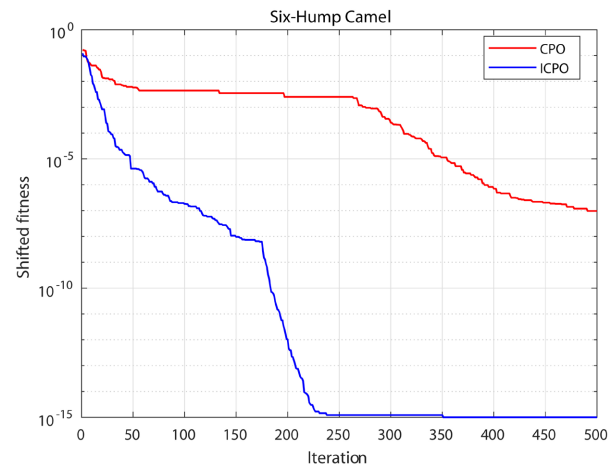


Figure 7. Convergence curves on F7
图 7. F7 函数收敛曲线

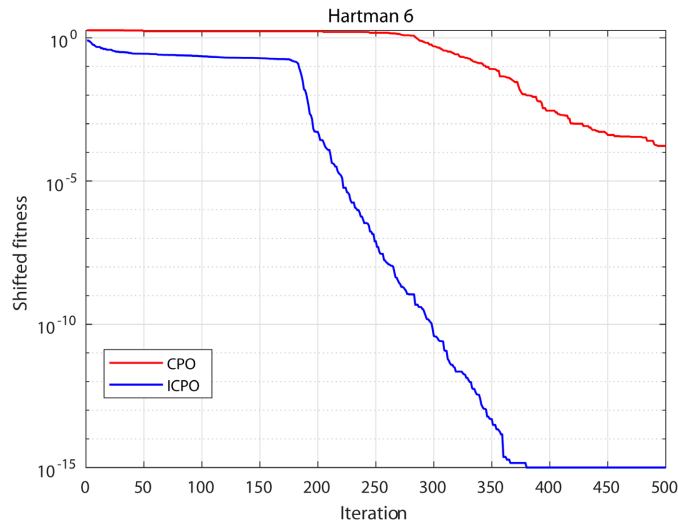


Figure 8. Convergence curves on F8
图 8. F8 函数收敛曲线

从图 1~8 可以看出，两种算法在迭代初期均能够实现较快的误差下降，CPO 的前期全局探索机制在 ICPO 中得到了有效保留。随着迭代推进，ICPO 在多数测试函数上表现出更快的收敛速度，并在较长迭代区间内保持更低的适应度水平，表明其在搜索效率、中后期开发能力及末期逼近质量方面整体优于 CPO。相比之下，CPO 在部分测试函数上表现出较为明显的收敛平台现象，导致其中后期搜索效率下降，整体收敛进程相对滞后。

进一步分析可知，ICPO 在复杂搜索环境下表现出更稳定的演化轨迹和更优的中后期优化行为，说明所引入的多策略改进机制有效缓解了 CPO 容易出现的中期停滞问题，并增强了算法在复杂非线性优化问题中的全局搜索能力与局部开发能力。结合多次独立运行结果可知，ICPO 在平均优化性能和整体收敛稳定性方面均优于 CPO。

4.3. 阶段划分阈值敏感性分析

Table 3. Sensitivity analysis results of stage division thresholds
表 3. 阶段划分阈值敏感性分析结果

ϕ_e	ϕ_m	平均秩	综合评价
0.25	0.75	3.8750	最优/默认
0.25	0.65	3.9375	较优
0.25	0.85	4.5000	较优
0.45	0.65	4.8750	中等
0.35	0.75	5.0000	中等
0.35	0.85	5.3125	中等
0.35	0.65	5.5000	一般
0.45	0.75	5.6250	一般
0.45	0.85	6.3750	一般

阶段划分阈值 ϕ_e 和 ϕ_m 是影响 ICPO 探索与开发平衡的重要参数。为分析算法对该类参数的敏感性，

本文固定其他参数不变, 设置 $\phi_e \in \{0.25, 0.35, 0.45\}$, $\phi_m \in \{0.65, 0.75, 0.85\}$, 共形成 9 组参数组合。每组参数在 8 个基准函数上独立运行 30 次, 并根据各组合在不同函数上的均值进行排序, 计算其在所有函数上的平均秩。平均秩越小, 表示该参数组合的综合性能越优。阶段划分阈值敏感性分析结果如表 3 所示。

由表 3 可知, 不同阶段阈值组合下 ICPO 的平均秩存在一定差异, 但整体变化幅度相对有限, 说明算法对 ϕ_e 和 ϕ_m 不具有强敏感性, 具有一定参数鲁棒性。其中, $\phi_e = 0.25$, $\phi_m = 0.75$ 取得最优平均秩, 表明适当缩短前期探索阶段并保留较充分的中期过渡过程, 有助于提高算法整体性能。因此, 本文采用 $\phi_e = 0.25$, $\phi_m = 0.75$ 作为 ICPO 的默认阶段划分参数。

5. 结论

本文针对中华穿山甲优化算法在中后期收敛速度较慢、局部开发能力不足以及复杂问题求解性能受限等问题, 提出了一种改进中华穿山甲优化算法(ICPO)。该算法在前期全局探索特性的基础上, 引入混沌对立初始化、阶段自适应搜索、差分进化增强以及局部精修等策略, 以增强算法在不同搜索阶段中的平衡能力、持续开发能力与整体收敛质量。为验证所提方法的有效性, 本文选取 8 个标准基准函数对 ICPO 与 CPO 进行了对比实验。实验结果表明, ICPO 在平均优化性能、收敛速度以及整体收敛稳定性方面均优于 CPO, 尤其在部分复杂测试函数上表现出更高的求解精度和更优的收敛特性, 体现出较强的综合优化能力。

基金项目

大学生创新创业训练计划资助项目(2025482, 2025496, 2025517)。

参考文献

- [1] 秦小林, 罗刚, 李文博, 等. 集群智能算法综述[J]. 无人系统技术, 2021, 4(3): 2096-5915.
- [2] 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): 3452-3467.
- [3] 莫程凯, 赵宇红, 段灵芝, 等. 两种群智能算法在 PID 参数优化中的应用分析[J]. 机械工程与自动化, 2022(1): 181-182+186.
- [4] 宫月红, 张少君, 王明雨, 等. 基于遗传-粒子群优化算 USV 路径规划方法[J]. 山东交通学院学报, 2022, 30(1): 29-34.
- [5] 王永海, 郭珂, 方岳, 等. 基于集群智能优化的模糊神经网络设计与应用[J]. 航空兵器, 2021, 28(1): 87-92.
- [6] 张梓嘉, 苏成利, 王宁, 等. 自适应正余弦搜索樽海鞘群优化算法[J]. 当代化工, 2022, 51(2): 407-412.
- [7] Guo, Z., Liu, G. and Jiang, F. (2025) Chinese Pangolin Optimizer: A Novel Bio-Inspired Metaheuristic for Solving Optimization Problems. *The Journal of Supercomputing*, **81**, Article No. 517. <https://doi.org/10.1007/s11227-025-07004-4>
- [8] 白芸. 基于强化学习的差分进化算法改进方法研究[J]. 科学技术创新, 2022(28): 49-52.