

多策略协同改进河马优化算法及应用

周建新 王 彬 焦江涛

(华北理工大学电气工程学院 唐山 063210)

摘要: 针对标准河马优化算法(HO)收敛速度慢、易陷入局部最优、搜索与开发不平衡等问题,提出了多策略协同改进的河马优化算法(MSCHO)。在初始化阶段引入 Circle 映射,增强初始化种群的多样性,减少初始化盲区;其次在 HO 位置更新阶段引入镜像-极差随机游走策略,增强寻找到最优解的可能;最后在 HO 探索阶段采用余弦自适应 t 分布变异,提升算法的收敛速度和稳定性。为验证算法的改进效果和性能,选取 5 个群智能优化算法对 8 个经典测试函数以及经典工程应用问题进行仿真实验,实验结果表明:与其他的算法相比,改进后的 MSCHO 具有更加出色的寻优能力和收敛速度。

关键词: 河马优化算法;circle 映射;镜像-极差随机游走;余弦自适应 t 分布变异

中图分类号: TP301.6;TN2 **文献标识码:** A **国家标准学科分类代码:** 520.1040

Multi-strategy collaborative improvement hippo optimization algorithm and its applications

Zhou Jianxin Wang Bin Jiao Jiangtao

(School of Electrical Engineering, North China University of Science and Technology, Tangshan 063210, China)

Abstract: Aiming at the problems that the standard hippopotamus optimization algorithm (HO) has slow convergence speed, easy to fall into local optimum and imbalance between search and development, this study proposes the multi-strategy cooperative hippopotamus optimization algorithm (MSCHO). Circle mapping is introduced in the initialization stage to enhance the diversity of the initialization population and reduce the blind area of initialization; secondly, the mirror-pole random wandering strategy is introduced in the HO position updating stage to enhance the possibility of finding the optimal solution; finally, nonlinear adaptive Cosine adaptive t-distribution variation is used in the HO exploration stage to improve the convergence speed and stability of the algorithm. In order to verify the improvement effect and performance of the algorithm, five swarm intelligent optimization algorithms are selected to simulate eight classical test functions and the experimental results show that compared with other algorithms, the improved MSCHO has more excellent optimization ability and convergence speed.

Keywords: hippo optimization algorithm; circle mapping; mirror-pole random wandering; cosine adaptive t-distribution variation

0 引言

近年来,面向复杂优化问题的群智能算法发展迅速,典型代表包括哈里斯鹰优化算法、灰狼优化算法和蜣螂优化算法等。虽然发展迅速,但在高维、约束和多峰场景下,仍常见早熟收敛、探索开发失衡、边界点聚集等问题。这些问题会导致解的稳定性与可重复性不足,在工程优化问题中表现明显。河马优化算法(hippopotamus optimization algorithm, HO)由 Amiri 等^[1]于 2024 年提出,是一种以河马群体行为为灵感、基于三阶段行为模型构建的启发式算

法。其具有一定的全局搜索能力与鲁棒性,但与多数群智能方法类似,在实际求解中仍存在收敛速度偏慢、易陷入局部最优等不足。

针对算法的缺陷,已有学者在其基础上进行改进和应用。Guo 等^[2]提出了一个利用 HO 和双深度 Q 网络来优化钢筋笼设计的框架,该框架用于优化复杂三维结构中钢筋的布局,提升自动化建造能力。研究表明,优化后的布局减少了应力集中,显著提高了钢筋笼的可建造性;Chaib 等^[3]提出了一种改进的布朗熊优化算法,将 HO 和分数阶混沌映射相结合,用于光伏(PV)模型参数的精确估

计,为光伏模型的参数提取提供了更可靠的解决方案。He 等^[4]提出了一种基于 HO 和变分模态分解 VMD 的早期轴承故障诊断方法,通过结合电流与振动信号的加权证据融合,有效提升了故障特征的提取和诊断精度,特别是在噪声干扰较强的情况下。

任庄欣等^[5]将 Piecewise 混沌映射用于初始化,结合自适应权重实现由全局到局部的步长衰减,并以透镜成像反向学习扩展搜索范围;袁磊^[6]提出:以准反向学习提升初始/中期多样性,在第 1 阶段结合正余弦算法提供持续振荡式探索,在防御与逃逸阶段分别引入切线飞行与 PID 搜索因子,实现长步长跳跃与后期自适应。整体上,这些工作在抑制早熟与提升精度方面取得了积极进展,但在调度与机理方面仍有提升空间;面对多目标或更复杂系统时,性能与稳健性仍存在改进余地,在算法的应用上也仍有探索空间。

针对上述的问题,本文提出一种多策略协同改进的河马优化算法(multi-strategy collaborative improvement hippo optimization algorithm, MSCHO)。采用基于个体极差的步长自适应,并配合随迭代递减的 t 变异强度实现从全局探索到局部精修,将各阶段策略有机衔接;在种群初始化阶段引入 Circle 混沌映射,提升初始多样性与可行域覆盖;在探索阶段融入镜像—极差自适应随机游走,以个体极差作为群体分散程度的近似表征,使扰动尺度在搜索过程中呈现由大到小的自适应趋势,平衡全局探索与局部搜索,避免早熟并稳步逼近潜优区域;在防御/逃逸阶段采用余弦自适应 t 分布变异策略,实现由重尾到轻尾的渐进精修,兼顾跳出局部与后期微调,提高算法整体的收敛效率与精度。将改进后的算法与其余算法在测试函数以及工程优化问题上进行仿真,改进后的算法在计算精度和收敛速度上具有显著的优势。相关消融实验进一步验证了各策略及其作用的有效性。

1 HO

HO 是一种受河马自然行为启发的新型元启发式优化算法。该算法主要模拟了河马的 3 种典型行为模式,依次构成了算法的 3 个阶段:第 1 阶段是河马在河流或池塘中的位置更新行为,模拟了河马群体在水域中的活动和位置变化,用于实现算法的探索过程;第 2 阶段是河马防御捕食者的行为,当河马遭遇捕食者威胁时,会迅速转向并张开大嘴发出巨大吼声以威慑捕食者,这一过程增强了算法的全局搜索能力,使算法能够有效避免陷入局部最优解;第 3 阶段是河马逃避捕食者的行为,当河马无法通过防御吓退捕食者时,会选择快速逃离至附近的水域,体现了算法的局部搜索和开发能力,有助于在局部区域内寻找更优解。

1.1 基于 Circle 混沌映射的种群初始化

在元启发式算法中,种群初始化的质量直接影响算法的全局搜索能力和收敛效率。传统 HO 每一个河马通过一个向量进行表示,其数量由矩阵进行表征,采用随机均匀分

布生成随机的初始解形成初始种群,虽操作简单,但可能导致种群多样性不足或分布不均,进而陷入局部最优。为提升 HO 初始种群质量,引入 Circle 混沌映射(又称圆形映射)重构种群初始化过程^[7]。Circle 混沌映射是一种具有遍历性、非周期性和初值敏感性的混沌系统,该映射能在 $[0,1)$ 区间生成分布均匀且不重复的混沌序列,其类随机性与内在规律性的结合,为 HO 种群初始化提供了理想的分布特性。根据混沌映射的特点, $a=0.5$ 使得系统表现出合适的混沌行为,避免进入周期性区域,保证了初始化时的多样性,而 $b=0.2$ 控制映射的周期变化,较小的 b 值可以有效避免系统产生过于频繁的周期性波动,使得混沌序列具有较好的分布特性,确保生成的混沌序列具有良好的随机性和均匀分布,从而提高优化算法的全局搜索能力^[8]。其数学表达式为:

$$c_{ij} = \text{mod}\left(c_{i-1,j} + b - \frac{a}{2\pi} \sin(2\pi c_{i-1,j})\right) \quad (1)$$

$$X_i: x_{ij} = lb + c_{ij} \times (ub - lb) \quad (2)$$

式中: c_{ij} 为混沌映射函数迭代生成的确定性序列,接着,将该序列线性映射至问题的解空间; lb 和 ub 分别为决策变量的下界和上界,将映射后的向量 $[X_1, X_2, \dots, X_N]^T$ 作为河马初始种群。相较于随机初始化,Circle 混沌映射显著提升了初始解的多样性和均匀性,为后续的河马社会行为模拟奠定了更优的搜索起点,从而增强算法全局探索能力。

1.2 基于镜像-极差随机游走的位置更新新策略

HO 模拟河马群社会结构及其行为。主导雄性个体驱逐成熟雄性个体。被驱逐者在搜索空间新区域探索,建立新解或者挑战主导者,争夺主导地位。雄性河马位置的数学表示如式(3)所示。

$$x_i^{Mhippo}: x_{ij}^{Mhippo} = x_{ij} + y_1 \times (Dhippo - I_1 \times x_{ij}) \quad (3)$$

$i = 1, 2, \dots, \frac{N}{2}, j = 1, 2, \dots, m$

式中: x_i^{Mhippo} 为第 i 头雄性河马在每个决策变量下的所有位置信息, $Dhippo$ 代表主导雄性河马,即当前迭代最优解的位置, y_1 为 0~1 的随机数, I_1 等于 1 或 2。

河马群体中幼年河马通常聚集在雌性附近,但少数会因好奇而远离,式(4)为雌性或未成熟河马的位置更新公式。

$$X_i^{FBhippo}: x_{ij}^{FBhippo} = \begin{cases} x_{ij} + h_1 \times (Dhippo - I_2 \times MG_i), & T > 0.6 \\ \Xi, & \text{其他} \end{cases}$$

$$\Xi = \begin{cases} x_{ij} + h_2 \times (MG_i - Dhippo), & r_6 > 0.5 \\ lb_j + r_7 \times (ub_j - lb_j), & \text{其他} \end{cases} \quad (4)$$

式中: $x_i^{FBhippo}$ 为第 i 头雌性或未成熟河马在每个决策变量下的所有位置信息,参数 T (随迭代变化)控制离群倾向:当 $T > 0.6$ 时,幼崽可能离开雌性进行独立探索;当 $T < 0.6$ 时,行为由随机数 r_6 决定: $r_6 > 0.5$ 使其远离雌性但仍靠近种群, $r_6 < 0.5$ 则可能使其完全迷失方向(受 r_7 影响),远离

群体和母亲。

$$h = \begin{cases} I_2 \times \vec{r}_1 + (\sim Q_1) \\ 2 \times \vec{r}_2 - 1 \\ \vec{r}_3 \\ I_1 \times \vec{r}_4 + (\sim Q_2) \\ r_5 \end{cases} \quad (5)$$

$$T = \exp\left(-\frac{t}{T}\right)$$

式中: $\vec{r}_1 \sim \vec{r}_4$ 是 0~1 的随机向量, r_5, r_7 是 0~1 的随机数, $\vec{h}_1$ 和 $\vec{h}_2$ 为从预定义的向量集合中随机选取的向量, 用于增加位置更新的多样性, Q_1, Q_2 为 0 或 1, 是整数随机数, t 是当前迭代次数, T 是最高迭代次数。通过比较雄性河马、雌性河马、幼年河马的适应度值, 对适应度值进行更新^[9]。

用概率选择的随机游走策略如式(6)所示^[10], 对各类河马位置进行再次更新, 为了平衡探索与利用的权衡, 选择概率 $P = 0.5$ 作为更新阈值。对每个个体生成随机数 $\text{rand} \in [0, 1]$, $\text{rand} > p$ 时执行随机游走更新, 否则保持当前位置不变。当触发更新时, 采用动态范围调整策略。其中 $\max(x_i)$ 和 $\min(x_i)$ 分别表示当前个体在维度上的极值, lb_j^{local} 和 ub_j^{local} 为变量边界, 引入分布信息, 使游走步长随搜索进程自适应调整: 初期种群分散时步长较大增强全局探索; 后期种群聚集时步长减小提升局部开发^[11]。阶段结束通过比较 $X_i, X_i^{\text{FBhippo}}, X_i^{\text{Mhippo}}$ 的适应度值, 选取其中最佳的作为更新后的个体位置。

$$X_{ij}^{\text{hippo}} = (X_{ij}^{\text{hippo}} - lb_j^{\text{local}}) \times \frac{\max(X_{ij}^{\text{hippo}}) - \min(X_{ij}^{\text{hippo}})}{ub_j^{\text{local}} - lb_j^{\text{local}}} + \min(X_{ij}^{\text{hippo}}) \quad (6)$$

传统硬边界约束采用截断式处理如式(7)所示, 虽然实现简单, 但会引发边界聚集效应和方向信息丢失问题。为解决这一局限, 在改进算法的随机游走策略和余弦自适应 t 分布变异策略中均采用双面镜反射边界处理理论如式(8)所示, 保留种群的探索潜力增强稳定性^[12]。

$$y = \begin{cases} ub, & y > ub \\ lb, & y < lb \end{cases} \quad (7)$$

$$y = \begin{cases} ub - \text{mod}(y - ub, ub - lb), & y > ub \\ lb + \text{mod}(lb - y, ub - lb), & y < lb \end{cases} \quad (8)$$

1.3 基于余弦自适应 t 分布变异的防御捕食者策略

抵御捕食者是河马选择群居生活的原因之一, 在面对捕食者时河马会选择转向捕食者并主动接近, 发出响亮的叫声, 以阻挡捕食者靠近, 诱导其撤退。捕食者的位置初始化公式为:

$$\text{Predator: Predator}_j = lb_j + \vec{r}_8 \times (ub_j - lb_j) \quad (9)$$

$j = 1, 2, \dots, m$

$\vec{r}_8$ 为随机向量, 用于搜索空间中随机初始化捕食者的位置。对于每个河马个体, 计算其与捕食者的距离, 以判断

捕食者的威胁程度。公式为:

$$\vec{D} = \|\text{Predator}_j - x_{i,j}\| \quad (10)$$

根据河马与捕食者的距离, 决定河马是向捕食者靠近以威慑它, 还是在一定范围内移动以保持警惕。公式为:

$$x_i^{\text{HippoR}}; x_{ij}^{\text{HippoR}} = \begin{cases} \vec{RL} \oplus \text{Predator}_j + \left(\frac{f}{(c-d \times \cos(2\pi g))}\right) \times \left(\frac{1}{\vec{D}}\right), & F_{\text{Predator}_j} < F_i \\ \vec{RL} \oplus \text{Predator}_j + \left(\frac{f}{(c-d \times \cos(2\pi g))}\right) \times \left(\frac{1}{2 \times \vec{D} + \vec{r}_g}\right), & F_{\text{Predator}_j} \geq F_i \end{cases}$$

$$i = \left[\frac{N}{2}\right] + 1, \left[\frac{N}{2}\right] + 2, \dots, N, j = 1, 2, \dots, m \quad (11)$$

在此基础上采用余弦自适应 t 分布变异策略来再次更新河马面对捕食者时的位置设置固定选择概率 0.5 作为更新阈值^[13], 当随机数大于更新阈值触发变异更新, 将迭代过程转化为自由度参数, 使自由度随进化进程余弦渐近增长^[14], 具体公式为:

$$tf = \frac{1 - \cos\left(\pi \times \frac{t}{T}\right)}{2} \quad (12)$$

若随机数满足 $\text{rand} > p$ 则以当前最优解为参考, 由式(16)所示生成新个体, 否则保持原个体不变。

$$x_{ij}^{\text{HippoR}} = x_{\text{best}} + x_{\text{best}} \times t \text{rand}(tf) \quad (13)$$

式中: t 是当前迭代次数, T 是最大迭代次数, 对于每个河马个体, 如果随机生成的数大于更新阈值, 其新位置通过式(13)进行计算, x_{best} 是当前种群中目标函数值最优的个体位置, 早期迭代 t 分布呈重尾特性, 变异步长较大, 后期迭代 t 分布逼近高斯分布, 变异步长减小, 进行精细调整^[15]。

在抵御捕食者阶段, 河马会根据 F_{Predator_j} 因子的大小采取不同的抵御措施, 当因子小于 F_i 证明捕食者与河马距离过近, 河马会立即转向捕食者迫使其撤退, 当因子大于 F_i 河马会发出叫声使捕食者意识到其的存在。式(11)中: x_{ij}^{HippoR} 代表河马面对捕食者时的位置, f, c, g, d 分别为在 $[2, 4], [1, 1.5], [2, 3], [-1, 1]$ 之间的均匀随机数, 其中 $\vec{r}_g$ 是维数 m 的一个随机向量, 用于增加额外的随机性。 $\vec{RL}$ 是遵循莱维分布的随机向量^[16], 莱维运动的数学模型如式(14)、(15)所示。

$$\text{Levy}(\vartheta) = 0.05 \times \frac{\omega \times \sigma_\omega}{|v|^{\frac{1}{\vartheta}}} \quad (14)$$

$$\sigma_\omega = \left[\frac{\Gamma(1+\vartheta) \times \sin\left(\frac{\pi\vartheta}{2}\right)}{\Gamma\left(\frac{1+\vartheta}{2}\right) \times \vartheta \times 2^{\frac{(\vartheta-1)}{2}}}\right]^{\frac{1}{\vartheta}} \quad (15)$$

式中: ω 和 v 是 0~1 的随机数, ϑ 是一个常数等于 1.5, Γ 是 Gamma 函数。在河马抵御捕食者的过程中, 会根据防御行为的结果, 更新河马个体的位置。若防御后 x_{ij}^{HippoR} 适应度值, 小于目标适应度的值则更新最佳位置, 否则维持之前位置。

1.4 基于余弦自适应 t 分布变异的逃离捕食者策略

当河马无法用防御行为击退捕食者的时候,会试图逃离此区域以避免伤害,通过模拟逃避行为,算法能够快速在局部区域内找到更优的解。在此基础上同样采用余弦自适应 t 分布变异策略^[17],当 $\text{rand} > p$ 时,由式(17)所示再次更新河马逃离捕食者时的位置。

$$x_{ij}^{\text{HippoE}}; x_{ij}^{\text{HippoE}} = x_{ij} + r_{10} \times (lb_j^{\text{local}} + s_1 \times (ub_j^{\text{local}} - lb_j^{\text{local}})) \quad (i = 1, 2, \dots, N, j = 1, 2, \dots, m) \quad (16)$$

$$x_{ij}^{\text{HippoE}} = x_{\text{best}} + x_{\text{best}} \times t \text{rand}(tf) \quad (17)$$

$$lb_j^{\text{local}} = \frac{lb_j}{t}, ub_j^{\text{local}} = \frac{ub_j}{t}, t = 1, 2, \dots, T$$

$$\tau = \begin{cases} 2 \times \vec{r}_{11} - 1 \\ r_{12} \\ r_{13} \end{cases} \quad (18)$$

式中: x_{ij}^{HippoE} 代表搜索后找到的最近安全位置, s_1 是从预定义的向量集合随机选取的向量, t 和 T 分别是当前迭代次数和最大迭代次数,用于控制局部搜索区域的范围^[18]。

r_{10} 是介于 0~1 的随机数,用于控制步长大小。 $\vec{r}_{11}$ 代表 0~1 的随机向量, r_{10} 和 r_{13} 表示在 0 和 1 的范围内产生的随机数, r_{12} 是正态分布的随机数^[19]。若 X_i^{HippoE} 的适应度值小于目标适应度的值则更新最佳位置,否则维持不变。更新式为:

$$X_i = \begin{cases} X_i^{\text{HippoE}}, & F_i^{\text{HippoE}} < F_i \\ X_i, & F_i^{\text{HippoE}} \geq F_i \end{cases} \quad (19)$$

2 算法实现

改进后的 HO 流程如图 1 所示。

3 实验与结果分析

3.1 对比算法和参数设置

为了验证改进后的 HO 性能,本文选取基础 HO、哈里斯鹰优化算法(harris hawks optimization, HHO)、灰狼优化算法(gray wolf optimizer, GWO)、蜣螂优化算法(dung beetle optimizer, DBO)进行对比实验。测试函数如表 1 所示,在每个测试函数中,对这 5 种算法执行 30 次重复测试,设置种群数量为 30,最大迭代次数为 500,分别在 30 维度和 100 维度下进行测试。

3.2 标准测试函数

为验证改进 HO 的寻优精度、收敛性能及改进策略的有效性,本文选取 8 个标准测试函数进行系统性实验。

3.3 收敛性分析

在 8 种测试函数下,6 种算法的收敛曲线如图 2 所示。在单峰测试函数(F1~F4)上,MSCHO 算法展现出了显著的收敛性能优势。其迭代速度明显快于其他对比算法,能够在更短的迭代次数内迅速逼近最优解区域,稳定地收敛到函数的理论最优值,证明了其在处理相对简单、梯度信息明确问题上的高效性与可靠性。对于测试函数 F5 和 F6,

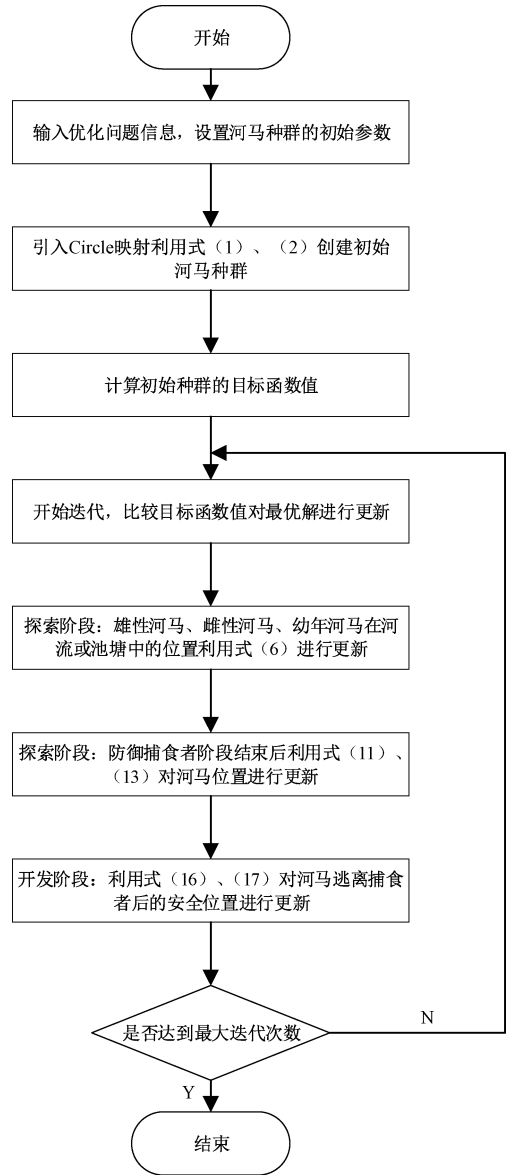


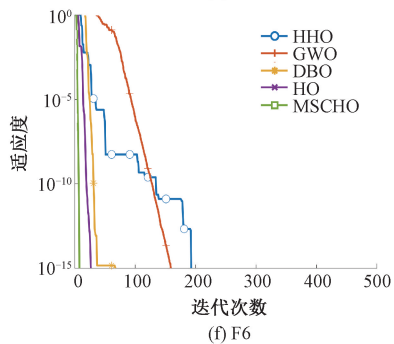
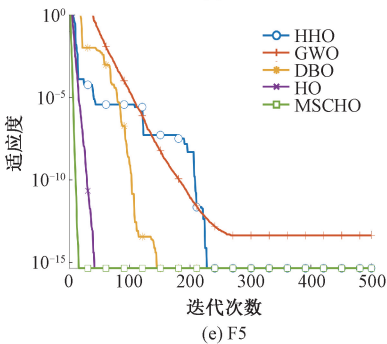
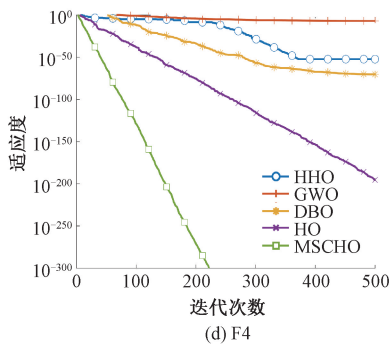
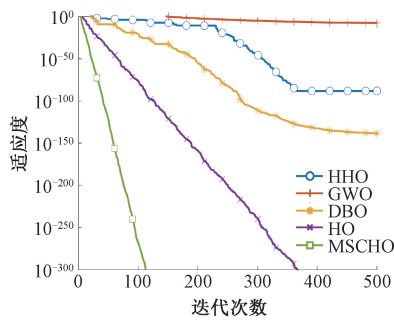
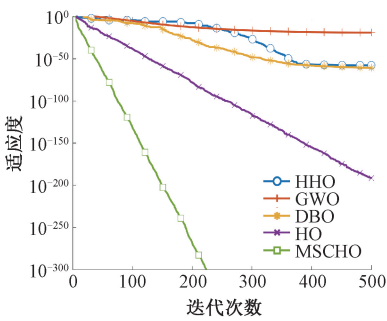
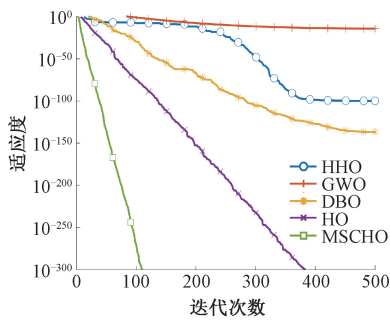
图 1 MSCHO 流程

Fig. 1 Flowchart of the hippo optimization algorithm with multi-strategy collaborative improvement

虽然所有参与比较的算法最终都能达到相近的最优解水平,收敛结果在精度上相差不大,但 MSCHO 算法的收敛效率优势依然突出,显著快于其他算法的收敛速度,如图 2(e)、(f)。在具有多个局部最优解的多峰复杂测试函数(F7 和 F8)上,改进后算法的优越性进一步得到体现。相较于对比算法,能以更快速度寻找到质量更高的全局最优解或接近全局最优的解,如图 2(g)、(h)所示。收敛曲线显示,MSCHO 在迭代早期就能获得较优的解,并持续向更优方向进化,有效避免了过早陷入局部最优陷阱。这充分说明本文的改进策略显著增强了算法在复杂搜索空间中的全局探索能力和跳出局部最优的能力。

表 1 标准测试函数
Table 1 Standard test function

编号	函数表达式	区间	最优值	类型
F1	$f(x) = \sum_{i=1}^n x_i^2$	$[-100,100]$	0	单峰
F2	$f(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	$[-10,10]$	0	单峰
F3	$f(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	$[-100,100]$	0	单峰
F4	$f(x) = \max_i x_i $	$[-100,100]$	0	单峰
F5	$f(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$	$[-32,32]$	0	多峰
F6	$f(x) = \frac{1}{4\,000}\sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1$	$[-600,600]$	0	多峰
F7	$f(x) = \frac{\pi}{n}\left\{10\sin(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})] + (y_i - 1)^2\right\} + \sum_{i=1}^n u(x_i, 10, 100, 4)$	$[-50,50]$	0	多峰
	$y_i = 1 + \frac{x_i + 1}{4}u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i - a \\ 0, & -a < x_i < a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$			
F8	$f(x) = 0.1\{\sin^2(3\pi x_i) + \sum_{i=1}^n (x_i - 1)^2[1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2[1 + \sin^2(2\pi x_n)]\} + \sum_{i=1}^n u(x_i, 5, 100, 4)$	$[-50,50]$	0	多峰



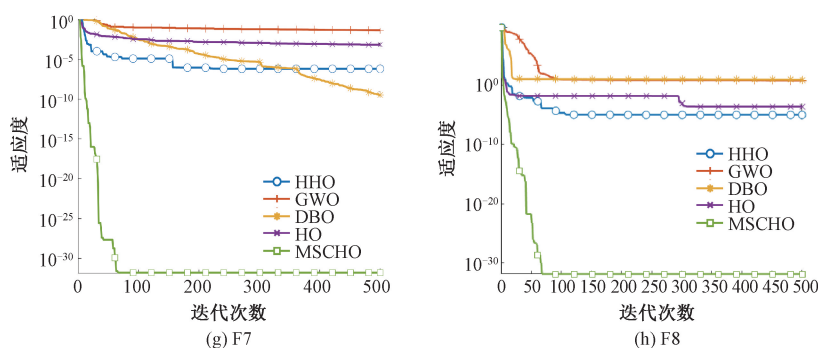


图 2 不同算法收敛曲线对比

Fig. 2 Comparison of convergence curves of different algorithms

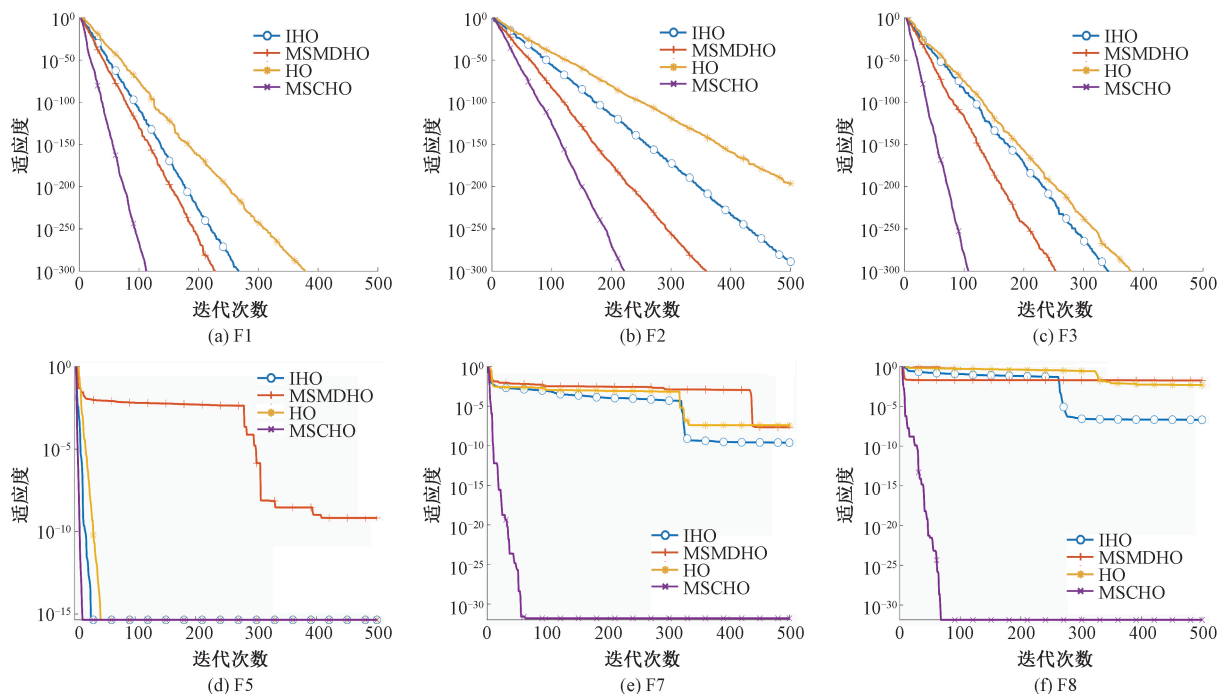


图 3 不同改进的河马优化算法收敛对比

Fig. 3 Convergence comparison of different improved hippo optimization algorithms

3.4 不同改进算法对比

为验证本文算法改进优越性,将 MSCHO 与最新改进的 HO 进行对比。引入准反向学习、正余弦优化策略、飞行策略和 PID 搜索因子的河马优化算法(improved hippo optimization algorithm based on multi-strategy and multi dimension fusion, MSMDHO)^[5]和引入 Piecewise 映射、自适应权重策略以及反向学习策略的河马优化算法(improved hippo optimization algorithm, IHO)^[6]。算法种群数量为 50,迭代次数为 500,结果如图 3 所示。结果表明在无论单峰函数还是多峰函数中,MSCHO 收敛速度最快,在多峰函数中其收敛精度能要优于 IHO。相较于其他算法,MSCHO 具有较大优势。

3.5 精度对比与分析

本研究通过比较各算法在标准测试函数上的平均值

和标准差来评估其性能。平均值作为寻优精度的度量,标准差则用于评估算法的稳定性水平。详细的仿真实验结果对比请参见表 2 和 3。实验表明无论是在 30 维还是 100 维,改进后的算法相比其他算法以及未改进前的算法,求解精度更加出色,具有更好的全局搜索能力和稳定性,求解能力得到很大提升。

3.6 改进策略有效性分析

为验证 Circle 映射、镜像极差随机游走策略和余弦自适应 t 分布变异对 HO 的改进效果,将基本算法、3 种单策略改进算法(MSCHO-1: Circle 映射; MSCHO-2: 镜像极差随机游走; MSCHO-3: 余弦自适应 t 分布变异)及综合策略的 MSCHO 进行对比。在 8 个标准测试函数上各运行 30 次,其结果如表 4 所示。平均值与标准差结果显示:在寻优精度(更优的平均解)以及收敛稳定性(更低的标

表 2 单峰、多峰测试函数 30 维度测试结果对比

Table 2 Comparison of 30-dimensional test results of single-peak and multi-peak test functions

算法	统计	F1	F2	F3	F4	F5	F6	F7	F8
HHO	Ave	2.39×10^{-100}	1.94×10^{-53}	2.62×10^{-86}	7.56×10^{-51}	4.44×10^{-16}	$0.00 \times 10^{+00}$	2.43×10^{-06}	3.32×10^{-05}
	Std	1.30×10^{-99}	7.10×10^{-53}	1.43×10^{-85}	2.81×10^{-50}	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	2.79×10^{-06}	5.01×10^{-05}
GWO	Ave	3.57×10^{-33}	8.42×10^{-20}	4.98×10^{-08}	2.39×10^{-08}	4.42×10^{-14}	5.73×10^{-03}	3.13×10^{-02}	3.99×10^{-01}
	Std	5.84×10^{-33}	6.83×10^{-20}	1.06×10^{-07}	3.79×10^{-08}	5.63×10^{-15}	7.35×10^{-03}	1.89×10^{-02}	2.34×10^{-01}
DBO	Ave	8.78×10^{-99}	8.18×10^{-60}	2.48×10^{-42}	3.84×10^{-56}	4.44×10^{-16}	4.19×10^{-04}	7.16×10^{-10}	7.21×10^{-02}
	Std	4.81×10^{-98}	4.47×10^{-59}	1.36×10^{-41}	2.01×10^{-55}	$0.00 \times 10^{+00}$	2.29×10^{-03}	1.02×10^{-09}	7.56×10^{-02}
HO	Ave	$0.00 \times 10^{+00}$	1.50×10^{-192}	$0.00 \times 10^{+00}$	7.88×10^{-191}	4.44×10^{-16}	$0.00 \times 10^{+00}$	4.48×10^{-05}	2.07×10^{-03}
	Std	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	1.45×10^{-04}	6.51×10^{-03}
MSCHO	Ave	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	1.57×10^{-32}	1.35×10^{-32}
	Std	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	5.57×10^{-48}	5.58×10^{-48}

表 3 单峰、多峰测试函数 100 维度测试结果对比

Table 3 Comparison of 100-dimensional test results of single-peak and multi-peak test functions

算法	统计	F1	F2	F3	F4	F5	F6	F7	F8
HHO	Ave	2.79×10^{-97}	1.60×10^{-51}	1.42×10^{-67}	6.77×10^{-51}	4.44×10^{-16}	$0.00 \times 10^{+00}$	2.07×10^{-06}	4.80×10^{-05}
	Std	1.53×10^{-96}	5.47×10^{-51}	7.80×10^{-67}	3.54×10^{-50}	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	3.97×10^{-06}	5.11×10^{-05}
GWO	Ave	3.79×10^{-15}	1.34×10^{-09}	$1.48 \times 10^{+02}$	2.19×10^{-01}	7.22×10^{-09}	2.95×10^{-03}	2.13×10^{-01}	$6.15 \times 10^{+00}$
	Std	3.02×10^{-15}	3.96×10^{-10}	$1.99 \times 10^{+02}$	3.04×10^{-01}	2.67×10^{-09}	6.87×10^{-03}	5.41×10^{-02}	5.18×10^{-01}
DBO	Ave	3.63×10^{-124}	5.39×10^{-63}	3.63×10^{-34}	9.52×10^{-49}	5.62×10^{-16}	$0.00 \times 10^{+00}$	2.03×10^{-02}	$7.21 \times 10^{+00}$
	Std	1.45×10^{-123}	2.21×10^{-62}	1.99×10^{-33}	5.15×10^{-48}	6.48×10^{-16}	$0.00 \times 10^{+00}$	1.51×10^{-02}	6.10×10^{-01}
HO	Ave	$0.00 \times 10^{+00}$	4.48×10^{-191}	$0.00 \times 10^{+00}$	5.29×10^{-191}	4.44×10^{-16}	$0.00 \times 10^{+00}$	6.23×10^{-05}	1.20×10^{-03}
	Std	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	2.16×10^{-04}	2.45×10^{-03}
MSCHO	Ave	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	4.44×10^{-16}	$0.00 \times 10^{+00}$	1.57×10^{-32}	1.35×10^{-32}
	Std	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	5.57×10^{-48}	5.58×10^{-48}

表 4 改进策略有效性结果对比

Table 4 Comparison of effectiveness results of improvement strategies

算法	统计	F1	F2	F3	F4	F5	F6	F7	F8
HO	Ave	$0.00 \times 10^{+00}$	7.33×10^{-190}	$0.00 \times 10^{+00}$	4.19×10^{-191}	4.44×10^{-16}	$0.00 \times 10^{+00}$	3.07×10^{-05}	7.38×10^{-04}
	Std	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	1.17×10^{-04}	3.27×10^{-03}
MSCHO-1	Ave	$0.00 \times 10^{+00}$	2.67×10^{-205}	$0.00 \times 10^{+00}$	1.30×10^{-197}	4.44×10^{-16}	$0.00 \times 10^{+00}$	1.44×10^{-05}	1.92×10^{-04}
	Std	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	1.84×10^{-04}	5.09×10^{-03}
MSCHO-2	Ave	$0.00 \times 10^{+00}$	7.78×10^{-214}	$0.00 \times 10^{+00}$	8.06×10^{-213}	4.44×10^{-16}	$0.00 \times 10^{+00}$	1.57×10^{-32}	5.17×10^{-12}
	Std	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	5.57×10^{-48}	2.83×10^{-11}
MSCHO-3	Ave	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	4.44×10^{-16}	$0.00 \times 10^{+00}$	4.12×10^{-06}	3.89×10^{-04}
	Std	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	4.21×10^{-06}	1.37×10^{-03}
MSCHO	Ave	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	4.44×10^{-16}	$0.00 \times 10^{+00}$	1.57×10^{-32}	3.91×10^{-32}
	Std	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	5.57×10^{-48}	1.40×10^{-31}

准差)上可以看出,改进算法(MSCHO-1至MSCHO-3及MSCHO)的性能均显著优于基本HO。充分验证了3种策略各自及协同整合的有效性,显著提升了算法的整体性能。

3.7 Wilcoxon 秩和检验

Wilcoxon 秩和检验具有不依赖数据分布、对异常值的

鲁棒性、适用于小样本数据,考虑相对差异,易于解释和理解等特点,所以在评估算法改进有效性时,可以帮助得到可靠的结果^[20]。本文在6个测试函数上,通过30次重复运行收集MSCHO及对比的4种算法性能数据,进行Wilcoxon 秩和检验,其结果如表5所示,除去在单峰函数

表 5 Wilcoxon 秩和检验结果
Table 5 Wilcoxon rank sum test results

算法	F1	F2	F3	F4	F7	F8
HHO	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}	1.7×10^{-12}	1.21×10^{-12}
GWO	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}	1.7×10^{-12}	1.21×10^{-12}
DBO	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}	1.7×10^{-12}	1.21×10^{-12}
HO	NaN	1.21×10^{-12}	NaN	1.21×10^{-12}	1.7×10^{-12}	1.21×10^{-12}

F1 和 F3 中与原算法算力相同,其余所有 P 值均小于 0.05,表明 MSCHO 与 4 个对比算法存在显著差异,证明了改进的有效性。

4 实际工程应用分析

为验证所提出的 MSCHO 算法在典型工程优化问题中的有效性与适用性,本文选取焊接梁优化问题(welded

beam design problem)、阶梯皮带轮设计问题(step-cone pulley design problem)各进行测试^[21]。实验参数设置种群规模为 30,最大迭代次数为 1 000,得到最优成本值和最佳变量,其结果如表 7、8 所示。

4.1 焊接梁优化问题(welded beam design problem)

为了验证所提出算法在工程设计优化问题中的有效性,本文选取经典的焊接梁设计问题(welded beam design

表 6 焊接梁设计问题
Table 6 Optimization problem of welded beams

参数	P	L	E	G	$\tau_{\max}$	$\sigma_{\max}$	$\delta_{\max}$
含义	外载荷	梁长度	弹性模量	剪切模量	最大剪应力	最大正应力	最大挠度
数值	6 000	14	30×10^6	12×10^6	13 600	30 000	0.25

表 7 焊接梁设计问题测试结果对比
Table 7 Comparison of test results for welded beam optimization problem

算法	最优成本	X_1	X_2	X_3	X_4
MSCHO	1.696 2	0.203 2	3.286 9	9.036 6	0.205 7
DBO	1.773 2	0.201 4	2.941 6	10.000 0	0.201 4
HHO	1.845 1	0.152 0	5.036 2	9.132 0	0.205 3
GWO	1.699 3	0.201 5	3.325 5	9.039 5	0.205 7
HO	2.220 1	0.356 4	2.214 0	6.863 2	0.356 7
MSMDHO	1.713 1	0.192 3	3.536 4	9.036 6	0.205 7
IHO	1.768 1	0.198 7	3.317 8	9.545 7	0.204 1

表 8 阶梯皮带轮设计问题测试结果对比
Table 8 Comparison of test results for the design issues of stepped pulleys

算法	最优成本	$D1$	$D2$	$D3$	$D4$	w
MSCHO	16.624	39.686 6	54.611 3	72.809 1	87.294 5	87.113 2
DBO	18.256	40.916 8	56.305 4	75.067 6	90.000 0	90.000 0
HHO	26.156	40.244 6	55.376 6	73.844 0	88.521 8	85.895 3
GWO	21.513	40.914 3	56.304 2	75.058 5	90.000 0	87.706 6
HO	16.827	40.162 1	55.266 0	73.682 1	88.340 2	86.098 0
MSMDHO	18.022	40.915 4	56.303 6	75.066 1	89.995 4	87.873 3
IHO	18.256	40.916 8	56.305 4	75.067 6	90.000 0	90.000 0

problem)作为测试算例。该问题旨在最小化焊接结构的制造成本,同时满足应力、挠度、几何及屈曲等约束条件,

是典型的非线性约束优化问题。定义设计变量 $x = [x_1, x_2, x_3, x_4]^T$, 其中 x_2 为焊缝厚度, x_3 为焊缝长度, x_4 为梁

高度, x_1 为梁厚度。焊接梁的制造成本可表示为:

$$\min f(x) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14 + x_2) + F_1 \quad (20)$$

式中: F_1 是问题的 7 个非线性约束条件, 分别为剪应力、正应力、挠度、几何及屈曲约束, 通过式(28)将约束转化为优化目标函数。

$$\begin{cases} g_1(x) = \tau - \tau_{\max} \leq 0 \\ g_2(x) = \sigma_x - \sigma_{\max} \leq 0 \\ g_3(x) = \delta_x - \delta_{\max} \leq 0 \\ g_4(x) = x_1 - x_4 \leq 0 \\ g_5(x) = P - P_c \leq 0 \\ g_6(x) = 0.125 - x_1 \leq 0 \\ g_7(x) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14 + x_2) - 5 \leq 0 \end{cases} \quad (21)$$

剪应力:

$$\begin{cases} \tau = \sqrt{\tau_1^2 + 2\tau_1\tau_2 \frac{x_2}{2R} + \tau_2^2} \\ \tau_1 = \frac{P}{\sqrt{2}x_1x_2} \\ \tau_2 = \frac{MR}{J} \end{cases} \quad (22)$$

弯曲应力:

$$\sigma_x = \frac{6PL}{x_4x_3^2} \quad (23)$$

挠度:

$$\delta_x = \frac{4PL^3}{Ex_4x_3^3} \quad (24)$$

临界载荷:

$$P_c = \frac{4.013E \sqrt{\frac{x_3^2x_4^6}{36}}}{L^2} (1 - \frac{x_3}{2L} \sqrt{\frac{E}{4G}}) \quad (25)$$

$$\begin{cases} M = P(L + 0.5x_2) \\ R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2} \end{cases} \quad (26)$$

$$J = 2\sqrt{2}x_1x_2 \left(\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2 \right) \quad (27)$$

$$F_1 = 10^{10} \times \sum_{i=1}^7 [\max(0, f_i)]^2 \quad (28)$$

4.2 阶梯皮带轮设计问题(step-cone pulley design problem)

阶梯皮带轮设计问题最早由 Rao 在 2011 年提出, 用于验证优化算法在多约束机械设计问题中的性能, 该问题要求在满足传动比、功率及张力比约束的条件下, 使皮带轮的总重量最小化。目标函数为:

$$\min f(x) = \rho\omega \frac{\pi}{4} \sum_{i=1}^4 d_i^2 \left(1 + \left(\frac{N_i}{N} \right)^2 \right) + F_2 \quad (29)$$

式中: $x = [d_1, d_2, d_3, d_4, w]$ 分别为 4 个轮径及带轮宽度, ρ 为材料密度, N_i 为从动轮转速, N 为主动轮转速。

$$\begin{cases} C_1 - C_2 = 0 \\ C_1 - C_3 = 0 \\ C_1 - C_4 = 0 \end{cases} C_i = \frac{\pi d_i}{2} \left(1 + \frac{N_i}{N} \right) + \frac{\left(\frac{N_i}{N} - 1 \right)^2 d_i^2}{4a} + 2a \quad (30)$$

式中: 3 条等式约束确保每级皮带长度一致。

$$\begin{cases} R_i = \exp\left[\mu\left(\pi - 2\arcsin \frac{\left(\frac{N_i}{N} - 1\right)d_i}{2a}\right)\right] \\ P_i = st\omega[1 - \exp(-\mu\left(\pi - 2\arcsin \frac{\left(\frac{N_i}{N} - 1\right)d_i}{2a}\right))] \frac{\pi d_i N_i}{60} \end{cases} \quad (31)$$

$$F_2 = 10^{10} \times \left(\sum_{i=1}^4 [\max(0, g_i)]^2 + \sum_{j=1}^3 [\max(h_j)]^2 \right) \quad (32)$$

式中: 将约束转化为优化目标函数; g_i 为不等式约束 R_i 和 P_i ; h_j 为等式约束。

张力比与功率约束为:

$$R_i \geq 2, P_i \geq 0.75 \times 745.6998, i = 1, 2, 3, 4,$$

$$\rho = 7200, a = 3, \mu = 0.35, s = 1.75 \times 10^6,$$

$$t = 8 \times 10^{-3}, N = 350, N_1 = 750, N_2 = 450,$$

$$N_3 = 250, N_4 = 150.$$

5 结 论

针对基本 HO 收敛慢、易陷局部最优及稳定性不足的问题, 本研究提出一种融合 Circle 混沌映射、镜像极差随机游走策略和余弦自适应 t 分布变异的多策略协同改进算法(MSCHO)。Circle 映射优化了种群初始化, 提升多样性; 随机游走策略增强了局部勘探能力; 余弦自适应 t 变异则有效平衡了探索与开发, 避免早熟收敛。在 8 个标准测试函数上的实验表明 MSCHO 显著优于其他算法, 尤其在多峰函数上性能提升明显。这充分验证了 3 种策略各自及协同改进的有效性, 显著提升了 HO 的全局寻优能力、收敛速度及鲁棒性。将 MSCHO 算法应用于焊接梁设计、阶梯皮带轮设计, 发现 MSCHO 算法在工程优化问题上表现出更低的生产成本, 具有更好的潜力。未来将探索 MSCHO 在工程优化等实际问题中的应用价值。

参考文献

- [1] AMIRI M H, HASHJIN N M, MONTAZERI M, et al. Hippopotamus optimization algorithm: A novel nature-inspired optimization algorithm[J]. Scientific Reports, 2024, 14(1): 5032.
- [2] GUO X F, HUANG Y M, ZHANG S P, et al. Constructability optimization of rebar cages using hippopotamus optimization and double deep Q-network[J]. Automation in Construction, 2025, 175: 106224.
- [3] CHAIB L, TADJ M, CHOUCHE A, et al. Hybrid

- brown-bear and hippopotamus algorithms with fractional order chaos maps for precise solar PV model parameter estimation[J]. Processes, 2024, 12: 2718.
- [4] HE X W, LIU X H, LIN C, et al. Early bearing fault diagnosis in PMSMs based on HO-VMD and weighted evidence fusion of current-vibration signals [J]. Sensors, 2025, 25: 4591.
- [5] 任庆欣,冯锋.多策略多维度融合改进的河马优化算法[J].计算机科学,2025,52(S1):313-320.
- REN Q X, FENG F. Improved hippo optimization algorithm based on multi-strategy and multi-dimension fusion[J]. Computer Science, 2025, 52(S1): 313-320.
- [6] 袁磊.融合多策略的改进河马优化算法及其应用[J].海南大学学报(自然科学版中英文),2025,43(3): 289-296.
- YUAN L. Improved hippopotamus optimization algorithm integrating multi-strategy and its applications[J]. Journal of Hainan University(Natural Science Edition), 2025, 43(3): 289-296.
- [7] 张勇,唐睿哲,刘畅,等.基于改进蛇优化算法的WSN覆盖优化研究[J].电子测量技术,2024,47(16): 24-32.
- ZHANG Y, TANG R Z, LIU C, et al. Research on WSN coverage optimization based on improved snake optimization algorithm [J]. Electronic Measurement Technology, 2024, 47(16): 24-32.
- [8] 张小庆,孙民民,张莉,等.基于改进双曲正余弦优化算法的无人机路径协同规划[J].电子测量与仪器学报,2025,39(9):137-149.
- ZHANG X Q, SUN M M, ZHANG L, et al. Cooperative path planning of UAVs based on improved hyperbolic sine cosine optimization algorithm [J]. Journal of Electronic Measurement and Instrumentation, 2025, 39(9): 137-149.
- [9] 张莲,贾浩,张尚德,等.融合自适应t分布和随机游走策略的松鼠优化算法的研究[J].计算机与数字工程,2024,52(8):2343-2347,2410.
- ZHANG L, JIA H, ZHANG S D, et al. Research on squirrel optimization algorithm integrating adaptive t-distribution and random walk strategy[J]. Computer and Digital Engineering, 2024, 52(8): 2343-2347,2410.
- [10] WANG W X, LI K S, TAO X Z, et al. An improved MOEA/D algorithm with an adaptive evolutionary strategy[J]. Information Sciences, 2020, 539: 1-15.
- [11] 陈洪芳,吴欢,王子帅,等.基于改进粒子群算法的三坐标测量机最佳测量区域评价方法[J].仪器仪表学报,2024,45(11):197-205.
- CHEN H F, WU H, WANG Z S, et al. Evaluation method for optimal measurement zones of coordinate measuring machines based on an improved particle swarm optimization algorithm[J]. Chinese Journal of Scientific Instrument, 2024, 45(11): 197-205.
- [12] 吴智祥,刘杰,覃涛,等.多策略改进的精英金豺优化算法[J].计算机工程与科学,2025,47(1): 1-15.
- WU Z X, LIU J, QIN T, et al. Multi-strategy improved elite golden jackal optimization algorithm[J]. Computer Engineering and Science, 2025, 47(1): 1-15.
- [13] 张颖,姜文刚,陈一鸣,等.基于改进蜘蛛蜂算法的无人机三维路径规划[J].电子测量技术,2024,47(11): 101-111.
- ZHANG Y, JIANG W G, CHEN Y M, et al. 3D path planning of UAV based on improved spider wasp algorithm[J]. Electronic Measurement Technology, 2024, 47(11): 101-111.
- [14] 李楠,薛建凯,舒慧生.基于自适应t分布变异麻雀搜索算法的无人机航迹规划[J].东华大学学报(自然科学版),2022,48(3):69-74.
- LI N, XUE J K, SHU H S. UAV path planning based on sparrow search algorithm with adaptive t-distribution mutation [J]. Journal of Donghua University(Natural Science Edition), 2022, 48(3): 69-74.
- [15] 陈悦,冯锋.基于多策略改进的蜣螂优化算法[J].物联网技术,2025,15(2):1-5.
- CHEN Y, FENG F. Improved dung beetle optimizer based on multi-strategy [J]. Internet of Things Technologies, 2025, 15(2): 1-5.
- [16] LAYEB A. Tangent search algorithm for solving optimization problems [J]. Neural Computing and Applications, 2022, 34(11): 8853-8884.
- [17] 刘金科,梁作鹏,蒲泽森,等.基于改进黏菌算法的无人船全局路径规划[J].电子测量与仪器学报,2025,39(9):111-125.
- LIU J K, LIANG Z P, PU Z S, et al. Global path planning of unmanned ship based on improved slime mould algorithm[J]. Journal of Electronic Measurement and Instrumentation, 2025, 39(9): 111-125.
- [18] 张彪,李永强.基于动态寻优蚁群算法的机器人路径规划[J].仪器仪表学报,2025,46(3):74-85.
- ZHANG B, LI Y Q. Mobile robot path planning based on dynamic optimization ant colony algorithm [J]. Chinese Journal of Scientific Instrument, 2025, 46(3): 74-85.
- [19] 毛清华,王迎港.融合改进Logistics混沌和余弦算子的自适应t分布海鸥算法[J].小型微型计算机系统,2022,43(11):2271-2277.

MAO Q H, WANG Y G. Adaptive t-distribution seagull algorithm integrating improved Logistics chaos and cosine operator[J]. Journal of Chinese Computer Systems, 2022, 43(11): 2271-2277.

[20] 张福兴,高腾,吴泓达. 多策略融合的改进黑猩猩优化算法[J]. 北京航空航天大学学报, 2025, 51(1): 235-247.

ZHANG F X, GAO T, WU H D. Improved chimpanzee optimization algorithm based on multi-strategy fusion[J]. Journal of Beijing University of Aeronautics and Astronautics, 2025, 51(1): 235-247.

[21] RAO R V, SAVSANI V J, VAKHARIA D P. Teaching-Learning-based optimization: A novel

method for constrained mechanical design optimization problems[J]. Computer-Aided Design, 2011, 43(3): 303-315.

作者简介

周建新,博士,副教授,主要研究方向为智能控制理论及应用。
E-mail:zhoujianxin1977@126.com

王彬(通信作者),硕士研究生,主要研究方向为电力设备状态检测与故障诊断。
E-mail:1290547848@qq.com

焦江涛,硕士研究生,主要研究方向为智能控制与模式识别。
E-mail:2495984756@qq.com