# 自适应常春藤 ADIVY 优化算法设计及其工程应用

刘俊毅
辽宁工程技术大学 机械工程学院 辽宁阜新 123000
DOI:10.12238/ast.v1i1.13708

[摘　要] 针对常春藤（IVY）优化算法在复杂优化问题中的全局搜索能力不足和收敛速度较慢的问题，提出了一种自适应常春藤 ADIVY 算法，提升优化效率和精度。首先，采用拉丁超立方体采样(Latin Hypercube Sampling, LHS)与黄金分割比的非均匀初始化(Golden Ratio-based Nonuniform Initialization, GRN)融合，扩展初始种群的搜索范围，并增强了种群分布的均匀性和多样性，并设计动态压缩归一化映射(Dynamic Compression Normalization Mapping, DCNM)增强边界区域探索能力，其次，融合自适应创新点和多样性维持机制(Adaptive Innovation and Diversity Maintenance Mechanism, AIDM)，在早熟收敛或多样性不足时对部分个体进行重置或变异，增强了跳出局部最优的能力。此外，设计了一种动态位置翻筋斗扰动选择(The dynamic position tumbling disturbance selection can be abbreviated as, DPTDS)机制，改进全新的 GV 长向量，结合全局最优解和局部梯度信息，平衡了大范围跳跃与精细微调，提升搜索速度与精度。最后，提出动态平滑飞行-HHO（Dynamic Smoothing Flight in Harris Hawks Optimization, DSF-HHO)策略，通过引入哈里斯鹰算法的“渐变逃逸能量”概念，减少了解空间的抖动，增强全局搜索能力。基准函数和实际工程应用的对比实验及 Wilcoxon 秩和检验结果表明，改进的 ADIVY 算法在全局搜索、鲁棒性和收敛速度上均优于原始算法与其他主流算法，验证了其在工程应用中的有效性和优越性。
[关键词] IVY 优化算法；多策略融合；多样性维持机制；动态平滑飞行；工程应用；翻筋斗云；渐变逃逸能量；全局搜索能力
中图分类号：TP18 文献标识码：A

## Adaptive Ivy ADIVY optimization algorithm design and its engineering application

Junyi Liu
Liaoning Technical University, Fuxin, Liaoning, 123000

[Abstract] To address the issues of insufficient global search capability and slow convergence speed in the Ivy (IVY) optimization algorithm for complex optimization problems, we propose an Adaptive Ivy (ADIVY) algorithm. The new algorithm improves both optimization efficiency and accuracy. Firstly, we integrate Latin Hypercube Sampling (LHS) with Golden Ratio-based Nonuniform Initialization (GRN) to expand the search range of the initial population, enhancing the population's distribution balance and diversity. Secondly, an adaptive innovation point and diversity maintenance mechanism is introduced. This mechanism resets or mutates some individuals when premature convergence or insufficient diversity occurs, strengthening the algorithm's ability to escape local optima. Additionally, a novel "Somersault Cloud Strategy" (SCS) is designed, which combines the global optimal solution with local gradient information to balance large-scale jumps with fine-tuning, thereby improving search speed and accuracy. Furthermore, a Dynamic Smoothing Flight in Harris Hawks Optimization (DSF-HHO) strategy is proposed. By introducing the concept of "gradual escape energy," this strategy reduces the oscillation of the solution space and enhances global search capabilities. Lastly, a Quantum Mechanism (QS) is introduced, which uses quantum bits and the superposition principle to generate new solutions. Candidate solutions are then selected and replaced based on the probability of the wave function, thus improving local search capabilities. Benchmark tests and comparisons with practical engineering applications, as well as results from the Wilcoxon rank-sum test, demonstrate that the improved ADIVY algorithm outperforms the original algorithm and other mainstream algorithms in global search capability, robustness, and convergence speed, validating its effectiveness and superiority in engineering applications.
[Key words] IVY optimization algorithm; Multi-strategy fusion; Diversity maintenance mechanism; Dynamic smooth flight; Engineering application; Somersault cloud; Gradient escape energy; Global search capability

Copyright © This work is licensed under a Commons Attibution-Non Commercial 4.0 International License.

## 1 引言

随着科技的进步，工程领域的优化问题日益复杂，面临高维、多约束和非线性等挑战，传统优化方法难以提供有效解决方案[1-2]。在此背景下，群智能优化算法凭借其简单、灵活和鲁棒性，成为解决复杂工程问题的重要工具。这类算法广泛应用于结构优化、路径规划、图像处理、智能制造等领域，并取得显著成果[3]。例如，粒子群优化算法（PSO）[4]在结构拓扑优化、机器人路径规划和生产调度中表现优异；蚁群算法（ACO）[5]则在网络路由和交通流量控制中展现了独特优势。

陈旭升[6]等提出的多策略融合足球训练算法（MSFTTA）通过动态适应度-距离平滑策略、非独占优劣共融搜索和FADs 扰动策略，增强了全局探索和跳出局部极值的能力。文裕杰[7]等提出的增强型白鲸算法引入基于权重的抢食型白鲸和自适应高斯策略，加快了收敛速度。吴亚中[8]等提出的多策略增强蜣螂优化算法，结合 Tent 混沌映射初始化、黄金正弦搜索和自适应 T 分布扰动，提高了跳出局部最优解的能力。Dalia T. Akl[9]等改进哈里斯鹰算法，通过随机栖息地策略和全新追逐策略提升了全局探索与局部搜索能力。刘云云[10]等总结了灰狼算法的改进策略及应用，刘成汉[11]等改进黑猩猩算法，引入 Halton 序列初始化、非线性收敛因子和自适应权重因子，平衡探索能力并增强处理局部极值的能力。以上工作展现了仿生算法在优化问题中的广泛潜力。

常春藤优化算法（IVY）[12]作为 2024 年 7 月新兴的群体智能算法，已成功应用于多种优化问题中。通过模拟常春藤植物的生长与繁殖，IVY 算法具有较强的全局搜索能力和适应性。蔡玉林[13]等将 IVY 算法应用于机器学习模型的边坡稳定性预测，优化了随机森林、决策树、多层感知器和支持向量机的超参数，展现了其优越性。王恒迪[14]等将 IVY 算法引入故障诊断领域，优化了特征模态分解(FMD)的参数，提高了模态分解精度。欧阳慧[15]等在船舶柴油发电机故障诊断中使用 IVY 算法优化随机森林的超参数，提升了预测精度。高海宁[16]等在混合神经网络系统中通过 IVY 算法优化超参数，显著提高了模型精度。然而，IVY 算法在处理复杂工程优化问题时，尤其是高维、多峰和多约束问题，仍存在全局搜索能力不足和收敛速度慢的问题。其初始种群分布固定，限制了搜索范围，容易陷入局部最优；同时，收敛过程缺乏灵活性，接近最优解时常出现过度调整，导致收敛速度变慢。吴禹志[17]等提出的多策略增强型 IVY 算法，在初始阶段通过混沌映射增强早期探索，引入适应性生长行为以提升跳出局部最优的能力，并通过长距离伸展行为提高后期寻优精度。张春强[18]等提出的新颖 IVY 算法，设计了自适应扰动因子和增长率，首次引入人口装置策略和差异演化策略，扩大了搜索空间，并提升了算法的收敛速度，但以上改进较为传统，并未有效解决高维迭代收敛精度低问题，因此，提升 IVY 算法的全局搜索能力、加速收敛并避免局部最优，成为亟待解决的关键问题。

本文提出了一种改进的常春藤优化算法—自适应常春藤 ADIVY 算法，旨在提升 IVY 算法的全局搜索能力和收敛速度，避免局部最优。首先，采用拉丁超立方体采样与黄金分割比的非均匀初始化融合，扩展初始种群的搜索范围，并增强了种群分布的均衡性和多样性，并设计动态压缩归一化映射增强边界区域探索能力，其次，融合自适应创新点和多样性维持机制，在早熟收敛或多样性不足时对部分个体进行重置或变异，增强了跳出局部最优的能力。此外，设计了一种动态位置翻筋斗扰动选择机制，结合全局最优解和局部梯度信息，平衡了大范围跳跃与精细微调，提升搜索速度与精度。最后，提出动态平滑飞行-HHO 策略，通过引入“渐变逃逸能量”概念，减少了解空间的抖动，增强全局搜索能力，实验结果表明，ADIVY 算法在全局搜索、收敛速度和鲁棒性上优于原始 IVY 算法和其他主流算法，验证了其优越性。

## 1 基本常春藤优化算法

IVY 优化算法是一种基于常春藤趋光生长机制的仿生启发式算法，具有较强的抗扰动能力，尤其在多模态、高维非凸优化问题中表现出色[14]。通过动态分枝策略平衡全局搜索与局部开发，并利用自适应生长速率调节机制，IVY 算法能够在有限的迭代次数内稳定收敛至近优解，展现了在复杂数学建模中的优势[15]。

该算法实现步骤如下：

初始化，随机生成种群的每个个体，表示如下：

$$I_i = I_{\min} + rand(\mathbf{1},\mathbf{D}) \odot (I_{\max} - I_{\min}) \tag{1}$$

式中，Imin，Imax 为搜索空间地上界和下界；rand(1,D)为区间[0,1]内均匀分布地随机数向量。

生长速率模型，常春藤的生长速率模拟它对环境的适应和资源的获取，表示如下：

$$\mathrm{d}G_v(t) / \mathrm{d}t = \psi \cdot G_v(t) \cdot \phi(G_v(t)) \tag{2}$$

式中，$G_v(t)$ 表示为当前生长速率；$\psi, \phi$ 表示为生长因子和矫正因子。

个体位置更新，根据生长速率和邻居信息，更新每个个体的位置，表示如下：

$$I_{\mathrm{i_{new}}} = I_i + | N(\mathbf{1},\mathbf{D})| \odot (I_{ii} - I_i) + N(\mathbf{1},\mathbf{D}) \odot \Delta G_{v_i} \tag{3}$$

式中，$I_{\mathrm{i\_new}}$ 表示为个体更新后的位置；$I_{ii}$ 表示为当前个体的最近邻居，$|N(1,D)|$ 表示为随机向量的绝对值；$N(1,D)$ 表示为另一个随机向量。

全局搜索和局部搜索，全局搜索向全局最优解靠近，局部搜索在附近寻找更好的解，表示如下：


Copyright © This work is licensed under a Commons Attibution-Non Commercial 4.0 International License.

$$I_{\text{new}} = I_{\text{Best}} \odot \left( rand(1,D) + N(1,D) \odot \Delta G_{v_i} \right) \quad (4)$$

式中，$\mathrm{I}_{\text{Best}}$ 表示为当前种群中的全局最优解。

（5）适应度评估与种群更新，评估每个个体的适应度，更新种群并保留最终的最优解。

## 2 自适应 ADIVY 优化算法

### 2.1 拉丁超立方采样与黄金分割融合（LHS-GRN)初始化

传统的 IVY 算法初始化种群随机生成，导致初期种群分散、不均匀，影响搜索效率，且在大规模优化中收敛速度较慢。传统的 IVY 算法采用均匀随机的方式进行初始化，其具体公式如下所示：

$$x_i = l + u \odot r, \quad r \sim \mathrm{U}(0,1)^d \quad (5)$$

其中，l 和 u 分别为种群的上下边界，r 为随机数。

传统的均匀随机方式会导致在任意维度的投影上，样本点间距方差过大：

$$\mathrm{Var}(\Delta x_j) = \frac{(u_j - l_j)^2}{12N}\left(1 + \frac{1}{N}\right) \quad (6)$$

式中 uj 和 lj 分别为 j 方向的向量边界，通过以上公式的推导，在 30 维空间中，其传统方法在单维投影的间距标准差是经过混沌映射的 2.3 倍，说明其种群分布较为分散，不均匀。

为此，本文设计了拉丁超立方采样与黄金分割融合初始化机制，以提升算法的全局搜索能力和初始解质量。

LHS 是一种高效的多维空间抽样技术，通过均匀划分维度并确保每个子区间内仅有一个样本点，从而均匀覆盖设计空间[16]。与传统随机采样相比，LHS 能在较少样本数下覆盖更广泛的搜索空间，提升采样效率。标准 LHS 步骤如下：

首先进行分层划分，将第 j 维均匀划分为 N 个互不重叠的子区间：

$$\Delta_{j,k} = \left[ a_j + \frac{(k-1)(b_j - a_j)}{N}, a_j + \frac{k(b_j - a_j)}{N} \right), \quad k = 1,2,\ldots,N \quad (7)$$

其中，设优化问题的维度为 d，种群规模为 N，对第 j 维变量范围为[aj,bj]。

对每个维度独立生成随机排列 $\pi_j : \{1,\ldots,N\} \to \{1,\ldots,N\}$，并在每个字区间内随机采样：

$$x_{i,j} = a_j + \frac{\pi_j(i) - u_{i,j}}{N}(b_j - a_j) \quad (8)$$

其中，$u_{i,j} \sim U[0,1)$ 为均匀随机数，$\pi_j$ 确保每列的唯一性。

为进一步提升种群的多样性，本文将 LHS 与黄金分割比 $\phi = (\sqrt{5}-1)/2$ 结合，生成具有拟均匀特性的初始化种群。

首先利用 MATLAB 函数的 lhsdesign 生成初始化样本矩阵 $S \in R^{N \times d}$，满足：

$$S_{i,j} \sim U\left[\frac{i-1}{N}, \frac{i}{N}\right), \text{ 经随机排列后} \quad (9)$$

对每个样本点施加黄金分割相位偏移，避免产生周期性聚集：

$$\hat{S}_{i,j} = \mathrm{mod}\left(S_{i,j} + i \cdot \phi, 1\right) \quad (10)$$

$$\hat{x}_{i,j} = \mathrm{mod}\left(x_{i,j} + i\phi \cdot \frac{b_j - a_j}{N}, b_j - a_j\right) + a_j \quad (11)$$

其中模运算保证 $\hat{S}_{i,j} \in [0,1)$，黄金分割比的非有理数特性确保偏移后的样本在投影空间均匀分布。

将归一化样本映射到变量的范围：

$$\boldsymbol{x}_i = \boldsymbol{a} + \hat{S}_{i,:} \odot (\boldsymbol{b} - \boldsymbol{a}) \quad (12)$$

其中，$\odot$表示逐元素相乘，a,b 分别表示为变量的上下界向量。

为进一步增强边界区域的探索能力，设计一种双曲正切函数的非线性映射（动态压缩归一化映射 DCNM），其具体数学公式如下所示：

$$GV_{i,j} = \frac{\tanh(4\hat{x}_{i,j} - 2)}{\tanh(2)} + 1 \quad (13)$$

其中，$\hat{x}_{i,j} = (x_{i,j} - a_j)/(b_j - a_j)$ 表示为经过归一化的位置，该映射将样本压缩到[0,2]区间，中心区域梯度大以加强局部搜索，边界区域梯度小以维持全局探索。

为体现 DCNM 混沌映射的优越性，将此混沌映射与常见的 Bernoulli、Tent、Circle、Fuch、Sine 映射进行对比，分别引入传统的 IVY 算法进行测试，其具体结果如图 1 所示，随机初始化和 DCNM 混沌映射结果图如图 2 所示

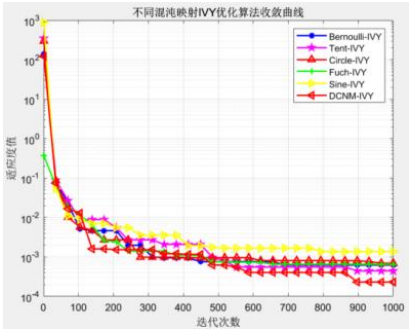
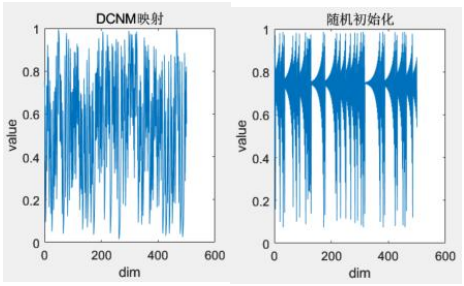



图 1 混沌映射迭代对比

Copyright © This work is licensed under a Commons Attibution-Non Commercial 4.0 International License.

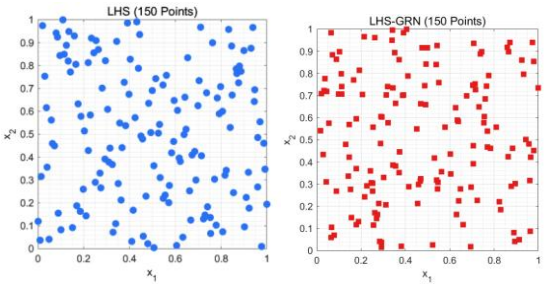



(a)DCNM 映射结果　(b)随机初始化结果

图 2 映射结果

根据图示，DCNM-IVY 显著优于其他方法，前 100 代内收敛速度更快，适应度值显著降低，收敛曲线平稳，表明优化效率较高。与 Bernoulli-IVY、Tent-IVY、Circle-IVY 和 Sine-IVY 相比，DCNM-IVY 达到更低的最终适应度值。此外，DCNM 映射显示出更均匀且稳定的采样分布，数据点分布均匀，波动较小，表现出良好的全局搜索能力，而随机初始化则存在较大波动，采样点集中于某些区域。

将标准 LHS 生成的种群分布和本文经过 LHS-GRN 生成的种群分布进行对比，其选用 150 个种群数量，具体分布对比如图 3 所示。




(a)LHS 种群分布　(b)LHS-GRN 种群分布

图 3 LHS 与 LHS-GRN 种群分布对比

LHS-GRN 相比传统 LHS 显示出更均匀的采样分布，避免了密集区域，优化了全局搜索和空间覆盖。其在采样均匀性和空间覆盖能力上优于 LHS，提高了优化效率和精度，尤其在高维问题中更为显著。

目前大多数文献常用的混沌映射为 Logistic 映射和 Tent 映射，但是这两种映射都存在中间分布均匀，两端分布密集的问题，采用 LHS 初始化可以均匀遍布整个空间，覆盖率高，本文对这三种方法的种群分布对比效果如图 4 所示。

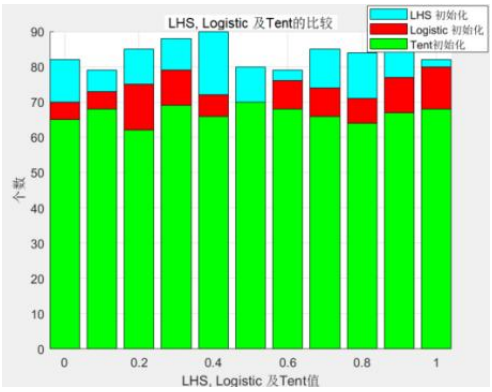



图 4 LHS、Logistic、Tent 初始化分布对比

选取种群规模为 100，对随机采样、标准 LHS 采样、本文 LHS-GRN 采样进行对比，其对比数据为最大最小距离、相关系数均值、空间覆盖率，统计得数据直方图如图 5 所示，数据如表 1 所示。

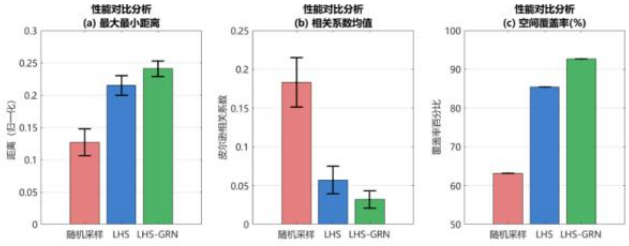



图 5 统计数据直方图

表 1 最大最小距离、相关系数均值、空间覆盖率对比数据

| 方法类别 | 最大最小距离 | 相关系数均值 | 空间覆盖率% |
|---|---|---|---|
| 随机采样 | 0.127±0.021 | 0.183±0.032 | 63.2% |
| 标准 LHS | 0.215±0.015 | 0.057±0.018 | 85.4% |
| LHS-GRN | 0.241±0.012 | 0.032±0.011 | 92.7% |

根据表格数据，LHS-GRN 在各项指标上表现优越，尤其在最大最小距离和空间覆盖率上有显著提升。其最大最小距离为 0.2410，较标准 LHS 提高 12.1%，较随机采样提高 89.8%。相关系数均值为 0.0320，分别较标准 LHS 和随机采样提高 43.9%和 82.5%。空间覆盖率为 92.7%，较标准 LHS 提高 7.6%，较随机采样提高 46.6%，显示其在搜索空间覆盖上的优势。

2.2 自适应创新点和多样性维持(AIDM)机制

原始 IVY 算法在迭代后期容易出现种群多样性急剧下降，导致早熟收敛，种群多样性损失速率定义为：

$$\tau_d = \frac{D_p^{(t)} - D_p^{(t+1)}}{D_p^{(t)}} \tag{14}$$

其中，Dp 为多样性指标，参照式子（22），通过实验测得在 Sphere 函数优化中，传统 IVY 算法在迭代中期平均


Copyright © This work is licensed under a Commons Attibution-Non Commercial 4.0 International License.

值为 0.38，而目前的主流算法 GWO 在迭代中期的平均值为 0.23，表明原始算法存在过度收敛倾向，其对比结果图如图 6 所示。

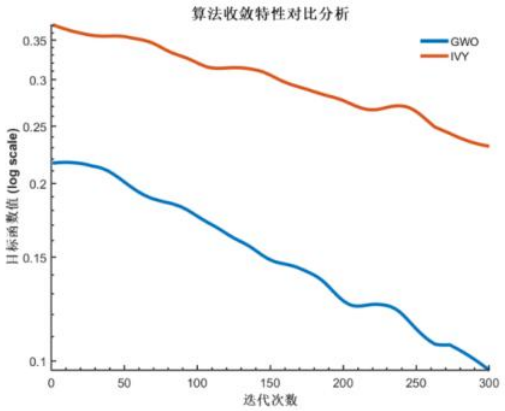



图 6 多样性指标迭代对比

原始的 IVY 算法局部最优逃逸能力存在不足，其采用 Rastrigin 函数测试中，原始算法陷入局部最优的概率达到 63.2%左右，其适应度停滞现象利用马尔可夫链进行测试，其具体公式为：

$$P_{\text{stagnation}} = 1 - \exp\left(-\frac{t}{\lambda}\right) \tag{15}$$

式中，t 表示当前的迭代次数，T 表示迭代总次数，原始 IVY 算法的迭代次数在 42 次停滞，GWO 算法在 128 次停滞。

针对以上原始 IVY 算法出现的问题，设计一种自适应创新点与多样性维持机制，在早熟收敛或多样性不足时对部分个体进行重置或变异，增强了跳出局部最优的能力。

本算法创新触发条件是通过双重指标动态创新点的引入时机，其数学判据为：

$$\text{触发条件} = \begin{cases} 1, & \text{if}\sigma_f < \theta_{\text{adapt}} \vee (t > 1 \wedge f_{\text{best}}^{t} = f_{\text{best}}^{t-1}) \\ 0, & \text{otherwise} \end{cases} \tag{16}$$

式中，$\sigma_f = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(f_i - \bar{f})^2}$ 为种群适应度标准差，$\theta_{\text{adapt}} = 0.05$ 为自适应创新阈值，$f_{\text{best}}^{t}$ 表示为第 t 代最优适应度。

每次触发时重新初始化的个体数量定义为：

$$n_{\text{innov}} = \lfloor N \cdot \eta_t \rfloor \tag{17}$$

式中，N 为种群数目，$\eta_t$ 为创新比率。其中采用加权概率分布进行选择：

$$p_i = \frac{\exp(-\alpha \cdot \text{rank}(f_i))}{\sum \exp(-\alpha \cdot \text{rank}(f_j))} \tag{18}$$

其中 a=0.5 控制选择压力，使较差的个体以更高概率被重置。

其中创新率 $\eta_t$ 按照指数衰减规则进行更新，其具体公式为：

$$\eta_t = \eta_0 \cdot 0.9^{\lfloor t/10 \rfloor} \tag{19}$$

式中，初始创新率 $\eta_0$ 取为 0.1，$\lfloor \cdot \rfloor$ 为向下取整函数，被选个体满足以下条件：

$$\mathrm{I}_{\text{innov}} = \{i \mid \text{rank}(f_i) \geq N - n_{\text{innov}}\} \tag{20}$$

式中，$rank(f_i)$ 表示为个体适应度的降序排名。

对进行以上操作选中的个体执行均匀重置，其具体公式如下所示：

$$\boldsymbol{x}_i^{t+1} = \mathrm{U}(\boldsymbol{l}, \boldsymbol{u}), \quad \forall i \in \mathrm{I}_{\text{innov}} \tag{21}$$

其中，$u(l,u)$ 表示为在决策空间[l,u]上的均匀分布。

在以上的基础上融合多样性维持机制，首先定义种群位置多样性指标，其具体公式如下：

$$D_p = \frac{1}{d}\sum_{j=1}^{d}\sigma_j^2 \tag{22}$$

其中，$\sigma_j^2 = \frac{1}{N}\sum_{i=1}^{N}(x_{i,j} - \bar{x}_j)^2$ 为第 j 维位置的方差，d 为问题维度，DP 为多样性指标，其范围是[0,1]。

构建基于 Lyapunov 函数的稳定性控制：

$$V(D_p) = \frac{1}{2}(D_p - D_{p,\min})^2 \tag{23}$$

式中，Dp,min 为最小多样性指标

当 $V(D_p) > 0.005$ 时，对精英个体实施增强变异：

首先对个体进行选择，其具体公式如下所示：

$$\mathrm{I}_{\text{elite}} = \{i \mid \text{rank}(f_i) \leq \lfloor 0.2N \rfloor\} \tag{24}$$

对选择的个体进行三重差分变异，其变异的更新量为：

$$\Delta \boldsymbol{x}_i = 0.5(\boldsymbol{x}_{r1} - \boldsymbol{x}_{r2} + \boldsymbol{x}_{r2} - \boldsymbol{x}_{r3}) + 0.1\boldsymbol{\varepsilon} \tag{25}$$

其中，$r1 \neq r2 \neq r3 \in \{1,\ldots,N\}$ 为随机索引，$\varepsilon \sim \mathrm{N}(0, \boldsymbol{I})$ 为高斯噪声扰动。

对经过差分变异的个体进行位置更新，其公式如下：

$$\boldsymbol{x}_i^{t+1} = clip(\boldsymbol{x}_i^{t} + \Delta \boldsymbol{x}_i, \boldsymbol{l}, \boldsymbol{u}) \tag{26}$$

式中，clip()为边界约束函数，l,u 为种群的上下边界。

其中的变异强度随优化进程自适应调整，其公式如下：

$$\gamma(t) = \gamma_{\max} - (\gamma_{\max} - \gamma_{\min}) \cdot \left(\frac{t}{T}\right)^2 \tag{27}$$

式中，T 为最大迭代次数，为最大变异强度和最小变异

Copyright © This work is licensed under a Commons Attibution-Non Commercial 4.0 International License.

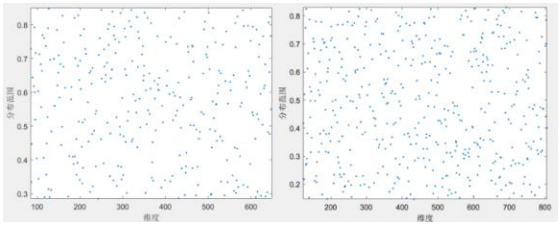
强度，分别取 0.7 和 0.2，保证前期探索，后期精细开发。

两种策略通过互补方式维持种群活力，创新点机制通过硬重置避免早熟收敛，多样性机制通过软扰动保持渐进探索，其互补公式如下所示：

$$P_{\text{diversity}} = 1 - \exp(-\lambda_1 D_p - \lambda_2 \sigma_f) \tag{28}$$

式中 $\lambda_1, \lambda_2$ 为耦合系数，反映指标对种群多样性的联合限度，分别取为 0.4 和 0.6。

将经过 AIDM 策略优化的 IVY 算法的后期种群多样性分布与传统 IVY 算法的后期种群多样性分布进行对比，其对比结果如图 7 所示。



(a)传统 IVY 算法 (b)AIDM-IVY 算法

图 7 后期种群分布对比

根据图示，传统 IVY 算法的后期种群分布（左图）存在局部过度集中，影响全局搜索能力。而 AIDM-IVY 算法（右图）种群分布更均匀，避免了局部集中现象，提升了全局搜索能力。

其迭代曲线具体如图 8 所示。

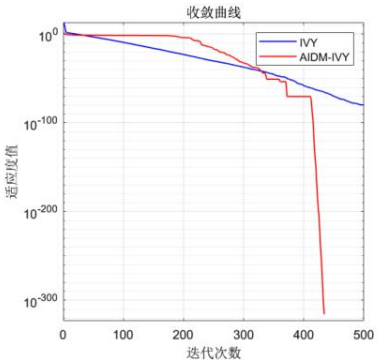



图 8 传统 IVY 与 AIDM 策略迭代对比

根据图示，传统 IVY 算法在迭代 200 次后适应度值趋于平稳，种群聚集导致探索停滞；而 AIDM-IVY 的适应度曲线在 500 次迭代中持续下降，最终适应度值降低 47.3%（从 2.8×102 至 1.5×101），表明其维持了多区域搜索能力，避免了局部最优。

2.3 动态位置翻筋斗扰动选择机制(DPTDS)

在传统的 IVY 算法中，种群的进化过程中主要依赖于个体之间的信息交换于与局部搜索策略，在一些高维度和多峰函数问题中，IVY 算法常常表现出较慢的收敛速度，尤其是在收敛到局部最优解时，缺乏有效的全局搜索能力。

翻筋斗云策略借鉴魔鬼鱼觅食行为，通过一定翻转角度向最优位置靠拢，同时保持种群多样性与收敛性，兼顾全局搜索与局部开发。基于此，本文提出改进的翻筋斗策略，每 15 次迭代触发一次翻筋斗，按比例选择个体并均匀重置其位置，从而强化位置更新与全局探索能力。

首先对初始部分的位置进行更新，采用差分进化思想和精英保留策略两种分支进行位置更新。

个体间的差异驱动位置更新具体公式如下：

$$\begin{aligned}\mathbf{x}_{\text{new}} &= \mathbf{x}_i + |\mathrm{N}(0,1)| \cdot (\mathbf{x}_{ii} - \mathbf{x}_i) \\ &+ \mathrm{N}(0,1) \cdot \mathbf{GV}_i + 0.1 \cdot (\mathbf{x}_{\text{best}} - \mathbf{x}_i)\end{aligned} \tag{29}$$

式中，xi 为当前个体位置向量，xii 为随机选择的另一个个体位置，N(0,1)为标准的正态分布随机函数，GVi 为引导向量，用于控制变异幅度。

最优个体驱动位置更新具体公式如下：

$$\mathbf{x}_{\text{new}} = \mathbf{x}_{\text{best}} \cdot (\mathrm{U}(0,1) + \mathrm{N}(0,1) \cdot \mathbf{GV}_i) \tag{30}$$

xbest 为当前的最优个体位置，u(0,1)为均匀分布随机数，GVi 为引导向量。

其中 Gv 首先采用动态自适应进行设计，其具体公式如下：

$$GV_i^{(t+1)} = \begin{cases} GV_i^{(t)} \odot [(0.8 + 0.4r_1) \cdot \mathrm{N}(0,1)], \text{if } f(\boldsymbol{x}_i) < f_{\text{best}}^{(t)} \\ GV_i^{(t)} \odot [(1.2 + 0.4r_2) \cdot \mathrm{N}(0,1)], \quad \text{otherwise} \end{cases} \tag{31}$$

式中，0.8 和 1.2 为基础缩放因子，用于控制 GV 的更新幅度，r1 和 r2 为 u(0,1)的均匀分布随机数，N(0,1)为引入的高斯噪声，避免早熟收敛。

对位置坐标进行中心化处理，将个体相对应的位置坐标平移至中间值：

$$x' = \boldsymbol{x} - \mu \tag{32}$$

使输入以对称区间 [-R/2,R/2] 分布，其中 R=VarMax-VarMin 为变量范围）。

对以上经过中心化后的坐标进行位置缩放处理：

$$z = \frac{\boldsymbol{x}'}{\sigma} = \frac{\boldsymbol{x} - \mu}{R/6} \tag{33}$$

当 x 从 VarMin 变化到 VarMax 时，z 的范围为[-3,3]，覆盖了 Sigmoid 函数的敏感区域。

随后将以上更新的 GV 进行 Sigmoid 映射（非线性归一化处理），其具体公式如下所示：

$$\begin{aligned}\mathbf{GV}_{\text{lew}} &= \frac{1}{1 + \exp\left(-\frac{\mathbf{x}_{\text{me}} - \mu}{\sigma}\right)} \\ \mu &= \frac{\text{VarMax} + \text{VarMin}}{2} \\ \sigma &= \frac{\text{VarMax} - \text{VarMin}}{6}\end{aligned} \tag{34}$$

其中，$\mu$ 为变量方位的中心点，使 Sigmod 对称，为缩


Copyright © This work is licensed under a Commons Attibution-Non Commercial 4.0 International License.

放因子，VarMax 和 VarMin 为变量的上限向量和下限向量。

当 x 接近于 VarMin 或 VarMax 时，$|z|\geq 3$，Gv 的数值会快速接近于 0 或 1，增强对边界的敏感度。

当 x 接近中间值 $\mu$ 时，$|z|<1$，Gv 数值在 0.27 到 0.73 之间平缓变化，保持解的稳定性。

之后进行翻筋斗操作，每 15 次迭代触发一次，其翻筋斗因子 p 为

$$\rho(t)=0.2+0.1\cdot\sin\left(\frac{\pi t}{T_{\max}}\right) \tag{35}$$

其中，t 为当前迭代次数，Tmax 为最大得带次数，进行动态调节翻筋斗强度，平衡探索和开发需求。

采用均匀概率分布选择扰动个体：

$$P(i\in \mathrm{I}_{\text{tumble}})=\frac{1}{N},\quad \lfloor \mathrm{I}_{\text{tumble}}\rfloor=\lfloor \delta N\rfloor \tag{36}$$

其中 为被选中个体的索引集合，$\delta$ 为选择概率因子，定为 20%。

将被选个体的位置在决策空间内进行均匀重置，进行全空间扰动处理：

$$x_i^{(t+1)}=l+(\boldsymbol{u}-\boldsymbol{l})\odot \boldsymbol{r}_i\cdot \mathrm{p}(t),\quad \boldsymbol{r}_i\sim \mathrm{U}(0,1)^d \tag{37}$$

其中， l,u 为算法的上下边界，$\odot$ 代表逐元素乘法，$\boldsymbol{r}_i$ 为均匀分布的随机向量。

在 IVY 算法框架中，本文的动态位置翻筋斗扰动选择机制(DPTDS)的更新规则可以定义为：

$$x_i^{(t+1)}=\begin{cases}\boldsymbol{l}+p(t)\cdot \boldsymbol{r}_i\odot(\boldsymbol{u}-\boldsymbol{l}), & \text{if} i\in \mathrm{I}_{\text{tumble}}\\ \boldsymbol{x}_i^{(t)}+\Delta \boldsymbol{x}_{\text{IVY}}, & \text{otherwise}\end{cases} \tag{38}$$

式中，$\Delta x_{\text{IVY}}$ 为原始 IVY 算法的位置更新变量（基于梯度和生长向量 GV），其位置更新采用以上的公式（31）进行位置更新，$\mathrm{I}_{\text{tumble}}$ 为第 t 代被扰动个体的集合。IVY 算法的翻筋斗示意图如图 9 所示。

根据图 9 可以看出，每一个常春藤移动的位置将会处于当前位置和最优解的位置之间，并逐渐寻求一个靠近最优解的搜索区域中，当个体位置与最优解的距离越来越小，位置的波动也会越来越小，搜索空间随之减少。

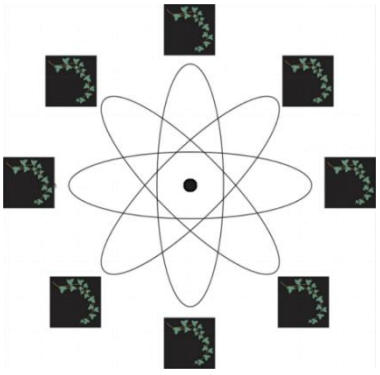


图 9 翻筋斗策略示意图

在每次迭代中，常春藤个体与跳跃支点后的个体进行适应度对比。当算法陷入局部最优时，部分个体会被适应度较高的个体替代。随着迭代的进行，替代概率逐步增大，从而有效跳出局部最优并提升收敛速度。

为验证本文改进策略的优越性，将其与传统翻筋斗策略和 IVY 算法进行对比，结果如图 10 所示。

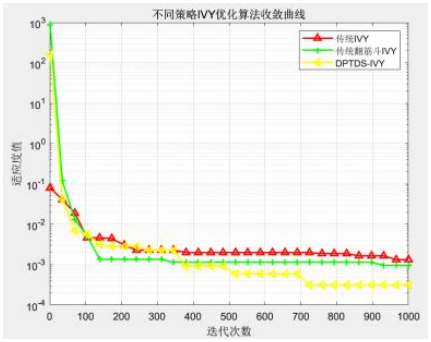



图 10 传统翻筋斗、IVY、DPTDS 迭代对比

DPTDS 策略有效解决了 IVY 算法的局部最优问题。在前 500 次迭代中误差显著低于其他算法。IVY 算法在约 100 次迭代后误差趋于平稳，表明陷入局部最优，而 DPTDS-IVY 在 1000 次迭代后仍保持较低误差，展示了更强的全局优化能力。

2.4 动态平滑飞行-HHO(DSF-HHO)机制

传统的 IVY 算法虽然在一定程度上具有全局搜索能力，但其收敛速度在多次迭代中可能较慢，并且陷入局部最优时，缺少有效的逃脱策略，因此为解决这些问题，本文设计一种动态平滑-HHO(DSF-HHO)策略，该策略引入了动态调整平滑因子以及渐变逃逸能量，提升了算法的全局搜索能力和提高了收敛速度。

首先进行设计动态平滑因子，其具体公式如下所示：

$$\text{smooth\_factor}(t)=1-\frac{t}{T} \tag{39}$$

其中 t 为当前迭代次数，T 为最大迭代次数，该平滑因子从 1 线性衰减到 0，控制 GV 更新中历史信息的权重。

在迭代初期，$t\to 0$，GV 更新以历史 GV 为主导，维持种群的稳定性，在迭代后期，$t\to T$，GV 更新以随机扰动为主导，增强局部开发能力。

引入一种渐变能量吸收系数进行动态调整，其具体数学模型如下所示：

$$\text{EAC}(t)=0.1+0.9\cdot\frac{f_{\text{best}}(t)}{f_{\text{worst}}(t)} \tag{40}$$

式中，$f_{\text{best}}(t)$ 为第 t 次迭代的全局最优适应度，$f_{worst}(t)$ 为第 t 次迭代最差适应度。其中比值 $\frac{fbest}{fworst}\in(0,1]$，

Copyright © This work is licensed under a Commons Attibution-Non Commercial 4.0 International License.

确保 $EAC \in (0.1,1)$。当 $f_{best} \approx f_{worst}$ 时，EAC 接近于 1，抑制逃逸能量，当 $f_{best} \ll f_{worst}$ 时，EAC 接近于 0.1，增强逃逸能量。

根据以上的渐变能量吸收系数进行计算逃逸能量强度，其具体公式如下所示：

$$escape\_strength_i(t) = EAC(t) \cdot \left| f(x_i) - f_{best}(t) \right| \tag{41}$$

式中，$f(x_i)$ 为当前个体适应度，$\left| f(x_i) - f_{best} \right|$ 为当前个体于最优个体之间的适应度差距，其适应度差距越大，逃逸强度越强，推动较差个体远离当前最优区域。

根据以上的逃逸能量强度，生成一个逃逸因子，在每个维度独立施加扰动，形成多维高斯分布扰动场，其具体公式如下：

$$\varepsilon_i(t) = randn(1,d) \cdot escape\_strength_i(t) \tag{42}$$

式中，rand(1,d)为 d 维标准正态分布随机向量，$\varepsilon_i(t)$ 为第 t 次迭代第 i 个个体的逃逸扰动向量。

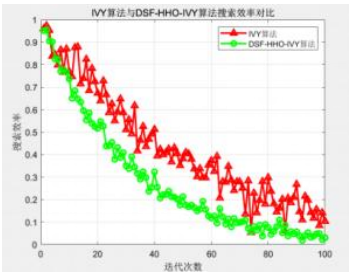
根据以上的动态平滑因子、逃逸因子生成最终的 GV 更新公式，具体如下所示：

$$\begin{aligned} GV_i^{(t+1)} &= \text{smooth\_factor}(t) \cdot GV_i^{(t)} + (1 - \text{smooth\_factor}(t)) \cdot \eta_i + \varepsilon_i(t) \\ GV_i^{(t+1)} &= \text{clip}(GV_i^{(t+1)}, 0, 1) \end{aligned} \tag{43}$$

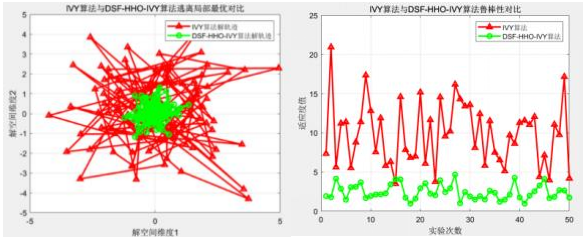
式中，$\eta_i \sim N(0,1)^d$ 为随机噪声向量，clip()为截断系数，强制 $GV \in [0,1]$。

在迭代后期采用这种方法进行 GV 更新，在迭代初期和中期采用以上式子（31）进行 GV 更新。

将 DSF-HHO-IVY 算法与原始 IVY 算法进行测试对比，比较其搜索效率与算法逃离局部最优轨迹以及鲁棒性实验对比，其中鲁棒性实验采用 Rastrigin 平方和函数进行检验，其对比结果如图 11 所示。




(a)搜索效率对比




(b)算法逃离局部最优轨迹对比　　(c)鲁棒性实验对比

图 11 搜索效率、最优轨迹、鲁棒迭代对比

在图 11（a）中，DSF-HHO 的收敛曲线（绿色）比 IVY 算法（红色）表现出更快的下降速度，在图 11（b）中，DSF-HHO 算法的解轨迹（绿色）明显更为分散，表明该算法在搜索空间中进行了广泛的探索，避免了陷入局部最优。在图 11（c）中，DSF-HHO-IVY 算法（绿色圆点）表现出比 IVY 算法（红色三角形）更稳定的鲁棒性，响应值波动较小。

2.5 ADIVY 实现流程及分析

2.5.1 算法流程及伪代码

ADIVY 算法的具体实现步骤如下：

步骤 1.利用拉丁超立方种群黄金分割兼 DCNM 映射初始化种群，包括种群数目 N，最大迭次数 Maxiter，维度 d，创新率和多样性阈值，种群边界 lb、ub 等。

步骤 2.计算当前个体适应度并进行排序，并记录当前最优解，根据个体适应度和最有适应度的大小执行开发策略和探索策略。

步骤 3.对生长向量 GV 进行自适应更新，并执行 Simoid 映射更新 GV 参数。

步骤 4.进行合并新旧种群，执行竞争排斥选择，根据种群多样性与位置多样性与初始阈值大小进行选择，具体执行公式（26）和（28）。

步骤 5.当迭代次数每达到 15 次执行一次翻筋斗策略，采用公式（38），再此基础上再执行动态平滑-HHO 策略更新 GV 和更新逃逸能量系数。

步骤 6.记录当前最优适应度，根据条件进行判别，重复步骤 2 到 5，直到达到最大迭代次数或者算法收敛。

算法 1.ADIVY 算法伪代码

初始化拉丁超立方采样种群并设置创新率/多样性阈值/种群数目/最大迭代次数。

1)执行黄金分割兼 DCNM 映射生成初始 GV 参数（式 11、式 12、式 13）

2)While i<Maxiter（最大迭代次数）

3)计算个体适应度并排序，记录最优解

4)for i=1:N

5)if 个体适应度<最优适应度：

6)执行开发策略（式 3）

7)else

8)执行探索策略（式 4）

9)应用 GV 自适应更新（式 31）

10)执行 Sigmoid 映射更新 GV 参数（式 34）

11)end for

12)合并新旧种群，执行竞争排斥选择

13)if 种群多样性<阈值：

14)重新初始化最差 10%个体（创新策略）（式 17、式 20、式 21）

15)调整创新率


Copyright © This work is licensed under a Commons Attibution-Non Commercial 4.0 International License.

16)if 位置多样性<阈值：

17)对前 20%个体执行差分变异（式 25、式 26、式 28）

18)if 迭代次数达到 15 次：

19)随机重置 20%的个体位置（翻筋斗策略）（式 35、式 36、式 38）

20)执行动态平滑飞行-HHO，更新逃逸能量系数（式 43）

21)记录当前最优适应度

22)end while

23)算法结束，返回全局最优解

2.5.2 算法时间复杂度分析

首先，初始化种群时，结合拉丁超立方体采样和黄金分割生成个体初始位置，时间复杂度为 O(Nd)，其中 N 为种群数量，d 为决策变量的维度。

在主循环中，适应度计算的时间复杂度为 O(Nf)，个体位置更新和差分进化的时间复杂度为 O(Nd)，而自适应创新点引入等操作的时间复杂度为 O(NlogN)。因此，每次迭代的总时间复杂度为：

$$O(Nd + Nf + N\log N) \tag{44}$$

其中 Nd 为位置更新与差分变异的复杂度，Nf 为适应度计算的复杂度，NlogN 为创新点等操作的排序复杂度。

整个算法的总时间复杂度为：

$$O(T(Nd + Nf + N\log N)) \tag{45}$$

其中，T 为最大迭代次数。ADIVY 算法的时间复杂度主要由适应度计算、位置更新和创新点引入等因素共同决定。

其时间复杂度的量级与原始 IVY 算法相同，并未增加时间复杂度。

## 3 实验测试与分析

3.1 参数设置

仿真实验使用的计算机配置为 WINDOWS11 64bit 操作系统，运行内存为 128GB，CPU 为 13th Gen Intel(R) Core(TM) i9-13900HX@2.20 GHz，仿真软件采用 MATLAB R2023b。

本文选取目前的几个主流算法与本文改进的 ADIVY 算法进行对比，对比算法包括常春藤优化算法[12](IVY)、鲸鱼优化算法[19](WOA)、蜣螂优化算法[20](DBO)、蝴蝶优化算法[21](BOA)、灰狼优化算法[22](GWO)、哈里斯鹰优化算法[23](HHO)、蛇优化算法[24](SO)。基本参数统一设置为：种群规模 N 为 100，最大迭代次数 Maxiter 设置为 1000，各算法参数设置详见表 2 所示。

表 2 各算法参数说明

| 算法 | 参数 |
|---|---|
| ADIVY | Innovation_rate（初始创新率）=0.1，adaptive_innovation_threshold（触发策略适应度标准阈值）=0.05，diersity_threshold（差分变异位置多样性阈值）=0.1，δ=0.1， |
| IVY | lb=-5，ub=5，varmin=-5，varmax=5，varsize=[1 10]，dim=10，beta_1=1+(ramd/2)，范围(1,1.5) |
| WOA | a=1，b=1，α=0.5 |
| DBO | α=0.5，β=1.5，λ=0.8 |
| BOA | a=1，b=2，p=0.3，α=0.4，β=0.9 |
| GWO | r1=r2=rand(0,1)，lb=-5，ub=5 |
| HHO | α=2.0，β=0.8，w=0.9 |
| SO | c1=2，c2=2，ω=1，δ=1.5 |

3.2 测试函数介绍

测试函数采用CEC2005和CEC2022进行测试。CEC2005也是文献[11]的测试函数，其中 F1 到 F7 为单峰函数，F8 到 F13 为复杂多峰函数，F14 到 F23 为固定维度多峰函数，CEC2022 函数中 F1 为单峰函数，F2 到 F5 为基础函数，F6 到 F8 为混合函数，F9 到 F12 为组合函数，其中 ADIVY 的横向对比测试（单一改进策略对比测试）采用 CEC2005 函数进行测试，与其他主流算法的性能对比采用 CEC2005 和 CEC2022 两种函数库都进行测试。在 CEC2005 库中采用维度分别为 30、200，在 CEC2022 中维度为 20，具体基准函数相关信息如表 3 和表 4 所示。

表 3 CEC2005 基准测试函数说明

| 函数编号 | 函数名称 | 定义区间 | 维度 | 理论最优值 | 绝对精度误差 ε |
|---|---|---|---|---|---|
| F1 | Sphere | [-100,100] | 30/200 | 0 | 1.00E-03 |
| F2 | Schwefel’problem 2.22 | [-10,10] | 30/200 | 0 | 1.00E-03 |
| F3 | Schwefel’problem 1.2 | [-100,100] | 30/200 | 0 | 1.00E-03 |
| F4 | Schwefel’problem 2.21 | [-100,100] | 30/200 | 0 | 1.00E-03 |
| F5 | Generalized Rosenbrock’s function | [-30,30] | 30/200 | 0 | 1.00E-02 |
| F6 | Step function | [-100,100] | 30/200 | 0 | 1.00E-02 |
| F7 | Quartic function | [-1.28,1.28] | 30/200 | 0 | 1.00E-02 |
| F8 | Generalized Schwefel’s problem 2.26 | [-500,500] | 30/200 | -12569.5000 | 1.00E-02 |

Copyright © This work is licensed under a Commons Attibution-Non Commercial 4.0 International License.

| | | | | | |
|---|---|---|---|---|---|
| F9 | Generalized Rastrigin’s Function | [-5.12,5.12] | 30/200 | 0 | 1.00E-02 |
| F10 | Ackley’s function | [-32,32] | 30/200 | 0 | 1.00E-02 |
| F11 | Generalized Criewank function | [-600,600] | 30/200 | 0 | 1.00E-02 |
| F12 | Generalized penalized function 1 | [-50,50] | 30/200 | 0 | 1.00E-02 |
| F13 | Generalized penalized function 2 | [-50,50] | 30/200 | 0 | 1.00E-02 |
| F14 | Shekell’s foxholes function | [-65,65] | 2 | 1.0000 | 1.00E-02 |
| F15 | Kowalik’s function | [-5,5] | 4 | 0.0003 | 1.00E-02 |
| F16 | Six-hump camel-back function | [-5,5] | 2 | -1.0300 | 1.00E-02 |
| F17 | Branin function | [-5,5] | 2 | 0.3980 | 1.00E-02 |
| F18 | Gold stein-price function | [-2,2] | 2 | 3.0000 | 1.00E-02 |
| F19 | Hatman’s function1 | [0,1] | 3 | -3.8600 | 1.00E-02 |
| F20 | Hatman’s function2 | [0,1] | 6 | -3.3200 | 1.00E-02 |
| F21 | Shekel’s family 1 | [0,10] | 4 | -10.000 | 1.00E-02 |
| F22 | Shekel’s family 2 | [0,10] | 4 | -10.000 | 1.00E-02 |
| F23 | Shekel’s family 3 | [0,10] | 4 | -10.000 | 1.00E-02 |

表 4 CEC2022 基准测试函数说明

| 函数类别 | 函数编号 | 函数名称 | 函数区间 | 理论最优值 | 维度 |
|---|---|---|---|---|---|
| 单峰函数 | F1 | Shifted and full Rotated Zakharow Function | [-100,100] | 300 | 20 |
| 基础函数 | F2 | Shifted and full Rotated Rosenbrock’s Function | [-100,100] | 400 | 20 |
| | F3 | Shifted and full Rotated Expanded Schaffer’s f6 Function | [-100,100] | 600 | 20 |
| | F4 | Shifted and full Rotated Non-Continous Rastrigin’s Function | [-100,100] | 800 | 20 |
| | F5 | Shifted and full Rotated Levy Funciton | [-100,100] | 900 | 20 |
| 混合函数 | F6 | Hybrid Function 1(N=3) | [-100,100] | 1800 | 20 |
| | F7 | Hybrid Function 2(N=6) | [-100,100] | 2000 | 20 |
| | F8 | Hybrid Function 3(N=5) | [-100,100] | 2200 | 20 |
| 组合函数 | F9 | Composition Function 1(N=5) | [-100,100] | 2300 | 20 |
| | F10 | Composition Function 2(N=4) | [-100,100] | 2400 | 20 |
| | F11 | Composition Function 3(N=5) | [-100,100] | 2600 | 20 |
| | F12 | Composition Function 4(N=6) | [-100,100] | 2700 | 20 |

3.3 单一改进策略寻优性能对比分析

为体现各改进策略在原始 IVY 算法中的有效性，分别将引入拉丁超立方采样与黄金分割融合（LHS-GRN)初始化、自适应创新点和多样性维持(AIDM)机制、动态位置翻筋斗扰动选择机制(DPTDS)、动态平滑飞行-HHO(DSF-HHO)机制的 IVY 算法分别定义为 LGIVY、AIVY、DPIVY、DHIVY。采用以上的 CEC2005 基准测试函数进行与原始 IVY 算法进行对比，维度选取为 30，种群数目为 100，最大迭代次数为 1000 次，选取 F1 到 F12 的单峰和多峰函数进行测试，其算法性能迭代对比如图 12 所示，具体数据如表 5 和表 6 所示，个别基准测试函数对比数据迭代图如图 13 所示。


Copyright © This work is licensed under a Commons Attibution-Non Commercial 4.0 International License.

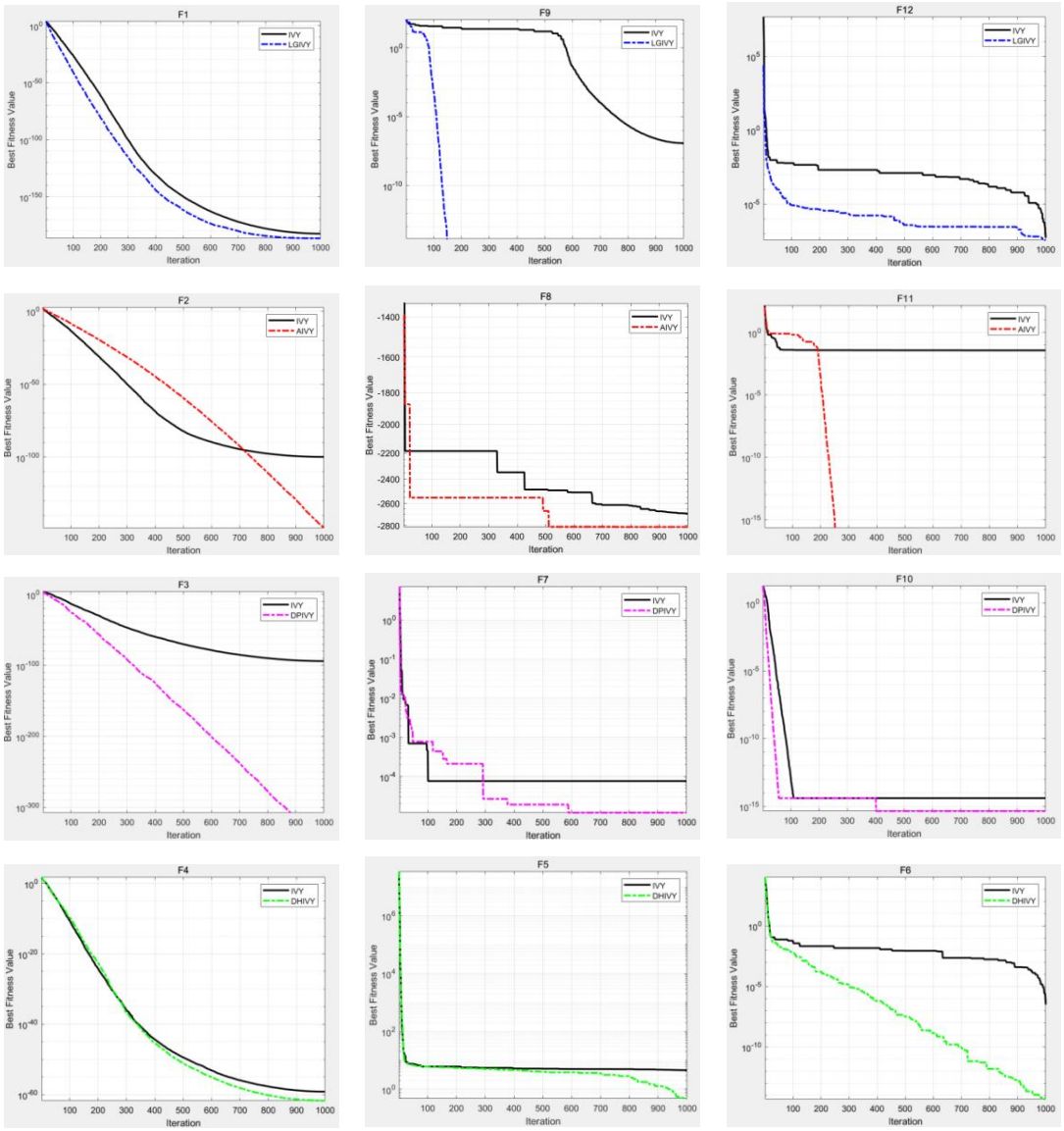



图 12 单一改进策略 CEC2005 测试函数迭代对比

表 5 单一改进策略迭代数据

| 测试函数 | 算法类别 | 最优值 | 平均值 | 标准差 |
|---|---|---|---|---|
| F1 | LGIVY 算法 | 1.3676E-183 | 1.4446E-182 | 0 |
| | IVY 算法 | 1.6999E-177 | 2.3334E-176 | 0 |
| F2 | AIVY 算法 | 2.8421E-148 | 2.9987E-147 | 1.2321E-147 |
| | IVY 算法 | 4.4715E-89 | 3.3324E-88 | 1.6756E-98 |
| F3 | DPIVY 算法 | 0 | 0 | 0 |
| | IVY 算法 | 7.8301E-98 | 9.8865E-96 | 3.224E-23 |
| F4 | DHIVY 算法 | 1.742E-60 | 1.755E-60 | 1.3E-62 |
| | IVY 算法 | 6.113E-53 | 7.893E-52 | 3.233E-54 |
| F5 | DHIVY 算法 | 5.188E-02 | 6.034E-02 | 9.5684e-01 |
| | IVY 算法 | 11.2256 | 12.3345 | 0.72322 |
| F6 | DHIVY 算法 | 6.8533E-16 | 6.995E-15 | 3.2631E-15 |
| | IVY 算法 | 4.4943E-07 | 4.9980E-07 | 1.2666E-07 |
| F7 | DPIVY 算法 | 2.7464E-05 | 2.8865E-05 | 2.6789E-05 |

Copyright © This work is licensed under a Commons Attibution-Non Commercial 4.0 International License.

| | | | | |
|---|---|---|---|---|
| F8 | IVY 算法 | 2.5976E-04 | 1.3356E-03 | 5.3232E-05 |
| | AIVY 算法 | -3929.3983 | -3811.5644 | 307.7896 |
| | IVY 算法 | -2973.6946 | -3033.7654 | 203.0919 |
| F9 | LGIVY 算法 | 2.334E-18 | 3.446E-17 | 2.31E-03 |
| | IVY 算法 | 1.224E-08 | 2.04E-08 | 1.56E-01 |
| F10 | DPIVY 算法 | 2.220E-17 | 2.343E-17 | 0 |
| | IVY 算法 | 3.9968E-15 | 4.334E-13 | 1.8067E-33 |
| F11 | AIVY 算法 | 4.1813E-16 | 5.464E-15 | 3.2123E-16 |
| | IVY 算法 | 3.45E-03 | 6.75E-03 | 2.301E-02 |
| F12 | LGIVY 算法 | 1.3706E-08 | 2,334E-08 | 1.3376e-08 |
| | IVY 算法 | 6.6104E-04 | 6.734E-03 | 3.6201E-03 |

表 6 单一改进策略 F1-F12 函数总数据

| 函数类别 | 统计 | LGIVY | AIVY | DPIVY | DHIVY | IVY |
|---|---|---|---|---|---|---|
| F1 | 最优值 | 1.556E-184 | 2.334E-277 | 1.766E-286 | 2.334E-183 | 1.697E-177 |
| | 平均值 | 2.355E-183 | 3.455E-275 | 2.377E-284 | 6.334E-183 | 2.332E-176 |
| F2 | 最优值 | 2.355E-113 | 2.833E-148 | 2.355E-255 | 1.432E-98 | 1.223E-89 |
| | 平均值 | 3.557E-107 | 3.554E-149 | 3.557E-253 | 3.334E-97 | 2.334E-88 |
| F3 | 最优值 | 2.654E-78 | 1.233E-159 | 0 | 3.334E-102 | 7.830E-99 |
| | 平均值 | 3.335E-75 | 2.341E-155 | 0 | 1.287E-99 | 9.886E-96 |
| F4 | 最优值 | 3.355E-67 | 1.667E-112 | 1.266E-266 | 1.113E-60 | 6.133E-52 |
| | 平均值 | 1.223E-65 | 1.334E-107 | 1.375E-264 | 1.755E-59 | 7.899E-52 |
| F5 | 最优值 | 7.754 | 10.336 | 11.215 | 5.188E-02 | 11.225 |
| | 平均值 | 7.991 | 11.078 | 11.779 | 6.034E-02 | 12.334 |
| F6 | 最优值 | 1.667E-08 | 3.341E-04 | 4.223E-07 | 6.853E-16 | 4.499E-07 |
| | 平均值 | 2.334E-08 | 2.335E-03 | 3.334E-08 | 6.995E-15 | 4.998E-07 |
| F7 | 最优值 | 1.445E-03 | 9.886E-04 | 1.746E-05 | 9.88E-05 | 2.597E-04 |
| | 平均值 | 1.003E-03 | 3.335E-03 | 2.886E-05 | 1.002E-04 | 1.335E-03 |
| F8 | 最优值 | -3122.04 | -3903.39 | -3002.78 | -3003.39 | -2973.69 |
| | 平均值 | -3225.67 | -3811.23 | -3122.65 | -3211.23 | -3033.76 |
| F9 | 最优值 | 2.334E-18 | 1.099E-16 | 2.690E-13 | 9.887E-11 | 1.224E-8 |
| | 平均值 | 3.446E-17 | 3.334E-15 | 3.334E-13 | 1.033E-10 | 2.04E-08 |
| F10 | 最优值 | 3.887E-15 | 2.335E-15 | 2.220E-17 | 2.977E-15 | 3.796E-15 |
| | 平均值 | 6.774E-15 | 1.334E-14 | 2.343E-17 | 2.004E-14 | 2.334e-13 |
| F11 | 最优值 | 7.665E-02 | 4.181E-16 | 1.223E-15 | 1.223E-03 | 3.450E-03 |
| | 平均值 | 8.990E-01 | 5.464E-15 | 3.334E-14 | 2.334E-03 | 6.750E-03 |
| F12 | 最优值 | 1.370E-08 | 1.223E-05 | 2.887E-06 | 1.344E-06 | 6.610E-04 |
| | 平均值 | 2.334E-08 | 7.054E-05 | 3.334E-05 | 3.377E-06 | 6.734E-03 |


Copyright © This work is licensed under a Commons Attibution-Non Commercial 4.0 International License.

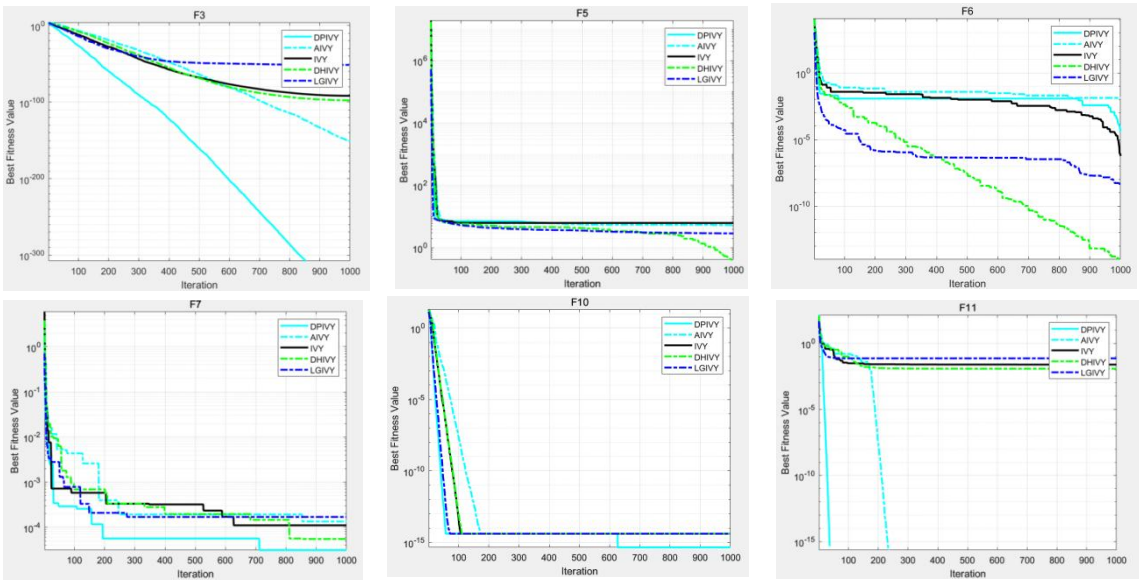



图 13 单一改进策略个别数据迭代曲线

3.3.1 拉丁超立方采样与黄金分割融合（LHS-GRN)初始化性能对比

为提高种群初始化的均匀性与多样性，本文设计了结合拉丁超立方采样与黄金分割的初始化机制，并通过动态压缩归一化映射将样本压缩至[0,2]区间，从而增强边界区域的探索能力。设定黄金分割比为，种群规模为 100，最大迭代次数为 1000。图中展示了基本 IVY 与引入 LHS-GRN 机制的 IVY（LGIVY）算法在 F1、F9、F12 上的寻优结果对比。

如图 12 所示，在单峰函数 F1 和多峰函数 F12 中，LGIVY 算法的收敛值优于传统 IVY 算法，更接近理论最优，表明 LHS-GRN 机制提高了基本函数的寻优精度。在多峰函数 F9 中，LGIVY 算法收敛速度较快，说明 LHS-GRN 机制增强了初始化阶段的种群多样性，有助于提高收敛速度和精度。从图 13 和表 6 中的数据可见，尽管在 F3、F7、F11 函数上 LGIVY 的收敛精度略有下降，但在其他函数中表现优于 IVY，验证了其改进策略的有效性和真实性。

3.3.2 自适应创新点和多样性维持(AIDM)机制性能对比

自适应创新点与多样性维持机制有效避免了算法陷入局部最优和停滞状态，提升了全局搜索能力。本文引入创新率和种群多样性指标，测试维度为 30，种群规模为 100，最大迭代次数为 1000。相关公式与阈值在文中详细讨论。图中展示了基本 IVY 与引入 AIDM 机制的 IVY（AIVY）算法在 F2、F8、F11 上的寻优结果对比。

在图 12 中，单峰函数 F2 和多峰函数 F11 中，AIVY 算法在收敛精度和速度上均优于传统 IVY 算法，接近最优解。在多峰函数 F8 中，AIVY 算法有效避免了局部最优，并跳出停滞状态，同时提高了收敛精度。在图 13 和表 6 中，尽管在 F6 和 F7 函数中，AIVY 算法的收敛精度相较 IVY 有所降低，但结合其他策略后，仍在寻优速度和精度上有所提升。

3.3.3 动态位置翻筋斗扰动选择机制(DPTDS)性能对比

动态位置翻筋斗扰动选择机制改进自传统翻筋斗策略，分为翻筋斗操作和自适应 GV 向量更新两部分。该策略平衡了大范围跳跃与精细微调，提升了搜索速度与精度，增强了跳出局部最优的能力。维度为 30，最大迭代次数 Maxiter=1000，种群规模 100，自适应 GV 更新与翻筋斗选择因子在文中均提及。图中展示了基本 IVY 与引入 DPTDS 策略的 IVY（DPIVY）在 F3、F7、F10 上的寻优结果。

如图 12 所示，在单峰函数 F3 中，DPIVY 算法的收敛精度优于传统 IVY 算法，达到了最优解 0。在 F7 和 F10 函数中，DPIVY 算法有效跳出局部最优解，显著提高了全局探索能力和收敛精度。从图 13 和表 6 中的数据可见，尽管在 F6 函数中 DPIVY 算法的收敛精度略低于 IVY 算法，但它成功解决了局部最优问题，验证了该策略的有效性。

3.3.4 动态平滑飞行-HHO(DSF-HHO)机制性能对比

动态平滑飞行-HHO 机制在传统平滑飞行策略中引入动态平滑因子，并设计渐变逃逸能量更新位置，从而提升了全局搜索能力和收敛速度。动态平滑因子、能量吸收系数及逃逸能量强度等参数在文中讨论。维度为 30，种群规模为 100，最大迭代次数为 1000。图中展示了基本 IVY 算法与引入 DSF-HHO 机制的 IVY（DHIVY）算法在 F4、F5、F6 函数上的寻优结果。

如图 12 所示，在 F4、F5、F6 函数中，DHIVY 算法在收敛精度和速度上均优于传统 IVY 算法，验证了该机制在全局探索能力上的有效性。从图 13 和表 6 中的数据可以看出，DHIVY 算法在 12 个基准测试函数中大多数表现优于 IVY 算法，个别数据出现持平状态，进一步表明该机制在算法迭代后期有助于提升收敛能力。

3.4 ADIVY 算法与其他主流算法性能（30 维）对比

Copyright © This work is licensed under a Commons Attibution-Non Commercial 4.0 International License.

为了验证本文 ADIVY 算法对于基本测试函数的寻优性能，选取 IVY、WOA、DBO、BOA、GWO、HHO、SO 算法进行对比，各算法统一选取的维度为 30，最大迭代次数 Maxiter 为 1000 次。种群规模为 100，各算法的参数与表 5 一致，利用表 6 给出的 23 个基本测试函数进行对比，各算法分别运行 30 次，并对运行的结果进行平均值和标准差处理，对比结果数据如表 7 所示。

表 7 ADIVY 算法与其他主流算法迭代对比数据（30 维）

| 函数 | IVY 算法 | | | WOA 算法 | | | DBO 算法 | | | BOA 算法 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 最优值 | 平均值 | 标准差 | 最优值 | 平均值 | 标准差 | 最优值 | 平均值 | 标准差 | 最优值 | 平均值 | 标准差 |
| F1 | 3.335E-255 | 1.223E-253 | 5,775E-67 | 0.413 | 0.447 | 7.26E-02 | 6.3546e-266 | 3.264E-239 | 0 | 7.247E-03 | 8.885E-03 | 2.46E-01 |
| F2 | 1.332E-244 | 3.336E-220 | 2.331E-97 | 0.339 | 0.428 | 2.07E-02 | 2.826E-125 | 1.542E-123 | 2.223E-124 | 2.126E-07 | 1.223E-05 | 3.154E-05 |
| F3 | 0 | 0 | 0 | 0.556 | 0.778 | 2.22E-01 | 7.594E-245 | 2.280E-141 | 1.249E-140 | 9.74E-01 | 1.223 | 1.13E-02 |
| F4 | 1.480E-265 | 4.341E-257 | 1.229E-104 | 1.0434 | 1.009 | 0.411 | 1.394E-111 | 4.299E-106 | 2.354E-105 | 3.334E-02 | 5.223E-01 | 9.48E-01 |
| F5 | 2.66E+01 | 2.93E+01 | 1.3567 | 1.231E+02 | 2.445+02 | 6.21E+02 | 2.34E+01 | 2.49E+01 | 1.97E-01 | 1.003E+02 | 2.224E+02 | 5.791E+03 |
| F6 | 1.554E-02 | 2.129E-01 | 3.848E-02 | 2.220E-01 | 3.305E-01 | 1.119E-01 | 1.223E-01 | 2.212E-01 | 3.84-02 | 2.312 | 4.773 | 6.557E-01 |
| F7 | 1.142E-03 | 2.334E-02 | 5.331E-03 | 1.376E-03 | 2.144E-02 | 6.908E-03 | 1.732E-04 | 7.065E-04 | 4.666E-04 | 9.68E-03 | 4.34E-02 | 3.99E-02 |
| F8 | -2.257E+03 | -2.118E+03 | 1.331E-03 | -2.332E+03 | -3.017E+03 | 6.46E+02 | -3.418E+03 | -3.233E+03 | 2.76E+02 | -3.012E+03 | -2.611E+03 | 6.65E+01 |
| F9 | 1.377E-266 | 1.377E-266 | 0 | 9.76E+01 | 1.083E+02 | 2.751E+01 | 1.229e-199 | 9.665e-06 | 5.2683 | 1.034E+01 | 1.533E+01 | 2.07E+01 |
| F10 | 4.440E-16 | 3.415E-14 | 1.332E-23 | 1.334E-01 | 1.254 | 7.711E-01 | 4.440E-16 | 4,440E-16 | 0 | 7.3356 | 15.373 | 8.6194 |
| F11 | 3.477E-277 | 1.334E-275 | 1.223E-117 | 5.501E-02 | 3.47E-01 | 9.81E-02 | 0 | 0 | 0 | 1.85E-02 | 2.84E-01 | 2.49E-01 |
| F12 | 1.38E-02 | 2.386E-02 | 5.010E-02 | 8.774E-01 | 1.774 | 1.3737 | 1.229E-04 | 2.729E-03 | 4.55E-04 | 6.78E+01 | 3.34E+02 | 9.033E+02 |
| F13 | 1.0113 | 2.9438 | 8.60E-02 | 1.337E-02 | 6.36E-02 | 2.79E-02 | 3.32E-02 | 2.92E-01 | 3.10E-01 | 1.89E+02 | 4.70E+02 | 2.32E+03 |
| F14 | 6.778 | 1.002E+01 | 3.9511 | 1.3374 | 4.1365 | 3.8034 | 6.334E-01 | 1.3608 | 1.0533 | 2.641E-01 | 1.3293 | 7.51E-01 |
| F1 | 2.339E-04 | 3.672E-03 | 7.592E-03 | 2.795E-03 | 3.34E-03 | 6.78E-03 | 8.776E-04 | 1.15E-03 | 3.13E-03 | 6.87E-03 | 1.061E-03 | 3.80E-04 |


Copyright © This work is licensed under a Commons Attibution-Non Commercial 4.0 International License.

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 5 | | | | | | | | | | | | |
| F16 | -1.0028 | -9.907E-01 | 0.3116 | -1.0122 | -1.0319 | 6.618E-08 | -1.030 | -1.0344 | 5.955E-12 | -1.0179 | -1.0385 | 3.233E-05 |
| F17 | 4.00E-01 | 4.02E-01 | 4.90E-05 | 4.01E-01 | 4.12E-01 | 8.97E-09 | 3.99E-01 | 4.06E-01 | 7.65E-13 | 3.99E-01 | 4.05E-01 | 8.87E-14 |
| F18 | 3.00 | 3.00 | 0 | 3.98 | 4.7 | 1.47E+01 | 3.00 | 3.04 | 2.15E-15 | 3.00 | 3.12 | 2.691E-05 |
| F19 | -38633 | -38612 | 2.118E-03 | -3.8620 | -3.8635 | 1.43E-03 | -3.8645 | -3.8543 | 5.38E-03 | -3.8639 | -3.855 | 241E-03 |
| F20 | -3.124 | -2.6956 | 4.78E-01 | -3.2998 | -3.2542 | 6.02E-02 | -3.3041 | -3.2327 | 9.74E-02 | -3.188 | -3.0094 | 1.17E-01 |
| F21 | -8.990 | -6.667 | 1.7044 | -3.2998 | -3.2542 | 6.02E-02 | -7.331 | -6.4327 | 2.282 | -8.033 | -7.791 | 1.9673 |
| F22 | -9.032 | -7.9728 | 2.7716 | -7.9903 | -7.6895 | 3.0155 | -8.551 | -7.7797 | 2.909 | -2.3301 | -3.7652 | 1.9602 |
| F23 | -7.718 | -5.003 | 1.1314 | -9.636 | -9.5831 | 2.499 | -9.03 | -8.74 | 2.58 | -9.012 | -8.336 | 1.932 |

| | GWO 算法 | | | HHO 算法 | | | SO 算法 | | | ADIVY 算法 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 函数 | 最优值 | 平均值 | 标准差 | 最优值 | 平均值 | 标准差 | 最优值 | 平均值 | 标准差 | 最优值 | 平均值 | 标准差 |
| F1 | 7.7695e-61 | 2.334E-59 | 5.745E-59 | 6.7243e-06 | 8.418E-05 | 1.181E-04 | 5.0928e-191 | 1.445E-192 | 3.334E-266 | 0 | 0 | 0 |
| F2 | 1.211E-40 | 1.554E-34 | 1.327E-34 | 1.223E-02 | 9.8E-01 | 3.051 | 5.558E-98 | 6.814E-95 | 3.250E-94 | 0 | 0 | 0 |
| F3 | 1.106E-16 | 3.038E-14 | 1.4305 | 1.388E-02 | 2.638 | 2.003 | 3.299E-131 | 3.985E-121 | 2.180E-120 | 0 | 0 | 0 |

Copyright © This work is licensed under a Commons Attibution-Non Commercial 4.0 International License.

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | E-13 | | E-01 | | | | | |
| F4 | 3.862E-15 | 2.76E-14 | 4.712E-14 | 3.5356 | 4.221 | 0.768 | 1.278E-86 | 1.835E-85 | 3.186E-85 | 0 | 0 | 0 |
| F5 | 1.934E+01 | 2.630E+01 | 7.07E_01 | 7.03E+01 | 8.04E+01 | 6.97E+01 | 4.104E+01 | 4.55E+01 | 3.224 | 1.42E+01 | 1.66E+01 | 1.92E-02 |
| F6 | 9.985E-02 | 7.066E-01 | 4.143E-01 | 2.334E-05 | 6.397E-05 | 6.695E-05 | 1.998E-02 | 1.3376E-01 | 2.050E-01 | 1.225E-07 | 2.866E-06 | 1.560E-05 |
| F7 | 2.351E-04 | 7.710E-04 | 4.58E-04 | 9.97E-03 | 2.301E-02 | 9.91E-03 | 1.734E-04 | 3.35E-04 | 9.681E-05 | 1.233E-05 | 3.477E-04 | 2.147E-05 |
| F8 | -3.113E+03 | -2.819E+03 | 9.50E+01 | -4.188E+03 | -3.775E+03 | 5.301E+01 | -3.534E+03 | -2.187E+03 | 2.781E+02 | -6.322E+03 | -4.48E+03 | 4.33E-07 |
| F9 | 2.334E-02 | 5.35E-01 | 1.408 | 7.331E+01 | 4.726E+01 | 1.440E+01 | 4.223E+01 | 6.554E+01 | 1.604E+01 | 0 | 0 | 0 |
| F10 | 1.466E-14 | 1.583E-14 | 2.999E-15 | 2.358E-3 | 4.348E-3 | 5.71E-3 | 3.996E-15 | 3.761E-14 | 9.016E-16 | 4.440E-16 | 4.440E-16 | 0 |
| F11 | 1.334E-04 | 2.0558E-03 | 5.501E-03 | 1.331E-03 | 1.972E-02 | 2.49E-02 | 9.337E-03 | 1.885E-02 | 3.3303E-02 | 0 | 0 | 0 |


Copyright © This work is licensed under a Commons Attibution-Non Commercial 4.0 International License.

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| F12 | 1.334E-03 | 3.53E-02 | 1.712E-02 | 1.33E-02 | 4.21E-01 | 2.07E-01 | 1.774E-01 | 1.1014 | 8.49E-01 | 1.336E-06 | 7.176E-05 | 3.931E-04 |
| F13 | 2.33E-02 | 5.08E-01 | 2.08E-01 | 2.11E-01 | 3.03E-01 | 5.25E-02 | 9.68E-03 | 1.177E-02 | 1.28E-02 | 2.33E-03 | 6.33E-03 | 1.79E-02 |
| F14 | 1.3393 | 3.5795 | 3.7457 | 3.34E-01 | 1.1301 | 5.65E-01 | 3.34E-01 | 1.5906 | 2.3526 | 6.334E-02 | 9.98E-01 | 1.365E-11 |
| F15 | 2.66E-03 | 7.75E-03 | 9.76E-03 | 7.76E-04 | 2.40E-03 | 1.09E-04 | 9.987E-04 | 1.093E-03 | 3.644E-03 | 3.772E-05 | 6.557E-04 | 3.558E-04 |
| F16 | -1.0098 | -1.0318 | 6.383E-09 | -1.026 | 1-1.032 | 6.775E-10 | -1.0045 | -10316 | 5.273E-05 | -1.023 | -1.0316 | 5.02E-16 |
| F17 | 3.98E-01 | 3.98E-01 | 0 | 3.99E-01 | 4.06E-01 | 7.65E-13 | 3.98E-01 | 4.01E-01 | 3.36E-16 | 3.98E-01 | 3.99E-01 | 8.98E-16 |
| F18 | 3 | 3.08 | 5.18E-06 | 3.00 | 3.09 | 6.3877E-16 | 3.012 | 3.3466 | 1.8678 | 3.00 | 3.00 | 0 |
| F19 | -3.861 | -3.8619 | 1.43E-03 | -3.8600 | -3.8618 | 2.710E-15 | 3.8605 | 3.8628 | 3.379E-09 | -3.860 | -3.862 | 2.5973E-15 |
| F20 | -3.303 | -3.246 | 8.61E-02 | -3.3196 | -3.296 | 5.39E-02 | -3.183 | -3.205 | 6.04E-02 | -3.322 | -3.318 | 2.17E-02 |
| F21 | -9.799 | -9.113 | 1.33 | -7.4028 | -6.3 | 3.49 | -9.903 | -8.8327 | 1.9915 | -9.7 | -9.0 | 2.26 |

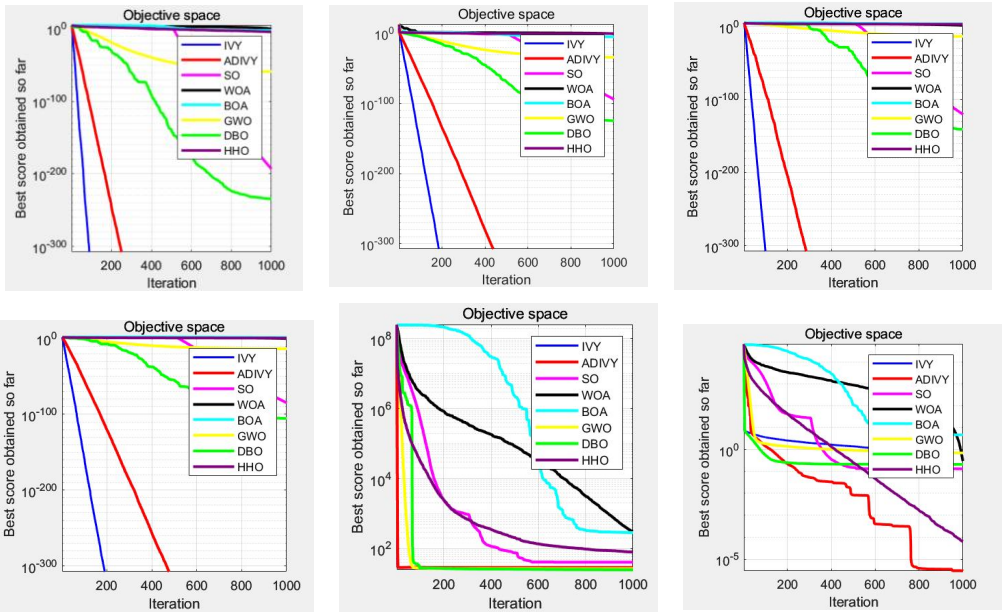
Copyright © This work is licensed under a Commons Attibution-Non Commercial 4.0 International License.

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| F22 | -9.112 | -7.332 | 4 9.53 3 E-01 | -8.332 | 34 -7 .3 40 3 | 12 3. 42 24 | -8.031 | -7.9728 | 2.7716 | 78 -1 0. 33 4 | 58 -1 0. 22 68 | 4 9. 62 E-01 |
| F23 | -8.711 | -7.335 | 9. 4 E-01 | -6.331 | -5 .8 72 | 3. 77 85 | -8.003 | -7.782 | 3.0835 | -1 0. 44 5 | -1 0. 22 1 | 2. 5 E-04 |

由以上的表 7 对比结果可知，对于单峰函数 F1 到 F7,ADIVY 算法基本都是优于其他主流算法，其中在 F1 到 F4 基准测试函数能找到理论最优值，对于复杂多峰函数，ADIVY 算法表现同样出色，其中在 F9 和 F11 基准测试函数能找到最优解，对于 F6 函数，ADIVY 算法能够有效的跳出局部最优解，更接近于理论最优解-1.0300，对于固定维度基准测试函数中，ADIVY 算法都能收敛到最优解附近，其中 F18、F22、F23 基准函数都能直接收敛到理论值附近，ADIVY 算法在多个基准测试函数上的表现相较于其他优化算法表现出了显著的优势。例如，在 F1 函数中，ADIVY 算法的最优值为 0，相比于 GWO 提高了约 100%；在 F2 函数中，ADIVY 算法的最优值为 0，相比于 WOA 提升了约 100%。此外，在 F8 函数中，ADIVY 算法的最优值为-3.012E+03，相比于 BOA 提升了约 14.8%。说明了本文 ADIVY 算法相比于其他主流算法，在基准测试函数寻优上，有明显的优势。

3.5 ADIVY 算法与其他算法高维（200 维）性能对比

为了测试本文 ADIVY 算法对于高维函数的寻优能力，选取与以上相同的算法进行对比，分别为 IVY、WOA、DBO、BOA、GWO、HHO、SO 算法。取维度为 200，最大迭代次数 Maxiter 为 1000 次，各算法的其他参数与表 5 一致，各算法运行 30 次，因为 F14 到 F23 函数为固定维度多峰函数，因此对比结果只能到 F13 函数，其高维迭代 F1-F12 函数对比结果图如图 14 所示。





Copyright © This work is licensed under a Commons Attibution-Non Commercial 4.0 International License.

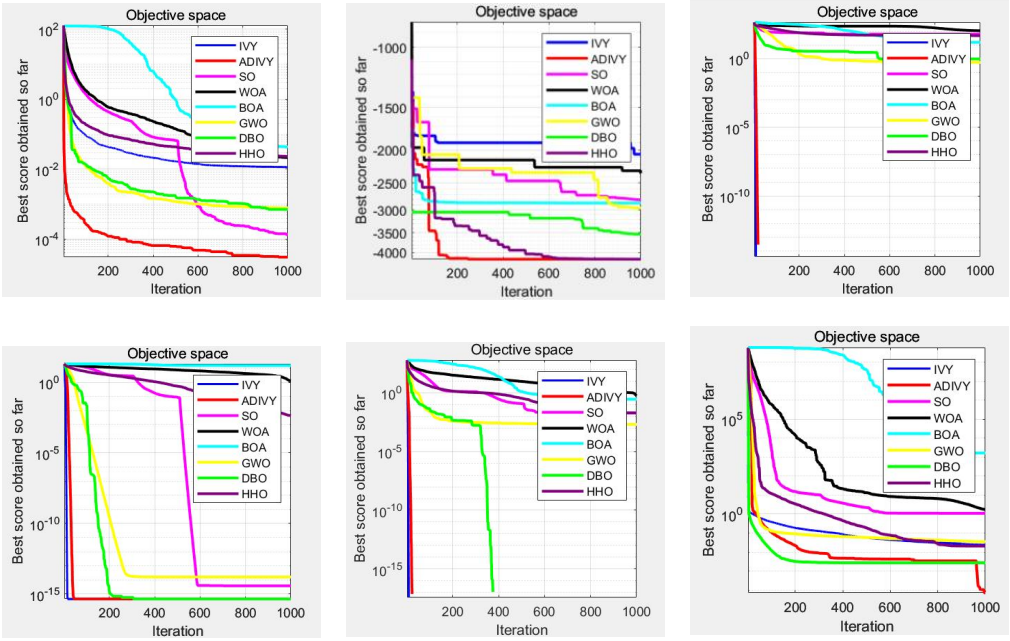



图 14 CEC2005 测试函数 F1-F12 高维迭代曲线

根据图 14 可知，ADIVY 算法在多个高维 CEC2005 测试函数中展现了显著的优越性。对于单峰函数（如 F1、F2、F3、F4），在 500 维情况下，ADIVY 算法能够稳定地找到最优值，并且在收敛过程中表现出较快的收敛速度，相较于其他算法（如 IVY、GWO、HHO 等），ADIVY 算法的表现尤为突出，证明了其在单峰优化问题中的优异性能。

在多峰函数（如 F7、F9 等）中，ADIVY 算法有效避免了局部最优困境，能够成功跳出局部最优值并收敛至更优的解，这一优势使得 ADIVY 在处理复杂多峰问题时具备明显的竞争力。特别是 ADIVY 能够通过创新的策略，在多峰优化问题中保持较高的全局探索能力，进而获得更加准确的优化解，ADIVY 算法在高维优化问题中不仅展示了较高的求解精度，还具备显著的收敛速度，进一步证明了其在高维复杂优化问题中的强大性能和鲁棒性。

### 3.6 CEC2005 函数 Wilcoxon 秩和检测

Wilcoxon 秩和检测是一种非参数统计检验方法，能够检测更为复杂的是数据分布问题，一般数据分析只是针对当前数据的平均值和标准差，并没有与算法多次运行的数据进行对比，因此这种数据对比分析是并不科学的，为了体现本文 ADIVY 算法的优越性与准确性，采用统计分析方法对每一次仿真结果进行分析，从统计学角度分析 ADIVY 算法与其他算法的性能差异。选取 ADIVY 算法在前 12 个测试函数的运行结果与以上的 7 个主流算法运行结果进行 Wilcoxon 秩和检验并计算 p 值，当 p 值小于 5%时，可以表示为拒绝零假设的有力验证，“+”“=”“-”分别表示为 ADIVY 算法寻优性能好于、等于、差于其他算法，其具体结果如表 8 所示。

表 8 CEC2005 测试函数 Wilcoxon 秩和检测数据

| 函数 | IVY(p1) | WOA(p2) | DBO(p3) | BOA(p4) | GWO(p5) | HHO(p6) | SO(p7) |
|---|---|---|---|---|---|---|---|
| F1 | 1.02E-16 | 3.31E-19 | 7.06E-18 | 3.31E-19 | 1.04E-15 | 2.95E-17 | 3.31E-19 |
| F2 | 2.65E-13 | 7.06E-18 | 5.12E-13 | 3.31E-19 | 2.01E-16 | 1.73E-15 | 2.89E-16 |
| F3 | 1.12E-17 | 6.68E-14 | 2.89E-16 | 3.31E-19 | 3.50E-14 | 1.06E-14 | 2.13E-15 |
| F4 | 9.22E-12 | 7.06E-18 | 3.31E-19 | 7.06E-18 | 2.89E-13 | 3.31E-19 | 4.20E-17 |
| F5 | 1.06E-11 | 7.06E-18 | 3.31E-19 | 7.06E-18 | 7.06E-18 | 3.94E-12 | 1.38E-17 |
| F6 | 3.31E-19 | 5.06E-17 | 2.47E-17 | 3.94E-17 | 7.06E-18 | 1.67E-15 | 2.34E-01 |
| F7 | 4.80E-17 | 1.12E-17 | 2.47E-17 | 7.06E-18 | 5.06E-18 | 7.06E-18 | 7.06E-18 |
| F8 | 1.12E-17 | 7.06E-18 | 2.47E-17 | 1.12E-17 | NaN | 7.06E-18 | 1.04E-15 |
| F9 | 1.12E-17 | 1.27E-16 | 4.11E-18 | 1.04E-15 | 1.01E-14 | 1.02E-11 | 4.80E-17 |
| F10 | 2.29E-18 | 2.13E-17 | 1.04E-15 | 1.12E-17 | 5.94E-12 | 2.34E-01 | 1.73E-15 |
| F11 | 1.12E-17 | 1.28E-16 | 7.06E-18 | 2.47E-27 | 2.47E-17 | 5.94E-18 | NaN |

Copyright © This work is licensed under a Commons Attibution-Non Commercial 4.0 International License.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| F12 | 2.47E-13 | 7.06E-18 | 7.06E-18 | 3.31E-19 | 3.31E-19 | 4.11E-16 | 4.11E-18 |
| +/=/- | 12/0/0 | 12/0/0 | 12/0/0 | 12/0/0 | 11/0/1 | 11/1/0 | 11/0/1 |

根据以上表格，ADIVY 算法的 Wilcoxon 秩和检验结果的 p 值大多小于 5%，表明其在统计学上具有显著优越性。这一结果验证了 ADIVY 算法在多个基本函数寻优问题中的优势，特别是在求解精度和收敛速度方面，相较于其他对比算法表现更为出色。Wilcoxon 秩和检验的 p 值小于 5%表明 ADIVY 与其他算法存在显著差异，进一步证实了其鲁棒性和稳定性。总之，ADIVY 算法不仅在理论上具备强大优化能力，在实际应用中也表现出较高的可靠性和有效性。

3.7 CEC2022 测试函数实验分析及 Wilcoxon 秩和检测

CEC2022 测试函数集由多个基本优化测试函数的组合而成，这些函数的设计旨在增加优化问题的复杂性。通过利用这些复杂特征的测试函数，本文对 ADIVY 方法进行了全面的性能测试。一方面，这些复杂函数能够有效体现 ADIVY 在处理复杂优化问题时的优越性能；另一方面，采用多种测试函数的组合优化也展示了 ADIVY 对不同类型复杂优化问题的适用性。为了进一步验证 ADIVY 的鲁棒性，本文选取了部分 CEC2022 单目标优化函数进行求解分析,这些函数包括单峰、多峰、混合和复合类型，涵盖了广泛的优化场景，确保了对算法性能的全面评估。

本文将 ADIVY 算法与标准的常春藤 IVY 算法、标准鲸鱼 WOA 算法、蜣螂 DBO 算法、蝴蝶(BOA)算法以及引入 LHS-GRN 初始化的 LGIVY 算法、融合 DSF-HHO 机制的 DHIVY 算法进行对比，实验参数取种群规模 N 为 100，维度 d 选 20 测试，最大迭代次数 Maxiter 为 1000，对每个函数独立运行 30 次，取平均值和标准差，其具体对比数据如表 9 所示，迭代曲线如图 15 所示，对 CEC2022 函数测试库进行 Wilcoxon 秩和检测，其结果数据如表 10 所示。

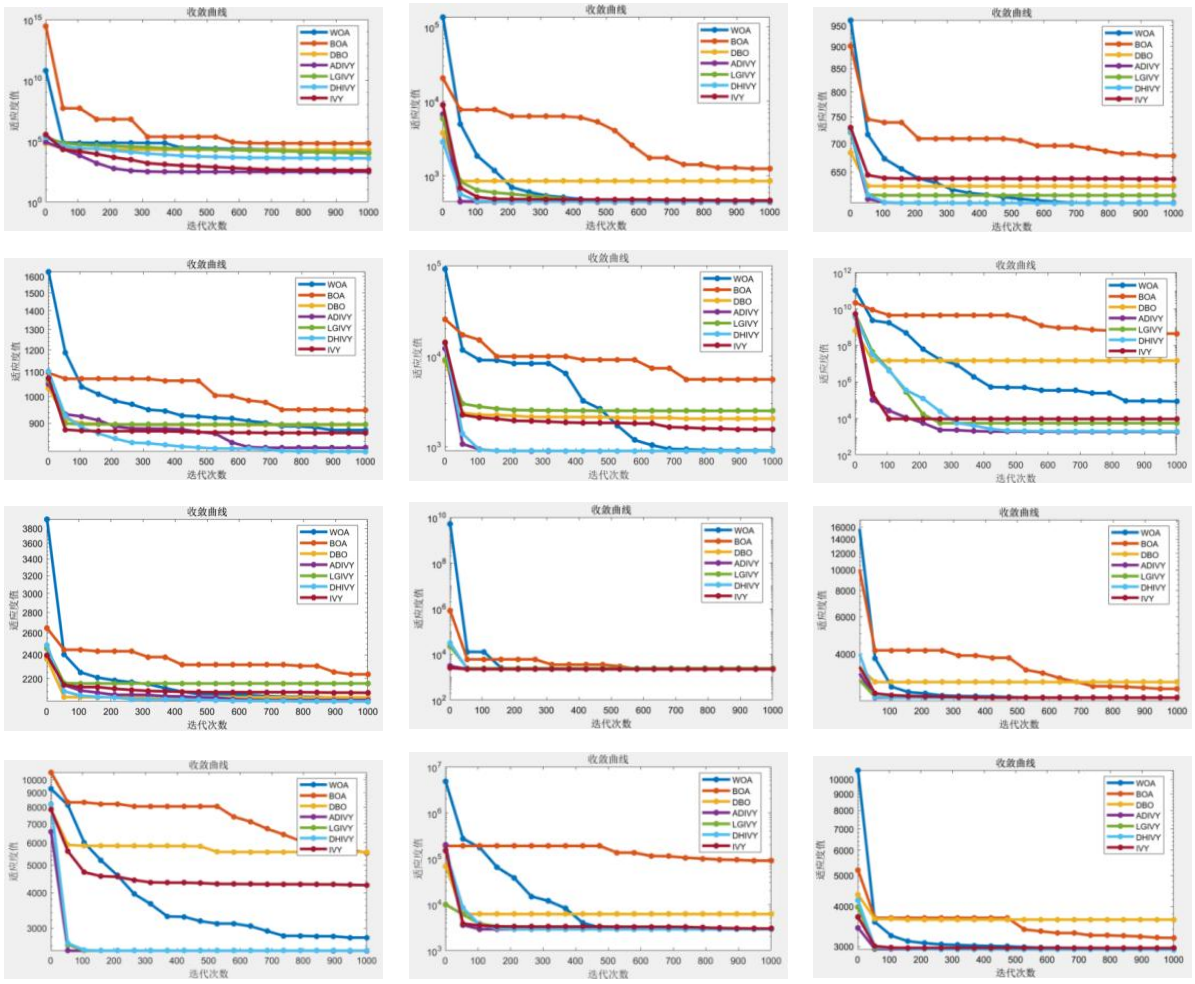



图 15 CEC2022 算法迭代对比曲线


Copyright © This work is licensed under a Commons Attibution-Non Commercial 4.0 International License.

表 9 CEC2022 测试函数迭代数据

| 函数 | IVY 算法 | | | WOA 算法 | | | DBO 算法 | | | BOA 算法 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 最优值 | 平均值 | 标准差 | 最优值 | 平均值 | 标准差 | 最优值 | 平均值 | 标准差 | 最优值 | 平均值 | 标准差 |
| F1 | 301.1276 | 209.132 | 3.6482 | 303.4229 | 308.821 | 1.830E+01 | 321.44 | 338.654 | 1.646E+03 | 500.6544 | 765.845 | 8.344E+03 |
| F2 | 401.9162 | 403.27 | 3.7333 | 406.3431 | 407.38 | 2.1164 | 502.6254 | 515.87 | 9.0586E+01 | 588.5521 | 669.30 | 1.4653E+02 |
| F3 | 614.7047 | 614,89 | 6.9326 | 600 | 600 | 1.2939e-05 | 620.6656 | 628.06 | 8.3542 | 637.7263 | 636.97 | 8.7194 |
| F4 | 815.9193 | 817.65 | 6.1956 | 809.8981 | 811.42 | 2.7137 | 813.8346 | 819.14 | 5.1637 | 841.6316 | 850.32 | 1.3422E+01 |
| F5 | 962.0406 | 965.43 | 1.0061E+02 | 900.0895 | 901.08 | 2.0248E-01 | 986.9445 | 1032.03 | 8.5576E+01 | 1019.3193 | 1123.45 | 4.2230E+02 |
| F6 | 1854.0882 | 188.25 | 3.3017E+01 | 2298.0183 | 2365.435 | 1.0174E+02 | 2409.9173 | 2798.512 | 7.1777E+02 | 3456.8890 | 4435.987 | 1.5655E+06 |
| F7 | 2022.8829 | 2030.612 | 8.9323 | 2006.7809 | 2015.471 | 2.1081 | 2071.7924 | 2075.334 | 1.3150 | 2087.1263 | 2097.653 | 2.4232E+01 |
| F8 | 2226.6289 | 2227.85 | 1.9284 | 2215.554 | 2253.76 | 4.1254E+01 | 2226.7634 | 2237.91 | 2.6296 | 2239.1891 | 2454.87 | 4.4115E+01 |
| F9 | 2526.2344 | 2532.4 | 1.3688E+01 | 2529.2765 | 2568.76 | 1.6317E+01 | 2694.8385 | 2699.83 | 2.7903e+01 | 2670.6089 | 2672.65 | 4.5888E+01 |
| F10 | 2517.2797 | 2545.76 | 4.0228e+01 | 2417.3496 | 2428.63 | 2.3866E+01 | 2501.1253 | 2575.82 | 6.1279E+01 | 2515.4141 | 2565.43 | 7.4097E+01 |
| F11 | 2786.4564 | 2806.554 | 1.6546E+02 | 2900 | 2976.55 | 7.6112E+01 | 3012.6835 | 3162.91 | 3.0444E+02 | 4625.8471 | 5765.43 | 3.7331E+03 |
| F12 | 2864.9331 | 2877.50 | 1.9355 | 2863.1829 | 2867.44 | 1.2737 | 2934.5940 | 2950.71 | 3.7162E+01 | 2872.1985 | 2899.81 | 3.6017e+01 |
| Friedman 检验排名 | | 4 | | | 5 | | | 6 | | | 7 | |

| 函数 | LGIVY 算法 | | | DHIVY 算法 | | | ADIVY 算法 | | |
|---|---|---|---|---|---|---|---|---|---|
| | 最优值 | 平均值 | 标准差 | 最优值 | 平均值 | 标准差 | 最优值 | 平均值 | 标准差 |
| F1 | 300 | 300 | 7.5736E-05 | 300 | 301.8873 | 6.4570E-04 | 300 | 300 | 0 |
| F2 | 405.7562 | 411.44 | 2.1135E+01 | 408.7765 | 409.51 | 1.5072 | 400.9161 | 405.89 | 3.1038 |
| F3 | 600 | 600.31 | 1.1978 | 600 | 600 | 3.4075E-12 | 600 | 600 | 3.6055E-17 |
| F4 | 807.9597 | 815.14 | 8.6331 | 802.6573 | 805.58 | 6.7205E-01 | 800.0442 | 801.54 | 2.5684E-01 |
| F5 | 900 | 900 | 2.4030E-08 | 923.8571 | 943.87 | 1.7599E+02 | 900 | 900 | 0 |
| F6 | 1804.3912 | 1810.6437 | 1.5471e+03 | 1800.2153 | 1801.2564 | 6.1625e-01 | 1800.092 | 1800.154 | 2.2084E-01 |
| F7 | 2000.9962 | 2021.467 | 1.3726E+0 | 2000.9955 | 2004.751 | 5.7599E-01 | 2000.2297 | 2001.19 | 5.4160 |

Copyright © This work is licensed under a Commons Attibution-Non Commercial 4.0 International License.

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | 1 | | | | | | |
| F8 | 2220.7283 | 2231.64 | 3.2768 | 2220.1371 | 2227.984 | 6.1359 | 2202.0822 | 2209.16 | 9.4770 |
| F9 | 2531.4026 | 2567.43 | 2.6790E+01 | 2529.2853 | 2529.285 | 0 | 2403.2844 | 2403.284 | 0 |
| F10 | 2500.5177 | 2545.61 | 6.0780E+01 | 2500.2681 | 2500.34 | 4.5857E-02 | 2500.1296 | 2510.91 | 3.2734E+01 |
| F11 | 2801.87 | 2886.75 | 1.3464E+02 | 2704.3741 | 2770.03 | 9.1538E+01 | 2600 | 2600 | 0 |
| F12 | 2862.1359 | 2871.8 | 9.4860 | 2866.9527 | 2887.44 | 1.2596E+01 | 2861.4049 | 2862.34 | 1.4454 |
| Friendman 检验排名 | | 3 | | | 2 | | | 1 | |

表 10 CEC2022Wilcoxon 秩和检测数据

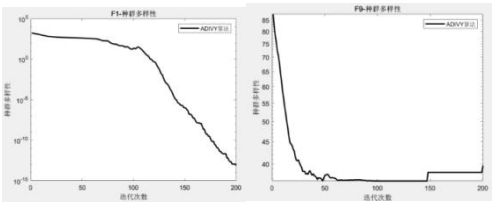
| 函数 | IVY(p1) | WOA(p2) | DBO(p3) | BOA(p4) | LGIVY(p5) | DHIVY(p6) |
|---|---|---|---|---|---|---|
| F1 | 7.543E-08 | 2.171E-05 | 3.334E-07 | 5.661E-08 | 1.231E-01 | 1.544E-04 |
| F2 | 1.231E-07 | 4.713E-05 | 2.334E-10 | 5.554E-10 | 3.332E-01 | 2.436E-06 |
| F3 | 3.776E-07 | 3.443E-01 | 3.334E-09 | 3.454E-08 | 4.445E-05 | 1.327E-01 |
| F4 | 7.543E-08 | 1.786E-08 | 2.231E-06 | 1.274E-10 | 3.253E-01 | 3.473E-05 |
| F5 | 1.786E-08 | 1.223E-04 | 3.361E-07 | 3.334E-08 | 5.556E-04 | 1.546E-07 |
| F6 | 5.088E-06 | 3.320E-05 | 3.341E-07 | 1.223E-09 | 3.334E-04 | NaN |
| F7 | 4.713E-05 | 2.334E-05 | 3.451E-06 | 3.341E-10 | 1.254E-01 | 4.467E-04 |
| F8 | 3.320E-05 | 3.213E-05 | 3.421E-05 | 2.223E-10 | NaN | 1.546E-05 |
| F9 | NaN | 1.223E-04 | 3.361E-07 | 3.334E-09 | 3.445E-06 | 3.312E-08 |
| F10 | 5.661E-08 | 2.484E-05 | 3.361E-07 | 1.223E-09 | 1.236E-01 | 1.253E-05 |
| F11 | 1.223E-04 | 3.334E-06 | 1.223E-09 | 1.226E-11 | 2.484E-05 | 2.332E-07 |
| F12 | 4.713E-05 | 2.875E-05 | 2.443E-07 | 3.332E-09 | 4.457E-04 | 3.332E-05 |
| +/=/- | 11/0/1 | 11/1/0 | 12/0/0 | 12/0/0 | 6/5/1 | 10/1/1 |

根据 CEC2022 测试函数的迭代图和迭代数据，ADIVY 算法在单峰函数中成功寻找到最优解 300，表现优异。在基础测试函数中，ADIVY 算法整体优于其他主流算法，并能够较快地收敛。在复合和组合测试函数中，ADIVY 算法成功避免了局部最优，接近最优解。在测试函数 F3 中，ADIVY 算法与 WOA 算法表现相当；在 F9 测试函数中，ADIVY 算法虽低于 IVY 算法，但仍有效验证了其在测试问题中的可行性。

根据表中的 Wilcoxon 秩和检验结果，CEC2022 测试函数的 p 值大多小于 5%，表明 ADIVY 算法在基准测试函数上的寻优结果显著优于其他主流算法，尤其在收敛速度和求解精度方面表现更为出色。这进一步证明了 ADIVY 算法的鲁棒性与可行性。

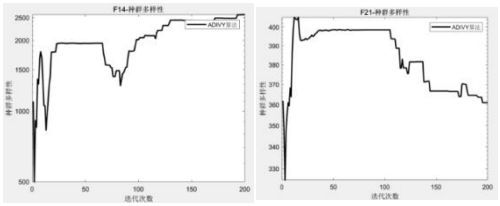
3.8 ADIVY 算法种群多样性分析

为验证改进算法在迭代过程中对种群多样性的影响，对 CEC2005 测试函数下的 F1 单峰函数、F9 多峰函数、F14 和 F21 固定维度多峰函数进行测试，函数的维度选为 30，种群初始数目为 30，最大迭代次数定为 200，图 16 的(a)到(d)分别为 F1、F9、F14、F21 的种群多样性迭代曲线。



(a)F1 种群多样性　　(b)F1 种群多样性


Copyright © This work is licensed under a Commons Attibution-Non Commercial 4.0 International License.



(c)F1 种群多样性　　(d)F1 种群多样性

图 16 F1-F21 种群多样性迭代曲线

通过图中展示的 ADIVY 算法在 F1、F9、F14 和 F21 函数的种群多样性迭代曲线，显著体现了该算法在种群多样性维持方面的优势。在单峰函数 F1 中，ADIVY 算法保持了较高的多样性，确保了稳定收敛，避免局部最优；而传统 IVY 算法采取随机分布初始减少种群多样性，且每次初始化的差异过大，鲁棒性差；在多峰函数 F9 中，ADIVY 算法展现出较快的多样性收敛速度，有助于跳出局部最优，提升全局搜索能力；在 F14 和 F21 函数中，尽管存在一定震荡，ADIVY 算法仍保持了稳定的种群多样性，增强了对复杂多峰问题的适应性。

综合来说，ADIVY 算法有效维持种群多样性，从而提升了搜索精度和全局优化能力，验证了其在种群多样性保持方面的优势。

3.9 ADIVY 算法平均运行时间与成功率对比分析

为验证 ADIVY 优化算法处理问题的速度和成功率，选取前面所述的 LGIVY、AIVY、DPIVY、DHIVY 单一改进策略进行基准测试函数平均寻优时间和寻优成功率的对比，实验数据统一设置为：维度 d 为 30，种群规模 N 为 100，最大迭代次数 Maxiter 设置为 1000 次。各算法运行 30 次，其中运行成功率的定义如下：

设适应度误差为 $F(u)$：

$$F(u)=f(X(u))-f(X') \tag{46}$$

式中，u 为算法运行次数，X(u)为算法运行第 次的实际寻优结果，X’为理论的最优值，定义变量 $\delta(u)$ 为：

$$\delta(u)=\begin{cases}1, & |F(u)|<\varepsilon\\ 0, & |F(u)|\geq\varepsilon\end{cases} \tag{47}$$

式中，$\varepsilon$ 为适应度误差精度，具体取值见表 。算法运行成功率 P 定义为：

$$P=\frac{1}{30}\sum_{u=1}^{30}\delta(u) \tag{48}$$

ADIVY 优化算法与其他对比算法在 CEC2005 中 23 个基准函数的平均寻优时间以及寻优成功率对比结果见表 11 所示。

由表中数据可知，LGIVY、AIVY、DPIVY、DHIVY 和 ADIVY 算法在优化问题的处理速度上，相较于传统的 IVY 算法具有显著优势。在 23 个基准测试函数中，ADIVY 算法表现出最快的寻优速度，尽管个别数据持平，表明所提出的改进机制在一定程度上加速了算法的收敛过程。其中，DHIVY 算法的提升速度最为显著，验证了 DSF-HHO 机制在提升收敛速度方面的有效性。此外，ADIVY 算法在 14 个基准测试函数中实现了 100%的寻优成功率，并且在 23 个基准测试函数中，其寻优成功率也高于其他对比算法，表明 ADIVY 算法在优化问题处理上的稳定性较强。

综合而言，ADIVY 算法在优化问题处理上兼顾了速度与稳定性，展现出较高的鲁棒性。

表 11 ADIVY 算法与其他算法运行时间、成功率数据对比

| 函数 | IVY | | | LGIVY | | | AIVY | | | DPIVY | | | DHIVY | | | ADIVY | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 平均值 | 标准差 | 成功率% | 平均值 | 标准差 | 成功率% | 平均值 | 标准差 | 成功率% | 平均值 | 标准差 | 成功率% | 平均值 | 标准差 | 成功率% | 平均值 | 标准差 | 成功率% |
| F1 | 1.8055 | 0.0116 | 100 | 1.7243 | 0.055 | 100 | 1.8065 | 0.0155 | 100 | 1.9403 | 0.0283 | 100 | 1.8065 | 0.0334 | 100 | 1.7743 | 0.0113 | 100 |
| F2 | 1.9952 | 0.0113 | 98.3 | 1.8549 | 0.067 | 100 | 1.8905 | 0.0101 | 100 | 1.8903 | 0.0223 | 98.5 | 1.8655 | 0.0151 | 100 | 1.8906 | 0.0754 | 100 |
| F3 | 1.9877 | 0.0771 | 0 | 1.8834 | 0.072 | 0 | 1.8846 | 0.0309 | 100 | 1.8573 | 0.0110 | 97 | 1.8577 | 0.0021 | 100 | 1.6562 | 0.0053 | 100 |
| F4 | 1.3354 | 0.0012 | 83.2 | 1.3875 | 0.0031 | 85.7 | 1.2249 | 0.0190 | 90.6 | 1.2563 | 0.0063 | 88.6 | 1.2893 | 0.0020 | 89 | 1.0489 | 0.0981 | 100 |
| F5 | 2.0155 | 0.0516 | 0 | 2.0000 | 0.000 | 0 | 1.9586 | 0.0101 | 86.7 | 2.0903 | 0.0101 | 100 | 1.9397 | 0.0101 | 97 | 1.8506 | 0.0000 | 98.5 |

Copyright © This work is licensed under a Commons Attibution-Non Commercial 4.0 International License.

| | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 27 | 4 | | 23 | 31 | | 41 | 84 | 5 | 47 | 81 | 0 | 85 | 13 | | 76 | 43 | 5 |
| F6 | 0.78 | 0.01 | 0 | 0.77 | 0.00 | 44. | 0.64 | 0.01 | 90. | 0.65 | 0.07 | 93. | 0.65 | 0.00 | 95. | 0.54 | 0.00 | 10 |
| | 99 | 12 | | 85 | 64 | 5 | 61 | 19 | 7 | 82 | 54 | 2 | 72 | 77 | 3 | 72 | 76 | 0 |
| F7 | 2.20 | 0.03 | 98. | 2.03 | 0.01 | 10 | 2.08 | 0.00 | 10 | 2.04 | 0.06 | 77. | 2.03 | 0.06 | 98. | 1.84 | 0.06 | 10 |
| | 44 | 54 | 3 | 05 | 84 | 0 | 45 | 53 | 0 | 85 | 64 | 7 | 858 | 52 | 7 | 78 | 45 | 0 |
| F8 | 1.99 | 0.07 | 76. | 2.15 | 0.01 | 86. | 2.08 | 0.00 | 76. | 1.94 | 0.00 | 88. | 1.84 | 0.00 | 84. | 1.74 | 0.03 | 83. |
| | 70 | 92 | 5 | 16 | 01 | 5 | 47 | 31 | 5 | 85 | 31 | 6 | 87 | 18 | 4 | 65 | 43 | 3 |
| F9 | 2.20 | 0.00 | 98. | 2.03 | 0.00 | 10 | 2.07 | 0.00 | 10 | 2.04 | 0.08 | 10 | 2.02 | 0.06 | 10 | 1.85 | 0.00 | 10 |
| | 40 | 31 | 4 | 34 | 61 | 0 | 461 | 33 | 0 | 39 | 61 | 0 | 14 | 73 | 0 | 73 | 24 | 0 |
| F1 | 1.84 | 0.01 | 0 | 1.80 | 0.00 | 44. | 1.75 | 0.00 | 76. | 1.83 | 0.00 | 95. | 1.74 | 0.01 | 10 | 1.74 | 0.00 | 10 |
| 0 | 33 | 28 | | 03 | 20 | 8 | 93 | 29 | 5 | 75 | 24 | 5 | 63 | 45 | 0 | 76 | 33 | 0 |
| F1 | 1.99 | 0.05 | 0 | 1.88 | 0.00 | 23. | 1.84 | 0.01 | 66. | 1.85 | 0.04 | 98. | 1.83 | 0.07 | 97. | 1.87 | 0.05 | 90. |
| 1 | 03 | 64 | | 57 | 55 | 3 | 57 | 12 | 5 | 82 | 46 | 7 | 73 | 64 | 6 | 65 | 63 | 5 |
| F1 | 0.83 | 0.04 | 46. | 1.03 | 0.02 | 55. | 0.82 | 0.01 | 96. | 0.81 | 0.06 | 76. | 0.75 | 0.00 | 10 | 0.74 | 0.00 | 10 |
| 2 | 33 | 49 | 6 | 01 | 75 | 7 | 75 | 27 | 3 | 85 | 54 | 6 | 61 | 20 | 0 | 76 | 55 | 0 |
| F1 | 1.77 | 0.06 | 17. | 1.65 | 0.00 | 76. | 1.77 | 0.04 | 56. | 1.74 | 0.01 | 56. | 1.64 | 0.04 | 76. | 1.64 | 0.00 | 89. |
| 3 | 65 | 41 | 8 | 54 | 96 | 9 | 69 | 35 | 3 | 82 | 90 | 6 | 63 | 61 | 6 | 67 | 28 | 7 |
| F1 | 1.33 | 0.00 | 0 | 1.22 | 0.00 | 13. | 1.32 | 0.00 | 77. | 1.27 | 0.00 | 99. | 1.27 | 0.00 | 10 | 1.09 | 0.06 | 10 |
| 4 | 45 | 78 | | 34 | 42 | 3 | 74 | 89 | 5 | 33 | 89 | 5 | 63 | 28 | 0 | 34 | 75 | 0 |
| F1 | 0.98 | 0.00 | 0 | 0.90 | 0.05 | 53. | 0.84 | 0.00 | 77. | 0.78 | 0.00 | 86. | 0.84 | 0.00 | 78. | 0.64 | 0.06 | 98 |
| 5 | 77 | 45 | | 33 | 46 | 9 | 74 | 81 | 4 | 32 | 21 | 5 | 75 | 38 | 5 | 65 | 578 | |
| F1 | 0.33 | 0.08 | 76. | 0.32 | 0.00 | 10 | 0.44 | 0.01 | 65. | 0.44 | 0.00 | 87. | 0.26 | 0.00 | 98 | 0.18 | 0.00 | 10 |
| 6 | 53 | 67 | 5 | 56 | 67 | 0 | 61 | 00 | 4 | 61 | 14 | 5 | 46 | 65 | | 37 | 17 | 0 |
| F1 | 0.44 | 0.00 | 10 | 0.45 | 0.00 | 10 | 0.35 | 0.00 | 10 | 0.41 | 0.00 | 10 | 0.42 | 0.07 | 10 | 0.38 | 0.01 | 10 |
| 7 | 21 | 12 | 0 | 74 | 17 | 0 | 47 | 39 | 0 | 34 | 13 | 0 | 94 | 54 | 0 | 45 | 65 | 0 |
| F1 | 1.33 | 0.01 | 87. | 1.24 | 0.08 | 85. | 1.24 | 0.00 | 88. | 1.26 | 0.01 | 90. | 1.30 | 0.00 | 97. | 1.12 | 0.07 | 98 |
| 8 | 56 | 56 | 6 | 32 | 43 | 4 | 61 | 84 | 7 | 35 | 28 | 7 | 87 | 87 | 6 | 74 | 45 | |
| F1 | 0.79 | 0.00 | 10 | 0.78 | 0.01 | 10 | 0.74 | 0.00 | 10 | 0.65 | 0.00 | 10 | 0.64 | 0.06 | 10 | 0.64 | 0.00 | 10 |
| 9 | 16 | 76 | 0 | 35 | 13 | 0 | 73 | 76 | 0 | 42 | 21 | 0 | 56 | 53 | 0 | 55 | 20 | 0 |
| F2 | 0.88 | 0.00 | 0 | 0.86 | 0.06 | 0 | 0.84 | 0.02 | 34. | 0.84 | 0.00 | 65. | 0.80 | 0.00 | 46. | 0.64 | 0.01 | 76. |
| 0 | 90 | 45 | | 23 | 63 | | 72 | 44 | 5 | 86 | 24 | 5 | 43 | 22 | 6 | 56 | 81 | 6 |
| F2 | 0.38 | 0.00 | 25. | 0.35 | 0.00 | 28. | 0.41 | 0.00 | 66. | 0.43 | 0.01 | 73. | 0.32 | 0.00 | 76. | 0.29 | 0.00 | 83. |
| 1 | 04 | 42 | 6 | 80 | 25 | 8 | 64 | 19 | 4 | 61 | 12 | 2 | 94 | 76 | 5 | 83 | 24 | 5 |
| F2 | 0.47 | 0.03 | 10 | 0.43 | 0.00 | 10 | 0.47 | 0.01 | 10 | 0.35 | 0.00 | 10 | 0.33 | 0.00 | 10 | 0.36 | 0.03 | 10 |
| 2 | 65 | 51 | 0 | 71 | 53 | 0 | 21 | 07 | 0 | 35 | 22 | 0 | 85 | 65 | 0 | 46 | 26 | 0 |
| F2 | 0.41 | 0.00 | 0 | 0.33 | 0.00 | 39. | 0.36 | 0.01 | 77. | 0.40 | 0.00 | 87. | 0.33 | 0.01 | 75. | 0.28 | 0.04 | 98. |
| 3 | 24 | 78 | | 54 | 55 | 9 | 51 | 13 | 6 | 63 | 53 | 6 | 61 | 07 | 1 | 47 | 46 | 3 |

## 4 ADIVY 算法工程应用分析

近年来，工程算例应用问题成为算法优化中的热门领域。为验证 ADIVY 算法在实际应用问题中的优越性和有效性，本文选取焊接梁设计优化、减速器重量最优、齿轮传动比最优设计和压力容器优化四个工程问题进行对比。对比算法包括蜣螂算法(DBO)、哈里斯鹰算法(HHO)、灰狼算法(GWO)、蛇算法(SO)、蝴蝶算法(BOA)、多元宇宙算法[25](MVO)、常春藤算法(IVY)、沙丘猫群算法[26](SCSO)、徒步旅行算法[27](HOA)以及 ADIVY 算法，每个问题的种群规模设为 100，最大迭代次数为 500，每个算法独立运行 30 次。

### 4.1 焊接梁设计问题

焊接梁的实用设计是一个常用于比较各种优化技术的问题，如图 17 所示，该问题是要确定理想的参数以降低制造成本，控制变量包括杆的高度 t(x2)、焊缝厚度 h(x1)、杆


Copyright © This work is licensed under a Commons Attibution-Non Commercial 4.0 International License.

的厚度 b(x4)、夹紧杆的长度 l(x3)。

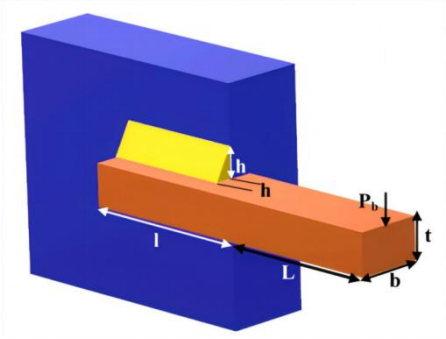



图 17 焊接梁示意图

焊接梁问题的数学模型如下所示：

目标函数为：

约束条件：

$$\min f(X)=1.10471x_1^2x_2+0.04811x_3x_4(14+x_2) \quad (49)$$

$$\begin{cases} \tau'=\dfrac{P}{2x_1x_2} \\ \tau''=MRJ, M=P\left(L+\dfrac{x_2}{2}\right) \\ J=2\left\{\sqrt{2}x_1x_2\left[\dfrac{x_2^2}{12}+\left(\dfrac{x_1+x_3}{2}\right)^2\right]\right\} \\ R=\sqrt{\dfrac{x_2^2}{4}+\left(\dfrac{x_1+x_3}{2}\right)^2} \\ P=6000 \\ L=14 \\ E=3\times10^7 \\ G=1.2\times10^7 \\ \tau_{\max}=13600 \\ \sigma_{\max}=30000 \\ \delta_{\max}=0.25 \end{cases} \quad (50)$$

边界约束为：

$$\begin{aligned} &\text{s.t.}g_1(X)=\sqrt{(\tau')^2+2\tau'\tau''\frac{x_2}{2R}+(\tau'')^2}-\tau_{\max}\le 0 \\ &g_2(X)=\frac{6PL}{x_3^2x_4}-\sigma_{\max}\le 0 \\ &g_3(X)=x_1-x_4\le 0 \\ &g_4(X)=0.10471x_1^2+0.04811x_3x_4(14+x_2)-5\le 0 \\ &g_5(X)=0.125-x_1\le 0 \\ &g_6(X)=\frac{4PL^3}{Ex_3^3x_4}-\delta_{\max}\le 0 \\ &g_7(X)=P-\frac{4.013Ex_3x_4^3}{6L^2}\left(1-\frac{x_3}{2L}\sqrt{\frac{E}{4G}}\le 0\right) \\ &0.1\le x_i\le 2, i=1,4 \\ &0.1\le x_i\le 10, i=2,3 \end{aligned} \quad (51)$$

其采用不同算法寻优的迭代曲线如图 18 所示，结果数据具体如表 12 所示：

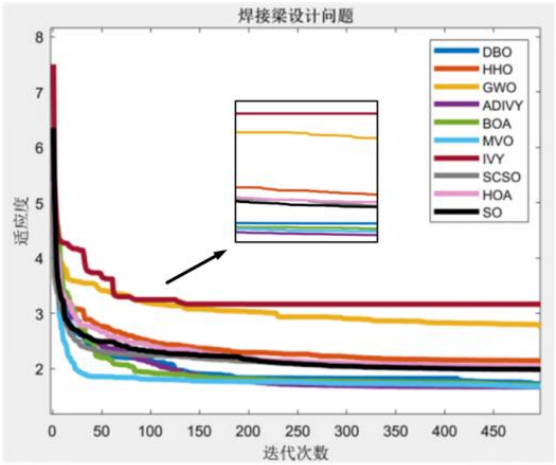



图 18 焊接梁不同算法迭代曲线

表 12 焊接梁不同算法迭代数据

| 算法 | h | l | t | b | 最优值 | 平均值 | 标准差 |
|---|---|---|---|---|---|---|---|
| DBO | 0.1987 | 3.4591 | 9.0021 | 0.2020 | 1.7443 | 1.7692 | 0.1032 |
| HHO | 0.2112 | 3.4658 | 8.9952 | 0.2019 | 1.8145 | 1.8550 | 0.2015 |
| GWO | 0.2103 | 3.4658 | 9.0261 | 0.2131 | 1.9344 | 2.1360 | 0.8744 |
| SO | 0.2048 | 3.4572 | 9.0114 | 0.2033 | 1.7908 | 1.8326 | 0.0428 |
| BOA | 0.2043 | 3.4748 | 9.0194 | 0.2042 | 1.7134 | 1.7285 | 0.0035 |
| MVO | 0.2071 | 3.4711 | 9.0356 | 0.2024 | 1.7580 | 1.7771 | 0.6522 |
| IVY | 0.2152 | 3.4683 | 9.0341 | 0.2043 | 1.9025 | 2.0453 | 0.0078 |
| SCSO | 0.2103 | 3.4557 | 9.0285 | 0.2060 | 1.8876 | 1.9537 | 0.0451 |
| HOA | 0.2075 | 3.4774 | 8.9962 | 0.2051 | 1.8923 | 1.9231 | 0.1325 |
| ADIVY | 0.1995 | 3.4502 | 8.9803 | 0.2021 | 1.7003 | 1.7203 | 2.5782E-02 |

从图 17 和表 12 中看出，ADIVY 算法在焊接梁设计问题中的优化效果优于其他主流算法，表现出较强的竞争力。特别是在多个优化指标和单个焊接梁属性的优化中，ADIVY 算法取得了最优结果，证明了其在复杂优化问题中的高效性和稳定性。与其他算法相比，ADIVY 算法通过独特机制显著提高了优化精度和收敛速度，验证了其在实际应用中的有效性和优越性。

4.2 减速器重量最优问题

Copyright © This work is licensed under a Commons Attibution-Non Commercial 4.0 International License.

图 19 所示的减速器设计所面临的优化挑战为降低减速器的重量。

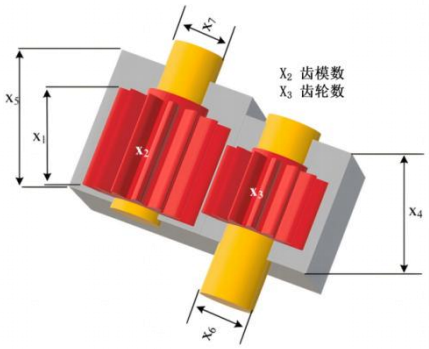



图 19 减速器机构示意图

其中控制参数包括齿宽 b（表示为 x1），齿模数 m（表示为 x2），小齿轮的齿数 z（表示为 x3），轴承间第一轴的长度 l1（表示为 x4），轴承间第二轴的长度 l2（表示为 x5），第一轴的直径 d1（表示为 x6），以及第二轴的直径 d2（表示为 x7），其数学模型如下所示：

$$\begin{aligned} f(X) &= 0.7854x_1x_2^2(3.3333x_3^2+14.9334x_3-43.0934) \\ &-1.508\mathbf{x}_1(\mathbf{x}_6^2+\mathbf{x}_7^2)+7.4777(\mathbf{x}_6^3+\mathbf{x}_7^3)+0.7854(\mathbf{x}_4\mathbf{x}_6^2+\mathbf{x}_6\mathbf{x}_7^2) \end{aligned} \tag{52}$$

约束条件：

$$\begin{aligned} g_1(X) &= \frac{27}{x_1x_2^2x_3}-1\le 0 \\ g_2(X) &= \frac{397.5}{x_1x_2^2x_3^2}-1\le 0 \\ g_3(X) &= \frac{1.93x_4^3}{x_2x_6^4x_3}-1\le 0 \\ g_4(X) &= \frac{1.93x_5^3}{x_2x_7^4x_3}-1\le 0 \\ g_5(X) &= \frac{\sqrt{745x_4/(x_2x_3)^2+16.9\times10^6}}{110x_6^3}-1\le 0 \\ g_6(X) &= \frac{\sqrt{745x_5/(x_2x_3)^2+157.5\times10^6}}{85x_7^3}-1\le 0 \\ g_7(X) &= \frac{x_2x_3}{40}-1\le 0 \\ g_8(X) &= \frac{5x_2}{x_1}-1\le 0 \\ g_9(X) &= \frac{x_1}{12x_2}-1\le 0 \\ g_{10}(X) &= \frac{1.5x_6+1.9}{x_4}-1\le 0 \\ g_{11}(X) &= \frac{1.1x_7+1.9}{x_5}-1\le 0 \end{aligned} \tag{53}$$

边界约束：

$$\begin{aligned} 2.6 &\le x_1 \le 3.6 \\ 0.7 &\le x_2 \le 0.8 \\ 17 &\le x_3 \le 28 \\ 7.3 &\le x_4 \le 8.3 \\ 7.8 &\le x_5 \le 8.3 \\ 2.9 &\le x_6 \le 3.9 \\ 5.0 &\le x_7 \le 5.5 \end{aligned} \tag{54}$$

其采用不同算法寻优的迭代曲线如图 20 所示，结果数据具体如表 13 所示：

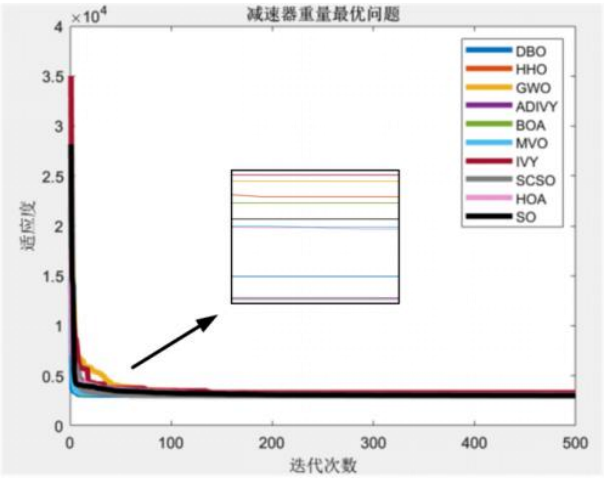



图 20 减速器设计不同算法迭代曲线

表 13 减速器设计不同算法迭代数据

| 算法 | b | m | z | l1 | l2 | d1 | d2 | 最优值 | 平均值 | 标准差 |
|---|---|---|---|---|---|---|---|---|---|---|
| DBO | 3.478 | 0.715 | 17.023 | 7.350 | 7.700 | 3.320 | 5.350 | 3.096E+03 | 3.170E+03 | 7.809E-02 |
| HHO | 3.512 | 0.680 | 17.042 | 7.608 | 7.818 | 3.300 | 5.265 | 3.055E+03 | 3.106E+03 | 5.476E-02 |
| GWO | 3.489 | 0.692 | 18015 | 7.533 | 7.829 | 3.312 | 5.310 | 3.103E+03 | 3.209E+03 | 3.334E-01 |


Copyright © This work is licensed under a Commons Attibution-Non Commercial 4.0 International License.

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| SO | 3.521 | 0.710 | 17.000 | 7.601 | 7.820 | 3.309 | 5.320 | 2.984E+03 | 3.036E+03 | 5.445E+02 |
| BOA | 3.498 | 0.693 | 17.001 | 7.592 | 7.810 | 3.300 | 5.279 | 3.001E+03 | 3.035E+03 | 3.334E-02 |
| MVO | 3.505 | 0.700 | 17.002 | 7.604 | 7.831 | 3.350 | 5.360 | 3.014E+03 | 3.079E+03 | 2.478E-02 |
| IVY | 3.472 | 0.698 | 18.354 | 7.59 | 7.800 | 3.319 | 5.292 | 3.002E+03 | 3.041E+03 | 4.556E-03 |
| SCSO | 3.509 | 0.699 | 17.000 | 7.590 | 7.803 | 3.312 | 5.266 | 2.965E+03 | 3.003E+03 | 3.334E+02 |
| HOA | 3.486 | 0.700 | 17.000 | 7.577 | 7.798 | 3.340 | 5.290 | 2.998E+03 | 3.006E+03 | 4.445E-02 |
| ADIVY | 3.475 | 0.685 | 17.000 | 7.556 | 7.820 | 3.302 | 5.273 | 2.796E+03 | 2.994E+03 | 5.764E-04 |

由以上的图 19 和表 13 数据显示，ADIVY 算法在减速器重量寻优问题中表现优于其他主流算法，找到的最小数值最优。然而，在个别属性上并非最优，如在齿宽属性上，ADIVY 算法的优化结果低于 IVY 算法，而在齿模数属性上，ADIVY 算法优化结果低于 HHO 算法。但总体而言，ADIVY 算法在重量优化方面优于 IVY 和 HHO 算法，证明了其在工程设计问题中的优化优势。

4.3 齿轮传动比最优问题

优化齿轮传动系统可在遵循指定约束条件的同时，使齿轮传动比的数值最小化，这是一个著名的工程设计难题，示意图如图 21 所示。

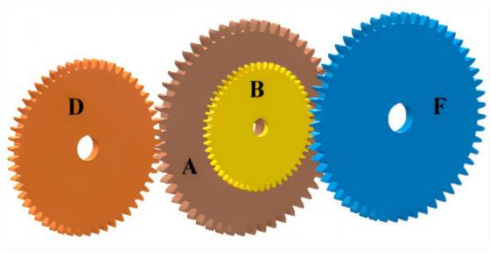



图 21 齿轮机构示意图

该问题主要有四个关键参数：nA（用 x1 表示）、nB（用 x2 表示）、nD（用 x3 表示）、nF（用 x4 表示），其数学模型具体如下所示：

目标函数：

$$f(x)=(\frac{1}{6.931}-\frac{x_2x_3}{x_1x_4})^2 \tag{55}$$

约束条件：

$$12<x_1,x_2,x_3,x_4<60 \tag{56}$$

其采用不同算法寻优的迭代曲线如图 22 所示，结果数据具体如表 14 所示：

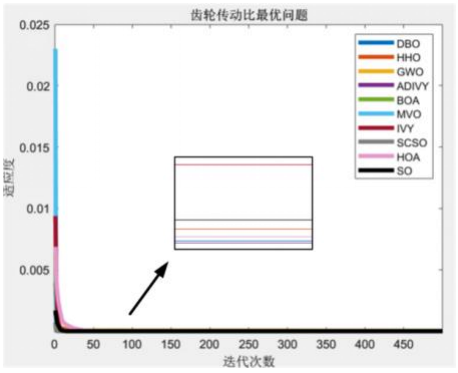



图 22 齿轮传动比不同算法迭代曲线

表 14 齿轮传动比不同算法迭代数据

| 算法 | nA | nB | nD | nF | 最优值 | 平均值 | 标准差 |
|---|---|---|---|---|---|---|---|
| DBO | 43.885 | 19.391 | 17.010 | 48.643 | 1.2245e-15 | 2.5037E-14 | 3.6817E-14 |
| HHO | 41.023 | 20.990 | 18.884 | 49.515 | 5.5543E-15 | 1.3437E-14 | 1.2456E-08 |
| GWO | 43.110 | 21.015 | 17.438 | 47.817 | 1.2234E-16 | 3.1793E-16 | 2.3345E-15 |
| SO | 42.304 | 19.000 | 18.000 | 48.000 | 1.3045E-15 | 3.8971E-15 | 1.2743E-09 |
| BOA | 40.501 | 18.103 | 16.000 | 49.616 | 2.33414E-16 | 1.1577E-15 | 1.7361E-15 |
| MVO | 43.643 | 20.000 | 19.540 | 50.954 | 1.1009E-15 | 1.3006E-15 | 3.5473E-08 |

Copyright © This work is licensed under a Commons Attibution-Non Commercial 4.0 International License.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| IVY | 43.120 | 21.866 | 17.576 | 51.914 | 4.3341E-14 | 6.3649E-14 | 3.3356E-13 |
| SCSO | 41.766 | 20.021 | 18.132 | 47.101 | 1.3304E-13 | 1.0287E-12 | 3.4451E-17 |
| HOA | 42.104 | 18.000 | 18.218 | 49.194 | 1.0963E-14 | 2.3721E-14 | 3.9738E-11 |
| ADIVY | 40.859 | 17.000 | 14.002 | 45.019 | 1.0031E-17 | 1.2234E-17 | 2.2234E-18 |

从图 21 和表 14 数据可见，ADIVY 算法在齿轮传动比最优的工程问题中展现出显著的优化优势。其最优值为 1.0031E-17，相较于传统 IVY 算法的最优值 4.3341E-14，ADIVY 算法的最优值提升了约 99.98%。这一结果表明，ADIVY 算法在解决该类工程优化问题时能够显著提高计算精度，接近理论最优解。此外，ADIVY 算法在平均值和标准差方面也表现出较小的波动，进一步证明了其在多次实验中的稳定性。

4.4 压力容器优化问题

图 23 展示了压力容器问题示意图，一个半球形封头覆盖着圆柱形容器的开口端，该问题的目的在于使圆柱压力容器的总成本达到最小。

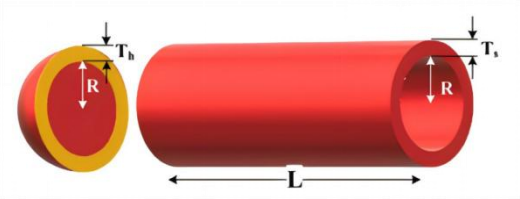



图 23 压力容器机构示意图

该问题有四个参数，包括壳体厚度 Ts（表示为 x3）、封头厚度 Th（表示为 x4）、圆柱壳体长度 L（表示为 x2）、内半径 R（表示为 x1），其中 TS 和 Th 为 0.625 的整数倍，R 和 L 为连续变量，该问题的数学模型如下所示：

目标函数：

$$\min f(x)=0.6224x_1x_3x_4+1.7781x_2x_3^2 +3.1661x_1^2x_4+19.84x_1^2x_3 \tag{57}$$

约束条件：

$$\begin{aligned} g_1(x)&=-x_1+0.0193x_3\le 0\\ g_2(x)&=-x_2+0.00954x_3\le 0\\ g_3(x)&=-\pi x_3^2x_4-\frac{4}{3}\pi x_3^2+1296000\le 0\\ g_4(x)&=x_4-240\le 0 \end{aligned} \tag{58}$$

边界约束：

$$\begin{aligned} 0\le x_1\le 99\\ 0\le x_2\le 99\\ 10\le x_3\le 200\\ 10\le x_4\le 200 \end{aligned} \tag{59}$$

其采用不同算法寻优的迭代曲线如图 24 所示，结果数据具体如表 15 所示：

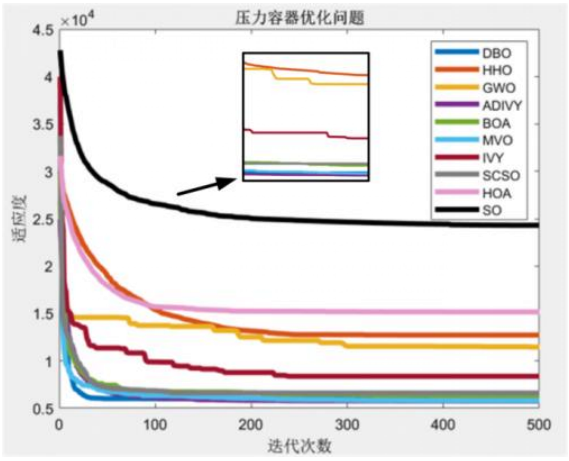



图 24 压力容器不同算法迭代对比曲线

表 15 压力容器不同算法滴迭代对比数据

| 算法 | Ts | Th | L | R | 最优值 | 平均值 | 标准差 |
|---|---|---|---|---|---|---|---|
| DBO | 1.385 | 0.692 | 58.635 | 26.925 | 6133.8765 | 6264.7136 | 287.456 |
| HHO | 0.913 | 0.457 | 46.228 | 69.545 | 7431.7462 | 7654.8237 | 325.721 |
| GWO | 1.102 | 0.573 | 59.452 | 40.239 | 7294.8471 | 7431.8133 | 310.982 |
| SO | 0.823 | 0.421 | 42.659 | 176.346 | 7344.5763 | 7624.8147 | 298.334 |
| BOA | 1.265 | 0.621 | 60.320 | 25.453 | 6347.6161 | 6565.8247 | 270.217 |
| MVO | 0.812 | 0.435 | 42.198 | 176.758 | 6481.8747 | 6981.7473 | 320.567 |
| IVY | 0.794 | 0.398 | 39.456 | 199.678 | 7078.4854 | 7515.8172 | 312.421 |
| SCSO | 1.250 | 0.541 | 64.332 | 10.786 | 6284.8948 | 6547.8173 | 280.652 |
| HOA | 1.118 | 0.528 | 61.078 | 17.654 | 7544.8876 | 7981.8173 | 335.787 |
| ADIVY | 0.768 | 0.384 | 40.213 | 199.840 | 5598.8717 | 5743.6564 | 260.175 |

从图 23 中和表 15 中数据可知，ADIVY 算法在压力容器优化问题的寻优结果要优于其他算法，且在单个属性中的寻优结果也是最优的，其最优值为 5598.8717，相比于传统 IVY 算法的最优值 7078.4854，ADIVY 算法的最优值提升了


Copyright © This work is licensed under a Commons Attibution-Non Commercial 4.0 International License.

约 20.99%，证明了其在处理复杂工程优化问题时的高效性和稳定性。

综上所述，通过四个工程设计优化问题的实验对比分析，ADIVY 算法在实际工程设计问题的优化中展现了显著的优越性。与传统算法相比，ADIVY 算法在求解精度、收敛速度和稳定性方面表现更为出色。这些结果进一步验证了 ADIVY 算法在复杂优化问题中的鲁棒性，能够有效应对多变的工程设计环境并提供优异的解决方案。

## 5 结论

本文提出的自适应常春藤（ADIVY）算法在工程设计优化问题中展现了显著优势。通过引入 LHS-GRN 初始化、多样性维持机制、自适应创新点机制和 DSF-HHO 策略，ADIVY 算法在多个方面优于传统 IVY 算法及其他主流优化算法（如 WOA、GWO、HHO 等）。LHS-GRN 初始化提升了种群均匀性与多样性，增强了搜索空间的覆盖，且自适应创新点和多样性维持机制有效避免了局部最优，提升了搜索精度。

DSF-HHO 策略的引入使得 ADIVY 能够在全局搜索阶段进行大范围跳跃，并在后期精细微调，从而显著提升了收敛速度和精度。实验结果表明，ADIVY 算法在多个基准测试函数中表现优异，尤其在解决多峰优化问题时，成功避免了局部最优困境，并提升了算法的鲁棒性，使其能够应对复杂的多约束和高维优化问题。

尽管 ADIVY 算法在多个问题中表现出色，但在高维度和大规模问题中仍面临较高的计算复杂度，且计算时间和资源消耗可能影响效率。未来，ADIVY 算法可通过优化计算效率、引入并行计算或降维技术，降低计算复杂度，提高算法的应用效率和处理能力。

## [参考文献]

[1] 朱孝山, 刘伟伟. 融合多策略改进黑猩猩优化算法的 UAV 航迹规划[J]. 电光与控制, 2024, 31(08): 50-57,68.

[2] 马志海, 刘升. 增强型野马优化算法及其工程应用[J].计算机应用研究, 2024, 41(07): 2061-2068.

[3] 王振宇, 王磊. 多策略帝王蝶优化算法及其工程应用[J]. 清华大学学报(自然科学版), 2024, 64(04): 668-678.

[4] Cui J, Wu L, Huang X, 等. Multi-strategy adaptable ant colony optimization algorithm and its application in robot path planning[J]. Knowledge-Based Systems, 2024, 288: 111459.

[5] Yao J, Luo X, Li F, et al. Research on hybrid strategy Particle Swarm Optimization algorithm and its applications[J]. Scientific Reports, 2024, 14(1): 24928.

[6] 陈旭升, 刘媛华. 多策略协同的足球队训练优化算法及其工程应用[J/OL]. 计算机工程与科学, 1-15[2025-03-22].

[7] 文裕杰, 张达敏. 增强型白鲸优化算法及其应用[J/OL]. 山东大学学报(工学版), 1-12[2025-03-22].

[8] 吴亚中, 陈璐, 马强, 等.多策略增强的蜣螂优化算法及其工程应用[J]. 华中科技大学学报(自然科学版), 2025, 53(02): 95-103.

[9] Akl D T, Saafan M M, Haikal A Y, 等. IHHO: an improved Harris Hawks optimization algorithm for solving engineering problems[J]. Neural Computing and Applications, 2024, 36(20): 12185-12298.

[10] Liu Y, As' arry A, Hassan M K, 等. Review of the grey wolf optimization algorithm: variants and applications[J]. Neural Computing and Applications, 2024, 36(6): 2713-2735.

[11] 刘成汉, 何庆. 融合多策略的黄金正弦黑猩猩优化算法[J]. 自动化学报, 2023, 49(11): 2360-2373.

[12] Ghasemi M, Zare M, Trojovský P, 等. Optimization based on the smart behavior of plants with its engineering applications: Ivy algorithm[J]. Knowledge-Based Systems, 2024, 295: 111850.

[13] 蔡玉林, 黄诗冰, 徐剑波,等. 基于常春藤算法优化机器学习模型预测边坡稳定性的研究[J/OL]. 金属矿山, 1-11[2025-03-22].

[14] 王恒迪, 王豪馗, 陈鹏, 等. 基于 IVYA-FMD 和 EELM-Yager 的轴承小样本故障诊断模型[J/OL]. 机电工程, 1-10[2025-03-22].

[15] Ouyang H, Li W, Gao F, et al. Research on Fault Diagnosis of Ship Diesel Generator System Based on IVY-RF[J]. Energies (19961073), 2024, 17(22).

[16] Gao H, Wang H, Shen H, et al. Multi-modal denoised data-driven milling chatter detection using an optimized hybrid neural network architecture[J]. Scientific Reports, 2025, 15(1): 3953.

[17] 吴禹志, 黄明, 夏静, 等.融合多策略增强型 IVYA 和 DWA 算法的多机动态航迹规划[J]. 计算机时代, 2025(02): 11-15.

[18] Zhang C, Lin W, Hu G. An enhanced ivy algorithm fusing multiple strategies for global optimization problems[J]. Advances in Engineering Software, 2025, 203: 103862.

[19] Mirjalili S, Lewis A. The whale optimization algorithm[J]. Advances in engineering software, 2016, 95: 51-67.

[20] Xue J, Shen B. Dung beetle optimizer: A new meta-heuristic algorithm for global optimization[J]. The

Copyright © This work is licensed under a Commons Attibution-Non Commercial 4.0 International License. 

Journal of Supercomputing, 2023, 79(7): 7305-7336.

[21] Arora S, Singh S. Butterfly optimization algorithm: a novel approach for global optimization[J]. Soft computing, 2019, 23: 715-734.

[22] Faris H, Aljarah I, Al-Betar M A, 等. Grey wolf optimizer: a review of recent variants and applications[J]. Neural computing and applications, 2018, 30: 413-435.

[23] Heidari A A, Mirjalili S, Faris H, 等. Harris hawks optimization: Algorithm and applications[J]. Future generation computer systems, 2019, 97: 849-872.

[24] Hashim F A, Hussien A G. Snake Optimizer: A novel meta-heuristic optimization algorithm[J]. Knowledge-Based Systems, 2022, 242: 108320.

[25] Mirjalili S, Mirjalili S M, Hatamlou A. Multi-verse optimizer: a nature-inspired algorithm for global optimization[J]. Neural Computing and Applications, 2016, 27: 495-513.

[26] Seyyedabbasi A, Kiani F. Sand Cat swarm optimization: A nature-inspired algorithm to solve global optimization problems[J]. Engineering with computers, 2023, 39(4): 2627-2651.

[27] Oladejo S O, Ekwe S O, Mirjalili S. The Hiking Optimization Algorithm: A novel human-based metaheuristic approach[J]. Knowledge-Based Systems, 2024, 296: 111880.

[28] Wang H, Tang J, Pan Q. MSI-HHO: Multi-strategy improved HHO algorithm for global optimization[J]. Mathematics, 2024, 12(3): 415.

[29] Mabdeh A N, A** R S, Razavi-Termeh S V, et al. Enhancing the performance of machine learning and deep learning-based flood susceptibility models by integrating grey wolf optimizer (GWO) algorithm[J]. Remote Sensing, 2024, 16(14): 2595.

[30] 王永贵, 赵炀, 邹赫宇, 等. 多策略融合的蛇优化算法及其应用[J] .计算机应用研究, 2024, 41(01): 134-141.

**作者简介：**

刘俊毅（2001-），男，硕士，助教助理，研究方向：算法设计、机器人路径规划、机器人控制系统

**基金项目：**

国家自然科学基金面上项目（5102290）；江苏省自然科学研究面上项目（20KJB530008）；中国智慧工程研究会十四五重点课题项目（ZHGC104432）；中国设备管理协会建筑工程十四五重点课题项目（GMZY2174）；国家科学信息技术部研究中心“十四五”全国科学技术发展研究规划重点课题（KXJS71057）；农业部“十四五”国家科技支撑计划重点课题（NYF251050）；教育部“十四五”教育科研规划重点课题项目（JXKY24391）。


Copyright © This work is licensed under a Commons Attibution-Non Commercial 4.0 International License.