

多策略改进的徒步优化算法及其应用^{*}

徐 明 王凤富 龙 文

(贵州财经大学数学与统计学院 贵阳 550025)

摘 要: 为了解决复杂数值优化问题,提出一种基于柯西逆累积分布算子和随机差分变异策略改进的徒步优化算法。该算法使用佳点集初始化种群,以此增加种群多样性;采用柯西逆累积分布算子,平衡全局搜索与局部开发能力;引入随机差分变异策略,降低过早陷入局部最优的风险。实验结果显示,该算法在 CEC2017 测试集上的平均性能优于 8 种对比算法。统计检验进一步证实了性能差异具有显著性。同时,从 CEC2017 测试集中选取 9 个有代表性的测试函数,通过对比试验,分别验证了该算法中三种改进策略的有效性。此外,将该算法应用到光伏模型参数辨识中,实现了较小的均方根误差 2.43×10^{-3} ,为所有比较算法中的最优值。在另外两类工程设计问题中,该算法均取得了最小目标函数值,优于对比算法。综上所述,改进的徒步优化算法在全局搜索能力、收敛速度和精度方面表现出色,有效提升了解决复杂数值优化问题的性能。

关键词: 徒步优化算法;佳点集;柯西逆累积分布算子;随机差分;光伏模型

中图分类号: TP18;TN02 **文献标识码:** A **国家标准学科分类代码:** 510.1040

Multiple strategies improved hiking optimization algorithm
and its application

Xu Ming Wang Fengfu Long Wen

(School of Mathematics and Statistics, Guizhou University of Finance and Economics, Guiyang 550025, China)

Abstract: To tackle complex numerical optimization problems, this paper proposes an improved hiking optimization algorithm based on Cauchy distribution operator and random differential mutation strategy (CDHOA). The algorithm enhances diversity through effective population initialization, balances global search with local exploitation using the inverse cumulative Cauchy distribution operator, and employs a random differential mutation strategy to boost exploitation and reduce local optima risks. Experimental results show the average performance of CDHOA on the CEC2017 test set is better than that of eight comparison algorithms. The statistical test further confirmed that the performance difference was significant. Nine representative test functions are selected from the CEC2017 test set, and the effectiveness of the three enhancement strategies in the algorithm is verified by comparative experiments. Additionally, it is applied to the parameter identification of photovoltaic model, and a small root mean square error of 2.43×10^{-3} is achieved, which has the best result of all comparison algorithms. In two kinds of engineering design problems, the algorithm achieves the minimum objective function value, which is better than the comparison algorithms. Overall, CDHOA performs well in global search ability, convergence speed and accuracy, which effectively improves the performance of solving complex numerical optimization problems.

Keywords: hiking optimization algorithm; good point set; Cauchy inverse cumulative distribution operator; stochastic differential; photovoltaic model

0 引 言

随着科学技术的不断发展,目标函数的形式趋于多样

化,具有高维、非线性、不可导等性质,传统的优化算法已不能满足求解复杂优化问题的需求,群智能算法因其强大的适用性得到大力发展,吸引无数学者开发新算法以及对经

收稿日期:2024-10-11

^{*} 基金项目:国家自然科学基金(12361106)、贵州省科技计划重点项目(黔科合基础-ZK[2023]重点 003)、贵州省高层次创新型人才项目(黔科合平台人才-GCC[2023]006)资助

典算法经典的改进。如为提升粒子群优化算法全局优化能力,提出的速度暂停粒子群优化算法(velocity pausing particle swarm optimization, VPPSO)^[1],受经典差分进化算法启发提出的高级差分进化算法(advanced differential evolution, ADE)^[2],受北极海鹦生存和捕食行为启发的北极海鹦优化算(arctic puffin optimization, APO)^[3]等。这些智能优化算法的不断提出和改进,使得现实生活中的复杂问题得以解决,如:障碍路径规划^[4]、智能环境中说话人跟踪问题^[5]、水下无人航行器勘察任务规划问题^[6]。

徒步优化算法(hiking optimization algorithm, HOA)是 Oladejo 等^[7]于 2024 年提出的一种新的启发式算法,该算法通过对徒步旅行者试图登顶山峰、丘陵或岩石的经历进行建模。与灰狼优化算法(grey wolf optimizer, GWO)^[8]等算法相比,HOA 在基准测试函数上具有较强的竞争力,在工程应用上有着的优越的性能。文献[9]将徒步优化算法与动态模态分解的优化极限学习机相结合,以此来实现水电机组故障诊断,实验表明该方法有极高的准确率。然而,HOA 与其他群智能算法一样,在解决复杂问题时存在易陷入局部最优、精度不高和收敛慢的问题。因此,本文提出一种改进的徒步优化算法(Cauchy distribution operator and random differential mutation hiking optimization algorithm, CDHOA),主要运用了以下方法:引入佳点集策略,增加种群多样性;加入逆累积柯西分布算子,平衡全局搜索与局部开发能力;利用随机差分变异策略,降低了过早陷入局部最优的风险。通过对 CEC2017 测试集中 29 个基准函数进行实验,结果表明,CDHOA 具有更高的寻优速度与精度。对改进策略的有效性进行分析,证明本文所用单一策略的有效性。非参数统计检验结果也显示了 CDHOA 算法与其他 8 种对比算法的差异具有显著性。最后,应用在求解光伏参数辨识模型^[10]、压力容器设计问题和三杆桁架设计问题^[11]上,进一步证明了本文算法在性能上具有明显的优势。

1 基本徒步优化算法

HOA 的灵感源自于徒步旅行者试图登顶山峰、丘陵或岩石的经历,是基于美籍瑞士籍地理学家和制图师 Waldo Tobler 提出的著名 Tobler 徒步函数(THF)^[12]。该函数作为一个指数模型,用以考量地形或路径的陡峭程度,从而确定徒步旅行者的行进速度。

Tobler 徒步函数为:

$$W_{i,t} = 6e^{-3.5|S_{i,t}+0.05|} \quad (1)$$

$$S_{i,t} = \frac{dh}{dx} = \tan\theta_{i,t} \quad (2)$$

式中: $W_{i,t}$ 为徒步者 i 在 t 时的速度, $S_{i,t}$ 为徒步者 i 在 t 时所处地形的陡峭程度, dh 、 dx 分别表示海拔与徒步者所走路程之差,倾斜角 $\theta_{i,t} \in [0^\circ, 50^\circ]$ 。

HOA 融合了徒步旅行者群体的社会思维以及个体徒

步旅行者的认知能力。而徒步者的速度则由 Tobler 徒步函数来决定,该函数受领队徒步者、当前徒步者的位置以及扫描因子的影响。函数表达式:

$$W_{i,t} = W_{i,t-1} + C(X_{best} - \alpha X_{i,t}) \quad (3)$$

式中:控制系数是 $C \in [0,1]$ 的随机数。 X_{best} 为领队徒步者, $X_{i,t}$ 为当前徒步者。 α 为扫描因子 $\alpha \in [1,3]$ 的随机整数, SF 确保徒步旅行者不会偏离领队太远,以便他们能够看到领队徒步旅行者的方向,并可以接收领队徒步旅行者的信号。

徒步者位置更新公式为:

$$X_{i,t+1} = X_{i,t} + W_{i,t} \quad (4)$$

HOA 的详细流程参考文献[7]。

2 改进的徒步优化算法

基本的 HOA 存在容易陷入局部最优、收敛速度较慢以及收敛精度较低等缺陷。这些缺陷在处理复杂问题时尤为明显,因为复杂问题往往具有更复杂的解空间,从而增加了算法寻找全局最优解的难度。因此,本文针对性地提出 3 种策略改进 HOA。值得注意的是,这 3 种策略的实施并未增加原算法的计算复杂度,而是专注于增强算法在解决实际问题时的效能。

2.1 基于佳点集的种群初始化

原算法采用随机初始化方法产生初始种群,该方法无法保证初始种群解的多样性。因此,本文引用佳点集来产生初始种群,确保生成的初始种群在解空间中分布更加均匀,从而避免随机初始化的点在某些区域过于聚集而其他区域过于稀疏的问题。

佳点集理论是由我国著名数学家华罗庚提出的,该理论描述了一种在多维空间中能够均匀分布的点集。

在欧式空间中 R^D 的单位立方体内,如果点集 $P_n(k)$ 中的点 r 满足特定的分布性质,即其偏差满足 $\phi(n) = C(r, \epsilon)n^{-1+\epsilon}$, 式中: $C(r, \epsilon)$ 是与点 r 和 ϵ 正小数有关的常数, $P_n(k)$ 则称为佳点集, r 为佳点。通常以 $r_k = 2\pi\cos(2\pi j/p)$ 来选取佳点,其中 p 是满足 $(p-3)/2 \geq D$ 的最小素数。以 100 个二维的佳点集与随机方法取得的随机点比较,如图 1 所示。

图 1(a)为佳点集所产生的初始化种群,图 1(b)为随机点所产生的初始化种群,对比可以看出,佳点集所产生的二维种群相较于随机分布产生的种群分布更均匀,这使得算法在开始的搜索过程中能够更全面地探索解空间,从而减少遗漏潜在最优解的可能性。在维度相同的情况下,佳点集产生的种群具有更强的稳定性。由于佳点集初始化减少了随机性的影响,算法在不同运行条件下表现出更强的稳定性和可重复性。因此,该方法提高了种群的多样性,有助于算法获得更好的全局寻优效果。

2.2 柯西逆累积分布算子

柯西分布,又称柯西——洛伦兹分布,是一种连续概率

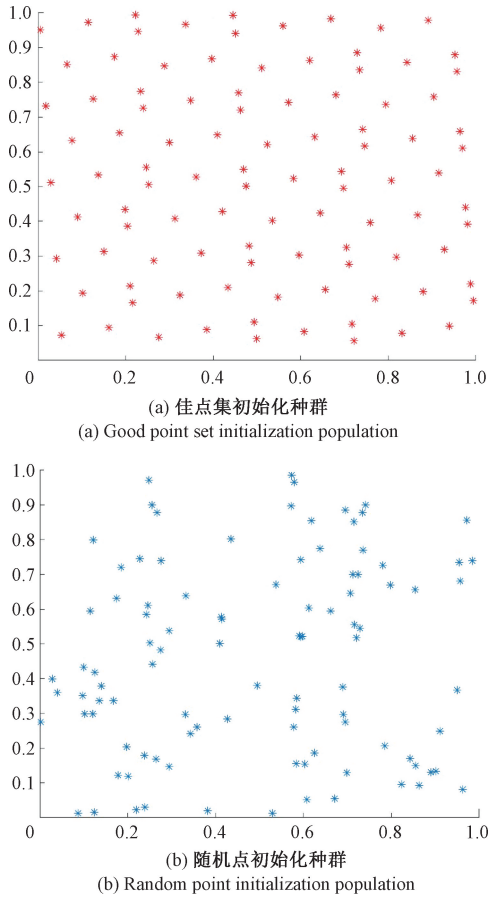


图 1 佳点集与随机点初始化种群对比图

Fig. 1 Comparison chart of good point set and random point initialization population

分布的函数。这种分布是由法国数学家奥古斯丁·路易·柯西(Augustin-Louis Cauchy)^[13]在 19 世纪提出的。其概率密度函数为:

$$f(x) = \frac{1}{\pi} \left[\frac{b}{(x-a)^2 + b^2} \right] \quad (5)$$

当 $a = 0, b = 1$ 时,为标准柯西分布,其累积分布函数为:

$$f(x) = \frac{1}{\pi} \arctan\left(\frac{x-a}{b}\right) + \frac{1}{2} \quad (6)$$

对应的逆函数为:

$$f(x)^{-1} = a + b \tan(\pi(r1 - 0.5)) \quad (7)$$

式中: $r1$ 为 $[0, 1]$ 的随机数,本文中 $a = 0$ 。由于其运动轨迹具有显著的不规则性,这种特性使得算法在搜索过程中能够避免过早地局限于某个局部区域,从而不易陷入局部最优解的困境。这种不规则的运动模式为算法提供了更多的探索机会,有助于在更广泛的解空间内进行搜索,进而提高了找到全局最优解的可能性。同时为了不使个体步行者不偏离范围太远,压缩倍数 $b = 0.01$ 。

改进后的式(4)为:

$$W_{i,t} = W_{i,t-1} + 0.01 \times \tan(\pi(r1 - 0.5)) \times (X_{best} - \alpha X_{i,t}) \quad (8)$$

2.3 随机差分变异策略

在原 HOA 的中,种群个体的更新方式依赖于最优个体。如果最优个体陷入局部最优,那么随着迭代的进行,种群个体会逐渐向这个局部最优的个体聚集,从而导致整个算法陷入局部收敛。鉴于此,CDHOA 融合差分进化算法^[14]中的随机差分变异策略,使算法能够跳出局部最优,平衡全局搜索与局部搜索能力。位置更新方式为:

$$X_{i,t+1} = X_{i,t} + r \cdot (X_{best} - X_{i,t}) + r \cdot (X_{r2} - X_{r3}) \quad (9)$$

式中: r 为 $[0, 1]$ 的随机数, $r2, r3$ 为 $[1, N]$ 的随机整数。

每次迭代都对个体随机的使用随机差分变异,产生新的个体,以此平衡算法的开发与探索,保持种群多样性,提高算法的全局搜索能力。

CDHOA 伪代码为:

算法 1 CDHOA

输入:种群数 N ,迭代总数 T ,候选解上、下界 ub, lb ,变量个数 dim
输出:最佳徒步者个体 $best$
1 初始化徒步者种群
2 计算目标函数值
3 for 1 to T do
4 找出领队徒步者
5 for 1 to N do
6 if $rand < 0.5$
7 计算 $S_{i,t}, W_{i,t}$
8 式(9)更新 $X_{i,t+1}$
9 else
10 式(10)更新 $X_{i,t+1}$
11 end if
12 计算 $X_{i,t+1}$ 目标函数值
13 if $F(X_{i,t+1}) < F(X_{i,t})$
14 更新目标函数值以及徒步者位置
15 end if
16 end for
17 end for

3 数值实验结果与分析

为了验证 CDHOA 的性能,本文仿真实验选取 CEC2017 测试集 29 个基准函数进行测试。其中 CEC2017 测试集 29 个基准函数涵盖了单峰函数($F1, F3$)、多峰函数($F4 \sim F10$)、混合函数($F11 \sim F19$)以及复合函数($F20 \sim F29$)4 种类型的函数。该测试集函数的名称、理论最优值可以参考文献^[15]中的表 1。

分别对 CEC2017 测试集中的 29 个基准函数进行 10 维和 30 维数值实验,利用 CDHOA 对上述优化问题进行求解,并与 HOA、ADE、VPPSO、正余弦优化算法(sine

表 1 CDHOA 与其他 8 种算法对 29 个测试函数的结果比 (Dim=10)

Table 1 Comparisons of CDHOA and other eight algorithms for 29 test function with Dim=10

函数	统计	CDHOA	HOA	ADE	VPPSO	SCA	DO	DBO	WOA	SSA
F1	平均值	1.00×10^0	1.86×10^9	1.55×10^4	3.64×103	7.69×10^8	4.47×10^3	3.22×10^5	2.46×10^6	3.74×10^3
	标准差	1.69×10^{-1}	1.06×10^9	1.45×10^4	2.82×103	2.80×10^8	3.19×10^3	1.44×10^6	4.44×10^6	3.51×10^3
F3	平均值	3.00×10^2	4.59×10^3	7.39×10^3	3.00×10^2	1.59×10^3	3.00×10^2	3.01×10^2	9.24×10^2	3.00×10^2
	标准差	1.04×10^{-14}	1.85×10^3	2.76×10^3	2.89×10^{-9}	9.43×10^2	8.50×10^{-3}	2.79×10^0	4.76×10^2	1.07×10^{-3}
F4	平均值	4.00×10^2	5.06×10^2	4.07×10^2	4.07×10^2	4.49×10^2	4.05×10^2	4.12×10^2	4.39×10^2	4.06×10^2
	标准差	1.72×10^{-7}	5.43×10^1	7.11×10^{-1}	1.16×10^1	3.14×10^1	2.36×10^0	2.86×10^1	4.98×10^1	1.19×10^1
F5	平均值	5.07×10^2	5.36×10^2	5.18×10^2	5.24×10^2	5.48×10^2	5.24×10^2	5.30×10^2	5.49×10^2	5.31×10^2
	标准差	2.52×10^0	1.16×10^1	2.99×10^0	8.39×10^0	7.78×10^0	7.31×10^0	8.48×10^0	1.80×10^1	1.37×10^1
F6	平均值	6.00×10^2	6.23×10^2	6.00×10^2	6.07×10^2	6.17×10^2	6.05×10^2	6.02×10^2	6.34×10^2	6.05×10^2
	标准差	1.34×10^{-3}	7.25×10^0	5.20×10^{-5}	6.00×10^0	2.91×10^0	5.41×10^0	2.23×10^0	1.53×10^1	5.76×10^0
F7	平均值	7.19×10^2	7.50×10^2	7.32×10^2	7.48×10^2	7.76×10^2	7.45×10^2	7.29×10^2	7.74×10^2	7.71×10^2
	标准差	2.89×10^0	1.27×10^1	3.82×10^0	1.34×10^1	1.11×10^1	1.83×10^1	8.95×10^0	2.83×10^1	2.56×10^1
F8	平均值	8.06×10^2	8.23×10^2	8.19×10^2	8.23×10^2	8.38×10^2	8.25×10^2	8.23×10^2	8.39×10^2	8.27×10^2
	标准差	1.87×10^0	6.88×10^0	3.20×10^0	7.91×10^0	6.44×10^0	1.19×10^1	1.03×10^1	1.44×10^1	1.03×10^1
F9	平均值	9.00×10^2	1.05×10^3	9.00×10^2	9.15×10^2	9.99×10^2	9.02×10^2	9.02×10^2	1.38×10^3	1.22×10^3
	标准差	3.34×10^{-2}	9.42×10^1	7.51×10^{-4}	2.19×10^1	3.78×10^1	8.12×10^0	7.04×10^0	2.75×10^2	3.23×10^2
F10	平均值	1.50×10^3	2.02×10^3	1.69×10^3	1.69×10^3	2.30×10^3	1.70×10^3	1.72×10^3	2.07×10^3	1.79×10^3
	标准差	1.44×10^2	3.07×10^2	1.25×10^2	2.40×10^2	1.88×10^2	2.80×10^2	2.56×10^2	2.81×10^2	3.74×10^2
F11	平均值	1.10×10^3	1.30×10^3	1.11×10^3	1.14×10^3	1.20×10^3	1.12×10^3	1.15×10^3	1.21×10^3	1.12×10^3
	标准差	1.53×10^0	2.06×10^2	1.77×10^0	1.78×10^1	3.94×10^1	9.97×10^0	6.96×10^1	7.49×10^1	1.33×10^1
F12	平均值	1.65×10^3	3.38×10^6	1.00×10^6	3.99×10^5	1.64×10^7	2.44×10^5	4.56×10^5	4.44×10^6	1.22×10^4
	标准差	2.11×10^2	3.13×10^6	6.88×10^5	5.36×10^5	1.36×10^7	3.05×10^5	1.69×10^6	4.86×10^6	1.02×10^4
F13	平均值	1.31×10^3	1.28×10^4	7.00×10^3	1.19×10^4	2.85×10^4	1.19×10^4	1.37×10^4	2.05×10^4	9.40×10^3
	标准差	6.75×10^0	6.99×10^3	6.48×10^3	8.05×10^3	2.10×10^4	8.03×10^3	1.28×10^4	1.63×10^4	8.62×10^3
F14	平均值	1.41×10^3	3.03×10^3	2.04×10^3	1.51×10^3	1.86×10^3	1.49×10^3	1.51×10^3	1.90×10^3	1.54×10^3
	标准差	8.31×10^0	1.86×10^3	1.46×10^3	3.77×10^1	8.92×10^2	1.09×10^2	5.09×10^1	8.67×10^2	9.46×10^1
F15	平均值	1.51×10^3	5.83×10^3	2.35×10^3	2.27×10^3	2.43×10^3	1.80×10^3	2.40×10^3	5.98×10^3	2.08×10^3
	标准差	1.19×10^1	2.65×10^3	1.41×10^3	8.45×10^2	8.91×10^2	4.14×10^2	1.54×10^3	4.50×10^3	7.21×10^2
F16	平均值	1.60×10^3	1.87×10^3	1.63×10^3	1.76×10^3	1.74×10^3	1.73×10^3	1.72×10^3	1.86×10^3	1.76×10^3
	标准差	3.32×10^0	1.17×10^2	2.28×10^1	9.84×10^1	7.70×10^1	1.22×10^2	7.72×10^1	1.01×10^2	1.46×10^2
F17	平均值	1.73×10^3	1.77×10^3	1.71×10^3	1.77×10^3	1.78×10^3	1.76×10^3	1.75×10^3	1.79×10^3	1.76×10^3
	标准差	4.95×10^0	2.20×10^1	1.06×10^1	3.24×10^1	1.05×10^1	3.00×10^1	2.33×10^1	4.03×10^1	3.81×10^1
F18	平均值	1.86×10^3	1.10×10^6	1.03×10^4	1.85×10^4	1.06×10^5	1.85×10^4	1.33×10^4	1.82×10^4	8.75×10^3
	标准差	7.74×10^1	5.87×10^6	6.34×10^3	1.17×10^4	8.06×10^4	1.32×10^4	1.10×10^4	1.23×10^4	7.67×10^3
F19	平均值	1.90×10^3	1.08×10^4	3.35×10^3	3.40×10^3	4.76×10^3	3.40×10^3	4.32×10^3	2.33×10^4	3.01×10^3
	标准差	3.32×10^0	1.09×10^4	2.98×10^3	4.35×10^3	3.89×10^3	3.07×10^3	7.68×10^3	2.30×10^4	1.48×10^3
F20	平均值	2.02×10^3	2.10×10^3	2.00×10^3	2.11×10^3	2.11×10^3	2.06×10^3	2.05×10^3	2.15×10^3	2.08×10^3
	标准差	4.56×10^0	5.12×10^1	5.19×10^{-1}	5.40×10^1	2.78×10^1	4.05×10^1	4.03×10^1	8.16×10^1	6.19×10^1
F21	平均值	2.27×10^3	2.30×10^3	2.29×10^3	2.26×10^3	2.26×10^3	2.30×10^3	2.22×10^3	2.32×10^3	2.30×10^3
	标准差	5.07×10^1	5.78×10^1	3.62×10^1	6.72×10^1	6.25×10^1	5.21×10^1	3.76×10^1	4.95×10^1	6.27×10^1
F22	平均值	2.30×10^3	2.42×10^3	2.30×10^3	2.30×10^3	2.36×10^3	2.30×10^3	2.30×10^3	2.32×10^3	2.30×10^3
	标准差	7.45×10^{-1}	1.69×10^2	8.20×10^0	2.04×10^1	1.65×10^1	1.02×10^1	1.11×10^1	1.31×10^1	1.79×10^1
F23	平均值	2.61×10^3	2.68×10^3	2.62×10^3	2.62×10^3	2.65×10^3	2.64×10^3	2.63×10^3	2.65×10^3	2.64×10^3

续表 1

Continuation table 1

函数	统计	CDHOA	HOA	ADE	VPPSO	SCA	DO	DBO	WOA	SSA
F24	标准差	2.52×10⁰	1.78×10 ¹	4.05×10 ⁰	9.44×10 ⁰	6.89×10 ⁰	1.20×10 ¹	7.71×10 ⁰	1.49×10 ¹	1.73×10 ¹
	平均值	2.70×10³	2.77×10 ³	2.74×10 ³	2.73×10 ³	2.73×10 ³	2.75×10 ³	2.74×10 ³	2.75×10 ³	2.75×10 ³
F25	标准差	8.01×10 ¹	8.86×10 ¹	3.03×10¹	7.73×10 ¹	9.59×10 ¹	6.94×10 ¹	7.30×10 ¹	8.10×10 ¹	8.54×10 ¹
	平均值	2.92×10 ³	3.00×10 ³	2.93×10 ³	2.91×10 ³	2.96×10 ³	2.91×10³	2.94×10 ³	2.96×10 ³	2.93×10 ³
F26	标准差	2.27×10 ¹	5.41×10 ¹	1.57×10 ¹	2.23×10 ¹	1.23×10¹	6.42×10 ¹	2.25×10 ¹	3.31×10 ¹	2.50×10 ¹
	平均值	2.91×10³	3.35×10 ³	3.01×10 ³	2.98×10 ³	3.09×10 ³	2.98×10 ³	3.03×10 ³	3.22×10 ³	3.04×10 ³
F27	标准差	2.78×10¹	2.57×10 ²	1.22×10 ²	2.10×10 ²	3.46×10 ¹	1.08×10 ²	1.95×10 ²	3.74×10 ²	2.21×10 ²
	平均值	3.09×10³	3.18×10 ³	3.09×10 ³	3.10×10 ³	3.10×10 ³	3.10×10 ³	3.10×10 ³	3.13×10 ³	3.11×10 ³
F28	标准差	1.35×10 ⁰	2.81×10 ¹	8.02×10⁻¹	1.21×10 ¹	1.69×10 ⁰	9.68×10 ⁰	4.51×10 ⁰	3.88×10 ¹	2.84×10 ¹
	平均值	3.40×10 ³	3.53×10 ³	3.33×10 ³	3.30×10 ³	3.27×10 ³	3.26×10³	3.35×10 ³	3.35×10 ³	3.35×10 ³
F29	标准差	5.60×10¹	1.56×10 ²	6.89×10 ¹	1.40×10 ²	5.73×10 ¹	1.37×10 ²	1.15×10 ²	1.16×10 ²	1.29×10 ²
	平均值	3.15×10³	3.25×10 ³	3.18×10 ³	3.22×10 ³	3.23×10 ³	3.22×10 ³	3.25×10 ³	3.30×10 ³	3.25×10 ³
F30	标准差	1.09×10¹	7.61×10 ¹	1.31×10 ¹	5.84×10 ¹	2.72×10 ¹	5.60×10 ¹	7.99×10 ¹	7.16×10 ¹	7.70×10 ¹
	平均值	8.52×10⁴	2.58×10 ⁶	1.40×10 ⁵	1.02×10 ⁵	6.91×10 ⁵	3.42×10 ⁵	3.85×10 ⁵	9.90×10 ⁵	2.30×10 ⁵
Friedman	标准差	2.45×10 ⁵	5.88×10 ⁶	1.44×10⁵	2.57×10 ⁵	4.25×10 ⁵	5.01×10 ⁵	5.32×10 ⁵	1.28×10 ⁶	3.73×10 ⁵
	平均排名	1.62	7.90	3.48	3.83	6.86	4.00	4.62	7.79	4.90
最终排名		1	9	2	3	7	4	5	7	6

cosine algorithm, SCA)^[16]、蒲公英优化算法(dandelion optimizer, DO)^[17]、蜣螂优化算法(dung beetle optimizer, DBO)^[18]、鲸鱼优化算法(whale optimization algorithm, WOA)^[19]、麻雀优化算法(sparrow search algorithm, SSA)^[20]9种算法进行对比。

为了比较的公平性,9种算法均采用相同的适应度评价次数(50 000次),即种群规模 $N=100$ 、最大迭代次数 $T=500$ 。其余算法详细参数均按照原基本算法参数设置。9种算法对单个函数独立运行30次,记录目标函数值的平均值、标准差,并进行Wilcoxon's秩和检验以及Friedman's检验^[21]。最佳算法的结果在表格中均以加粗字体突出

显示。

另外,对CEC2017测试集中的9个测试函数(单峰函数F1,多峰函数F5、F8,混合函数F16、F18、F19,复合函数F22、F23、F28)进行维度 $Dim=30$ 、种群规模 $N=100$ 、最大迭代次数 $T=500$ 的算法改进策略有效性分析。

算法均在MATLAB R2024a软件编程实现,电脑硬件为:Windows 11 64位操作系统,AMD Ryzen 7 7840HS w/Radeon 780M Graphics 3.80 GHz,内存32 GB。

3.1 CDHOA 与其他算法的比较

当维度 $Dim=10$,实验结果的平均值与标准差如表1所示,表2为Wilcoxon秩和检验结果。

表 2 Wilcoxon 检验在 10 维 CEC2017 测试集上的测试结果对比

Table 2 Comparison outcomes attained by Wilcoxon test on the 10-dimensional CEC2017 test suite

函数	HOA	ADE	VPPSO	SCA	DO	DBO	WOA	SSA
单峰函数	1.70×10^{-21}	1.70×10^{-21}	7.76×10^{-21}	1.70×10^{-21}	7.76×10^{-21}	1.66×10^{-19}	1.70×10^{-21}	1.04×10^{-17}
多峰函数	4.85×10^{-3}	6.58×10^{-2}	2.14×10^{-2}	9.77×10^{-3}	2.46×10^{-2}	4.17×10^{-2}	3.77×10^{-3}	8.75×10^{-3}
混合函数	3.88×10^{-46}	4.87×10^{-21}	2.95×10^{-27}	1.65×10^{-32}	7.93×10^{-22}	1.30×10^{-23}	1.43×10^{-38}	1.35×10^{-23}
复合函数	1.95×10^{-5}	8.66×10^{-2}	2.45×10^{-1}	3.28×10^{-2}	1.21×10^{-1}	1.05×10^{-1}	1.82×10^{-3}	4.89×10^{-2}

由表1可知,CDHOA求解CEC2017测试函数的平均值与标准差几乎都优于原算法,根据Friedman检验结果,CDHOA排在第1位,HOA排在第9位。此外,由表2Wilcoxon秩和检验结果可得,CEC2017测试集的基准函数p值均小于5%,说明两者存在明显的差异。

由表1可知,CDHOA对17个基准测试函数(F1、

F3~F5、F7、F8、F11~F16、F18、F19、F23、F26、F29)求解所得最优值的平均值、标准差均优于8种对比算法。在F6、F9、F17、F20、F22、F25函数上最优值的平均值相较于最优算法次之;在F10、F24、F27、F30函数上得到的平均值最优,但标准差不是最小的。在F1、F3、F4、F6、F9、F11、F19上求解所得平均值为对应函数的理论值。Friedman

检验的结果可知,CDHOA 排在第 1 位,ADE 排在第 2 位,VPPSO 排在第 3 位,而 HOA 排在第 9 位。

由表 2 的 Wilcoxon 秩和检验结果可知,CDHOA 与 8 种对比算法在求解单峰函数、混合函数的结果中存在 5% 水平差异;在求解多峰函数上,CDHOA 与 HOA、VPPSO、SCA、DO、WOA、SSA 共 6 种算法在 5% 水平上存在差异,在 10% 水平上与 8 种算法均也存在差异;在求解复合函数上,与 HOA、VPPSO、SCA、WOA、SSA 在 5% 水平上存在差异,在 10% 水平上与其他 8 种算法也存在

差异。

由于篇幅有限,本文仅展示具有代表性的 9 个基准测试(F1、F5、F8、F12、F13、F18、F23、F26、F29)迭代收敛图。如图 2 所示,可以明显看到 CDHOA 的收敛曲线终点均位于最下方,说明在求解函数过程中,CDHOA 的开发和探索能力都得到了显著提升。具体来说,CDHOA 在一定程度上避免了陷入局部最优解,展现出了更强的全局搜索能力,且与其他 8 个算法相比,CDHOA 的收敛速度更快、收敛精度更高,这进一步证明了其在优化问题上的优势。

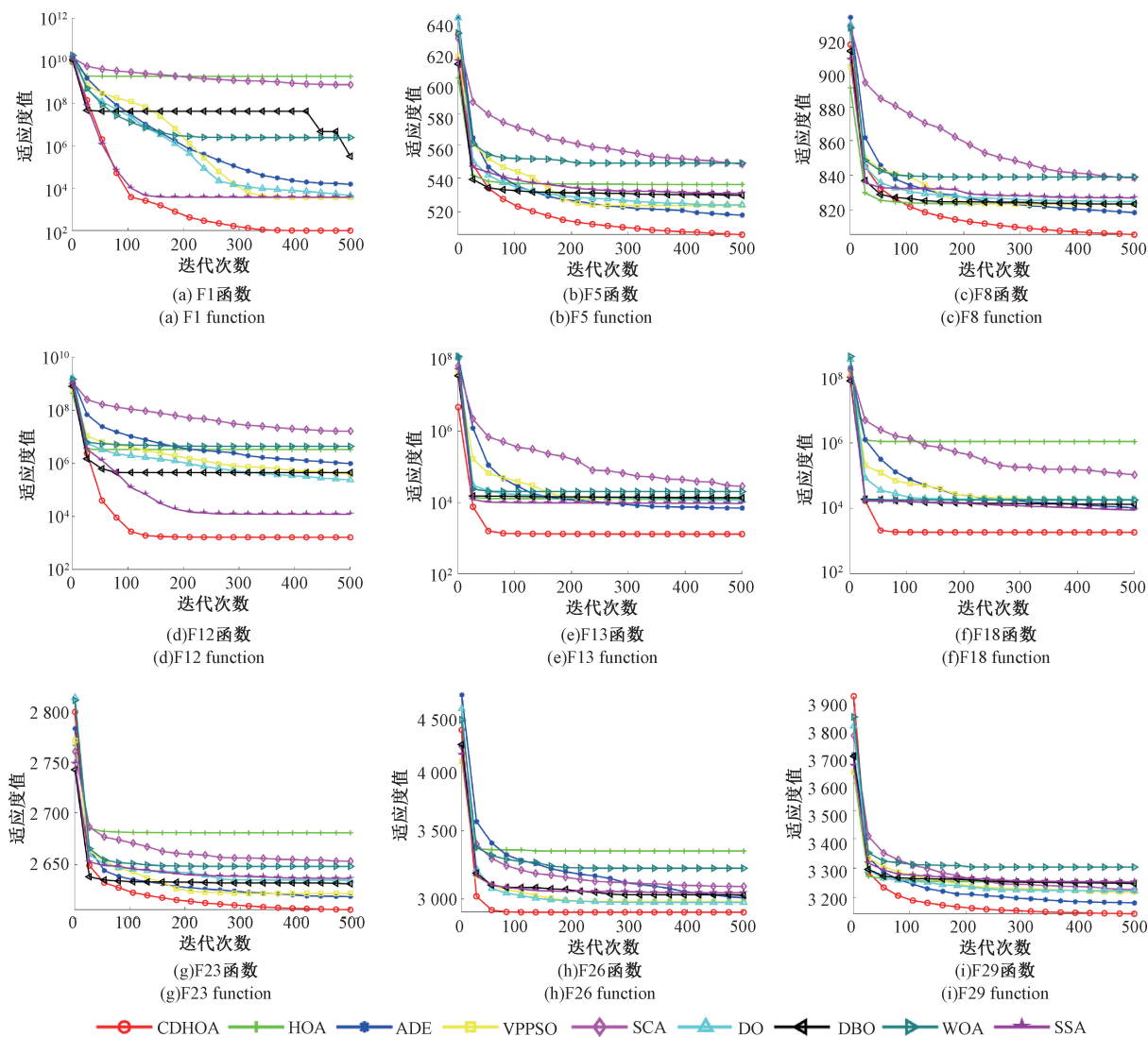


图 2 九种算法对 29 个测试函数的收敛曲线 Dim=10

Fig. 2 Convergence curves of nine algorithms for twenty-nine text functions with Dim=10

由表 3 可以看出,当维度增高到 Dim=30 时,CDHOA 相较于 Dim=10 的收敛效果有所下降,但同其余 8 个对比算法相比,CDHOA 仍有很大优势,且比 HOA 有更好的求解性能。在 11 个基准测试函数(F12~F15、F17~F20、F22、F29、F30)上,求解所得结果平均值及标准差均优于 8 种对比算法;在求解 F5、F7、F8、F16、F21、F23、F24 上所得

平均值优于 8 种对比算法;在 F1、F4、F6、F9、F25~F28 上求解所得平均值次之。Friedman 检验结果显示 CDHOA 排在第 1 位,DO 排在第 2 位,VPPSO 排在第 3 位,HOA 排在第 7 位。

表 4 的 Wilcoxon 秩和检验结果显示,有 87.5% 的数据存在明显差异,获得的 p 值小于 5%。

表 3 CDHOA 与 8 种算法对 CEC2017 中 29 个测试函数的结果比(Dim=30)

Table 3 Comparisons of CDHOA and other eight algorithms for 29 test functions in CEC2017 with Dim=30

函数	统计	CDHOA	HOA	ADE	VPPSO	SCA	DO	DBO	WOA	SSA
F1	平均值	1.03×10^6	2.71×10^{10}	5.90×10^6	7.90×10^3	1.61×10^{10}	1.10×10^5	3.99×10^7	5.52×10^8	8.48×10^3
	标准差	7.98×10^5	5.86×10^9	3.36×10^6	7.59×10^3	2.39×10^9	9.33×10^4	5.85×10^7	2.37×10^8	5.91×10^3
F3	平均值	3.41×10^3	5.99×10^4	1.53×10^5	3.38×10^4	6.11×10^4	2.65×10^3	9.12×10^4	2.34×10^5	4.77×10^4
	标准差	1.67×10^3	7.64×10^3	2.37×10^4	9.00×10^3	1.35×10^4	1.51×10^3	2.22×10^4	5.48×10^4	7.04×10^3
F4	平均值	5.01×10^2	4.54×10^3	5.17×10^2	5.07×10^2	2.02×10^3	5.03×10^2	5.70×10^2	7.02×10^2	4.99×10^2
	标准差	2.64×10^1	1.18×10^3	1.58×10^1	1.97×10^1	5.53×10^2	1.82×10^1	8.92×10^1	8.56×10^1	2.31×10^1
F5	平均值	6.35×10^2	7.49×10^2	6.99×10^2	6.62×10^2	8.15×10^2	6.63×10^2	6.99×10^2	8.07×10^2	7.47×10^2
	标准差	2.46×10^1	3.85×10^1	1.08×10^1	3.85×10^1	2.43×10^1	3.16×10^1	5.69×10^1	5.86×10^1	5.10×10^1
F6	平均值	6.06×10^2	6.54×10^2	6.01×10^2	6.42×10^2	6.58×10^2	6.41×10^2	6.23×10^2	6.71×10^2	6.51×10^2
	标准差	2.78×10^0	7.04×10^0	2.33×10^{-1}	9.12×10^0	5.11×10^0	1.11×10^1	7.02×10^0	1.12×10^1	9.68×10^0
F7	平均值	9.08×10^2	1.14×10^3	9.50×10^2	9.58×10^2	1.18×10^3	9.81×10^2	9.62×10^2	1.26×10^3	1.22×10^3
	标准差	2.74×10^1	6.43×10^1	1.55×10^1	6.10×10^1	4.03×10^1	6.67×10^1	8.04×10^1	6.30×10^1	1.06×10^2
F8	平均值	9.07×10^2	1.02×10^3	9.91×10^2	9.13×10^2	1.07×10^3	9.47×10^2	1.00×10^3	1.03×10^3	9.70×10^2
	标准差	2.77×10^1	2.85×10^1	1.22×10^1	2.74×10^1	2.19×10^1	3.47×10^1	4.57×10^1	4.24×10^1	3.31×10^1
F9	平均值	2.16×10^3	5.27×10^3	2.06×10^3	3.74×10^3	6.77×10^3	5.04×10^3	6.05×10^3	1.00×10^4	5.32×10^3
	标准差	8.09×10^2	1.04×10^3	5.54×10^2	9.10×10^2	1.41×10^3	1.66×10^3	1.92×10^3	3.06×10^3	2.93×10^2
F10	平均值	6.06×10^3	6.89×10^3	7.33×10^3	4.62×10^3	8.54×10^3	4.99×10^3	5.19×10^3	6.67×10^3	5.36×10^3
	标准差	4.77×10^2	7.28×10^2	3.30×10^2	7.30×10^2	2.78×10^2	7.16×10^2	7.47×10^2	8.53×10^2	5.96×10^2
F11	平均值	1.27×10^3	3.79×10^3	2.06×10^3	1.32×10^3	2.59×10^3	1.25×10^3	1.53×10^3	4.68×10^3	1.26×10^3
	标准差	5.83×10^1	1.08×10^3	5.45×10^2	4.21×10^1	5.06×10^2	4.23×10^1	1.43×10^2	2.10×10^3	6.55×10^1
F12	平均值	7.14×10^5	3.29×10^9	2.27×10^7	1.20×10^7	1.90×10^9	5.91×10^6	3.19×10^7	1.33×10^8	1.42×10^6
	标准差	6.64×10^5	1.00×10^9	1.30×10^7	9.24×10^6	6.08×10^8	3.45×10^6	5.37×10^7	7.40×10^7	1.08×10^6
F13	平均值	1.34×10^4	6.00×10^8	1.89×10^6	1.04×10^5	7.84×10^8	1.11×10^5	2.49×10^6	5.55×10^5	1.70×10^4
	标准差	1.42×10^4	4.75×10^8	1.94×10^6	5.28×10^4	2.70×10^8	7.50×10^4	6.34×10^6	4.24×10^5	1.83×10^4
F14	平均值	1.58×10^3	1.06×10^6	5.37×10^5	6.45×10^4	4.53×10^5	4.41×10^4	1.11×10^5	2.25×10^6	5.25×10^4
	标准差	5.50×10^1	5.36×10^5	4.50×10^5	4.34×10^4	2.87×10^5	3.43×10^4	1.19×10^5	2.40×10^6	5.37×10^4
F15	平均值	2.15×10^3	1.44×10^7	7.63×10^5	4.95×10^4	3.60×10^7	4.86×10^4	6.74×10^4	1.99×10^5	9.14×10^3
	标准差	1.96×10^2	2.82×10^7	6.50×10^5	3.71×10^4	3.03×10^7	2.71×10^4	5.00×10^4	2.44×10^5	9.80×10^3
F16	平均值	2.39×10^3	3.75×10^3	2.99×10^3	2.83×10^3	3.88×10^3	2.68×10^3	3.09×10^3	3.86×10^3	2.91×10^3
	标准差	2.44×10^2	3.37×10^2	1.59×10^2	2.62×10^2	2.03×10^2	2.61×10^2	3.51×10^2	4.60×10^2	3.80×10^2
F17	平均值	1.94×10^3	2.51×10^3	2.27×10^3	2.23×10^3	2.61×10^3	2.22×10^3	2.42×10^3	2.72×10^3	2.57×10^3
	标准差	7.99×10^1	2.29×10^2	1.04×10^2	2.07×10^2	1.56×10^2	2.33×10^2	2.08×10^2	3.16×10^2	2.52×10^2
F18	平均值	6.24×10^3	5.69×10^6	2.28×10^6	6.83×10^5	8.08×10^6	8.11×10^5	2.20×10^6	4.53×10^6	4.61×10^5
	标准差	7.61×10^3	4.96×10^6	1.20×10^6	7.33×10^5	7.44×10^6	6.60×10^5	3.71×10^6	5.07×10^6	4.99×10^5
F19	平均值	2.16×10^3	8.80×10^6	5.25×10^5	6.92×10^5	5.49×10^7	1.15×10^5	8.50×10^5	9.59×10^6	1.31×10^4
	标准差	1.70×10^2	1.32×10^7	6.45×10^5	4.53×10^5	3.18×10^7	1.29×10^5	2.25×10^6	6.67×10^6	1.17×10^4
F20	平均值	2.31×10^3	2.68×10^3	2.54×10^3	2.50×10^3	2.81×10^3	2.61×10^3	2.63×10^3	2.88×10^3	2.81×10^3
	标准差	7.91×10^1	2.28×10^2	1.02×10^2	1.54×10^2	1.47×10^2	2.04×10^2	2.27×10^2	2.01×10^2	1.82×10^2
F21	平均值	2.41×10^3	2.55×10^3	2.49×10^3	2.45×10^3	2.58×10^3	2.45×10^3	2.48×10^3	2.59×10^3	2.51×10^3
	标准差	2.66×10^1	3.24×10^1	1.21×10^1	3.17×10^1	1.62×10^1	3.96×10^1	2.92×10^1	5.79×10^1	4.94×10^1
F22	平均值	2.31×10^3	6.73×10^3	5.45×10^3	4.38×10^3	8.83×10^3	4.79×10^3	5.58×10^3	7.55×10^3	5.68×10^3
	标准差	3.55×10^0	1.61×10^3	1.04×10^3	2.03×10^3	2.33×10^3	1.95×10^3	2.00×10^3	1.43×10^3	1.95×10^3
F23	平均值	2.79×10^3	3.30×10^3	2.84×10^3	2.83×10^3	3.04×10^3	2.87×10^3	2.88×10^3	3.10×10^3	2.92×10^3
	标准差	3.72×10^1	8.78×10^1	1.47×10^1	4.52×10^1	4.52×10^1	6.59×10^1	6.80×10^1	1.00×10^2	5.59×10^1
F24	平均值	2.96×10^3	3.54×10^3	3.04×10^3	2.97×10^3	3.20×10^3	3.06×10^3	3.04×10^3	3.19×10^3	3.11×10^3

续表 3

Continuation table 3

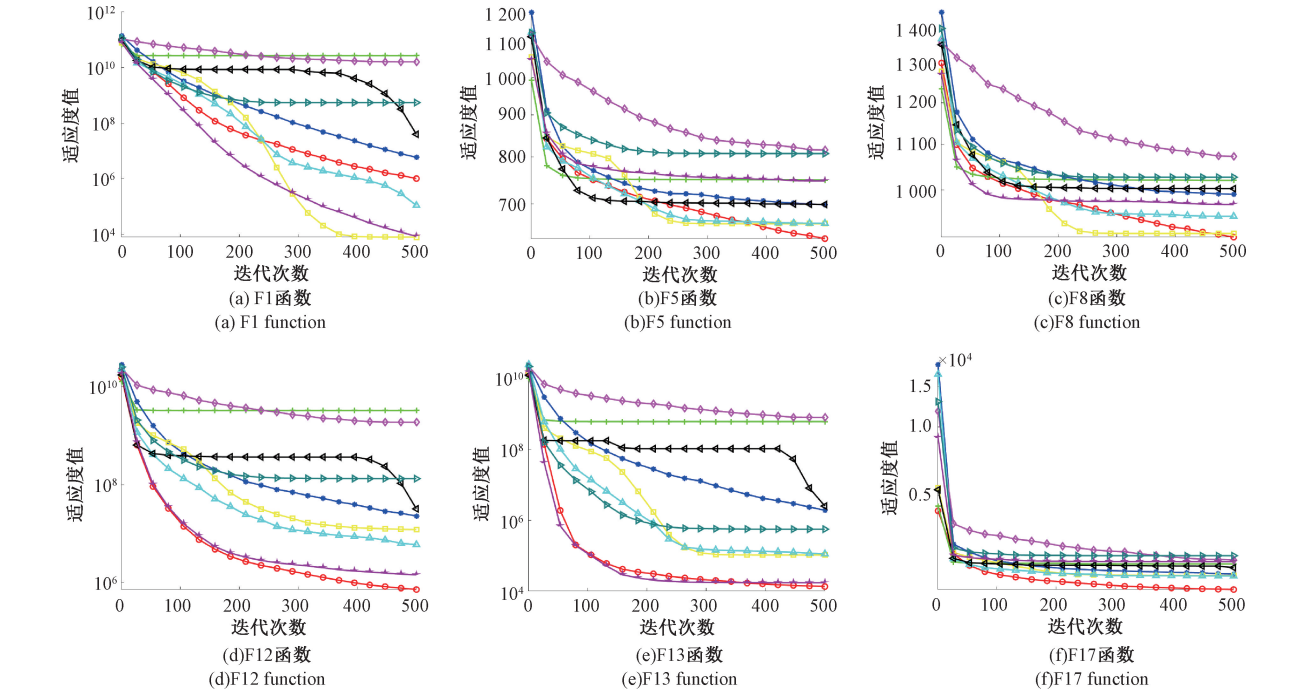
函数	统计	CDHOA	HOA	ADE	VPPSO	SCA	DO	DBO	WOA	SSA
F25	标准差	2.65×10^1	1.14×10^2	1.60×10^1	4.04×10^1	2.95×10^1	5.83×10^1	6.24×10^1	8.83×10^1	8.72×10^1
	平均值	2.90×10^3	3.53×10^3	2.90×10^3	2.92×10^3	3.41×10^3	2.89×10^3	2.94×10^3	3.06×10^3	2.90×10^3
F26	标准差	2.14×10^1	1.33×10^2	8.68×10^0	2.25×10^1	1.54×10^2	1.23×10^1	5.74×10^1	5.04×10^1	1.87×10^1
	平均值	4.75×10^3	8.48×10^3	5.55×10^3	4.72×10^3	7.44×10^3	5.49×10^3	5.78×10^3	7.79×10^3	6.13×10^3
F27	标准差	1.02×10^3	5.04×10^2	1.90×10^2	1.22×10^3	2.74×10^2	1.14×10^3	9.52×10^2	1.41×10^3	1.20×10^3
	平均值	3.24×10^3	3.95×10^3	3.23×10^3	3.29×10^3	3.48×10^3	3.27×10^3	3.27×10^3	3.42×10^3	3.27×10^3
F28	标准差	1.81×10^1	1.91×10^2	7.31×10^0	6.31×10^1	7.03×10^1	2.63×10^1	3.63×10^1	1.03×10^2	4.60×10^1
	平均值	3.24×10^3	5.09×10^3	3.30×10^3	3.29×10^3	4.07×10^3	3.23×10^3	3.37×10^3	3.46×10^3	3.24×10^3
F29	标准差	3.56×10^1	4.51×10^2	1.66×10^1	2.21×10^1	1.78×10^2	2.11×10^1	7.41×10^1	6.37×10^1	2.31×10^1
	平均值	3.67×10^3	5.19×10^3	4.20×10^3	4.21×10^3	5.00×10^3	4.09×10^3	4.11×10^3	5.08×10^3	4.17×10^3
F30	标准差	1.09×10^2	3.87×10^2	1.25×10^2	2.00×10^2	3.32×10^2	2.18×10^2	2.65×10^2	3.89×10^2	2.53×10^2
	平均值	1.51×10^4	1.27×10^8	3.59×10^5	4.83×10^6	1.20×10^8	6.04×10^5	1.39×10^6	2.76×10^7	1.71×10^4
Friedman	标准差	6.14×10^3	9.27×10^7	4.21×10^5	3.03×10^6	5.57×10^7	3.30×10^5	3.03×10^6	2.75×10^7	1.00×10^4
	平均排名	1.66	7.69	4.41	3.28	8.00	2.93	5.14	7.76	4.14
最终排名		1	7	5	3	9	2	6	8	4

表 4 Wilcoxon 检验在 30 维 CEC2017 测试集上的测试结果对比

Table 4 Comparison outcomes attained by Wilcoxon's test on the 30-dimensional CEC2017 test suite

函数	HOA	ADE	VPPSO	SCA	DO	DBO	WOA	SSA
单峰函数	2.35×10^{-6}	6.45×10^{-6}	4.09×10^{-2}	2.35×10^{-6}	2.48×10^{-3}	9.32×10^{-5}	1.63×10^{-6}	3.54×10^{-1}
简单多模态函数	3.79×10^{-14}	6.22×10^{-2}	1.64×10^{-1}	2.73×10^{-14}	3.44×10^{-2}	9.14×10^{-3}	3.75×10^{-9}	4.15×10^{-3}
混合函数	1.50×10^{-57}	3.07×10^{-39}	5.31×10^{-27}	4.74×10^{-56}	1.94×10^{-24}	4.68×10^{-31}	2.58×10^{-55}	1.51×10^{-21}
复合函数	1.17×10^{-29}	9.73×10^{-8}	8.39×10^{-4}	2.34×10^{-20}	2.57×10^{-4}	1.67×10^{-7}	5.55×10^{-16}	2.36×10^{-6}

受篇幅限,选取上述相同的函数展示收敛效果图,如图 3 所示,CDHOA 在求解 F1 函数上的收敛精度与 VPPSO、DO 和 SSA 相比较差,在求解 F6、F9 和 F27 函数上略逊于 ADE,其余函数中,CDHOA 均优于 8 种比较算法。且从图中可以得出,CDHOA 在求解多峰函数上有明显的优势。



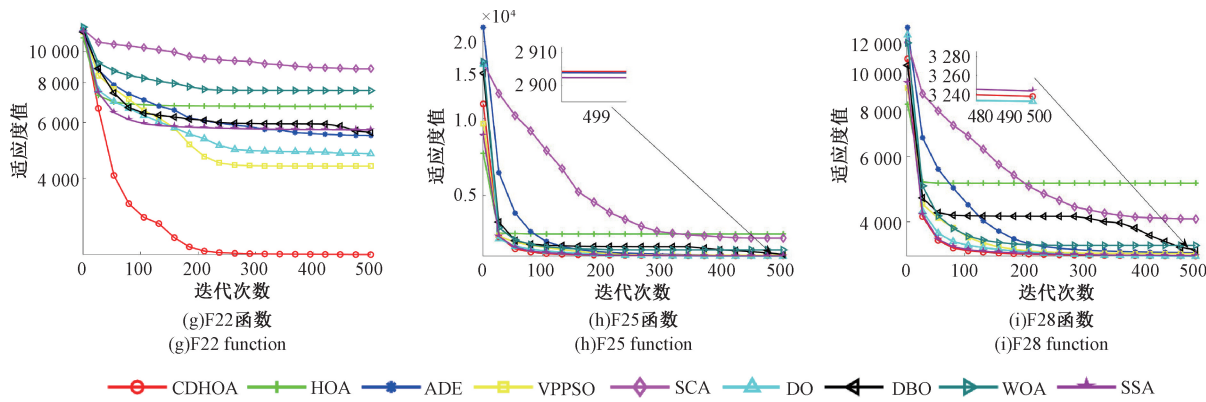


图 3 九种算法对 29 个测试函数的收敛曲线 Dim=30

Fig. 3 Convergence curves of nine algorithms for twenty-nine text functions with Dim=30

3.2 三种改进策略的有效性分析

经 3.1 节分析,CDHOA 在 CEC2017 测试集上的求解性能较原算法得到显著提升,但这是综合 3 个策略的结果。

为了分析 3 种策略的有效性,选择 CEC2017 测试集中的 9 个测试函数(单峰函数 F1,多峰函数 F5、F8,混合函数 F16、F18、F19,复合函数 F22、F23、F28)进行有效性数值实

验,设置 Dim=30。对基本 HOA 分别设置:仅使用佳点集初始化种群策略(记为 GHOA)、仅使用柯西逆累积分布算子策略(记为 CHOA)、仅使用随机差分策略(记为 DHOA),以此进行数值实验。算法参数设置和实验环境,都与 3.1 节一致。表 5 列出了 CDHOA、HOA、GHOA、CHOA 和 DHOA 经过 30 次运行后的平均值、标准差,图 4 为目标函数值曲线收敛图。

表 5 CDHOA 不同策略有效性分析结果

Table 5 Analysis of the effectiveness of different strategies for CDHOA

函数	统计	CDHOA	HOA	GHOA	CHOA	DHOA
F1	平均值	1.16×10^6	2.61×10^{10}	3.26×10^{10}	2.97×10^9	4.12×10^3
	标准差	6.14×10^5	4.88×10^9	4.11×10^9	5.07×10^8	4.81×10^3
F5	平均值	6.19×10^2	7.48×10^2	7.79×10^2	7.68×10^2	6.45×10^2
	标准差	3.18×10^1	3.52×10^1	2.71×10^1	1.84×10^1	2.72×10^1
F8	平均值	9.04×10^2	1.01×10^3	9.98×10^2	1.06×10^3	9.12×10^2
	标准差	2.41×10^1	2.43×10^1	2.15×10^1	2.29×10^1	2.38×10^1
F16	平均值	2.33×10^3	3.75×10^3	3.71×10^3	2.98×10^3	2.60×10^3
	标准差	2.96×10^2	4.28×10^2	3.33×10^2	1.33×10^2	2.69×10^2
F18	平均值	9.10×10^3	6.81×10^6	5.76×10^6	5.62×10^5	1.05×10^4
	标准差	8.78×10^3	5.26×10^6	3.73×10^6	2.30×10^5	6.08×10^3
F19	平均值	2.10×10^3	7.76×10^6	5.56×10^6	2.31×10^5	3.36×10^3
	标准差	5.95×10^1	6.33×10^6	4.24×10^6	9.45×10^4	3.60×10^3
F22	平均值	2.50×10^3	7.25×10^3	7.11×10^3	4.12×10^3	2.54×10^3
	标准差	1.03×10^3	1.55×10^3	1.32×10^3	1.30×10^3	9.08×10^2
F23	平均值	2.79×10^3	3.28×10^3	3.25×10^3	2.91×10^3	2.92×10^3
	标准差	3.59×10^1	8.15×10^1	5.44×10^1	6.09×10^1	5.36×10^1
F28	平均值	3.23×10^3	4.82×10^3	5.16×10^3	3.67×10^3	3.23×10^3
	标准差	2.90×10^1	4.27×10^2	2.48×10^2	6.53×10^1	2.40×10^1
Friedman 平均排名		1.11	4.44	4.22	3.22	2
最终排名		1	5	4	3	2

由表 5 实验结果和图 4 收敛效果图可知,除 F1 以外,CDHOA 在各函数上的求解结果平均值均优于其他 4 种比

较算法;F1 的求解结果平均值次之。就标准差而言,CDHOA 在求解 F19、F23 函数上相较于其他比较算法有

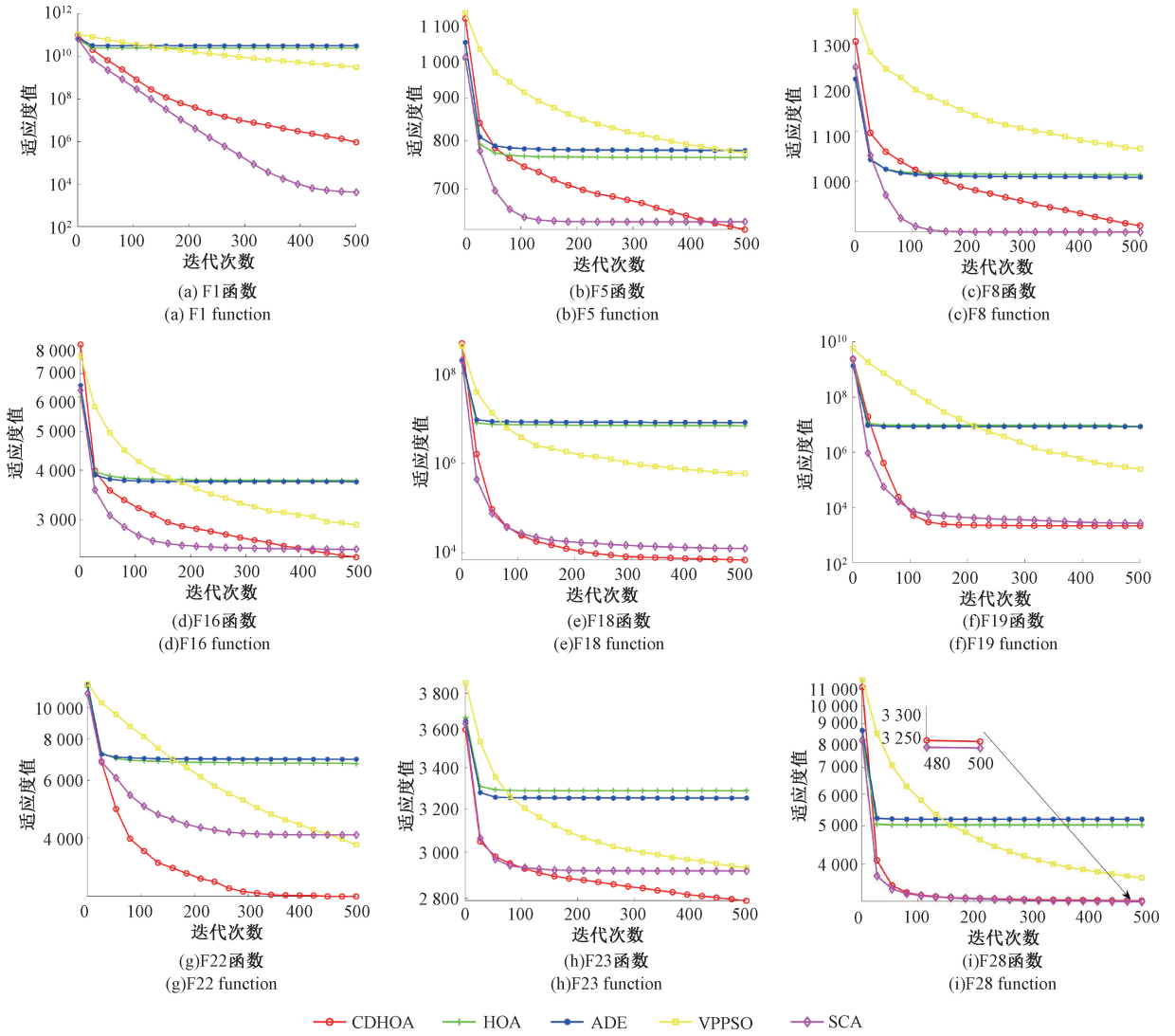


图4 五种算法对12个测试函数的收敛曲线 Dim=10

Fig. 4 Convergence curves of five algorithms for twelve text functions with Dim=10

更好的稳定性,但在其他函数上的稳定性整体略逊一筹。可以看到引入随机差分策略的 DHOA 与使用柯西逆累积分布算子的 CHOA 性能表现不错,但整体上仍与 CDHOA 有所差距。CHOA、DHOA 实验结果整体上比原算法 HOA 都有所提升。此外,Friedman 检验的结果显示,CDHOA 排在第1位,而原算法 HOA 排在第5位。说明3种策略综合对原算法改进均是有效的。

4 CDHOA 用于光伏模型参数辨识

随着全球对电能需求的不断上升,能源结构现代化和可再生能源收集系统的研究变得至关重要。风能、太阳能和水资源等可再生能源越来越受到重视。在这些可再生能源中,太阳能被认为是最有潜力满足不断增长的能源需求的最佳选择之一,其利用光伏(photovoltaic,PV)系统将太阳能转化成电能^[22]。

为了模拟光伏组件的电学行为,特别是在不同光照条件和温度下的性能表现,参考文献[22],使用光伏组件模型(photovoltaic module model,PMM)。

其输出电流计算公式为^[22]:

$$I_L = N_p (I_{ph} - I_d - I_{sh}) \quad (10)$$

$$I_d = I_{sd} (\exp(q(\frac{V_L}{N_s} + R_s \cdot \frac{I_L}{N_p}) / (akT) - 1)) \quad (11)$$

$$I_L = \frac{1}{R_{sh}} (\frac{V_L}{N_s} + R_s \cdot \frac{I_L}{N_p}) \quad (12)$$

式中: I_{ph} 为光电电流, I_d 为二极管电流, I_{sd} 为二极管饱和电流, R_s 、 R_{sh} 为串、并联电阻数, V_L 为输出电压, $q=1.602\ 176\ 464 \times 10^{-19}$ 为电子电荷, $k=1.380\ 650\ 3 \times 10^{-23}$ 为玻尔兹曼常量, a 为二极管理想系数。由式(10)~(12)可知,在光伏组件模型中有5个未知参数(I_{ph} , I_{sd} , R_s , R_{sh} , a)需要辨识。

PV 模型的参数辨识可转化为用均方根误差 (RMSE) 来度量模拟值与真实值之间误差最小的优化问题:

$$RMSE(x) = \left(\frac{1}{N} \sum_{i=1}^n [I_{i,calculated}(x) - I_{i,measure}]^2 \right)^{0.5}$$

(13)

式中: N 为数据规模, $x = (I_{ph}, I_{sd}, R_s, R_{sh}, a)$ 。

本文利用 CDHOA 对光伏组件模型 (PMM) 的参数进

行辨识, 并将结果与 HOA、ADE、VPPSO、SCA、DO、DBO、WOA、SSA 进行对比。在数值实验中, 使用文献[22]的模型组件数据。为确保比较的公平性, 设置种群规模 $N=100$, 最大迭代次数 $T=500$, 其余算法详细参数与各原算法的标准参数一致。每个算法独立运行 30 次, 并取最优结果, 结果如表 6 所示。记录各算法所得目标函数值的最好值、最差值、平均值和标准差, 如表 7 所示。

表 6 九种算法在 PMM 上独立运行 30 次的最佳结果

Table 6 The best outcomes attained by nine methods over 30 independent runs on PMM

算法	$I_{ph} [A]$	$I_{sd} [\mu A]$	$R_s [\Omega]$	$R_{sh} [\Omega]$	a	RMSE	rank
CDHOA	1.030 51	3.482 26	1.201 27	981.982 26	48.642 83	2.43×10^{-3}	1
ADE	1.030 44	2.719 99	1.229 74	896.782 92	47.709 21	2.56×10^{-3}	6
VPPSO	1.030 09	3.774 63	1.194 81	1 102.585 66	48.950 52	2.45×10^{-3}	4
GWO	1.069 98	4.735 67	1.094 99	230.750 37	50.000 00	1.56×10^{-2}	9
SCA	1.027 89	4.861 39	1.101 43	613.161 30	50.000 00	8.87×10^{-3}	8
DO	1.029 00	4.479 43	1.174 93	1 385.324 46	49.625 15	2.52×10^{-3}	5
DBO	1.030 20	3.753 39	1.193 13	1 058.223 61	48.931 72	2.43×10^{-3}	3
WOA	1.029 19	4.790 69	1.165 51	1 393.815 18	49.896 95	2.59×10^{-3}	7
SSA	1.030 96	3.237 48	1.209 19	908.817 26	48.365 42	2.43×10^{-3}	2

表 7 求解 PMM 所得目标函数值

Table 7 Objective function values of algorithms for solving PMM

算法	最好值	最差值	平均值	标准差
CDHOA	2.43×10^{-3}	2.43×10^{-3}	2.43×10^{-3}	1.90×10^{-17}
HOA	2.57×10^{-3}	1.44×10^{-1}	3.86×10^{-2}	4.26×10^{-2}
ADE	2.45×10^{-3}	2.63×10^{-3}	2.53×10^{-3}	5.46×10^{-5}
VPPSO	1.57×10^{-2}	9.97×10^{-2}	4.72×10^{-2}	2.40×10^{-2}
SCA	8.87×10^{-3}	2.74×10^{-1}	6.55×10^{-2}	6.33×10^{-2}
DO	2.52×10^{-3}	1.06×10^{-2}	3.60×10^{-3}	1.87×10^{-3}
DBO	2.43×10^{-3}	4.97×10^{-3}	2.68×10^{-3}	4.40×10^{-4}
WOA	2.59×10^{-3}	2.74×10^{-1}	5.74×10^{-2}	1.02×10^{-1}
SSA	2.43×10^{-3}	2.74×10^{-1}	1.21×10^{-2}	4.95×10^{-2}

由表 6 的结果可知, CDHOA 在辨识 PMM 模型参数得到一组最优的参数, 其对应的 MRSE 值为 2.43×10^{-3} , 这是所有比较算法中的最小值。此外, 该 MRSE 值也与文献[23]中提出算法所得的最小 MRSE 值相同。而 SSA 算法排在第 2 位。

CDHOA 所辨识出的最优参数生成的电流-电压曲线图如图 5(a)所示, 功率-电压曲线如图 5(b)所示。从图中可以清晰地看出, 在电压范围内, CDHOA 得到的模拟参数与实测数据呈现出高度一致。

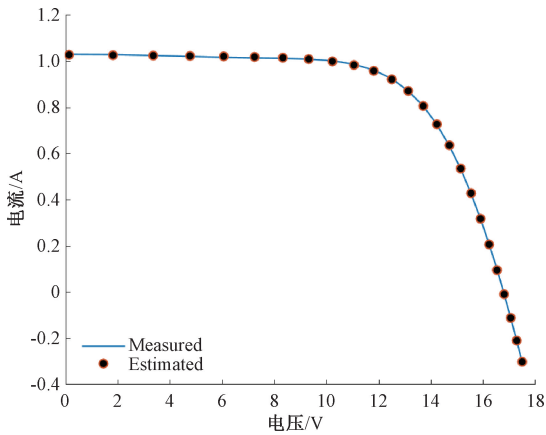
此外, 通过图 6 (a) 的收敛曲线图, 可以发现 CDHOA 在求解 PMM 模型参数辨识问题中展现出良好的鲁棒性。由图 6 (b) 箱线图可知, 独立运行 30 次, HOA、VPPSO、SCA 与 WOA 所得的结果较为分散; 疑似异常值较为明显的有 HOA、SCA、WOA 和 SSA; 而 CDHOA 从图中可以

看出结果相对集中, 疑似异常值均与平均值相近。

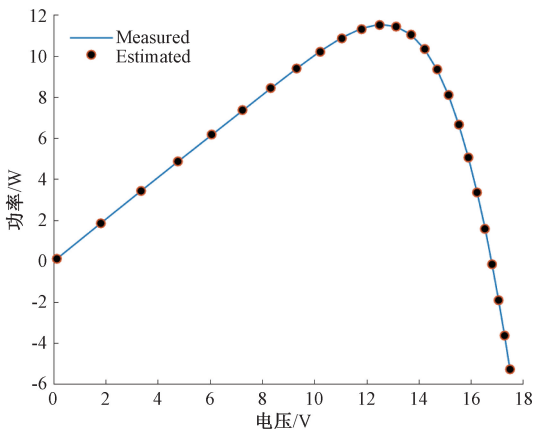
由表 7 可知, 算法 CDHOA 所得目标函数值的最好值、最差值、平均值和标准差均优于比较算法, 且标准差要远小于比较算法。

5 工程设计问题的求解

为了检验改进算法求解工程设计问题的能力, 将 CDHOA 算法和上述 8 种对比算法应用于压力容器设计和三杆桁架设计问题。这两个工程问题具有各自不同的求解难度和约束条件, 能够很好的测试出算法在工程问题中的适应性。种群规模 $N=100$, 最大迭代次数 $T=500$, 本文使用惩罚系数 $\lambda=10^3$ 处理约束条件, 为避免实验结果的偶然性, 共运行 30 次, 记录各算法在不同问题所得目标函数值的最好值、最差值、平均值和标准差。



(a) 电流-电压图
(a) I-V curves



(b) 功率-电压图
(b) P-V curves

图 5 PMM 的 CDHOA 测量数据与实验数据的比较

Fig. 5 Comparisons between measured data and experimental data attained by CDHOA for PMM

5.1 压力容器设计问题

压力容器设计问题的目的是使压力容器制作成本最小。数学模型如下:

$$\begin{cases} \min f(x) = 0.622 \, 4x_1x_3x_4 + 1.778 \, 1x_2x_3^2 + \\ \quad 3.166 \, 1x_1^2x_4 + 19.84x_1^2x_3 \\ s. t. \, g_1 = -x_1 + 0.019 \, 3x_3 \leq 0 \\ g_2 = -x_2 + 0.009 \, 54x_3 \leq 0 \\ g_3 = -\pi x_3^2x_4 + \frac{4}{3}\pi x_3^3 + 1 \, 296 \, 000 \leq 0 \\ g_4 = x_4 - 240 \leq 0 \\ 0 \leq x_1, x_2 \leq 100 \\ 0 \leq x_3, x_4 \leq 200 \end{cases} \quad (14)$$

式中: x_1 为半径, x_2 为无头圆柱长度, x_3 为壳体厚度, x_4 为封头厚度。

利用算法 CDHOA 求解压力容器设计问题,并于 3.1 节中的 8 种对比算法进行对比。算法中的参数与原算法文献相同。由表 8 可知,算法 CDHOA 求解压力容器设

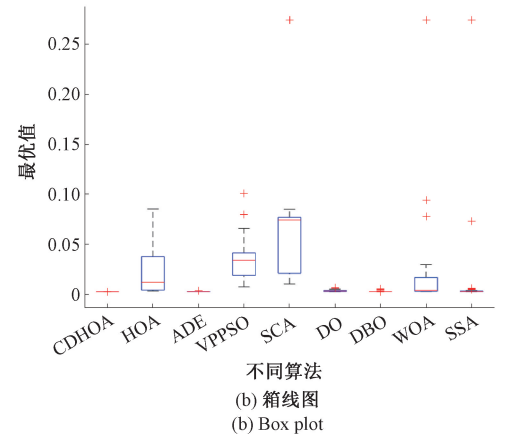
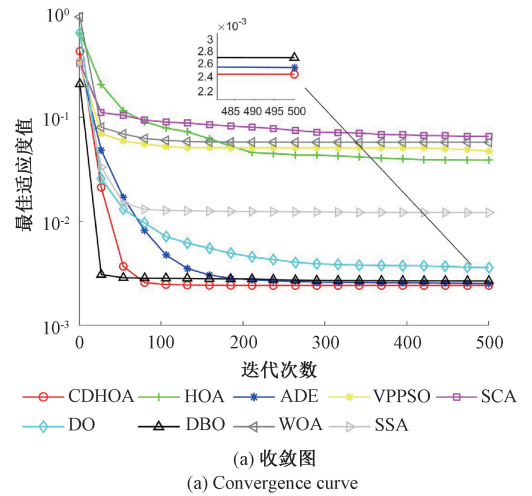


图 6 九种算法在 PMM 上独立运行 30 次的结果分析图

Fig. 6 The outcomes attained by nine methods over 30 independent runs on PMM

计问题所得目标函数的最好值、最差值、平均值和标准差均优于对比算法所得结果。CDHOA 求解压力容器设计问题所得的解 $x = \{0.778 \, 168 \, 641, 0.384 \, 649 \, 163, 40.319 \, 618 \, 7, 200\}$ 对应的最小目标函数值为 $f = 5 \, 885.332 \, 77$ 。

5.2 三杆桁架设计问题

三杆桁架设计问题的目的是通过调整横截面积大小来最小化三杆桁架的体积。数学模型为:

$$\begin{cases} \min f(x) = a(2\sqrt{2}x_1 + x_2) \\ s. t. \, g_1 = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2}P - \sigma \leq 0 \\ g_2 = \frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2}P - \sigma \leq 0 \\ g_3 = \frac{1}{\sqrt{2}x_2 + x_1}P - \sigma \leq 0 \\ a = 100 \, \text{cm}, P = 2 \, \text{kN/cm}^2 \\ \sigma = 2 \, \text{kN/cm}^2 \\ 0 \leq x_1, x_2 \leq 1 \end{cases} \quad (15)$$

表 8 求解压力容器设计问题

Table 8 Algorithms for solving pressure vessel design problem

算法	最优值	最差值	平均值	标准差
CDHOA	5 885.332 77	5 885.332 78	5 885.332 77	6.62×10^{-7}
HOA	7 057.939 81	23 712.460 75	10 596.415 43	3.69×10^3
ADE	5 922.868 05	7 319.000 70	6 346.039 08	4.16×10^2
VPPSO	5 890.285 88	7 333.771 43	6 688.293 08	4.78×10^2
SCA	6 014.867 02	8 197.162 13	6 802.179 68	5.50×10^2
DO	5 885.658 73	7 316.338 16	6 344.524 77	5.41×10^2
DBO	6 682.071 10	239 839.906 60	40 538.651 32	4.84×10^4
WOA	6 519.393 12	14 163.692 50	7 983.936 89	1.72×10^3
SSA	7 172.886 36	152 134.299 68	25 288.071 23	3.40×10^4

利用算法 CDHOA 求解三角桁架设计问题,并于 3.1 节中的 8 种对比算法进行对比。算法中的参数与原算法文献相同。由表 9 可知,算法 CDHOA 求解两个工程问题所得目标函数的最好值、最差值、平均值和标准差均优

于对比算法所得结果。CDHOA 求解三杆桁架设计问题所得的解为 $x = \{0.788\ 675\ 133\ 82, 0.408\ 248\ 292\ 67\}$, 对应的最小目标函数值为 $f = 263.895\ 84$ 。

表 9 求解三杆桁架设计问题

Table 9 Algorithms for solving three bar truss design

算法	最好值	最差值	平均值	标准差
CDHOA	263.895 84	263.895 84	263.895 84	1.49×10^{-14}
HOA	263.895 87	264.003 54	263.912 41	2.34×10^{-2}
ADE	263.895 86	263.899 43	263.896 42	7.07×10^{-4}
VPPSO	263.895 85	263.909 08	263.896 93	2.50×10^{-3}
SCA	263.903 07	264.125 42	263.990 69	6.64×10^{-2}
DO	263.895 85	263.897 65	263.896 14	4.54×10^{-4}
DBO	263.951 66	266.715 20	264.884 17	8.11×10^{-1}
WOA	263.895 91	268.681 78	264.618 08	1.08×10^0
SSA	263.915 03	270.540 39	264.284 36	1.20×10^0

6 结 论

本文所提的多策略改进徒步算法,通过综合应用佳点集初始化、柯西逆累积分布算子和随机差分变异策略,显著提升了算法的全局搜索性能和解的质量。将 CDHOA 算法应用于 CEC2017 测试集的全局优化求解中,其 10 维和 30 维的运行结果总体性能均优于 8 种性能优越的代表性对比算法;Wilcoxon 秩和检验及 Friedman 检验则验证了 CDHOA 算法与各对比算法之间的优势差异具有显著性,且排名第一。最后,将该算法成功运用在光伏模型参数辨识和工程设计问题中,得到结果:辨识得到的参数与实测数据高度一致;求解工程设计问题精确度较好。综上所述,CDHOA 算法得到了更好的全局搜索能力,具有更好的收敛速度和精度,具备解决实际问题的良好适用性和有效性。未来的工作将致力于进一步优化算法性能,探索其在更广泛的工程问题中的应用,如大规模变量优

化、约束优化问题以及无人机路径规划等。

参考文献

[1] SHAMI T M, MIRJALILI S, Al-ERYAN Y, et al. Velocity pausing particle swarm optimization: A novel variant for global optimization[J]. Neural Computing and Applications, 2023, 35(12): 9193-9223.

[2] ABBASI B, MAJIDNEZHAD V, MIRJALILI S. ADE: Advanced differential evolution [J]. Neural Computing and Applications, 2024, 36 (25): 15407-15438.

[3] WANG W CH, TIAN W C, XU D M, et al. Arctic puffin optimization: A bio-inspired metaheuristic algorithm for solving engineering design optimization[J]. Advances in Engineering Software, 2024, 195: 103694.

[4] 万怡华,张雪梅. 混合多策略改进蜣螂算法的避障路径规划[J]. 电子测量技术, 2024, 47(2): 69-78.

- WAN Y H, ZHANG X M. Hybrid multi-strategy improved dung beetle algorithm for obstacle avoidance path planning[J]. *Electronic Measurement Technology*, 2024, 47(2): 69-78.
- [5] 李卓, 李欣. 考虑粒子记忆传播的说话人精确跟踪方法[J]. *国外电子测量技术*, 2022, 41(6): 31-37.
- LI ZH, LI X. Accurate speaker tracking method considering particle memory propagation[J]. *Foreign Electronic Measurement Technology*, 2022, 41(6): 31-37.
- [6] 王宏健, 鄂鑫, 张凯, 等. 改进蚁群算法解决 UUV 集群任务规划问题[J]. *仪器仪表学报*, 2022, 43(9): 238-254.
- WANG H J, E X, ZHANG K, et al. Improved ant colony algorithm to solve UUV cluster task planning problem[J]. *Chinese Journal of Scientific Instrument*, 2022, 43(9): 238-254.
- [7] OLADEJO O S, EKWE O S, MIRJALILI S, et al. The hiking optimization algorithm: A novel human-based metaheuristic approach[J]. *Knowledge-Based Systems*, 2024, 296: 111880.
- [8] MIRJALILI S, MIRJALILI S M, LEWIS A. Grey wolf optimizer[J]. *Advances in Engineering Software*, 2014, 69: 46-61.
- [9] LIN D, WANG Y, XIN H, et al. Fault diagnosis method for hydropower units based on dynamic mode decomposition and the hiking optimization algorithm-extreme learning machine [J]. *Energies*, 2024, 17(20): 5159.
- [10] LONG W, WU T B, XU M, et al. Parameters identification of photovoltaic models by using an enhanced adaptive butterfly optimization algorithm[J]. *Energy*, 2021, 229: 120750.
- [11] KUMAR A, WU G H, ALI M Z, et al. A test-suite of non-convex constrained optimization problems from the real-world and some baseline results[J]. *Swarm and Evolutionary Computation*, 2020, 56: 100693.
- [12] GOODCHILD M F. Beyond toiler's hiking function [J]. *Geographical Analysis*, 2020, 52(4): 558-569.
- [13] WANG K, ROHDE G K, XIAO J. Edge feature enhancement approach using Hilbert transform of Cauchy distribution and its applications [J]. *IET Image Processing*, 2021, 15(12): 2891-2909.
- [14] DAS S, SUGANTHAN P N. Differential evolution: A survey of the state-of-the-art[J]. *IEEE Transactions on Evolutionary Computation*, 2010, 15(1): 4-31.
- [15] 梁昔明, 史兰艳, 龙文. 基于混合策略改进的蛇优化算法及其应用[J]. *计算机工程与科学*, 2024, 46(4): 693-706.
- LIANG X M, SHI L Y, LONG W. An improved snake optimization algorithm based on hybrid strategies and its application[J]. *Computer Engineering & Science*, 2024, 46(4): 693-706.
- [16] MIRJALILI S. SCA: A sine cosine algorithm for solving optimization problems[J]. *Knowledge Based Systems*, 2016, 96: 120-133.
- [17] ZHAO SH J, ZHANG T R, MA SH L, et al. Dandelion optimizer: A nature-inspired metaheuristic algorithm for engineering applications[J]. *Engineering Applications of Artificial Intelligence*, 2022, 114: 105075.
- [18] XUE J K, SHEN B. Dung beetle optimizer: A new meta-heuristic algorithm for global optimization[J]. *The Journal of Supercomputing*, 2023, 79(7): 7305-7336.
- [19] MIRJALILI S, LEWIS A. The whale optimization algorithm [J]. *Advances in Engineering Software*, 2016, 95: 51-67.
- [20] XUE J K, SHEN B. A novel swarm intelligence optimization approach: sparrow search algorithm[J]. *Systems Science & Control Engineering*, 2020, 8(1): 22-34.
- [21] ONAY F K, AYDEMİR S B. Chaotic hunger games search optimization algorithm for global optimization and engineering problems [J]. *Mathematics and Computers in Simulation*, 2022, 192: 514-536.
- [22] YU K J, LIANG J J, QU B Y, et al. Multiple learning backtracking search algorithm for estimating parameters of photovoltaic models [J]. *Applied Energy*, 2018, 226: 408-422.
- [23] SIOWIK A, CPAIKA K, XUE Y, et al. An efficient approach to parameter extraction of photovoltaic cell models using a new population-based algorithm[J]. *Applied Energy*, 2024, 364: 123208.

作者简介

徐明, 教授, 硕士生导师, 博士, 主要研究方向为智能优化算法、机器学习。

王风富, 硕士研究生, 主要研究方向为智能优化算法。

龙文(通信作者), 教授, 硕士生导师, 博士, 主要研究方向为智能优化算法、机器学习。

E-mail: lw227@mail.gufe.edu.cn