

基于T分布扰动和透镜成像反向学习的算术优化算法及应用

徐黎卓¹, 陆莎^{2*}, 王文静¹, 陈美珍¹

¹南宁师范大学数学与统计学院, 广西 南宁

²南宁师范大学广西应用数学中心, 广西 南宁

收稿日期: 2025年3月28日; 录用日期: 2025年4月23日; 发布日期: 2025年4月30日

摘要

针对算术优化算法(Arithmetic optimization algorithm, AOA)易陷入局部最优、收敛速度慢等问题, 提出一种基于T分布透镜成像反向学习策略的算术优化算法(TOBLAOA), 设计动态平衡机制调节全局探索强度与局部开采深度的配比, 从而突破早熟收敛瓶颈并提高解精度稳定性。对2005、2017、2015测试集中的部分基准函数(共15个)进行仿真实验, 首先引入4种不同改进策略的改进透镜成像反向学习策略的算术优化算法(OBLAOA)算法进行比较, 再与哈里斯鹰优化算法、雷电附着优化算法等6个优化算法进行了实验结果对比和差异显著性Wilcoxon秩和检验, 结果表明改进后的算术优化算法在求解精度、收敛速度上均有显著提升。最后将算法应用于工程优化中的常见的压力容器设计、三杆桁架设计、齿轮系设计中, 进一步验证此算法的有效性。

关键词

算术优化算法, T分布扰动, 透镜成像反向学习, 工程优化问题

Arithmetic Optimization Algorithm Based on T-Distribution Perturbation and Lens Imaging Opposition-Based Learning with Its Applications

Lizhuo Xu¹, Sha Lu^{2*}, Wenjing Wang¹, Meizhen Chen¹

¹School of Mathematics and Statistics, Nanning Normal University, Nanning Guangxi

²Guangxi Center for Applied Mathematics, Nanning Normal University, Nanning Guangxi

*通讯作者。

文章引用: 徐黎卓, 陆莎, 王文静, 陈美珍. 基于 T 分布扰动和透镜成像反向学习的算术优化算法及应用[J]. 应用数学进展, 2025, 14(4): 1051-1075. DOI: 10.12677/aam.2025.144226

Abstract

To address issues such as the Arithmetic Optimization Algorithm (AOA) being prone to local optima and having slow convergence speed, this paper proposes a T-distribution lens imaging opposition-based learning arithmetic optimization algorithm (TOBLAOA). The algorithm designs a dynamic balance mechanism to regulate the ratio between global exploration intensity and local exploitation depth, thereby breaking through the bottleneck of premature convergence and improving the stability of solution accuracy. Simulation experiments were conducted on selected benchmark functions (totaling 15) from the 2005, 2017, and 2015 test sets. The study first compares four improved OBLAOA algorithms with different enhancement strategies, and then performs experimental result comparisons and Wilcoxon rank-sum significance tests with six optimization algorithms including Harris Hawk Optimization and Lightning Attachment Procedure Optimization. The results demonstrate that the enhanced arithmetic optimization algorithm achieves significant improvements in both solution accuracy and convergence speed. Finally, the algorithm was applied to common engineering optimization problems such as pressure vessel design, three-bar truss design, and gear train design, further verifying its effectiveness.

Keywords

Arithmetic Optimization Algorithm, T-Distribution Perturbation, Lens Imaging Reverse Learning, Engineering Optimization Problems

Copyright © 2025 by author(s) and Hans Publishers Inc.

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

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



Open Access

1. 引言

在信息化时代技术革新驱动下，多学科领域涌现出大量非线性化、高维度的待解难题。具有高效求解机制及实现门槛的智能优化算法日益成为学界研究热点，其设计理念多源于对自然规律的建模与重构。早期科研团队通过仿生学原理推导出遗传算法(GA) [1]、粒子群优化(PSO) [2]等经典方法，随着研究深入，学界持续拓展算法体系，陆续设计出包含蝴蝶优化算法(BOA) [3]、哈里斯鹰动态捕猎算法(HHO) [4]、正弦余弦调控算法(SCA) [5]、鲸鱼协同觅食优化算法(WOA) [6]等创新型计算框架。此类方法在工业调度、机器人路径规划等场景中展现出显著的应用潜力。

算术优化算法(Arithmetic optimization algorithm, AOA) [7]作为新一代智能优化工具，由 Abualigah 等人于 2021 年提出。其核心创新在于建立数学运算机制与搜索过程的内在映射。该方法通过解析加减乘除操作的数学特性，构建局部搜索与全局搜索阶段的动态平衡机制：乘除运算拓展解空间覆盖广度，加减操作实现邻域精细化调节。目前学者研究常见的改进方向有：郑婷婷等[8]引入概率分布调节参数，黄学雨等[9]建立三角函数辅助的局部逃逸策略，兰周新等[10]构建多模态特征融合框架。特别在机械结构优化、能源系统调度等领域，多代改进版 AOA 已被证实具有较强工程适用性。

针对现有算法在动态环境适应性和早熟收敛方面的不足，本研究设计了融合光学成像原理的算法增强方案[11]。首先改造数学优化加速器机制(MOA)以动态调节全局 - 局部搜索权重，其次在解空间挖掘阶段嵌入透镜反向聚焦策略，该机制通过模拟光学成像的正逆向传播过程提升种群多样性，从而增强复

杂模型解空间的遍历能力。仿真实验证明, 该混合策略能有效突破传统算法的收敛瓶颈。

2 算术优化算法

2.1. 初始化条件

首先对种群进行初始化, 生成一个均匀分布的随机种群, 其生成公式如下:

$$X_i(t) = a \times (Ub - Lb) + Lb \quad (2.1.1)$$

其中 Ub 为搜索上界, Lb 为搜索下界, a 为 $[0,1]$ 间的随机数, $X_i(t) (i=1,2,3,\dots,N)$ 为初始化后的初始位置。

2.2. 函数加速器(Math Optimizer Accelerated, MOA)

在判断算法进行全局搜索还是局部寻优时, 需要计算 MOA 的值并与 $[0,1]$ 区间上的随机数 r_1 作比较, 当 $r_1 < MOA(t)$ 时, 算法进入全局寻优阶段, 反之则进入局部探索阶段。我们参考郑婷婷等运用 t 分布变异策略和动态边界策略混合改进的算术优化算法[8], 采用 MOA 计算公式如下:

$$MOA(t) = Min + t \times \frac{Max - Min}{T} \quad (2.2.1)$$

其中, 加速函数值为 $MOA(t)$, 加速函数的最小值为 Min , 最大值和 Max , 当前迭代次数为 t , 最大迭代次数为 T 。

2.3. 探索阶段

当控制系数 MOA 满足 $r_1 < MOA(t)$ 时, 算法启动全局探索模式, 在此阶段, 个体位置更新机制采用基于乘法的高离散度运算模型, μ 作为搜索过程中的控制系数, ε 为随机比例系数, 通过下列表达式更新候选解:

$$X_i(t+1) = \begin{cases} X_{best}(t) \div (MOP + \varepsilon) \times ((Ub - Lb) \times \mu + Lb), & r_2 < 0.5 \\ X_{best}(t) \times MOP \times ((Ub - Lb) \times \mu + Lb), & \text{其他} \end{cases} \quad (2.3.1)$$

其中, r_2 是 $[0,1]$ 区间上的随机数, $X_i(t+1)$ 为下一次迭代个体的位置, $X_{best}(t)$ 为当前个体最优解。乘除算子的非线性特性可有效扩大搜索空间覆盖半径, 提升解域的探索广度, 为探索解空间的不同区域, 保证解的多样化, 算法引入数学优化概率(Math Optimizer Probability, MOP), 其计算模型定义为:

$$MOP(t) = 1 - \left(\frac{t}{T} \right)^{\frac{1}{a}} \quad (2.3.2)$$

其中, $a > 0$ 是定义开发精度的敏感参数(根据参考文献[12]确定该值固定为 5), 该参数通过指数调节控制迭代中后期解集的收敛密度。

当 $r_1 > MOA(t)$ 时, 算法切换至邻域精细开发模式, 此阶段采用低离散度的加减法运算模型进行解空间局部遍历, 增强候选解的精度锁定能力。其位置迭代形式可表示为:

$$X_i(t+1) = \begin{cases} X_{best}(t) - MOP \times ((Ub - Lb) \times \mu + Lb), & r_3 < 0.5 \\ X_{best}(t) + MOP \times ((Ub - Lb) \times \mu + Lb), & \text{其他} \end{cases} \quad (2.3.3)$$

其中, r_3 是 $[0,1]$ 区间上的随机数。

2.4. AOA 实现步骤

设在 D 维搜索空间中有 N 个算子, 算子 i 在 t 时刻的位置以及该算子在搜寻过程中所确定的最好位置分别记为 $x_i(t)$ 和 x_{est} , 引入函数加速器 MOA 与随机数 r_1 作比较判断搜索方式, μ 作为搜索过程中的控制系数, ε 为随机比例系数控制算子在区域搜索的范围及方式, 取 $\mu=0.499$ 、 $a=5$ 、 $Min=0.2$ 、 $Max=1$ 则 AOA 算法的实现步骤如下表 1:

- Step 1: 给定种群大小 N 、问题维度 D 、目标函数 $f(x)$ 、最大迭代次数 T 、令 $t:=0$
- Step 2: 利用式(2.1.1)初始化种群 $x_i(t)(i=1,2,\dots,N)$, 取 $\mu=0.499$ 、 $a=5$ 、 $Min=0.2$ 、 $Max=1$
- Step 3: 评估每个算子的适应度值 $f(x_i(t))(i=1,2,\dots,N)$
- Step 4: 将生成随机数 r_1 与利用式(2.2.1)计算所得 $MOA(t)$ 作比较, 判断搜索方式
- Step 5: 利用式(2.3.2)计算数学优化概率
- Step 6: 若 $r_1 < MOA(t)$ 则通过式(2.3.1)更新位置, 反之用(2.3.3)
- Step 7: 检查新位置的可能性并进行更新
- Step 8: 评估新位置的适应度值
- Step 9: 判断是否达成终止条件, 若是转 step 10; 否则转 step 4
- Step 10: 输出算子中当前最优记忆位置作为目标函数的解

Table 1. Pseudocode of the AOA

表 1. 算术优化算法伪代码

算法 1	算术优化算法(AOA)
1.	输入: 初群体规模 N 、最大迭代次数 T 、令 $t:=0$, 初始化 $x_i(t)(i=1,2,\dots,N)$, 取 $\mu=0.499$ 、 $a=5$ 、 $Min=0.2$ 、 $Max=1$
2.	While $t < T$ do
3.	计算适应度并找到最小解 $X_{\text{best}}(t)$, 更新 $MOA(t)$ 、 $MOP(t)$ 的值
4.	for $i=1,2,3,\dots,N$ do
5.	生成随机数 r_1 、 r_2 、 r_3
6.	if $r_1 > MOA(t)$ do
7.	计算 $X_i(t+1)=\begin{cases} X_{\text{best}}(t) \div (MOP + \varepsilon) \times ((Ub - Lb) \times \mu + Lb), & r_2 < 0.5 \\ X_{\text{best}}(t) \times MOP \times ((Ub - Lb) \times \mu + Lb), & \text{其他} \end{cases}$
8.	else
9.	计算 $X_i(t+1)=\begin{cases} X_{\text{best}}(t) - MOP \times ((Ub - Lb) \times \mu + Lb), & r_3 < 0.5 \\ X_{\text{best}}(t) + MOP \times ((Ub - Lb) \times \mu + Lb), & \text{其他} \end{cases}$
10.	end if
11.	end for
12.	$t = t + 1$
13.	end while
14.	输出最优解 X_{best}

3. 基于不同优化策略下的 OBLAOA

算术优化算法凭借其计算框架简洁、调节维度低等特征, 在工程应用等优化领域都展现了一定的优

势,但是该算法仍然存在一定的局限性。一为开发深度不足,由于该算法的核心建立在四则运算的基础上,使得其迭代新解的过程容易陷入停止;二为算法参数敏感,其设置还需进行优化。针对上述缺陷,引用透镜成像反向学习策略的算术优化算法(OBLAOA)该策略示意图[13]如图 1 所示。引用该策略改进后的算法则可以在一定程度上地解决问题,但是其收敛速度、精度还有提升空间,并且该算法在一些情况下仍存在易陷入局部最优的问题。就上述问题,提出几种算法的改进策略,其中具有 T 分布扰动和透镜成像反向学习策略融合的算术优化算法(TOBLAOA)展现出更好地性能,该算法较于本章提出的其他几种混合策略的 OBLAOA 更有优势,下对提出的几种策略进行说明。

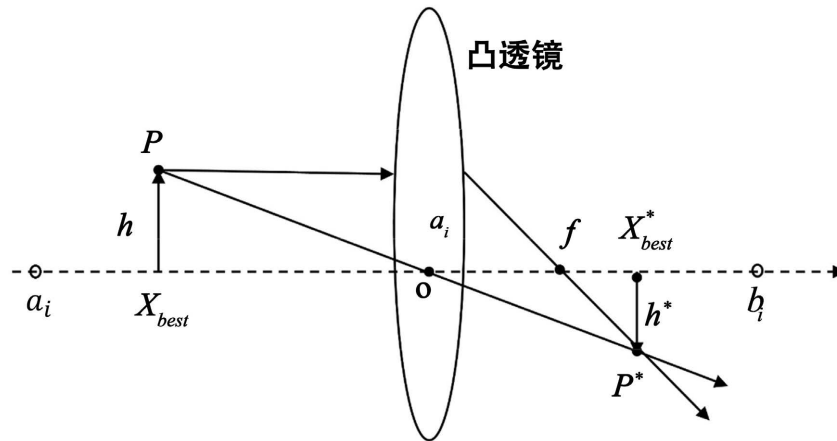


Figure 1. Schematic diagram of image formation by a convex lens
图 1. 凸透镜成像示意图

3.1. T 分布扰动策略(TOBLAOA)

当算法进入末尾阶段时,通常会达到一种局部搜索状态,但是由于搜索的不确定性,此时如果偏离前期的目标值,则会使得在算子在该局部的寻优都是无效的,为保障算子在后期寻优时的有效性,保证算子可以在最后寻优阶段仍然能跳出局部区域,将搜索群体随机弹送到距离目前区域更远的新区域,以此来达到寻求更有解的目的,达到真正的全局开发。本策略的改进目的在于增加变异扰动项和改变游走方式,引入随机性,与 OBLAOA 算法中生成的反向解相辅相成,增加解的多样性,从而避免早熟收敛。

T 分布扰动策略(T-distribution Perturbation Strategy, TDPS)基于 T 分布的长尾特性生成随机扰动步长,平衡优化算法的全局探索与局部开发能力。通过自由度的动态调节,在迭代早期根据厚尾特性增强探索能力,后期近似高斯分布聚焦开发能力。其具体公式如下:

1) 自由度 ν 动态衰减公式

$$\nu(t) = \nu_1 - (\nu_1 - \nu_2) \cdot \frac{t}{T} \quad (3.1.1)$$

其中 t 为当前迭代次数, T 为最大迭代次数, $\nu_1 = 30$ 为自由度的初始值, $\nu_2 = 1$ 为自由度的终值,当 ν 的值去向无穷大时,可以等同于标准的高斯分布,当 $\nu = 1$ 时,则可等同于柯西分布[14]。

2) 扰动生成公式:

$$X_i(t+1) = X_i(t) + \eta \cdot T(\nu(t)) \cdot (X_{best}(t) - X_i(t)) \quad (3.1.2)$$

受文献[15]中柯西分布、t 分布和高斯分布函数分布图的启发,经过实验确定扰动强度系数 η 的取值在区间 $[0.1, 0.5]$ 内, $T(\nu(t))$ 为服从自由度为 $\nu(t)$ 的 T 分布随机数(表 2)。

Table 2. Pseudocode of the TOBLAOA algorithm**表 2.** T 分布扰动及透镜成像反向学习的算术优化算法伪代码

算法 2 T 分布扰动及透镜成像反向学习的算术优化算法(TOBLAOA)

1. 输入：初群体规模 N 、最大迭代次数、令 $t := 0$ ，初始化参数 $X_i(t)$ 、 v 初始值 $v_1 = 30$ ，终值 $v_2 = 1$ 、取 $\mu = 0.499$ 、 $a = 5$ 、 $Min = 0.2$ 、 $Max = 1$
2. 初始化：生成初始种群 $x_i(t) (i = 1, 2, \dots, N)$
3. 计算初始适应度 $f(x)$ ，记录全局最优解 $X_{best}(t)$ 和适应度
4. While $t < T$ do
5. 更新 $MOA(t) = Min + t \times \frac{Max - Min}{T}$
6. 更新 $MOP(t) = 1 - \left(\frac{t}{T}\right)^{\frac{1}{a}}$
7. 更新 $n(t) = \left(1 + \left(\frac{t}{T}\right)^{\frac{1}{2}}\right)^{10}$
8. for $i = 1$ to N do
9. 生成随机数 r_1 、 r_2 、 r_3 及透镜成像的上下界 a_i 、 b_i
10. if $r_1 > MOA(t)$ then (执行乘除法搜索策略)
11.
$$X_i(t+1) = \begin{cases} X_{best}(t) \div (MOP + \varepsilon) \times ((Ub - Lb) \times \mu + Lb), & r_2 < 0.5 \\ X_{best}(t) \times MOP \times ((Ub - Lb) \times \mu + Lb), & \text{其他} \end{cases}$$
8. Else (执行加减法搜索策略)
9.
$$X_i(t+1) = \begin{cases} X_{best}(t) - MOP \times ((Ub - Lb) \times \mu + Lb), & r_3 < 0.5 \\ X_{best}(t) + MOP \times ((Ub - Lb) \times \mu + Lb), & \text{其他} \end{cases}$$
10. end if
11. 根据 $\frac{(a_i + b_i)/2 - X_{best}(t)}{X_{best}(t) - (a_i + b_i)/2} = \frac{h}{h^*}$ 更新种群原始最优解 $X_{best}(t)$ $v(t) = v_1 - (v_1 - v_2) \cdot \frac{t}{T}$
12. 根据 $X_{best}^*(t) = \frac{a_i + b_i}{2} + \frac{a_i + b_i}{2n} - \frac{X_{best}(t)}{n}$ 计算原始反向最优解 $X_{best}^*(t)$
13. 将原始反向最优解 $X_{best}^*(t)$ 替换原始最优解 $X_{best}(t)$ 后参与迭代
14. if $t > T$ do
15. 计算动态自由度
16. 计算最优扰动方向 $X_i(t+1) = X_i(t) + \eta \cdot T(v(t)) \cdot (X_{best}(t) - X_i(t))$
13. end if
- 更新个体最优
14. end for
15. $t = t + 1$
16. end while
17. 输出全局最优解

3.2. 差分变异扰动策略(DMOBLAOA)

差分变异扰动策略(Differential Mutation Perturbation, DMP)差分变异通过种群中个体的向量差异生成扰动,生成新解,增强搜索方向的多样性,并可根据缩放因子通知扰动强度,控制扰动幅度的全局探索能力,当 $F > 1$ 时,放大差异向量,则增强全局搜索,当 $F < 1$ 时,压制差异向量,则偏向局部开发,具体公式如下:

$$V_i^{t+1} = X_{r_1}^t + F \cdot (X_{r_2}^t - X_{r_3}^t) \quad (3.2.1)$$

其中, $X_{r_1}, X_{r_2}, X_{r_3}$ 为迭代中随机选取的三个不同的个体, 并且 $r_1 \neq r_2 \neq r_3$; F 为区间 $[0, 2]$ 的放缩因子, 通过控制扰动幅度来控制对全局的探索能力。

3.3. 三角形游走策略(TWOBLAOA)

三角形游走策略(Triangular Walk Strategy, TWS)基于三角形概率分布生成随机步长, 其概率密度函数在区间 $[a, b]$ 内呈对称或不对称的三角形形状, 其形式为:

$$f(x) = \begin{cases} \frac{2(x-a)}{(b-a)(c-a)}, & a \leq x \leq c \\ \frac{2(b-x)}{(b-a)(b-c)}, & c \leq x \leq b \end{cases} \quad (3.3.1)$$

其中 a, b 为分布的范围边界, 即搜索空间; c 为众数的位置, 即计算过程中当前最优解的附近。

3.4. 高斯游走策略(GWOBLAOA)

高斯游走策略(Gaussian Random Walk, GRW), 扰动步长服从高斯分布, 强调局部精细化搜索, 步长自适应缩小, 收敛稳定性强, 高维条件下仍能保持解的质量。

$$X_i^{t+1} = X_i^t + \sigma(t) \cdot N(0, 1) \quad (3.4.1)$$

其中 $\sigma(t)$ 为标准差, 会随着迭代动态衰减。

4. 仿真实验及结果分析

4.1. 基本参数设置

本仿真测试环境为: 操作系统 Windows 10, 处理器 12th Gen Intel(R) Core(TM) i7-12700H, 主频 2.30 GHz, 内存 16.0 GB。

Table 3. Algorithm parameter settings
表 3. 算法参数设置

算法	参数设置
所有算法	$N = 30; T = 300$
TOBLAOA	$v_1 = 30; v_2 = 1$
HHO	$c_1 = 2; c_2 = 2$
SA	$Tol = 10E-10$
SCA	$a = 2$

本文首先基于不同改进策略下的 OBLAOA 算法(即 TOBLAOA、DWOBLAOA、TWOBLAOA、

SWOBLAOA、AOA)进行对比, 再与蝴蝶优化算法(BOA)、雷电附着优化算法(LAPO) [16]、哈里斯鹰优化算法(HHO)和遗传算法(SA)、黄金正弦算法(SCA)、神经网络优化算法(NNA) [17]算法进行对比。为保证实验的公平性、可比性、客观性, 本文对一些算法的参数数值的设定进行了限制, 具体要求见表 3。

4.2. 基准测试函数

本文选取了不同测试函数集中具有不同性质的测试函数进行仿真实验。选取 2005 测试集中具有单峰($F_1 \sim F_5$)和多峰($F_6 \sim F_8$)特性的测试函数, 选取 2017 测试集中具有多峰($F_9 \sim F_{10}$)特性的测试函数, 选取 2019 测试集中单目标测试函数($F_{11} \sim F_{15}$), 对上述算法逐一进行仿真实验测试, 测试函数具体信息如表 4~6 所示。

Table 4. CEC2005 benchmark test functions table

表 4. CEC2005 基准测试函数表

函数表达式	维度	范围	理论最小值
$F_1(x) = \sum_{i=1}^n x_i^2$	30	$[-100, 100]$	0
$F_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i \sum_{i=1}^n X_i Y_i$	30	$[-10, 10]$	0
$F_3(x) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2$	30	$[-100, 100]$	0
$F_4(x) = \max_i \{ x_i , 1 \leq i \leq n\}$	30	$[-100, 100]$	0
$F_5(x) = \sum_{i=1}^n i x_i^4 + \text{random}[0,1]$	30	$[-1.28, 1.28]$	0
$F_6(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10] - 418.98 \times \text{Dim}^n$	30	$[-5.12, 5.12]$	0
$F_7(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$	30	$[-30, 32]$	0
$F_8(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	$[-600, 600]$	0

Table 5. CEC2017 benchmark test functions table

表 5. CEC2017 基准测试函数表

函数表达式	维度	范围	理论最小值
$F_9(x) = g(x_1, x_2) + g(x_2, x_3) + \dots + g(x_{D-1}, x_D) + g(x_D, x_1)$ $g(x, y) = 0.5 + \frac{\sin^2\left(\sqrt{x^2 + y^2}\right) - 0.5}{1 + 0.001(x^2 + y^2)^2}$	30/100	$[-100, 100]$	600
$F_{10}(x) = \min\left(\sum_{i=1}^D (\bar{x}_i - \mu_0)^2, dD + s \sum_{i=1}^D (\bar{x}_i - \mu_1)^2\right) + 10\left(D - \sum_{i=1}^D \cos(2\pi \bar{z}_i)\right)$ $\mu_0 = 2.5, \mu_1 = -\sqrt{\frac{\mu_0^2 - d}{s}}, s = 1 - \frac{1}{2\sqrt{D+20} - 8.2}, d = 1$ $y = \frac{10(x - o)}{100}, \bar{x}_i = 2\text{sign}(x_i^*)y_i + \mu_0, \text{ for } i = 1, 2, \dots, D$ $z = \Lambda^{100}(\bar{x}_i - \mu_0)$	30/100	$[-100, 100]$	600

Table 6. CEC2019 benchmark test functions table
表 6. CEC2019 基准测试函数表

函数	函数名称	维度	范围	理论最小值
$F_{11}(x)$	Stron's Chebyshev Polynomial Fitting Problem	9	$[-8192, 8192]$	1
$F_{12}(x)$	Inverse Hilbert Matrix Problem	16	$[-16,384, 16,384]$	1
$F_{13}(x)$	Rastrigin's Function	10	$[-100, 100]$	1
$F_{14}(x)$	Modified Schwefel's Function	10	$[-100, 100]$	1
$F_{15}(x)$	Expanded Schaffer's F6 Function	10	$[-100, 100]$	1

4.3. 不同改进策略算法的对比结果

对第 3 小节中给出的 4 种嵌入改进策略的 OBLAOA 算法与 TOBLAOA 进行对比测试。为了比较寻优性能, 分别从 min (最优适应度)、std (标准差)、avg (平均适应度)、worse (最差适应度)四个指标进行评价。每个算法均在基准测试函数上独立运行三十次, 具体结果如表 7:

Table 7. Experimental results of TOBLAOA with different improvement strategies
表 7. 不同改进策略的 TOBLAOA 实验结果表

测试函数	Index	TOBLAOA	DMOBLAOA	TWEOBLAOA	GWOBLAOA	AOA
F_1	min	0	0	0	0	5.89E-189
	std	0	0	0	0	0
	avg	0	0	0	0	5.97E-177
	worse	0	0	0	0	2.95E-176
F_2	min	0	0	0	0	3.36E-91
	std	0	0	0	0	6.92E-77
	avg	0	0	0	0	3.09E-77
	worse	0	0	0	0	1.55E-76
F_3	min	0	0	0	0	2.18E-167
	std	0	0	0	0	5.57E-141
	avg	0	0	0	0	2.49E-141
	worse	0	0	0	0	1.25E-140
F_4	min	0	0	0	0	6.30E-85
	std	0	0	0	0	8.97E-72
	avg	0	0	0	0	4.01E-72
	worse	0	0	0	0	2.01E-71
F_5	min	4.71E-05	3.53E-05	5.51E-05	4.21E-05	2.91E-04
	std	2.14E-05	1.63E-05	3.03E-05	1.55E-05	1.87E-04
	avg	5.78E-06	1.16E-06	1.14E-05	5.71E-06	5.45E-04
	worse	1.42E-05	1.42E-05	1.82E-05	8.04E-06	7.09E-04
F_6	min	-3.63E+03	-3.13E+03	-2.58E+03	-2.59E+03	-4.23E+03
	std	-4.09E+03	-3.40E+03	-2.95E+03	-3.03E+03	387.6514
	avg	-5.03E+03	-4.31E+03	-3.15E+03	-3.31E+03	-3.89E+03
	worse	-3.92E+03	-3.19E+03	-3.12E+03	-3.03E+03	-3.27E+03

续表

F_7	min	0	0	0	0	0
	std	0	0	0	0	0
	avg	0	0	0	0	0
	worse	0	0	0	0	0
F_8	min	4.44E-16	4.44E-16	4.44E-16	4.44E-16	4.44E-16
	std	4.44E-16	4.44E-16	4.44E-16	4.44E-16	0
	avg	4.44E-16	4.44E-16	4.44E-16	4.44E-16	4.44E-16
	worse	4.44E-16	4.44E-16	4.44E-16	4.44E-16	4.44E-16
F_9	min	0	0	0	0	5.89E-189
	std	0	0	0	0	0
	avg	0	0	0	0	5.97E-177
	worse	0	0	0	0	2.95E-176
F_{10}	min	0	0	0	0	3.36E-91
	std	0	0	0	0	6.92E-77
	avg	0	0	0	0	3.09E-77
	worse	0	0	0	0	1.55E-76
F_{11}	min	1	1	1	1	1
	std	1	1	1	1	1.32E-13
	avg	1	1	1	1	1
	worse	0	0	0	0	1
F_{12}	min	1	1	1	1	1
	std	5	5	5	5	4.4614
	avg	5	5	5	5	0.2325
	worse	5	5	5	5	4.8636
F_{13}	min	0	0	0	8.88E-16	4.9956
	std	5	5	5	5	5
	avg	1	1	1	1	1
	worse	1	1	1	1	0
F_{14}	min	1	1	1	1	1
	std	0	0	0	0	1
	avg	1	1	1	1	1
	worse	1	1	1	1	1
F_{15}	min	1	1	1	1	0
	std	1	1	1	1	1
	avg	0	0	0	0	1
	worse	1	1	1	1	1

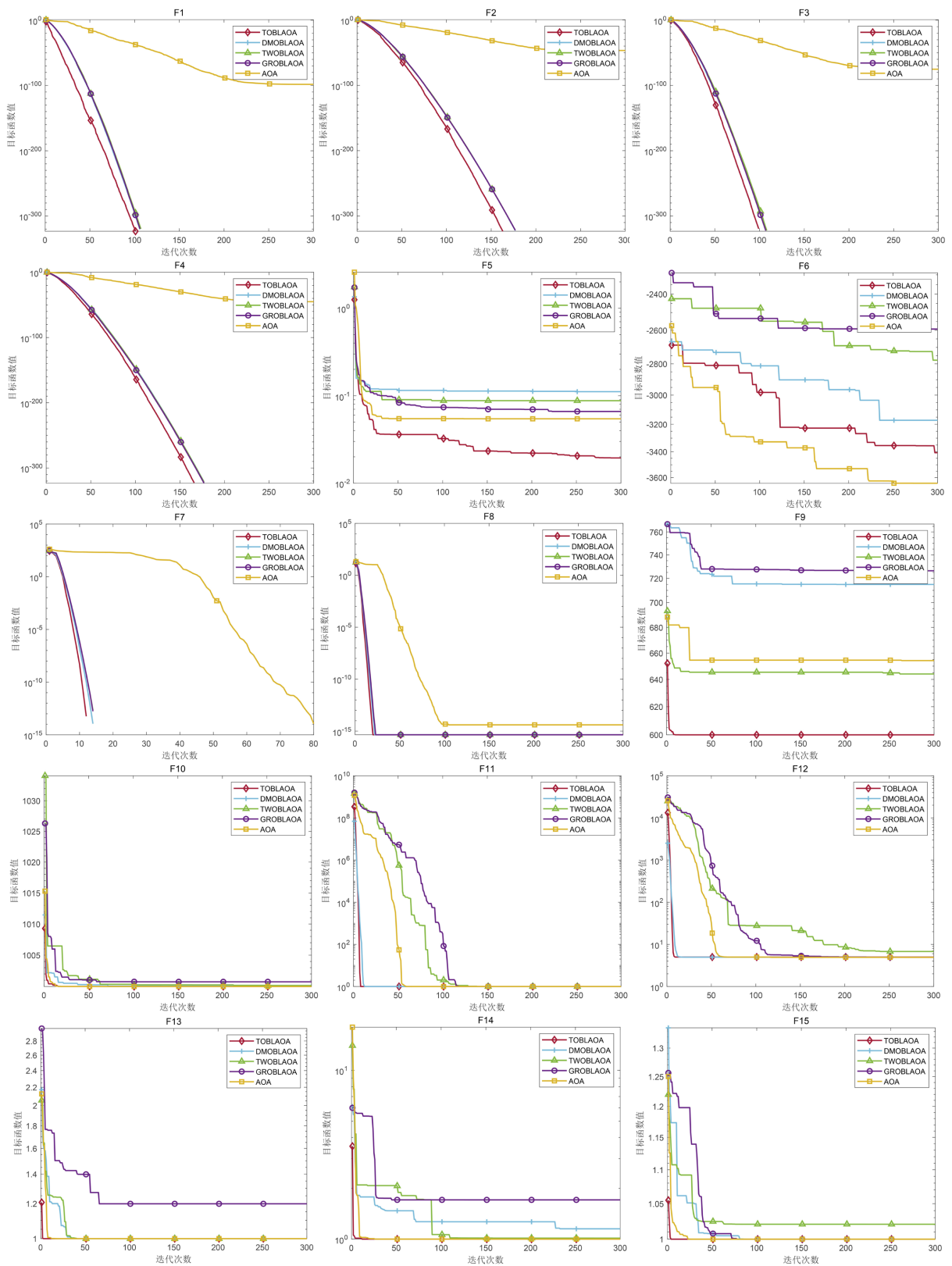


Figure 2. Convergence curves of TOBLAOA with different improvement strategies on benchmark test functions

图 2. TOBLAOA 不同改进策略实验测试函数收敛曲线

根据实验结果可知, 在 OBLAOA 算法改进后的算法, 其寻优性能都远远地优于原始的 AOA 算法, 其中 AOA 只搜索到了 $F_7, F_{11}, F_{12}, F_{14}$ 的理论最优值, 而给出的四种策略都可以直接搜索到函数 $F_1, F_2, F_3, F_4, F_7, F_9, F_{10}, F_{11}, F_{12}, F_{14}, F_{15}$ 的理论最优值, 与传统 AOA 相比有显著的提升; 对于没有直接搜索到理论最优值的函数 F_5, F_6, F_8, F_{13} , 将 TOBLAOA 与 AOA 进行比较可知, TOBLAOA 的最有适应度值的精度也优于 AOA。上述实验结果说明, 在这 5 种嵌入改进策略的算法中, OBLAOA 对于原始 AOA 的改进策略最为有效, 在提高求解精度的同时还提升了稳定性。但是由于在数据上对于集中改进策略比较不明显, 故继续对 TOBLAOA、DWOBLAOA、TWOBLAOA、SWOBLAOA、AOA 进行仿真测试, 对比在不同基准测试函数中算法的收敛速度, 得到收敛曲线图, 见图 2。

仿真实验结果表明, 对于所有基准测试函数 TOBLAOA 可以在更短的迭代次数内更快地接近目标函数值, 其中由函数 F_5, F_8 的收敛曲线图可知, TOBLAOA 的收敛曲线呈现多处曲折, 这表明在收敛过程中, TOBLAOA 多次且有效地跳出局部最优解, 以更快的收敛速度逼近最优解, 对于部分高维的函数 F_{10} 到 F_{15} , 也可以看出, 即使各改进策略之间差异非常小, TOBLAOA 仍然是最快接近最优解的策略, 以上结果都表明 TOBLAOA 具有更好的寻优性能跟收敛速度。

4.4. TOBLAOA 与其它优化算法的对比分析

为进一步验证 TOBLAOA 的有效性和求解精度, 接下来将其与 BOA、LAOP、HHO、SA、SCA、NNA 进行比较, 通过 min (最优适应度)、std (标准差)、avg (平均适应度)、worse (最差适应度)四个指标进行评价。表 8 为各算法在每个基准测试函数上独立运行 30 次所得到的结果:

Table 8. Comparison table of experimental results of different algorithms of TOBLAOA (Dim = 50)

表 8. TOBLAOA 不同算法实验结果对比表(Dim = 50)

$D = 50$	index	OBLAOA	BOA	LAPO	HHO	SA	SCA	NNA
F_1	min	0	2.71E-12	8.06E-61	1.10E-105	1.35E-26	4.66E-52	1.71E-23
	std	0	2.27E-12	2.14E-61	2.20E-106	1.15E-25	3.74E-45	2.88E-17
	avg	0	1.91E-12	3.68E-62	1.78E-117	1.28E-25	1.68E-45	1.17E-17
	worse	0	2.22E-12	6.95E-62	4.75E-113	2.83E-25	8.38E-45	1.37E-16
F_2	min	0	1.72E-09	6.53E-29	2.57E-57	5.39E-23	1.01E-29	9.41E-13
	std	0	1.33E-09	2.46E-29	5.25E-58	3.14E-18	9.71E-27	2.71E-10
	avg	0	1.07E-09	6.54E-30	9.54E-62	1.72E-18	9.34E-27	1.99E-10
	worse	0	1.22E-09	2.07E-29	1.43E-59	7.26E-18	2.59E-26	9.47E-10
F_3	min	0	2.48E-12	5.90E-52	4.95E-104	8.29E-09	2.06E-46	0.8841
	std	0	2.00E-12	1.94E-52	9.91E-105	1.71E-07	7.66E-38	30.3553
	avg	0	1.35E-12	2.77E-53	4.21E-115	9.00E-08	3.80E-38	32.8361
	worse	0	2.34E-12	8.71E-53	1.93E-108	3.95E-07	1.74E-37	123.7664
F_4	min	0	1.42E-09	1.26E-25	7.44E-55	4.53E-11	3.16E-24	0.0012
	std	0	1.14E-09	6.14E-26	1.60E-55	1.52E-09	4.91E-21	0.0066
	avg	0	8.28E-10	2.44E-26	5.68E-59	1.39E-09	2.47E-21	0.0079
	worse	0	1.20E-09	3.89E-26	1.25E-56	3.37E-09	1.12E-20	0.0276
F_5	min	4.11E-05	0.0019	0.0012	1.03E-04	0.0871	0.0112	0.0048
	std	1.55E-05	0.001	5.61E-04	4.11E-05	0.0609	0.0781	0.0197
	avg	3.39E-06	4.89E-04	1.28E-04	1.59E-05	0.1547	0.074	0.0286
	worse	1.41E-05	8.22E-04	5.08E-04	3.29E-05	0.2457	0.1853	0.0733

续表

F_6	min	-4.00E+03	-3.61E+03	-4.70E+03	-1.26E+04	-1.25E+04	-4.06E+03	0
	std	-4.32E+03	-3.85E+03	-5.17E+03	-1.26E+04	164.4148	231.5899	0.375
	avg	-4.45E+03	-4.27E+03	-6.04E+03	-1.26E+04	-1.22E+04	-3.84E+03	0.1722
	worse	-4.40E+03	-3.70E+03	-4.94E+03	-1.26E+04	-1.21E+04	-3.59E+03	0.995
F_7	min	0	165.6542	3.90E-10	0	0.9951	0.0012	2.67E-12
	std	0	33.1308	8.84E-11	0	1.5068	57.7601	1.06E-08
	avg	0	0	7.39E-13	0	2.5937	57.9847	3.95E-09
	worse	0	0	5.12E-12	0	4.9748	135.1036	5.45E-08
F_8	min	4.44E-16	6.19E-09	1.56E-06	4.44E-16	1.02E-04	0.0221	0.0197
	std	4.44E-16	5.75E-09	7.81E-07	4.44E-16	0.0025	0.168	0.0385
	avg	4.44E-16	5.39E-09	1.59E-07	4.44E-16	0.0024	0.2733	0.0707
	worse	4.44E-16	5.73E-09	9.47E-07	4.44E-16	0.0063	0.477	0.1599
F_9	min	600	600.2146	600.0001	600.2529	600	600.0027	600.0419
	std	600	600.203	600	600.1711	4.05E-04	0.0046	600.0086
	avg	600	600.1862	600	600.1027	600.0003	600.007	600
	worse	600	600.206	600	600.1254	600.001	600.014	600.0005
F_{10}	min	700	700	710.0106	700	700	700	710.0106
	std	700	700	705.0751	700	4.4943	1.49E-12	708.0085
	avg	700	700	700.0079	700	7.08E+02	700	700
	worse	700	700	7.04E+02	700	710.0876	700	710.0106
F_{11}	min	1	3.03E+04	946.9517	1	1	1.0171	1
	std	0	1.64E+05	2.81E+04	0	0	2.25E+06	0
	avg	1	2.19E+05	2.29E+04	1	1	1.22E+06	1
	worse	1	4.45E+05	7.08E+04	1	1	5.23E+06	1
F_{12}	min	4.9975	415.7586	68.5309	5	2.46E+03	1.55E+03	1.75E+03
	std	0.0011	406.1479	46.6666	0	3.42E+03	1.56E+03	2.36E+03
	avg	4.9995	984.5102	122.6961	5	5.75E+03	3.91E+03	5.81E+03
	worse	5	1.55E+03	189.009	5	1.15E+04	5.88E+03	1.03E+04
F_{13}	min	1	1	1	1	1	1	1
	std	0	0	5.58E-10	0	2.97E-13	0	0
	avg	1	1	1	1	1	1	1
	worse	1	1	1	1	1	1	1
F_{14}	min	1	1	1	1	1	1	1
	std	0	0	2.19E-05	0	0.0342	0	0
	avg	1	1	1	1	1.025	1	1
	worse	1	1	1.0001	1	1.0625	1	1
F_{15}	min	1	1	1	1	1	1	1
	std	0	0	3.17E-09	0	6.45E-14	0	0
	avg	1	1	1	1	1	1	1
	worse	1	1	1	1	1	1	1

根据实验结果可知,在寻优性能上,TOBLAOA 可以直接搜索到函数 $F_1, F_2, F_3, F_4, F_7, F_9, F_{11}, F_{13}, F_{14}, F_{15}$ 的理论最优值,而 BOA 和 HHO 只搜索到了 $F_{11}, F_{13}, F_{14}, F_{15}$ 的理论最优值。NNA 只搜索到了 $F_6, F_{11}, F_{13}, F_{14}, F_{15}$ 的理论最优值,SA 和 SCA 只搜索到了 $F_6, F_{11}, F_{13}, F_{14}, F_{15}$ 的理论最优值并且标准差不为 0, OBLAOA 算法更具有稳定性。对于没有直接搜索到理论最优值的函数 F_5, F_6, F_8 , TOBLAOA 求解精度更高。上述实验结果表明,与其他算法相比,TOBLAOA 在求解精度和稳定性上都具有可比性。为进一步验证该结论,本文还对各算法在高维情况下的表现进行仿真实验,以下表 9, 表 10 算法在 100 维和 500 维下的实验结果。

Table 9. Comparison table of experimental results of different algorithms of TOBLAOA ($Dim = 100$)

表 9. TOBLAOA 不同算法实验结果对比表($Dim = 100$)

$D = 100$	index	TOBLAOA	BOA	LAPO	HHO	SA	SCA	NNA
F_1	min	0	2.71E-12	8.06E-61	1.10E-105	1.35E-26	4.66E-52	1.78E-100
	std	0	2.27E-12	2.14E-61	2.20E-106	1.15E-25	3.74E-45	6.69E-102
	avg	0	1.91E-12	3.68E-62	1.78E-117	1.28E-25	1.68E-45	8.93E-119
	worse	0	2.22E-12	6.95E-62	4.75E-113	2.83E-25	8.38E-45	7.05E-110
F_2	min	0	1.72E-09	6.53E-29	2.57E-57	5.39E-23	1.01E-29	3.12E-51
	std	0	1.33E-09	2.46E-29	5.25E-58	3.14E-18	9.71E-27	1.05E-52
	avg	0	1.07E-09	6.54E-30	9.54E-62	1.72E-18	9.34E-27	1.21E-61
	worse	0	1.22E-09	2.07E-29	1.43E-59	7.26E-18	2.59E-26	9.73E-57
F_3	min	0	2.48E-12	5.90E-52	4.95E-104	8.29E-09	2.06E-46	1.42E-87
	std	0	2.00E-12	1.94E-52	9.91E-105	1.71E-07	7.66E-38	4.75E-89
	avg	0	1.35E-12	2.77E-53	4.21E-115	9.00E-08	3.80E-38	4.40E-107
	worse	0	2.34E-12	8.71E-53	1.93E-108	3.95E-07	1.74E-37	1.09E-96
F_4	min	0	1.42E-09	1.26E-25	7.44E-55	4.53E-11	3.16E-24	1.86E-50
	std	0	1.14E-09	6.14E-26	1.60E-55	1.52E-09	4.91E-21	1.01E-51
	avg	0	8.28E-10	2.44E-26	5.68E-59	1.39E-09	2.47E-21	1.73E-59
	worse	0	1.20E-09	3.89E-26	1.25E-56	3.37E-09	1.12E-20	7.12E-55
F_5	min	4.11E-05	0.0019	0.0012	1.03E-04	0.0871	0.0112	4.15E-04
	std	1.55E-05	0.001	5.61E-04	4.11E-05	0.0609	0.0781	8.85E-05
	avg	3.39E-06	4.89E-04	1.28E-04	1.59E-05	0.1547	0.074	7.73E-07
	worse	1.41E-05	8.22E-04	5.08E-04	3.29E-05	0.2457	0.1853	6.68E-05
F_6	min	-4.00E+03	-3.61E+03	-4.70E+03	-1.26E+04	-1.25E+04	-4.06E+03	0
	std	-4.32E+03	-3.85E+03	-5.17E+03	-1.26E+04	164.4148	231.5899	0
	avg	-4.45E+03	-4.27E+03	-6.04E+03	-1.26E+04	-1.22E+04	-3.84E+03	0
	worse	-4.40E+03	-3.70E+03	-4.94E+03	-1.26E+04	-1.21E+04	-3.59E+03	0
F_7	min	0	165.6542	3.90E-10	0	0.9951	0.0012	4.44E-16
	std	0	33.1308	8.84E-11	0	1.5068	57.7601	4.44E-16
	avg	0	0	7.39E-13	0	2.5937	57.9847	4.44E-16
	worse	0	0	5.12E-12	0	4.9748	135.1036	4.44E-16

续表

F_8	min	4.44E-16	6.19E-09	1.56E-06	4.44E-16	1.02E-04	0.0221	0
	std	4.44E-16	5.75E-09	7.81E-07	4.44E-16	0.0025	0.168	0
	avg	4.44E-16	5.39E-09	1.59E-07	4.44E-16	0.0024	0.2733	0
	worse	4.44E-16	5.73E-09	9.47E-07	4.44E-16	0.0063	0.477	0
F_9	min	600	6.19E-09	1.56E-06	4.44E-16	1.02E-04	0.0221	4.44E-16
	std	600	5.75E-09	7.81E-07	4.44E-16	0.0025	0.168	4.44E-16
	avg	600	5.39E-09	1.59E-07	4.44E-16	0.0024	0.2733	4.44E-16
	worse	600	5.73E-09	9.47E-07	4.44E-16	0.0063	0.477	4.44E-16
F_{10}	min	700.0845	600.2165	600	600.2529	600	600.0007	600.1238
	std	700.0212	600.1946	600	600.2026	0.0034	0.0033	600.0267
	avg	700.0001	600.1853	600	600.0015	600.0019	600.0045	600.0001
	worse	700.0081	600.1859	600	600.2529	600.0078	600.0084	600.0007
F_{11}	min	1	4.38E+06	3.32E+04	1	1	2.04E+06	10.4068
	std	1	1.62E+06	2.00E+04	1	1	1.19E+06	2.57E+06
	avg	1	1.33E+05	3.93E+03	1	1	2.71E+05	1.99E+06
	worse	0	1.77E+06	1.05E+04	0	0	8.67E+05	9.33E+06
F_{12}	min	5	1.71E+03	204.3887	5	7.84E+03	5.38E+03	975.302
	std	5	875.0103	123.6975	4.9173	5.00E+03	2.81E+03	1.65E+03
	avg	5	78.1537	66.4558	4.5863	3.84E+03	1.54E+03	3.14E+03
	worse	4.27E-09	619.6823	60.6121	0.185	1.63E+03	1.60E+03	7.01E+03
F_{13}	min	1	1	1	1	1	1	1
	std	0	0	3.28E-09	0	6.07E-14	0	0
	avg	1	1	1	1	1	1	1
	worse	1	1	1	1	1	1	1
F_{14}	min	1	1	1	1	1	1	1
	std	0	0	1.30E-05	0	5.1927	0	0
	avg	1	1	1	1	3.3534	1	1
	worse	1	1	1	1	12.6422	1	1
F_{15}	min	1	1	1	1	1	1	2
	std	0	0	1.80E-09	0	9.98E-14	0	0
	avg	1	1	1	1	1	1	2
	worse	1	1	1	1	1	1	2

Table 10. Comparison table of experimental results of different algorithms of TOBLAOA ($Dim = 500$)**表 10.** TOBLAOA 不同算法实验结果对比表($Dim = 500$)

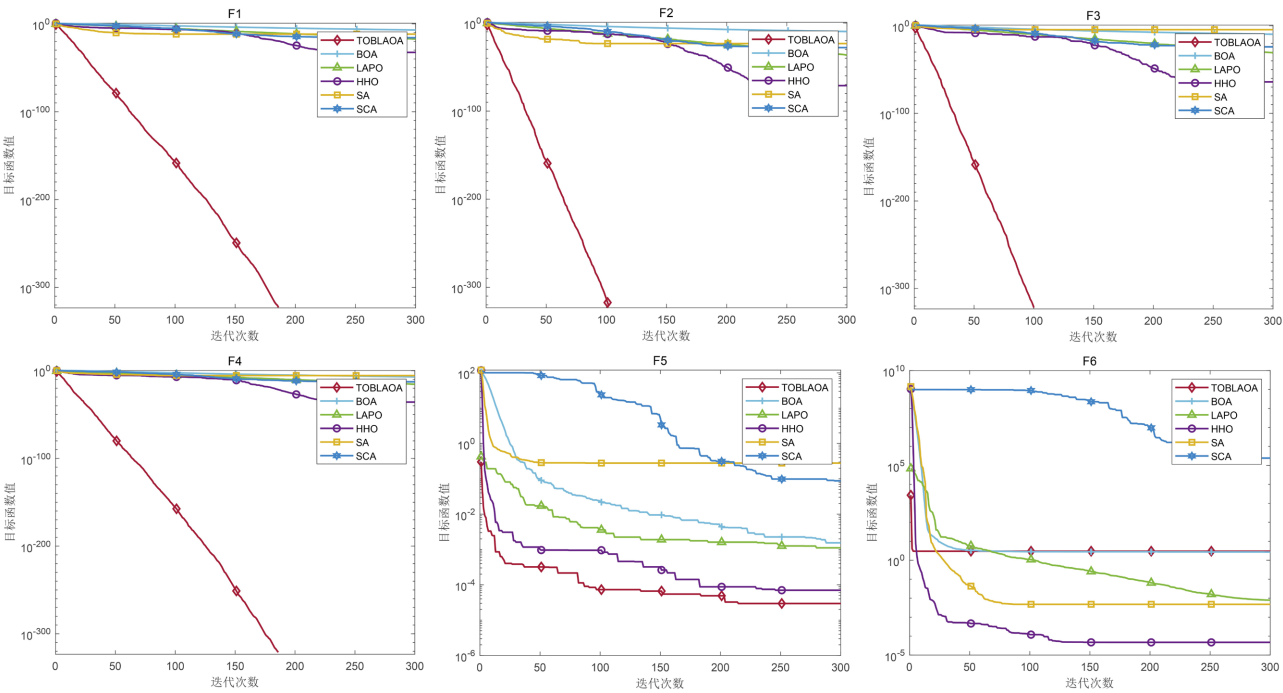
$D = 500$	index	TOBLAOA	BOA	LAPO	HHO	SA	SCA	NNA
F_1	min	0	2.71E-12	8.06E-61	1.10E-105	1.35E-26	4.66E-52	7.13E-96
	std	0	2.27E-12	2.14E-61	2.20E-106	1.15E-25	3.74E-45	2.38E-97
	avg	0	1.91E-12	3.68E-62	1.78E-117	1.28E-25	1.68E-45	2.92E-125
	worse	0	2.22E-12	6.95E-62	4.75E-113	2.83E-25	8.38E-45	1.08E-107

续表

F_2	min	0	1.72E-09	6.53E-29	2.57E-57	5.39E-23	1.01E-29	5.49E-54
	std	0	1.33E-09	2.46E-29	5.25E-58	3.14E-18	9.71E-27	2.80E-55
	avg	0	1.07E-09	6.54E-30	9.54E-62	1.72E-18	9.34E-27	3.36E-62
	worse	0	1.22E-09	2.07E-29	1.43E-59	7.26E-18	2.59E-26	7.88E-57
F_3	min	0	2.48E-12	5.90E-52	4.95E-104	8.29E-09	2.06E-46	2.95E-91
	std	0	2.00E-12	1.94E-52	9.91E-105	1.71E-07	7.66E-38	2.06E-92
	avg	0	1.35E-12	2.77E-53	4.21E-115	9.00E-08	3.80E-38	8.04E-112
	worse	0	2.34E-12	8.71E-53	1.93E-108	3.95E-07	1.74E-37	7.47E-98
F_4	min	0	1.42E-09	1.26E-25	7.44E-55	4.53E-11	3.16E-24	1.36E-49
	std	0	1.14E-09	6.14E-26	1.60E-55	1.52E-09	4.91E-21	4.94E-51
	avg	0	8.28E-10	2.44E-26	5.68E-59	1.39E-09	2.47E-21	4.65E-58
	worse	0	1.20E-09	3.89E-26	1.25E-56	3.37E-09	1.12E-20	5.40E-55
F_5	min	4.11E-05	0.0019	0.0012	1.03E-04	0.0871	0.0112	4.17E-04
	std	1.55E-05	0.001	5.61E-04	4.11E-05	0.0609	0.0781	9.56E-05
	avg	3.39E-06	4.89E-04	1.28E-04	1.59E-05	0.1547	0.074	8.84E-07
	worse	1.41E-05	8.22E-04	5.08E-04	3.29E-05	0.2457	0.1853	7.13E-05
F_6	min	-4.00E+03	-3.61E+03	-4.70E+03	-1.26E+04	-1.25E+04	-4.06E+03	0
	std	-4.32E+03	-3.85E+03	-5.17E+03	-1.26E+04	164.4148	231.5899	0
	avg	-4.45E+03	-4.27E+03	-6.04E+03	-1.26E+04	-1.22E+04	-3.84E+03	0
	worse	-4.40E+03	-3.70E+03	-4.94E+03	-1.26E+04	-1.21E+04	-3.59E+03	0
F_7	min	0	165.6542	3.90E-10	0	0.9951	0.0012	4.44E-16
	std	0	33.1308	8.84E-11	0	1.5068	57.7601	4.44E-16
	avg	0	0	7.39E-13	0	2.5937	57.9847	4.44E-16
	worse	0	0	5.12E-12	0	4.9748	135.1036	4.44E-16
F_8	min	4.44E-16	6.19E-09	1.56E-06	4.44E-16	1.02E-04	0.0221	0
	std	4.44E-16	5.75E-09	7.81E-07	4.44E-16	0.0025	0.168	0
	avg	4.44E-16	5.39E-09	1.59E-07	4.44E-16	0.0024	0.2733	0
	worse	4.44E-16	5.73E-09	9.47E-07	4.44E-16	0.0063	0.477	0
F_9	min	1	3.03E+04	946.9517	1	1	1.0171	600.2529
	std	0	1.64E+05	2.81E+04	0	0	2.25E+06	600.1938
	avg	1	2.19E+05	2.29E+04	1	1	1.22E+06	600.0077
	worse	1	4.45E+05	7.08E+04	1	1	5.23E+06	600.2529
F_{10}	min	4.9975	415.7586	68.5309	5	2.46E+03	1.55E+03	700.0005
	std	0.0011	406.1479	46.6666	0	3.42E+03	1.56E+03	700
	avg	4.9995	984.5102	122.6961	5	5.75E+03	3.91E+03	700
	worse	5	1.55E+03	189.009	5	1.15E+04	5.88E+03	700

续表

F_{11}	min	1	1	1	1	1	1	1
	std	0	0	5.58E-10	0	2.97E-13	0	1
	avg	1	1	1	1	1	1	1
	worse	1	1	1	1	1	1	1
F_{12}	min	1	1	1	1	1	1	5
	std	0	0	2.19E-05	0	0.0342	0	4.995
	avg	1	1	1	1	1.025	1	4.8492
	worse	1	1	1.0001	1	1.0625	1	5
F_{13}	min	1	1	1	1	1	1	1
	std	0	0	3.17E-09	0	6.45E-14	0	1
	avg	1	1	1	1	1	1	1
	worse	1	1	1	1	1	1	1
F_{14}	min	0	2.71E-12	8.06E-61	1.10E-105	1.35E-26	4.66E-52	1
	std	0	2.27E-12	2.14E-61	2.20E-106	1.15E-25	3.74E-45	1
	avg	0	1.91E-12	3.68E-62	1.78E-117	1.28E-25	1.68E-45	1
	worse	0	2.22E-12	6.95E-62	4.75E-113	2.83E-25	8.38E-45	1
F_{15}	min	0	1.72E-09	6.53E-29	2.57E-57	5.39E-23	1.01E-29	2
	std	0	1.33E-09	2.46E-29	5.25E-58	3.14E-18	9.71E-27	2
	avg	0	1.07E-09	6.54E-30	9.54E-62	1.72E-18	9.34E-27	2
	worse	0	1.22E-09	2.07E-29	1.43E-59	7.26E-18	2.59E-26	2



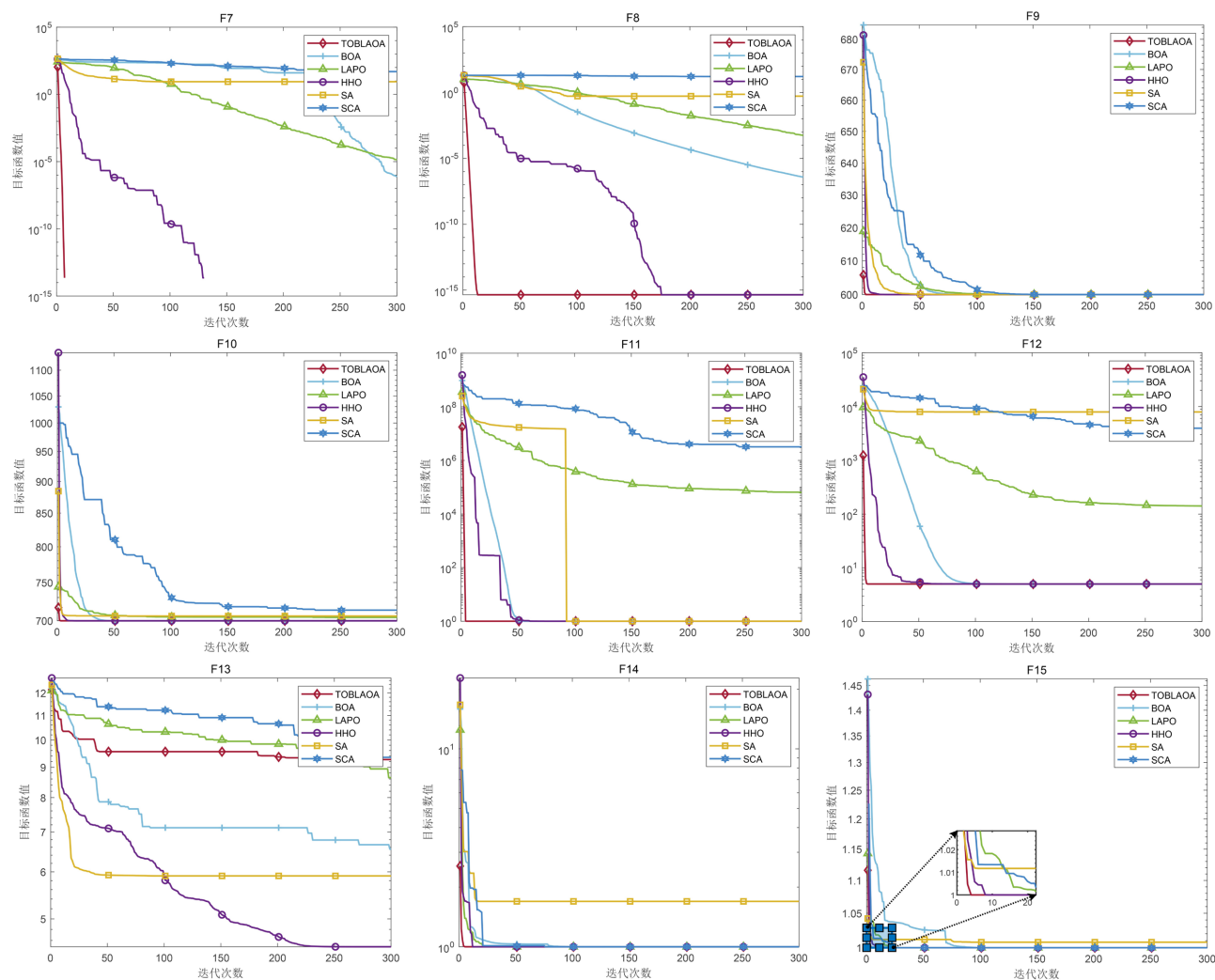


Figure 3. Experimental test function convergence curves of different algorithms for TOBLAOA

图 3. TOBLAOA 不同算法实验测试函数收敛曲线

根据实验结果可知, TOBLAOA 在高维情况下仍与低维表现一致, 其寻优精度并没有因为维度的升高而波动, 这说明该算法具有良好的稳定性, 进一步说明了算法改进的效果。

为了进一步验证本文 TOBLAOA 有效性, 本文给出 TOBLAOA 与 BOA、LAOP、HHO、SA、SCA、NNA 在不同基准测试函数下的收敛曲线图, 见图 3。

从收敛曲线可以明显看出 TOBLAOA 在测试函数仿真实验中, 相较于其他算法收敛性更好, 并且下降曲线并没有因为迭代次数的增多而出现停滞现象, 表明算法能够在更短的迭代周期内寻到最优解, 并且此算法可以弥补传统 AOA 在迭代后期出现收敛速度慢的缺陷, TOBLAOA 对陷入局部最优值问题处理性能更优, 快速跳出局部最优解的能力更强, 实验证明本文改进策略的优化效果良好。

4.5. Wilcoxon 秩和检验

为建立改进算法的统计显著性证据链, 本研究采用非参数统计方法——基于 Wilcoxon 秩和检验[18]的对比实验设计。具体仿真流程为:

1) 对照实验设置: 在 15 个基准测试函数上, 对 TOBLAOA 与 6 种对比算法分别执行 30 次独立重

复试验以消除一定的随机误差。

2) 显著性判据: 以双侧检验的 p 值作为统计差异性指标(显著性阈值 $\alpha = 0.05$), 当 $p < 0.05$ 时拒绝原假设(H_0 : 算法性能无差异), 接受备择假设(H_1 : TOBLAOA 结果具有显著优势)。

实验结果(见表 11)显示: TOBLAOA 与比较的算法大部分都拒绝原假设, 满足 $p < 0.05$ 的条件, 表明改进后的 TOBLAOA 算法的优越性在统计上是显著的。

Table 11. The Wilcoxon rank sum test p-value table for the differences between TOBLAOA and other algorithms

表 11. TOBLAOA 与其他算法差异性的 Wilcoxon 秩和检验 p 值表

	BOA	LAP0	HHO	SA	SCA	SCA
F_1	0.0079	0.0079	0.0079	0.0079	0.0079	0.0079
F_2	0.0079	0.0079	0.0079	0.0079	0.0079	0.0079
F_3	0.0079	0.0079	0.0079	0.0079	0.0079	0.0079
F_4	0.0079	0.0079	0.0079	0.0079	0.0079	0.0079
F_5	0.0079	0.0079	0.0952	0.0079	0.0079	0.0057
F_6	0.0317	0.0079	0.0079	0.0079	0.0317	0.0025
F_7	0.4444	0.0079	1	0.0079	0.0079	0.0079
F_8	0.0079	0.0079	1	0.0079	0.0079	0.0061
F_9	0.0079	0.0159	0.0079	0.3095	0.6905	0.01395
F_{10}	0.0079	0.1508	0.0079	0.0952	0.0317	0.0583
F_{11}	0.0079	0.0079	0.0079	0.0079	0.5476	0.0079
F_{12}	0.0079	0.0079	0.0079	0.0079	0.0079	0.0079
F_{13}	1	1	1	0.0079	1	1
F_{14}	1	0.0079	1	0.1667	1	0.0034
F_{15}	1	0.0079	1	0.0476	1	1

5. TOBLAOA 在实际工程问题中的应用

现将改进后的 TOBLAOA 算法应用与实际工程应用问题中, 并进行仿真实验测试, 将其结果与其他 6 个优化算法进行对比, 由此进一步验证算法可行性及有效性。

5.1. 应用 1: 压力容器设计问题

压力容器设计[19]如图 4 所示, 该问题的优化目标为最小化压力容器的总制造成本, 共包含 4 个决策变量: 圆柱壳厚度 x_1 , 半球形头厚度 x_2 , 内半径 x_3 , 圆柱段长度 x_4 。其中 x_1 和 x_2 为离散变量且取值为 0.0625 英寸的整数倍, x_3 和 x_4 为连续变量。

$$\text{Consider: } X = [x_1, x_2, x_3, x_4] = [T_s, T_h, R, L]; \quad (5.1.1)$$

$$\text{Minimize: } f(X) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3, \quad (5.1.2)$$

$$\begin{aligned} \text{Subject to: } g_1(X) &= -x_1 + 0.0193x_3 \leq 0 \\ g_2(X) &= -x_2 + 0.00954x_3 \leq 0 \\ g_3(X) &= -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \leq 0 \\ g_4(X) &= x_4 - 240 \leq 0 \end{aligned} \quad (5.1.3)$$

其中 $x_1, x_2 \in \{1 \times 0.0625, 2 \times 0.0625, 3 \times 0.0625, \dots, 1600 \times 0.0625\}, 10 \leq x_3, x_4 \leq 200$.

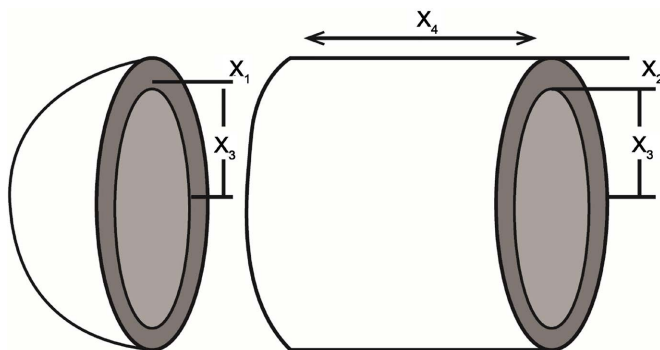


Figure 4. Schematic diagram of pressure vessel problem design
图 4. 压力容器设计问题示意图

5.2. 应用 2：三杆桁架设计问题

三杆桁架是结构工程问题[20]中重要的一种结构类型，该问题的优化目标为最小化桁架结构总质量，共包含 3 个决策变量： x_1, x_2, x_3 分别表示三杆桁架中三个杆的横截面积，其中 x_1 与 x_3 对称。其设计如图 5 所示，每一个桁架上均收到应力 σ 的约束。

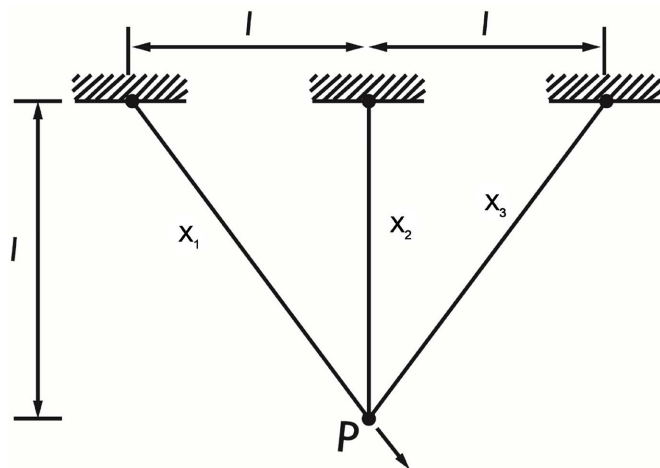


Figure 5. Schematic diagram of the three-bar truss design problem
图 5. 三杆桁架设计问题示意图

$$\text{Consider: } X = [x_1, x_2] = [A_1, A_2]; \quad (5.2.1)$$

$$\text{Minimize: } f(X) = (2\sqrt{2}x_1 + x_2)l; \quad (5.2.2)$$

$$\begin{aligned} \text{Subject to: } g_1(x) &= \frac{\sqrt{2}x_1 + x_2}{\sqrt{2x_1^2 + 2x_1x_2}} P - \sigma \leq 0; \\ g_2(x) &= \frac{x_2}{\sqrt{2x_1^2 + 2x_1x_2}} P - \sigma \leq 0; \\ g_3(x) &= \frac{1}{\sqrt{2x_2 + x_1}} P - \sigma \leq 0; \end{aligned} \quad (5.2.3)$$

Variable range : $0 \leq x_1, x_2 \leq 1;$

(5.2.4)

其中 $l = 100\text{ cm}$, $P = 2\text{ kN/cm}^2$, $\sigma = 2\text{ kN/cm}^2$ 。

5.3. 应用 3：齿轮系设计问题

齿轮系结构设计[19]如图 6 所示，该结构设计的目的是使齿轮系的转动比成本最小，其中传动比的定义为输出轴的角速度与输入轴角速度的比。该问题共有 4 个整数决策变量，分别表示 4 个齿轮的齿数，取值范围为 12 到 60 之间。其数学模型为：

Consider : $X = [x_1, x_2, x_3, x_4]$

(5.3.1)

Minimize : $f(X) = \left(\frac{1}{6.391} - \frac{x_3 x_2}{x_1 x_4} \right)^2$

(5.3.2)

Variable range : $x_1, x_2, x_3, x_4 \in \{12, 13, \dots, 60\}$

(5.3.3)

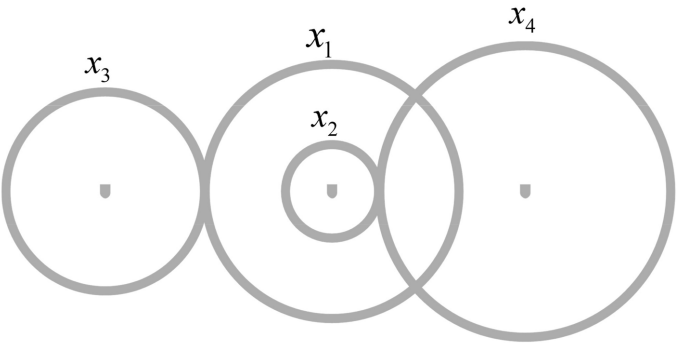


Figure 6. Gear train design problem diagram
图 6. 齿轮系设计问题示意图

5.4. 实验结果分析

本文将优化后的 TOBLAOA 算法与蝴蝶优化算法(BOA)、雷电附着优化算法(LAPO)、哈里斯鹰优化算法(HHO)和遗传算法(SA)、黄金正弦算法(SCA)、神经网络优化算法(NNA)进行测试对比，并将种群数量统一设置为 50，迭代次数为 100，将每种算法在所给出的三个工程应用问题中分别进行 30 次的独立实验。三种应用问题类型维度及约束类型如表 12 所示。

Table 12. Comparison table of three kinds of applied problems
表 12. 三种应用问题比较表

问题名称	变量类型	变量维度	约束类型
压力容器	混合离散	4	非线性不等式
三杆桁架	连续	2	线性与非线性组合
齿轮系	整数	4	无显示约束

应用 1：

实验结果如表 13 所示，其收敛曲线图如图 7 所示，由图表可知，在迭代早期，TOBLAOA 收敛速度虽然不是最快、更接近最优解的算法，但是在经过 50 次迭代之后就快速收敛到了极值，其他算法中

只有 HHO 算法可以在 200 次后追上 TOBLAOA 算法，当迭代次数到达 300 次的时候，TOBLAOA 与其余 6 种优化算法比较结果最优，在迭代速度上更能快速收敛。

Table 13. The comparison of the optimal solutions for pressure vessel design problems
表 13. 压力容器设计问题最优解比较

算法	最优值	x_1	x_2	x_3	x_4
TOBLAOA	5885.30	2.3211	8.9210	57.3581	54.4564
BOA	74480.00	4.8722	8.7144	65.0223	35.6284
LAPO	44063.00	0.7782	0.3847	40.3199	199.9964
HHO	6285.20	1.0778	0.5435	55.2280	63.8594
SA	5922.80	1.1191	0.5532	57.9870	56.9864
SCA	6498.20	1.1087	0.5780	55.7353	85.0521
NNA	5987.10	1.1029	0.5451	57.1379	62.1391

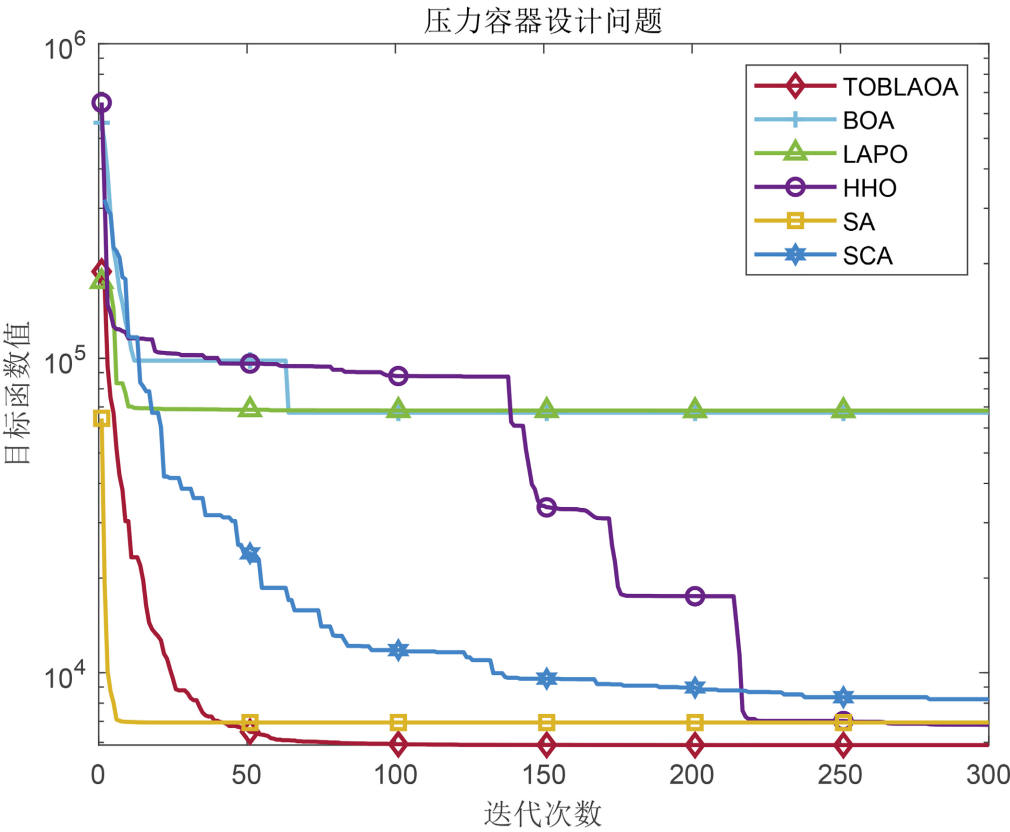


Figure 7. Convergence curves of different algorithms for pressure vessel design problems
图 7. 压力容器设计问题不同算法收敛曲线图

应用 2:

实验结果如表 14 所示，其收敛曲线图 8 如示；由图表可知，在迭代早期，TOBLAOA 收敛速度最快、更接近最优解。在经过 50 次迭代之后就收敛到了极值。TOBLAOA 不仅在寻优结果上优于其余 6 种优化算法，在迭代速度上也展现出了该算法的优异性能。

Table 14. Comparison of the optimal solutions for the design problem of a three-bar truss
表 14. 三杆桁架设计问题最优解比较

算法	最优值	x_1	x_2
TOBLAOA	263.5959	0.7729	0.4085
BOA	265.3015	0.7983	0.4057
LAPO	264.1305	0.7886	0.4686
HHO	263.8976	0.8052	0.3651
SA	264.1446	0.8235	0.3561
SCA	263.9486	0.7916	0.4023
NNA	263.8959	0.7901	0.4047

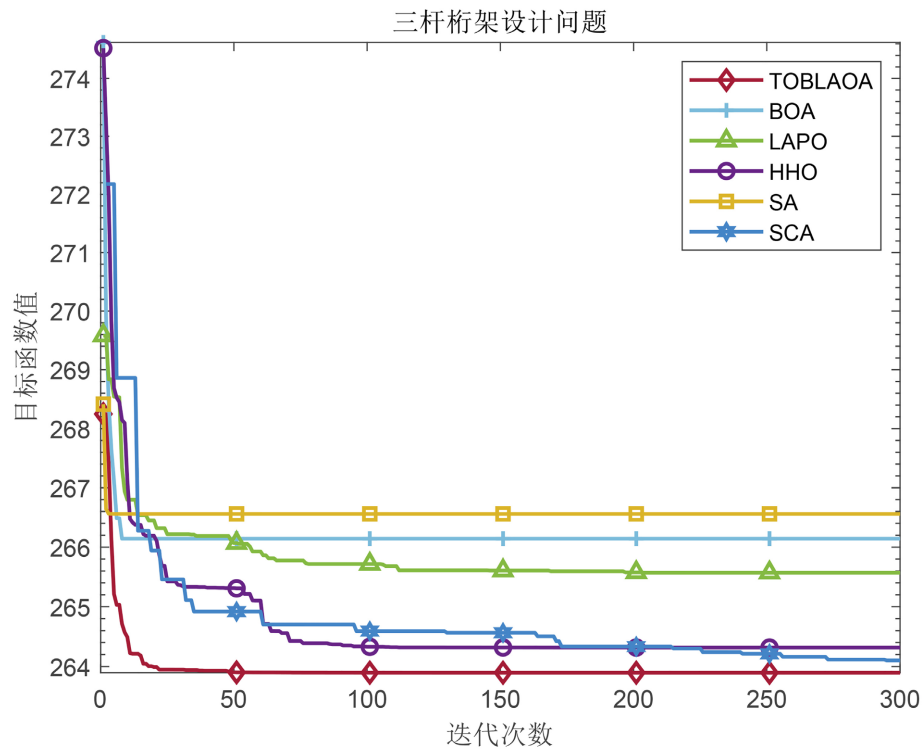


Figure 8. Convergence curves of different algorithms for the design problem of a three-bar truss
图 8. 三杆桁架设计问题不同算法收敛曲线图

Table 15. Comparison of optimal solutions for gear train design problems
表 15. 齿轮系设计问题最优解比较

算法	最优值	x_1	x_2	x_3	x_4
TOBLAOA	2.70E-12	44	19	17	53
BOA	5.52E-08	34	16	15	42
LAPO	9.92E-10	48	25	15	53
HHO	2.31E-11	51	21	16	43
SA	8.70E-09	45	21	16	47
SCA	1.17E-10	52	19	21	49
NNA	2.70E-12	53	17	22	48

应用 3:

实验结果如表 15 所示, 其收敛曲线图 9 如示, 由可知, 在迭代早期, TOBLAOA 收敛速度最快, 收敛曲线近乎垂直, 并在 50 次迭代内就收敛到了极值。TOBLAOA 在寻优结果和迭代速度上都优于其余 6 种优化算法。

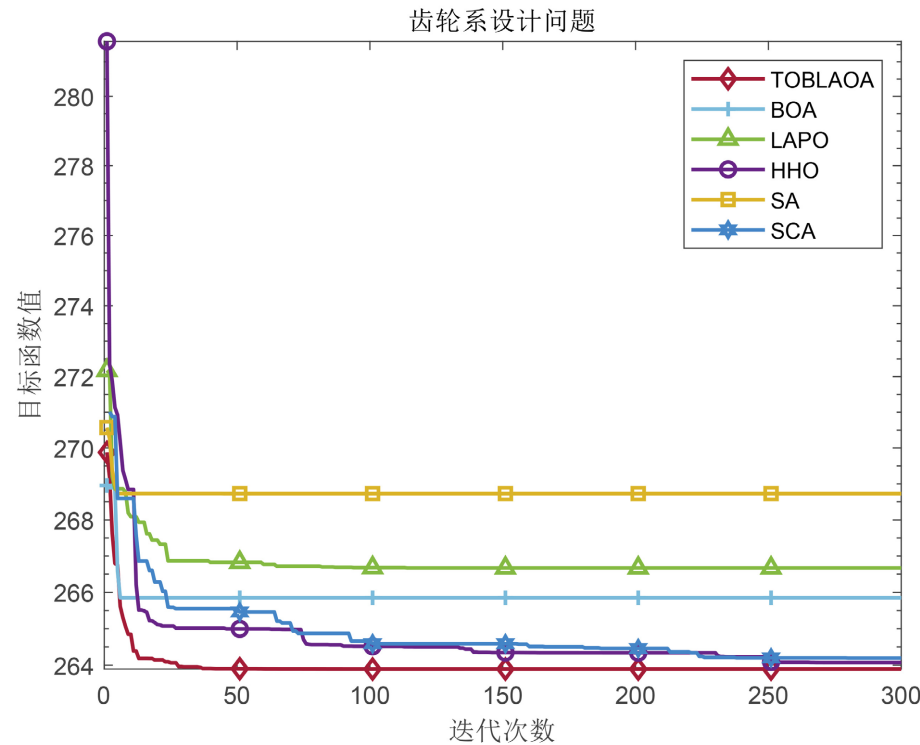


Figure 9. Convergence diagram of gear train design problem
图 9. 齿轮系设计问题收敛曲线图

6. 小结

传统算术优化算法存在收敛速度慢精度不高, 且易陷入局部最优等问题, 故提出了具有 T 分布扰动和透镜成像反向学习策略融合的算术优化算法(TOBLAOA), 该算法性能相较于本章提出的其他几种混合策略的 OBLAOA 更有优势, 并且在与其他优化算法的仿真实验中占据上风地位。TOBLAOA 还可以应用于实际的工程问题当中, 经过仿真实验发现有助于提高工程性能及其实际效率。综上, TOBLAOA 具有很大前景, 可以进一步拓展到更多领域。

基金项目

广西自然科学基金(2022GXNSFAA035618)。

参考文献

- [1] Salomon, R. (1996) Re-Evaluating Genetic Algorithm Performance under Coordinate Rotation of Benchmark Functions. A Survey of Some Theoretical and Practical Aspects of Genetic Algorithms. *Biosystems*, **39**, 263-278. [https://doi.org/10.1016/0303-2647\(96\)01621-8](https://doi.org/10.1016/0303-2647(96)01621-8)
- [2] Kennedy, J. and Eberhart, R. (1995) Particle Swarm Optimization. *Proceedings of ICNN'95—International Conference on Neural Networks*, Perth, 27 November-1 December 1995, 1942-1948. <https://doi.org/10.1109/icnn.1995.488968>

-
- [3] Arora, S. and Singh, S. (2018) Butterfly Optimization Algorithm: A Novel Approach for Global Optimization. *Soft Computing*, **23**, 715-734. <https://doi.org/10.1007/s00500-018-3102-4>
- [4] Habibpour Roudsari, M.N. (2024) Improved Task Scheduling in Heterogeneous Distributed Systems Using Intelligent Greedy Harris Hawk Optimization Algorithm. *Evolutionary Intelligence*, **17**, 4199-4226. <https://doi.org/10.1007/s12065-024-00979-8>
- [5] Mirjalili, S. (2016) SCA: A Sine Cosine Algorithm for Solving Optimization Problems. *Knowledge-Based Systems*, **96**, 120-133. <https://doi.org/10.1016/j.knosys.2015.12.022>
- [6] Mirjalili, S. and Lewis, A. (2016) The Whale Optimization Algorithm. *Advances in Engineering Software*, **95**, 51-67. <https://doi.org/10.1016/j.advengsoft.2016.01.008>
- [7] Abualigah, L., Diabat, A., Mirjalili, S., Abd Elaziz, M. and Gandomi, A.H. (2021) The Arithmetic Optimization Algorithm. *Computer Methods in Applied Mechanics and Engineering*, **376**, Article ID: 113609. <https://doi.org/10.1016/j.cma.2020.113609>
- [8] 郑婷婷, 刘升, 叶旭. 自适应 t 分布与动态边界策略改进的算术优化算法[J]. 计算机应用研究, 2022, 39(5): 1410-1414.
- [9] 黄学雨, 罗华. 融合正余弦策略的算术优化算法[J]. 计算机工程与科学, 2023, 45(7): 1320-1330.
- [10] 兰周新, 何庆. 多策略融合算术优化算法及其工程优化[J]. 计算机应用研究, 2022, 39(3): 758-763.
- [11] 杨文珍, 何庆. 融合微平衡激活的小孔成像算术优化算法[J]. 计算机工程与应用, 2022, 58(13): 85-93.
- [12] 邵彝君, 朱良宽, 付雪, 等. 融合混沌和差分变异的正余弦算术优化算法[J]. 计算机仿真, 2024, 41(11): 393-402.
- [13] Tizhoosh, H.R. (2006) Opposition-Based Reinforcement Learning. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, **10**, 578-585. <https://doi.org/10.20965/jaciii.2006.p0578>
- [14] 杜鹏桢, 唐振民, 孙研. 一种混合蜂群算法的自适应细菌觅食优化算法[J]. 计算机工程, 2014, 40(7): 138-142.
- [15] 韩斐斐, 刘升. 基于自适应 t 分布变异的缎蓝园丁鸟优化算法[J]. 微电子学与计算机, 2018, 35(8): 117-121.
- [16] Nematollahi, A.F., Rahiminejad, A. and Vahidi, B. (2017) A Novel Physical Based Meta-Heuristic Optimization Method Known as Lightning Attachment Procedure Optimization. *Applied Soft Computing*, **59**, 596-621. <https://doi.org/10.1016/j.asoc.2017.06.033>
- [17] Sadollah, A., Sayyaadi, H. and Yadav, A. (2018) A Dynamic Metaheuristic Optimization Model Inspired by Biological Nervous Systems: Neural Network Algorithm. *Applied Soft Computing*, **71**, 747-782. <https://doi.org/10.1016/j.asoc.2018.07.039>
- [18] Derrac, J., García, S., Molina, D. and Herrera, F. (2011) A Practical Tutorial on the Use of Nonparametric Statistical Tests as a Methodology for Comparing Evolutionary and Swarm Intelligence Algorithms. *Swarm and Evolutionary Computation*, **1**, 3-18. <https://doi.org/10.1016/j.swevo.2011.02.002>
- [19] 张宁. 乌鸦搜索算法与侏儒猫鼬优化算法分析及应用研究[D]: [硕士学位论文]. 南宁: 广西民族大学, 2023.
- [20] 汪逸晖, 高亮. 乌鸦搜索算法的改进及其在工程约束优化问题中的应用[J]. 计算机集成制造系统, 2021(7): 1871-1883.