

一种基于引导学习策略改进的Adam梯度下降优化算法

曾钰清, 郭 权, 徐紫玉

赣南科技学院智能制造与材料工程学院, 江西 赣州

收稿日期: 2026年4月23日; 录用日期: 2026年5月21日; 发布日期: 2026年5月28日

摘 要

针对原始Adam梯度下降优化算法在复杂多峰及高维优化问题中易陷入局部最优、全局探索与局部开发难以有效平衡的局限, 文章提出了一种基于引导学习策略(GLS)改进的Adam梯度下降优化算法(GAGDO)。该算法在AGDO的渐进梯度动量积分与动态梯度交互框架基础上, 引入GLS的反馈-引导机制, 通过周期性评估种群分布的标准差, 动态自适应地调节探索与开发行为, 有效克服早熟收敛, 增强跳出局部最优的能力。为验证算法性能, 在CEC2022基准测试集与压力容器设计问题上开展仿真实验。结果表明, GAGDO在收敛速度、寻优精度及鲁棒性方面均优于原始AGDO及GWO、DBO、PSO、SSA等主流算法; 在压力容器设计问题中, GAGDO获得最低制造成本且多次运行稳定性优异, 充分证明了其在复杂非线性约束问题中的可靠性。

关键词

Adam梯度下降优化算法, 引导学习策略, 工程设计问题

An Adam Gradient Descent Optimization Algorithm Improved Based on the Guided Learning Strategy

Yuqing Zeng, Quan Guo, Ziyu Xu

School of Intelligent Manufacturing and Material Engineering, Gannan University of Science and Technology, Ganzhou Jiangxi

Received: April 23, 2026; accepted: May 21, 2026; published: May 28, 2026

Abstract

In response to the limitations of the original Adam gradient descent optimization algorithm, which

文章引用: 曾钰清, 郭权, 徐紫玉. 一种基于引导学习策略改进的 Adam 梯度下降优化算法[J]. 计算机科学与应用, 2026, 16(5): 300-311. DOI: 10.12677/csa.2026.165185

is prone to getting stuck in local optima in complex multi-modal and high-dimensional optimization problems, and where it is difficult to effectively balance global exploration and local exploitation, this paper proposes an improved Adam gradient descent optimization algorithm based on the guided learning strategy (GLS) (GAGDO). This algorithm is built upon the progressive gradient momentum integration and dynamic gradient interaction framework of AGDO, and introduces the feedback-guidance mechanism of GLS. By periodically evaluating the standard deviation of the population distribution, it dynamically and adaptively adjusts the exploration and exploitation behaviors, effectively overcoming premature convergence and enhancing the ability to escape local optima. To verify the performance of the algorithm, simulation experiments were conducted on the CEC2022 benchmark test set and the pressure vessel design problem. The results show that GAGDO outperforms the original AGDO and mainstream algorithms such as GWO, DBO, PSO, and SSA in terms of convergence speed, optimization accuracy, and robustness; in the pressure vessel design problem, GAGDO achieves the lowest manufacturing cost and has excellent stability in multiple runs, fully demonstrating its reliability in complex nonlinear constrained problems.

Keywords

Adam Gradient Descent Optimization Algorithm, Guiding Learning Strategy, Engineering Design Problem

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

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

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



Open Access

1. 引言

在当今科学研究与工程实践中，全局优化问题无处不在。无论是复杂的工程结构设计、工业生产中的资源分配与调度，还是人工智能领域的机器学习模型参数调优，其本质往往都可以归结为在给定的约束条件下寻找某一目标函数的最优解[1]。然而，随着现代实际应用场景的日益复杂化，这类优化问题通常表现出高度非线性、多峰值以及高维数等极具挑战性的特征。传统的基于梯度的确定性优化方法在处理这些复杂问题时，往往容易陷入局部最优，且对初始解的依赖性较强，难以高效地探索庞大而复杂的解空间。

为克服传统确定性优化方法的局限性，元启发式算法(Meta-Heuristic Algorithms)作为一种强大的近似全局优化工具应运而生，并在近年来获得了学术界与工业界的广泛关注。这类算法通常受到自然界生物行为[2]、物理现象[3]或人类社会演化过程[4]的启发，无需借助目标函数的梯度信息，而是凭借其内在的随机搜索机制和启发式规则，在解空间的全局探索与局部开发之间寻求动态平衡。元启发式算法因其出色的鲁棒性、易于实现以及广泛的适用性，已成为解决当代复杂科学与工程优化问题的核心利器[5] [6]。

近期，受深度学习中 Adam 优化器的启发，Adam 梯度下降优化算法(Adam Gradient Descent Optimizer, AGDO)作为一种基于数学规律的创新元启发式算法被提出[7]。该算法通过融合渐进式梯度动量积分、动态梯度交互系统以及系统优化算子，在全局探索与局部开发之间取得了出色的平衡。然而，尽管 AGDO 表现优异，但在复杂任务中仍存局限：它在复杂多峰地形中易陷入局部最优，在低维问题中效能略有衰减；且因缺乏高效变异机制，面对高维复杂问题时探索与开发易失衡，导致早熟收敛。

针对上述不足，本文提出一种基于引导学习策略(Guided Learning Strategy, GLS) [8]改进的 Adam 梯度下降优化算法(GAGDO)，该算法在 AGDO 原有框架(渐进梯度动量积分、动态梯度交互系统、系统优化算子)的基础上，引入 GLS 的反馈 - 引导机制，通过周期性评估种群离散度动态调整探索与开发行为，

从而在保持 AGDO 快速收敛特性的同时, 增强跳出局部最优的能力。

2. AGDO 算法的工作原理

AGDO 算法核心工作原理旨在突破传统进化范式的局限, 通过将数学梯度特性与随机搜索过程相融合, 在解空间的全局探索与局部开发之间实现高效平衡。AGDO 的运行框架主要由四个部分构成: 初始化、渐进梯度动量积分、动态梯度交互系统以及系统优化算子。

2.1. 初始化

在算法运行之初, AGDO 需要在边界约束内构建初始种群。具体而言, 系统会随机生成一个种群大小为 N 的初始种群。每个个体均由 Dim 维决策变量构成。此过程的具体数学定义如下:

$$x_{ij} = rand(N, Dim) \times (UB_j - LB_j) + LB_j \quad (1)$$

式中, x_{ij} 表示第 i 个个体在第 j 个维度上的坐标位置; $rand(N, Dim)$ 代表维度为 $N \times Dim$ 的随机分布矩阵; UB_j 与 LB_j 分别对应第 j 维空间的上限与下限。在算法初始化阶段, AGDO 会评估并选出具有最佳适应度的个体作为最优解候选(记为 $X_{superior}$), 以此来近似引导后续梯度下降的寻优方向。

2.2. 渐进梯度动量积分

该阶段主要主导算法前期的全局探索。AGDO 通过引入随迭代次数动态调整的非线性惯性权重 w 和随机角度变量 α_i , 构建了渐进式的动量积分框架:

$$X_{i, new_1}^{t+1} = \begin{cases} w * X_i^t + \alpha_i * X_i^t & k=1 \\ X_{i, new_1}^t + \sin(2 * \pi * dim * t) * X_{i, new_1}^t & 1 < k \leq 2/dim \end{cases} \quad (2)$$

其中, X_i^t 是搜索代理在第 t 次迭代时的位置, t 表示连续梯度动量整合的次数。动态参数 w 和随机角度变量 α_i 的计算公式分别为:

$$w = rand() * \left(\frac{1}{t_{total}^2} t_{now}^2 - \frac{2}{t_{total}} t_{now} + \frac{1}{2} \right) \quad (3)$$

$$\alpha_i = \cos(1 - rand() * 2\pi) \quad (4)$$

这种动态的积分方式使得搜索代理在迭代初期具备较大的步长和高度的随机性, 能够快速且广泛地遍历整个解空间。

2.3. 动态梯度交互系统

该阶段是 AGDO 局部开发的核心, 其灵感直接来源于 Adam 算法的一阶与二阶动量估计机制。为精确引导 AGDO 的搜索过程, 动态梯度交互系统引入了方向指针(ξ)。系统计算随机位置与当前位置的适应度差值, 并将其离散化为 1 或 -1。

在预开发阶段, 系统旨在通过广泛遍历解空间来锁定潜在的优质解。为此, 算法采用了一种复合型位置更新策略:

$$X_{i, new_{2a}}^{t+1} = X_{i, new_1}^{t+1} + \xi * a * (G - X_{idx_1}^t) - a * (X_{i, new_1}^{t+1} - X_{idx_2}^t) \quad (5)$$

$$a = \left(1 - \frac{1}{T} t \right) * rand(1, Dim) \quad (6)$$

进入后开发阶段, 随着迭代推进, 目标转向解的微调, 以确保收敛的稳定性和解的质量。此阶段采

用更加直接高效的更新公式，利用索引位置直接向最优解 G 的方向进行调整。具体公式如下：

$$X_{i,new_{2b}}^{t+1} = X_{idx_1}^t + a * (G - X_{idx_2}^t) \quad (7)$$

关键参考点 G 是通过矩估计方法确定的最优位置，用于引导和调整搜索过程：

$$G = X_{i,best}^t - \frac{\eta}{\sqrt{\hat{v}_t + \vartheta}} \hat{m}_t \quad (8)$$

式中， \hat{m}_t 和 \hat{v}_t 分别是梯度的一阶矩和二阶矩的偏置校正估计：

$$\begin{cases} \hat{v}_t = \frac{v_t}{1 - \beta_2^t} \\ \hat{m}_t = \frac{m_t}{1 - \beta_1^t} \end{cases} \quad (9)$$

此外，系统引入了严格的信赖域策略，控制优化步伐，防止每次迭代中的位置更新幅度过大而导致不稳定或无效搜索。

2.4. 系统优化算子

为防范动态梯度交互系统在开发后期可能出现的搜索停滞，AGDO 引入了系统优化算子。该机制以 Levy 飞行为核心，扩大算法的全局搜索范围，增强系统的寻优性能。其操作公式定义为：

$$X_{i,new_4}^{t+1} = X_{suprior} + \text{levy}(\varphi) * \delta * (X_{suprior} - X_i^t * \theta) \quad (10)$$

该更新操作在一个随迭代次数增加而概率性增强的条件下执行：

$$\text{if } \text{rand} \leq \left[1 + e^{-\left(\frac{18^t}{T} - 12\right)} \right]^{-1} \quad (11)$$

其中， $X_{suprior}$ 是当前迭代中的最优位置。 $\text{levy}(\varphi)$ 是 levy 飞行函数，其计算方式为：

$$\text{levy}(\varphi) = 0.01 \times \frac{r_1 \times \tau}{|r_2|^{1/\sigma}} \quad (12)$$

通过引入 Levy 飞行的长步长特性，该算子弥补了利用阶段探索不足的缺陷，在保证精确收敛的同时，帮助算法跳出局部最优，实现对解空间的更全面探索。

3. GLS 策略的基本原理

该策略的设计灵感首先源自建构主义学习理论，模拟了教学过程中教师为掌握知识的学习者提供适时反馈与引导，以纠正其搜索偏差的机制。同时，它深入借鉴了艾宾浩斯遗忘曲线中的间隔训练原理，通过设定合理的周期性评估与反馈间隔，将短期的寻优轨迹有效转化为长期的全局搜索。GLS 主要由两个阶段构成：反馈阶段与引导阶段。

策略的执行流程由两个关键参数控制：

- 1) C_{\max} ：该参数的取值大小直接影响引导的频率和时长： C_{\max} 越大，引导次数越少，单次引导时间越长； C_{\max} 越小，引导次数越多，单次引导时间越短。
- 2) α ：引导选择阈值，用于决定当前应执行探索还是开发引导。

3.1. 反馈阶段

在反馈阶段算法依据反馈指标 V_t 实施动态引导。具体而言，通过计算历史种群(即学习经验)的标准

差，可以量化个体的空间离散度，进而诊断算法当前的搜索状态：离散度较高表明算法正处于全局探索阶段；反之，若个体分布高度集中，则意味着算法已进入局部开发状态。

$$V_o = std(St) * B \tag{13}$$

$$B = 200 / (ub - lb) \tag{14}$$

式中， $std(St)$ 为历史个体位置的标准差， St 存储最近 C_{max} 迭代中所有个体的位置信息， ub 和 lb 分别为问题的上、下界。系数 B 用于消除不同问题边界范围对反馈结果的尺度影响，使策略具有良好的通用性。

3.2. 引导阶段

获得反馈结果 V_o 后，GLS 将其与预设参数 α 进行比较，选择相应的引导方式：

当 $V_o > \alpha$ 时，表明种群分布较分散，算法当前偏向探索，因此需要加强开发。此时，GLS 采用公式 (15)，围绕历史最优个体 X_{best} 生成新位置，引导种群向最优区域收敛：

$$X_{new} = X_{best} + \tan(R * \pi) * (ub - lb) / V_o \tag{15}$$

其中， R 为 $[0,1]$ 之间的随机数。

当 $V_o \leq \alpha$ 时，表明种群分布较集中，算法当前偏向开发，因此需要加强探索。此时，GLS 采用公式 (16)，在全局范围内随机生成新位置，以增强种群多样性，避免陷入局部最优：

$$X_{new} = R * (ub - lb) \tag{16}$$

通过这种反馈 - 引导机制 GLS 实现了对探索与开发过程的精准平衡，且由于其结构极其简单，带来的计算复杂度微乎其微，它容易被嵌入并混合到各类现有的元启发式或群智能优化算法中，从而显著提升算法在处理复杂约束工程问题时的收敛精度与整体寻优性能。反馈 - 引导机制流程图如图 1 所示：

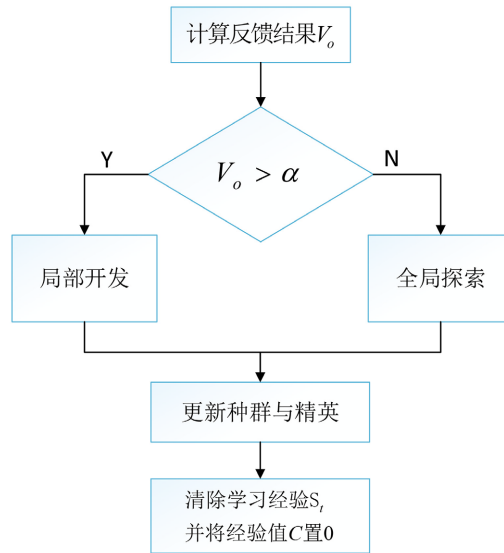


Figure 1. Feedback-guidance mechanism flowchart
图 1. 反馈 - 引导机制流程图

4. GAGDO 算法的实现

该算法在 AGDO 原有的三阶段优化框架(渐进梯度动量整合、动态梯度交互系统、系统优化算子)基础上，引入 GLS 的反馈 - 引导机制，通过周期性评估种群离散程度，动态调整算法的探索与开发行为，

从而在保持 AGDO 快速收敛特性的同时，增强跳出局部最优的能力。

基于引导学习策略改进的 Adam 梯度下降优化算法(GAGDO)的算法流程归纳如下：

- 1) 初始化阶段：设置 Adam 参数(β_1, β_2, lr)与 GLS 引导阈值，随机生成初始种群并将其存入历史存储矩阵。
- 2) 计算动态权重：在每次迭代开始时，更新时变权重 ω 、缩放因子 a 和 Levy 扰动系数 ξ ，以调节搜索步长。
- 3) 渐进梯度动量集成：结合周期性算子 α 对当前位置进行初步偏移预测，利用数学周期性规律增强算法跳出局部最优的能力。
- 4) 动态梯度交互系统：根据邻域个体的适应度差异确定梯度方向，并应用类 Adam 优化器的动量与方差修正机制更新候选解。
- 5) 引导学习策略(GLS)介入：定期计算存储矩阵中各维度的标准差，若某维度离散度过高，则引导个体向全局最优靠拢。
- 6) 系统优化算子(Levy 飞行)：根据 Sigmoid 概率函数决定是否触发 Levy 飞行，利用随机的长短步长对全局最优位置进行精细化微调。
- 7) 贪婪选择与存储更新：通过目标函数评估新解，仅保留适应度更优的个体，并实时更新 GLS 存储矩阵中的位置信息。
- 8) 迭代终止：重复执行搜索与引导过程，直至达到最大迭代次数，最终输出全局最优解及收敛曲线。

5. 实验分析

5.1. 标准测试函数集测试实验

为全面评估改进算法的寻优性能，本文选用 CEC2022 基准测试集[9]开展仿真实验。该集合涵盖单峰(F1)、基础(F2~F5)、混合(F6~F8)及组合(F9~F12)共四类 12 个不同特征的函数(见表 1)，旨在通过模拟多样的复杂优化环境，客观评价算法在实际工程应用中的鲁棒性与综合求解精度。

Table 1. CEC 2022 function set

表 1. CEC2022 函数集

| 函数 | 名称 | 范围 | 最优值 |
|-----|--|-------------|------|
| F1 | Shifted and Full Rotated Zakharov Function | [-100, 100] | 300 |
| F2 | Shifted and Full Rotated Rosenbrock's Function | [-100, 100] | 400 |
| F3 | Shifted and Full Rotated Expanded Schaffer's F6 Function | [-100, 100] | 600 |
| F4 | Shifted and Full Rotated Non-Continuous Rastrigin's Function | [-100, 100] | 800 |
| F5 | Shifted and Full Rotated Levy Function | [-100, 100] | 900 |
| F6 | Hybrid Function 1 (N = 3) | [-100, 100] | 1800 |
| F7 | Hybrid Function 2 (N = 6) | [-100, 100] | 2000 |
| F8 | Hybrid Function 3 (N = 5) | [-100, 100] | 2200 |
| F9 | Composition Function 1 (N = 5) | [-100, 100] | 2300 |
| F10 | Composition Function 2 (N = 4) | [-100, 100] | 2400 |
| F11 | Composition Function 3 (N = 5) | [-100, 100] | 2600 |
| F12 | Composition Function 4 (N = 6) | [-100, 100] | 2700 |

5.2. 对比算法与参数设置

为了客观评估所提 GAGDO 算法的综合寻优性能, 将其与原始 AGDO 以及其他四种经典的群智能优化算法展开对比实验。选取的对照算法包括: 灰狼优化算法(Grey Wolf Optimizer, GWO) [10]、蜣螂优化算法(Dung Beetle Optimizer, DBO) [11]、粒子群优化算法(Particle Swarm Optimization, PSO) [12]和麻雀搜索算法(Sparrow Search Algorithm, SSA) [13]。为保证性能评估的公平性与严谨性, 所有参评算法的通用参数均保持一致: 种群规模 $N = 30$, 最大迭代次数为 500 次, 搜索空间维度设定为 20。此外, 考虑到元启发式算法固有的随机性, 本实验规定各算法在每个基准测试函数上均独立运行 30 次, 以确保实验结果的统计可靠性。

5.3. 收敛性分析

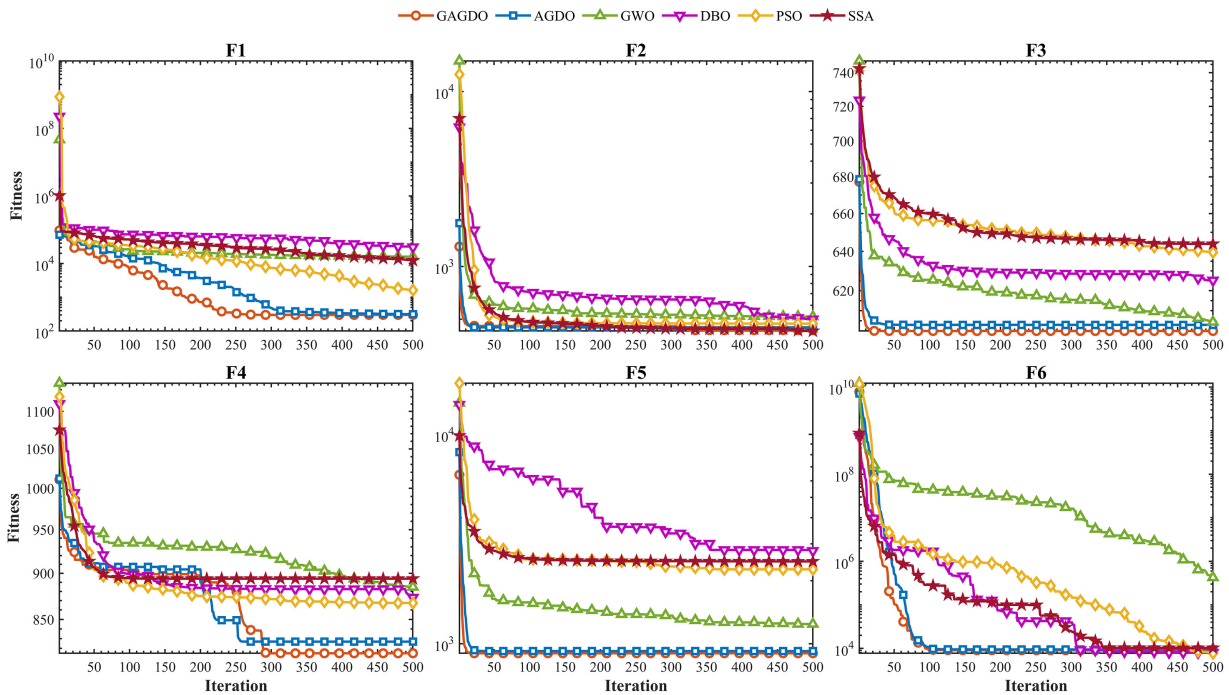
在 12 个基准测试函数下, 6 种算法的平均收敛曲线对比如图 2 所示, 图示结果表明, GAGDO 在所有类别函数上均表现出更优的综合性能:

1) 在单峰函数 F1 上, GAGDO 在迭代初期就呈现最为陡峭的下降趋势, 收敛速度显著优于其他五种算法, 且最终达到最低适应度值, 曲线平滑无振荡, 展现了优异的局部开发能力。

2) 在基础函数 F2~F5 上, GAGDO 同样表现突出, 尤其在 F3 和 F5 中, 迭代初期下降幅度远超过其他算法, 后期平稳收敛至最低值; 在混合函数 F6~F8 上, GAGDO 的优势进一步扩大, 表现出极强的全局探索能力和逃逸局部最优的能力, AGDO 的最终适应度值始终高于 GAGDO, 其他算法则在整个迭代过程中均未达到理想精度且后期振荡严重。

3) 在组合函数 F9~F12 上, GAGDO 凭借其卓越的全局探索与局部开发平衡能力, 在收敛速度、最终精度和稳定性方面均显著优于其他对比算法。

上述结果充分验证了 GAGDO 算法能够有效平衡全局勘探与局部开发, 在不同类型的测试函数上均展现出了更快的收敛速度、更高的寻优精度以及更强的鲁棒性。



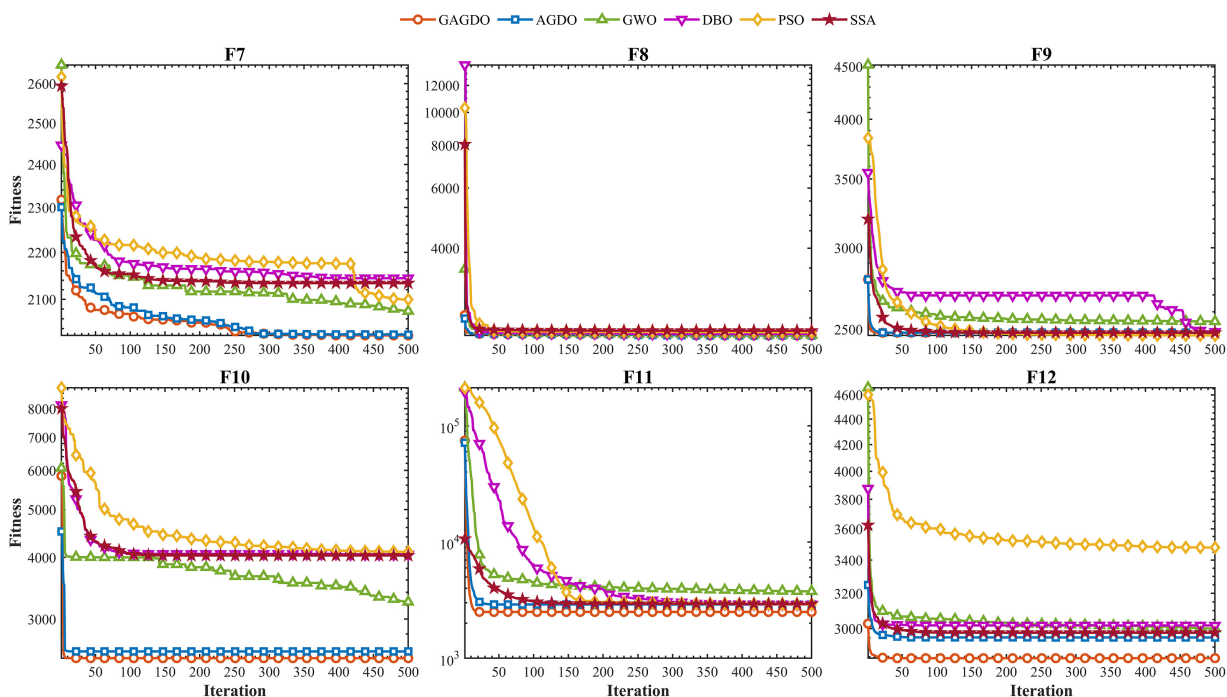


Figure 2. Convergence curve comparison chart
图 2. 收敛曲线对比图

5.4. 综合性能对比分析

为直观呈现各算法的综合性能，本文通过雷达图进行展示。考虑到所用指标均为极小化指标且存在量纲差异，为符合“面积越大性能越优”的视觉逻辑，本文引入反向最大-最小归一化对数据进行线性变换，将其映射至[0.1, 1.0]区间：

$$S_{ij} = \alpha + (1 - \alpha) \times \frac{X_{\max} - X_{ij}}{X_{\max} - X_{\min}} \quad (17)$$

其中， X_{\max} 和 X_{\min} 为相应指标下的极值， $\alpha = 0.1$ 用于规避最差值映射至原点导致的图形重叠。变换后，各维度的最优性能均对应雷达图的最外廓（即 1.0 处）。由此，雷达图中覆盖面积越大、形状越趋于正多边形的算法，代表其在该函数上的综合表现与稳定性越优。

基于反向最大-最小归一化处理，图 3 直观地展示了各算法在 12 个基准测试函数上 30 次独立运行统计结果（最优值 Best、最差值 Worst、均值 Mean、中值 Median 及标准差 Std）的雷达图。

从整体分布来看，红色多边形在绝大多数测试函数维度上几乎完全贴合雷达图的最外层边界，尤其在单峰函数 F1、基础函数 F2~F5 以及混合函数 F6~F8 上，各指标点均稳定落于 1.0 附近，表明 GAGDO 不仅在最优解质量上达到领先水平，而且在多次独立运行中保持了极低的方差与波动。在组合函数 F9~F12 这类高难度测试上，GAGDO 的多边形依然饱满无内缩，进一步验证了其在复杂多峰地形中跳出局部最优的强劲能力。与部分对比算法呈现出的极不规则、尖锐甚至局部凹陷的形态相比，GAGDO 的多边形形态最为饱满均衡，整体更接近正五边形，这充分证明了 GAGDO 不仅在全局求解精度上占据显著优势，更在算法稳定性和抗随机性干扰方面展现出了卓越的能力。

综上所述，GAGDO 凭借引导学习策略的有效嵌入，实现了在多类型、多难度测试函数上的一致性优异表现，其多维度的综合性能显著优于其他参与对比的优化算法。

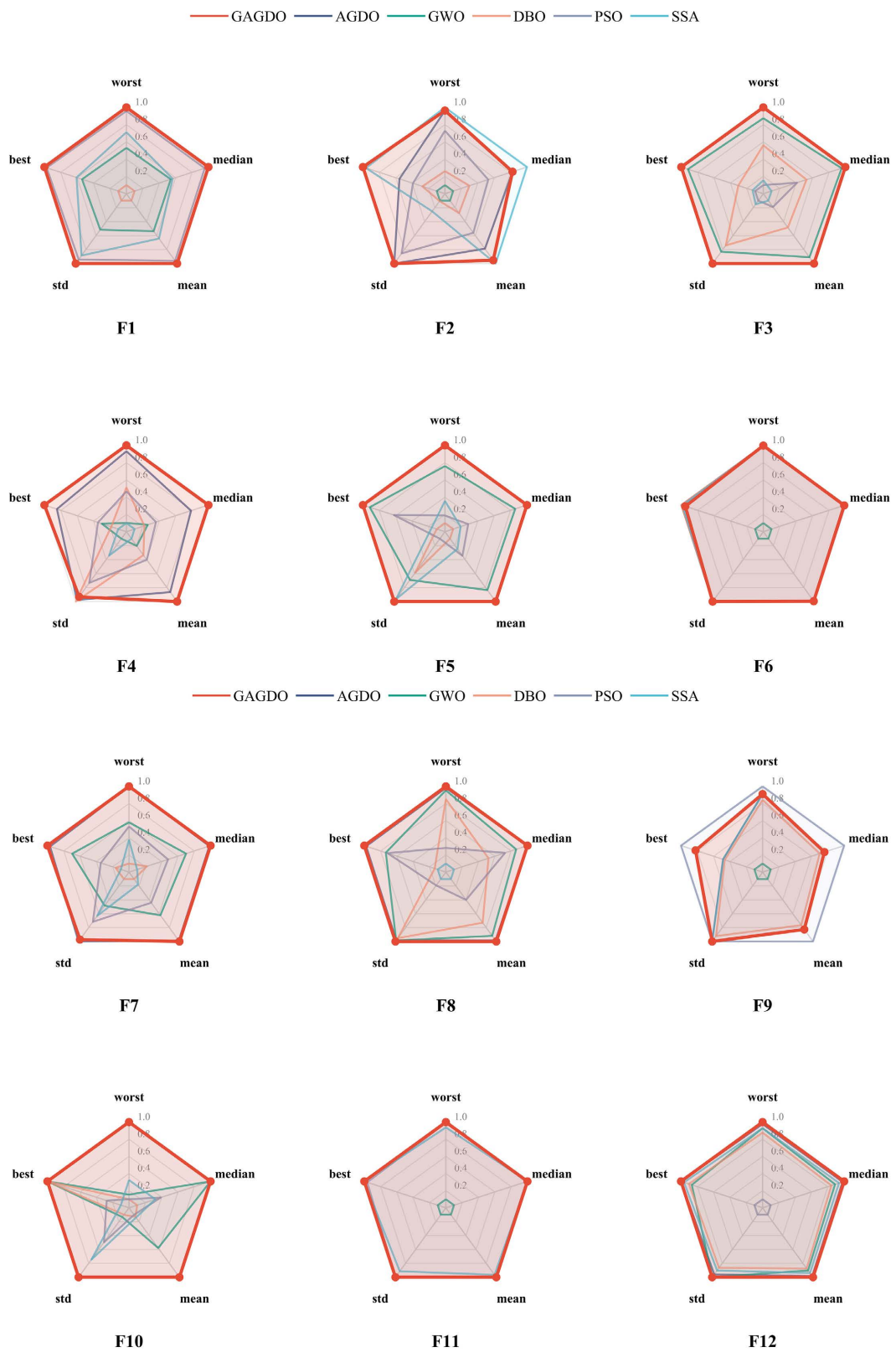


Figure 3. Test function indicator comparison radar chart
图 3. 测试函数指标对比雷达图

6. 压力容器设计问题的求解分析

本文将 GAGDO 算法应用于压力容器设计问题[14], 通过引入 DBO、PSO、GWO、SSA 及基准算法 AGDO 作为对比对象, 从多维度验证其在复杂搜索空间中的优化性能。

压力容器设计问题是一个经典的非线性受限工程优化案例, 其核心目标是在满足结构功能与安全规范的前提下, 通过优化关键设计参数来最小化总制造成本。该问题的决策变量由四个连续或离散变量组成: 壳体厚度(T_s)、封头厚度(T_h)、内半径(R)以及不含封头的圆柱截面长度(L)。压力容器设计结构图如图 4 所示, 制造成本的函数构建综合考虑了材料费用、焊接成本以及封头成型的加工开支。在优化过程中, 算法必须严格遵循一系列复杂的非线性约束, 包括材料力学强度要求、最小容积限制以及制造工艺规定的尺寸比例:

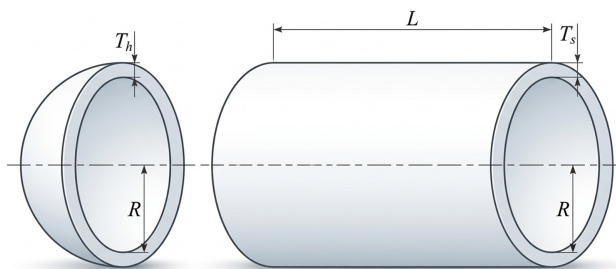


Figure 4. Pressure vessel design structure diagram
图 4. 压力容器设计结构图

四个变量壳体厚度(T_s)、封头厚度(T_h)、内半径(R)以及不含封头的圆柱截面长度(L), 记为 $\bar{x} = [x_1, x_2, x_3, x_4] = [T_s, T_h, R, L]$ 。目标函数为制造成本最小化:

$$f(\bar{x}) = 0.6224x_1x_2x_3x_4 + 1.778x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \quad (18)$$

约束条件如下:

$$\begin{aligned} g_1(\bar{x}) &= -x_1 + 0.0193x_3 \leq 0, \\ g_2(\bar{x}) &= -x_3 + 0.00954x_3 \leq 0, \\ g_3(\bar{x}) &= -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \leq 0, \\ g_4(\bar{x}) &= x_4 - 240 \leq 0. \end{aligned} \quad (19)$$

各变量的取值范围为: $0 \leq x_1, x_2 \leq 99$, $10 \leq x_3, x_4 \leq 200$ 。

Table 2. Comparison of test results on pressure vessel design issues
表 2. 压力容器设计问题测试结果对比

| | DBO | PSO | GWO | SSA | AGDO | GAGDO |
|-------|-------------|--------------|-------------|-------------|-------------|--------------------|
| 平均值 | 6562.783631 | 6216.961965 | 6079.318675 | 6412.455184 | 5949.417435 | 5910.536414 |
| 标准差 | 595.2190905 | 133.2800771 | 374.9518016 | 491.6418634 | 120.1482202 | 51.23545532 |
| 最优值 | 5885.356958 | 6023.922881 | 5892.021671 | 5886.79528 | 5885.376428 | 5885.332773 |
| x_1 | 0.778172571 | 0.852009756 | 0.778403154 | 0.779023212 | 0.778174152 | 0.778168641 |
| x_2 | 0.38465107 | 0.421147854 | 0.385762049 | 0.385071607 | 0.384654042 | 0.384649162 |
| x_3 | 40.31967753 | 44.14547695 | 40.33040682 | 40.36389658 | 40.31985363 | 40.31961872 |
| x_4 | 200 | 199.99681011 | 200 | 200 | 199.384537 | 152.8210106 |

表 2 展现了 GAGDO 算法显著的综合优势：首先，在寻优精度方面，GAGDO 成功寻得了所有参评算法中的最低制造成本，其平均成本亦远优于 DBO、GWO 及 SSA 等传统算法，证明了其强大的收敛能力。其次，在算法稳定性上，GAGDO 的标准差仅为 51.23545532，较基础 AGDO 算法有大幅降低，显著提升了在处理复杂工程约束时的稳健性。最后，从决策变量的分布来看，当其他对比算法普遍陷入局部最优区域时，GAGDO 凭借优化的搜索策略成功挖掘出更优解空间，充分验证了其卓越的全局探索能力与规避早熟收敛的有效性。表明了 GAGDO 能够为该受限工程设计提供更具经济性与可靠性的优化方案。

7. 结论

针对原始 AGDO 算法在处理复杂优化问题时易陷入局部最优、高维探索与开发易失衡等局限性，本文提出了一种基于引导学习策略(GLS)改进的 Adam 梯度下降优化算法(GAGDO)。主要研究可归纳为以下几点：

1) GAGDO 算法在 AGDO 原有的渐进梯度动量整合、动态梯度交互等框架基础上引入了 GLS 策略。该机制通过周期性计算种群的空间离散程度，能够精准诊断算法所处的搜索状态，从而动态调节全局探索与局部开发行为，有效克服了算法早熟收敛的缺陷。

2) 在涵盖单峰、基础、混合及组合特征的 12 个 CEC2022 基准测试函数中，GAGDO 算法展现出了卓越的性能。与原始 AGDO 以及 GWO、DBO、PSO、SSA 经典群智能算法相比，GAGDO 在收敛速度、最终寻优精度以及逃逸局部最优的能力上均实现了显著超越。雷达图分析进一步证实，GAGDO 具有最饱满均衡的指标形态，展现出极佳的算法稳定性与抗随机干扰能力。

3) 在高度非线性且约束复杂的压力容器设计工程问题中，GAGDO 算法展现了极高的经济实用价值与鲁棒性。实验结果表明，GAGDO 不仅成功跳出了局部最优陷阱，寻得了最低制造成本，更以极低的标准差证明了其在面对实际受限工程问题时具备卓越的可靠性。

综上所述，GAGDO 通过引入引导学习策略，在全局探索与局部开发之间实现了有效平衡，显著提升了原 AGDO 算法的综合性能，为求解复杂科学与工程优化问题提供了一种高效可靠的解决方案。后续研究将进一步探索 GLS 策略在其他元启发式算法中的适用性，并将其拓展至更多实际工程应用场景。

基金项目

江西省教育厅科学技术研究项目：基于多智能体的智慧路灯微电网群协同优化调度研究(GJJ2203620)。

参考文献

- [1] Dhiman, G. (2021) SSC: A Hybrid Nature-Inspired Meta-Heuristic Optimization Algorithm for Engineering Applications. *Knowledge-Based Systems*, **222**, Article ID: 106926. <https://doi.org/10.1016/j.knosys.2021.106926>
- [2] Dehghani, M., Montazeri, Z., Trojovská, E. and Trojovský, P. (2023) Coati Optimization Algorithm: A New Bio-Inspired Metaheuristic Algorithm for Solving Optimization Problems. *Knowledge-Based Systems*, **259**, Article ID: 110011. <https://doi.org/10.1016/j.knosys.2022.110011>
- [3] Rashedi, E., Nezamabadi-Pour, H. and Saryazdi, S. (2009) GSA: A Gravitational Search Algorithm. *Information Sciences*, **179**, 2232-2248. <https://doi.org/10.1016/j.ins.2009.03.004>
- [4] Feng, Z., Niu, W. and Liu, S. (2021) Cooperation Search Algorithm: A Novel Metaheuristic Evolutionary Intelligence Algorithm for Numerical Optimization and Engineering Optimization Problems. *Applied Soft Computing*, **98**, Article ID: 106734. <https://doi.org/10.1016/j.asoc.2020.106734>
- [5] Dokeroglu, T., Sevinc, E., Kucukyilmaz, T. and Cosar, A. (2019) A Survey on New Generation Metaheuristic Algorithms. *Computers & Industrial Engineering*, **137**, Article ID: 106040. <https://doi.org/10.1016/j.cie.2019.106040>
- [6] 张梦婷, 等. 元启发式算法研究综述[J]. 计算机工程与应用, 2026, 62(2): 40-53.
- [7] Xia, Y. and Ji, Y. (2025) Application of a Novel Metaheuristic Algorithm Inspired by Adam Gradient Descent in Distributed Permutation Flow Shop Scheduling Problem and Continuous Engineering Problems. *Scientific Reports*, **15**,

Article No. 21692. <https://doi.org/10.1038/s41598-025-01678-9>

- [8] Jia, H. and Lu, C. (2024) Guided Learning Strategy: A Novel Update Mechanism for Metaheuristic Algorithms Design and Improvement. *Knowledge-Based Systems*, **286**, Article ID: 111402. <https://doi.org/10.1016/j.knosys.2024.111402>
- [9] 田云娜, 李奕轩, 王凯欣. 混合策略改进的鱼鹰优化算法及其工程应用[J]. 计算机工程与应用, 2025, 61(18): 114-131.
- [10] Mirjalili, S., Mirjalili, S.M. and Lewis, A. (2014) Grey Wolf Optimizer. *Advances in Engineering Software*, **69**, 46-61. <https://doi.org/10.1016/j.advengsoft.2013.12.007>
- [11] Xue, J. and Shen, B. (2022) Dung Beetle Optimizer: A New Meta-Heuristic Algorithm for Global Optimization. *The Journal of Supercomputing*, **79**, 7305-7336. <https://doi.org/10.1007/s11227-022-04959-6>
- [12] 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>
- [13] Xue, J. and Shen, B. (2020) A Novel Swarm Intelligence Optimization Approach: Sparrow Search Algorithm. *Systems Science & Control Engineering*, **8**, 22-34. <https://doi.org/10.1080/21642583.2019.1708830>
- [14] Zhao, S., Zhang, T., Ma, S. and Chen, M. (2022) Dandelion Optimizer: A Nature-Inspired Metaheuristic Algorithm for Engineering Applications. *Engineering Applications of Artificial Intelligence*, **114**, Article ID: 105075. <https://doi.org/10.1016/j.engappai.2022.105075>