



## 融合经验反思机制的教与学优化算法

吴迪, 贾鹤鸣, 刘庆鑫, 齐琦, 王爽

引用本文:

吴迪,贾鹤鸣,刘庆鑫,齐琦,王爽. 融合经验反思机制的教与学优化算法[J]. 智能系统学报, 2023, 18(3): 629–641.

WU Di, JIA Heming, LIU Qingxin, QI Qi, WANG Shuang. Teaching and learning optimization algorithm based on empirical reflection mechanism[J]. *CAAI Transactions on Intelligent Systems*, 2023, 18(3): 629–641.

在线阅读 View online: <https://dx.doi.org/10.11992/tis.202112043>

## 您可能感兴趣的其他文章

### 融合动态反向学习的阿奎拉鹰与哈里斯鹰混合优化算法

Hybrid Aquila and Harris hawks optimization algorithm with dynamic opposition-based learning

智能系统学报. 2023, 18(1): 104–116 <https://dx.doi.org/10.11992/tis.202108031>

### 融合振幅随机补偿与步长演变机制的改进原子搜索优化算法

Improved atom search optimization by combining amplitude random compensation and step size evolution mechanism

智能系统学报. 2022, 17(3): 602–616 <https://dx.doi.org/10.11992/tis.202103033>

### 基于混合身份搜索黏菌优化的模糊C-均值聚类算法

An optimization fuzzy C-means clustering algorithm based on the hybrid identity search and slime mold algorithms

智能系统学报. 2022, 17(5): 999–1011 <https://dx.doi.org/10.11992/tis.202107011>

### 分组教学蚁群算法改进及其在机器人路径规划中应用

Improvement of ant colony algorithm in group teaching and its application in robot path planning

智能系统学报. 2022, 17(4): 764–771 <https://dx.doi.org/10.11992/tis.202108020>

### 随机交叉自学策略改进的教与学优化算法

Teaching-learning-based optimization algorithm based on random crossover-self-learning strategy

智能系统学报. 2021, 16(2): 313–322 <https://dx.doi.org/10.11992/tis.201910045>

### 具有Levy变异和精英自适应竞争机制的蚁狮优化算法

Ant lion optimizer with levy variation and adaptive elite competition mechanism

智能系统学报. 2018, 13(2): 236–242 <https://dx.doi.org/10.11992/tis.201706091>

DOI: 10.11992/tis.202112043

网络出版地址: <https://kns.cnki.net/kcms/detail/23.1538.tp.20230320.1345.006.html>

## 融合经验反思机制的教与学优化算法

吴迪<sup>1</sup>, 贾鹤鸣<sup>2</sup>, 刘庆鑫<sup>3</sup>, 齐琦<sup>3</sup>, 王爽<sup>2</sup>

(1. 三明学院 教育与音乐学院, 福建 三明 365004; 2. 三明学院 信息工程学院, 福建 三明 365004; 3. 海南大学 计算机科学与技术学院, 海南海口 570228)

**摘要:** 针对传统教与学算法存在易陷入局部最优、收敛速度慢和求解精度低等问题, 提出一种融合经验反思机制的教与学优化算法 (empirical reflection teaching learning based optimization, ERTLBO)。首先在教师阶段引入经验反思机制, 遴选精英个体引导普通个体向教师靠近, 提高班级整体水平, 从而提高算法全局探索能力。其次在学习阶段引入动态自适应权重, 能够根据学生的适应度值对位置进行自适应扰动, 进而实现个体位置的动态更新, 提高算法跳出局部最优的能力。仿真实验选取 23 个基准测试函数对 ERTLBO 同其他变体和流行算法进行性能测试。实验结果表明, ERTLBO 算法具有更好的寻优性能和求解稳定性。最后, 通过 2 个工程设计问题进一步验证 ERTLBO 解决实际问题的有效性和优越性。

**关键词:** 教与学优化算法; 经验反思机制; 动态自适应权重; 元启发式算法; 基准函数; 压力容器设计问题; 焊接梁设计问题; Wilcoxon 秩和检验

中图分类号: TP301.6 文献标志码: A 文章编号: 1673-4785(2023)03-0629-13

中文引用格式: 吴迪, 贾鹤鸣, 刘庆鑫, 等. 融合经验反思机制的教与学优化算法 [J]. 智能系统学报, 2023, 18(3): 629-641.

英文引用格式: WU Di, JIA Heming, LIU Qingxin, et al. Teaching and learning optimization algorithm based on empirical reflection mechanism[J]. CAAI transactions on intelligent systems, 2023, 18(3): 629-641.

## Teaching and learning optimization algorithm based on empirical reflection mechanism

WU Di<sup>1</sup>, JIA Heming<sup>2</sup>, LIU Qingxin<sup>3</sup>, QI Qi<sup>3</sup>, WANG Shuang<sup>2</sup>

(1. School of Education and Music, Sanming University, Sanming 365004, China; 2. School of Information Engineering, Sanming University, Sanming 365004, China; 3. School of Computer Science and Technology, Hainan University, Haikou 570228, China)

**Abstract:** In this paper, an empirical reflection teaching learning based optimization (ERTLBO) algorithm is proposed to solve the problems of easy falling into local optimum, slow convergence speed and low accuracy in traditional teaching-learning-based optimization. Firstly, the empirical reflection mechanism is introduced in the teaching stage to select elite individuals, guide ordinary individuals to approach teachers, and improve the overall level of the class, so as to improve the overall exploration ability of the algorithm. Secondly, the dynamic adaptive weight is introduced in the learning stage, which can adaptively disturb the position according to the fitness value of students, so as to realize dynamic updating of individual position and improve the ability of the algorithm to jump out of local optimum. In the experiment, 23 benchmark functions are selected to test the performance of ERTLBO, variant, and popular algorithms. Experimental results show that ERTLBO has better optimization performance and solution stability. Finally, the effectiveness and superiority of ERTLBO in solving practical problems are further verified through two engineering design problems.

**Keywords:** teaching and learning-based optimization; empirical reflection mechanism; dynamic adaptive weight; meta-heuristic algorithm; benchmark functions; pressure vessel design problem; welded beam design problem; Wilcoxon rank-sum test

收稿日期: 2021-12-30. 网络出版日期: 2023-03-24.

基金项目: 全国教育科学规划教育部重点课题 (DIA220374).

通信作者: 贾鹤鸣. E-mail: [jiaheminglucky99@126.com](mailto:jiaheminglucky99@126.com).

近年来不同学科和工程领域中的优化问题日益复杂与多样化, 诸如牛顿法、梯度下降法和共

轭梯度法等传统优化方法已经不能满足实际需求。因此,许多研究者受自然启发,通过模仿生物或物理现象提出了各种元启发式优化算法。与传统优化算法不同,元启发式优化算法概念简单,不需要梯度信息,灵活且易于实现,可广泛应用于不同领域下的各类优化问题<sup>[1-3]</sup>。其中,教与学优化算法 (teaching-learning-based optimization, TLBO)<sup>[4]</sup> 是印度学者 Rao 等于 2011 年提出的一种以班级内教师教学和学生学习的思想为设计思想的群体智能优化算法,该算法将优化求解空间模拟为一个班级,而班级里的每位学生就对应于解空间中的搜索个体,每个学生的成绩对应于解的适应度值,教师则代表当前整个群体中适应度值最好的解。该算法思想清晰,来源于基本的教育理论和教育方法,原理简单且参数较少,具有较快的收敛速度,已被成功应用于多个学术或工业领域,如:天线优化设计<sup>[5]</sup>、电力系统优化调度<sup>[6]</sup>、结构可靠性分析<sup>[7]</sup>、手势识别技术<sup>[8]</sup>等。然而,无免费午餐理论<sup>[9]</sup>指出没有一种算法能独立解决所有的优化问题,即每种优化算法都具有其自身的优势和局限性,只能在某些领域中表现出较好的效果。因此,不断有学者提出各种新颖或改进的优化算法试图解决不同的优化问题。

李丽荣等<sup>[10]</sup>提出了一种基于头脑风暴策略的改进教与学优化算法,在学习阶段选择优秀的学生个体与当前个体进行头脑风暴式学习,提高算法开发能力。欧阳城添等<sup>[11]</sup>提出一种改进天牛须搜索的教与学优化算法,在教学阶段引入天牛须搜索,提高解的精度,学习阶段利用混合变异,提高算法跳出局部最优能力。李丽荣等<sup>[12]</sup>为克服算法易出现早熟收敛和搜索精度差的问题,在教学部分引入自适应因子,提高教师自身学习能力,很好的保持了种群的多样性。Lakshmi 等<sup>[13]</sup>提出一种鲸鱼与教与学混合优化算法,在 23 个基准测试函数表现出一定求解能力,并将其应用于面部情绪识别领域。Xu 等<sup>[14]</sup>提出基于动态反向学习的教与学优化算法,利用非对称搜索空间显著提高了种群获得全局最优解的概率,整体上提高了种群的开发能力。Tsai 等<sup>[15]</sup>在教学和学习阶段引入不同的突变策略来提高算法探索与开发能力。Shukla 等<sup>[16]</sup>引入混沌映射与动态自适应权重提高种群多样性与收敛能力,并应用于目标分类中。Taheri 等<sup>[17]</sup>提出一种平衡教与学优化算法,引入辅导阶段和重启阶段平衡算法探索和开发能力。

上述研究方法对 TLBO 的收敛速度与收敛精度均有改善,但上述诸多研究思想都是从数学公式的改进或者融入一些生物机制角度进行算法效果的提升,往往与原算法作者提出的教育学理论不一致,忽视了元启发式算法的设计初衷。本文的主旨在于吸收学习教育学的基本理论和思想,从教育学原理角度对教师的教和学生的学进行改进,利用当下主流的教育思想来设计算法的改进公式,提出一种融合经验反思机制的教与学优化算法。在教学阶段引入经验反思策略,鼓励精英个体引导普通个体向最优位置靠近,使得不同类型个体之间具有更强的联系,从而提升班级整体水平。学习阶段引入动态自适应权重,根据个体适应度值进行位置扰动,实现动态位置更新,提升算法避免早熟收敛能力,提高收敛精度。通过 23 个测试函数、Wilcoxon 秩和检验、焊接梁设计问题以及压力容器设计问题进行测试实验,实验结果表明改进算法具有更优越的寻优性能和鲁棒性,解决实际工程问题时效果显著。

## 1 教与学优化算法

教与学优化算法主要灵感来源于课堂上教师教学行为与学生的学习行为。TLBO 将种群集合称为班级,种群中个体称为学生,当前适应度值最佳的个体称为教师。该算法具体过程分为教师教学阶段与学生学习阶段。

### 1.1 教学阶段

在教学阶段,教师向学生传授知识,以提高班级的整体水平。考虑到教学成果受到班级平均成绩影响,因此位置更新方式表达为

$$\mathbf{X}_i^{\text{new}}(t+1) = \mathbf{X}_i(t) + \text{rand} \times (\mathbf{X}_{\text{teacher}}(t) - T_F \times \mathbf{X}_{\text{mean}}(t)) \quad (1)$$

式中:  $\mathbf{X}_i^{\text{new}}(t+1)$  与  $\mathbf{X}_i(t)$  分别学习后的个体与学习前的个体; rand 表示均匀分布在 0~1 的随机数;  $\mathbf{X}_{\text{teacher}}(t)$  代表种群内最佳个体,称为教师;  $T_F$  称为教学因子,随机确定为 1 或 2;  $\mathbf{X}_{\text{mean}}(t)$  是种群平均位置。

### 1.2 学习阶段

在学习阶段,学生之间相互交流,来提高自己的成绩。具体来说,学生  $\mathbf{X}_{r1}(t)$  随机选择  $\mathbf{X}_{r2}(t)$  进行交流学习,比较学习成绩,选出学习优秀的学生引导另一位学习,其数学表达式为

$$\mathbf{X}_i^{\text{new}}(t+1) = \begin{cases} \mathbf{X}_i^{\text{old}}(t) + \text{rand} \times (\mathbf{X}_{r1}(t) - \mathbf{X}_{r2}(t)), & f(\mathbf{X}_{r1}(t)) > f(\mathbf{X}_{r2}(t)) \\ \mathbf{X}_i^{\text{old}}(t) + \text{rand} \times (\mathbf{X}_{r2}(t) - \mathbf{X}_{r1}(t)), & \text{其他} \end{cases} \quad (2)$$

其中,  $f(\cdot)$  表示需要优化的目标函数;  $\mathbf{X}_{r1}(t)$  与



$X_{r2}(t)$  代表 2 个随机个体。TLBO 算法伪代码如下所示。

- 1) 设置种群规模  $N$ , 当前迭代次数  $t$ , 最大迭代次数  $T$
- 2) 初始化种群位置
- 3) While  $t < T$
- 4) 计算个体适应度值并选取最佳个体为教师
- 5) For  $i = 1$  to  $N$
- 6) 教学阶段: 通过式 (1) 更新种群位置
- 7) 学习阶段: 通过式 (2) 更新种群位置
- 8) End For
- 9)  $t = t + 1$
- 10) End While

## 2 融合经验反思机制的教与学优化算法

### 2.1 经验反思教学模式

教与学算法在教学阶段强调学生从教师的授课中获得知识。依据教学相长理论, 教师需要不断提升自身素养才能促进学生不断学习。原算法中的策略是教师依据学生的平均成绩进行教学, 然而这种教学模式没有获得学生反馈, 无法根据学生所学情况进行教学模式的调整。同时, 该模式也没有考虑更多学生个体不同的个性情况, 容易以点盖面片面地对学生进行教授。这种情况是由于当下缺少一种教师对学生学习情况经验反思的机制导致的。现有的机制并不能够保障将获得的成功的教学经验或者失败的经验吸收转化, 忽视了持续改进的作用, 从而无法进一步提升教学质量(水平)。因此本文改进思想正是在教学阶段加入一个基于经验反思的机制, 通过反思教师可以对学生的学习行为进行及时监控, 根据学生学习情况对自身进行有效调整, 从而可以保证教学效果的持续改进, 因此融入反思机制的教与学算法, 相比于其他算法, 具有循环增强和持续改进的特点。

在标准教与学算法的教学阶段中, 教师向学生传授知识, 希望整体平均位置向教师靠近, 从而提高全班的平均成绩。然而, 该模式下可能会导致算法探索能力不强, 影响实际寻优效果。针对这一问题, 提出一种经验反思策略, 根据不同学生的学习情况调整教学方式, 提高整体教学质量, 具体描述如下:

$$X_i^{\text{new}}(t+1) = X_i^{\text{old}}(t) + e^{-\frac{10t}{T}} \times (X_{\text{teacher}}(t) - X_{r3}(t)) + \text{rand} \times (X_{\text{bp}}(t) - X_{\text{np}}(t)) \quad (3)$$

式中:  $T$  为最大迭代次数;  $X_{r3}(t)$  代表随机个体位置;  $X_{\text{bp}}(t)$  与  $X_{\text{np}}(t)$  分别代表成绩排名前 30% 与后 70% 的随机个体 (即适应度排名前 30% 和后 70% 的个体)。

由上述可知, 在完成教学阶段后, 教师会对学生成绩 (适应度值评估) 进行观测并收集反馈意见, 记录成绩没有提升的学生个体数量。若学生数量没有超过该班级学生总数量的 60% (反思阈值  $R$ ), 即大部分个体适应度值均不理想。此时教师对自己的教学行为进行反思, 依据不同层次的学生来调整教学模式, 对剩下的个体进行经验反思式教学, 选拔优秀适应度值个体辅导普通个体, 使整体种群位置朝最优点靠近, 进而提高整体的学习质量。

### 2.2 动态自适应学习模式

在学习阶段中, 学生之间互相学习, 与其他同学交流知识, 从而提高对知识的理解程度。TLBO 随机选择 2 个学生, 比较其适应度值, 将优秀学生的位置减去次优学生, 以提高次优学生知识水平。然而, 仅通过随机选取的 2 个体无法有效提高学生成绩, 容易在局部最优值附近停止, 导致算法收敛速度慢。合适的选择动态权值能够加速算法寻优效果, 提高算法收敛速度<sup>[18]</sup>。因此, 提出一种动态自适应学习模式, 提升算法避免早熟收敛能力, 数学表达式如下:

$$w_i(t) = \begin{cases} k \times \sin(\text{rand} \times \pi) \times \left( \frac{f_i(t) - f_{\min}(t)}{f_{\text{avg}}(t) - f_{\min}(t)} \right), & f_i(t) \leq f_{\text{avg}}(t) \\ k, & f_i(t) > f_{\text{avg}}(t) \end{cases} \quad (4)$$

式中:  $k$  为常量;  $f_i(t)$  为第  $i$  个个体所获适应度值;  $f_{\min}(t)$  为全局最优适应度值 (以求极小值为例);  $f_{\text{avg}}(t)$  为种群平均适应度值。

根据式 (4) 可知, 若当前个体适应度值小于种群平均适应度, 惯性权值将较小, 对算法的扰动减小, 允许当前个体充分开发邻近区域, 提升局部寻优能力。反之, 若当前个体适应度值大于种群平均适应度时, 借助扰动因子  $k$  可有效扩大搜索范围, 明显增强其全局探索能力。图 1 代表某个体基于式 (4) 所得自适应权值曲线图, 图中自适应权值呈振荡下降趋势。结合式 (4) 可以看出, 个体适应度值时常小于种群平均适应度值, 因此产生一个较小的权值对位置进行扰动, 可令种群充分开发邻近区域, 提升求解精度。为了帮助个体选择最佳搜索模式, 本文经过大量实验分析, 将扰动因子  $k$  设置为 0.5 最为合适。

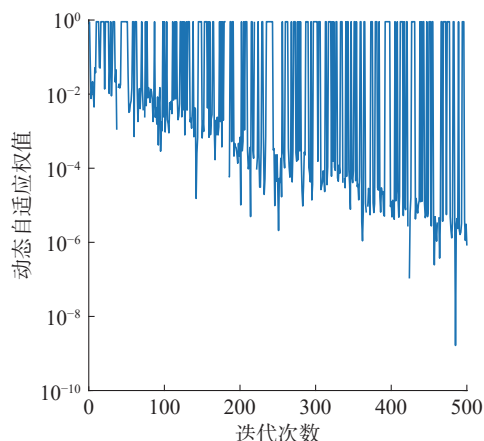


图 1 动态自适应权重曲线  
Fig. 1 Dynamic adaptive weight curve

基于上述分析, 本文将动态自适应权重与学习阶段相结合, 提出了改进的动态自适应学习模式, 改进更新公式为

$$X_i^{\text{new}}(t+1) = \begin{cases} w_i(t) \times X_i(t) + \text{rand} \times (X_{r1}(t) - X_{r2}(t)), & f(X_{r1}(t)) > f(X_{r2}(t)) \\ w_i(t) \times X_i(t) + \text{rand} \times (X_{r2}(t) - X_{r1}(t)), & \text{其他} \end{cases} \quad (5)$$

综上两节, 将经验反思机制融合到教学模式中, 允许教师在教学环境中参考不同层次学生的知识掌握情况, 进而调整教学模式。与此同时, 鼓励学习成绩不好的学生 (适应度值排名位于后 40% 个体) 向成绩好的学生学习 (适应度值排名位于前 60% 个体), 进而提升班级整体成绩, 提升算法的全局探索能力。此外, 式 (3) 中的  $e^{-\frac{10t}{T}}$  是从 1 到 0 线性递减的参数, 迭代前期主要依靠老师完成教学任务, 但随着迭代次数的增加, 一些成绩好的学生能够与老师互相配合, 提升整体适应度值水平。学习阶段引入动态自适应系数, 根据个体适应度值进行位置扰动, 帮助个体调整最佳搜索模式, 从而增加个体找到最优解的几率, 同时有效避免算法易陷入局部最优, 提升算法收敛精度。上述改进策略分别增强了算法的全局探索与局部开发能力, 利用经验反思提升了算法整体探索能力, 迭代前期主要由教师完成教学活动, 当班级整体学习效果不佳时, 教师利用经验反思策略引导学生学习, 提升班级整体水平。利用动态自适应学习系数令个体进行适当扰动, 提升了局部开发能力, 避免算法陷入局部极值, 使得算法在全局和局部的能力上达到有效平衡。

### 2.3 算法实现

将经验反思机制与动态自适应权重加入到教与学优化算法中, 改进传统教与学优化算法的寻优能力。ERTLBO 算法伪代码如下, 执行流程如图 2 所示。

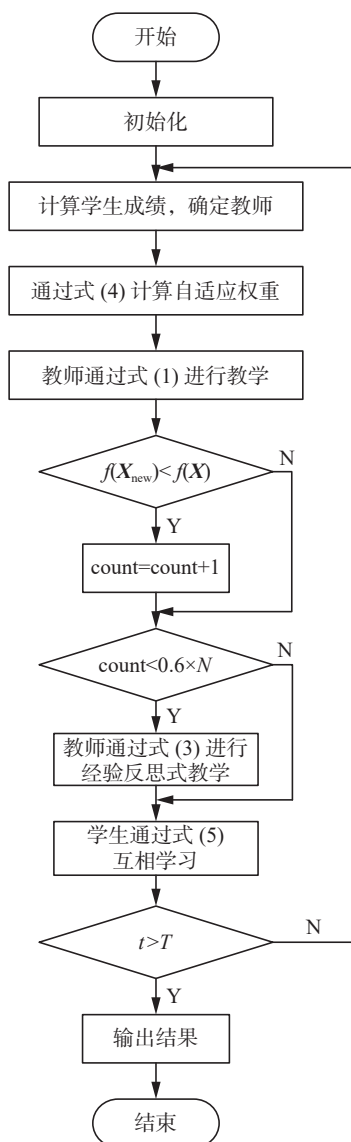


图 2 ERTLBO 算法实现流程图  
Fig. 2 Flowchart of ERTLBO algorithm

ERTLBO 算法如下:

- 1) 初始化种群规模  $N$ , 当前迭代  $t$ , 最大迭代次数  $T$ , 反思阈值  $\text{count}$
- 2) 初始化种群位置
- 3) While  $t < T$
- 4) 计算个体适应度值并选取最佳个体为教师
- 5) 根据式 (4) 计算权重因子  $w$
- 6) For  $i = 1$  to  $N$
- 7) 通过式 (1) 计算种群位置
- 8) If  $f(X_{\text{new}}) < f(X)$
- 9)  $\text{count} = \text{count} + 1$
- 10) End If
- 11) If  $\text{count} < 0.6 \times N$
- 12) 通过经验反思机制 (式 (3)) 更新种群位置
- 13) End If
- 14) 通过式 (4) 计算种群位置
- 15) End For

16)  $t = t+1$

17) End While

## 2.4 运算复杂度分析

由1、2节内容可知, TLBO 算法运算复杂度主要取决于以下过程: 初始化阶段、适应度值评估以及种群位置更新阶段。定义种群规模  $N$ 、问题维度  $D$ 、最大迭代次数  $T$ 。则初始化阶段复杂度为  $O(N \times D)$ , 适应度值评估部分复杂度为  $O(N \times T)$ 。种群位置更新分为教师教学阶段及学生学习阶段, 运算复杂度为  $O(N \times D \times T)$ 。因此 TLBO 算法运算复杂度为  $O(N \times D \times T)$ 。同理, ERTLBO 算法运算复杂度取决于初始化阶段、适应度值评估以及种群位置更新阶段。初始化阶段复杂度为  $O(N \times D)$ , 适应度值评估部分复杂度为  $O(N \times T)$ 。种群位置更新分为教学阶段、经验反思教学模式以及学生相互学习阶段, 因此该部分运算复杂度为  $O(N \times D \times T)$ 。综上所述, ERTLBO 运算复杂度为  $O(N \times D \times T)$ , 可知改进算法与原始算法运算复杂度相同, 并未对算法产生负面影响。

## 3 实验仿真与分析

### 3.1 仿真环境

本次实验环境为 64 位 Windows 10 系统, 处理器型号 Intel(R) Core(TM) i5-9500 CPU, 主频为 3.00 GHz, 内存 24 GB, 使用基于 R2018b 版 MATLAB 实现。为验证本文提出改进算法的有效性, 采用 23 个标准测试函数对 ERTLBO 进行性能测试<sup>[19]</sup>。标准测试函数包括 7 个单峰函数, 6 个多峰测试函数与 10 个固定维多峰测试函数。其中函数 F1~F7 只有一个全局最优解, 验证算法的局部开发能力与收敛速度。函数 F8~F23 具有多个局部最优值与一个全局最优值, 可验证算法跳出局部最优与全局搜索能力。因此, 通过不同类型的测试函数, 可充分验证算法的寻优性能。

### 3.2 对比算法与参数设置

为了更好地验证 ERTLBO 算法性能, 本文采用两组优化算法进行对比。其中一组为 TLBO 改进算法, 如 BASTLBO、ITLBOBSO 与 DSLTLBO。另一组为目前主流优化算法如 SSA<sup>[20]</sup>、WOA<sup>[21]</sup>、GWO<sup>[22]</sup> 以及 GA<sup>[23]</sup>。这些算法被证实具有良好的寻优能力。为确保实验结果有效性, 统一设定种群规模  $N=30$ , 维度  $D=100$ , 最大迭代次数  $T=500$ , 各算法独立运算 100 次进行统计分析。算法参数设置如表 1 所示。选取平均值、标准差与 Wilcoxon 秩和检验作为评价指标。在解决极小值问题中, 平均值可以反映算法收敛精度与寻优能力, 标准差可以反映算法搜索结果鲁棒性, Wil-

coxon 秩和检验可反映两种算法的显著性差异。

表 1 各算法参数设置  
Table 1 Parameter setting of algorithms

算法	参数设置
ERTLBO	$k=0.5; R=0.6$
ITLBOBSO <sup>[10]</sup>	$\alpha=[0.5, 1]; k=[0.3, 0.6]$
BASTLBO <sup>[11]</sup>	$\delta=1; d=2$
DSLTLBO <sup>[12]</sup>	$k=3$
TLBO <sup>[4]</sup>	$T_F=[1, 2] \in N$
SSA <sup>[20]</sup>	$c_1 \in [0, 1]; c_2 \in [0, 1]$
WOA <sup>[21]</sup>	$a_1=[2, 0]; a_2=[-2, 1]; b=1$
GWO <sup>[22]</sup>	$a=[2, 0]$
GA <sup>[23]</sup>	$P_c=0.7; P_m=0.3$

### 3.3 收敛精度分析

实验结果如表 2 和表 3 所示, 其中 Mean 代表平均值, Std 为标准差, 最优结果用粗体标出。从表 2 数据可以看出, ERTLBO 在求解单峰测试函数 F1~F4 和 F7 时均达到了全局最优值, 收敛精度分别达到了  $10^{-199}$ 、 $10^{-106}$ 、 $10^{-76}$ 、 $10^{-88}$  和  $10^{-5}$  量级, 明显强于其他改进 TLBO 和原始算法。从标准差上看 ERTLBO 的求解稳定性也强于其他算法, 说明动态自适应学习模式对局部寻优效果有显著增强。对于 F5 与 F6 函数, ERTLBO 并未收敛至全局最优值, 但平均值和标准差数据仅次于 ITLBOBSO 和 BASTLBO。从单峰测试函数结果可以看出, 动态自适应学习模式可根据适应度值对个体位置进行动态自适应调整, 提高算法避免早熟收敛能力, 增强算法求解精度。在多峰测试函数 F8~F13 中, ERTLBO 仍能取得不错的效果。对于函数 F9~F11, ERTLBO 收敛精度达到理论最优值且标准差最小。但在求解 F9 和 F11 测试函数时, 有些对比算法同样取得了理论最优值, 此时结合图 3 的收敛曲线可以看出 ERTLBO 更快收敛速度, 说明 ERTLBO 的搜索性能更佳。对于函数 F10, 仅有 ERTLBO 收敛到理论最优值。对于函数 F12~F13, ERTLBO 效果不及 BASTLBO 和 ITLBOBSO。在固定维多峰测试函数中, 对于函数 F14、F16、F17~F21, ERTLBO 取到理论最优值, 仅有个别函数测试时的稳定性不如对比算法, 体现出 ERTLBO 算法在全局探索上的稳定性。对于函数 F15, 虽未取到理论最优值, 但收敛到了更接近的全局最优值, 并且稳定性高。对于函数 F22 和 F23, 虽未达到理论最优, 但相比基本算法 TLBO, ERTLBO 还是有进步的趋势。综上所述, 经验反思机制与动态自适应权重策略能够有效提升 TLBO 算法求解精度, 赋予算法不错的稳定性, 搜索效率得到提升。

表 2 ERTLBO 及其对比算法在单峰测试函数实验结果

Table 2 Experimental results of ERTLBO and compared algorithms in unimodal benchmark functions

函数	统计值	ERTLBO	BASTLBO	ITLBOBSO	DSLTLBO	TLBO	SSA	WOA	GWO	GA
F1	Mean	<b><math>4.86 \times 10^{-199}</math></b>	$1.93 \times 10^{-17}$	$1.35 \times 10^{-100}$	$2.29 \times 10^{-78}$	$3.52 \times 10^{-71}$	$1.38 \times 10^3$	$2.34 \times 10^{-71}$	$1.57 \times 10^{-12}$	$3.45 \times 10^4$
	Std	<b>0.00</b>	$5.66 \times 10^{-18}$	$4.54 \times 10^{-100}$	$4.87 \times 10^{-78}$	$4.69 \times 10^{-71}$	$3.70 \times 10^2$	$1.21 \times 10^{-70}$	$1.33 \times 10^{-12}$	$7.87 \times 10^3$
F2	Mean	<b><math>8.93 \times 10^{-106}</math></b>	$5.23 \times 10^{-8}$	$9.10 \times 10^{-50}$	$4.88 \times 10^{-40}$	$2.81 \times 10^{-36}$	$4.73 \times 10^1$	$5.15 \times 10^{-49}$	$4.24 \times 10^{-8}$	$1.60 \times 10^2$
	Std	<b><math>2.13 \times 10^{-105}</math></b>	$5.57 \times 10^{-9}$	$2.56 \times 10^{-49}$	$2.76 \times 10^{-40}$	$1.89 \times 10^{-36}$	7.79	$2.88 \times 10^{-48}$	$1.55 \times 10^{-8}$	$6.94 \times 10^1$
F3	Mean	<b><math>9.54 \times 10^{-76}</math></b>	$1.12 \times 10^{-16}$	$2.71 \times 10^{-53}$	$2.45 \times 10^{-34}$	$1.44 \times 10^{-6}$	$5.14 \times 10^4$	$1.07 \times 10^6$	$5.78 \times 10^2$	$1.43 \times 10^5$
	Std	<b><math>7.68 \times 10^{-75}</math></b>	$1.05 \times 10^{-16}$	$8.95 \times 10^{-37}$	$8.85 \times 10^{-34}$	$4.52 \times 10^{-6}$	$2.44 \times 10^4$	$3.26 \times 10^5$	$6.51 \times 10^2$	$5.89 \times 10^4$
F4	Mean	<b><math>2.18 \times 10^{-88}</math></b>	$1.21 \times 10^{-5}$	$8.09 \times 10^{-50}$	$1.03 \times 10^{-35}$	$3.01 \times 10^{-29}$	$2.75 \times 10^1$	$7.55 \times 10^1$	1.01	$3.96 \times 10^1$
	Std	<b><math>4.80 \times 10^{-88}</math></b>	$1.39 \times 10^{-5}$	$2.70 \times 10^{-49}$	$7.32 \times 10^{-36}$	$2.31 \times 10^{-29}$	3.18	$2.58 \times 10^1$	1.28	4.67
F5	Mean	$9.61 \times 10^1$	1.47	<b><math>4.88 \times 10^{-2}</math></b>	$9.75 \times 10^1$	$9.63 \times 10^1$	$1.76 \times 10^5$	$9.82 \times 10^1$	$9.80 \times 10^1$	$2.65 \times 10^7$
	Std	$1.03 \times 10^{-1}$	$9.17 \times 10^{-1}$	<b><math>2.38 \times 10^{-2}</math></b>	$7.24 \times 10^{-1}$	$8.27 \times 10^{-1}$	$8.36 \times 10^4$	$1.93 \times 10^{-1}$	$5.66 \times 10^{-1}$	$1.26 \times 10^7$
F6	Mean	$1.92 \times 10^1$	<b><math>5.14 \times 10^{-7}</math></b>	5.08	6.89	2.18	$1.47 \times 10^3$	4.07	$1.00 \times 10^1$	$3.61 \times 10^4$
	Std	$9.95 \times 10^{-1}$	<b><math>3.52 \times 10^{-7}</math></b>	1.09	1.01	$5.37 \times 10^{-1}$	$4.87 \times 10^2$	1.16	1.00	$8.79 \times 10^3$
F7	Mean	<b><math>2.38 \times 10^{-5}</math></b>	$7.99 \times 10^{-4}$	$3.37 \times 10^{-4}$	$2.28 \times 10^{-4}$	$1.47 \times 10^{-3}$	2.82	$3.84 \times 10^{-3}$	$6.87 \times 10^{-3}$	$3.99 \times 10^1$
	Std	<b><math>2.14 \times 10^{-5}</math></b>	$3.47 \times 10^{-4}$	$2.23 \times 10^{-4}$	$1.30 \times 10^{-4}$	$4.92 \times 10^{-4}$	$6.88 \times 10^{-1}$	$4.39 \times 10^{-3}$	$2.58 \times 10^{-3}$	$1.88 \times 10^1$

表 3 ERTLBO 及其对比算法在多峰测试函数实验结果

Table 3 Experimental results of ERTLBO and compared algorithms in multimodal benchmark functions

函数	统计值	ERTLBO	BASTLBO	ITLBOBSO	DSLTLBO	TLBO	SSA	WOA	GWO	GA
F8	Mean	$-7.49 \times 10^3$	$-2.31 \times 10^4$	$-9.59 \times 10^3$	$-1.83 \times 10^4$	$-1.72 \times 10^4$	$-2.17 \times 10^4$	<b><math>-3.49 \times 10^4</math></b>	$-1.64 \times 10^4$	$-7.49 \times 10^3$
	Std	<b><math>6.57 \times 10^2</math></b>	$3.93 \times 10^3$	$7.17 \times 10^2$	$3.13 \times 10^3$	$3.92 \times 10^3$	$1.72 \times 10^3$	$5.95 \times 10^3$	$2.58 \times 10^3$	$1.11 \times 10^3$
F9	Mean	<b>0.00</b>	$2.03 \times 10^2$	$1.59 \times 10^1$	<b>0.00</b>	8.69	$2.34 \times 10^2$	$2.27 \times 10^{-15}$	8.73	$8.57 \times 10^2$
	Std	<b>0.00</b>	$3.35 \times 10^1$	$3.67 \times 10^1$	<b>0.00</b>	$3.40 \times 10^1$	$4.42 \times 10^1$	$2.27 \times 10^{-14}$	7.38	$4.87 \times 10^1$
F10	Mean	<b><math>8.88 \times 10^{-16}</math></b>	$1.04 \times 10^{-8}$	$9.24 \times 10^{-16}$	$4.44 \times 10^{-15}$	$6.71 \times 10^{-2}$	$1.02 \times 10^1$	$3.98 \times 10^{-15}$	$1.29 \times 10^{-7}$	$1.54 \times 10^1$
	Std	<b>0.00</b>	$6.36 \times 10^{-9}$	$3.55 \times 10^{-16}$	<b>0.00</b>	$6.71 \times 10^{-1}$	1.23	$2.46 \times 10^{-15}$	$5.12 \times 10^{-8}$	$8.04 \times 10^{-1}$
F11	Mean	<b>0.00</b>	$5.70 \times 10^{-7}$	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	$1.42 \times 10^1$	$3.64 \times 10^{-3}$	$4.74 \times 10^{-3}$	$3.33 \times 10^{-2}$
	Std	<b>0.00</b>	$5.70 \times 10^{-6}$	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	4.00	$3.64 \times 10^{-2}$	$1.04 \times 10^{-2}$	$7.52 \times 10^1$
F12	Mean	$7.11 \times 10^{-1}$	<b><math>3.11 \times 10^{-4}</math></b>	$1.51 \times 10^{-1}$	$9.22 \times 10^{-2}$	$2.08 \times 10^{-2}$	$3.56 \times 10^{-1}$	$5.29 \times 10^{-2}$	$2.94 \times 10^{-1}$	$7.80 \times 10^{-6}$
	Std	$8.94 \times 10^{-2}$	<b><math>3.11 \times 10^{-3}</math></b>	$4.29 \times 10^{-2}$	$2.64 \times 10^{-2}$	$7.26 \times 10^{-3}$	$1.14 \times 10^{-1}$	$2.61 \times 10^{-2}$	$8.10 \times 10^{-2}$	$7.53 \times 10^{-6}$
F13	Mean	9.97	$2.09 \times 10^{-3}$	<b><math>6.09 \times 10^{-3}</math></b>	9.74	5.25	$6.55 \times 10^{-3}$	2.90	6.74	$5.00 \times 10^{-7}$
	Std	$9.51 \times 10^{-3}$	$4.33 \times 10^{-3}$	<b><math>3.24 \times 10^{-5}</math></b>	$8.01 \times 10^{-1}$	$7.35 \times 10^{-1}$	$1.29 \times 10^{-4}$	1.25	$4.09 \times 10^{-1}$	$3.23 \times 10^{-7}$
F14	Mean	<b><math>9.98 \times 10^{-1}</math></b>	<b><math>9.98 \times 10^{-1}</math></b>	1.63	1.04	<b><math>9.98 \times 10^{-1}</math></b>	1.29	2.79	4.12	4.00
	Std	1.81	$7.06 \times 10^{-17}$	1.30	$2.79 \times 10^{-1}$	<b>0.00</b>	$6.65 \times 10^{-1}$	3.11	3.85	2.91
F15	Mean	<b><math>3.14 \times 10^{-4}</math></b>	$3.90 \times 10^{-4}$	$3.20 \times 10^{-4}$	$3.29 \times 10^{-4}$	$5.67 \times 10^{-4}$	$2.44 \times 10^{-3}$	$7.38 \times 10^{-4}$	$6.26 \times 10^{-3}$	$3.70 \times 10^{-3}$
	Std	<b><math>8.56 \times 10^{-5}</math></b>	$1.50 \times 10^{-4}$	$7.67 \times 10^{-4}$	$1.15 \times 10^{-4}$	$2.01 \times 10^{-3}$	$5.33 \times 10^{-3}$	$5.09 \times 10^{-4}$	$1.01 \times 10^{-2}$	$6.06 \times 10^{-3}$
F16	Mean	<b>-1.03</b>	<b>-1.03</b>	<b>-1.03</b>	<b>-1.03</b>	<b>-1.03</b>	<b>-1.03</b>	<b>-1.03</b>	<b>-1.03</b>	<b>-1.03</b>
	Std	<b><math>6.03 \times 10^{-16}</math></b>	$2.10 \times 10^{-5}$	$1.01 \times 10^{-5}$	$6.68 \times 10^{-16}$	$6.56 \times 10^{-16}$	$2.27 \times 10^{-14}$	$1.19 \times 10^{-9}$	$2.18 \times 10^{-8}$	$5.48 \times 10^{-9}$
F17	Mean	<b><math>3.98 \times 10^{-1}</math></b>	<b><math>3.98 \times 10^{-1}</math></b>	<b><math>3.98 \times 10^{-1}</math></b>	<b><math>3.98 \times 10^{-1}</math></b>	<b><math>3.98 \times 10^{-1}</math></b>	<b><math>3.98 \times 10^{-1}</math></b>	<b><math>3.98 \times 10^{-1}</math></b>	<b><math>3.98 \times 10^{-1}</math></b>	<b><math>3.98 \times 10^{-1}</math></b>
	Std	$1.03 \times 10^{-3}$	0.00	$1.07 \times 10^{-6}$	0.00	0.00	$7.54 \times 10^{-15}$	$9.95 \times 10^{-6}$	$9.24 \times 10^{-7}$	<b><math>6.97 \times 10^{-13}</math></b>



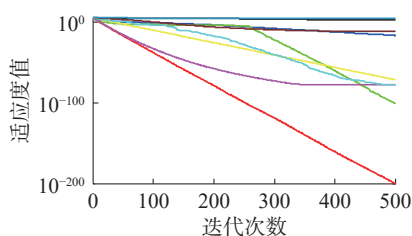
续表 3

函数	统计值	ERTLBO	BASTLBO	ITLBOBSO	DSLTLBO	TLBO	SSA	WOA	GWO	GA
F18	Mean	<b>3.00</b>	<b>3.00</b>	<b>3.00</b>	<b>3.00</b>	<b>3.00</b>	<b>3.00</b>	<b>3.00</b>	<b>3.00</b>	<b>3.00</b>
	Std	$8.13 \times 10^{-4}$	$1.24 \times 10^{-15}$	$8.01 \times 10^{-4}$	$6.45 \times 10^{-16}$	<b><math>6.44 \times 10^{-16}</math></b>	$2.21 \times 10^{-13}$	2.70	$4.44 \times 10^{-5}$	$1.88 \times 10^{-11}$
F19	Mean	<b>-3.86</b>	<b>-3.86</b>	<b>-3.86</b>	<b>-3.86</b>	<b>-3.86</b>	<b>-3.86</b>	<b>-3.86</b>	<b>-3.86</b>	<b>-3.86</b>
	Std	$9.45 \times 10^{-3}$	$2.14 \times 10^{-15}$	$1.08 \times 10^{-4}$	<b><math>2.20 \times 10^{-15}</math></b>	$2.23 \times 10^{-15}$	$6.35 \times 10^{-11}$	$9.45 \times 10^{-3}$	$4.58 \times 10^{-3}$	$9.93 \times 10^{-3}$
F20	Mean	<b>-3.32</b>	-3.16	-3.28	-3.31	-3.30	-3.22	-3.19	-3.26	-3.17
	Std	<b><math>5.54 \times 10^{-12}</math></b>	$8.98 \times 10^{-2}$	$7.11 \times 10^{-2}$	$3.09 \times 10^{-2}$	$4.68 \times 10^{-2}$	$5.75 \times 10^{-2}$	$2.78 \times 10^{-1}$	$7.29 \times 10^{-2}$	$1.09 \times 10^{-1}$
F21	Mean	<b><math>-1.02 \times 10^1</math></b>	-5.06	-5.75	-5.81	$-1.00 \times 10^1$	-7.75	-8.29	-9.27	-7.16
	Std	$3.72 \times 10^{-1}$	<b><math>8.73 \times 10^{-14}</math></b>	1.33	1.59	$6.68 \times 10^{-1}$	3.29	2.58	2.07	3.21
F22	Mean	-7.03	<b><math>-1.04 \times 10^1</math></b>	-5.61	-6.05	$-1.01 \times 10^1$	-7.66	-7.00	$-1.02 \times 10^1$	-8.10
	Std	$9.09 \times 10^{-1}$	<b><math>3.59 \times 10^{-15}</math></b>	1.20	1.86	1.36	3.40	3.20	1.18	3.09
F23	Mean	-7.93	<b><math>-1.05 \times 10^1</math></b>	-6.01	-7.24	-9.97	-8.60	-7.37	$-1.03 \times 10^1$	-8.23
	Std	$6.24 \times 10^{-3}$	<b><math>2.27 \times 10^{-15}</math></b>	1.56	2.42	1.82	3.18	3.45	1.26	3.12

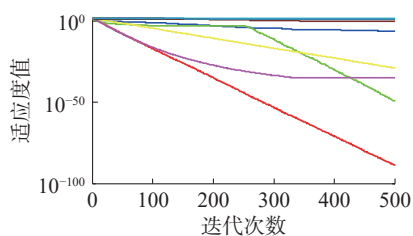
### 3.4 收敛曲线

为了更直观地对比各算法寻优性能, 图 3 为 9 种算法在不同测试函数上进行测试的结果。对于单峰测试函数 F1 和 F4, 迭代初期 ERTLBO 和 DSLTLBO 收敛速度相当。然而随着迭代次数的增加, ERTLBO 收敛能力逐渐增强, 并于迭代后期收敛到全局最优解, 从收敛精度上看明显强于其他对比算法。对于函数 F6, 除 BASTLBO 外其他算法均陷入不同程度的局部最优值。对于函数 F7, 从收敛曲线可以看出 ERTLBO 具有多个梯度下降, 表明算法有效跳出局部最优值, 更接近理论最优值。从多峰测试函数收敛曲线可以看出 ERTLBO 的全局搜索能力同样得到提高。对于函数 F8, ERTLBO 收敛到全局最优值。对于函数 F9 和 F11, ERTLBO 不仅找到了理论最优值, 且收

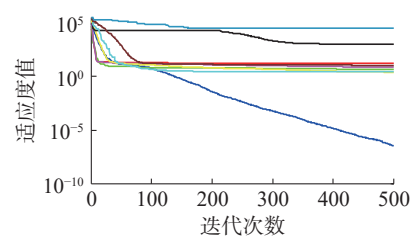
敛速度最快。对于 F15, ERTLBO 不仅收敛到全局最优值, 且收敛速度也保持最佳水平。对于函数 F16~F19, 虽然各算法均收敛到理论最优值, 但 ERTLBO 所需迭代次数更少, 表现出优越的收敛速度。对于函数 F20、F22 和 F23, ERTLBO 均保持最快的收敛速度。综上分析, ERTLBO 算法整体收敛速度快于大部分改进 TLBO 算法以及流行算法, 表现出更好的收敛性能。结合收敛精度分析, 可以得出结论: 在求解单峰测试函数中, 动态自适应学习模式有效提升了算法的收敛精度, 提高算法跳出局部极值点能力。对于多峰测试函数和固定维多峰测试函数, 经验反思机制提升了整体搜索性能, 在迭代前期就能表现出优秀的收敛速度, 可以看出本文提出的改进策略对算法的收敛速度和跳出局部最优能力有积极影响。



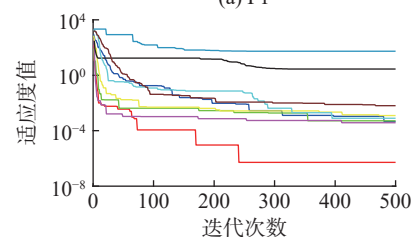
(a) F1



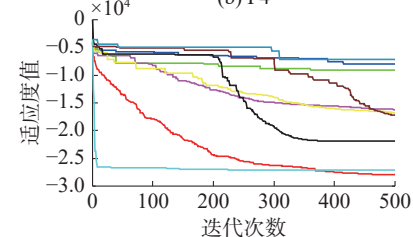
(b) F4



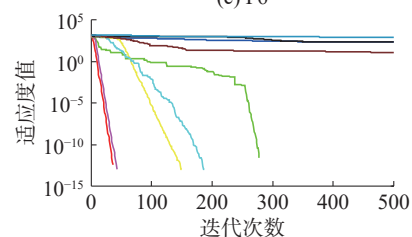
(c) F6



(d) F7



(e) F8



(f) F9



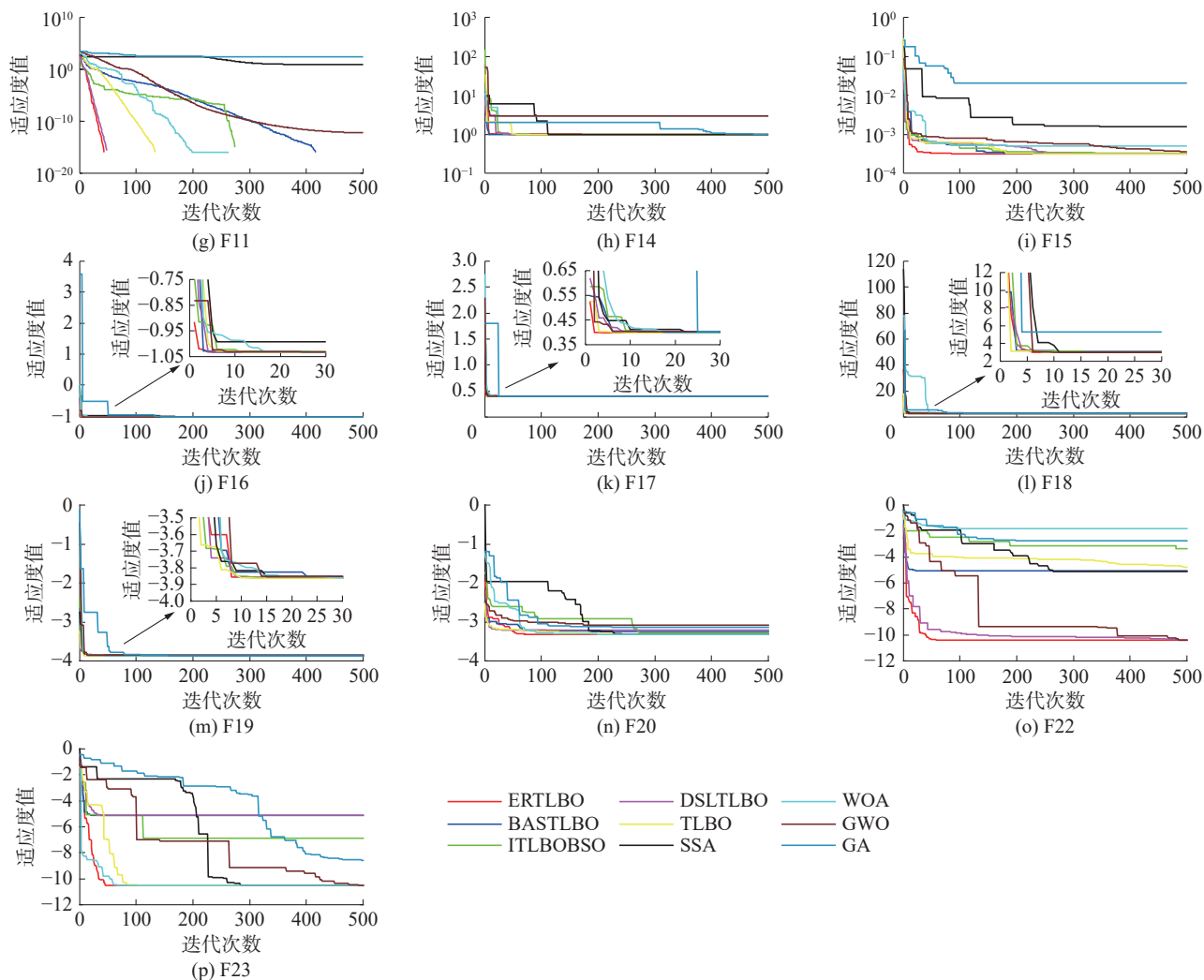


图 3 各算法求解部分测试函数收敛曲线

Fig. 3 Convergence curve behavior obtained by algorithms on some benchmark functions

### 3.5 Wilcoxon 秩和检验

上述实验仅采用了平均值与标准差进行评价, 为了进一步验证改进算法有效性, 本文采用 Wilcoxon 秩和检验进行统计检验<sup>[24]</sup>。显著性水平设置为 5%, 若  $p$  值小于 5%, 则拒绝零假设, 说明两种算法之间具有显著性差异, 否则说明两种算法性能相差不大。Wilcoxon 秩和检验  $p$  值结果如表 4 所示, 因为算

法无法与自身对比, 故表中不再列出 ERTLBO 的  $p$  值。表中  $p$  值大于 0.05 已用粗体展示, NaN 表示数据无效, 即实验样本数据相同, 算法性能相当。

由表 4 可知, 大部分  $p$  值均小于 5%, 表明 ERTLBO 算法优越性在统计检验上是显著的。可以认为 ERTLBO 算法相比于对比算法, 表现出更高的收敛性能。

表 4 各算法得到的 Wilcoxon 秩和检验结果

Table 4 Statistical results of Wilcoxon rank sum test obtained by algorithms

函数	BASTLBO	ITLBOBSO	DSLTLBO	TLBO	SSA	WOA	GWO	GA
F1	$2.56 \times 10^{-34}$	$2.56 \times 10^{-34}$	$2.56 \times 10^{-34}$	$2.56 \times 10^{-34}$	$2.56 \times 10^{-34}$	$2.56 \times 10^{-34}$	$2.56 \times 10^{-34}$	$2.56 \times 10^{-34}$
F2	$2.56 \times 10^{-34}$	$2.56 \times 10^{-34}$	$2.56 \times 10^{-34}$	$2.56 \times 10^{-34}$	$2.56 \times 10^{-34}$	$2.56 \times 10^{-34}$	$2.56 \times 10^{-34}$	$2.56 \times 10^{-34}$
F3	$2.56 \times 10^{-34}$	$1.63 \times 10^{-33}$	$2.56 \times 10^{-34}$	$2.56 \times 10^{-34}$	$2.56 \times 10^{-34}$	$2.56 \times 10^{-34}$	$2.56 \times 10^{-34}$	$2.56 \times 10^{-34}$
F4	$2.56 \times 10^{-34}$	$2.56 \times 10^{-34}$	$2.56 \times 10^{-34}$	$2.56 \times 10^{-34}$	$2.56 \times 10^{-34}$	$2.56 \times 10^{-34}$	$2.56 \times 10^{-34}$	$2.56 \times 10^{-34}$
F5	$2.56 \times 10^{-34}$	$2.56 \times 10^{-34}$	$6.48 \times 10^{-34}$	$3.46 \times 10^{-34}$	$2.56 \times 10^{-34}$	$2.70 \times 10^{-33}$	$2.19 \times 10^{-33}$	$2.56 \times 10^{-34}$
F6	$2.56 \times 10^{-34}$	$2.56 \times 10^{-34}$	$2.56 \times 10^{-34}$	$2.56 \times 10^{-34}$	$2.56 \times 10^{-34}$	$2.56 \times 10^{-34}$	$2.56 \times 10^{-34}$	$2.56 \times 10^{-34}$
F7	$2.56 \times 10^{-34}$	$2.26 \times 10^{-33}$	$6.54 \times 10^{-33}$	$2.56 \times 10^{-34}$	$2.56 \times 10^{-34}$	$2.13 \times 10^{-33}$	$2.56 \times 10^{-34}$	$2.56 \times 10^{-34}$

续表4

函数	BASTLBO	ITLBOBSO	DSLTLBO	TLBO	SSA	WOA	GWO	GA
F8	$2.56 \times 10^{-34}$	$3.74 \times 10^{-33}$	$2.56 \times 10^{-34}$	$4.27 \times 10^{-34}$	$2.56 \times 10^{-34}$	$2.56 \times 10^{-34}$	$4.38 \times 10^{-26}$	<b><math>7.37 \times 10^{-1}</math></b>
F9	$5.64 \times 10^{-39}$	$7.58 \times 10^{-7}$	NaN	<b><math>3.22 \times 10^{-1}</math></b>	$5.64 \times 10^{-39}$	<b><math>3.22 \times 10^{-1}</math></b>	$5.64 \times 10^{-39}$	$5.64 \times 10^{-39}$
F10	$5.79 \times 10^{-38}$	$1.70 \times 10^{-39}$	$7.30 \times 10^{-3}$	$6.97 \times 10^{-41}$	$5.79 \times 10^{-38}$	<b><math>9.51 \times 10^{-1}</math></b>	$5.79 \times 10^{-38}$	$5.79 \times 10^{-38}$
F11	NaN	NaN	NaN	NaN	$5.64 \times 10^{-39}$	<b><math>1.58 \times 10^{-1}</math></b>	$5.64 \times 10^{-39}$	$5.64 \times 10^{-39}$
F12	$2.56 \times 10^{-34}$	$2.56 \times 10^{-34}$	$2.56 \times 10^{-34}$	$2.56 \times 10^{-34}$	$2.56 \times 10^{-34}$	$2.56 \times 10^{-34}$	$2.89 \times 10^{-34}$	$2.56 \times 10^{-34}$
F13	$2.56 \times 10^{-34}$	$2.56 \times 10^{-34}$	$1.25 \times 10^{-32}$	$2.56 \times 10^{-34}$	$2.56 \times 10^{-34}$	$2.56 \times 10^{-34}$	$2.56 \times 10^{-34}$	$2.56 \times 10^{-34}$
F14	$1.78 \times 10^{-37}$	$6.12 \times 10^{-12}$	$8.10 \times 10^{-39}$	$5.64 \times 10^{-39}$	$1.11 \times 10^{-23}$	$1.08 \times 10^{-4}$	<b><math>4.52 \times 10^{-1}</math></b>	$8.93 \times 10^{-5}$
F15	<b><math>2.47 \times 10^{-1}</math></b>	$7.71 \times 10^{-22}$	$2.43 \times 10^{-29}$	$3.00 \times 10^{-6}$	$9.00 \times 10^{-32}$	$1.84 \times 10^{-24}$	$1.32 \times 10^{-11}$	$1.91 \times 10^{-30}$
F16	$3.33 \times 10^{-36}$	$3.85 \times 10^{-6}$	$1.62 \times 10^{-38}$	$1.03 \times 10^{-37}$	$2.56 \times 10^{-34}$	$2.72 \times 10^{-34}$	$2.11 \times 10^{-32}$	$2.80 \times 10^{-34}$
F17	$5.64 \times 10^{-39}$	$4.24 \times 10^{-32}$	$5.64 \times 10^{-39}$	$5.64 \times 10^{-39}$	$2.09 \times 10^{-34}$	$6.99 \times 10^{-26}$	$1.47 \times 10^{-31}$	$2.40 \times 10^{-34}$
F18	$1.28 \times 10^{-34}$	<b><math>1.79 \times 10^{-1}</math></b>	$1.47 \times 10^{-35}$	$9.85 \times 10^{-37}$	$2.56 \times 10^{-34}$	$8.81 \times 10^{-16}$	$9.61 \times 10^{-15}$	$2.56 \times 10^{-34}$
F19	$2.01 \times 10^{-36}$	$1.90 \times 10^{-31}$	$4.27 \times 10^{-38}$	$8.10 \times 10^{-39}$	$2.56 \times 10^{-34}$	<b><math>3.93 \times 10^{-1}</math></b>	$1.61 \times 10^{-13}$	$1.65 \times 10^{-31}$
F20	$2.33 \times 10^{-37}$	$3.96 \times 10^{-22}$	$1.03 \times 10^{-32}$	$4.46 \times 10^{-28}$	$8.00 \times 10^{-6}$	$7.81 \times 10^{-9}$	$2.28 \times 10^{-13}$	$3.16 \times 10^{-2}$
F21	$5.79 \times 10^{-38}$	$1.08 \times 10^{-20}$	$3.60 \times 10^{-21}$	$2.50 \times 10^{-35}$	$1.26 \times 10^{-9}$	$1.52 \times 10^{-22}$	$1.61 \times 10^{-27}$	$2.78 \times 10^{-8}$
F22	$1.62 \times 10^{-38}$	$7.35 \times 10^{-18}$	$5.74 \times 10^{-20}$	$9.83 \times 10^{-33}$	$2.45 \times 10^{-13}$	$2.01 \times 10^{-9}$	$2.56 \times 10^{-34}$	$1.83 \times 10^{-3}$
F23	$2.31 \times 10^{-37}$	$4.97 \times 10^{-23}$	$6.75 \times 10^{-26}$	$6.97 \times 10^{-28}$	$2.40 \times 10^{-14}$	$1.62 \times 10^{-8}$	$3.89 \times 10^{-29}$	$3.77 \times 10^{-11}$

通过分析算法收敛精度、收敛性和统计性检验结果,可以得出结论:经验反思机制与动态自适应权重策略有效增强了TLBO收敛能力,表现出更优秀的收敛性能及稳定性。

### 3.6 仿真时间

本节对ERTLBO和其他算法在23个测试函数上的运行时间进行分析,如表5所示,其中最佳结果用粗体展示。从实验数据上看,对于

大部分测试函数,WOA、GWO、GA、TLBO、SSA相较于其他算法所需的运行时间更少。但结合表2、表3的实验数据,上述算法并不能获得一个较好的解。尽管ERTLBO运行时间不是最快的,但从收敛精度上看ERTLBO相较于其他算法具有显著优势。此外,考虑到无免费午餐定理,适当增加一些运行时间来获得更高精度的解是可接受的。

表5 各算法所需运行时间

Table 5 Statistical results of runtime obtained by algorithms

s

函数	ERTLBO	BASTLBO	ITLBOBSO	DSLTLBO	TLBO	SSA	WOA	GWO	GA
F1	0.2397	0.3548	0.1199	0.2249	0.1256	0.1450	0.0948	0.1902	<b>0.0702</b>
F2	0.2499	0.3831	0.1482	0.2383	0.1496	0.1401	0.1181	0.2333	<b>0.0937</b>
F3	1.8536	3.2118	0.8565	1.1293	0.7901	0.5252	0.4308	0.5524	<b>0.0142</b>
F4	0.1983	0.3187	0.0952	0.2167	0.1302	0.1305	0.1007	0.2159	<b>0.0709</b>
F5	0.2476	0.4192	0.1244	0.2552	0.1490	0.1367	0.1279	0.2506	<b>0.1236</b>
F6	0.1924	0.3519	0.0913	0.2308	0.1041	0.1328	0.0915	0.2012	<b>0.0442</b>
F7	1.0536	1.7270	0.4365	0.6911	0.4548	<b>0.3126</b>	0.2557	0.3643	0.5991
F8	0.3519	0.5840	0.2023	0.3099	0.1994	0.1650	<b>0.1110</b>	0.2427	0.1540
F9	0.2646	0.5725	0.1327	0.2557	0.1187	0.1648	<b>0.0978</b>	0.2206	0.1650
F10	0.2328	0.4602	0.1335	0.2379	0.1419	0.1689	<b>0.0872</b>	0.2246	0.1642
F11	0.3166	0.5420	0.1485	0.2863	<b>0.1420</b>	0.2067	0.1654	0.3346	0.8134
F12	2.7467	4.8544	4.1141	1.6169	2.9185	1.1044	2.1335	<b>0.7198</b>	3.3038
F13	2.6362	4.8122	1.1436	1.5587	5.1400	4.1887	<b>0.6731</b>	2.0365	5.8029
F14	1.6302	10.9205	0.7577	3.6405	2.6188	<b>0.4010</b>	0.3176	0.2700	0.8149
F15	0.1012	0.2646	0.0832	0.1501	0.0956	0.0462	<b>0.0340</b>	0.0447	0.0578

续表 5

函数	ERTLBO	BASTLBO	ITLBOBSO	DSLTLBO	TLBO	SSA	WOA	GWO	GA
F16	0.1042	0.2681	0.0813	0.1393	0.1017	0.0404	0.0366	<b>0.0349</b>	0.0829
F17	0.0815	0.2210	0.0797	0.1320	0.0649	0.0348	<b>0.0265</b>	0.0342	0.0517
F18	0.0799	0.2533	0.0913	0.1215	0.0656	0.0436	<b>0.0293</b>	0.0310	0.0497
F19	0.1178	0.1888	0.1198	0.1629	0.0837	0.0614	<b>0.0358</b>	0.0406	0.0901
F20	0.1233	0.3223	0.1118	0.1556	0.1173	0.0595	<b>0.0411</b>	0.0576	0.0667
F21	0.1297	0.3612	0.1384	0.1681	0.1237	0.0545	<b>0.0493</b>	0.0496	0.0999
F22	0.1470	0.4036	0.1366	0.1792	0.1139	0.0632	0.0470	0.0649	0.1126
F23	0.1705	0.2779	0.1281	0.1776	0.1292	0.0810	0.0597	0.0756	0.1135

#### 4 工程设计问题

为了验证 ERTLBO 算法解决工程问题的性能和有效性, 选用焊接梁设计问题<sup>[25]</sup>与压力容器问题<sup>[26]</sup>进行测试。上述工程设计问题是约束优化问题的具体应用。以优化最小值问题为例, 约束优化问题可以描述为

$$\text{最小化 } f(\mathbf{X}), \mathbf{X} = [x_1 \ x_2 \ \cdots \ x_n] \quad (6)$$

$$\text{约束条件 } \begin{cases} g_j(\mathbf{X}) \leq 0, j = 1, 2, \dots, q \\ h_j(\mathbf{X}) = 0, j = q+1, q+2, \dots, m \end{cases} \quad (7)$$

式中:  $f(\mathbf{X})$  为目标函数;  $\mathbf{X}$  为满足边界条件的决策变量;  $g_j(\mathbf{X})$  为不等式约束;  $h_j(\mathbf{X})$  为等式约束。

对于这类问题, 除了考虑目标函数外, 还要考虑约束条件, 其中罚函数是最常用的约束处理技术之一, 其思想为: 基于个体的约束违反程度构造罚数项, 通过对目标函数增加罚数项来构造惩罚目标函数, 从而将约束优化问题转为无约束优化问题<sup>[27]</sup>。具体的数学模型如下:

$$\text{最小化 } F(\mathbf{X}) = f(\mathbf{X}) + \omega \times \sum_{i=1}^q \max(0, g_i(\mathbf{X})) \quad (8)$$

式中:  $F(\mathbf{X})$  为惩罚目标函数;  $f(\mathbf{X})$  为原目标函数;  $\omega$  为惩罚因子。下节将使用 ERTLBO 求解惩罚目标函数下的焊接梁设计问题和压力容器设计问题。

##### 4.1 焊接梁设计问题

焊接梁设计(如图 4)是一个著名的约束工程设计问题<sup>[25]</sup>, 目标使其设计成本最小化。该问题的决策变量为: 焊缝宽度  $h(z_1)$ 、焊接梁厚度  $b(z_2)$ 、焊接梁长度  $l(z_3)$  以及焊接梁高度  $t(z_4)$ 。

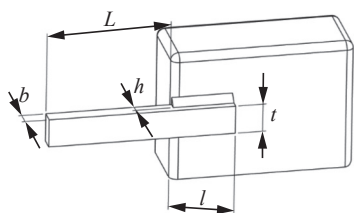


图 4 焊接梁设计问题

Fig. 4 Welded beam design problem

目标函数  $f(\cdot)$ 、约束条件  $g$  以及变量范围如下所示:

优化参数及目标函数如下:

$$\mathbf{z} = [z_1 \ z_2 \ z_3 \ z_4] = [h \ l \ t \ b],$$

$$f(\mathbf{z}) = 1.104 \ 71 z_1^2 z_2 + 0.048 \ 11 z_3 z_4 (14.0 + z_2) \quad (9)$$

问题的约束条件为

$$\begin{cases} g_1(\mathbf{z}) = \tau(\mathbf{z}) - \tau_{\max} \leq 0 \\ g_2(\mathbf{z}) = \sigma(\mathbf{z}) - \sigma_{\max} \leq 0 \\ g_3(\mathbf{z}) = \delta(\mathbf{z}) - \delta_{\max} \leq 0 \\ g_4(\mathbf{z}) = z_1 - z_4 \leq 0, \\ g_5(\mathbf{z}) = P - P_c(\mathbf{z}) \leq 0 \\ g_6(\mathbf{z}) = 0.125 - z_1 \leq 0 \\ g_7(\mathbf{z}) = 1.104 \ 71 z_1^2 + 0.048 \ 11 z_3 z_4 (14.0 + z_2) - 5.0 \leq 0 \end{cases}$$

取值范围:

$$0.05 \leq z_1 \leq 2.00, 0.25 \leq z_2 \leq 1.30, 2.00 \leq z_3 \leq 15.0$$

其中

$$\begin{cases} \tau(\mathbf{z}) = \sqrt{\tau'^2 + 2\tau'\tau'' \frac{z_2}{2R} + \tau''^2}, \tau' = \frac{P}{\sqrt{2}z_1 z_2} \\ \tau'' = \frac{MR}{J}, M = P \left( L + \frac{z_2}{2} \right) \\ R = \sqrt{\frac{z_2^2}{4} + \left( \frac{z_1 + z_3}{2} \right)^2}, J = 2 \left\{ \sqrt{2} z_1 z_2 \left[ \frac{z_2^2}{12} + \left( \frac{z_1 + z_3}{2} \right)^2 \right] \right\} \\ \sigma(\mathbf{x}) = \frac{6PL}{z_4 z_3^2} \\ \delta(\mathbf{z}) = \frac{4PL^3}{E z_3^3 z_4}, P_c(\mathbf{z}) = \frac{4.013E \sqrt{\frac{z_3^2 z_4^6}{36}}}{L^2} \left( 1 - \frac{z_3}{2L} \sqrt{\frac{E}{4G}} \right) \\ P = 2 \ 721.55 \text{ kg}, L = 35.56 \text{ cm}, E = 2.07 \times 10^8 \text{ kPa} \\ G = 8.27 \times 10^7 \text{ kPa} \end{cases}$$

图 5 给出了各算法求解焊接梁设计问题所得收敛曲线, 可以看出 ERTLBO 收敛速度快于其他算法。表 6 统计了 8 种算法求解焊接梁设计问题实验结果, 其中较好结果用粗体展示。从中可以看出, ERTLBO 在  $h=0.205 \ 73$ 、 $l=3.253$ 、 $t=9.036 \ 6$ 、 $b=0.205 \ 73$  处得到的最小代价为 1.695 2, 说明本文算法对于求解此类工程问题具有良好的

性能。

表6 8种算法求解焊接梁设计问题实验结果

Table 6 Experimental results of eight algorithms for solving welded beam design problems

算法	$h$	$l$	$t$	$b$	$f(x)$
ERTLBO	<b>0.2057</b>	<b>3.253</b>	<b>9.0366</b>	<b>0.20573</b>	<b>1.6952</b>
BASTLBO	0.2085	3.2204	8.9964	0.20871	1.7102
ITLBOBSO	0.1970	3.4227	9.0342	0.20588	1.7058
DSLTLBO	0.1937	3.4874	9.0275	0.20625	1.7111
TLBO	0.2089	3.2319	9.0003	0.20976	1.7210
SSA	0.1937	3.4011	9.3469	0.2052	1.7468
WOA	0.2057	3.2530	9.0366	0.2057	1.7185
GWO	0.2046	3.2797	9.0327	0.2060	1.6988
GA	0.2331	2.6193	9.9920	0.2427	2.0964

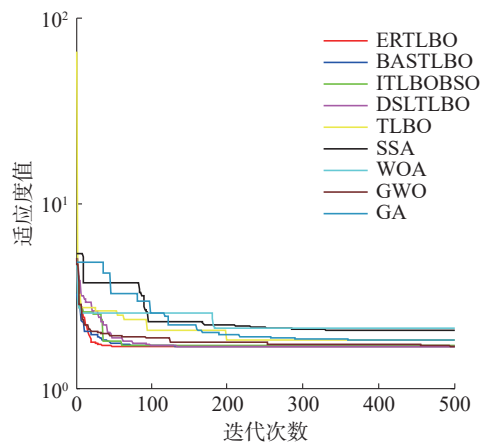


图5 焊接梁问题收敛曲线

Fig. 5 Convergence curve of welded beam design problem obtained by algorithms

#### 4.2 压力容器设计问题

压力容器设计问题<sup>[26]</sup>试图在满足压力要求下使设计总成本最小化,如图6所示。该问题的决策变量为:壳体厚度 $T_s(x_1)$ 、封头厚度 $T_h(x_2)$ 、壳体半径 $R(x_3)$ 、圆柱形截面长度 $L(x_4)$ 。

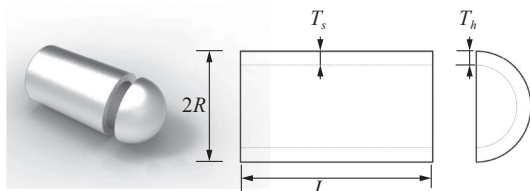


图6 压力容器设计问题

Fig. 6 Pressure vessel problem

优化参数及目标函数为

$$\begin{aligned} \mathbf{x} &= [x_1 \ x_2 \ x_3 \ x_4] = [T_s \ T_h \ R \ L], \\ f(\mathbf{x}) &= 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + \\ &\quad 3.1661x_1^2x_4 + 19.84x_1^2x_3 \end{aligned} \quad (10)$$

问题的约束条件为:

$$\begin{cases} g_1(\mathbf{x}) = -x_1 + 0.0193x_3 \leq 0 \\ g_2(\mathbf{x}) = -x_3 + 0.00954x_3 \leq 0 \\ g_3(\mathbf{x}) = -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 + 1\,296\,000 \leq 0 \\ g_4(\mathbf{x}) = x_4 - 240 \leq 0 \end{cases}$$

取值范围:

$$0 \leq x_1 \leq 99, 0 \leq x_2 \leq 99, 10 \leq x_3 \leq 200, 10 \leq x_4 \leq 200$$

图7为各算法求解压力容器设计问题所得收敛曲线,可以看出ERTLBO于迭代初期便取得了较高的收敛精度,并于迭代后期收敛到全局最优值。表7统计了8种算法在求解压力容器问题时得到的最佳解决方案数据。可以看出ERTLBO取得了最低设计方案,设计成本远低于原始TLBO算法,充分说明了ERTLBO对压力容器设计问题优良可信的求解能力。

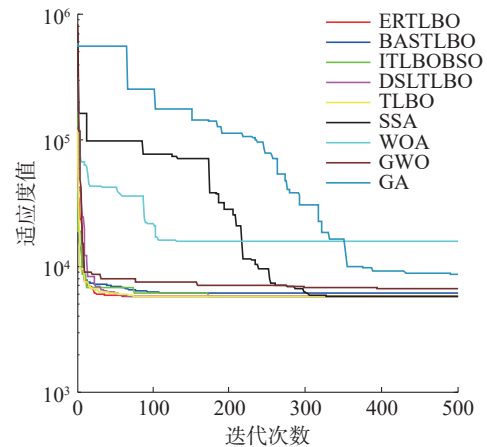


图7 压力容器设计问题收敛曲线

Fig. 7 Convergence curve of pressure vessel design problem obtained by algorithms

表7 各算法应用压力容器设计问题优化结果

Table 7 Optimization results of pressure vessel design problem by algorithms

算法	$T_s(x_1)$	$T_h(x_2)$	$R(x_3)$	$L(x_4)$	$f(\mathbf{x})$
ERTLBO	<b>0.7527</b>	<b>0.4138</b>	<b>42.2128</b>	<b>176.8576</b>	<b>5984.622</b>
BASTLBO	0.8827	0.4310	47.2537	121.7488	6027.769
ITLBOBSO	0.8828	0.4311	47.2707	121.5889	6028.433
DSLTLBO	0.8202	0.3841	45.1290	142.5473	6054.547
TLBO	0.9015	0.4390	48.2043	113.2618	6072.663
SSA	0.8759	0.4279	46.9243	124.7872	6043.665
WOA	0.7942	0.4064	44.5199	150.4761	6057.799
GWO	0.7371	0.3555	41.6434	186.6832	6050.765
GA	0.9008	0.4344	48.0603	115.3819	6092.908

通过对上述典型著名工程设计问题的测试,可以看出本文改进算法ERTLBO具有较强的局



部开发能力,在求解工程优化设计问题具有较好的应用潜力,能够提供优秀的解决方案。

## 5 结束语

为了改善传统 TLBO 算法存在收敛速度慢、收敛精度低,易陷入局部最优等问题,本文提出一种融合经验反思机制的教与学优化算法 (ERTLBO)。ERTLBO 源于教育学原理的思想,运用教育学基本理论设计策略对原算法进行改进,有效地增强并平衡了全局探索和局部开发过程。具体表现为:教学阶段引入经验反思机制,提升班级整体水平,增强算法全局探索性能。学习阶段引入动态自适应权重,实现学生的动态位置更新,提升种群多样性,避免算法陷入局部最优,进而在探索和开发之间取得较好的平衡。

仿真实验采用较高维度的基准函数对改进算法的优化性能进行测试,同时选取了其他改进 TLBO 算法和流行算法作为对比,实验结果发现针对不同类型测试函数,ERTLBO 均具有更佳的收敛速度和精度,具有一定的竞争性和稳定性。最后通过 2 个经典工程设计问题验证改进算法解决实际问题能力,结果表明 ERTLBO 可用于解决实际问题并具有良好的工程应用前景。然而 ERTLBO 也不可避免地提高了运行时间,因此未来的工作方向是在现有基础上继续改进,保证优化性能的同时减少运行时间开销,提高计算效率。其次,可将 ERTLBO 应用到多目标优化中,并结合实际问题作进一步探究。

## 参考文献:

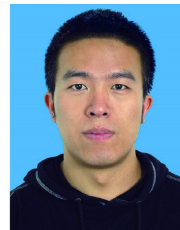
- [1] 贾鹤鸣,李瑶,孙康健. 基于遗传乌燕鸥算法的同步优化特征选择[J]. 自动化学报, 2022, 48(6): 1601–1615.  
JIA Heming, LI Yao, SUN Kangjian. Simultaneous feature selection optimization based on hybrid sooty tern optimization algorithm and genetic algorithm[J]. Acta automatica sinica, 2022, 48(6): 1601–1615.
- [2] 贾鹤鸣,姜子超,李瑶. 基于改进秃鹰搜索算法的同步优化特征选择[J]. 控制与决策, 2022, 37(2): 445–454.  
JIA Heming, JIANG Zichao, LI Yao. Simultaneous feature selection optimization based on improved bald eagle search algorithm[J]. Control and decision, 2022, 37(2): 445–454.
- [3] 贾鹤鸣,刘宇翔,刘庆鑫,等. 融合随机反向学习的黏菌与算术混合优化算法[J]. 计算机科学与探索, 2022, 16(5): 1182–1192.  
JIA Heming, LIU Yuxiang, LIU Qingxin, et al. Hybrid algorithm of slime mould algorithm and arithmetic optimization algorithm based on random opposition-based learning[J]. Journal of frontiers of computer science and technology, 2022, 16(5): 1182–1192.
- [4] RAO R V, SAVSANI V J, VAKHARIA D P. Teaching-learning-based optimization: an optimization method for continuous non-linear large scale problems[J]. Information sciences, 2012, 183(1): 1–15.
- [5] 杨闰. 改进 TLBO 算法及其在天线优化设计中的应用[D]. 杭州: 杭州电子科技大学, 2019: 44–54.  
YANG Run. Improved TLBO algorithm and its application in antenna optimization design[D]. Hangzhou: Hangzhou Dianzi University, 2019: 44–54.
- [6] 麻利新. 基于教与学算法的风-火-蓄电力系统优化调度[D]. 银川: 宁夏大学, 2020: 31–38.  
MA Lixin. Optimal dispatching of wind power-thermal power-pumped storage power system based on teaching and learning algorithm[D]. Yinchuan: Ningxia University, 2020: 31–38.
- [7] 曾恋捷. 基于教与学优化算法的结构可靠性分析方法[D]. 广州: 暨南大学, 2020: 59–71.  
ZENG Lianjie. Reliability analysis method based on teaching and learning optimization algorithm[D]. Guangzhou: Jinan University, 2020: 59–71.
- [8] 李木子. 基于神经网络的手势识别技术研究[D]. 石家庄: 河北地质大学, 2021: 35–46.  
LI Muzi. Research on gesture recognition technology based on neural network[D]. Shijiazhuang: Hebei GEO University, 2021: 35–46.
- [9] WOLPERT D H, MACREADY W G. No free lunch theorems for optimization[J]. IEEE transactions on evolutionary computation, 1997, 1(1): 67–82.
- [10] 李丽荣,杨坤,王培崇. 融合头脑风暴思想的教与学优化算法[J]. 计算机应用, 2020, 40(9): 2677–2682.  
LI Lirong, YANG Kun, WANG Peichong. Improved teaching & learning based optimization with brain storming[J]. Journal of computer applications, 2020, 40(9): 2677–2682.
- [11] 欧阳城添,周凯. 融合改进天牛须搜索的教与学优化算法[J]. 计算机工程与应用, 2022, 58(4): 91–99.  
OUYANG Chengtian, ZHOU Kai. Teaching-learning based optimization algorithm with improved beetle antennae search[J]. Computer engineering and applications, 2022, 58(4): 91–99.
- [12] 李丽荣,李木子,李崔灿,等. 具有动态自适应学习机制的教与学优化算法[J]. 计算机工程与应用, 2020, 56(19): 62–67.  
LI Lirong, LI Muzi, LI Cuican, et al. Teaching and learning based optimization with dynamic self-adaptive learning[J]. Computer engineering and applications, 2020,

- 56(19): 62–67.
- [13] LAKSHMI V A, MOHANAIHAH P. WOA-TLBO: whale optimization algorithm with Teaching-learning-based optimization for global optimization and facial emotion recognition[J]. *Applied soft computing*, 2021, 110: 107623.
- [14] XU Yunlang, YANG Zhile, LI Xiaoping, et al. Dynamic opposite learning enhanced teaching–learning-based optimization[J]. *Knowledge-based systems*, 2020, 188: 104966.
- [15] TSAI H C. Confined teaching-learning-based optimization with variable search strategies for continuous optimization[J]. *Information sciences*, 2019, 500: 34–47.
- [16] SHUKLA A K, SINGH P, VARDHAN M. An adaptive inertia weight teaching-learning-based optimization algorithm and its applications[J]. *Applied mathematical modelling*, 2020, 77: 309–326.
- [17] TAHERI A, RAHIMIZADEH K, RAO R V. An efficient balanced teaching-learning-based optimization algorithm with Individual restarting strategy for solving global optimization problems[J]. *Information sciences*, 2021, 576: 68–104.
- [18] LI Yu, ZHAO Yiran, LIU Jingsen. Dynamic sine cosine algorithm for large-scale global optimization problems[J]. *Expert systems with applications*, 2021, 177: 114950.
- [19] 贾鹤鸣, 刘庆鑫, 刘宇翔, 等. 融合动态反向学习的阿奎拉鹰与哈里斯鹰混合优化算法 [J]. 智能系统学报, 2023, 18(1): 104–116.
- JIA Heming, LIU Qingxin, LIU Yuxiang, et al. Hybrid Aquila and Harris Hawks optimization algorithm with dynamic opposition-based learning[J]. *CAAI transactions on intelligent systems*, 2023, 18(1): 104–116.
- [20] MIRJALILI S, GANDOMI A H, MIRJALILI S Z, et al. Salp swarm algorithm: a bio-inspired optimizer for engineering design problems[J]. *Advances in engineering software*, 2017, 114: 163–191.
- [21] MIRJALILI S, LEWIS A. The whale optimization algorithm[J]. *Advances in engineering software*, 2016, 95: 51–67.
- [22] MIRJALILI S, MIRJALILI S M, LEWIS A. Grey wolf optimizer[J]. *Advances in engineering software*, 2014, 69: 46–61.
- [23] 吕冰焱, 姜志翔, 宁春玉. 基于 PSO 和 GA 混合优化的 FCM 算法 [J]. 长春理工大学学报(自然科学版), 2021, 44(6): 125–130.
- LYU Bingyao, JIANG Zhiao, NING Chunyu. FCM algorithm based on hybrid optimization of PSO and GA[J]. *Journal of Changchun University of Science and Technology (natural science edition)*, 2021, 44(6): 125–130.
- [24] LIU Qingxin, LI Ni, JIA Heming, et al. Modified remora optimization algorithm for global optimization and multi-level thresholding image segmentation[J]. *Mathematics*, 2022, 10(7): 1014.
- [25] ABUALIGAH L, YOUSRI D, ABD ELAZIZ M, et al. Aquila Optimizer: a novel meta-heuristic optimization algorithm[J]. *Computers & industrial engineering*, 2021, 157: 107250.
- [26] LIU Qingxin, LI Ni, JIA Heming, et al. A hybrid arithmetic optimization and golden sine algorithm for solving industrial engineering design problems[J]. *Mathematics*, 2022, 10(9): 1567.
- [27] 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, 27(7): 1871–1883.
- WANG Yihui, GAO Liang. Improvement of crow search algorithm and its application in engineering constrained optimization problems[J]. *Computer integrated manufacturing systems*, 2021, 27(7): 1871–1883.

#### 作者简介:



吴迪, 副教授, 博士, 主要研究方向为元启发式优化算法、现代教育技术。



贾鹤鸣, 教授, 主要研究方向为群体智能优化算法与工程应用。主持福建省自然科学基金等项目 10 余项。发表学术论文 60 余篇。



刘庆鑫, 硕士研究生, 主要研究方向为智能优化算法及现实应用。