马云鹏, 张水长, 张淑玥, 等. 基于差分变异算子的教与学优化算法研究[J]. 智能计算机与应用,2026,16(7):172-180.
DOI:10.20169/j.issn.2095-2163.24112101

基于差分变异算子的教与学优化算法研究

马云鹏[1], 张水长[2], 张淑玥[2], 董雄飞[2]
(1 天津商业大学 信息工程学院, 天津 300134; 2 天津商业大学 理学院, 天津 300134)

摘 要: 教与学算法是一种基于学校教学活动的启发式优化算法,具有计算量小、参数设置少、收敛精度高、收敛速度快等特点,但存在容易陷入局部最优等问题。对原始的教与学算法进行了研究,提出了改进的差分变异教与学优化算法(DVTLBO),在原始的算法上引入了新的种群更新机制。在教学阶段,引入了教学效率因子,用来控制学生向老师学习新知识的能力。在学习阶段,融合了差分进化算法中的差分变异思想,设置新的种群更新机制。通过 14 种无约束测试函数对 DVTLBO 进行评估,并与其他几种启发式优化算法和 TLBO 进行对比,结果表明 DVTLBO 具有更高的搜索精度和更快的收敛速度。

关键词: 启发式优化算法; 教与学优化算法; 教学效率; 差分变异
中图分类号: TP183 **文献标志码**: A **文章编号**: 2095-2163(2026)07-0172-09

Research on Differential-Variation Teaching-Learning-Based Optimization algorithm

MA Yunpeng[1], ZHANG Shuichang[2], ZHANG Shuyue[2], DONG Xiongfei[2]
(1 School of Electronic and Computer Engineering, Tianjin University of Commerce, Tianjin 300134, China;
2 School of Sciences, Tianjin University of Commerce, Tianjin 300134, China)

Abstract: Teaching-Learning-Based Optimization (TLBO) algorithm is a heuristic optimization algorithm based on school teaching activities, but it is easy to fall into local optimal problems. It has the characteristics of less computation, less parameter setting, high convergence accuracy and fast convergence speed. This paper proposes an improved Differential-Variation Teaching-Learning-Based Optimization algorithm (DVTLBO), which introduces a new population updating mechanism over the original algorithm. In the teaching stage, the teaching efficiency factor is introduced to control students´ ability to learn new knowledge from teachers. In the learning stage, the idea of differential variation in differential evolution algorithm is integrated to set up a new population renewal mechanism. DVTLBO is evaluated by 14 unconstrained test functions, and compared with other heuristic optimization algorithms and TLBO, the results show that DVTLBO has higher search accuracy and faster convergence speed.

Key words: heuristic optimization algorithms; Teaching-Learning-Based Optimization algorithm; teaching efficiency; differential variation

0 引 言

教与学优化算法是一种基于教学活动的启发式优化算法,是由 Rao 等学者[1]在 2011 年提出具有计算量小、参数设置少、收敛精度高、收敛速度快等特点。文献[2]对教与学算法最新的进展作了一个全面的研究, 总结了 TLBO 的研究现状和存在的问题:在教学阶段,种群容易过早收敛;在学习阶段,种群容易脱离局部最优,种群搜索空间有限,难以进一步增加种群多样性等。

基金项目: 国家自然科学基金青年项目(62203332)。
作者简介: 张水长(2000—),男,硕士研究生。主要研究方向:智能优化算法; 张淑玥(2001—),女,硕士研究生。主要研究方向:智能优化算法; 董雄飞(2001—),男,硕士研究生。主要研究方向:智能优化算法。
通信作者: 马云鹏(1989—),男,博士,副教授。主要研究方向:智能优化算法,机器学习。Email: mayunpeng@tjcu.edu.cn。
收稿日期: 2024-11-21

针对目前 TLBO 算法存在的问题，文献[3]提出了一种结合动态环邻域拓扑结构的方法。该方法有效地降低全局最优解对个体的影响，从而提高算法的搜索性能。文献[4]在教学阶段使用黄金正弦搜索模型来解决原点偏移问题，在学习阶段之后加入多种群学习者阶段，提高了算法的搜索精度和收敛速度。文献[5]融合了黏模算法和基于教学的优化的算法，以平衡开发和探索能力，并引入了一个莱维飞行变异，以最大限度地提高算法的探索能力。文献[6]通过平衡开发和探测能力来提升算法的性能，引入自我反馈阶段以保持良好的开发能力，引入交叉变异阶段以提高算法的探索能力。文献[7]提出一种三角形变化的更新机制，在种群数量的线性增长阶段，利用种群的自适应均值和方差，产生服从高斯分布的新个体；在种群数量的线性递减阶段，删除一些高度相似的个体，降低了原 TLBO 的计算成本，并将其扩展到人工神经网络(ANN) 的参数优化中。文献[8]把更新的学生和当前的学生的适应度水平进行了对比，将教和学阶段分为“低，中，高”三个级别，并对 5 种光伏电池和模块进行了研究，证明其在双二极管模型中具有优秀的性能表现。文献[9]提出了一种新的距离适应度学习策略，引导整个种群根据微分算子同时从教师和距离最优候选者学习，以分别提高开发和探索能力。文献[10]引入变异算子和动态边界约束策略以提高算法的搜索能力和收敛速度，并提出了一种贪婪梯度信息估计方法，与变异算子相结合，建立了一种基于梯度的变异策略，在探索和利用之间取得了较好的平衡。文献[11]提出了一种自适应混合自学习的 TLBO 算法，融合了高斯分布，2 种模式切换的更新机制一定程度上克服了处理具有移位解的问题时表现不佳的问题。文献[12]将连续算法转化为离散算法，更新后的算法没有任何控制参数，收敛速度快，适应于不同的配电系统，可用来解决离散问题。文献[13]应用人工神经网络(ANN)和 TLBO 来预测土耳其的能源消耗，并与反向传播训练的神经网络模型(ANN-BP)进行对比，结果表明 ANN-TLBO 模型具有更佳的预测准确性。文献[14]运用随机键表示法设计基于扩展升序排列规则，将连续解映射到舰载机的起飞顺序和起飞跑道，设计了基于教与学优化的舰载机起飞出动动态调度算法。文献[15]使用 MapReduce 编程范式在分布式系统上实现 TLBO 算法，创建了一种称为 MapReduce 的基于教学的优化的新方法。文献[16]提出一种扰动疏散机制，避免繁育蜣螂群体聚集，同时引入自适应惯性权重策略，提高了算法的收敛速度。文献[17]设计了动态的课堂进化，融合了学生的多元搜索和教师的适应性自学，提出了一种基于教学-学习的动态优化算法。文献[18]提出了一种基于粒子群优化算法-门控循环单元的粮食产量预测模型，利用 PSO 算法对 GRU 模型的关键参数进行寻优，解决了 GRU 易陷入局部最优解的问题。文献[19]在可靠性分析中采用了教与学算法及三角形变种群，以克服 HLRF 算法或其对一阶可靠性方法的改进中较易陷入周期解和混沌等不稳定状态数值的困难。

为了提高算法的搜索精度和收敛速度，本文提出一种改进的差分变异的教与学优化算法(Differential-Variation Teaching Learning Based Optimization，DVTLBO)。该算法在“教阶段”和“学阶段”引入了新的种群更新机制：

(1)在“教阶段”，引入教学效率因子和惯性权值。教学效率因子用来控制学生向老师学习新知识的效率，惯性权值因子用来控制学生对已经学过知识的影响。

(2) 在“学阶段”，融合差分进化算法中的变异思想，引入差分变异因子，设置新的种群更新机制。

(3)在“教阶段”和“学阶段”，将学生个体分为“优生”和“差生”。“优生”有较强的自主学习能力，可以自己学习，“差生”自主学习能力较差，侧重于向老师学习。为了验证 DVTLBO 算法的性能，用 14 种无约束测试函数进行评估，并与其他几种启发式优化算法和原始的教与学算法进行对比，结果 表明 DVTLBO 具有更高的搜索精度和更快的收敛速度。

1　教与学优化算法

1.1　教与学优化算法

教与学算法是一种以教学过程为基础的启发式优化算法，旨在模拟教室中教师和学生之间的互动。该算法主要包括教学阶段和学生阶段 2 个部分，核心思想是通过这 2 个阶段来促进解的优化。

在初始化阶段，需要定义种群规模、函数、维度、最大迭代次数、当前迭代次数、循环次数、循环条件和终止条件等各种初始化参数。

1.2　教学阶段

在教师阶段，算法将当前最佳解视为教师，其他解视为学生。教师尝试“教授”学生，使学生的解逐

渐接近最佳解。该阶段的种群更新公式如下：

$$X_{\text{new},i} = X_{\text{old},i} + r_i \times (M_{\text{new}} + T_f \times M_i) \quad (1)$$

$$T_f = \text{round}[1 + \text{rand}(0, 1)] \quad (2)$$

其中，$X_{\text{new},i}$ 为更新后的个体位置；$X_{\text{old},i}$ 为更新前的个体位置；T_f 为教学系数，取值为 1 或 2；r_i 为随机数，取值范围为(0 ,1)；M_{new} 表示教师的最优成绩；M_i 为平均成绩。

1.3 学习阶段

在学生阶段，学生之间相互学习。每个学生都会与随机选择的其他学生进行比较，并尝试从更好的学生那里学习。如果一个学生的解比另一个学生好，后者就会尝试更新自己的解，朝着前者的解靠近。$f(X_i)$ 和 $f(X_j)$ 为第 i 个和第 j 个个体的适应度值。该阶段种群更新公式如下：

$$X_{\text{new},i} = \begin{cases} X_{\text{old},i} + r_i \times (X_i - X_j), & \text{if } f(X_i) < f(X_j) \\ X_{\text{old},i} + r_i \times (X_j - X_i), & \text{if } f(X_i) > f(X_j) \end{cases} \quad (3)$$

2 差分变异的教与学优化算法

在原始的 TLBO 算法基础上，对种群更新机制进行改进，得到了 DVTLBO 算法。首先，把学生个体分为 2 类，包括“优生”和“差生”；然后，在“教阶段”引入教学效率因子 和惯性权值分别用来控制学生向老师学习的效率和用来控制学生对已经学过知识的影响；最后，在“学阶段”引入差分变异因子，设置新的种群更新机制。该算法具体的更新机制如下所述。

2.1 教学阶段

在现实的课堂教学活动中，教学效率影响着学生向老师学习的效率。老师的教学技能水平、教学状态、与同学的熟悉程度等因素都会影响教学效率。引入教学效率因子可以很好地控制学生向老师学习的效率。班级的平均分反映了一个班学生的整体成绩水平，在平均分之上的学生视为“优生”，在平均分之下的学生视为差生，用这样的方式合理地把学生分为“优生”和“差生”两类。“优生”有较强的自主学习能力，主要向最优秀的学生学习来获取知识，其在算法中的个体位置更新公式为：

$$X_{\text{new},i} = X_{\text{old},i} \times y_i + \text{rand} \times (X_{\text{best}} + X_{\text{old},i}) \quad (4)$$

其中，y_i 表示控制学生对已经学过知识的惯性权值，X_{best} 表示代表班级成绩最优秀的学生。

“差生”主要向老师学习来获取知识，其在算法中的个体位置更新公式如下：

$$X_{\text{new},i} = X_{\text{old},i} + (\text{rand} - 0.5) \times 2 \times (X_{\text{mean}} - X_{\text{old},i}) \times w_1 \times x_l + \text{diff} \times w_2 \quad (5)$$

其中，X_{mean} 表示班级学生成绩均值，x_l 表示控制学生向老师学习效率的教学效率因子。教学效率因子的表达式为：

$$x_l = 1 - e^{-\alpha \times \text{iter}} \quad (6)$$

其中，iter 表示当前迭代次数。

惯性权值因子的表达式分别如下：

$$y_i = e^{-\alpha \times \text{iter}} \quad (7)$$

$$w_1 = \sin\left(\frac{\pi}{2} \times \frac{\text{iter}}{\text{Maxiter}}\right) \quad (8)$$

$$w_2 = \cos\left(\frac{\pi}{2} \times \frac{\text{iter}}{\text{Maxiter}}\right) \quad (9)$$

其中，Maxiter 表示最大迭代次数，w_1 和 w_2 表示 2 个惯性权值，用来加快学生向老师获取知识的速度，加快算法的搜索速度，提高全局搜索能力，避免陷入局部最优。

2.2 学习阶段

在“学习阶段”，同“教学阶段”一样，学生将被分为“优生”和“差生”两组。“优生”具有较好的自学能力，不仅可以从最优秀的学生那里获取知识，也可以独立学习。在“优生”的个体位置更新机制中引入差分变异因子 F，预先设置 3 个个体 $X^1_{\text{new},i}$，$X^2_{\text{new},i}$，$X^3_{\text{new},i}$，利用更新机制生成新的个体 X_1，X_2，X_3，运用差分变异的更新机制，生成 $X_{\text{new},i}$ 进入下一次迭代。引入差分变异更新机制能够增加种群多样性，并加快收敛速度。“差生”自学能力较差，主要向老师学习来获取知识。生成新个体的更新机制公式为：

if $f(X^1_{\text{new},i}) > f(X^2_{\text{new},i})$

$$X_1 = X^1_{\text{new},i} + x_l \times F \times (X^2_{\text{new},i} - X^1_{\text{new},i}) \quad (10)$$

else

$$X_1 = X^1_{\text{new},i} + (\text{rand} - 0.5) \times 2 \times (X_{\text{upper limit}} - X_{\text{lower limit}}) \quad (11)$$

$$X_{\text{new},i} = X_1 + x_l \times F \times (X_1 - X_2) + x_l \times F \times (X_2 - X_3) \quad (12)$$

其中，$f(X^1_{\text{new},i})$ 表示第 $X^1_{\text{new},i}$ 个个体的适应度值；x_l 表示教学效率因子；F 表示差分变异因子，数学表达式如下：

$$F = \frac{w_1}{1 + w_2} \quad (13)$$

进一步推得差生个体位置更新公式为：

$$X_{\text{new},i} = X_{\text{old},i} + y_i \times w_2 \times (X_{\text{best}} + X_{\text{old},i}) \quad (14)$$

其中，w_1 和 w_2 表示惯性权值。

DVTLBO 算法的流程如图 1 所示。

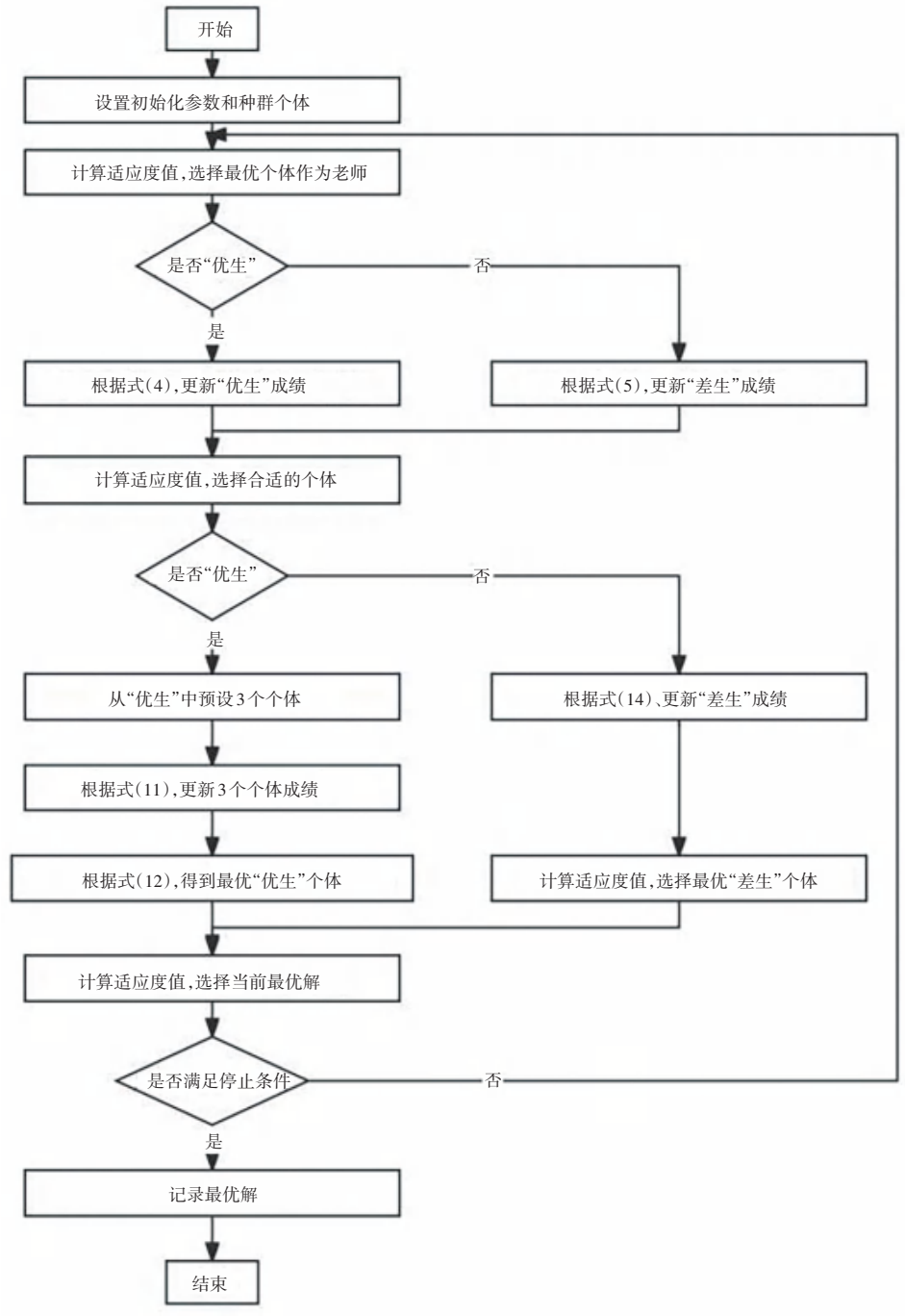


图 1　DVTLBO 算法流程图

Fig. 1　Flowchart of the DVTLBO algorithm

3　性能测试

3.1　测试函数和参数设置

为了验证 DVTLBO 算法的性能，采用了 14 个基准数学函数来测试算法的收敛速度和收敛精度。表 1 详细记录了这些测试函数，包括搜索范围、理论全局最优和可接受解。表 1 中，F_1 ~ F_5 为单峰高维函数，F_6 ~ F_{10} 为多峰高维函数，其余 4 个为旋转函数。单峰高维函数只有一个局部极小值点，优化算法通常能够相对较快地收敛到这个最优解，这有利于验证算法的收敛速度。多峰高维函数在定义域内有多个局部极小值和(或)极大值点，存在多个局部最优解，找到全局最优解变得更加困难，传统的优化算法可能会陷入局部最优解而无法找到全局最优解，这有利于验证算法避免陷入局部最优的能力和探索能力。表 2 记录了各个算法的参数设置。其中，种群规模设置为 20，每一次运行迭代 1 000 次，基准函数依次设置为 10 维、30 维。为了减小统计误差和随机性影响，每个算法独立运行 30 次。

表 1 实验中使用的 14 个测试函数

Table 1 14 test functions used in the experiment

测试函数	测试范围	最优解	最优解点
F_1 (sphere)	[-100, 100]	0	$[0]^n$
F_2 (quadric)	[-100, 100]	0	$[0]^n$
F_3 (sumsquares)	[-100, 100]	0	$[0]^n$
F_4 (zakharov)	[-10, 10]	0	$[0]^n$
F_5 (rosenbrock)	[-2.048, 2.048]	0	$[1]^n$
F_6 (ackley)	[-32.768, 32.768]	0	$[0]^n$
F_7 (rastrigin)	[-5.12, 5.12]	0	$[0]^n$
F_8 (weierstrass)	[-0.5, 0.5]	0	$[0]^n$
F_9 (griewank)	[-600, 600]	0	$[0]^n$
F_{10} (schwefel)	[-500, 500]	0	$[420.96]^n$
F_{11} (ackleyrotfunc)	[-0.5, 0.5]	0	$[0]^n$
F_{12} (rastriginrotfunc)	[-pi, pi]	-1	$[0]^n$
F_{13} (weierstrassrot)	[-3, 1]	0	$[0]^n$
F_{14} (griewankrotfunc)	[-100, 100]	0	$[0]^n$

表 2 各算法的参数设置

Table 2 Parameter setting of each algorithm

算法	种群规模	迭代次数	维数
POA	20	1 000	10,30
PSO	20	1 000	10,30
COA	20	1 000	10,30
GWO	20	1 000	10,30
TLBO	20	1 000	10,30
DVTLBO	20	1 000	10,30

3.2 DVTLBO 与原始算法比较

为了更直观地分析算法的性能,将 DVTLBO 算法与其他传统的启发式优化算法进行对比实验。表 3 和表 4 分别验证了 DVTLBO 算法与其他算法在 10 维和 30 维独立运行 30 次的均值和标准差的性能。实验表明,均值越接近最优解,算法的求解精度越高,表现最佳的结果以加粗字体表示。

表 3 POA、PSO、COA、GWO、TLBO、DVTLBO 在 10 维函数上得到的平均值和标准差

Table 3 The mean and standard deviation of POA, PSO, COA, GWO, TLBO, DVTLBO on the 10-dimensional functions

函数	POA		PSO		COA		GWO		TLBO		DVTLBO	
	均值	标准差	均值	标准差	均值	标准差	均值	标准差	均值	标准差	均值	标准差
F_1	1.89×10^{-212}	0	1.26×10^{2}	1.06×10^{1}	1.00×10^{1}	4.23×10^{-17}	1.73×10^{-97}	4.25×10^{-97}	2.64×10^{-226}	0	**0**	**0**
F_2	1.39×10^{-221}	0	1.28×10^{2}	1.39×10^{1}	1.00×10^{1}	4.23×10^{-17}	8.56×10^{-97}	4.10×10^{-96}	1.80×10^{-227}	0	**0**	**0**
F_3	1.05×10^{-210}	0	7.32×10^{2}	6.38×10^{1}	5.51×10^{-1}	1.13×10^{-16}	7.50×10^{-94}	4.10×10^{-93}	2.96×10^{-226}	0	**0**	**0**
F_4	3.77×10^{-217}	0	1.69×10^{7}	1.21×10^{7}	8.25×10^{-2}	4.23×10^{-17}	3.70×10^{-55}	1.58×10^{-54}	9.42×10^{-110}	2.61×10^{-109}	**0**	**0**
F_5	6.27	9.75×10^{-1}	4.16×10^{3}	1.85×10^{-12}	5.49×10^{-1}	3.39×10^{-16}	6.59	5.46×10^{-1}	**3.37**	**6.85×10^{-1}**	7.80×10^{1}	2.21
F_6	3.52×10^{-15}	1.23×10^{-15}	1.19×10^{1}	2.91×10^{-1}	1.88×10^{-1}	8.47×10^{-17}	4.83×10^{-15}	1.53×10^{-15}	4.00×10^{-15}	0	**4.44×10^{16}**	**0**
F_7	**0**	**0**	2.42×10^{2}	1.57×10^{1}	5.21×10^{-2}	0	3.66×10^{-1}	1.39	3.65	3.10	**0**	**0**
F_8	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
F_9	**0**	**0**	1.05	6.74×10^{-2}	4.22×10^{-1}	5.65×10^{-17}	1.72×10^{-2}	1.60×10^{-2}	1.25×10^{-2}	1.85×10^{-2}	**0**	**0**
F_{10}	1.20×10^{3}	2.74×10^{2}	2.38×10^{3}	1.39×10^{-12}	2.38×10^{3}	1.50×10^{-32}	1.55×10^{3}	2.76×10^{2}	**7.70×10^{2}**	**3.30×10^{2}**	1.81×10^{3}	6.01×10^{2}
F_{11}	3.32×10^{-15}	9.01×10^{-16}	2.17×10^{1}	1.08×10^{-14}	1.47×10^{-1}	2.82×10^{-17}	6.18×10^{-1}	8.88×10^{-1}	5.37×10^{-1}	6.72×10^{-1}	**0**	**0**
F_{12}	**0**	**0**	3.16×10^{2}	1.16×10^{-13}	3.57×10^{-2}	0	7.74	4.44	1.40×10^{1}	1.24×10^{1}	**0**	**0**
F_{13}	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
F_{14}	**0**	**0**	1.21	3.58×10^{-2}	4.88×10^{-2}	3.53×10^{-17}	8.74×10^{-2}	7.61×10^{-2}	2.27×10^{-2}	5.02×10^{-2}	**0**	**0**

表 3 的实验数据显示,DVTLBO 算法在 F_1、F_2、F_3、F_4、F_6、F_7、F_{11} 这些函数上比其他所有算法的均值和标准差都要低,在 F_9、F_{12}、F_{14} 这些函数上与 POA 算法共同表现出了最好的性能。在 F_5 和 F_{10} 上,TLBO 算法表现出了最好的性能,证明了 TLBO 算法在某些功能上优于其他算法。在 F_8 和 F_{13} 上 DVTLBO 算法和其他算法都能取得最优的均值和标准差。从表 4 可以看出,DVTLBO 算法在 F_1、F_2、F_3、F_4、F_6、F_7、F_8、F_9、F_{10}、F_{11}、F_{12}、F_{13}、F_{14} 均取得了最佳的均值和标准差,COA 算法在 F_5 上取得了最佳的均值和标准差。

算法的收敛速度是其在迭代过程中向最优解逼近的速度,是评估算法性能的重要指标,收敛曲线图可以很好地反映算法的收敛速度。为了更直观地显示 DVTLBO 算法的收敛速度,图 2 和图 3 分别给出了 DVTLBO 和其他算法在 10 维、30 维的部分函数收敛曲线。图 2 给出了 F_1 ~ F_4 在 10 维的收敛曲线对比图,图 3 给出了 F_7、F_9 在 30 维各算法的收敛曲线对比图。从图 2 和图 3 可以看出,DVTLBO 在 F_1、F_2、F_3、F_4、F_7、F_9上相较于其他算法具有最快的收敛速度。图 2 和图 3 表明在大多数的函数下,DVTLBO 收敛速度快,可靠性好。

表 4　POA、PSO、COA、GWO、TLBO、DVTLBO 在 30 维函数上得到的平均值和标准差

Table 4　The mean and standard deviation of POA, PSO, COA, GWO, TLBO, DVTLBO on the 30-dimensional functions

函数	POA		PSO		COA		GWO		TLBO		DVTLBO	
	均值	标准差	均值	标准差	均值	标准差	均值	标准差	均值	标准差	均值	标准差
F_1	1.25×10^{-211}	0	3.27×10^{2}	2.19×10^{1}	3.01×10^{-1}	5.65×10^{-17}	1.22×10^{-49}	2.97×10^{-49}	1.40×10^{-189}	0	**0**	**0**
F_2	9.16×10^{-212}	0	3.29×10^{2}	2.25×10^{1}	3.01×10^{-1}	5.65×10^{-17}	1.22×10^{-49}	2.68×10^{-49}	1.47×10^{-189}	0	**0**	**0**
F_3	4.48×10^{-208}	0	5.14×10^{3}	3.90×10^{2}	4.66	9.03×10^{-16}	4.31×10^{-49}	8.99×10^{-49}	1.05×10^{189}	0	**0**	**0**
F_4	1.82×10^{-215}	0	1.40×10^{10}	1.71×10^{10}	3.48×10^{1}	2.17×10^{-14}	6.68×10^{-11}	3.31×10^{-10}	1.69×10^{-23}	6.36×10^{-23}	**0**	**0**
F_5	-8.33×10^{-1}	3.79×10^{-1}	1.34×10^{4}	9.25×10^{-12}	1.77	9.03×10^{-16}	2.72×10^{1}	7.36×10^{-1}	2.55×10^{1}	3.57×10^{-1}	**6.70**	**1.66×10^{1}**
F_6	-8.67×10^{-1}	3.46×10^{-1}	1.14×10^{1}	2.48×10^{-1}	1.88×10^{-1}	8.47×10^{-17}	2.16×10^{-14}	4.98×10^{-15}	5.06×10^{-15}	1.66×10^{-15}	**4.44×10^{-16}**	**0**
F_7	0	0	6.57×10^{2}	2.37×10^{1}	1.56×10^{-1}	0	5.71×10^{-1}	1.55	1.88×10^{1}	2.39×10^{1}	**0**	**0**
F_8	0	0	-7.33×10^{-1}	4.50×10^{-1}	0	0	0	0	0	0	**0**	**0**
F_9	0	0	1.08	5.90×10^{-3}	5.25×10^{-1}	1.13×10^{-16}	3.70×10^{-3}	8.70×10^{-3}	0	0	**0**	**0**
F_{10}	5.18×10^{3}	8.53×10^{2}	7.15×10^{3}	1.85×10^{-12}	7.15×10^{3}	1.85×10^{12}	6.57×10^{3}	7.27×10^{2}	4.82×10^{3}	1.00×10^{3}	**3.88×10^{3}**	**2.19×10^{3}**
F_{11}	**3.20×10^{-15}**	**1.08×10^{-15}**	**1.81×10^{1}**	**7.23×10^{-15}**	**3.88×10^{-2}**	**0**	**1.31×10^{-1}**	**2.92×10^{-1}**	**2.72×10^{-2}**	**4.82×10^{-2}**	0	0
F_{12}	0	0	9.57×10^{2}	4.63×10^{-13}	1.28×10^{-1}	0	6.33×10^{1}	3.76×10^{1}	9.97×10^{1}	6.36×10^{1}	**0**	**0**
F_{13}	0	0	0	0　0	0	0	0	0	0	0	**0**	**0**
F_{14}	0	0	1.35	3.32×10^{-2}	4.73×10^{-2}	0	2.70×10^{-2}	7.63×10^{-2}	0	0	**0**	**0**

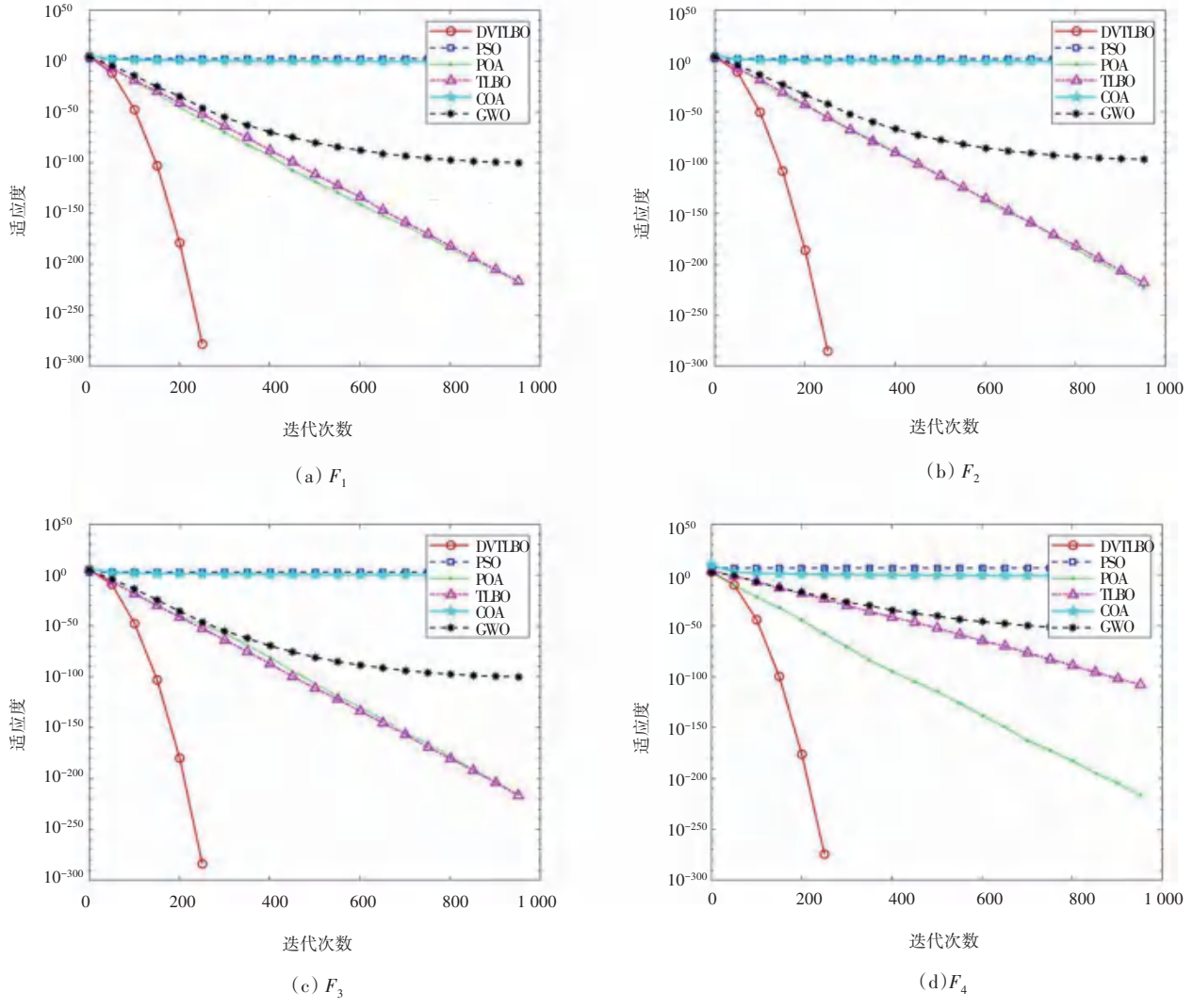


图 2　F_1~F_4 在 10 维的收敛曲线对比图

Fig. 2　Comparison of convergence curves of F_1~F_4 in 10 dimensions

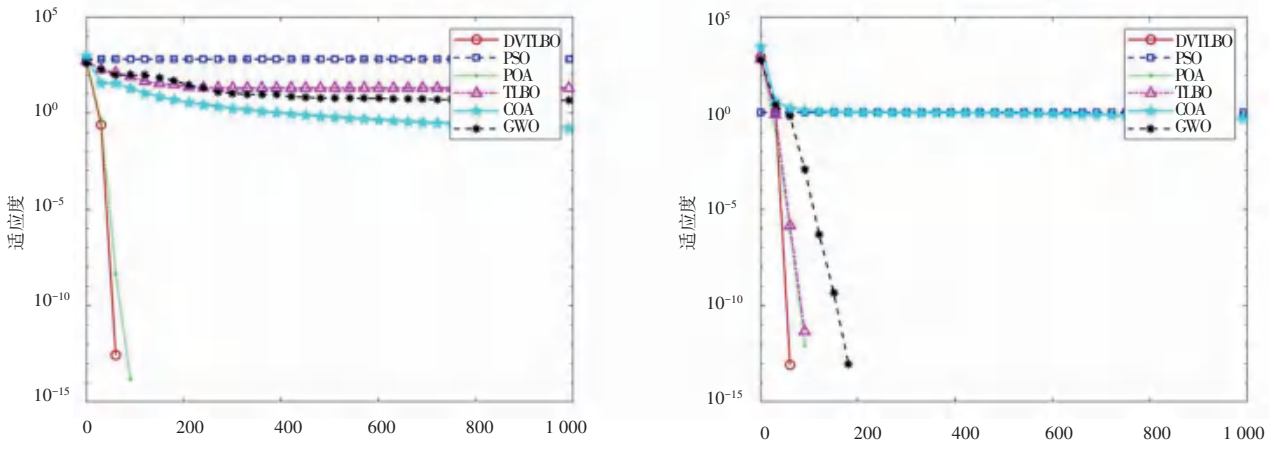


图 3 F_7、F_9 在 30 维的收敛曲线对比图

Fig. 3 Comparison of convergence curves of F_7 and F_9 in 30 dimensions

3.3 DVTLBO 与改进的算法比较

为了进一步验证 DVTLBO 的性能,将算法与其他改进教与学优化算法进行对比。ETLBO、sawTLBO、VTLBO 算法的实验结果来源于于文献[14]的前期工作。表 5 和表 6 分别记录了 DVTLBO 算法与 ETLBO、sawTLBO、VTLBO 算法在 10 维和 30 维的独立运行 30 次的均值和标准差。

根据表 5 实验数据显示,DVTLBO 算法在 F_1、F_2、F_3、F_4、F_6、F_7、F_9、F_{11}、F_{12}、F_{14} 这些函数上比其他所有算法的均值和标准差都要低,表现出最好的性能。ETLBO 算法则在 F_5 上表现出最好的性能,VTLBO 算法在 F_{10} 上表现出最好的性能。在 F_8 和 F_{13} 上 DVTLBO 算法和其他算法都能取得最优的均值和标准差。从表 6 可以看出,DVTLBO 算法在 F_1、F_2、F_3、F_4、F_5、F_6、F_7、F_{10}、F_{11}、F_{12} 均取得了最佳的均值和标准差,表现出最好的性能,在 F_8、F_9、F_{13}、F_{14} 上和其他改进的算法共同取得最优的均值和标准差。

表 5 ETLBO、sawTLBO、VTLBO、DVTLBO 在 10 维函数上得到的平均值和标准差

Table 5 The mean and standard deviation obtained by ETLBO, sawTLBO, VTLBO, DVTLBO on the 10-dimensional functions

函数	ETLBO		sawTLBO		VTLBO		DVTLBO	
	均值	标准差	均值	标准差	均值	标准差	均值	标准差
F_1	2.84×10^{-166}	4.27×10^{-167}	3.01×10^{-64}	4.86×10^{-64}	3.56×10^{-296}	0	**0**	**0**
F_2	3.22×10^{-79}	5.07×10^{-79}	4.59×10^{-50}	1.12×10^{-49}	3.50×10^{-130}	5.05×10^{-130}	**0**	**0**
F_3	6.50×10^{-169}	5.49×10^{-170}	4.86×10^{-67}	1.54×10^{-66}	4.63×10^{-298}	0	**0**	**0**
F_4	2.94×10^{-87}	3.10×10^{-87}	6.87×10^{-53}	1.61×10^{-52}	1.02×10^{-139}	3.21×10^{-139}	**0**	**0**
F_5	1.46×10^{-1}	1.38×10^{-1}	1.90	5.28×10^{-1}	**1.13**	**5.06×10^{-1}**	7.80×10^{1}	2.21
F_6	3.37×10^{-15}	1.05×10^{-15}	2.84×10^{-15}	1.50×10^{-15}	1.78×10^{-15}	1.87×10^{-15}	**4.44×10^{16}**	**0**
F_7	3.02	1.86	8.57	3.23	1.09	1.52	**0**	**0**
F_8	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
F_9	2.42×10^{-2}	3.69×10^{-2}	1.33×10^{-2}	2.22×10^{-2}	3.82×10^{-8}	1.21×10^{-7}	**0**	**0**
F_{10}	**7.03×10^{2}**	**1.81×10^{2}**	8.25×10^{2}	1.22×10^{2}	7.92×10^{2}	1.90×10^{2}	1.81×10^{3}	6.01×10^{2}
F_{11}	3.49×10^{-15}	3.27×10^{-16}	3.20×10^{-15}	1.12×10^{-15}	2.13×10^{-15}	1.83×10^{-15}	**0**	**0**
F_{12}	3.00	1.22	1.34×10^{1}	4.28	2.55	2.14	**0**	**0**
F_{13}	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
F_{14}	6.77×10^{-3}	5.15×10^{-3}	1.91×10^{-2}	2.92×10^{-2}	5.40×10^{-10}	1.70×10^{-9}	**0**	**0**

表 6　ETLBO、sawTLBO、VTLBO、DVTLBO 在 30 维函数上得到的平均值和标准差

Table 6　The mean and standard deviation obtained by ETLBO, sawTLBO, VTLBO, DVTLBO on the 30-dimensional functions

函数	ETLBO		sawTLBO		VTLBO		DVTLBO	
	均值	标准差	均值	标准差	均值	标准差	均值	标准差
F_1	2.66×10^{-95}	1.84×10^{-95}	2.61×10^{-65}	3.09×10^{-65}	4.85×10^{-158}	1.06×10^{-157}	**0**	**0**
F_2	3.42×10^{-22}	4.72×10^{-22}	1.68×10^{-26}	1.78×10^{-26}	1.24×10^{-32}	1.79×10^{-32}	**0**	**0**
F_3	8.21×10^{-96}	1.11×10^{-95}	6.13×10^{-66}	1.12×10^{-65}	2.50×10^{-158}	2.48×10^{-158}	**0**	**0**
F_4	1.92×10^{-11}	1.93×10^{-11}	1.23×10^{-8}	2.54×10^{-8}	8.06×10^{-18}	1.11×10^{-17}	**0**	**0**
F_5	2.38×10^{1}	8.57×10^{-1}	2.36×10^{1}	8.68×10^{-1}	2.28×10^{1}	4.23×10^{-1}	**6.70**	$\mathbf{1.66\times10^{1}}$
F_6	3.55×10^{-15}	0	3.55×10^{-15}	0	3.55×10^{-15}	0	$\mathbf{4.44\times10^{16}}$	**0**
F_7	1.22×10^{1}	9.07	2.36×10^{1}	6.04	1.25×10^{1}	7.27	**0**	**0**
F_8	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
F_9	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
F_{10}	4.55×10^{3}	7.81×10^{2}	5.26×10^{3}	6.41×10^{2}	4.70×10^{3}	4.22×10^{2}	$\mathbf{3.88\times10^{3}}$	$\mathbf{2.19\times10^{3}}$
F_{11}	3.55×10^{-15}	0	3.55×10^{-15}	0	3.55×10^{-15}	0	**0**	**0**
F_{12}	1.36×10^{1}	7.85	4.09×10^{1}	1.15×10^{1}	1.41×10^{1}	5.55	**0**	**0**
F_{13}	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
F_{14}	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

4　结束语

基于原始的教与学优化算法[20]，为了进一步优化算法的性能，提出了一种新的优化方法 DVTLBO。在 DVTLBO 算法中，在教学阶段和学习阶段引入教学效率因子和差分变异的思想，分别设计了新的种群个体更新机制，并在其中应用了惯性权值。通过与其他算法在基准函数的对比实验，验证 DVTLBO 的性能。实验结果表明，DVTLBO 具有较好的收敛速度和收敛精度，性能得到提高。

在未来的工作中，DVTLBO 的性能将进一步得到提高。设计多目标的 DVTLBO 算法，解决多目标优化问题。同时，进一步改进算法，用以解决约束优化问题。

参考文献

[1] Rao R V, Savsani V J, Vakharia D P. Teaching learning based optimization: A novel method for constrained mechanical design optimization problems[J]. Computer Aided Design, 2011, 43(3): 303-315.

[2] Zhou G, Zhou Y Q, Deng W, et al. Advances teaching learning based optimization algorithm: A comprehensive survey (ICIC2022)[J]. Neurocomputing, 2023, 561: 126898.

[3] Zeng Z B, Dong H, Xu Y L, et al. Teaching learning based optimization algorithm with dynamic neighborhood and crossover search mechanism for numerical optimization[J]. Applied Soft Computing, 2024, 154: 111332.

[4] Xing A S, Chen Y, Suo J Y, et al. Improving teaching learning based optimization algorithm with golden-sine and multipopulation for global optimization[J]. Mathematics and Computers in Simulation, 2024, 221: 94-134.

[5] Kundu T, Garg H. LSMA-TLBO: A hybrid SMA-TLBO algorithm with lévy flight based mutation for numerical optimization and engineering design problems[J]. Advances in Engineering Software, 2022, 172: 103185.

[6] Ji X Y, Hu Y E, Zhou J X, et al. An improved teaching learning based optimization algorithm and its application to a combinatorial optimization problem in foundry industry[J]. Applied Soft Computing, 2017, 57: 504-516.

[7] Chen D B, Lu R Q, Zou F, et al. Teaching learning based optimization with variable population scheme and its application for ANN and global optimization[J]. Neurocomputing, 2016, 173(3): 1096-1111.

[8] Abdel-basset M, Mohamed R, Chakrabortty R K, et al. An efficient teaching learning based optimization algorithm for parameters identification of photovoltaic models: Analysis and validations[J]. Energy Conversion and Management, 2021, 227: 113614.

[9] Xu Y L, Peng Y, Su X Y, et al. Improving teaching learning based optimization algorithm by a distance fitness learning strategy[J]. Knowledge-Based Systems, 2022, 257: 108271.

[10] Dong H, Xu Y L, Cao D, et al. An improved teaching learning based optimization algorithm with a modified learner phase and a new mutation restarting phase[J]. Knowledge-Based Systems, 2022, 258: 109989.

[11] Chen Z Y, Liu Y, Yang Z L, et al. An enhanced teaching learning based optimization algorithm with self adaptive and learning operators and its search bias towards origin[J]. Swarm and Evolutionary Computation, 2021, 60: 100766.

[12] Lotfipour A, Afrakhte H. A discrete teaching learning based optimization algorithm to solve distribution system reconfiguration in presence of distributed generation[J]. International Journal of Electrical Power & Energy Systems, 2016, 82: 264-273.

[13] Uzlu E, Kankal M, Akpinar A, et al. Estimates of energy consumption in Turkey using neural networks with the teaching learning based optimization algorithm[J]. Energy, 2014, 75: 295-303.

[14] 刘翱. 基于教与学优化的舰载机起飞出动动态调度[J]. 智能计算机与应用,2024,14(4): 45-51.

[15] Umbarkar A J, Sheth P M, Hong W C, et al. MapReduce teaching learning based optimization algorithm for solving CEC-2013 LSGO benchmark Testsuit[J]. Intelligent Systems with Applications, 2024, 24: 200460.

[16] 王乐遥,顾磊. 带扰动的自适应蜣螂优化算法[J]. 智能计算机与应用, 2026, 16(5): 40-48.

[17] Wa J, Li D B, Tang H T, et al. A dynamical teaching-learning-based optimization algorithm for fuzzy energy-efficient parallel batch processing machines scheduling in fabric dyeing process, Applied Soft Computing, 2024, 167: 112413.

[18] 袁世一. 基于粒子群优化 GRU 的粮食产量预测[J]. 智能计算机与应用, 2024, 14(8): 109-114.

[19] Ou Y J, Cai Yu, Zeng L J, et al. A teaching-learning-based optimization algorithm for reliability analysis with an adaptive penalty coefficient[J]. Structures, 2024, 65: 106695.

[20] Ma Y P, Zhang X X, SONG J C, et al. A modified teaching learning based optimization algorithm for solving optimization problem[J]. Knowledge-Based Systems, 2021, 212: 106599.