

DOI: 10.3969/j.issn.1671-0673. 2025. 06. 010

改进的指数三角优化算法及工程应用

王云峰, 冯 琪, 杨自强

(陆军防化学院, 北京 102205)

摘要: 针对指数三角优化算法(ETO)易陷入局部最优、收敛速度慢等问题, 提出一种多策略改进的指数三角优化算法(IETO)。IETO 算法在种群初始化阶段, 利用拉丁超立方体的均匀分布特性, 增强种群多样性; 探索 and 开发阶段, 引入自适应分段更新策略, 动态调整搜索过程中探索与开发的平衡; 采用多样性保持机制, 避免算法早熟收敛并跳出局部最优; 引入周期性退火局部搜索策略, 在开发阶段精细搜索, 从而提高最终解的质量。在求解常见工程应用问题中, 三杆桁架、压缩弹簧、容器压力设计问题所得结果均有良好表现, 在遍历寻优方面表现更出色, 具有更强的适用性和优越性。

关键词: 指数三角优化算法; 拉丁超立方体; 自适应更新; 多样性保持; 局部搜索

中图分类号: TP311.1

文献标识码: A

文章编号: 1671-0673(2025)06-0697-09

Improved Exponential-Trigonometric Optimization Algorithm and Its Engineering Application

WANG Yunfeng, FENG Qi, YANG Ziqiang

(Institute of NBC Defense, Beijing 102205, China)

Abstract: To address the problems of local optima and slow convergence speed in the exponential-trigonometric optimization algorithm (ETO), a multi-strategy improved exponential-trigonometric optimization algorithm (IETO) is proposed. In the population initialization stage of IETO algorithm, the uniform distribution characteristics of Latin hypercube sampling are utilized to enhance population diversity. During the exploration and development stages, an adaptive segmented update strategy is introduced to dynamically adjust the balance between exploration and development in the search process. A diversity maintenance mechanism is adopted to avoid premature convergence and escape local optima. A periodic annealing local search strategy is introduced to perform fine-grained searches during the development stage, thereby improving the quality of the final solution. In solving common engineering application problems such as the three-bar truss, compressed spring, and container pressure design problems, the results obtained all show good performance, with superior performance in traversing to find the optimal value, demonstrating stronger applicability and superiority.

Key words: exponential-trigonometric optimization algorithm; Latin hypercube; adaptive updating; diversity preservation; local search

随着科学技术进步, 优化问题广泛出现在生活中各个领域, 传统的基于数学的优化算法面对大规模、复杂性高的问题时, 求解精度和求解效率

已经很难满足实际需求^[1]。为此, 研究人员提出了一系列元启发优化算法。元启发算法是受自然界中的物理规律、物种进化以及生物种群行为等启

收稿日期: 2025-07-31; 修回日期: 2025-10-31

作者简介: 王云峰(1990—), 男, 工程师, 硕士, 主要研究方向为智能算法及应用。E-mail: wangyunfengie@126.com

发设计,具有执行效率高,全局搜索能力强等特点,如模拟灰狼捕食行为和社会等级的灰狼优化算法^[2](Grey Wolf Optimization, GWO),基于蜣螂滚球、觅食、繁殖的蜣螂算法^[3](Dung Beetle Optimizer, DBO),模拟白鲸游泳、捕食、鲸落的白鲸算法^[4](Beluga Whale Optimization, BWO),模拟麻雀觅食与反捕食行为的麻雀算法^[5](Sparrow Search Algorithm, SSA)以及粒子群算法^[6](Particle Swarm Optimization, PSO)等。

近年许多研究提出改进算法,金将等^[7]在蚁群算法的状态转移概率公式中引入动态目标导向函数,通过劣质蚂蚁的分层信息素更新个体位置;高纪元等^[8]在改进的蜣螂算法中,采用改进后的Logistic映射初始化种群,增加透镜成像学习策略增强算法的局部逃逸能力;杨佳等^[9]在改进蜻蜓算法中,引入变异算子和边界控制机制,增强算法全局寻优能力;张晶等^[10]提出融合正余弦优化的距离定位算法,利用正余弦的波动性寻找最优解,平衡算法前期、后期的搜索能力;徐浩等^[11]提出一种改进鲸鱼算法,引入余弦因子避免算法陷入局部最优。专家学者在研究中发现相较传统元启发算法,新型元启发优化算法如鱼鹰优化算法^[12]、黑猩猩搜索算法^[13]、改进的蜣螂优化算法^[14]、精英金豺优化算法^[15]、金鹰优化算法^[16]等,在函数寻优方面表现优异,但仍存在收敛速度较慢和容易陷入局部最优等不足,面对局部极值较多、变量维度较高复杂问题时尤为突出^[17]。

指数三角优化算法^[18](Exponential-Trigonometric Optimization, ETO)是一种新颖启发式优化方法,基于指数函数和三角函数的组合,合理调整代理位置,有效平衡探索与开发,其设计结构简单、易于实现,能够灵活实现快速寻优。基于以上思想,提出一种改进型指数三角优化算法(Improved Exponential-Trigonometric Optimization, IETO)。种群初始化阶段,利用拉丁超立方的均匀分布特性,增强种群多样性;探索和开发阶段引入自适应分段更新策略,动态调整搜索过程中探索与开发的平衡;采用多样性保持机制,避免算法早熟收敛并跳出局部最优;引入周期性退火局部搜索策略,在开发阶段精细搜索,从而提高最终解的质量。仿真对比 IETO 与 PSO、SSA、GWO、DBO、BWO、ETO 等算法性能,结果表明 IETO 算法收敛性更强,寻优精度更高,且引入的策略能够有效作用,全面提升算法的性能。应用在常见工程设计问题

的仿真更直观显示,IETO算法在求解三杆桁架、弹簧压力、容器压力设计问题所得适应度最优值优于其他算法,具有更强的适用性和优越性。

1 指数三角优化算法

ETO算法通过约束探索和自适应,切换探索与开发阶段,使算法在迭代中能够灵活跳出局部最优,提升全局搜索能力。

1.1 约束探索

ETO算法提出一种约束探索方法,促进空间探索和开发效率,利于优化搜索过程,减少所需时间和计算资源,同时确保不会错过最优解。 E_i 和 E_{i+1} 表示当前第 i 个及后续第 $(i+1)$ 个约束探索方法的迭代次数:

$$E_{i+1} = \text{floor}\left[2 - 2 \times t \times (I_{\text{iter_max}} - E_i \times a)\right] + E_i \quad (1)$$

$$E_i = \text{floor}\left(1 + \frac{I_{\text{iter_max}}}{b}\right) \quad (2)$$

式中: t 为当前迭代次数; $I_{\text{iter_max}}$ 为总迭代次数; $\text{floor}(\cdot)$ 为向下取整函数; a 、 b 为调整系数,分别取4.6和1.55。

在应用约束探索策略后,更新搜索空间的上限和下限,确保搜索空间能够动态适应,以提高优化过程效率。上限和下限的表达式为:

$$U_i = r_1 \times \left(1 - \frac{t}{I_{\text{iter_max}}}\right) \times |r_2 \times X_j^{\text{best}} - X_j^s| + X_j^{\text{best}} \quad (3)$$

$$L_i = -r_1 \times \left(1 - \frac{t}{I_{\text{iter_max}}}\right) \times |r_2 \times X_j^{\text{best}} - X_j^s| + X_j^{\text{best}} \quad (4)$$

式中: U_i 和 L_i 为搜索空间的上限和下限; r_1 、 r_2 为[0,1]的随机数; X_j^{best} 为最优解位置; X_j^s 为第 j 个索引处的次优解位置。此处添加次优解 X_j^s 来更新上下限,使搜索沿着次优向最优逐步扩展,目的是避免搜索过度依赖最优解 X_j^{best} ,造成搜索空间迅速以其为中心收缩,导致局部最优。

1.2 种群初始化

ETO算法使用常见的反向学习策略(Opposition-Based Learning, OBL)^[19]初始化种群,初始位置公式为

$$X_{ij} = (U_j - L_j) \times R_{\text{rand}} + L_j \quad (5)$$

根据式(6)计算迭代次数阈值 T :

$$T = \text{floor}\left(1.2 + \frac{I_{\text{iter_max}}}{2.25}\right) \quad (6)$$

当 $t \leq T$ 时,以全局探索为主,通过大范围随机扰动寻找潜在最优区域。当 $t > T$ 时,局部开发为主,围绕当前最优解搜索,从而通过阈值 T 实现探索和开发阶段切换。

1.3 第1探索阶段

优化过程早期,种群初始化且搜索空间的边界被确定后,个体开始探索搜索域,并逐渐移动到最佳位置。在第1探索阶段,个体位置公式为:

$$X_{t+1}^{ij} = \begin{cases} X_j^{\text{best}} + R_{\text{rand}} \times \alpha_1 \times |X_j^{\text{best}} - X_t^{ij}|, q_1 \leq 0.5; \\ X_j^{\text{best}} - R_{\text{rand}} \times \alpha_1 \times |X_j^{\text{best}} - X_t^{ij}|, q_1 > 0.5. \end{cases} \quad (7)$$

$$\alpha_1 = 3 \times R_{\text{rand}} \times \left(\frac{t}{I_{\text{iter_max}}} - 0.85 \right) \times \exp\left(\frac{d_1}{d_2} - 1\right) \quad (8)$$

式中: X_t^{ij} 和 X_{t+1}^{ij} 为当前和后续迭代中第 i 个解的第 j 个位置; q_1 和 R_{rand} 为 $[0,1]$ 的随机数; α_1 为权重系数,利用指数函数的衰减特性,使得探索逐渐趋近最优解。引入 d_1 、 d_2 变量目的是创建新的、高质量的搜索空间,随着迭代次数增加,搜索空间会逐渐缩小并趋于稳定,以提高收敛精度和速度。 d_1 、 d_2 取值范围为 $[-0.1,0.1]$,其表达式为:

$$d_1 = 0.1 \times e^{-0.01t} \times \cos\left[0.5 \times I_{\text{iter_max}} \times \left(1 - \frac{t}{I_{\text{iter_max}}}\right)\right] \quad (9)$$

$$d_2 = -0.1 \times e^{-0.01t} \times \cos\left[0.5 \times I_{\text{iter_max}} \times \left(1 - \frac{t}{I_{\text{iter_max}}}\right)\right] \quad (10)$$

1.4 第2探索阶段

第2探索阶段,个体移动向下一最优位置由当前最优解位置和当前解位置共同决定,其位置公式为:

$$X_{t+1}^{ij} = \begin{cases} X_t^{ij} + 3 \times |R_{\text{rand}} \times \alpha_2 \times X_j^{\text{best}} - X_t^{ij}|, q_2 \leq 0.5; \\ X_t^{ij} - 3 \times |R_{\text{rand}} \times \alpha_2 \times X_j^{\text{best}} - X_t^{ij}|, q_2 > 0.5. \end{cases} \quad (11)$$

$$\alpha_2 = R_{\text{rand}} \times \exp\left(\tanh\left(1.5 \times \left(\frac{t}{I_{\text{iter_max}}} - 0.75\right)\right) - R_{\text{rand}}\right) \quad (12)$$

式中: q_2 为 $[0,1]$ 的随机数; α_2 为影响第2探索阶段的权重因数。通过 α_2 控制移动幅度,由 $\tanh$ 函数平滑过渡,指数函数调节衰减速度。

1.5 第1开发阶段

为确保全面开发搜索域,开发过程分为两个阶段,每个阶段在整个迭代过程中展开。第1开发阶段,第 i 个解的第 j 个位置 X_{t+1}^{ij} 的位置公式为:

$$X_{t+1}^{ij} = \begin{cases} X_j^{\text{best}} + q_4 \times \alpha_3 \times |R_{\text{rand}} \times X_j^{\text{best}} - X_t^{ij}|, q_3 \leq 0.5; \\ X_j^{\text{best}} - q_4 \times \alpha_3 \times |R_{\text{rand}} \times X_j^{\text{best}} - X_t^{ij}|, q_3 > 0.5. \end{cases} \quad (13)$$

$$\alpha_3 = 3 \times R_{\text{rand}} \times \left(\frac{t}{I_{\text{iter_max}}} - 0.85 \right) \times \exp\left(\left|\frac{d_1}{d_2}\right| - 1.3\right) \quad (14)$$

式中: q_3 、 q_4 为 $[0,1]$ 的随机数; α_3 为第1开发阶段权重系数,用于控制个体在潜在最优解附近搜索域开发,并逐渐扩大其开发半径。 α_3 从较大数值逐渐增加,潜在解可以从近到远开发其周围域。

1.6 第2开发阶段

在第2开发阶段,ETO算法围绕目前最优解进行更密集的开发潜在解。随着迭代的进行,这种局部探索的强度增强,其位置公式为:

$$X_{t+1}^{ij} = X_t^{ij} + c \times |R_{\text{rand}} \times \alpha_2 \times X_j^{\text{best}} - X_t^{ij}| \quad (15)$$

$$c = \exp\left(\tan\left|\frac{d_1}{d_2}\right|\right) \quad (16)$$

1.7 阶段转换机制

为了能够在探索和开发阶段之间灵活切换,采用一种高效的转换技术,由式(17)实现:

$$M = 0.01 \times R_{\text{rand}} \times \left(\frac{t}{I_{\text{iter_max}}} \right)^{\frac{1}{2} \tan\left(\frac{d_1}{d_2}\right)} \quad (17)$$

在初始阶段,这种转换机制主要用于探索,但也进行少量开发。随着迭代增加,转换机制逐渐集中在开发,但仍保持少量探索。当 $M > 1$ 时,ETO算法处于探索模式;当 $M < 1$ 时,ETO算法切换到开发模式。确保平衡整个搜索域的探索和开发,同时在后续迭代中跳出局部最优。

2 改进指数三角优化算法

2.1 拉丁超立方体初始化

拉丁超立方体初始化是一种高效的多维分层抽样技术。相较于传统的混沌映射、佳点集映射等初始化策略,其基于分层抽样原理,实现非重叠、变量范围全覆盖采样,兼顾种群初始化的随机性和均匀性。通过确保每个变量的取值区间均被代表性采样,增强算法对全局最优区域的探索,适用于高维、非线性及计算代价高昂的黑箱函数优化问题^[20-21]。

将取值范围为 $[v, h]$ 的维度 j 划分为 R 等份,每个子区间长度为 $\Delta j = \frac{v-h}{R}$ 。第 i 个样本在第 j 维的采样点计算公式为

$$X_{ij} = a + (s_{ij} - u_{ij}) \times \Delta j \quad (18)$$

式中: s_{ij} 为第 i 个样本在第 j 维的子区间随机编号; $u_{ij} \sim U(0,1)$ 为均匀分布随机数。各维度的采样点随机组合,形成最终的多维样本矩阵,每个子区间仅含1个样本点,且样本点均匀覆盖整个分布空间的拉丁方阵结构。

2.2 自适应分段更新策略

ETO算法在探索和开发阶段,虽然新位置更新倾向于向当前最优解靠近,但仍然保留了一定的随机性。因此,引入自适应因子 β ,如式(19)所示:

$$\beta = \exp\left(-\frac{t}{I_{\text{iter_max}}}\right) \quad (19)$$

β 随迭代次数增加而变少,根据当前迭代次数 t 与最大迭代次数 $I_{\text{iter_max}}$ 比例决定更新方式。 β 用于动态调整搜索过程中的探索与开发行为,有助于在ETO算法早期进行广泛地探索,而在后期专注于更精细地局部搜索,提高算法找到全局最优解的能力。优化迭代过程中,动态更新 β ,种群不断向最优解靠近,从而提升算法的寻优精度,加快算法收敛速度。

继而,第1、2探索和第1、2开发阶段位置公式分别更新为:

$$X_{t+1}^{ij} = \begin{cases} X_j^{\text{best}} + R_{\text{rand}} \times \alpha_1 \times \beta \times |X_j^{\text{best}} - X_t^{ij}|, q_1 \leq 0.5; \\ X_j^{\text{best}} - R_{\text{rand}} \times \alpha_1 \times \beta \times |X_j^{\text{best}} - X_t^{ij}|, q_1 > 0.5. \end{cases} \quad (20)$$

$$X_{t+1}^{ij} = \begin{cases} X_t^{ij} + 3 \times |R_{\text{rand}} \times \alpha_2 \times \beta \times X_j^{\text{best}} - X_t^{ij}|, q_2 \leq 0.5; \\ X_t^{ij} - 3 \times |R_{\text{rand}} \times \alpha_2 \times \beta \times X_j^{\text{best}} - X_t^{ij}|, q_2 > 0.5. \end{cases} \quad (21)$$

$$X_{t+1}^{ij} = \begin{cases} X_j^{\text{best}} + q_4 \times \alpha_3 \times \beta \times |R_{\text{rand}} \times X_j^{\text{best}} - X_t^{ij}|, q_3 \leq 0.5; \\ X_j^{\text{best}} - q_4 \times \alpha_3 \times \beta \times |R_{\text{rand}} \times X_j^{\text{best}} - X_t^{ij}|, q_3 > 0.5. \end{cases} \quad (22)$$

$$X_{t+1}^{ij} = X_t^{ij} + c \times |R_{\text{rand}} \times \alpha_2 \times \beta \times X_j^{\text{best}} - X_t^{ij}| \quad (23)$$

2.3 多样性保持机制

当种群中个体过于相似时,ETO算法可能会陷入局部最优解而无法继续有效地探索更优解。为防止早熟收敛,引入多样性保持机制,计算个体间平均距离 d ,并与多样性阈值 Q 比较:

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \quad (24)$$

$$X_{ij} = (U_j - L_j) \times R_{\text{lhsdesign}} + L_j, d \leq Q \quad (25)$$

式中:平均距离 d 越小,表示种群内个体间越相似,反之则表明种群具有较高的多样性; $R_{\text{lhsdesign}}$ 表示拉丁超立方抽样过程,取值范围为[0,1];归一化空间下的多样性阈值通常设为[0.01,0.1]^[22],本文取 $Q=0.05$ 。当 $d \leq Q$ 时,对30%的最相似个体重新拉丁超立方体初始化^[23],可打破个体间同质化趋势,有效恢复种群多样性。

多样性保持在算法中较为关键,用于防止种群过早收敛到局部最优解,并维持种群的探索能力。该机制通过监测种群的相似度并采取措施增加个体间的差异性来实现,确保算法整体性能不下降,从而提高找到全局最优解概率,增强算法的适应性和可靠性。

2.4 周期性退火局部搜索策略

为增强局部搜索能力,基于模拟退火思想,允许接受比当前解稍差的新解,采用每隔10代执行1

次周期性模拟退火机制。周期性局部搜索(如每5~15代执行1次)是平衡全局探索与局部开发的常用策略,避免频繁调用导致算法陷入局部优化^[24],如式(26)和式(27)所示:

$$p = \exp\left(\frac{1}{t}(f - f_{\text{new}})\right) \quad (26)$$

$$X_{t+1}^{ij} = X_{\text{new}}, \text{mod}(t, 10) = 0, p > R_{\text{rand}} \quad (27)$$

式中: f, f_{new} 为当前解和新生成的解; p 为接受新解的概率。随着迭代次数 t 的增加,接受较差解的概率逐渐减小,即模拟退火中的“冷却”过程^[25]。算法初始阶段,由于接受较差解的概率较高,算法能够在更大范围内探索,有助于跳出局部最优;后期阶段,随着 t 逐渐增加,接受较差解的概率下降,算法更专注于寻找当前区域内的最优解。

2.5 IETO算法流程

IETO算法流程图如图1所示。

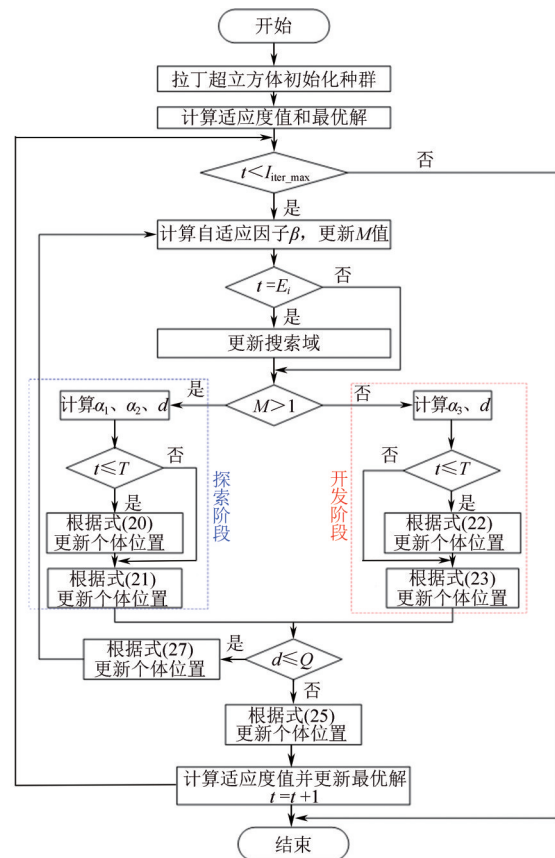


图1 IETO算法流程图

3 改进指数三角优化算法性能

3.1 定性分析

使用CEC2022基准测试套件的12维函数评估算法性能,通过搜索历史、平均适应度、第1维度轨迹和收敛曲线这4个指标直观展示IETO算法性能,其部分结果如图2所示。

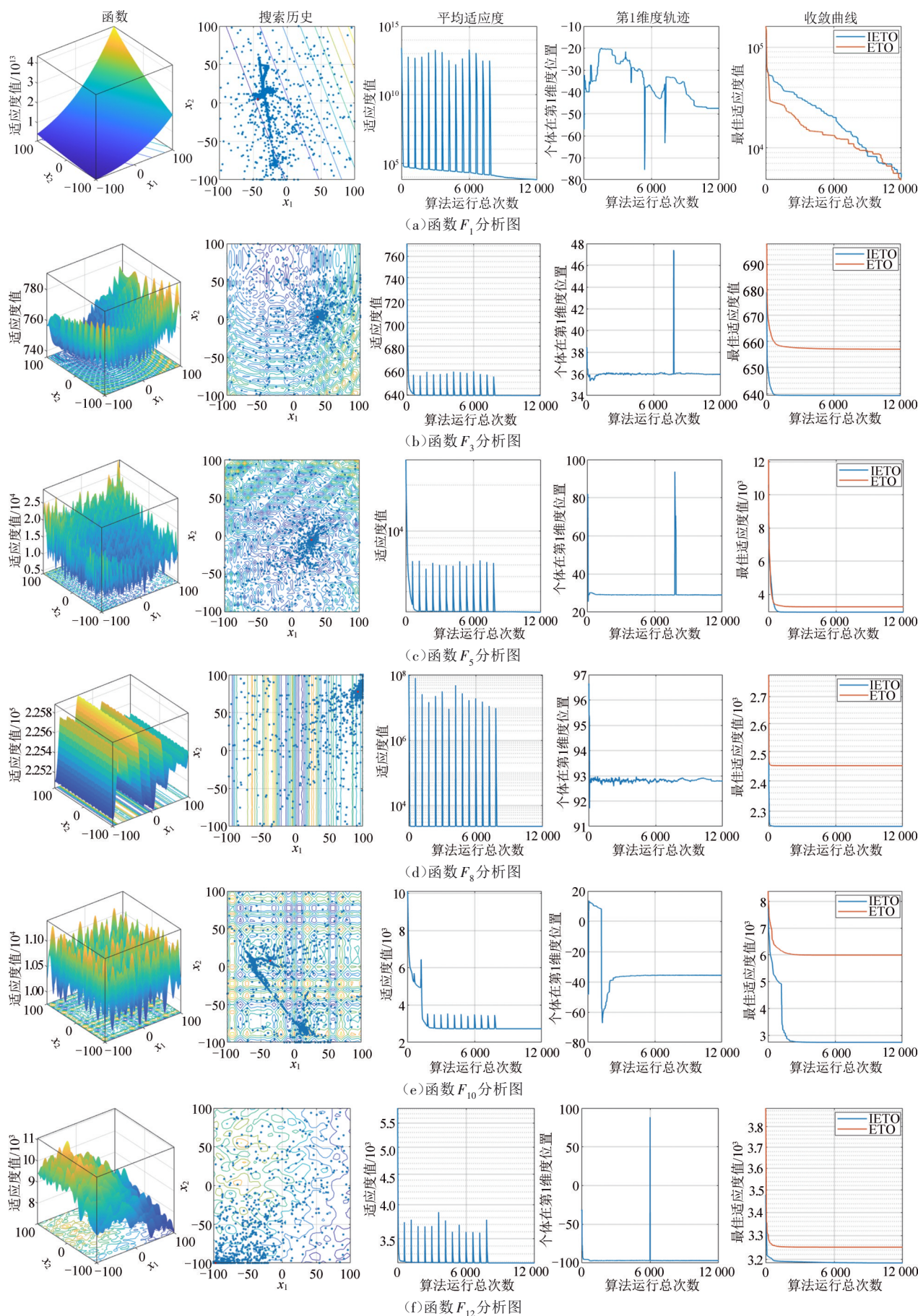


图2 IETO算法在CEC2022基准测试函数上的部分对比结果

图2中: F_1 、 F_3 表示混合函数; F_5 、 F_8 、 F_{10} 表示复合函数; F_{12} 表示简单函数。搜索历史用于可视化 IETO 算法在迭代过程中种群个体的分布与演化过程,直观反映了算法的种群多样性与搜索广度。平均适应度指标记录整个种群在迭代过程中的平均性能变化,结合最优适应度可用于评估种群的收敛趋势与多样性保持能力。第1维度轨迹展示最优个体在第1维度上的位置演化,有助于观察其收敛行为。收敛曲线则描绘了每代最优解的目标函数值变化,是衡量算法收敛速度与精度的核心指标。

1)从搜索历史可以看出,个体位置点较分散,但呈现有方向性的集中,说明 IETO 算法在进行有效的探索,寻找更优的解决方案,表明算法既保持了全局探索的能力,也展示了局部逐步逼近最优的趋势。

2)从平均适应度可以看出,每隔10代的周期性模拟退火局部搜索机制,能够使算法跳出局部最优,重新搜索空间,再次寻找最优解,继而使适应度值持续下降逼近最优。

3)从第1维度轨迹可以看出,出现短暂峰值并立刻下降现象,这是种群多样性保持机制起到了作用。当个体间平均距离 $d<$ 阈值 Q (仿真取0.05)时,即个体相似度较高,IETO算法重新拉丁超立方体初始化30%个体,生成的新个体可能瞬间跳到搜索空间的边界,造成轨迹图上的峰值,随后这些差解被算法快速修正,轨迹立刻降下来,使算法能够持续跳出局部最优,不断搜寻最优适应度。

4)从收敛曲线可以看出,IETO算法仅在测试简单函数 F_1 时,适应度值逊于ETO算法。测试 F_3 、 F_5 、 F_8 、 F_{10} 、 F_{12} 时,IETO算法收敛速度显著快于ETO算法,且寻优精度高,能快速趋近适应度理论值。

综上,IETO算法能够高效搜索空间,其自适应分段更新策略、多样性保持机制、周期性退火局部搜索策略有效,能够充分平衡探索与开发阶段,促使算法跳出局部最优。随着迭代次数增加,每次算法运行的最优解逐渐减小,趋近理论值,IETO算法收敛速度和适应度值均优于ETO算法。

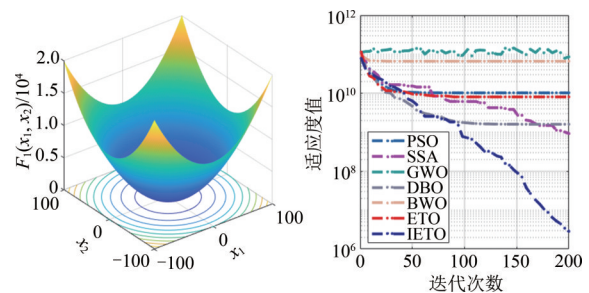
3.2 对比分析

为验证算法的普适性,全面测试 IETO 算法的泛化能力,使用 CEC2017 基准函数对比测试算法性能,测试套件包括2个单峰函数(F_1 、 F_3)、6个简单多峰函数($F_4\sim F_{10}$)、10个混合函数($F_{11}\sim F_{20}$)和11个组合函数($F_{21}\sim F_{30}$)^[26]。

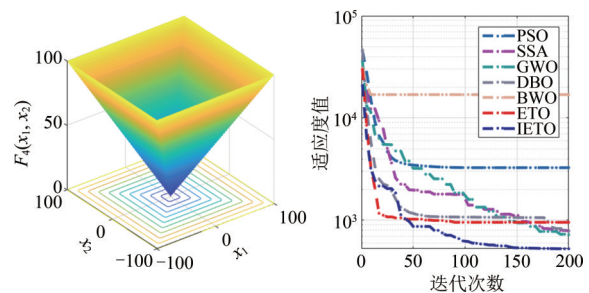
实验仿真选择具有代表性的6个函数 F_1 、 F_4 、 F_5 、 F_{11} 、 F_{16} 、 F_{22} ,对比测试 PSO、SSA、GWO、DBO、BWO、ETO 和 IETO 算法。其中,每个算法运行30次取平均值,最大迭代次数设置为200次,种群设置为50,维度大小为30,其他参数设置如表1所示。对比结果如图3所示。

表1 各算法参数设置

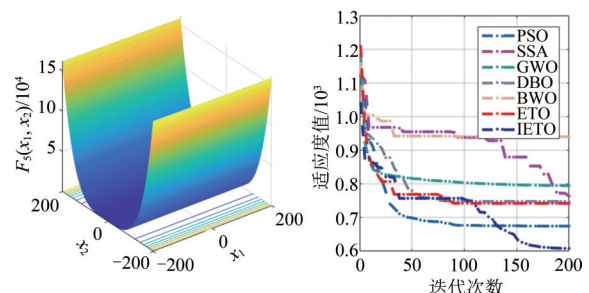
算法	描述
PSO	全局和个人学习因子 C_1 、 C_2 为1;惯性因子 $\omega_{\min}$ 、 $\omega_{\max}$ 分别设置为0.2、0.5。
SSA	发现者阈值为0.8;捕食麻雀占比70%、警觉麻雀占比20%。
GWO	参数 $C\in[0,2]$ 随机数; A 取1; L 为计数器; T 为迭代次数; $a=2-L\times(2/T)$,在 $[0,2]$ 区间线性递减。
DBO	滚球行为个体占比20%;挠度系数 $K=0.1$; $b=0.3$;自然系数 $\alpha=\pm 1$ 。
BWO	r_1 、 r_2 、 r_3 、 r_4 为 $[0,1]$ 的随机数。
ETO	约束探索: $a=4.6$, $b=1.55$ 。
IETO	约束探索: $a=4.6$, $b=1.55$;多样性保持机制,阈值 $Q=0.05$ 。



(a)函数 F_1 收敛曲线



(b)函数 F_4 收敛曲线



(c)函数 F_5 收敛曲线

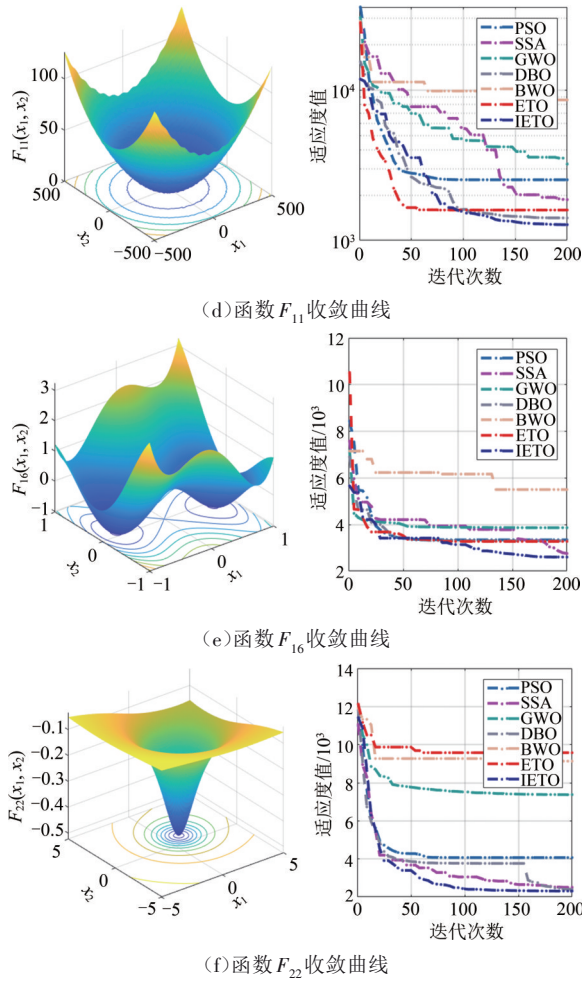


图3 IETO算法在CEC2017基准测试函数上的部分对比结果

从图3可以看出,IETO算法在单峰函数、简单多峰函数、混合函数、组合函数均有良好表现,较其他6种算法,均具有最优的适应度值,验证了其在不同类型问题下的广泛适用性。收敛性方面,IETO算法测试 F_1 时,收敛最快,体现了其高效的局部开发能力。测试 F_4 、 F_5 、 F_{11} 、 F_{16} 、 F_{22} 多峰和复杂混合函数时,在迭代初期,其收敛性逊于ETO等算法,这是由于IETO算法主动维持较高的种群多样性,避免过早陷入局部最优。当趋于第100次迭代时,周期性退火局部搜索策略起作用,对当前最优解进行精细优化,同时种群多样性保持策略有效引导搜索跳出局部陷阱,使IETO算法加快收敛,最终适应度值优于其他算法。

3.3 计算复杂度

计算复杂度方面,ETO算法未提及其复杂度。根据ETO算法流程,初始化阶段,生成种群数量 N , D 维变量,复杂度为 $O(N \times D)$;经过 T (即 $I_{\text{iter_max}}$)次迭代的适应度值计算,复杂度为 $O(T \times N \times D)$;每轮迭代更新 N 个种群最优解,复杂度为 $O(T \times N)$;开发阶段,种群位置更新, N 个种群、 D 维、更新 T 轮,复杂

度为 $O(T \times N \times D)$;阶段转换机制,涉及随机数等普通运算,复杂度为 $O(1)$,可忽略不计。ETO算法总计算复杂度为 $O(T \times N \times D)$ 。

IETO算法在ETO算法基础上,增加了每10代退火搜索,复杂度为 $O(N \times D/10)$,可简化为 $O(N \times D)$,拉丁超立方初始化和多样性保持机制复杂度均为 $O(N \times D)$,在此基础上迭代 T 次,总计算复杂度仍为 $O(T \times N \times D)$,与ETO算法处于同一量级。但通过引入多样性保持机制、周期性退火局部搜索策略,有效增强了算法局部开发与全局探索的能力,显著加快了算法的收敛速度,提高了求解精度,且未引入额外的高阶计算开销,体现了IETO算法在效率与性能之间的良好平衡。

4 改进指数三角优化算法在工程设计中的应用

4.1 三杆桁架设计

三杆桁架是一种常见的结构形式,其设计结构如图4所示。

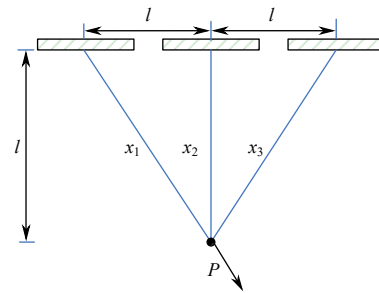


图4 三杆桁架设计示意图

3个杆的横截面积分别用变量 x_1 、 x_2 和 x_3 表示,由对称性可知 $x_1 = x_3$ 。三杆桁架设计的目的是通过调整横截面积 x_1 、 x_2 使三杆桁架的体积最小^[27],每个桁架构件均受到应力 σ 的约束,优化设计求解一个非线性适应度函数。

为将该约束优化问题转化为无约束优化问题,便于与优化算法结合求解,采用外点法(外罚函数法)来处理约束条件。式(28)为目标函数,式(29)~式(31)为约束条件,其数学模型如下:

$$\min f(x) = (2\sqrt{2}x_1 + x_2)l \quad (28)$$

$$g_1(x) = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2} \cdot P - \sigma \leq 0 \quad (29)$$

$$g_2(x) = \frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2} \cdot P - \sigma \leq 0 \quad (30)$$

$$g_3(x) = \frac{1}{\sqrt{2}x_2 + x_1} \cdot P - \sigma \leq 0 \quad (31)$$

式中： $0.001 \leq x_1, x_2 \leq 1$ ； $l=100$ cm； $P=2$ kN/cm²； $\sigma=2$ kN/cm²。

将 PSO、GWO、DBO、BWO、ETO 和 IETO 这 6 种算法应用于求解桁架最小体积，种群 $N=50$ ，迭代次数 $T=100$ ，每个算法运行 20 次。第 20 次运行时，6 种算法适应度值随迭代次数增加变化的比较如图 5 所示，实验结果如表 2 所示。

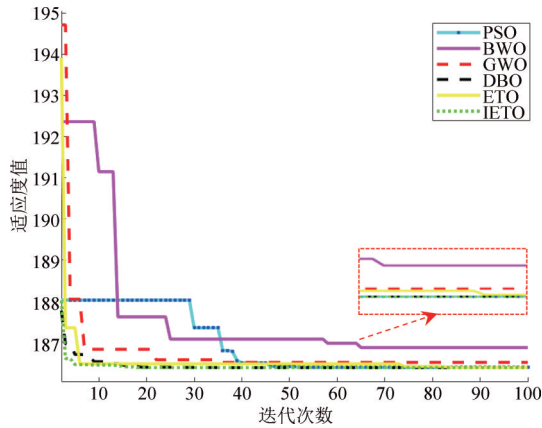


图5 三杆桁架设计问题应用中6种算法的比较

表2 三杆桁架设计问题实验结果

算法	最优值	平均值	最差值	标准差
PSO	186.385 9	186.624 8	189.453 1	3.002 700 0
GWO	186.388 9	187.787 3	194.665 2	1.462 300 0
DBO	186.385 9	187.106 4	188.733 4	0.000 740 2
BWO	186.886 2	186.387 1	192.428 6	4.282 000 0
ETO	186.388 1	186.934 2	193.868 5	1.353×10^{-6}
IETO	186.385 9	186.386 2	186.388 2	7.686×10^{-9}

应用 IETO 算法求解时，前期收敛速度快，迭代 10 次以内即取到接近最优适应度值，且后期适应度最优值、平均值均优于其他算法。

4.2 压缩弹簧设计

压缩弹簧设计如图 6 所示。

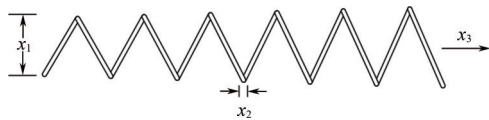


图6 压缩弹簧设计示意图

该问题的目标是 minimized 弹簧的重量，即最优优化问题，其中包含最小挠度、剪切应力、振荡频率以及外径限制 4 个不等式约束^[28]。

弹簧重量由 3 个变量确定：弹簧线径 x_1 、平均线圈直径 x_2 和有效线圈数 x_3 。式(32)为目标函数，式(33)~式(36)为约束条件，其数学模型为：

$$\min f(x) = x_1^2 x_2 x_3 + 2x_1^2 x_2 \quad (32)$$

$$g_1(x) = 1 - \frac{x_2^2 x_3}{71\,785 x_1^4} \leq 0 \quad (33)$$

$$g_2(x) = \frac{4x_2^2 - x_1 x_2}{12\,566(x_1^3 x_2 - x_1^4)} + \frac{1}{5\,108 x_1^2} \leq 0 \quad (34)$$

$$g_3(x) = 1 - \frac{140.45 x_1}{x_2^3 x_3} \leq 0 \quad (35)$$

$$g_4(x) = \frac{x_1 + x_2}{1.5} - 1 \leq 0 \quad (36)$$

式中： $0.05 \leq x_1 \leq 2$ ； $0.25 \leq x_2 \leq 1.3$ ； $2 \leq x_3 \leq 15$ 。

将 PSO、GWO、DBO、BWO、ETO 和 IETO 这 6 种算法应用于求压缩弹簧的最小重量，种群 $N=50$ ，迭代次数 $T=100$ ，每个算法运行 20 次。第 20 次运行时，6 种算法适应度值随迭代次数增加变化的比较如图 7 所示，实验结果如表 3 所示。

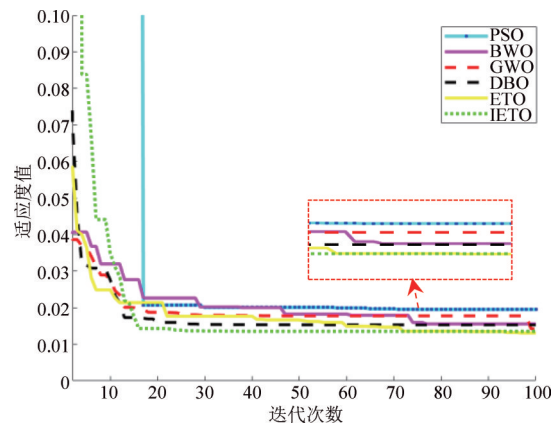


图7 压缩弹簧设计问题应用中6种算法的比较

表3 压缩弹簧设计问题实验结果

算法	最优值	平均值	最差值	标准差
PSO	0.012 821	0.018 264	0.049 857	0.009 597 5
GWO	0.012 785	0.014 702	0.017 769	0.001 917 8
DBO	0.012 688	0.013 624	0.015 559	0.000 810 2
BWO	0.012 695	0.013 669	0.017 769	0.001 607 8
ETO	0.012 717	0.013 534	0.017 762	0.001 209 1
IETO	0.012 713	0.012 983	0.017 748	0.000 370 6

应用 IETO 算法求解时，迭代初期算法收敛速度快，算法很快寻找到适应度最优值并保持，IETO 算法适应度最优值优于其他算法。

4.3 压力容器设计

压力容器设计如图 8 所示。压力容器设计有 4 个决策变量^[29]：半径 x_1 、无头圆柱长度 x_2 、壳体厚度 x_3 、封头厚度 x_4 。式(37)为目标函数，式(38)~式(41)为约束条件，其数学模型为：

$$\min f(x) = 0.622\,4x_1 x_3 x_4 + 1.778\,1x_2 x_3^2 + 3.166\,1x_1^2 x_4 + 19.84x_1^2 x_3 \quad (37)$$

$$g_1(x) = -x_1 + 0.019\,3x_3 \leq 0 \quad (38)$$

$$g_2(x) = -x_2 + 0.009\,54x_3 \leq 0 \quad (39)$$

$$g_3(x) = -\pi x_3^2 x_4 - 4/3\pi x_3^3 + 1\,296\,000 \leq 0 \quad (40)$$

$$g_4(x) = x_4 - 240 \leq 0 \quad (41)$$

式中: $0 \leq x_1, x_2 \leq 100; 10 \leq x_3, x_4 \leq 200$ 。

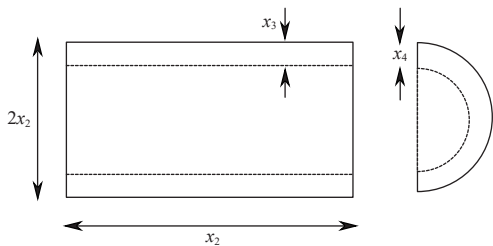


图8 压力容器设计示意图

将 PSO、GWO、DBO、BWO、ETO 和 IETO 这 6 种算法应用于求解容器最小成本,种群 $N=50$,迭代次数 $T=100$,每个算法运行 20 次。第 20 次运行时,6 种算法适应度值随迭代次数增加变化的比较如图 9 所示,实验结果如表 4 所示,应用 IETO 算法求解结果明显优于其他算法。

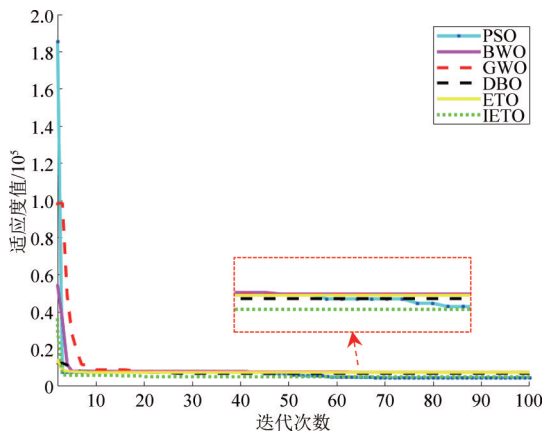


图9 压力容器设计问题应用中6种算法的比较

表4 压力容器设计问题实验结果

算法	最优值	平均值	最差值	标准差
PSO	4 527.366 0	4 679.156 3	19 123.447 1	3 055.805 4
GWO	4 534.295 3	7 093.928 6	9 839.448 3	957.156 8
DBO	4 529.348 6	5 658.396 8	7 535.014 1	962.381 1
BWO	4 536.336 2	6 802.843 8	7 535.188 1	1 138.398 1
ETO	4 530.465 4	5 510.632 2	7 565.011 8	1 082.306 5
IETO	4 527.268 0	4 832.059 9	5 575.359 6	343.343 8

5 结束语

提出一种改进的指数三角优化算法,在种群初始化阶段,利用拉丁超立方的均匀分布特性,增强种群多样性。采用自适应分段更新策略和多样性保持机制,动态调整搜索过程中探索与开发的平衡,避免算法早熟收敛并跳出局部最优。引入周期性退火局部搜索策略,提高最终解的质量。在求解常见工程应用问题中,如三杆桁架、压缩弹簧、容器压力设计问题,应用 IETO 算法所得结果均

有良好表现,在遍历寻优方面表现更出色,具有更强的适用性和优越性。

下步工作将聚焦提升 IETO 算法在高维复杂问题中的性能,同时减少算法单次迭代的计算开销,探索其在多目标优化场景下的扩展应用,进一步增强算法的自适应能力和计算效率。

参考文献:

[1] 匡鑫,阳波,马华,等.多策略改进的蜣螂优化算法[J].计算机工程,2024,50(10):119-136.

[2] MIRJALILI S, MIRJALILI S M, LEWIS A. Grey wolf optimizer[J]. Advances in Engineering Software, 2014, 69: 46-61.

[3] XUE J K, SHEN B. Dung beetle optimizer: a new meta-heuristic algorithm for global optimization[J]. The Journal of Supercomputing, 2023, 79(7): 7305-7336.

[4] ZHONG C T, LI G, MENG Z. Beluga whale optimization: a novel nature-inspired metaheuristic algorithm[J]. Knowledge-Based Systems, 2022, 251: No.109215.

[5] KENNEDY J, EBERHART R. Particle swarm optimization[C]//Proceedings of ICNN'95 - International Conference on Neural Networks. Piscataway, USA: IEEE, 1995: 1942-1948.

[6] XUE J K, SHEN B. A novel swarm intelligence optimization approach: sparrow search algorithm[J]. Systems Science & Control Engineering, 2020, 8(1): 22-34.

[7] 金将,王小平,臧铁钢,等.基于改进蚁群算法的机器人避障路径规划[J].计算机工程与设计,2025,46(4): 950-958.

[8] 高纪元,刘杰,陈昌盛,等.混合策略改进的蜣螂优化算法[J].计算机工程与科学,2025,47(1): 171-179.

[9] 杨佳,汤嘉乐,田朋,等.基于改进蜻蜓算法的WSNs分簇路由算法[J].计算机工程与设计,2025,46(6): 1625-1631.

[10] 张晶,贺媛媛.融合正余弦优化与跳距优化的DV-Hop定位算法[J].计算机工程与科学,2022,44(4): 645-653.

[11] 徐浩,吴宇航.改进鲸鱼算法对线激光条纹识别模型的优化[J].计算机仿真,2025,42(6): 356-362.

[12] 李奕轩,田云娜,王凯欣.多策略改进的鱼鹰优化算法及其应用[J].计算机技术与发展,2025,35(1): 132-139.

[13] 张福兴,高腾,吴泓达,等.多策略融合的改进黑猩猩优化算法[J].北京航空航天大学学报,2025,51(1): 235-247.

[14] 姜革伟,郑永煌,陈均,等.混合多策略改进的蜣螂优化算法[J].计算机工程与应用,2024,60(24): 97-109.

[15] 吴智祥,刘杰,覃涛,等.多策略改进的精英金豺优化算法[J/OL].计算机工程与科学,2024-06-28.<https://kns.cnki.net/kcms/detail/43.1258.TP.20240627.1446.004.html>.

(下接第 739 页)