

面向夹层结构柔顺度优化的代理模型辅助 差分进化算法

杨 赞^{1,2} 朱紫华¹ 孙观观¹ 邱浩波^{3*} 高 亮³

1. 南昌大学先进制造学院, 南昌, 330031

2. 泰豪科技股份有限公司, 南昌, 330096

3. 华中科技大学机械科学与工程学院, 武汉, 430074

摘要: 夹层结构因其高刚度-质量比特性而被广泛应用于航空航天等领域, 其优化设计时的柔顺度仿真分析计算成本显著高于重量约束, 但现有同类型方法假设目标与约束评估成本等价而导致优化适应性差、效率低下。为此, 设计了基于可行率划分的约束-目标两阶段优化架构, 为实时优化轨迹匹配适应性优化方向。第一阶段提出探索型变异-约束松弛筛选及开发型变异-不确定度筛选的双子种群协同优化策略, 从而同步提高约束优化水平与代理模型可靠性, 并设计了部分评估策略以节省高耗时目标评估; 第二阶段结合可行解聚类分析与动态阈值划定搜索类型, 自适应调整代理模型建模与进化策略。在3种经典载荷下, 该算法相较于梯度算法及其他最先进的同类型算法均获得最优结构, 证实了其在实际应用中的有效性。

关键词: 夹层结构; 昂贵优化; 代理模型; 差分进化算法

中图分类号: TP301

DOI:10.3969/j.issn.1004-132X.2026.03.011

开放科学(资源服务)标识码(OSID):



Surrogate-assisted Differential Evolution Algorithm for Compliance Optimization of Sandwich Structures

YANG Zan^{1,2} ZHU Zihua¹ SUN Guanguan¹ QIU Haobo^{3*} GAO Liang³

1.School of Advanced Manufacturing, Nanchang University, Nanchang, 330031

2.Tellhow Sci-Tech Co., Ltd., Nanchang, 330096

3.School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan, 430074

Abstract: Sandwich structures were widely used in aerospace and other fields due to their high stiffness-to-weight ratio characteristics. The calculation costs of compliance simulation analyses in optimization design processes were significantly higher than that of weight constraints. However, the existing homogeneous algorithms assumed that the objectives were equivalent to the costs of constraint evaluation, which led to poor optimization adaptability and low efficiency. Thus, a constraint-objective two-stage optimization framework was designed based on feasibility rate to match adaptive optimization direction for real-time optimization paths. In the first stage, a dual offspring population collaborative optimization strategy of exploratory mutation-constraint relaxation screening and exploitative mutation-uncertainty screening was proposed to simultaneously enhance the level of constraint optimization and the reliability of the surrogate model, and the partial evaluation strategy was designed to save time-consuming objective evaluation. In the second stage, the search type was defined by combining feasible solution clustering analyses and dynamic threshold, and the surrogate model modeling and evolution strategy were adjusted adaptively. Under three classical loads, the proposed algorithm obtains optimal structures comparing with the gradient algorithm and other state-of-the-art algorithms of the same type, which confirms the effectiveness in practical applications.

Key words: sandwich structure; expensive optimization; surrogate model; differential evolution algorithm

0 引言

夹层结构是由两个面板和一个芯材组成的复

合结构, 通常由蜂窝或泡沫材料制成^[1]。该结构凭借其显著的高刚度-质量比特性使其在航空航天、交通运输等对轻量化要求严苛的领域具有突出优势。

为了获得高刚度夹层结构, 可通过优化芯材

的材料分布来实现。当前,拓扑优化已成为解决夹层结构优化设计问题的有效手段之一。传统的隐式拓扑优化算法主要分为两类:密度法和边界法。密度法的典型方法包括渐进结构优化(evolutionary structural optimization, ESO)^[2]方法和固体各向同性材料惩罚(solid isotropic material with penalization, SIMP)^[3]方法,而边界法则以水平集法^[4]为代表。这两类方法都需要将设计域离散化为有限元网格,但分别采用不同的优化策略。在密度法中,材料密度变量用于表征设计域内的材料分布。该方法通过优化网格单元上的密度值,实现材料的移除或添加,从而达到结构优化的目的。水平集法则采用水平集函数表征结构的边界,其优化过程依赖于网格节点处水平集函数的演化。为了获得清晰的结构轮廓,传统拓扑优化算法通常需要离散化出密集的网格单元,这导致有限元分析存在显著的计算负担^[5]。相较于传统的隐式拓扑优化算法,基于可移动变形组件(movable morphable components, MMC)^[6]的显式拓扑优化方法具有显著优势。MMC方法以参数化构件(如杆)为基本单元,并将构件的几何参数(如尺寸、位置和方向)作为优化设计变量。这种显式拓扑优化方法可将设计变量个数减少至几十,并且不依赖于密集的网格,从而显著降低与拓扑优化相关的计算负担。

由于MMC方法所涉及的结构单元通常具有较大的设计空间,传统的梯度方法虽然效率高,但易陷入局部最优,因此,可将MMC框架与具有更强全局搜索能力的进化算法相结合,有望获得更优秀的优化结构。此外,在夹层结构优化领域,柔顺度计算依赖耗时的有限元分析,属于昂贵优化问题。然而,进化算法通常需要大量的函数评估才能充分发挥搜索性能,直接应用其解决昂贵优化问题通常效率低下^[7]。因此,可以引入代理辅助进化算法(surrogate-assisted evolutionary algorithm, SAEA)^[8]解决夹层结构优化问题。代理模型是一种机器学习模型,通过使用获得的昂贵样本近似昂贵问题的目标或约束^[9]。在优化过程中,代理模型的预测值可替代夹层结构优化问题中部分有限元分析以节省计算资源^[10]。在前期工作^[11]中,在MMC框架内引入了多分量拓扑描述函数,并显式构建了基于多折线与杆件夹芯的夹层结构(polyline-based core sandwich structure, PCSS)优化设计问题。通过将PCSS与SAEA技术深度融合,可实现夹层结构的高效优化。

针对PCSS问题优化,当前仍面临三大挑战:

首先,柔顺度有限元分析耗时比约束高出2~3个数量级,计算成本差异显著;其次,问题的可行域呈现出狭小且高度碎片化的分布;最后,子可行域的大小差异显著。

对于第一个挑战,现有SAEA方法普遍假设目标与约束的评估耗时处于同一量级,导致算法对两者的建模精度与评估频次采用同质化处理。直接应用传统SAEA将因频繁执行昂贵有限元分析而导致计算效率低下。对于第二个和第三个挑战,PCSS具有复杂的可行域结构,因此直接应用传统SAEA方法易发生早熟收敛。

综上所述,直接应用梯度算法易陷入局部最优解,进化算法则由于其需要消耗大量耗时的柔顺度评估次数而难以被接受,而传统的SAEA方法优化效果亦不理想。因此,针对PCSS及类似问题所面临的三大挑战,设计了用以解决夹层结构柔顺度优化问题的代理模型辅助差分进化算法(surrogate-assisted differential evolution algorithm for compliance optimization, CO-SADE)。

1 PCSS优化问题的构建

在PCSS中,夹层板由两个薄面和一个芯形成。两个面板用于构建设计域,但在优化过程中不被作为参数变量考虑。夹层由基于折线的芯结构(polyline-based core structure, PCS)组成。此外,PCS由重复的单胞阵列构成。由于PCS具有周期性结构,我们可以通过只关注单个单胞而不是整个结构来进行优化。在此基础上,如图1所示,单胞由多个2D基本组件组成。矩形结构两端的中心点坐标为 (x_1, y_1) 和 (x_2, y_2) ,它们同时对应着两个圆形结构的坐标,通过调整矩形结构两端的中心点坐标和矩形结构的厚度,可以实现2D组件的移动和扩展。为了进一步减少设计变量的数量, t_d 被设定为固定值。因此,设计变量被设计为各个2D基本组件的 y_i 以及 t_{2D} 。最后,上下面板与PCS组装以构建最终的PCSS。PCSS的结构如图2所示。

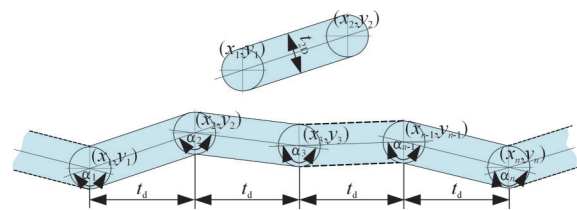


图1 单胞在 xy 平面横截面的几何参数描述
Fig. 1 Depiction for xy plane cross section of the unit cell

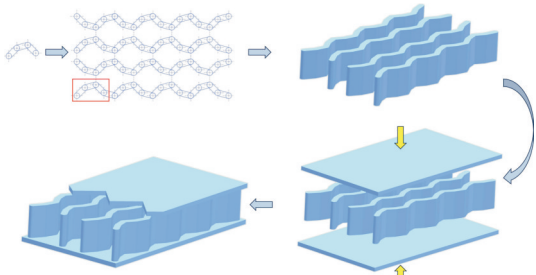


图2 PCSS的构造示意图

Fig. 2 Illustration for construction of PCSS

为了实现夹层结构的高刚度-质量比并且保证其可制造性,PCSS优化问题的优化目标是 minimized 柔顺度,质量约束可通过体积分数约束代替,而可制造性则通过角度约束控制。通过调用 MATLAB 可以分别评估柔顺度、体积分数约束和角度约束。其中,评估一次柔顺度耗时 3~5 min,体积分数约束评估耗时 0.4 s 左右,角度约束评估耗时可忽略不计。

PCSS 优化问题可由下式定义:

$$\begin{aligned}
 & \text{find } \tilde{D} = (y_1, y_2, \dots, y_n, t_{2D})^T \\
 & \min C = \int_D H(\phi(x; \tilde{D})) f u dV + \int_{\Gamma_t} t u dS \\
 & \text{s.t. } \int_D H(\phi(x; \tilde{D}))^q E : \epsilon(u) : \epsilon(v) dV = \\
 & \int_D H(\phi(x; \tilde{D})) f v dV + \int_{\Gamma_t} t v dS, \forall v \in U_{ad} \\
 & \int_D H(\phi(x; \tilde{D})) dV \leq \bar{V} \\
 & \cos(\alpha_l) \leq \cos(\underline{\alpha}) \\
 & \tilde{D} \subset U_{\tilde{D}} \\
 & u \subset \bar{u}, \text{ on } \Gamma_u \\
 & l = 1, 2, \dots, n
 \end{aligned} \quad (1)$$

式中: \tilde{D} 为PCSS问题的设计变量,由单胞中各2D基本组件的 y_i 以及 t_{2D} 构成; $U_{\tilde{D}}$ 为设计变量上下界; C 为柔顺度; f 和 t 分别为诺依曼边界 Γ_t 上的体积力密度和预设的表面牵引力; u 为位移场; \bar{u} 为狄利克雷边界 Γ_u 上的规定位移,且其值在本研究中被设置为0; $H(\cdot)$ 为赫维赛德函数; $\phi(x; \tilde{D})$ 为构建的多分量拓扑函数; q 为惩罚因子,其值在本研究中被设置为2; E 为给定弹性模量 E 和泊松比 ν 的各向同性固体材料的四阶弹性张量; ϵ 为二阶线性应变张量; U_{ad} 为虚拟位移向量 v 的可允许集合; \bar{V} 为固体材料可用体积的上界; α_l 为第 $(l-1)$ 个与第 l 个2D基本组件之间的夹角, $\underline{\alpha}$ 为此角度的下界。

详细的推导过程见文献[11]。

2 相关技术与背景方法介绍

2.1 部分评估

在处理昂贵优化问题时,评估策略可以分为全面评估和部分评估两种方式。当目标函数和约

束条件无法分开处理时,只能采用全面评估方法。近年来,对于可以将目标函数与约束条件分开处理的优化问题,部分评估方法的研究逐渐得到了关注。RAHI等^[12]提出了基于代理模型预测的约束值来确定约束评估顺序的策略,并在遇到约束冲突时停止评估,然而,该方法仅以约束信息指导真实函数评估。ZHANG等^[13]对此作出重要改进,通过融合父代个体的约束违反值和目标值,以及子代个体的预测约束违反值和目标值来指导评估,存在约束指导评估和目标指导评估两种情况,进一步提高了算法的优化效率。

2.2 种群分布调控

种群的分布是影响进化算法性能的关键因素之一。若种群分布过于分散,会造成计算资源的浪费,进而降低优化效率;而若种群分布过于集中,则可能限制算法的搜索能力,甚至引发早熟收敛。为了避免上述情况,LI等^[14]引入了最优个体的适应度值改进信息,若改进过小,则根据种群分布情况判定算法是处于早熟收敛还是停滞状态。在早熟收敛的情况下,种群会被分散,而在停滞的情况下,种群会被聚集。类似地,LIU等^[15]将种群的多样性融入SAEA中,在种群分布变得过于密集时,利用非支配排序重启种群。ZENG等^[16]则在每一代中探索不同类型的区域,根据区域的特性自适应地选择用于构建全局代理模型的个体以及用于生成子代的父代个体,并自适应选择变异方式,缓解了种群多样性丧失后无法跳出局部最优解的情况。

2.3 径向基函数

径向基函数(radial basis function, RBF)^[17]是一种简单的插值函数,基本原理是通过一组基函数的线性组合来近似复杂的函数。RBF模型相对而言具有较低的计算复杂度,因此,本研究采用RBF作为代理模型。

给定 N 个不同的点 $x_1, \dots, x_N \in \mathbb{R}^n$ 及其相应的函数值 $f(x_1), \dots, f(x_N) \in \mathbb{R}$,RBF可以用以下数学表达式表示:

$$\hat{f}(x) = w^T \phi = \sum_{i=1}^N \omega_i \phi(x - x_i) \quad (2)$$

$$w = (\Phi^T \Phi)^{-1} \Phi^T y \quad (3)$$

$$\Phi = \begin{bmatrix} \phi(x_1 - x_1) & \dots & \phi(x_1 - x_N) \\ \vdots & & \vdots \\ \phi(x_N - x_1) & \dots & \phi(x_N - x_N) \end{bmatrix} \quad (4)$$

$$y = (f(x_1), \dots, f(x_N))^T \quad (5)$$

式中: x_i 为第 i 个基函数的中心点; x 为需要被预测的点; ω_i 为与每个基函数关联的权重;由于立方核函数 $\phi(r) = r^3$

在 SAEA 中得到了广泛的应用,因此在本研究中使用 $\phi(r)=r^3$; φ 为一个包含基函数数值的 $N \times 1$ 向量。

方差可以表示 RBF 模型的预测不确定性,可按下述形式进行计算^[18]:

$$\sigma^2(x) = -M\Phi^{-1}M^T \quad (6)$$

$$M = (\phi(x-x_1), \phi(x-x_2), \dots, \phi(x-x_N)) \quad (7)$$

2.4 差分进化算法

在众多优化算法中,差分进化(differential evolution, DE)^[19]算法、粒子群优化(particle swarm optimization, PSO)^[20]算法和遗传算法(genetic algorithm, GA)^[21]得到了广泛应用。其中,DE 在处理连续优化问题时展示了较强的全局搜索能力和稳定性。此外,DE 存在多种具备明确搜索特性的变异操作,如极度全局搜索(DE/rand/1 和 DE/rand/2)、极度局部搜索(DE/current/1)以及全局与局部相平衡(DE/current-to-rand/1)。这意味着代理模型与 DE 有多种结合方式来实现不同的搜索目标,进而提高算法的优化效率。而 PSO 和 GA 性能虽然强大,但不满足本研究对多变体的要求。因此,在本文所提的所有方法中,均采用 DE 作为主要搜索引擎。

给定一个 n 维向量 $x_i = (x_{i,1}, \dots, x_{i,n})$, 根据当前种群中每个目标向量的变异操作生成变异向量 $v_i = (v_{i,1}, v_{i,2}, \dots, v_{i,n})$ 。本研究涉及的变异操作如下所示:

1) DE/rand/1:

$$v_i = x_{r1} + F_1(x_{r2} - x_{r3}) \quad (8)$$

2) DE/rand/2:

$$v_i = x_{r1} + F_1(x_{r2} - x_{r3}) + F_2(x_{r4} - x_{r5}) \quad (9)$$

3) DE/current/1:

$$v_i = x_{cur} + F_1(x_{r1} - x_{cur}) \quad (10)$$

4) DE/current-to-rand/1:

$$v_i = x_{cur} + F_1(x_{r1} - x_{cur}) + F_2(x_{r2} - x_{r3}) \quad (11)$$

式中: $x_{r1}, x_{r2}, x_{r3}, x_{r4}$ 和 x_{r5} 为从父代个体中随机选择的 5 个个体; x_{cur} 为当前的父代个体; F_1, F_2 为缩放因子,决定了变异的步长。

二项交叉如下所示:

$$u_{i,j} = \begin{cases} v_{i,j} & rand_j \leq C_R \text{ 或 } j = j_{rand} \\ x_{i,j} & \text{其他} \end{cases} \quad (12)$$

$$j = 1, 2, \dots, n$$

式中: $rand_j$ 为范围在 $(0, 1)$ 的随机数; C_R 为交叉概率参数; j_{rand} 为 $[0, n]$ 的随机整数; n 为设计变量的维数; $v_{i,j}, x_{i,j}$ 分别为变异向量和父代向量的第 j 维值; u_i 为执行交叉操作后的个体向量。

在交叉之后,会在父代个体以及 u_i 之间进行选择操作。

2.5 ϵ -约束方法

ϵ -约束(ϵ -constraint, EC)^[22]方法将原始问题的约束条件引入松弛参数 ϵ 以形成新的松弛约束。这意味着原始约束 $G(x) \leq 0$ 将替换为 $G(x) \leq \epsilon$ 。 ϵ 值的计算方法如下:

$$\epsilon(0) = G(x_\theta) \quad (13)$$

$$\epsilon(g_{en}) = \begin{cases} \epsilon(0)(1 - \frac{g_{en}}{G_{en}})^p & 0 < g_{en} < T_c \\ 0 & \text{其他} \end{cases} \quad (14)$$

$$c_p = -\frac{\log(\epsilon(0)) + \kappa}{\log(1 - T_c/G_{en})} \quad (15)$$

式中: $\epsilon(0)$ 为 ϵ 的初始值,每次迭代后都会更新 ϵ ,直至到达截止代数 T_c ; x_θ 为约束违反值排名第 θ 的个体向量; $G(\cdot)$ 为约束违反值; g_{en} 为当前迭代数; G_{en} 为总迭代数。

在本研究中, θ 取 $0.15N_{um}$, N_{um} 为对 ϵ 初始化时不可行解的数量; T_c 取 ϵ 初始化时剩余迭代次数的一半; κ 的大小能直接影响 ϵ 下降的趋势, κ 取 5。

当 x 和 y 进行比较时,当且仅当满足以下条件时,判断 x 优于 y :

$$\begin{cases} f(x) < f(y) & G(x) \leq \epsilon \text{ 且 } G(y) \leq \epsilon \\ f(x) < f(y) & G(x) = G(y) \\ G(x) < G(y) & \text{其他} \end{cases} \quad (16)$$

式中: $f(\cdot)$ 为柔顺度值; $G(\cdot)$ 为约束违反值。

当 ϵ 的值为 0 时,它等同于可行性法则(feasibility rule, FR)^[23]。

3 算法实现

3.1 算法的整体框架

算法的整体框架如算法 1 所示,首先通过拉丁超立方采样(LHS)^[24]方法在设计空间内均匀采样,获得大小为 N_p 的初始优化种群 Pop 。然后评估 Pop 的柔顺度和约束,并计算柔顺度评估与约束评估所消耗的时间,进而获得柔顺度评估与约束评估之间的成本差异。评估后的柔顺度以及约束信息分别存入 $Data_{Obj}$ 与 $Data_{Con}$,用于后续 RBF 代理模型建模。需要注意的是,当同时评估柔顺度与约束时,函数评估次数(function evaluations, F_{Es})会增加 1。然而,在第一阶段迭代过程中,夹层结构的柔顺度与约束会分开评估,且二者评估次数不相等。因此,每次柔顺度或约束评估后, F_{Es} 增加的数值会根据柔顺度或约束评估在一次完整评估中所占时间的比例 $ProObj$ 和 $ProCon$ 来调整。

算法 1: 整体框架

输入: 种群大小 N_p ; 最大允许函数评估次数 $\max F_{Es}$; 最大样本建模数量系数 r ; 柔顺度真实评估数据库

$Data_{Obj}$:约束真实评估数据库 $Data_{Con}$ 。

输出:最优解 x_{best} 。

通过LHS产生初始种群 Pop 。

评估 Pop 的柔顺度以及约束,并将真实评估后的数据存入 $Data_{Obj}$ 和 $Data_{Con}$ 。

$ProObj, ProCon \leftarrow$ 计算柔顺度评估和约束评估在一次完整评估中的时间占比。

$F_{Es} \leftarrow N_p$ 。

While $F_{Es} \leq \max F_{Es}$

If Pop 的可行率 ≤ 0.9

调用算法2执行第一阶段策略。

Else

调用算法4执行第二阶段策略。

End If

记录最优解并更新迭代信息。

End While

在每次迭代期间, Pop 的可行率不大于 0.9

时,算法处于第一阶段,否则,处于第二阶段。设置阈值为 0.9,可以有效减少不必要的约束评估。因为部分父代个体初始位置远离可行域,若等待其进入可行域,会导致种群迭代时额外进行大量的约束评估。最后,在迭代结束后记录最优解信息,并更新迭代相关参数,种群持续迭代直至评估次数 F_{Es} 超过允许的最大评估次数 $\max F_{Es}$ 。

本算法将约束优化及目标优化分为两个阶段。第一阶段主要优化夹层结构的约束,第二阶段则优化柔顺度,搜索全局最优解。

为更直观地理解 CO-SADE,在图3中给出了算法的主要流程。图中 CV 为约束违反值, Obj 为柔顺度(目标)值; $Offspring1$ 和 $Offspring2$ 表示两个子代种群, $Offspring$ 是从两个子代种群分别筛选出的 2 个个体中再筛选出的最佳候选个体。

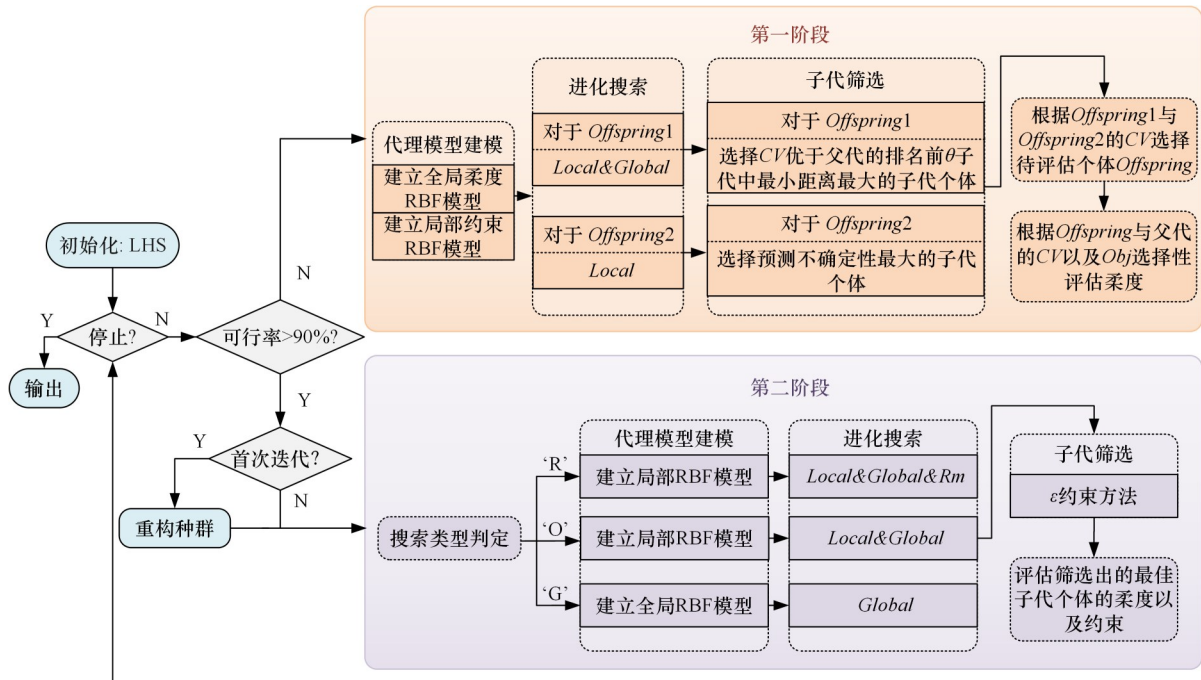


图3 CO-SADE流程图

Fig. 3 Flow chart of CO-SADE

3.2 第一阶段

第一阶段的具体实施过程如算法2所示,首先对柔顺度和约束建立RBF代理模型。根据建模所使用的解的数量不同,代理模型有全局或局部构建两种方式。全局代理模型是通过整合数据库中所有历史评估数据来构建,能够捕捉函数的宏观趋势并识别潜在优质区域^[25]。局部代理模型则通过筛选邻域样本来构建,具备更强的局部逼近能力^[16]。基于上述特性,本文采用混合建模策略。对柔顺度函数采用全局RBF模型以保持全局探索导向;对约束则构建局部RBF模型以提高

边界区域的预测精度。构建局部RBF模型的点数量为 $\min(\text{size}(Data_{Con}), 2r \times Dim_x)$ 。 r 是控制参与局部代理模型建模的点数量的参数,因为是双子种群协同进行优化,约束评估次数为常规的两倍,所以取 $2r$ 。 Dim_x 是设计变量的维度。

算法2:第一阶段

输入:种群大小 N_p ;已评估次数 F_{Es} ;种群 Pop 。

输出:更新后的评估次数 F_{Es} ;下一代种群 $NextPop$ 。

For $i = 1: N_p$

$p \leftarrow Pop(i)$ //父代个体

构建约束局部RBF模型。

构建柔顺度全局RBF模型。

```

//一个父代产生两个子代种群,细节如下:
//第一个子代种群逐步优化约束
OffType←Local&Global。
X←根据 OffType 进行差分进化。
 $X_s \leftarrow CV_x \leq CV_\beta$  的个体。
 $X_{sort} \leftarrow X$ , 中  $CV$  排名前  $\theta$  的个体。
Offspring1← $X_{sort}$  中最小距离最大个体。
//第二个子代种群提升约束 RBF 模型精度
OffType←Local。
X←根据 OffType 进行差分进化。
Offspring2← $X$  中预测方差最大个体。
调用算法 3 对约束和柔顺度进行评估。
NextPop←根据 FR 替换  $p$ 。
End For

```

双子代种群协同优化策略如算法 2 所示,其核心在于通过安排对应的变异操作以及筛选策略来实现不同的搜索目标。该策略中变异操作有 *Local&Global* 与 *Local* 两类, *Local&Global* 代表 DE/rand/1、DE/rand/2 和 DE/current-to-rand/1; *Local* 代表 DE/current/1。其中 DE/rand/1 和 DE/rand/2 通过随机个体差分扰动实现强大的全局探索能力,DE/current-to-rand/1 则通过当前个体与随机个体的混合差分平衡探索与开发;DE/current/1 通过当前个体邻域内的密集采样强化局部搜索能力。基于此,第一个子代种群通过组合 DE/rand/1、DE/rand/2 和 DE/current-to-rand/1 变异操作,在父代个体邻域内进行广泛探索;第二个子代种群则利用 DE/current/1 生成高密度候选解,专注于父代邻域的精炼开发。此外,进化的父代是参与构建约束局部 RBF 模型的个体。

然后调用算法 3 对 *Offspring1* 和 *Offspring2* 进行评估。最后,根据 FR 来替换父代种群。

算法 3:第一阶段部分评估策略

输入:第一个待评估子代个体 *Offspring1*;第二个待评估子代个体 *Offspring2*;已评估次数 F_{Es} ;柔顺度和约束在一次函数评估中的时间占比 *ProObj*, *ProCon*;父代个体 p 。

输出:更新后的评估次数 F_{Es} ;

$ConV_{Offspring1}$, $ConV_{Offspring2}$ ← 分别评估 *Offspring1* 和 *Offspring2* 的约束,并计算约束违反值。

$F_{Es} \leftarrow F_{Es} + 2 \times ProCon$ 。

If $ConV_{Offspring1} \leq ConV_{Offspring2}$

Offspring ← *Offspring1*。

Else

Offspring ← *Offspring2*。

End If

If 父代个体 p 是可行个体

① *Offspring* 可行且 RBF 模型预测目标值比父

代更优秀,则评估 *Offspring* 柔顺度, $F_{Es} \leftarrow F_{Es} + ProObj$;

② *Offspring* 可行但 RBF 模型预测目标值不优秀,则不评估 *Offspring* 柔顺度;

③ *Offspring* 不可行,不评估 *Offspring* 柔顺度。

ElseIf 父代个体 p 不是可行个体

① *Offspring* 可行则评估 *Offspring* 柔顺度, $F_{Es} \leftarrow F_{Es} + ProObj$;

② *Offspring* 不可行,则不评估 *Offspring* 柔顺度。

End If

部分评估策略如算法 3 所示,首先对 *Offspring1* 和 *Offspring2* 的约束进行评估,并选择约束违反值小的作为待柔顺度评估个体 *Offspring*。然后,根据父代个体以及 *Offspring* 的可行性和柔顺度值来决定是否评估柔顺度。上述约束和柔顺度评估的数据分别存入 $Data_{Obj}$ 与 $Data_{Con}$ 中。通过第一阶段的部分评估策略, $Data_{Obj}$ 中的可行解柔顺度评估信息来自各个子可行域中,显著减少了优化约束过程中不必要的柔顺度评估。

3.3 第二阶段

第二阶段的具体实施方案如算法 4 所示,如果当前迭代为第二阶段的第一次迭代,则通过 K-Means 方法对种群可行解聚 N_p 个类,并选择每个类中的柔顺度最小的个体,重构出新种群并初始化 ϵ 值。此后在每次迭代中,通过基于聚类的方法判断各个父代个体对应的搜索类型。首先使用 K-Means 方法对 $Data_{Obj}$ 中的可行解聚 N_p 个类,然后根据欧氏距离计算出各个类中包含的 Pop 中个体的数量 num 。最后,如下面伪代码所示,根据 num 与阈值 tol 的关系来决定对应类中所含的父代个体的搜索类型。

搜索类型判断

If all(num) < tol

所有父代个体的 $flag$ 均设置为‘O’。

Else If

① 若父代个体所属于的类 $num > tol$ 且最优解不属于该类:

$flag \leftarrow 'G'$

② 若父代个体所属于的类 $num > tol$ 且最优解属于该类:

$flag \leftarrow 'O'$

③ 若父代个体所属于的类 $num < tol$:

$flag \leftarrow 'R'$

End If

阈值 tol 的值由下式计算:

$$4 + (N_p/2 - 4) \cdot (F_{Es} - k) / (\max F_{Es} - k)$$

式中: k 为第一阶段结束后进入第二阶段时的评估次数; tol 的值取 4 为下界,随迭代次数增加逐渐增至一半种群

大小。

代理模型建模和进化搜索方式如算法4所示。对于搜索类型为‘O’的父代个体,构建局部RBF预测模型,并且进化搜索类型设置为 $Local&Global$ 。这种设置维持了原有的搜索方式,具有较强的邻域探索能力。构建柔顺度局部RBF模型的解数量为 $\min(size(Data_{obj}), r \times Dim_x)$,而对于体积分数约束的建模,选择位于柔顺度代理模型区域内的所有解(来自 $Data_{con}$)来构建。

算法4:第二阶段

输入:种群大小 N_p ;已评估次数 F_{Es} ;种群 Pop 。

输出:更新后的评估次数 F_{Es} ;下一代种群 $NextPop$ 。

```

If当前为第二阶段的第一代
  重构种群,初始化 $\epsilon$ 值
End If
For  $i = 1: N_p$ 
   $p \leftarrow Pop(i)$ . //父代个体
   $flag \leftarrow$ 判断搜索类型
  //代理模型建模:
  Switch  $flag$ 
  Case ‘R’ or ‘O’ 构建局部RBF模型
  Case ‘G’ 构建全局RBF模型
  //进化搜索:
  Switch  $flag$ 
  Case ‘R’  $OffType \leftarrow Local&Global&Rm$ 
  Case ‘O’  $OffType \leftarrow Local&Global$ 
  Case ‘G’  $OffType \leftarrow Global$ 
  根据 $OffType$ 产生候选子代 $X$ 。
  RBF模型预测 $X$ 柔顺度和体积分数约束。
  调用真实函数计算 $X$ 的角度约束。
  使用EC筛选出最佳候选个体。
  调用真实函数进行评估柔顺度和约束。
   $F_{Es} \leftarrow F_{Es} + 1$ 。
   $NextPop \leftarrow$ 根据FR替换 $p$ 。
End For

```

对于搜索类型为‘R’的父代个体,与搜索类型为‘O’相比,其进化搜索方式被设置为 $Local&Global&Rm$,这意味着会从生成的子代中剔除位于 num 超出阈值 tol 的类中的子代个体。一些工况下,次优解常分布于比全局最优解更大的可行域,这种分布特性易导致种群进化方向偏离最优区域。因此,通过上述方法可有效抑制次优解对种群进化轨迹的误导作用。此外,对于搜索类型为‘R’和‘O’的父代个体,构建柔顺度代理模型的个体被用来作为进化的父代。

对于搜索类型为‘G’的父代个体,其所属聚类为次优解所在的区域。由于使用邻域进化和局

部代理,种群难以逃离该区域,因此,需要调整代理模型的建模方式和进化搜索策略,以实现计算资源的有效转移。具体而言,柔顺度和体积分数约束采用全局代理模型,通过全局变异操作来探索潜在的优秀区域,以引导该类型的父代个体向最优解的方向转移。 $Global$ 代表 $DE/rand/1$ 和 $DE/rand/2$ 这两种变异操作。此外,进化的父代是通过 $Data_{obj}$ 中的可行解和不可行解分别聚 N_p 个类,并由类中柔顺度值最优秀的个体构成。

因为角度约束的评估的代价可忽略不计,为降低RBF预测误差带来的负面影响,在第二阶段子代的角度约束使用真实函数评估获得,而柔顺度和体积分数约束通过RBF代理模型预测获得。

在子代筛选方面,EC被用来筛选最佳候选子代个体。对于PCSS类夹层优化问题,整体刚度随体积分数的增大而增大,即全局最优解位于体积分数约束的边界处。而EC放松了原始约束条件,即允许柔顺度较小的不可行子代个体作为最佳候选子代个体。 ϵ 值随着迭代逐渐收敛,筛选出的子代个体逐渐逼近可行域。评估此类不可行解不仅有利于增强可行域边界处代理模型的预测精度,而且在约束复杂时可从不可行域和可行域两个方向逼近全局最优解。考虑到EC方法选择出的目标值优秀的不可行解可能比可行解更具有价值,因此在第二阶段不采用部分评估策略。

4 实验分析

4.1 实验设置

在本研究中,通过在3种经典载荷下的实验,分别与传统梯度算法和其他先进算法比较,以验证CO-SADE的性能。每个比较算法在 F_{Es} 达到1000次时终止。PCSS优化问题设计域的设置如图4所示。

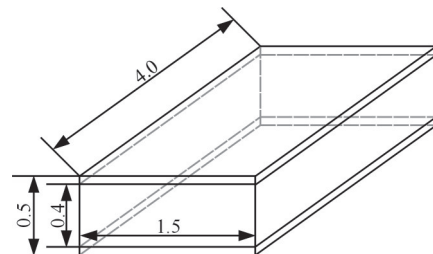


图4 PCSS优化问题的设计域

Fig. 4 Design domain of PCSS optimization problem

在所有实例中,位移分析采用统一的六面体单元离散设计域,并根据替代材料模型进行有限元分析。本实验所有涉及的数量均为量纲一的数量。固体材料的弹性模量 $E=1$,且 $E_{min}=10^{-6}$ 。泊

松比为 $\nu=0.3$ 。

设计域的尺寸为 $4\text{ mm} \times 1.5\text{ mm} \times 0.5\text{ mm}$ ，两个面板的厚度均为 0.05 。在本研究中，仅针对芯进行优化，因此实际设计域的尺寸为 $4\text{ mm} \times 1.5\text{ mm} \times 0.4\text{ mm}$ 。在计算 PCSS 优化问题的柔顺度和体积分数约束时，设计域被离散为 $160\text{ mm} \times 60\text{ mm} \times 20\text{ mm}$ 的有限元网格。在 3 种经典载荷的实验中，设计域参数和有限元网格划分方式保持一致，而施加的载荷和固定方式有所不同。

柔顺度、体积分数约束与角度约束均在 MATLAB 中评估。具体地，在 MATLAB 中划分网格后调用 `pcg` 函数求解出位移 U ，再通过 $C = F^T U$ 求解出柔顺度，其中， C 为柔顺度， F 为外部载荷向量。体积分数的计算是将材料体积除以整个设计域的体积。角度约束则通过计算两杆件之间的余弦值来衡量。

优化约束包括体积分数小于 0.3 ，与两杆件角度下界为 $\alpha = (2/3)\pi$ 。

在实例 1 中，如图 5 所示，PCSS 通过底部面板四个角的平面接缝约束固定，并在 PCS 表面长度的 $1/2$ 处沿着与高度平行的方向施加分布垂直荷载 $p=1$ 。

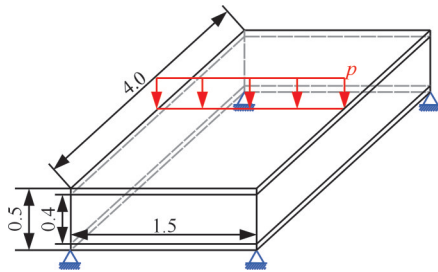


图 5 实例 1 的载荷示意图

Fig. 5 Schematic diagram of the load of case 1

在实例 2 中，如图 6 所示，PCSS 固定底部沿宽度方向的两边缘，并在 PCS 表面长度的 $1/2$ 处沿着与长度、宽度与高度平行的方向同时施加分布荷载 $p=1$ 。

在实例 3 中，如图 7 所示，PCSS 固定底部沿宽度方向的两边缘，并在 PCS 表面长度的 $1/2$ 处

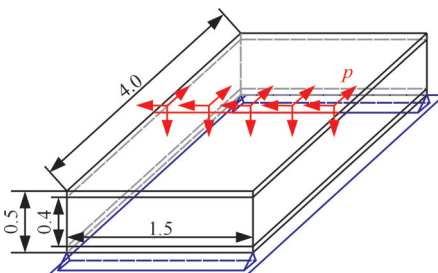


图 6 实例 2 的载荷示意图

Fig. 6 Schematic diagram of the load of case 2

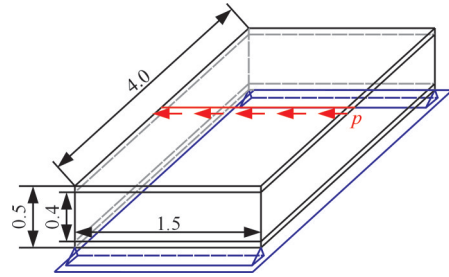


图 7 实例 3 的载荷示意图

Fig. 7 Schematic diagram of the load of case 3

沿着与宽度平行的方向施加分布荷载 $p=1$ 。

4.2 参数设置

在 CO-SADE 中，参数设置如下。种群大小 N_p 取 40；构造局部代理数量的系数 $r=7$ ；DE 相关参数 $F=0.2:0.2:1$ ； $C_r=0.4:0.2:1$ ；Global 中的各个变异操作子代数目设置为 1000，其余代号中的各个变异操作子代数目设置为 500；第一阶段筛选参数 θ 设置为 70%。

比较算法的参数参考其论文中原始参数。

4.3 消融实验

表 1 中列出了各个变体的名称及具体含义。

表 1 CO-SADE 的变体

Tab.1 Variants of CO-SADE

CO-SADE_FR	将第二阶段的筛选法则修改为 FR
CO-SADE_NF	移除第一阶段双子代种群协同优化和部分评估策略
CO-SADE_NS	移除第二阶段的基于聚类的自适应搜索策略

在表 2 中列出了 CO-SADE 各个变体的比较结果。CO-SADE 在实例 1 表现次优，在实例 2 和实例 3 中均表现最优。这证明了各个部件充分发挥了作用。

表 2 CO-SADE 变体的比较结果

Tab.2 Comparison results of the variants of CO-SADE

变体名	实例 1	实例 2	实例 3
CO-SADE	406.917	35.610	78.089
CO-SADE_FR	406.659	36.114	82.259
CO-SADE_NF	488.971	37.077	91.764
CO-SADE_NS	408.095	35.668	78.113

CO-SADE 与 CO-SADE_FR 的比较结果表明，运用 EC 直接逼近全局最优解在 PCSS 问题中是有效的。因为 FR 倾向于可行解，会使得柔顺度评估集中在可行域内，进而导致可行域边界的柔顺度预测精度较低，因此在实例 2 和实例 3 中仅搜索到次优解。然而，FR 会加速种群的收敛，这是 CO-SADE_FR 在实例 1 中表现最优的原因。

CO-SADE 与 CO-SADE_NF 的比较结果表明，第二阶段的优化建立在第一阶段的基础上，若缺乏第一阶段搜索到的子可行域，第二阶段的自

适应搜索策略反而会起到负面作用,影响算法的收敛速度。因此,在3个实例中,CO-SADE_NF均表现最差。

CO-SADE与CO-SADE_NS的比较结果表明,在3个实例中,CO-SADE的性能均获得了提高。这证明了基于聚类的自适应搜索策略能够有效地引导种群向更具潜力的区域转移。

4.4 CO-SADE与梯度算法的性能比较





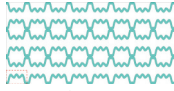
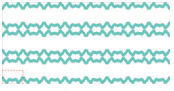

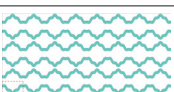
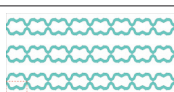
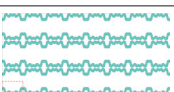
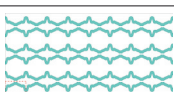

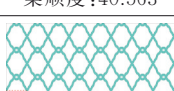
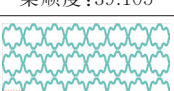

在本节中,CO-SADE与传统的梯度算法(traditional gradient-based method, TGB)进行实验对比。针对TGB实验,选取了3种不同的初始

解对PCSS进行优化。迭代优化方法设置为序列二次规划和内点法,并选取其中柔顺度值最小的结果与CO-SADE进行对比。

CO-SADE与TGB的对比结果如表3所示。可见,TGB在3种不同初始解下获得的最终结构柔顺度差异较大。这意味着TGB算法的性能和优化效果对初始解的选择具有较强的依赖性。其次,在3个测试实例中,TGB算法所获得的优化结果均不及CO-SADE。这说明CO-SADE算法在求解结构优化问题时展现出了更强的搜索能力和优化效果。

表3 CO-SADE和TGB的比较结果

Tab.3 Comparison results between CO-SADE and TGB

	TGB			CO-SADE
初始解				
实例1	 柔顺度:459.195	 柔顺度:430.474	 柔顺度:574.852	 柔顺度:406.917
实例2	 柔顺度:39.187	 柔顺度:40.563	 柔顺度:39.105	 柔顺度:35.610
实例3	 柔顺度:83.324	 柔顺度:89.319	 柔顺度:93.728	 柔顺度:78.089

4.5 CO-SADE与同类型算法的性能比较

将CO-SADE算法分别与FMSADE^[11]、MPMLS^[26]、SParEA^[12]和GLoSADE^[25]算法进行对比,以验证所提算法的优越性。

4.5.1 实验结果分析

表4~表6结果显示,CO-SADE、FMSADE、MPMLS与SParEA算法在3个测试实例中均可在有限评价次数内找到可行解,而GLoSADE算法仅在其中2个实例中成功。由于CO-SADE算法能够有效探索多个子可行域并动态调控种群分布,它在所有实例中均获得了柔顺度最优的结构,表现最佳。

FMSADE算法凭借其动态调整进化策略的能力,在探索与开发之间取得平衡,整体表现次优。MPMLS算法虽收敛较快,但未调控种群分布,导致搜索范围有限并早熟收敛。SParEA算法尽管采用部分评估策略,但因仅对可行解进行目标评估,其柔顺度代理模型精度不足,优化效果受限。

综合来看,SParEA算法与GLoSADE算法在可行解筛选机制上存在偏好,难以适应PCSS

问题中复杂可行域的挑战,这进一步印证了CO-SADE算法采用基于EC的子代筛选策略的有效性。

4.5.2 效率分析

表7列出了CO-SADE在获得相较于上文所提比较算法的更优结构时所消耗的时间占比。CO-SADE在3个实例中分别至少可将时间消耗缩减至53.9%、52.1%、62.4%,这验证了CO-SADE在优化过程中有效减少计算资源消耗的能力,显著提高了PCSS问题的优化效率;最多可分别缩减至4.4%、4.9%、4.3%,这是因为部分算法搜索可行域消耗时间过多,而CO-SADE在快速定位可行域的同时,可搜索到更优越的结构。

4.6 夹层结构三点弯曲试验验证

对实例1中柔顺度优化结果值相近的CO-SADE、FMSADE、MPMLS、GLoSADE算法所获得的夹层结构进行三点弯曲试验。

4.6.1 试样制备

试样制造使用Lite 800 3D打印机,以立体光固化成形技术,采用光固化树脂(C-UV9400EN)材料,3D打印出等比例放大30倍的夹层结构。3D

表 4 CO-SADE 和最先进算法在实例 1 上的比较结果

Tab.4 Comparison results between CO-SADE and state-of-the-art algorithms on case 1

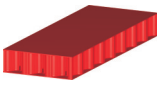


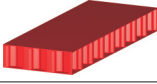

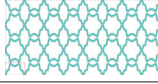
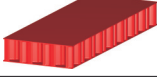
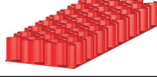
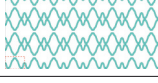
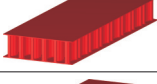
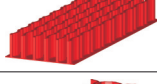
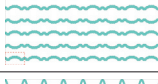
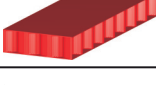
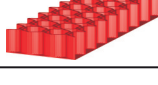
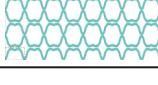
算法	柔顺度	CV	上板不隐藏	上板隐藏	xy平面横截面
CO-SADE	406.92	0			
FMSADE	435.03	0			
MPMLS	508.61	0			
SParEA	879.44	0			
GLoSADE	430.80	0			

表 5 CO-SADE 和最先进算法在实例 2 上的比较结果

Tab.5 Comparison results between CO-SADE and state-of-the-art algorithms on case 2

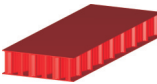
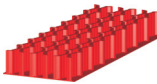
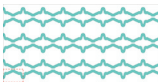
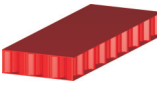

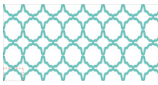
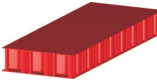
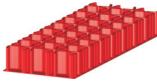
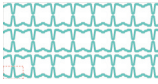
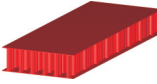
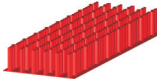

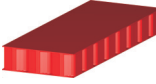
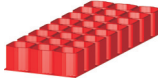

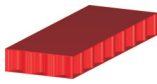
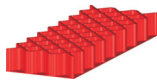

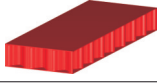
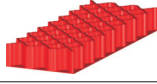

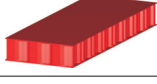
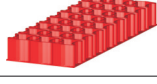
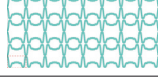
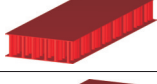
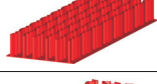
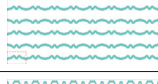
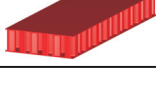
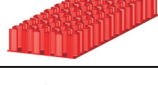
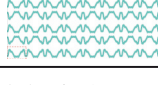
算法	柔顺度	CV	上板不隐藏	上板隐藏	xy平面横截面
CO-SADE	35.61	0			
FMSADE	38.30	0			
MPMLS	40.07	0			
SParEA	40.20	0			
GLoSADE	42.85	0			

表 6 CO-SADE 和最先进算法在实例 3 上的比较结果

Tab.6 Comparison results between CO-SADE and state-of-the-art algorithms on case 3

算法	柔顺度	CV	上板不隐藏	上板隐藏	xy平面横截面
CO-SADE	78.09	0			
FMSADE	79.46	0			
MPMLS	92.29	0			
SParEA	164.28	0			
GLoSADE	131.82	3.03			

打印机外观及制备的试样如图 8 所示。

4.6.2 加载设备及过程

使用通用万能试验机,按照 ASTM C393 试验标准,对夹层结构试样三点弯曲试验。其中,支座跨

距长度设置为 80 mm,加载速度恒定为 2 mm/min,测试完成后得到夹层结构的载荷-位移曲线。

4.6.3 试验结果分析

CO-SADE、FMSADE、MPMLS 和 GLo-

表7 CO-SADE超越最先进算法时的所耗时间占比

Tab.7 The proportion of time consumed when CO-SADE exceeds state-of-the-art algorithms %

比较算法	实例1	实例2	实例3
FMSADE	43.9	52.1	62.4
MPMLS	4.7	40.1	30.4
SParEA	4.4	40.1	5.1
GLoSADE	53.9	4.9	4.3

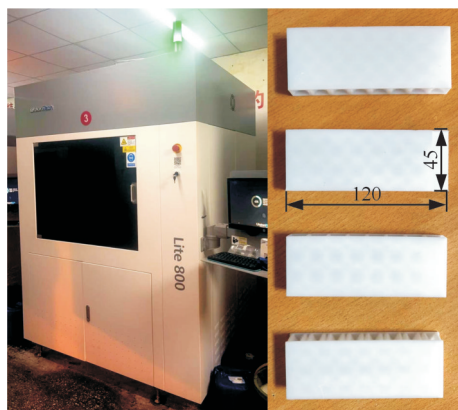


图8 3D打印机外观及制备的试样

Fig. 8 3D printer appearance and the prepared samples

SADE算法对应夹层结构的载荷-位移曲线见图9。在位移较小时,载荷与位移近似线性,斜率越大代表刚度越大。在位移达到3 mm左右时,夹层结构发生断裂,载荷急剧下降。图中结果与理论计算基本相符,CO-SADE(柔顺度406)的刚度和最大载荷都最大。FMSADE(柔顺度435)与GLoSADE(柔顺度430)理论计算柔顺度值相近,在图中的表现也相近。虽然MPMLS(柔顺度508)的物理试验结果比理论计算表现更优秀,但其在2.5 mm处斜率出现下降拐点,这代表其相较于其他3个算法,可能更早发生了结构的破坏。

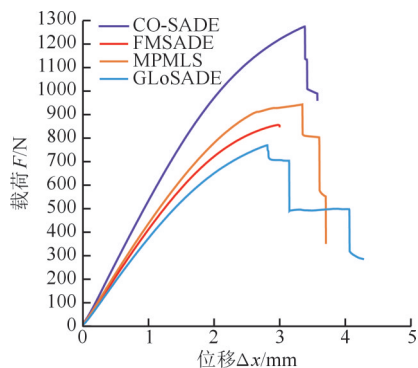


图9 载荷位移图

Fig. 9 Load displacement diagram

试验结果表明,本研究提出的CO-SADE算法的性能相较于同类型先进算法更加优秀,证明了CO-SADE算法的有效性。

5 结束语

本研究提出了一种用于解决夹层结构柔顺度优化问题的代理辅助进化算法(CO-SADE)。针对基于多折线与杆件夹芯的夹层结构(PCSS)优化中柔顺度评估与约束评估成本差异显著的特性,CO-SADE采用两阶段优化框架实现目标与约束的分步处理,同时引入部分评估策略,选择性进行柔顺度评估。针对PCSS优化中可行域狭小而又碎片化分布的特性,第一阶段通过双子种群协同优化策略探索可行解空间并提高约束代理模型精度。针对PCSS优化中可行域大小不均的特性,第二阶段设计了基于聚类的自适应搜索策略以动态调整种群搜索方向。

CO-SADE算法消融实验验证了所提策略的有效性:

1)第一阶段的双子种群协同优化策略搜索到大量可行域,部分评估策略节省昂贵的柔顺度有限元分析,两者结合显著提高了算法优化性能,并为第二阶段柔顺度的优化奠定基础。

2)基于聚类的自适应搜索策略有效增强全局探索能力,实现了计算资源向更具潜力区域的分配。其次,CO-SADE显著提高了PCSS问题的优化效率,分别在3个实例中,至少将时间成本缩减至53.9%、52.1%与62.4%。最后,通过三点弯曲试验进一步验证了所提算法的有效性。

在未来的工作中,拟将CO-SADE拓展至多目标。此外,将算法应用于处理目标与约束评估成本差异显著的其他工程问题也具有研究价值。

参考文献:

- [1] FAN Haitao, WANG Hai, CHEN Xiuhua. Optimization of Multi-sandwich-panel Composite Structures for Minimum Weight with Strength and Buckling Considerations [J]. Science and Engineering of Composite Materials, 2018, 25(2): 229-241.
- [2] XIE Y M, STEVEN G P. A Simple Evolutionary Procedure for Structural Optimization[J]. Computers & Structures, 1993, 49(5): 885-896.
- [3] BENDSØE M P, SIGMUND O. Material Interpolation Schemes in Topology Optimization[J]. Archive of Applied Mechanics, 1999, 69(9): 635-654.
- [4] WANG M Y, WANG Xiaoming, GUO Dongming. A Level Set Method for Structural Topology Optimization [J]. Computer Methods in Applied Mechanics and Engineering, 2003, 192(1/2): 227-246.
- [5] 陈继顺. 一种新的显式三维拓扑优化方法与基于MMC方法的拓扑优化尺寸控制[D]. 大连: 大连理工大学, 2018. CHEN Jishun. A New Explicit Three Dimension Topology Optimization Approach and Length Scale Control Based on MMC Framework[D]. Dalian: Dalian University of Tech-

- nology, 2018.
- [6] GUO Xu, ZHANG Weisheng, ZHONG Wenliang. Doing Topology Optimization Explicitly and Geometrically—a New Moving Morphable Components Based Framework [J]. *Journal of Applied Mechanics*, 2014, 81(8): 081009.
- [7] JIAO Ruwang, ZENG Sanyou, LI Changhe, et al. A Complete Expected Improvement Criterion for Gaussian Process Assisted Highly Constrained Expensive Optimization[J]. *Information Sciences*, 2019, 471: 80-96.
- [8] ZHANG Zhiyao, WANG Yong, LIU Jiao, et al. A Two-phase Kriging-assisted Evolutionary Algorithm for Expensive Constrained Multiobjective Optimization Problems[J]. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2024, 54(8): 4579-4591.
- [9] DONG Huachao, SONG Baowei, DONG Zuomin, et al. SCGOSR: Surrogate-based Constrained Global Optimization Using Space Reduction[J]. *Applied Soft Computing*, 2018, 65: 462-477.
- [10] LIU Yuanchao, LIU Jianchang, JIN Yaochu, et al. A Surrogate-assisted Two-stage Differential Evolution for Expensive Constrained Optimization[J]. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2023, 7(3): 715-730.
- [11] CHU Sheng, YANG Zan, XIAO Mi, et al. Explicit Topology Optimization of Novel Polyline-based Core Sandwich Structures Using Surrogate-assisted Evolutionary Algorithm[J]. *Computer Methods in Applied Mechanics and Engineering*, 2020, 369: 113215.
- [12] RAHI K H, SINGH H K, RAY T. Partial Evaluation Strategies for Expensive Evolutionary Constrained Optimization[J]. *IEEE Transactions on Evolutionary Computation*, 2021, 25(6): 1103-1117.
- [13] ZHANG Yong, JI Xinfang, GAO Xiaozhi, et al. Objective-constraint Mutual-guided Surrogate-based Particle Swarm Optimization for Expensive Constrained Multimodal Problems[J]. *IEEE Transactions on Evolutionary Computation*, 2023, 27(4): 908-922.
- [14] LI Chunlei, SUN Gaoji, DENG Libao, et al. A Population State Evaluation-based Improvement Framework for Differential Evolution [J]. *Information Sciences*, 2023, 629: 15-38.
- [15] LIU Jiansheng, YUAN Bin, YANG Zan, et al. Population State-driven Surrogate-assisted Differential Evolution for Expensive Constrained Optimization Problems with Mixed-integer Variables [J]. *Complex & Intelligent Systems*, 2024, 10(5): 6009-6030.
- [16] ZENG Yong, CHENG Yuansheng, LIU Jun. A Surrogate-assisted Constrained Optimization Evolutionary Algorithm by Searching Multiple Kinds of Global and Local Regions[J]. *IEEE Transactions on Evolutionary Computation*, 2025, 29(1): 61-75.
- [17] GUTMANN H M. A Radial Basis Function Method for Global Optimization[J]. *Journal of Global Optimization*, 2001, 19(3): 201-227.
- [18] SONG Zhenshou, WANG Handing, JIN Yaochu. A Surrogate-assisted Evolutionary Framework with Regions of Interests-based Data Selection for Expensive Constrained Optimization [J]. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2023, 53(10): 6268-6280.
- [19] DENG Libao, LI Chunlei, SUN Gaoji. An Adaptive Dimension Level Adjustment Framework for Differential Evolution [J]. *Knowledge-Based Systems*, 2020, 206: 106388.
- [20] 王桂荣, 倪志强, 周坤, 等. 多策略改进粒子群算法的机械臂时间最优轨迹规划[J]. *中国机械工程*, 2025, 36(5): 1044-1053.
- WANG Guirong, NI Zhiqiang, ZHOU Kun, et al. Time-optimal Trajectory Planning of Robotic Arms Based on MIPSO Algorithm [J]. *China Mechanical Engineering*, 2025, 36(5): 1044-1053.
- [21] KATOCH S, CHAUHAN S S, KUMAR V. A Review on Genetic Algorithm: Past, Present, and Future[J]. *Multimedia Tools and Applications*, 2021, 80(5): 8091-8126.
- [22] TAKAHAMA T, SAKAI S. Efficient Constrained Optimization by the ϵ Constrained Adaptive Differential Evolution [C]//*IEEE Congress on Evolutionary Computation*. Barcelona, 2010: 1-8.
- [23] DEB K. An Efficient Constraint Handling Method for Genetic Algorithms[J]. *Computer Methods in Applied Mechanics and Engineering*, 2000, 186(2/3/4): 311-338.
- [24] MCKAY M D, BECKMAN R J, CONOVER W J. A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code [J]. *Technometrics*, 2000, 42(1): 55-61.
- [25] WANG Yong, YIN Daqing, YANG Shengxiang, et al. Global and Local Surrogate-assisted Differential Evolution for Expensive Constrained Optimization Problems with Inequality Constraints[J]. *IEEE Transactions on Cybernetics*, 2019, 49(5): 1642-1656.
- [26] LI Genghui, ZHANG Qingfu. Multiple Penalties and Multiple Local Surrogates for Expensive Constrained Optimization[J]. *IEEE Transactions on Evolutionary Computation*, 2021, 25(4): 769-778.

(编辑 王旻玥)

作者简介: 杨赞,男,1994年生,讲师、博士。研究方向为复杂装备智能设计、智能优化算法、拓扑优化等。E-mail: yangzan@ncu.edu.cn。 邱浩波*(通信作者),男,1974年生,教授。研究方向为数字化设计与制造、复杂装备智能设计及其可靠性。E-mail: hobbyqiu@163.com。

本文引用格式:

杨赞,朱紫华,孙观观,等. 面向夹层结构柔顺度优化的代理模型辅助差分进化算法[J]. *中国机械工程*, 2026, 37(3): 612-623.

YANG Zan, ZHU Zihua, SUN Guanguan, et al. Surrogate-assisted Differential Evolution Algorithm for Compliance Optimization of Sandwich Structures[J]. *China Mechanical Engineering*, 2026, 37(3): 612-623.