

黑箱模型约束动态松弛的近似优化方法

马帅超, 武泽平*, 杨家伟, 高经纬

(国防科技大学 空天科学学院, 湖南 长沙 410073)

摘要:基于代理模型的优化方法因其高效的寻优能力为高精度仿真模型在优化设计中的应用提供了一种有效的技术途径。针对优化问题中高耗时黑箱约束处理问题,提出了一种基于改进可行性规则的多约束自适应采样方法,建立了精英档案驱动的非精确搜索方法和 ε -约束保持的伪可行域构造方法,通过迭代过程中动态缩放可行域以接受高质量非可行样本,增强了算法对可行域边界的探索能力,提高了基于代理模型的寻优能力。对进化计算会议约束优化标准函数的仿真计算结果表明,相较于现有方法, ε -约束保持优化方法可以有效解决多约束代理模型优化问题。对固体火箭发动机后翼柱装药设计结果表明算法具有应用在复杂工程问题中的潜力。

关键词:代理模型;约束处理;自适应采样; ε -约束保持

中图分类号:V421.1

文献标志码:A

文章编号:1001-2486(2025)05-125-09



Approximate optimization method for constraints dynamic relaxation of black box model

MA Shuaichao, WU Zeping*, YANG Jiawei, GAO Jingwei

(College of Aerospace Science and Engineering, National University of Defense Technology, Changsha 410073, China)

Abstract: The surrogate model-based optimization method provides an effective technical approach for the application of high-precision simulation models in optimal design due to its efficient search capability. To address the problem of time-consuming black-box constraint processing for optimization problems, a multi-constraint adaptive sampling method based on improved feasible rules was proposed, an elite archive-driven inexact search method and an ε -constraint-preserving pseudo-feasible domain construction method were established, and the algorithm's ability to explore the boundaries of the feasible domain was enhanced by dynamically scaling the feasible domain to accept high-quality nonfeasible samples during the iterative process, which improved the surrogate model-based optimization search ability. Simulation results of the Congress on Evolutionary Computation constraint optimization standard function indicate that the ε -constraint maintenance optimization method is effective in solving the multi-constraint surrogate model optimization problem compared with the existing methods. The results for the solid rocket motor rear wing pillar charge design show that the algorithm has the potential to be applied to complex engineering problems.

Keywords: surrogate model; constraint processing; adaptive sampling; ε -constraint holding

在以飞行器设计为代表的复杂工程设计问题中,为提高设计性能、降低研制成本、缩短设计周期,通常需要对产品性能、成本等设计指标进行优化设计^[1]。同时为了提高仿真计算结果的可信度,高耗时仿真模型在实际工程设计中的应用越来越广泛,而模型仿真精度的提升会导致仿真时长指数增长,致使计算成本上升的同时优化效率低下。为解决上述问题,基于代理模型的优化方

法被广泛应用于飞行器优化设计过程以达到降低设计、分析和计算消耗的目标^[2]。在基于代理模型的设计方法中,自适应采样策略根据现有代理模型和样本点分布确定最佳采样点来进行高耗时模型仿真,从而获得更有利于优化集成的样本点。上述采样过程需要按照采样准则搜索得到新样本点输出并扩充具有高精度特性的样本集。根据采样准则优化指标的不同,采样准则分为局部开发、

收稿日期:2023-02-16

基金项目:国家自然科学基金资助项目(52005502)

第一作者:马帅超(2000—),男,河南平顶山人,硕士研究生,Email:mashuaichao18@nudt.edu.cn

*通信作者:武泽平(1990—),男,山西吕梁人,副研究员,博士,硕士生导师,Email:zeping123@nudt.edu.cn

引用格式:马帅超,武泽平,杨家伟,等.黑箱模型约束动态松弛的近似优化方法[J].国防科技大学学报,2025,47(5):125-133.

Citation: MA S C, WU Z P, YANG J W, et al. Approximate optimization method for constraints dynamic relaxation of black box model[J].

Journal of National University of Defense Technology, 2025, 47(5): 125-133.

全局探索和开发/探索平衡三种策略^[3]。对于黑箱约束优化问题,现有算法无法直接应用,需要与约束处理方法相结合方可得到满足约束的最优可行解。同时约束优化问题中存在的可行域边界也进一步增加了算法的困难与挑战。

在实际工程问题中,除目标函数之外,大多数约束也同样具有类似的性质。在引入约束条件之后,约束类型、不同约束个数、可行域大小、有效约束个数等复杂因素都会导致优化算法效率降低、全局最优解搜索困难等问题^[4]。为此,近年来研究者们提出了针对近似优化方法的约束处理方法,使算法能够探索到空间可行域。Huang 等提出的可行性规则法^[5]易于实现且不需要烦琐的调参过程,在差分进化(differential evolution, DE)算法和粒子群优化(particle swarm optimization, PSO)算法等进化算法中广泛应用。但其对约束的处理过程太过严格,可能导致算法无法考虑约束边界的高质量不可行解,从而降低算法的寻优能力。

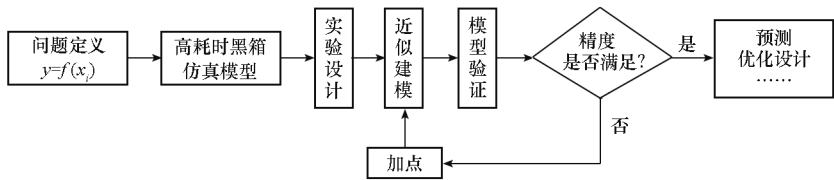


图 1 基于代理模型的优化方法基本流程

Fig. 1 Basic process of surrogate model-based model optimization method

实验设计能够为代理模型提供在设计空间上均匀分布的训练样本,通常采用拉丁超立方实验设计方法获取^[7]。近似建模通过训练样本输入输出关系拟合出原始模型的代理模型,使用径向基函数近似建模方法^[8]构建代理模型。自适应采样是通过现有样本点的分布以及代理模型的构建去搜索选择下一个样本点,以期在拓展样本点集中选出最优解。

Mckay 提出拉丁超立方实验设计方法^[9](Latin hypercube design, LHD),能够使样本点较为均匀分布在设计空间,目前已经成为主流的实验设计方法。针对 n 个 m 维样本点的设计结果是一个矩阵,通过采用拉丁超立方实验设计方法^[10]对此问题设计求解,可以在降低仿真实验次数的基础上快速得出满足均匀性要求的实验设计排列。其中样本与空间中对应位置如图 2 所示。

构建代理模型的方法有径向基函数(radial basis function, RBF)建模方法^[6]、Kriging 建模方法^[11]、多项式建模方法^[5]、高斯过程回归(Gaussian

针对约束处理中存在的问题,在进行优化过程中的自适应采样时需要选取合适的约束处理方法才可能选取到高质量可行解。在基于可行性规则法处理约束的基础上,首先引入 ε -约束保持法^[6],通过先接受不可行解用以探索可行域边界,再不断放缩伪可行域直至可行域,确保优化结果在约束内。基于此,本文将可行性规则法进行改进,增强其探索约束边界最优的能力,并提出约束动态松弛的近似优化(constraints dynamic relaxation approximate optimization, CDRAO)方法,最后通过数学算例和工程算例对提出的方法进行验证。

1 代理模型多约束优化方法基本框架

工程问题中的黑箱模型输出结果需要通过大量计算得出,为了提高计算效率,使用代理模型代替高耗时黑箱模型进行优化设计。基于代理模型的优化方法主要包括实验设计、代理模型构建、自适应采样三个方面,其基本流程如图 1 所示。

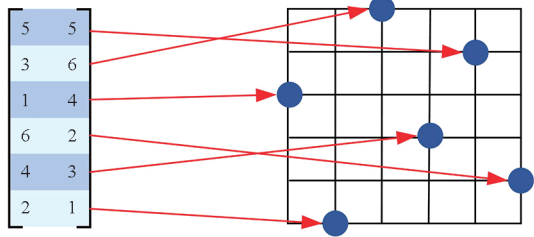


图 2 拉丁超立方实验设计方法

Fig. 2 Latin hypercube experimental design methodology

process regression, GPR) 建模方法^[12]、深度学习(deep learning, DL)建模方法^[13]等。RBF 模型相比其他建模方法具有设计简单、计算量小、泛化能力强等优点^[14],因此选择 RBF 建模方法构建代理模型。

标准 RBF 模型基本形式为

$$\hat{f}(\boldsymbol{x}) = \sum_{i=1}^n \omega_i \varphi_i(r) \quad (1)$$

式中, n 为训练集大小, $\boldsymbol{x}$ 为待预测的点, ω_i 为基函数系数, $\varphi_i(r)$ 为高斯基函数。其中高斯基函数可写为

$$\varphi_i(r) = \exp\left(-\frac{r^2}{c_i^2}\right) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}_i\|^2}{c_i^2}\right) \quad (2)$$

式中, c_i 为形状参数。

形状参数值 c_i 确定^[15-16]后,反算式(1),即可计算出对应的权系数 ω_i 。通过确定基函数系数,就可以获得任意样本的预测输出。

自适应采样需要基于当前代理模型和样本点分布,构造准则并对下一个样本点进行搜索选择,再通过代理模型计算新样本点的输出,实现样本集的扩充。自适应采样主要分为三种:局部开发、全局探索以及开发/探索平衡。局部开发准则基本思想是在代理模型精确逼近真实模型的假设下,寻找出代理模型的最优点,代入真实模型中验证并用来更新样本数据集。全局探索准则侧重于模型的全局最优,通过在近似模型的样本点稀疏区域选取采样点,从而更易于求出模型的全局最优解。局部开发准则易陷入局部最优,全局探索准则由于需要探索设计空间全局而易造成计算资源的浪费,因此提出非精确搜索的采样准则,通过选取既能满足结果更优又能保证选取点与已选点之间的距离较远的采样点,实现更效率的寻优采样。

但在实际工程问题中,目标函数通常有约束限制从而成为约束优化问题,且约束多为黑箱约束。使用无约束代理模型优化算法优化约束问题时,由于约束类型、可行域大小、约束个数等多方面因素的影响,难以完成目标函数寻优。如图3所示,考虑可行域内空心采样点建立的代理模型如图中虚线所示,由此得出的最优解并不是实际的最优解。因此应开展多约束条件下的采样方法研究。

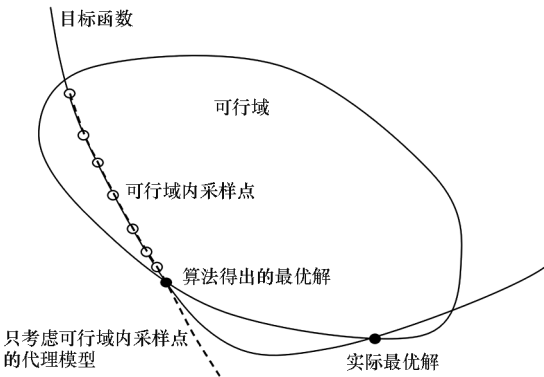


图3 带约束优化存在的问题

Fig.3 Problems with constraint optimization

2 多约束自适应采样方法

2.1 基于约束冲突函数的约束处理方法

单目标非线性约束优化问题可按如下形式定义:

$$\begin{aligned} & \text{Find } \mathbf{X} \\ & \min f(\mathbf{X}) \\ & \mathbf{X}^L \leq \mathbf{X} \leq \mathbf{X}^U \\ & \text{s. t. } \begin{cases} g_j(\mathbf{X}) \leq 0 & j=1,2,\dots,p \\ h_k(\mathbf{X}) = 0 & k=p+1,p+2,\dots,m \end{cases} \end{aligned} \quad (3)$$

式中: $\mathbf{X}$ 为设计变量,通常为 d 维向量; $f(\mathbf{X})$ 为目标函数(标量函数); $\mathbf{X}^L$ 、 $\mathbf{X}^U$ 分别为设计变量的上、下限; $g_j(\mathbf{X}) \leq 0$ ($j=1,2,\dots,p$) 为不等式约束; $h_k(\mathbf{X}) = 0$ ($k=p+1,p+2,\dots,m$) 为非设计变量上、下限的不等式约束。

在工程设计问题中多数约束计算同目标函数具有相似的复杂特征,难以找寻可行解,因此需要采用约束处理方法对约束进行处理。本文引入约束冲突函数,用以表征样本点与可行域之间的距离,使算法能够平衡约束与目标函数的关系并提升算法对可行域边界的探索能力,从而能够更好地找寻全局最优可行解。将等式约束做如下处理:

$$|h_k(\mathbf{X})| - \delta \leq 0 \quad (4)$$

式中, $h_k(\mathbf{X})$ 表示第 k 个等式约束, δ 为一小量。

应用式(4),将等式约束转化为不等式约束,定义个体 $\mathbf{X}$ 违反第 l 个约束条件的程度为:

$$G_l(\mathbf{X}) = \begin{cases} \max\{0, g_l(\mathbf{X})\} & 1 \leq l \leq p \\ \max\{0, |h_l(\mathbf{X})| - \delta\} & p+1 \leq l \leq p+m \end{cases} \quad (5)$$

定义个体 $\mathbf{X}$ 违反所有约束的程度为:

$$G(\mathbf{X}) = \sum_{l=1}^m G_l(\mathbf{X}) \quad (6)$$

由于不同约束的尺度特征不同,需进行归一化处理,以避免个别约束对 $G(\mathbf{X})$ 的值起决定性作用:

$$G_{\text{nor}}(\mathbf{X}_i) = \frac{\sum_{l=1}^m G_l(\mathbf{X}_i) / G_l^{\max}}{m} \quad (7)$$

式中: $G_{\text{nor}}(\mathbf{X}_i)$ 为粒子 $\mathbf{X}_i$ 的标准化约束冲突值; $G_l^{\max}$ 为群体针对单个约束的最大冲突值,即

$$G_l^{\max} = \max_{i=1,\dots,N} (G_l(\mathbf{X}_i)) \quad l \in \{1,2,\dots,m\} \quad (8)$$

N 为精英档案中样本点的个数。若 $G_{\text{nor}}(\mathbf{X}_i) \leq 0$, 则表明个体 $\mathbf{X}_i$ 是可行个体;否则 $\mathbf{X}_i$ 为不可行个体,且 $G_{\text{nor}}(\mathbf{X}_i)$ 值越大说明个体 $\mathbf{X}_i$ 在群体中相对约束冲突越大。

2.2 基于 ε -约束保持改进的可行性规则法

可行性规则法容易实现且优化过程中不用考虑相关参数调整,但是忽略了位于可行域边界的高质量不可行解。特别是对于代理模型优化问题,在优化初期探索阶段难以定位最优解区域,而

在可行域边界外附近的高质量不可行解能够提升算法对可行域边界的搜索能力,进而提升算法的探索能力。因此提出 ε -约束保持的思想^[6],在优化初期对可行域进行放大,随着算法的迭代,可行域也不断缩小,直至与真实可行域重合为止。在优化前期算法对可行域进行探索时将高质量不可行解加入样本点集,提升了算法对可行域边界的探索能力。如图 4 所示,在原有采样点的基础上将伪可行域内采样点考虑进去,使得代理模型更加逼近于目标函数,得出更为真实的可行最优解。算法的基本流程基本与可行性规则法相同,但将可行域改成包含可行域边界外的可放缩伪可行域。改进后三条规则如下:

- 1) 伪可行解优于不可行解;
- 2) 同为伪可行解,目标函数性能更理想的更优;
- 3) 同为不可行解,约束冲突值小的更优。

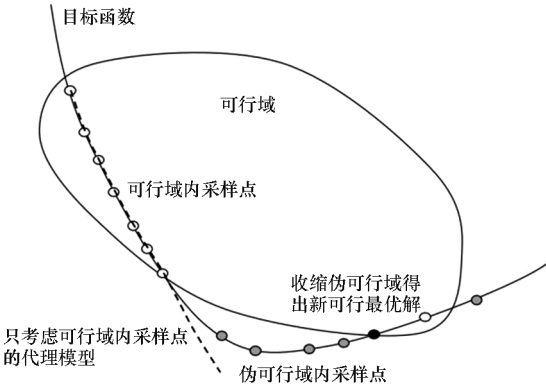


图 4 考虑伪可行域内采样点后优化

Fig. 4 Optimization after considering sampling points in the pseudo-feasible domain

引入松弛参数 ε ,将判断可行标准由 $G_{\text{nor}}(X_i) \leq 0$ 改进为 $G_{\text{nor}}(X_i) \leq \varepsilon$,通过调整值 ε 放缩伪可行域。由于 $G_{\text{nor}}(X)$ 经过归一化处理,其取值在 $[0,1]$ 之间,因此针对不同约束优化问题,松弛参数 ε 可以采用相同调整规律。为了确保最优点处于可行域内, ε 应随迭代次数增加而逐渐减小为 0,以促使伪可行域边界逐渐向可行域边界靠拢并最终重合。

2.3 基于改进可行性规则的多约束自适应采样方法

引入精英档案的思想来表征采样点性能的提升,并将精英档案初始化为实验设计样本点,经过后续迭代优化保存性能最优的若干样本点且保持数量不变。在每次迭代过程中如果有新样本点优于精英档案内存储点,则将其进行替换从而保持精英档案的最优。精英档案更新流程及其整体算

法流程如图 5 和图 6 所示。结合基于 ε -约束保持法改进的可行性规则法,提出改进可行性规则的多约束自适应采样方法,按照“先探索可行域,再寻优”的思想,采样过程分两阶段进行:

- 1) 若精英档案中存在不可行解,算法应在探索到可行点后停止搜索,并在当前的种群中找到可行且密度最小的点。
- 2) 若精英档案中全部是伪可行解,算法应在探索到优于精英档案最优的点后停止搜索,并在当前采样点中找到优于精英档案最优解且密度最小的点。

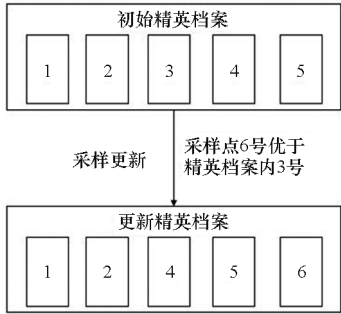


图 5 更新精英档案示意图

Fig. 5 Update elite profile diagram

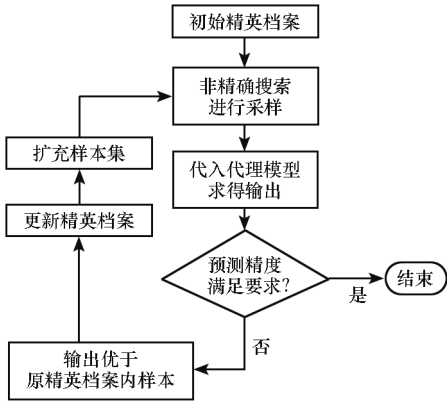


图 6 非精确搜索采样方法流程示意图

Fig. 6 Non-exact search sampling method flow diagram

满足 $G_{\text{nor}}(X) \leq \varepsilon$ 的 X 称为伪可行解,其中 ε 由下式得出:

$$\varepsilon = 0.1 \times c_j$$

(9)

式中, c_j 为随算法迭代次数的增加从 1 减小至 0 的系数,定义

$$c_j = \frac{\sum_{i=1}^s d_i(\mathbf{x}_N)}{\sum_{i=1}^s d_i(X_N)}$$

(10)

其中, $\mathbf{x}_N$ 代表当前精英档案样本集, s 表示样本集中距离不同点的对数, d_i 表示不同样本间的欧几里得距离。通过对样本进行归一化处理确保各个

变量影响权重一致。 $c_j = 1$ 表明精英档案为均匀分布的初始样本; $c_j \approx 0$ 表明精英档案近似收敛到同一点上,此时算法终止。定义收敛精度 e , 若 $c_j \leq e$, 则算法收敛。

优化初始阶段设 $G_{\text{nor}}(\mathbf{X}) \leq 0.1 \times 1$, 即所有的约束边界被放宽 10% 的裕度, 低于这个裕度的解可作为可行解进行阶段 1 采样。优化后期要求所有约束满足 $G_{\text{nor}}(\mathbf{X}) \leq 0$, 算法只能开发可行域直到定位全局最优。值得注意的是, 阶段 1 的采样和阶段 2 的采样并不具有单向性, 当前采样点被判断为伪可行解之后, 随着约束松弛因子 ε 减小, 可行域不断收敛, 原先的伪可行解变为不可行解, 此时需要重新转到阶段 1 进行采样以保证种群的可行性。

由于采样过程根据采样点特征分成两种策略, 精英档案的更新策略同样要适应采样策略。精英档案更新策略按如下定义:

- 1) 若精英档案中存在不可行解, 精英档案中约束违反程度最大的样本点被更新为阶段 1 采样获得的样本点。
- 2) 若精英档案中全为伪可行解, 精英档案中目标函数最差的解被更新为阶段 2 采样获得的样本点。

采样过程如图 7 所示。

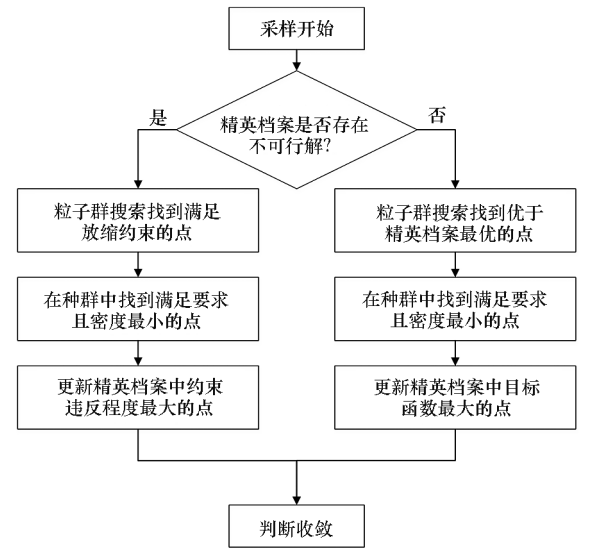


图 7 基于改进可行性规则的多约束自适应采样方法
Fig. 7 Multi-constraint adaptive sampling method based on improved feasible rules

外部代理模型优化的最优解追踪调整为: 优先追踪精英档案中满足约束且目标函数值最小的点, 若精英档案中不存在满足约束的点, 最优解设置为当前采样点中约束冲突最小点。若采样点都不可行, 最优解可以帮助其逐渐向可行域收敛。采样点都可行时, 最优解要保证其在目标函数上

的优越性。面向多约束问题的非精确采样代理模型优化方法流程如图 8 所示, 其中流程中的代理模型构建采用基于样本局部密度的径向基建模方法, 搜索采样过程采用改进粒子群算法。

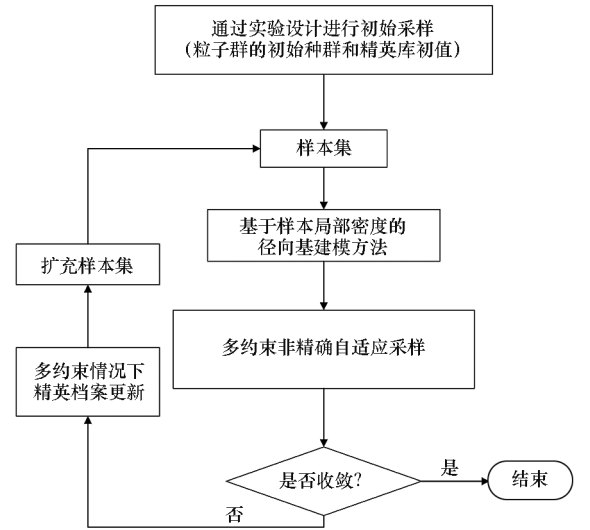


图 8 多约束问题的非精确采样代理模型优化流程
Fig. 8 Non-exact sampling surrogate model optimization process for multi-constraint problems

3 方法验证

3.1 数值算例

为了验证约束动态松弛优化方法的有效性, 从进化计算会议 (IEEE Congress on Evolutionary Computation, CEC) 约束优化标准函数中选取 8 个约束优化进行测试, 其目标函数、约束条件、自变量范围以及函数全局极小值如表 1 所示。CEC 标准测试问题是 CEC 组委会面向不同问题类型发布的标准测试算例, 这些算例为全局最优解已知的函数, 被广泛用于优化方法性能验证^[17-18]。

为充分测试方法的适用性和高效性, 选取的测试问题设计变量维度涵盖 2 维到 13 维, 约束形式包括不等式与等式约束。在处理等式约束时, 由于难以找到严格满足等式约束的解, 故将其做如下处理:

$$h(\mathbf{X}) = 0 \rightarrow |h(\mathbf{X})| - \delta \leq 0 \quad (11)$$

式中, δ 为一小量, 通常取 0.001。

在进行测试算例时, 针对每一个算例均独立运行 25 次以上, 确保运行结果的稳定。将优化结果与近年来发表的约束优化混沌灰狼算法^[19] (chaotic grey wolf optimization, CGWO)、扩展平衡排序方法^[20] (the extended balanced ranking method, E-BRM) 和全局人工蜂群算法^[21] (modified global best artificial bee colony, MGABC) 进行对比, 得出结果如表 2 所示。

表 1 约束优化标准函数

Tab. 1 Constraint optimization criteria functions

标准函数	表达式	约束函数	自变量范围	全局极小值
g01	$f(x) = (x_1 - 10)^3 + (x_2 - 20)^3$	$g_1(x) = -(x_1 - 5)^2 - (x_2 - 5)^2 + 100 \leq 0$ $g_2(x) = (x_1 - 6)^2 + (x_2 - 5)^2 - 82.81 \leq 0$	$13 \leq x_1 \leq 100$ $0 \leq x_2 \leq 100$	-6 961.813 87
g02	$f(x) = x_1^2 + (x_2 - 1)^2$	$h(x) = x_2 - x_1^2 = 0$	$-1 \leq x_i \leq 1$ ($i = 1, 2$)	-0.749 9
g03	$f(x) = -x_1 - x_2$	$g_1(x) = -2x_1^4 + 8x_1^3 - 8x_1^2 + x_2 - 2 \leq 0$ $g_2(x) = -4x_1^4 + 32x_1^3 - 88x_1^2 + 96x_1 + x_2 - 36 \leq 0$	$0 \leq x_1 \leq 3$ $0 \leq x_2 \leq 4$	-5.508 013
g04	$f(x) = 5.357\ 854\ 7x_3^2 + 0.835\ 689\ 1x_1x_5 + 37.293\ 239x_1 - 40\ 792.141$	$g_1(x) = 85.334\ 407 + 0.005\ 685\ 8x_2x_5 + 0.000\ 626\ 2x_1x_4 - 0.002\ 205\ 3x_3x_5 - 92 \leq 0$ $g_2(x) = -85.334\ 407 - 0.005\ 685\ 8x_2x_5 - 0.000\ 626\ 2x_1x_4 + 0.002\ 205\ 3x_3x_5 \leq 0$ $g_3(x) = 80.512\ 49 + 0.007\ 131\ 7x_2x_5 + 0.002\ 995\ 5x_1x_2 + 0.002\ 181\ 3x_3^2 - 110 \leq 0$ $g_4(x) = -80.512\ 49 - 0.007\ 131\ 7x_2x_5 - 0.002\ 995\ 5x_1x_2 - 0.002\ 181\ 3x_3^2 + 90 \leq 0$ $g_5(x) = 9.300\ 961 + 0.004\ 702\ 6x_3x_5 + 0.001\ 254\ 7x_1x_3 + 0.001\ 908\ 5x_3x_4 - 25 \leq 0$ $g_6(x) = -9.300\ 961 - 0.004\ 702\ 6x_3x_5 - 0.001\ 254\ 7x_1x_3 - 0.001\ 908\ 5x_3x_4 + 20 \leq 0$	$78 \leq x_1 \leq 102$ $33 \leq x_2 \leq 45$ $27 \leq x_i \leq 45$ ($i = 3, 4, 5$)	-30 665.5
g05	$f(x) = 5 \sum_{i=1}^4 x_i - 5 \sum_{i=1}^4 x_i^2 - \sum_{i=5}^{13} x_i$	$g_1(x) = 2x_1 + 2x_2 + x_{10} + x_{11} - 10 \leq 0$ $g_2(x) = 2x_1 + 2x_3 + x_{10} + x_{12} - 10 \leq 0$ $g_3(x) = 2x_2 + 2x_3 + x_{11} + x_{12} - 10 \leq 0$ $g_4(x) = -8x_1 + x_{10} \leq 0$ $g_5(x) = -8x_2 + x_{11} \leq 0$ $g_6(x) = -8x_3 + x_{12} \leq 0$ $g_7(x) = -2x_4 - x_5 + x_{10} \leq 0$ $g_8(x) = -2x_6 - x_7 + x_{11} \leq 0$ $g_9(x) = -2x_8 - x_9 + x_{12} \leq 0$	$0 \leq x_i \leq 1$ ($i = 1, 2, \cdots, 9, 13$), $0 \leq x_i \leq 100$ ($i = 10, 11, 12$)	-15
g06	$f(x) = \frac{\sin^3(2\pi x_1) + \sin(2\pi x_2)}{x_1^3(x_1 + x_2)}$	$g_1(x) = x_1^2 - x_2 + 1 \leq 0$ $g_2(x) = 1 - x_1 + (x_2 - 4)^2 \leq 0$	$0 \leq x_1 \leq 10$ $0 \leq x_2 \leq 10$	-0.095 825
g07	$f(x) = x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 + (x_3 - 10)^2 + 4(x_4 - 5)^2 + (x_5 - 3)^2 + 2(x_6 - 1)^2 + 5x_7^2 + 7(x_8 - 11)^2 + 2(x_9 - 10)^2 + (x_{10} - 7)^2 + 45$	$g_1(x) = 4x_1 + 5x_2 - 3x_7 + 9x_8 - 105 \leq 0$ $g_2(x) = 10x_1 - 8x_2 - 17x_7 + 2x_8 \leq 0$	$-10 \leq x_i \leq 10$ ($i = 1, \cdots, 10$)	24.306
g08	$f(x) = (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 + 10x_5^6 + 7x_6^2 + x_7^4 + 4x_6x_7 - 10x_6 - 8x_7$	$g_1(x) = 2x_1^2 + 3x_2^4 + x_3 + 4x_4^2 + 5x_5 - 127 \leq 0$ $g_2(x) = 7x_1 + 3x_2 + 10x_3^2 + x_4 - x_5 - 282 \leq 0$ $g_3(x) = 23x_1 + x_2^2 + 6x_6^6 - 8x_7 - 196 \leq 0$ $g_4(x) = 4x_1^2 + x_2^2 - 3x_1x_2 + 2x_3^2 + 5x_6 - 11x_7 \leq 0$	$-10 \leq x_i \leq 10$ ($i = 1, \cdots, 7$)	680.63

根据表 2 中结果,将不同算法对每一个标准函数求出的最优可行解加粗,CDRAO 算法在八个测试函数上的优化结果均为最优的效果。而 CGWO、E-BRM 和 MGABC 都无法完全寻优出文中测试算例的最优,且 CGWO、E-BRM 无法在部分算例上找

出可行域,导致寻优结果无效,MGABC 算法不能够优化出 g05 函数和 g07 函数的全局最优。

CDRAO 算法针对上述测试函数均能得到约束全局最优,且结果稳定性能够满足要求。针对每个函数迭代运算 25 次,计算出迭代收敛时迭代

表2 约束采样方法算例测试结果

Tab.2 Constraint sampling method study test results

标准函数	统计量	g01	g02	g03	g04	g05	g06	g07	g08
CDRAO	平均值	-6 961.81	0.749 9	-5.508	-30 665.5	-15	-0.095 8	24.306	680.63
	标准差	2.4×10^{-4}	0	0	1.50×10^{-2}	1.3×10^{-4}	0	2.1×10^{-5}	0
CGWO	平均值	-6 591.48	0.719 0	-5.308 7	-31 492.2	-14.900 5	-321.04	60.227 8	676.670
	标准差	56.4	1.28×10^{-2}	3.08	137	6.30×10^{-3}	6.25	0.121	3.08
E-BRM	平均值	-6 991.75	0.880 5	-5.508	-30 953.5	-14.998 5	-0.095 8	42.132 4	603.816
	标准差	5.10×10^{-2}	7.95×10^{-2}	0	63	6×10^{-4}	0	7.215×10^{-3}	0
MGABC	平均值	-6 959.49	0.750 0	-5.508	-30 665.5	-13.553 5	-0.095 8	27.654 0	680.63
	标准差	1.18	0	0	0	16.4	0	1.33×10^{-2}	0

次数的平均值,每次计算设置的样本点个数保持不变,采用 CDRAO 算法对 8 个测试函数优化的迭代次数如表 3 所示。

表3 CDRAO 算法迭代次数统计

Tab.3 Statistics on the number of iterations of CDRAO algorithm

标准函数	样本点个数	收敛时迭代次数
g01	5	143
g02	5	87
g03	5	134
g04	10	122
g05	27	145
g06	5	65
g07	20	167
g08	15	145

相比于文献中其他算法的收敛速率,CDRAO 算法收敛速率在 200 次以内,且低维函数和高维函数次数相差不大,适用性较强,表明提出的 CDRAO 算法是性能优越的约束优化算法,拥有进行复杂工程约束优化应用的潜力。

3.2 固体发动机后翼柱装药设计算例

采用 CDRAO 方法对固体火箭发动机装药设计进行优化,优化目标为药柱燃烧过程燃面稳定性,约束条件为发动机药柱质量,以验证算法的有效性。

3.2.1 计算模型

仿真模型采用固体火箭发动机性能匹配框架^[22]实现。药柱形状选用翼柱形,主要考虑后翼柱装药,图 9 展示了算例的装药构型结构。优化

设计变量选择后翼柱的 5 个几何结构参数,其中包含 4 个连续变量和 1 个离散变量,其类型及变量范围如表 4 所示。除此之外,其他常数参数设置见表 4。装药几何结构可由表中设计变量唯一确定。

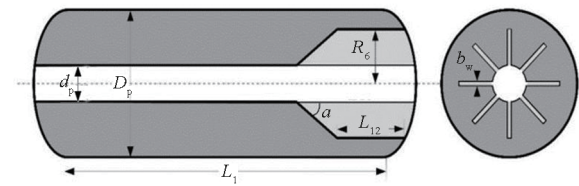


图9 后翼柱装药结构示意图

Fig.9 Schematic diagram of the rear wing pillar loading structure

表4 后翼柱型装药设计变量

Tab.4 Rear wing column type charge design variables

变量类型	参数	设计区间(值)
设计变量	后翼长度 L_{12} /mm	[100,800]
	后翼宽度 b_w /mm	[15,30]
	后翼深度 R_6 /mm	[45,130]
	后翼倾角 $\alpha/(^\circ)$	[0.15,30]
	后翼个数(离散)	[1,12]
其他变量	装药外径 D_p /mm	300
	柱段长度 L_1 /mm	2 000
	装药内径 d_p /mm	75

为了量化燃面设计的稳定性,定义算例优化问题为:

$$\begin{aligned} \min f_{\text{STD}}(A(X)) \\ \text{s. t. } m > m_0 \end{aligned} \tag{12}$$

式中, X 表示装药几何构型设计变量, $A(X)$ 表示根据当前构型计算出燃面燃烧厚度变化曲线, f_{STD} 表示计算燃面曲线的标准差, m 为装药质量, m_0 为初始装药质量最小值。本算例中 m_0 取 240 kg。

3.2.2 计算结果

初始算法种群设置为 10,优化过程中目标函数的收敛情况和优化结果的燃面曲线如图 10、图 11 所示。

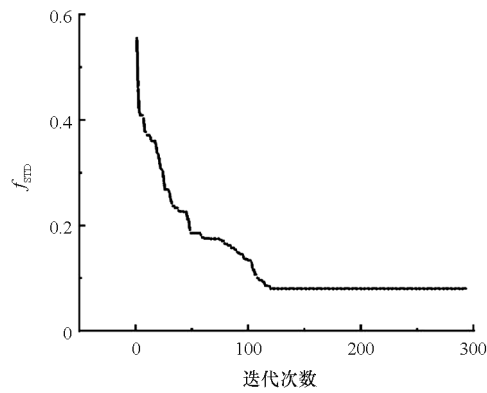


图 10 f_{STD} 历史收敛曲线

Fig. 10 f_{STD} historical convergence curve

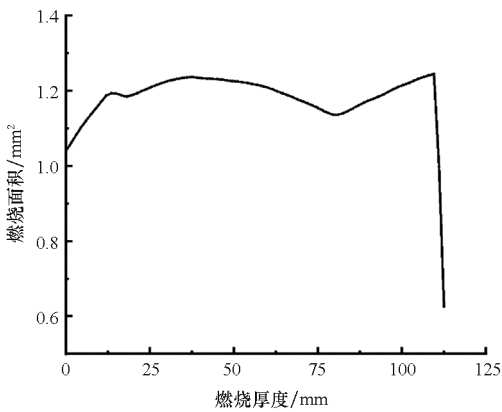


图 11 后翼柱装药设计燃面曲线

Fig. 11 Rear wing pillar charge design combustion surface curve diagram

最终优化设计结果如表 5 所示。

表 5 后翼柱型装药优化结果

Tab.5 Optimized results for rear wing pillar type loading		
变量类型	参数	优化结果
设计变量	后翼长度 L_{12}/mm	392
	后翼宽度 b_w/mm	17.4
	后翼深度 R_6/mm	135
	后翼倾角 $a/(\text{°})$	16.59
	后翼个数(离散)	5
约束变量	质量/kg	240.24

由图 10、图 11 得出,CDRAO 方法能够快速设计出燃面稳定的后翼柱型装药结构,迭代次数仅需 120 多次,展现出其高效性。而优化设计结果中装药质量为 240.24 kg,满足模型中关于约束条件的限制。证明 CDRAO 方法对约束处理的可靠性,能够应用于实际工程问题中。

4 结论

针对高耗时模型优化设计问题,对代理模型自适应采样阶段开展研究。考虑目标函数存在的各种类型约束,提出了多约束自适应采样方法,通过数学算例和工程算例对文中提出方法展开验证。

1)在基于约束冲突函数处理约束问题的基础上提出基于 ε - 约束保持改进的可行性规则法,制定“先探索可行域,再寻优”的二阶段采样策略,增强算法探索可行域附近高质量不可行解的能力,提升算法对约束边界的探索精度。

2)提出了基于改进可行性规则的多约束自适应采样方法,通过引入精英档案的思想迭代更新可行域的高质量解,通过缩小 ε 的值不断使当前域逼近于可行域,避免了优化过程中采样可行域边界精度不够的问题,从而提高了优化结果的准确性。

3)采用 CEC 优化约束标准函数对 CDRAO 方法进行测试,并与其他算法进行比对,得出算法能够较好地得出约束函数的最优值且具有较高的稳定性。将 CDRAO 方法应用到固体火箭发动机后翼柱装药设计算例中,得出满足设计要求的优化设计结果。表明文章提出的方法可以应用于飞行器设计方面的优化问题。

参考文献(References)

[1] 李正良,彭思思,王涛. 基于混合加点点准则的代理模型优化设计方法[J]. 工程力学, 2022, 39(1): 27-33.
LI Z L, PENG S S, WANG T. A surrogate-based optimization design method based on hybrid infill sampling criterion[J]. Engineering Mechanics, 2022, 39 (1): 27 - 33. (in Chinese)

[2] WANG W J, WU Z P, WANG D H, et al. Hypersonic vehicle aerodynamic optimization using field metamodel-enhanced sequential approximate optimization [J]. International Journal of Aerospace Engineering, 2021, 2021: 8889593.

[3] WANG X J, NI B W, ZENG L X, et al. An adaptive sampling strategy for construction of surrogate aerodynamic model[J]. Aerospace Science and Technology, 2021, 112: 106594.

[4] SU Y P, XU L H, GOODMAN E D. Hybrid surrogate-based constrained optimization with a new constraint-handling

method [J]. IEEE Transactions on Cybernetics, 2022, 52(6): 5394 – 5407.

[5] HUANG D, ALLEN T T, NOTZ W I, et al. Global optimization of stochastic black-box systems via sequential Kriging meta-models [J]. Journal of Global Optimization, 2006, 34(3): 441 – 466.

[6] PARK J, SANDBERG I W. Universal approximation using radial-basis-function networks [J]. Neural Computation, 1991, 3(2): 246 – 257.

[7] FORRESTER A I J, KEANE A J. Recent advances in surrogate-based optimization [J]. Progress in Aerospace Sciences, 2009, 45(1/2/3): 50 – 79.

[8] CHEN G D, ZHANG K, XUE X M, et al. A radial basis function surrogate model assisted evolutionary algorithm for high-dimensional expensive optimization problems [J]. Applied Soft Computing, 2022, 116: 108353.

[9] JIN R C, CHEN W, SUDJANTO A. An efficient algorithm for constructing optimal design of computer experiments[J]. Journal of Statistical Planning and Inference, 2005, 134(1): 268 – 287.

[10] WANG Q Y, NAKASHIMA T, LAI C G, et al. Modified algorithms for fast construction of optimal Latin-hypercube design[J]. IEEE Access, 2020, 8: 191644 – 191658.

[11] KRIGE D G. A statistical approach to some basic mine valuation problems on the Witwatersrand[J]. Journal of the Southern African Institute of Mining and Metallurgy, 1951, 52(6): 119 – 139.

[12] RASMUSSEN C E. Evaluation of Gaussian processes and other methods for non-linear regression [D]. Toronto: University of Toronto, 1997.

[13] BENGIO Y, LAMBLIN P, POPOVICI D, et al. Greedy layer-wise training of deep networks[M]//SCHÖLKOPF B, PLATT J, HOFMANN T. Advances in Neural Information Processing Systems 19: Proceedings of the 2006 Conference, Cambridge: MIT Press, 2007: 153 – 160.

[14] TSAI C C, HUANG H C, LIN S C. Adaptive neural network control of a self-balancing two-wheeled scooter[J]. IEEE Transactions on Industrial Electronics, 2010, 57(4): 1420 – 1428.

[15] BERKEMEIER M, PEITZ S. Derivative-free multiobjective trust region descent method using radial basis function surrogate models [J]. Mathematical and Computational Applications, 2021, 26(2): 31.

[16] 武泽平. 序列近似优化方法及其应用研究[D]. 长沙: 国防科学技术大学, 2013.

WU Z P. Study on sequential approximate optimization and its application[D]. Changsha: National University of Defense Technology, 2013. (in Chinese)

[17] 李雅丽, 王淑琴, 陈倩茹, 等. 若干新型群智能优化算法的对比研究[J]. 计算机工程与应用, 2020, 56(22): 1 – 12.

LI Y L, WANG S Q, CHEN Q R, et al. Comparative study of several new swarm intelligence optimization algorithms[J]. Computer Engineering and Applications, 2020, 56(22): 1 – 12. (in Chinese)

[18] 潘燕娜, 冯翔, 虞慧群. 基于自适应资源分配池的竞争合作群协同优化算法[J]. 计算机科学, 2022, 49(2): 182 – 190.

PAN Y N, FENG X, YU H Q. Competitive-cooperative coevolution for large scale optimization with computation resource allocation pool [J]. Computer Science, 2022, 49(2): 182 – 190. (in Chinese)

[19] KOHLI M, ARORA S. Chaotic grey wolf optimization algorithm for constrained optimization problems[J]. Journal of Computational Design and Engineering, 2018, 5(4): 458 – 472.

[20] DE CASTRO RODRIGUES M, GUIMARÃES S, DE LIMA B S L P. E-BRM: a constraint handling technique to solve optimization problems with evolutionary algorithms [J]. Applied Soft Computing, 2018, 72: 14 – 29.

[21] BANSAL J C, JOSHI S K, SHARMA H. Modified global best artificial bee colony for constrained optimization problems[J]. Computers & Electrical Engineering, 2018, 67: 365 – 382.

[22] WU Z P, WANG D H, ZHANG W H, et al. Solid-rocket-motor performance-matching design framework[J]. Journal of Spacecraft and Rockets, 2017, 54(3): 698 – 707.