Scaling Bayesian Optimization for High-Dimensional and Large-Scale Constrained Spaces

Maathuis, Hauke F.; Breuker, Roeland De; Castro, Saullo G. P.

DOI
10.2514/1.J065252

Publication date
2025

Document Version
Final published version

Published in
AIAA Journal

Scaling Bayesian Optimization for High-Dimensional and Large-Scale Constrained Spaces

Hauke F. Maathuis,* Roeland De Breuker,† and Saullo G. P. Castro‡
Delft University of Technology, 2629HS Delft, The Netherlands

https://doi.org/10.2514/1.J065252

Design optimization offers the potential to develop lightweight aircraft structures with reduced environmental impact. Due to the high number of design variables and constraints, these challenges are typically addressed using gradient-based optimization methods to maintain efficiency, however overlooking the global design space. Moreover, gradients are frequently unavailable. Bayesian optimization presents a promising gradient-free alternative, enabling sample-efficient global optimization through probabilistic surrogate models. Although Bayesian optimization has shown its effectiveness for problems with a small number of design variables, it struggles to scale to high-dimensional problems, particularly when incorporating large-scale constraints. This challenge is especially pronounced in aeroelastic tailoring, where directional stiffness properties are integrated into the structural design to manage aeroelastic deformations and enhance both aerodynamic and structural performance. Ensuring the safe operation of the system requires simultaneously addressing constraints from various analysis disciplines, making global design space exploration even more complex. This study seeks to address this issue by employing high-dimensional Bayesian optimization combined with dimensionality reduction to tackle the optimization challenges in aeroelastic tailoring. The proposed approach is validated through experiments on a well-known benchmark case, as well as its application to the aeroelastic tailoring problem, demonstrating the feasibility of Bayesian optimization for high-dimensional problems with large-scale constraints.

I. Introduction

THE design of modern aircraft with enhanced efficiency is crucial for enabling more sustainable aviation. Achieving this involves optimizing structural designs to reduce energy consumption. Aeroelastic tailoring emerges as a key technique that has the potential to reduce the weight of aeroelastically efficient high-aspect-ratio wings. Pioneered by Shirk et al. [1], aeroelastic tailoring incorporates directional stiffness properties to effectively carry and control the aeroelastic deformations. Performing aeroelastic tailoring is a multidisciplinary design and optimization (MDO) effort, involving aerodynamics for the outer-mold- shape definition and calculation of the loads acting over the wing, structural design that usually defines the layout of the main structural components of the wingbox, structural analysis to define and evaluate the relevant failure modes that should be considered as constraints, aeroelasticity that couples the aerodynamic loads with the inertial and elastic properties of the wing to characterize the flutter behavior, and optimization to properly explore the design space. Other disciplines are also involved, such as manufacturing, typically resulting in additional constraints for the design variables.

Evaluating these complex aeroelastic models is computationally expensive, therefore necessitating efficient optimization algorithms that require fewer analyses before finding an optimum solution. Due to the high number of design variables describing the structural properties of the system, gradient-based optimization algorithms are commonly used, leading to an efficient convergence toward the optimal solution. However, the computation of gradients is not always feasible, especially if the model's source code is unavailable. In such cases, the model must be treated as a black box, relying on methods such as finite differences to obtain the design sensitivities, which can lead to prohibitively high computational costs that would ultimately motivate the use of gradient-free methods. Furthermore, many engineering problems, such as noisy responses or experimental results, possess inherent complexities that can render gradient-based approaches less effective or even impractical. Additionally, the response surface for feasible designs in aeroelastic tailoring is often multimodal. This complexity can cause gradient-based methods to become trapped in local optima, overlooking the broader global design space and hindering the discovery of superior designs. Therefore, it is essential to develop methods that efficiently explore the global design space, optimizing structures to achieve lighter aircraft configurations.

The optimization problem at hand can be formulated as follows:

$$\min_{\boldsymbol{x}\in\mathcal{X}\subset\mathbb{R}^D} f(\boldsymbol{x}) \text{ s.t. } \forall j \in \{1,\ldots,G\}, c_j(\boldsymbol{x}) \leq 0 \tag{1}$$

where $\mathcal{X} \subset \mathbb{R}^D$ is a D-dimensional space of potential designs, $f(\boldsymbol{x}) : \boldsymbol{x} \in \mathcal{X} \rightarrow \mathbb{R}$ the objective function, and G constraints arising from the multidisciplinary analyses. Overall, aeroelastic tailoring can be seen as an optimization problem consisting of high-dimensional inputs and outputs, where the utilized models are able to map the vector of design variables to the objective function $f(\boldsymbol{x}) \in \mathbb{R}$ and all G constraints $\boldsymbol{c}(\boldsymbol{x}) \in \mathbb{R}^G$.

The simultaneous consideration of multiple disciplines can lead to large-scale constraints where $G \gg 10^3$, combining buckling, aeroelastic stability, maximum stress, maximum strain, and various others. In aeroelastic tailoring, the optimal stiffness distribution is achieved by means of a sizing optimization that, in the case of laminated composite wings, consists of finding the best set of lamination parameters and the optimum thickness for one or more composite regions [2]. Lamination parameters allow a condensed and theoretical representation of the membrane, bending, and coupled stiffness terms of a laminate with continuous variables [3], making the sizing optimization more convex and more adequate to established continuous optimization techniques, where the design variables can be treated as continuous variables. Once this sizing optimization is complete, a second discrete optimization is performed to retrieve a manufacturable set of ply orientations. Yet, the presence of multiple design regions to maintain design freedom can still result in the number of design variables being in the order of hundreds or thousands.

Presented as Paper 2024-2012 at the AIAA SCITECH 2024 Form, Orlando, FL, January 8–12, 2024; received 20 December 2024; accepted for publication 26 April 2025; published online 30 July 2025.

*Ph.D. Candidate, Faculty of Aerospace Engineering, Kluyverweg 1; h.f.maathuis@tudelft.nl (Corresponding Author).

†Associate Professor, Faculty of Aerospace Engineering, Kluyverweg 1; r.debreuker@tudelft.nl. Associate Fellow AIAA.

‡Associate Professor, Faculty of Aerospace Engineering, Kluyverweg 1; s.g.p.castro@tudelft.nl.

Given the expensive nature of evaluating an aeroelastic model to obtain the objective function values and associated constraints, a sample-efficient optimization algorithm is crucial. Compared to other gradient-free approaches such as Random Search, Genetic Algorithms, and others, Bayesian optimization (BO) has proven to be a powerful method for complex and computationally costly problems [4] and has been extensively applied across various domains, including aircraft design [5]. BO addresses the challenge of expensive evaluations by using computationally inexpensive probabilistic surrogate models, such as Gaussian Processes ($\mathcal{GP}$). These models replace the black-box functions representing the objective and constraints, significantly improving optimization efficiency [6]. Although many authors have shown that for lower-dimensional problems BO methods perform well, high-dimensional cases pose significant challenges due to the curse of dimensionality [7,8], resulting from the fact that high-dimensional search spaces are difficult to explore exhaustively. However, BO offers a probabilistic approach to efficiently search the design space to find promising regions and to reduce the computational burden. Although these algorithms offer a variety of advantages, including the learning-from-data aspect, uncertainty quantification, the lack of need for gradients, the ability to fuse data in a multifidelity context, and the capability to learn the correlation between simulation and experimental data, their scalability to high-dimensional problems with many constraints, as is often the case in engineering design, remains a significant challenge.

The present study focuses on employing high-dimensional BO algorithms for aeroelastic tailoring while considering large-scale constraints arising from the multidisciplinary analyses, as formulated in Eq. (1). The novelty of this paper lies in the application of a high-dimensional BO method with a dimensionality reduction approach that significantly lowers the computational burden arising from the incorporation of a large number of constraints. Subsequently, the methodology is applied to the $7D$ speed reducer benchmark problem with 11 black-box constraints [9] before its application to aeroelastic tailoring is presented.

The structure of the paper is as follows. First, Sec. II introduces $\mathcal{GP}$s as a probabilistic surrogate modeling technique, as well as BO for both unconstrained and constrained problems, highlighting scalability challenges. Section III then explores dimensionality reduction in the context of constrained BO, followed by a discussion of the numerical results in Sec. IV. Finally, the paper concludes with a discussion and directions for future work in Sec. V.

II. High-Dimensional Constrained Bayesian Optimization

This section briefly introduces BO within the context of high dimensionality and constraints. $\mathcal{GP}$s are introduced as the herein employed surrogate modeling technique. Subsequently, $\mathcal{GP}$s are linked to unconstrained BO, which is then expanded to address the constrained scenario, followed by an outline of the challenges encountered in this work.

A. Gaussian Processes

A $\mathcal{GP}$ in the context of BO serves as a probabilistic surrogate model that efficiently represents an unknown function $f(\boldsymbol{x})$. Recall that $\mathcal{X} \subset \mathbb{R}^D$ is a D-dimensional domain and the corresponding minimization problem is presented in Eq. (1), beginning with a Design of Experiments (DoE) denoted by $\mathcal{D}_0 = \{\boldsymbol{x}_i, f(\boldsymbol{x}_i)\}_{i=1,\ldots,N}$, where $\boldsymbol{x}_i \in \mathcal{X} \subset \mathbb{R}^D$ is the ith of N samples and $f(\boldsymbol{x}_i): \mathcal{X} \to \mathbb{R}$ the objective function, mapping from the design space to a scalar value. $\mathcal{GP}$s are commonly employed within BO to construct a surrogate model $\hat{f}(\boldsymbol{x}): \mathcal{X} \to \mathbb{R}$ of the objective function f from this given data set $\mathcal{D}$. Therefore, it is assumed that the objective function f follows a $\mathcal{GP}$, which is also called a multivariate normal distribution $\mathcal{N}$. By defining the mean $m: \mathcal{X} \to \mathbb{R}$ and covariance $k: \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, a noise-free surrogate can thus be denoted as

$$f(\boldsymbol{x})|\mathcal{D} \sim \mathcal{GP}(m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}')) \tag{2}$$

also known as the prior. The prior encapsulates the initial belief that observations are normally distributed. A common choice for the covariance function, also called kernel, is the squared exponential kernel $k(\boldsymbol{x}, \boldsymbol{x}')$ defined as

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sigma_f \exp\left(-\frac{1}{2}\sum_{i=1}^{D}\left(\frac{x_i - x_i'}{l_i}\right)^2\right) \tag{3}$$

which encodes the similarity between two chosen points $\boldsymbol{x}$ and $\boldsymbol{x}'$ [10]. The parameter l_i for $i = 1, \ldots, D$ is called the length scale and measures the distance for the correlation along x_i. Together with σ_f, often called the signal variance, the parameters form a set of so-called hyperparameters $\boldsymbol{\theta} = \{l_1, \ldots, l_D, \sigma_f\}$, in total $D + 1$ parameters, which need to be determined to train the model with respect to the target function. The kernel matrix is defined as $\boldsymbol{K} = [k(\boldsymbol{x}_i, \boldsymbol{x}_j)]_{i,j=1,\ldots,N} \in \mathbb{R}^{N \times N}$. The kernel needs to be defined such that $\boldsymbol{K}$ is symmetric positive definite to ensure its invertibility. The positive definite symmetry is guaranteed if and only if the used kernel is a positive definite function, as detailed in Schoenberg [11]. The values of the hyperparameters $\boldsymbol{\theta}$ are determined by maximizing the marginal likelihood, written as

$$\log\, p(\boldsymbol{f}|\mathcal{D}, \boldsymbol{\theta}) = -\frac{1}{2}\boldsymbol{f}^\top \boldsymbol{K}^{-1}\boldsymbol{f} - \frac{1}{2}\log|\boldsymbol{K}| - \frac{N}{2}\log 2\pi \tag{4}$$

Computing the partial derivative with respect to the hyperparameters $\boldsymbol{\theta}$ gives

$$\frac{\partial}{\partial \theta_j}\log\, p(\boldsymbol{f}|\mathcal{D}, \boldsymbol{\theta}) = \frac{1}{2}\boldsymbol{f}^\top \boldsymbol{K}^{-1}\frac{\partial \boldsymbol{K}}{\partial \theta_j}\boldsymbol{K}^{-1}\boldsymbol{f} - \frac{1}{2}\mathrm{tr}\left(\boldsymbol{K}^{-1}\frac{\partial \boldsymbol{K}}{\partial \theta_j}\right) \tag{5}$$

which can be used within a gradient-based optimization for model selection, or in other words, hyperparameter tuning. More detailed information can be found in Rasmussen and Williams [10].

Considering a new query point $\boldsymbol{x}_+ \in \mathcal{X}$, the stochastic process in Eq. (2) can be used to predict the new query point

$$f(\boldsymbol{x}_+)|\mathcal{D} \sim \mathcal{N}(\mu(\boldsymbol{x}_+), k(\boldsymbol{x}_+, \boldsymbol{x}_+)) \tag{6}$$

The posterior mean $\mu(\bullet)$ and covariance function $\sigma(\bullet)$ are computed by

$$\mu(\boldsymbol{x}_+) = \boldsymbol{k}(\boldsymbol{x}_+, \boldsymbol{X})\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X})^{-1}\boldsymbol{f} \tag{7}$$

$$\sigma(\boldsymbol{x}_+) = k(\boldsymbol{x}_+, \boldsymbol{x}_+) - \boldsymbol{k}(\boldsymbol{x}_+, \boldsymbol{X})\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X})^{-1}\boldsymbol{k}(\boldsymbol{X}, \boldsymbol{x}_+) \tag{8}$$

where $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N] \subset \mathcal{D}$ is the collection of samples and $\boldsymbol{f} = [f_1, f_2, \ldots, f_N] \subset \mathcal{D}$ of computed objective values in $\mathcal{D}$.

B. Unconstrained Bayesian Optimization

Up to this stage, the $\mathcal{GP}$ has been computed using the initial samples contained in $\mathcal{D}_0$. BO now proceeds iteratively to enhance the accuracy of the surrogate model by enriching $\mathcal{D}$ while exploring the design space. Thus, leveraging the acquired data, the endeavor is to identify regions expected to yield optimal values. The problem at hand can be written as

$$\min_{\boldsymbol{x} \in \mathcal{X}} f(\boldsymbol{x}) \tag{9}$$

An acquisition function $\alpha: \mathcal{X} \to \mathbb{R}$ is used to guide the optimization through the design space while trading off exploration and exploitation based on the posterior mean and variance defined in Eq. (7). The former describes the exploration of the whole design space, whereas the latter tries converging to an optimum based on the data observed. This can be written as

$$\boldsymbol{x}_+ \in \underset{\boldsymbol{x} \in \mathcal{X}}{\mathrm{argmax}}\, \alpha(\boldsymbol{x}|\mathcal{D}) \tag{10}$$

Numerous acquisition functions exist, often making use of the predictive mean $\hat{\mu}(\boldsymbol{x})$ and variance $\hat{\sigma}(\boldsymbol{x})$. Popular choices for such an acquisition function are, for example, Expected Improvement (EI) [12] or Thompson Sampling (TS) [13].

C. Constrained Bayesian Optimization

Most engineering design problems involve constraints, which can be integrated into the earlier introduced BO method, discussed in, for example, Gardner et al. [14], Gelbart et al. [15], and Hernández-Lobato et al. [16]. This assumes that the output of a model evaluation at design point $\boldsymbol{x}_i$ includes not only the objective function $f(\boldsymbol{x}_i)$, but also a mapping from the design space to a collection of G constraints $\boldsymbol{c}(\boldsymbol{x}_i): \mathcal{X} \to \mathbb{R}^G$. Consequently, the DoE for this scenario is represented as $\mathcal{D} = \{\boldsymbol{x}_i, f(\boldsymbol{x}_i), \boldsymbol{c}(\boldsymbol{x}_i)\}_{i=1,\ldots,N}$. The new design point found needs to lie in the feasible space $\mathcal{X}_f$, written as $\boldsymbol{x}_+ \in \mathcal{X}_f \subset \mathcal{X}$, where $\mathcal{X}_f \coloneqq \{\boldsymbol{x} \in \mathcal{X} | c_j(\boldsymbol{x}) \leq 0, j = 1, \ldots, G\}$. Gardner et al. [14] propose modeling each constraint $c_j(\boldsymbol{x}), j = 1, \ldots, G$ with an independent surrogate model, akin to how the objective function is modeled:

$$c_j(\boldsymbol{x}) | \mathcal{D} \sim \mathcal{GP}(m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}')) \tag{11}$$

leading to $G + 1\mathcal{GP}$ models in total. Accordingly, these surrogate models can then be used within a constrained acquisition strategy, solving the optimization problem formulated as

$$\boldsymbol{x}_+ \in \underset{\boldsymbol{x} \in \mathcal{X}_f \subset \mathcal{X}}{\operatorname{argmax}} \alpha_c(\boldsymbol{x} | \mathcal{D}) \tag{12}$$

where α_c denotes a constrained acquisition function. This subsection serves to introduce the fundamental aspects of constrained BO concisely, emphasizing that each constraint must be modeled via a separate $\mathcal{GP}$ model. Of course, a multitude of constrained acquisition functions exist. Among these approaches, for instance, is the use of Thompson Sampling [13] as an acquisition function [17], extended to the constrained setting in Eriksson et al. [18]. A major advantage is its scalability to larger batch sizes. The latter study also demonstrates the superiority of this approach compared to constrained EI, which is why constrained TS is employed in the course of this work and is explained in Algorithm 1. Therein, for each $\mathcal{GP}$ used for modeling the objective function and the G constraints, the posterior is computed. For a batch size of Q points, a sample is drawn to get realizations of the surrogate models. Then, N_c candidate points are evaluated on the $\mathcal{GP}$s to obtain either a set of feasible points with optimal objective value or points with a minimum total constraint violation $\boldsymbol{X}_+$.

Algorithm 1: Constrained Thompson Sampling

Input: $\mathcal{D}_k$ of kth iteration, Q batch size, $\boldsymbol{X}_c = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{N_c}]$ with N_c candidates
while Computational budget is not exhausted **do**
  $\boldsymbol{X}_+ = \{\}$
  Compute current posterior $\mathrm{p}(\boldsymbol{\theta}|\mathcal{D}_k)$ for $f, c_1, \ldots, c_G$
  for $q = 1{:}\mathrm{Q}$ **do**
    Sample $\boldsymbol{\theta}$ from $p(\boldsymbol{\theta}|\mathcal{D}_k)$ to obtain realisations for $\hat{f}, \hat{c}_1, \ldots, \hat{c}_G$
    Evaluate $\{\boldsymbol{x}_i | i \in \mathbb{N}, 1 \leq i \leq N_c\}$ on $\hat{f}(\boldsymbol{x}_i), \hat{c}_1(\boldsymbol{x}_i), \ldots, \hat{c}_G(\boldsymbol{x}_i)$ from the respective posterior distribution
    Obtain $\hat{f}(\boldsymbol{x}_i), \hat{c}_1(\boldsymbol{x}_i), \ldots, \hat{c}_G(\boldsymbol{x}_i)$
    Choose $\mathbb{X}_f = \{\boldsymbol{x}_i | \hat{c}_l(\boldsymbol{x}_i) \leq 0 \text{ for } 1 \leq l \leq G\}$
    if $\mathbb{X}_f \neq \varnothing$ **then** $\boldsymbol{x}_+^q = \operatorname{argmax}_{\boldsymbol{x} \in \mathbb{X}_f} \hat{f}(\boldsymbol{x})$
    else Obtain the minimum of total violation by computing $\boldsymbol{x}_+^q = \arg\min_{\boldsymbol{x} \in \boldsymbol{X}_c} \sum_{i=1:G} \max(\hat{c}_i(\boldsymbol{x}), 0)$
    end if
    $\boldsymbol{X}_+ = \boldsymbol{X}_+ \cup \{\boldsymbol{x}_+^q\}$
  end for
end while

D. High-Dimensional Bayesian Optimization: Challenges and Advances

BO algorithms consist of two main components, namely the probabilistic surrogate model, $\mathcal{GP}$s, which are based on Bayesian statistics [10], and an acquisition function to guide the selection where to query the next point to converge toward the minimizer of the objective function. Although these algorithms have been proven to be very efficient for lower-dimensional problems [19], scaling them to higher dimensions implies some difficulties:

1) The curse of dimensionality dictates that as the number of dimensions increases, the size of the design space grows exponentially, making an exhaustive search impractical.

2) With higher dimensions, there is an increase in the number of tunable hyperparameters $\boldsymbol{\theta} \in \mathbb{R}^{D+1}$, resulting in a more cumbersome $\mathcal{GP}$ model learning, possibly leading to increased uncertainty.

3) Higher-dimensional problems necessitate more samples N to construct an accurate surrogate model. The inversion of the covariance matrix $\boldsymbol{K} \in \mathbb{R}^{N \times N}$ becomes computationally intensive with a complexity for inference and learning of $\mathcal{O}(N^3)$ and $\mathcal{O}(N^2)$ for memory.

4) Insufficient data collection results in sparse sampling across the D-dimensional hyperspace, causing samples to be widely dispersed from each other. This dispersion hinders effective correlation among the samples.

5) Acquisition function optimization faces increased uncertainty in high-dimensional settings, requiring more evaluations of the surrogate model [19].

Various strategies have been employed to address the challenge of high-dimensional input spaces in scenarios with few or no constraints. In Wang et al. [20], random projections are utilized to reduce high-dimensional inputs to a lower-dimensional subspace, allowing for the construction of the $\mathcal{GP}$ model directly in this reduced space, thereby reducing the number of hyperparameters. Similarly, Raponi et al. [21] and Antonov et al. [22] employ (kernel) Principal Component Analysis on the input space to identify a reduced set of dimensions based on evaluated samples, followed by training the surrogate model in this reduced dimensional space. In contrast, Eriksson and Jankowiak [7] adopt a hierarchical Bayesian model that assumes varying importance among design variables, using a sparse axis-aligned prior on the length scale to discard dimensions unless supported by accumulated data. However, Santoni et al. [23] demonstrates high computational overhead in this approach. Additionally, decomposition techniques, such as additive methods, are employed to partition the original space, as demonstrated in Kandasamy et al. [24] and Ziomek and Bou-Ammar [25].

The Trust-Region Bayesian Optimization (TuRBO) algorithm, described in Eriksson et al. [18], takes a different route, where the design space is partitioned into multiple independent trust regions (TRs). Results from Eriksson et al. [18] demonstrate promising outcomes for this approach, particularly in high-dimensional problems where gathering sufficient data to construct a globally accurate surrogate model is challenging due to the curse of dimensionality. Instead, surrogates are focused on these defined TRs, which adjust in size during optimization. TRs are defined as hyperrectangles of size $L \in \mathbb{R}$, centered at the best solution found so far and initialized with $L \leftarrow L_{\text{init}}$, a user-defined parameter. The size L_{TR} of each TR is determined using the length scale l_i of the $\mathcal{GP}$, defined in Eq. (3), and a base length scale L:

$$L_{\text{TR}} = \frac{l_i L}{\left(\prod_{j=1}^{D} l_j\right)^{1/D}} \tag{13}$$

In each optimization iteration, a batch of q samples are drawn within the TR. When the design space is normalized to $\mathcal{X} \in [-1, 1]$ and L spans the entire design space with $L \to 2$ kept constant, the TR approach resembles a standard BO algorithm as outlined in Frazier [6]. The evolution of L significantly influences the convergence of this method, and specific hyperparameters governing its adaptation are detailed in Eriksson et al. [18].

All the algorithms discussed earlier focus exclusively on unconstrained optimization problems. The TR approach, however, addresses constraints explicitly by adapting the batched Thompson Sampling method from Thompson [13] as an acquisition function for constrained problems [26], detailed in Algorithm 1. This method, known as scalable constrained Bayesian optimization (SCBO), employs separate $\mathcal{GP}$s to model each constraint within the current TR. Scaling BO to high-dimensional problems necessitates addressing significant challenges through specific assumptions. Although existing approaches demonstrate promising results, handling large-scale constraints, such as those encountered in aircraft design problems where $G > 10^3$, remains insufficiently addressed. This work adopts the constrained TuRBO algorithm SCBO for high-dimensional BO due to its explicit treatment of constraints. Next, an extension of this method is introduced to address the challenge posed by large-scale constraints.

III. Large-Scale Constrained Bayesian Optimization via Latent- Space Gaussian Processes

Recall the optimization problem formulated in Eq. (1). By using constrained BO methods, as shown earlier, each of the G constraints needs to be modeled with an independent $\mathcal{GP}$, denoted as $\hat{c}_j(\boldsymbol{x})$. This work follows the idea of Higdon et al. [27] to construct the surrogates on a lower-dimensional, latent output space. Let $\mathcal{V} \subset \mathbb{R}^G$ denote a G-dimensional space. The objective of this work is to identify a latent space $\mathcal{V}' \subset \mathbb{R}^g$ such that $\mathcal{V}' \subset \mathcal{V}$, where $g \ll G$. This subspace may be found by using dimensionality reduction methods such as Principal Component Analysis (PCA) [28] on the training data in $\mathcal{D}_k$. An extended nonlinear version of PCA is the kernel PCA (kPCA), presented by Schölkopf et al. [29].

During the DoE, alongside the samples $\boldsymbol{x}_i$ and their corresponding objective function values f_i, constraint values $\boldsymbol{c}: \mathcal{X} \to \mathbb{R}^G$ are also collected in $\mathcal{D}$. This enables the construction of a matrix $\boldsymbol{C}(\boldsymbol{x})$ given by

$$\boldsymbol{C}(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{c}(\boldsymbol{x}_1)^\top \\ \boldsymbol{c}(\boldsymbol{x}_2)^\top \\ \vdots \\ \boldsymbol{c}(\boldsymbol{x}_N)^\top \end{bmatrix} = \begin{bmatrix} c_1(\boldsymbol{x}_1) & c_2(\boldsymbol{x}_1) & \dots & c_G(\boldsymbol{x}_1) \\ c_1(\boldsymbol{x}_2) & c_2(\boldsymbol{x}_2) & \dots & c_G(\boldsymbol{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ c_1(\boldsymbol{x}_N) & c_2(\boldsymbol{x}_N) & \dots & c_G(\boldsymbol{x}_N) \end{bmatrix} \in \mathbb{R}^{N \times G} \tag{14}$$

Here, N represents the number of samples and G denotes the number of constraints.

A. Principle Component Analysis (PCA)

Within PCA, a linear combination with maximum variance is sought, such that

$$\boldsymbol{C}\boldsymbol{v} = \lambda \boldsymbol{v} \tag{15}$$

where $\boldsymbol{v}$ is a vector of constants. These linear combinations are called the principle components of the data contained in $\boldsymbol{C}$. After centering the data with $\bar{\boldsymbol{C}} = \boldsymbol{C} - \boldsymbol{I}_N \mu$ with $\mu = (1/N) \sum_{i=1}^{N} \boldsymbol{c}_i$, a covariance matrix $\mathcal{C}$ is computed:

$$\mathcal{C} = \frac{1}{N-1} \bar{\boldsymbol{C}}^\top \bar{\boldsymbol{C}} \in \mathbb{R}^{G \times G} \tag{16}$$

Subsequently, PCA seeks the set of orthogonal vectors that capture the maximum variance in the data. This is achieved by performing an eigenvalue decomposition of $\mathcal{C}$, to obtain the corresponding eigenvalues λ and eigenvectors $\boldsymbol{v}$ such that

$$\mathcal{C}\boldsymbol{v}_i = \lambda_i \boldsymbol{v}_i, \quad \forall\ i = 1, 2, \dots, G \tag{17}$$

with $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_G \geq 0$. The eigendecomposition of $\mathcal{C}$ is then written as

$$\mathcal{C} = \boldsymbol{\Psi}\boldsymbol{\Lambda}\boldsymbol{\Psi}^{-1} \tag{18}$$

The matrix $\boldsymbol{\Psi} = [\Psi_1, \dots, \Psi_G] \in \mathbb{R}^{G \times G}$ has orthonormal columns such that $\boldsymbol{\Psi}^\top \boldsymbol{\Psi} = \boldsymbol{I}_{\boldsymbol{\Psi}}$ and $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \dots, \lambda_G) \in \mathbb{R}^{G \times G}$ is a diagonal matrix, containing the eigenvalues. By investigating the eigenvalues in $\boldsymbol{\Lambda}$, and choosing the ones with the g-highest values, the truncated decomposition is obtained, consisting of the reduced basis containing g orthogonal basis vectors in $\boldsymbol{\Psi}_g \in \mathbb{R}^{G \times g}$ with $g \ll G$. The new basis vectors can subsequently be used as a projection $\boldsymbol{\Psi}_g^\top : \mathcal{V} \subset \mathbb{R}^G \to \mathcal{V}' \subset \mathbb{R}^g$ to project the matrix $\boldsymbol{C}$ onto the reduced subspace $\tilde{\boldsymbol{C}} \in \mathbb{R}^{N \times g}$, written as

$$\tilde{\boldsymbol{C}} = \boldsymbol{C}\boldsymbol{\Psi}_g \tag{19}$$

Summarizing, the G constraints $\boldsymbol{c}(\boldsymbol{x})$ can be represented on a reduced subspace through the mapping $\boldsymbol{\Psi}_g$ while the eigenvalues λ_i give an indication about the loss of information, potentially drastically lowering the number of constraints that need to be modeled. A graphical interpretation is depicted in Fig. 1a. For a more thorough derivation of this method, the reader is referred to Jolliffe and Cadima [28].

B. kPCA

Whereas PCA can be seen as a linear dimensionality reduction technique, in Schölkopf et al. [29] the authors present an extension, called kPCA, using a nonlinear projection step to depict nonlinearities in the data. Similarly to the PCA algorithm, the starting point is the (centered) samples $\boldsymbol{c}_i(\boldsymbol{x}_i) \in \mathcal{V} \subset \mathbb{R}^G\ \forall\ i \in \{1, \dots, N\}$.

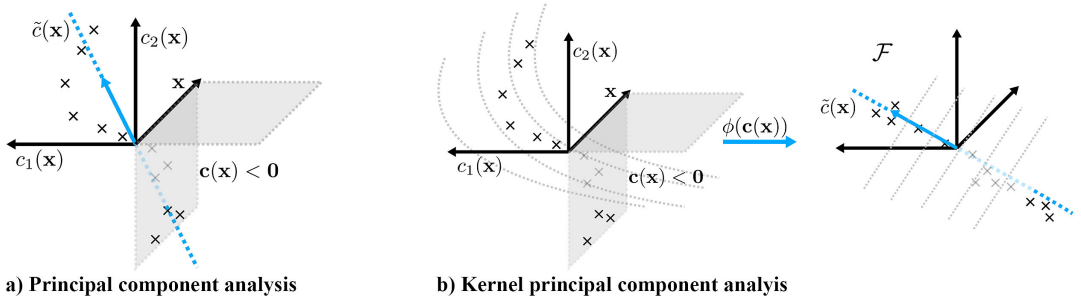
Let $\mathcal{F}$ be a dot product space (in the following, also called feature space) of arbitrary large dimensionality. A nonlinear map $\boldsymbol{\phi}(\boldsymbol{x})$ is defined as $\boldsymbol{\phi}: \mathbb{R}^G \to \mathcal{F}$. This map is used to construct a covariance matrix $\mathcal{C}$, similar to PCA, defined as

$$\mathcal{C} = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{\phi}(\boldsymbol{c}(\boldsymbol{x}_i)) \boldsymbol{\phi}(\boldsymbol{c}(\boldsymbol{x}_i))^\top \tag{20}$$

The corresponding eigenvalues and eigenvectors in $\mathcal{F}$ are computed by solving

$$\mathcal{C}\boldsymbol{v} = \lambda \boldsymbol{v} \tag{21}$$

As stated earlier, because the function $\boldsymbol{\phi}$ maps possibly to a very high-dimensional space $\mathcal{F}$, solving the eigenvalue problem therein



a) Principal component analysis b) Kernel principal component analyis

Fig. 1 Graphical interpretation of dimensionality reduction for constraints.

may be costly. A workaround is used to avoid computations in $\mathcal{F}$. Therefore, similar to the formulation of the $\mathcal{GP}$ models in Sec. II.A, a kernel $k:\mathbb{R}^G \times \mathbb{R}^G \to \mathbb{R}$ is defined as

$$k(\boldsymbol{c}(\boldsymbol{x}_i), \boldsymbol{c}(\boldsymbol{x}_j)) = \langle \boldsymbol{\phi}(\boldsymbol{c}(\boldsymbol{x}_i)), \boldsymbol{\phi}(\boldsymbol{c}(\boldsymbol{x}_j)) \rangle = \boldsymbol{\phi}(\boldsymbol{c}(\boldsymbol{x}_i))^\top \boldsymbol{\phi}(\boldsymbol{c}(\boldsymbol{x}_j)) \quad (22)$$

and the corresponding kernel matrix $\boldsymbol{K}_{ij}$ as

$$\boldsymbol{K}_{ij} \coloneqq (\boldsymbol{\phi}(\boldsymbol{c}(\boldsymbol{x}_j)), \boldsymbol{\phi}(\boldsymbol{c}(\boldsymbol{x}_j))) \in \mathbb{R}^{N \times N} \quad (23)$$

By solving the eigenvalue problem for nonzero eigenvalues

$$\boldsymbol{K}\boldsymbol{\alpha}_i = \lambda_i \boldsymbol{\alpha}_i \quad (24)$$

the eigenvalues $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_N$ and eigenvectors $\boldsymbol{\alpha}^1, \ldots, \boldsymbol{\alpha}^N$ are obtained. This part can be seen as the linear PCA, as presented before, although in the space $\mathcal{F}$. To map a test point $\boldsymbol{c}_+(\boldsymbol{x})$ from the feature space $\mathcal{F}$ to the qth principle component $\boldsymbol{v}^q$ of Eq. (21), the following relationship is evaluated:

$$((\boldsymbol{v}^q)^\top \boldsymbol{\phi}(\boldsymbol{c}_+(\boldsymbol{x}))) = \sum_{i=1}^{N} \boldsymbol{\alpha}_i^q (\boldsymbol{\phi}(\boldsymbol{c}(\boldsymbol{x}_i))^\top \boldsymbol{\phi}(\boldsymbol{c}_+(\boldsymbol{x}))) \equiv \tilde{\boldsymbol{c}}_+(\boldsymbol{x}_+) \quad (25)$$

A graphical interpretation can be found in Fig. 1b. The kernel function in Eq. (22) can also be replaced by another a priori chosen kernel function.

C. Dimensionality Reduction for Large-Scale Constraints

When large-scale constraints are involved, the computational time as well as the needed storage scales drastically because one $\mathcal{GP}$ model has to be constructed and trained for each constraint. Therefore, describing the constraints on a latent space allows to significantly lower the computational burden. This idea is based on the work of Higdon et al. [27], who project the simulation output onto a lower-dimensional subspace where the $\mathcal{GP}$ models are constructed. Other works then extended this method by employing, among others, kPCA as well as manifold learning techniques to account for nonlinearities [30,31]. However, the aforementioned authors try to approximate partial differential equations (PDE) model simulations with high-dimensional outputs, whereas, to the best of the authors' knowledge, the combination of dimensionality reduction techniques for use in high-dimensional BO with large-scale constraints for design optimization is novel.

The methods herein presented are capable of extracting the earlier introduced, most important principle components of available data, reducing the required amount of $\mathcal{GP}$ models to g instead of G, with $\boldsymbol{v}_j$ as the jth orthogonal basis vector. After projecting the data onto the lower-dimensional subspace by using either PCA as in Eq. (19) or kPCA as in Eq. (25), $\mathcal{GP}$s are constructed on the latent output space as independent batch $\mathcal{GP}$s, formulated as

$$\tilde{\boldsymbol{c}}_i \sim \mathcal{GP}(m_i(\boldsymbol{x}), k_i(\boldsymbol{x}, \boldsymbol{x}')) \;\forall\, i \in \{1, \ldots, g\} \quad (26)$$

These constraint surrogates on the latent space are then used to navigate through the design space to ultimately find a feasible and optimal design. A graphical interpretation is depicted in Fig. 1, inspired by Schölkopf et al. [29]. In the following, the projection of the constraints onto the lower-dimensional subspace in the ith iteration is denoted as $\mathcal{P}_i:\mathbb{R}^G \to \mathbb{R}^g$.

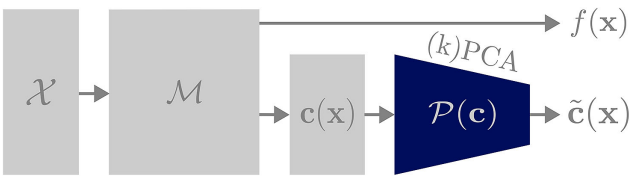


Fig. 2 Schematic illustration of (k)PCA-GP.

A schematic illustration of the $\mathcal{GP}$ construction is presented in Fig. 2, where $\mathcal{M}:\mathcal{X} \to \mathbb{R}^{G+1}$ denotes the numerical model, mapping from the design space $\mathcal{X}$ to the objective $f:\mathcal{X} \to \mathbb{R}$ and constraints $\boldsymbol{c}:\mathcal{X} \to \mathbb{R}^G$ as outputs. The constraints are then projected via (k)PCA onto a lower-dimensional representation $\tilde{\boldsymbol{c}}$ where the independent $\mathcal{GP}s$ are constructed. It is important to emphasize that the validity of a feasible design, where no constraints are violated, is checked in the original space rather than within the lower-dimensional subspace. This is made possible because in each iteration, a batch of q new samples is obtained and evaluated using the expensive-to-evaluate model. Hereinafter, the two methods are called PCA-GP SCBO and kPCA-GP SCBO.

D. Related Work and Complexity Considerations

To tackle the issue of many outputs, several works have been published. The Intrinsic Co-Regionalization Model can be related to the Linear Model of Co-Regionalization, presented in Alvarez et al. [32] and based on Multitask Gaussian Processes [33]. However, due to taking into account intertask correlation, the size of the covariance matrix increases drastically. Whereas in independent $\mathcal{GP}$ models inference and learning typically have a complexity of $\mathcal{O}((G+1)N^3)$ and $\mathcal{O}((G+1)N^2)$ for storage, the size of multitask models extends due to their Kronecker structure to complexities of $\mathcal{O}(N^3(G+1)^3)$ for inference and learning, with $G+1$ denoting the number of constraints plus the objective. Similarly, the storage complexity also scales to $\mathcal{O}(N^2(G+1)^2)$, posing significant computational challenges when the number of tasks/constraints and/or data points becomes large. The benefit of (k)PCA-GPs now is the fact that by mapping the outputs/constraints onto a g-dimensional subspace while no intertask correlations are respected, the computational costs for inference and learning only scale linearly to $\mathcal{O}((g+1)N^3 + G^3)$, where $\mathcal{O}(G^3)$ accounts for the eigendecomposition during (k)PCA and $\mathcal{O}((g+1)N^2)$ for storage, where $g \ll G$.

To address some of the issues, apart from Higdon et al. [27], Zhe et al. [34] present scalable High-Order $\mathcal{GP}$s (HOGP) and show that their method is superior to (k)PCA-GP in terms of accuracy. Because (k)PCA-GPs assume a linear structure of the outputs, meaning that the output is a linear combination of bases vectors, HOGP does not impose this kind of structure, thus claiming to be more flexible. The authors in Maddox et al. [35] then extend multitask GPs and later HOGP for a large number of outputs by employing Mathoron's rule to alleviate the computational burden of sampling from the posterior. Additionally, Bruinsma et al. [36] introduce a method that tackles the problem of needing a high number of linear basis vectors in PCA-GP, which still scale cubically in the dimensionality of the subspace when intertask correlations are taken into account. They leverage the statistics of data to achieve linear scaling. However, all these works take into account intertask correlation and thus scale poorly compared to k(PCA)-GP, as concluded by Zhe et al. [34]. Due to the fact that in engineering design problems the dimensionality and constraints can become very large, and thus high values for N and G can be expected, this work uses batched, independent $\mathcal{GP}$s in the reduced latent space, as originally proposed by Higdon et al. [27]. Due to the use in BO and the continuous retraining of the surrogates, the approach employed here significantly accelerates computations while maintaining acceptable accuracy, as presentbed in Zhe et al. [34].

IV. Numerical Experiments

In this section, the presented methodology is applied to a benchmark case before results for the aeroelastic tailoring optimization problem are shown. For comparison purposes, we adopt the reasoning of Hernández-Lobato et al. [16], where a feasible solution is always preferred over an infeasible one. Therefore, we use the maximum value from all feasible solutions as the default for all infeasible solutions. To leverage the capabilities of existing, well-performing frameworks, this study employs BoTorch [37] and GPyTorch [38] to make use of their extensive capabilities.

A. *7D* Speed Reducer Problem with *11* Black-Box Constraints

The *7D* speed reducer problem from Lemonge et al. [9] includes 11 black-box constraints. The known optimal value for this problem is $f^* = 2996.3482$. The results for all three evaluated methods (SCBO, PCA-GP SCBO, and kPCA-GP SCBO) are shown in Fig. 3. Twenty experiments are performed, where the solid line represents the mean objective value over the 20 experiments and the shaded area the standard deviation. f^* denotes the known optimal value of this problem. The eigenvalues of the matrix $\boldsymbol{C}$ with $N = 10$ samples are plotted on the right. Additionally, the decay of the eigenvalues λ of the constraint matrix $\boldsymbol{C} \subset \mathcal{D}$ is depicted. In this example where $G = 11$, $g = 4$ principal components are chosen. The SCBO hyperparameters are defined according to Eriksson and Poloczek [26]. The batch size is defined as $q = 1$ and $N = 20$ initial samples. The results are compared in Table 1. All methods find a feasible and optimal design. It is obvious that the original SCBO method converges faster than the ones employing latent $\mathcal{GP}$s. In SCBO, each constraint is modeled independently via batched $\mathcal{GP}$s. However, besides the fact that the proposed methods are significantly faster, see Table 1, it is shown that both are ultimately converging to an optimum very close to the one obtained via SCBO and the analytical solution f^*. kPCA-GP SCBO uses the Gaussian kernel, written as

$$k(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(\frac{-\|\boldsymbol{x} - \boldsymbol{x}'\|^2}{2\sigma^2}\right) \tag{27}$$

Here, PCA-GP SCBO converges slightly faster than kPCA-GP SCBO. It needs to be emphasized that this problem also does not show a fast decay of the eigenvalues, as can be seen in Fig. 3 (left).

In addition, the influence of the number of principal components, g, is studied in Fig. 4. It can be observed that g affects the convergence of the optimization. Notably, when $g = 2$, although convergence is slower, the mean value found is close to the analytic value f^*. However, when $g = 1$ the subspace does not cover enough of the feasible design space, resulting in no feasible value being found.

Lastly, the dimensionality of both the input and output spaces are examined. After demonstrating in this section that the method is generally effective on a lower- dimensional problem, the benchmark is now extended to explore cases with either high-dimensional inputs or high-dimensional outputs. To achieve this, the benchmark is modified in two ways. First, it is embedded into a 100-dimensional input space. Second, artificial constraints are introduced, increasing the number of constraints to $G = 500$, while preserving the original optimization problem's characteristics. This is ensured by adding nonviolated constraints such that each additional constraint $c_k \leq 0$. These modifications allow for an in-depth investigation of different components. The corresponding results are presented in Fig. 5, where the following is determined:

1) Case 1 represents the benchmark problem embedded in a high-dimensional input space with $D = 100$ and $G = 11$.

2) Case 2 retains the original input space dimension with $D = 7$ but extends the number of constraints to $G = 500$.

(k)PCA-GP SCBO is applied to both cases, where the number of principal components is set to $g = 6$. Additionally, SCBO is included for Case 1, where constraints are directly modeled using independent $\mathcal{GP}$s. However, Case 2 exceeds memory resources because SCBO tries to construct 501 $\mathcal{GP}$s, one for the objective and 500 for the constraints. Moreover, $N = 20$ initial samples are used, with a batch size of $q = 3$ and an evaluation budget of 200 over 20 experiments. The experiments demonstrate that our proposed method successfully handles both high-dimensional input spaces (Case 1) and large-scale constraints (Case 2), whereas the original approach fails in the latter scenario due to the need for excessive surrogate model construction, exceeding memory resources. Although both cases converged to

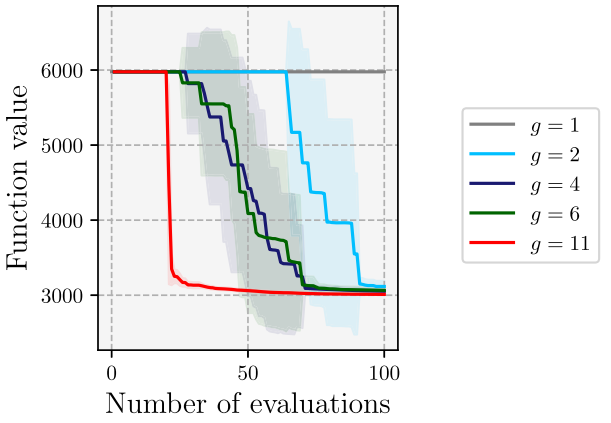


Fig. 4 The influence of the number of principal components g on the result.

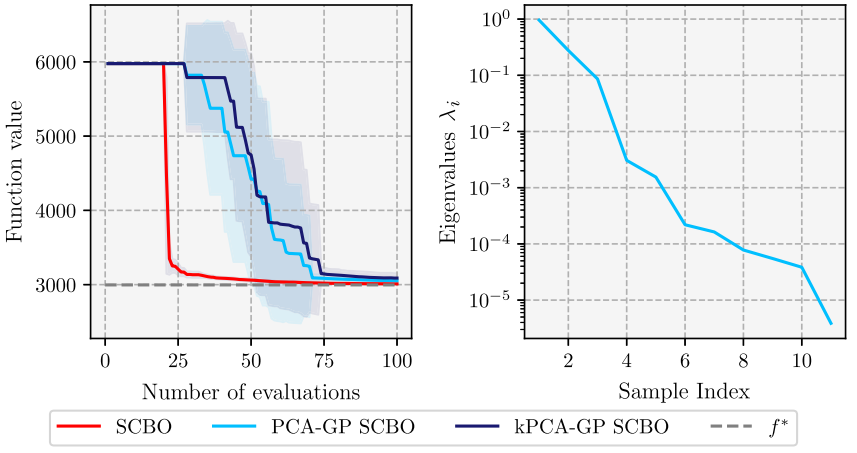


Fig. 3 7*D* speed reducer problem with 11 black-box constraints from [9].

Table 1 Computational time for speed reducer benchmark

Method	$\tilde{f}^*$, -	$(\tilde{f}^* - f^*)/f^*$, %	Time, s	Time saving, %	Successful runs, -
SCBO	3007.20	0.36	501.38	— —	20/20
PCA-GP SCBO	3053.30	1.90	201.38	59.83	20/20
kPCA-GP SCBO	3088.39	3.07	216.96	56.73	20/20

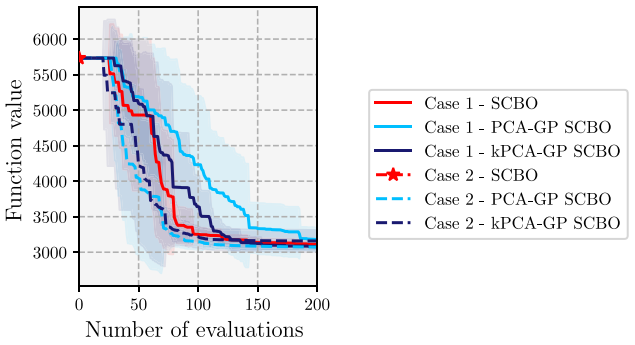


Fig. 5 The influence of the input and output dimensionality.

similar function values, it can be observed that Case 1 converges at a slower rate than Case 2. This discrepancy can be attributed to the curse of dimensionality, which affects the efficiency of surrogate modeling and exploration in high-dimensional spaces. In contrast, although a large number of constraints increases computational cost, it does not fundamentally alter the exploration process when efficient dimensionality reduction is possible, allowing for faster convergence. These findings confirm the effectiveness of our method in tackling high-dimensional and large-scale constrained optimization problems. Summarizing, the lower-dimensional subspace is constructed based on the constraint values in $\mathcal{D}$. Assuming that the global optimum lies on the boundary of the feasible space $\mathcal{X}_f$, the success of the method highly depends on how accurately the lower-dimensional subspace captures the original space. That stresses the importance of computing the projection matrix $\mathcal{P}_i$ in every iteration. However, we find that for this specific case fixing $\mathcal{P}_i = \mathcal{P}_0$, Algorithm 2 exhibits a better performance, presumably due to the rather low dimensionality and low number of constraints in combination with the use of the TR.

Algorithm 2: SCBO with Latent-Space Gaussian Processes

Input: Input space $\mathcal{X}$, Number of candidates N_c, batch size q_c, number of initial samples N_i, SCBO hyperparameters, number of eigenvalues N_{ev} or tolerance τ_{ev}

Compute DoE $\mathcal{D}_0 = \{\boldsymbol{x}_i, f(\boldsymbol{x}_i), \boldsymbol{c}(\boldsymbol{x}_i)\}_{i=1:N_i}$

$k = 0$

while Computational budget is not exhausted **do**

With $\boldsymbol{c}(\boldsymbol{x}) \subset \mathcal{D}_k$ compute projection $\mathcal{P}_k$

Project constraints onto lower-dimensional subspace $\tilde{\boldsymbol{c}}(\boldsymbol{x}) = \mathcal{P}_k(\boldsymbol{c}(\boldsymbol{x}))$

Fit $\mathcal{GP}$ for $f(\boldsymbol{x}), \tilde{c}_1(\boldsymbol{x}), \ldots, \tilde{c}_g(\boldsymbol{x})$

$\boldsymbol{x}_+ \leftarrow$ constrainedThompsonSampling (see Algorithm 1)

Evaluate $\boldsymbol{x}_+$ and observe $f(\boldsymbol{x}_+), \boldsymbol{c}(\boldsymbol{x}_+)$

Update TuRBO state

$\mathcal{D}_{k+1} = \mathcal{D}_k \cup \{\boldsymbol{x}_+, f(\boldsymbol{x}_+), \boldsymbol{c}(\boldsymbol{x}_+)\}$

$k \leftarrow k + 1$

end while

B. Aeroelastic Tailoring: An MDO Problem with *108D* and *1786* Black-Box Constraints

The MDO problem of aeroelastic tailoring addressed in this work presents a high-dimensional problem with large-scale constraints, involving both high-dimensional inputs and outputs. Unlike the aforementioned benchmark problem where it is practical to construct a $\mathcal{GP}$ for each constraint, this is computationally infeasible here, where the number of constraints is $10^3 < G < 10^5$. Therefore, the methodology presented in this study facilitates the process by modeling these constraint $\mathcal{GP}$s in a latent space. Figure 6 depicts the wing to be aeroelastically tailored by optimizing the stiffness and thickness of the wingbox. The wingbox is spanwise discretized in three sections, where top skin, bottom skin, front spar, and rear spar can take on different stiffness and thickness values. The wingspan exhibits $b = 12.28$ m, with a $c = 2.068$ m chord at the root and $c = 1.113$ m chord at the tip. The front and rear spar are located at $x_{\text{fs}} = 0.15c$ and $x_{\text{rs}} = 0.65c$, respectively. In total, $D = 108$ design variables are defined, consisting of the lamination parameters $\xi \in [-1, 1]$ and the thickness $t \in [0.002, 0.03]$ m of each panel, respectively. Each panel is described by a set of parameters $\boldsymbol{x}_i^{\text{lam}}$.

$$\begin{aligned} \boldsymbol{x} &= \{\boldsymbol{x}_1^{\text{lam}}, \boldsymbol{x}_2^{\text{lam}}, \ldots, \boldsymbol{x}_{n_p}^{\text{lam}}\} \in \mathbb{R}^{108} \text{ with} \\ \boldsymbol{x}_i^{\text{lam}} &= \{\xi_1^A, \xi_2^A, \xi_3^A, \xi_4^A, \xi_1^D, \xi_2^D, \xi_3^D, \xi_4^D, t\} \in \mathbb{R}^9 \end{aligned} \tag{28}$$

Based on the classical laminate theory, the following constitutive equations are used to relate the distributed forces N and moments M, with the in-plane ϵ^0 and curvature κ strains:

$$\begin{bmatrix} N \\ M \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}(\boldsymbol{x}) & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{D}(\boldsymbol{x}) \end{bmatrix} \begin{bmatrix} \epsilon^0 \\ \kappa \end{bmatrix} \tag{29}$$

The so-called ABD-matrix can be calculated by means of lamination parameters according to Tsai and Pagano [39] as follows:

$$\begin{aligned} \boldsymbol{A}(\boldsymbol{x}) &= t(\boldsymbol{\Gamma_0} + \boldsymbol{\Gamma_1}\xi_1^A + \boldsymbol{\Gamma_2}\xi_2^A + \boldsymbol{\Gamma_3}\xi_3^A + \boldsymbol{\Gamma_4}\xi_4^A) \\ \boldsymbol{D}(\boldsymbol{x}) &= \frac{t^3}{12}(\boldsymbol{\Gamma_0} + \boldsymbol{\Gamma_1}\xi_1^D + \boldsymbol{\Gamma_2}\xi_2^D + \boldsymbol{\Gamma_3}\xi_3^D + \boldsymbol{\Gamma_4}\xi_4^D) \end{aligned} \tag{30}$$

where $\boldsymbol{\Gamma}_i$ are material invariants, defined in Tsai and Pagano [39]. Equation (30) encodes the dependency of the design variables $\boldsymbol{x}$ with the stiffness of the system [40]. The constraints result from the incorporation of two load cases. These multiple load cases are often one of the reasons why the number of constraints can become very high. The aforementioned constraints arise from the multidisciplinary analyses, summarized in Table 2 and leading to a total number of $G = 1786$, similarly depending on the input variables $\boldsymbol{x}$. More information on the aeroelastic tailoring optimization problem can be found in Maathuis et al. [41]. Apart from the mathematical reasoning to find a latent space of the output data, the premise of the introduced methodology lies in the consistency of the physics governing the constraints across load cases, where eventually only the load changes. This stresses the potential for compressing this information due to the unchanged underlying physics for varying load cases.

The lamination parameter feasibility constraints are, however, closed-form equations. These analytical equations do not need to be modeled via surrogates because their behavior is known in the design space. Thus, these constraints are taken into account inherently within the sampling process via rejection sampling. Every candidate point in N_c is only added if it is not violating one of these feasibility constraints.

The aforementioned aeroelastic tailoring model is used to compute the DoE $\mathcal{D}$ with $N = 416$ samples. Sampling was performed via Latin Hypercube Sampling. One evaluation of this low-fidelity model takes ≈10 s due to parallelization. Anyway, subsequently PCA is applied on the matrix $\boldsymbol{C}$ to investigate its eigenvalues. Figure 7 depicts the decay of these computed eigenvalues. If it is the same error metric as in Sec. IV.A, eigenvalues up to approx $\lambda_i \approx 10^{-2}$, thus $g = 29$ principal components might be enough to construct a lower-dimensional subspace of sufficient accuracy.

As noted earlier, the high number of constraints stems from the incorporation of multiple load cases. Consequently, it becomes intriguing to explore how the eigenvalues vary when the number of load cases is altered. Recall that the eigenvalues denote the importance of their corresponding eigenvector, which serves as a measure of where to truncate the projection matrix. Beyond that, in Fig. 7 we compared the eigenvalues of $n_{\text{lc}} = 1$ and $n_{\text{lc}} = 2$ load cases. It can be observed that, even though the number of constraints in the original space has doubled, from $G = 893$ to $G = 1786$, if the eigenvalues $\lambda_i > 10^{-2}$ are used, no more principal components have to be taken into account. For $\lambda_i > 10^{-3}$, however, only 27 more components are needed to maintain the same error. Beyond that, the threshold of the eigenvalues is commonly set based on experience, thus it can be seen as a hyperparameter of the method.

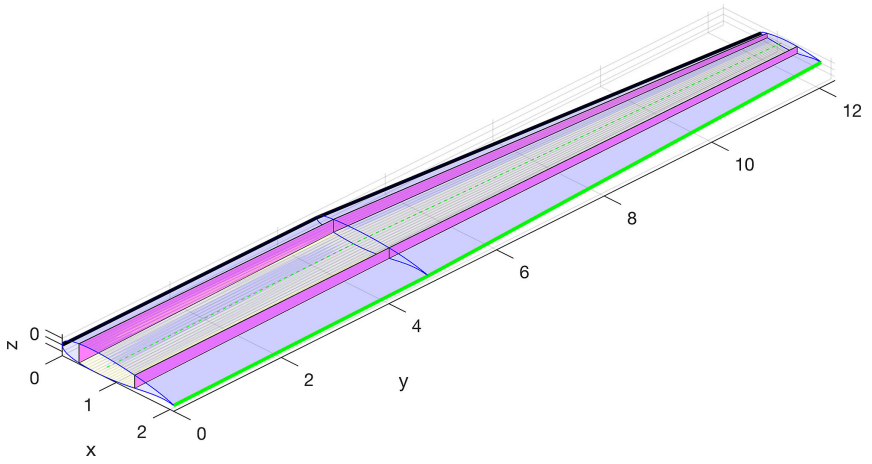


Fig. 6 Wing structure consisting of wingbox and airfoil shape.

Table 2 Aeroelastic tailoring constrained optimization problem

Type	Parameter	Symbol	#	
Objective	Minimize wing mass, kg	f	— —	
Design variables (D)	Lamination parameter	$\boldsymbol{x}$	— —	
	Laminate thickness		— —	— —
Constraints (G)	Laminate feasibility	$\boldsymbol{c}_{\text{lf}}$	72	— —
	Static strength	$\boldsymbol{c}_{\text{tw}}$	96	/load case
	Buckling	$\boldsymbol{c}_{b}$	768	/load case
	Aeroelastic stability	$\boldsymbol{c}_{\text{ds}}$	10	/load case
	Aileron effectiveness	c_{ae}	1	/load case
	Local angle of attack	$\boldsymbol{c}_{\text{AoA}}$	18	/load case

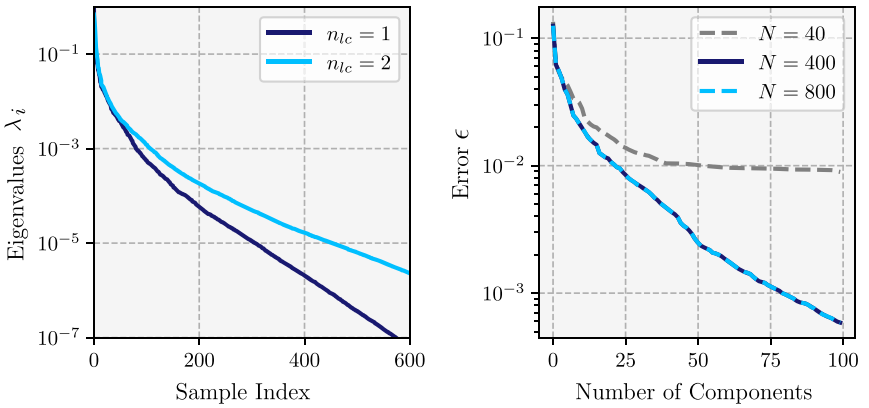


Fig. 7 Investigating the constraints in $\mathcal{D}$.

To compute the projection error, some unseen data $\boldsymbol{C}_*$ is mapped onto the lower-dimensional subspace $\tilde{\boldsymbol{C}}_* = \boldsymbol{\Psi}_g^\top \boldsymbol{C}_*$. Because PCA is a linear mapping, the inverse mapping can be simply computed by $\hat{\boldsymbol{C}}_* = \tilde{\boldsymbol{C}}_* \boldsymbol{\Psi}$. The approximation error can then be computed by

$$\epsilon = \frac{\|\boldsymbol{C}_* - \hat{\boldsymbol{C}}_*\|_F^2}{\|\boldsymbol{C}_*\|_F^2} \tag{31}$$

In Fig. 7 (right), the trend reveals that including more components leads to a reduced error, even for unseen data. Furthermore, to investigate how the construction of the lower-dimensional subspace behaves with sample size variation, the error ϵ is shown for $N = 40$, $N = 416$, and $2N$ samples. It can be seen that the error is approximately the same for the latter two cases. As anticipated, an insufficient initial sample size N results in limited information availability during the subspace construction, consequently leading to a larger error. Moreover, the conclusion drawn is that even with $N = 416$ samples, sufficient data is available to attain a reasonable subspace. Furthermore, increasing the number of samples in the DoE does not contribute to higher accuracy. To mitigate this issue, the projection matrix is recalculated in every iteration of the optimization process to incorporate as much data as possible.

Figure 8 (left) shows the results for the 108D aeroelastic tailoring problem, comparing the results of SCBO, (k)PCA-GP SCBO, Random Search, and Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES) [42]. Again, kPCA-GP SCBO uses the Gaussian kernel defined in Eq. (27). A total of five experiments are performed per method on a conventional computer with the following: Intel Xeon w3-2423, 6 cores, and 32GB RAM. The original SCBO method crashes due to insufficient memory after the first iteration, while trying to construct 1786 high-dimensional $\mathcal{GP}$ surrogates. However, a good convergence can be observed for the PCA-GP SCBO and kPCA-GP SCBO, with $g = 35$, where again PCA

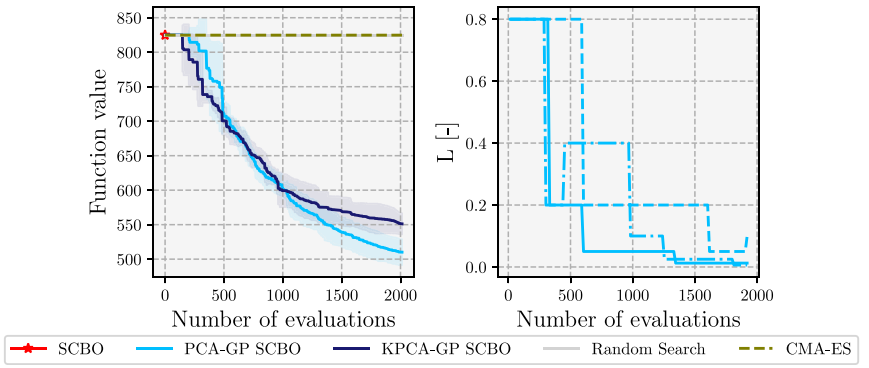


Fig. 8 Optimization results of aeroelastic tailoring case (left) and history of TR hyperrectangle size L (right).

performs better than kPCA. Additionally, it is important to note that given the size of the DoE $\mathcal{D}_0$ being $N = D$, a feasible design point can be efficiently identified, even if all points in the DoE at iteration $k = 0$ were initially infeasible. This is also highlighted by the results of the random search and CMA-ES, which both fail in finding a feasible point.

Due to the high-dimensional design space and the high number of constraints, the probability of finding a feasible point where no constraints are violated is extremely low with random search, which was unable to find a single feasible design point. Therefore, the proposed method renders the observed advantage of finding efficiently feasible points even when $\mathcal{D}_0$ only contains infeasible ones. Figure 8 (right) illustrates the size of the TR over the number of model evaluations for three randomly chosen runs. It can be observed that the size generally decreases. However, as seen for instance in the dark blue curve, the optimizer occasionally gets stuck, increases the TR size to escape the locality while the evaluation budget is not exhausted, and then restarts to decrease it. Furthermore, for this specific example, we can alternatively perform a gradient-based optimization for comparison. Using this approach, an objective value of $f^* = 402.06$ kg is obtained.

For the sake of completeness, we compare the proposed method to the so-called constraint aggregation approach, using the Kreisselmeier–Steinhauser (KS) function, written as

$$\mathrm{KS}(\boldsymbol{x}) = c_{\max} + \frac{1}{\rho}\log\left[\sum_{j=1}^{m} e^{\rho c_j(\boldsymbol{x})}\right] \tag{32}$$

This function aggregates multiple constraints, arising for example from a buckling or strength analysis into one constraint function. We implement this to lower the number of needed surrogates and compare the results against the best candidate so far. We aggregate the strain and buckling constraints for each load case individually for which we construct the $\mathcal{GP}$, whereas the other constraints are modeled independently, leading to a reduced number of constraints $g = 66$. It should be noted that, compared to PCA-GP SCBO/kPCA-GP SCBO where $g = 35$ principal components were used, in the aggregation approach 66 surrogate models need to be constructed, needing approximately twice as long for surrogate construction. Thus, downsides are the increased number of needed surrogate models in high-dimensional space as well as the additional hyperparameters needed to define which constraints to aggregate, as well as the hyperparameter ρ, which we set in this case to $\rho = 100$. For more information the reader is referred to Martins and Poon [43].

It should be pointed out that in the constraint aggregation case it not only requires more $\mathcal{GP}$s to be constructed, increasing the need for computational resources, but also the structure of the constraints needs to be known such that only constraints arising from one discipline are aggregated. This is additionally needed knowledge that might be not available, drastically lowering the generality of this approach. The corresponding results can be found in Fig. 9, where we compare SCBO with the aggregation technique with PCA-GP SCBO.

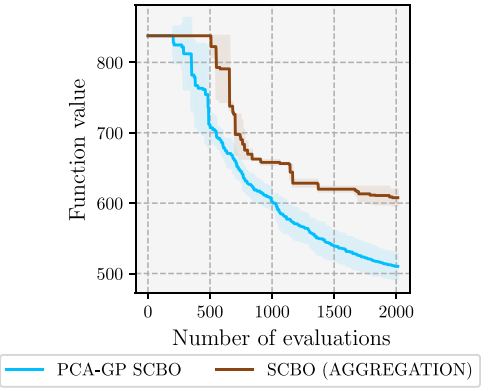


Fig. 9 Comparison of best result with constraint aggregation.

Hypothesizing why the aggregation method performed worse than the herein-introduced approaches is first of all its conservativeness, and second, the high order of the output function due to approximating all the constraints, possibly leading to a quasi-nonsmooth function, which is cumbersome to approximate. However, further research has to be performed to confirm these statements.

V. Conclusions

The aeroelastic tailoring problem exemplifies a high-dimensional multidisciplinary design optimization challenge characterized by large-scale constraints. Conducting a global design space search is inherently complex, particularly when dealing with black-box optimization problems where computing gradients is problematic. Constrained Bayesian optimization (BO) faces scalability issues due to the extensive number of constraints involved. To mitigate the scalability shortcomings of the aforementioned methods, Gaussian Processes ($\mathcal{GP}$s) are constructed on the latent space of the high-dimensional outputs in combination with a trust-region-based approach. By significantly reducing the number of required $\mathcal{GP}$s, substantial computational savings can be realized, making certain problems feasible and aligning with the objectives to reduce computational expenses. These savings are even more pronounced in high-dimensional settings, where large evaluation budgets can occur, thus increasing the computational costs for surrogate construction.

Within aeroelastic tailoring, feasible designs can be found relatively easily by increasing the thickness of each panel. However, this simplicity does not extend to other problems. The presented approach demonstrates the capability to drastically reduce computational time, thus making constrained Bayesian optimization feasible for such problems. Numerical investigations confirm the applicability of this method to aeroelastic tailoring, showcasing its effectiveness for multiple load cases with minimal additional principal components required.

An analytical example further illustrates that the proposed method converges to approximately the same objective function value. Whereas the authors' work primarily addresses aeroelastic

tailoring, the method's generality allows for application to various problems involving large-scale constraints. This flexibility is supported by numerical evidence showing the ease of application to diverse high-dimensional constraint problems.

Additionally, any dimensionality reduction method, such as autoencoders, can be seamlessly integrated into the methodology. When compared to other methods for handling large-scale constraints, such as penalty and constraint aggregation methods, the authors' proposed method demonstrates superior results without relying on specific knowledge about constraint categories. Whereas the herein presented method works with a fixed user-defined or eigenvalue-based number of principal components g, a promising path could be an extension of this method, using an adapting number g such that the approximation error of the latent space is minimized. This might further improve the method. Moreover, future research will focus on simultaneously reducing input and output spaces. The authors' current methodology necessitates training latent $\mathcal{GP}$s on the full-dimensional input space. The introduced approaches in Ch. II.D perform dimensionality reduction in the input space, offering promising avenues for further improvements. Simultaneously reducing input and output space would highly increase the scalability of this approach. Moreover, the efficient utilization of gradients, if available, will be explored to combine gradient-based and surrogate approaches. This could facilitate the use of active subspaces, potentially enhancing performance.

Besides its application in BO, this research also holds promise for design under uncertainty. $\mathcal{GP}$s offer a distinct advantage in providing a measure of variance. When addressing systems with high-dimensional outputs, this method becomes particularly advantageous. By leveraging Principal Component Analysis, the authors facilitate an efficient mapping back to the original high-dimensional space. This approach is particularly pertinent for engineering challenges where multiple model outputs are commonplace, offering a scalable solution for variability assessment. Furthermore, this method's potential application in a multifidelity optimization strategy will be explored to bolster computational efficiency and practical feasibility.

Acknowledgments

The authors would like to express their sincere gratitude to Embraer S.A., especially Pedro Higino Cabral and Alex Pereira do Prado, for their invaluable support and collaboration within the Aeroelastic Tailoring Enabled Design project. Their expertise, resources, and guidance have been instrumental in the successful completion of this study. Artificial intelligence (AI) was used in the writing process to assist with language refinement and clarity. All AI-generated content was reviewed and verified by the authors to ensure accuracy and alignment with the research objectives.

References

[1] Shirk, M. H., Hertz, T. J., and Weisshaar, T. A., "Aeroelastic Tailoring—Theory, Practice, and Promise," *Journal of Aircraft*, Vol. 23, No. 1, 1986, pp. 6–18. https://doi.org/10.2514/3.45260

[2] Werter, N. P., "Aeroelastic Modelling and Design of Aeroelastically Tailored and Morphing Wings," Ph.D. Thesis, Delft Univ. of Technology, Delft, The Netherlands, 2017. https://doi.org/10.4233/uuid:74925f40-1efc-469f-88ee-e871c720047e

[3] Dillinger, J. K., Klimmek, T., Abdalla, M. M., and Gürdal, Z., "Stiffness Optimization of Composite Wings with Aeroelastic Constraints," *Journal of Aircraft*, Vol. 50, No. 4, 2013, pp. 1159–1168. https://doi.org/10.2514/1.C032084

[4] Mockus, J., *Bayesian Approach to Global Optimization: Theory and Applications*, Springer, Netherlands, Dordrecht, 1989. https://doi.org/10.1007/978-94-009-0909-0

[5] Saves, P., Bartoli, N., Diouane, Y., Lefebvre, T., Morlier, J., David, C., Nguyen Van, E., and Defoort, S., "Multidisciplinary Design Optimization with Mixed Categorical Variables for Aircraft Design," *AIAA SCITECH 2022 Forum*, AIAA Paper 2022-0082, 2022. https://doi.org/10.2514/6.2022-0082

[6] Frazier, P. I., "A Tutorial on Bayesian Optimization," 2018. https://doi.org/10.48550/arXiv.1807.02811, arXiv: 1807.02811 [cs, math, stat]

[7] Eriksson, D., and Jankowiak, M., "High-Dimensional Bayesian Optimization with Sparse Axis-Aligned Subspaces," *37th Conference on Uncertainty in Artificial Intelligence (UAI 2021)*, PMLR 161:493–503, 2021. https://doi.org/10.48550/arXiv.2103.00349, arXiv: 2103.00349 [cs, stat].

[8] Priem, R., "Optimisation Bayésienne Sous Contraintes et en Grande Dimension Appliquée à la Conception Avion Avant Projet," Ph.D. Thesis, ISAE-SUPAERO, 2020.

[9] Lemonge, A., Barbosa, H., Borges, C., and Silve, F., "Constrained Optimization Problems in Mechanical Engineering Design Using a Real-Coded Steady-State Genetic Algorithm," *Mecánica Computacional*, Vol. 29, 2010, pp. 9287–9303.

[10] Rasmussen, C. E., and Williams, C. K. I., "Gaussian Processes for Machine Learning," *Adaptive Computation and Machine Learning*, MIT Press, Cambridge, MA, 2006.

[11] Schoenberg, I. J., "Metric Spaces and Positive Definite Functions," *Transactions of the American Mathematical Society*, Vol. 44, No. 3, 1938, pp. 522–536. https://doi.org/10.1090/S0002-9947-1938-1501980-0

[12] Močkus, J., "On Bayesian Methods for Seeking the Extremum," *Optimization Techniques IFIP Technical Conference Novosibirsk*, edited by Marchuk, G. J., Optimization Techniques 1974, Lecture Notes in Computer Science, Vol. 27, Springer, Berlin, Heidelberg, July 1975, pp. 117–129. https://doi.org/10.1007/3-540-07165-2_55

[13] Thompson, W. R., "On the Likelihood that One Unknown Probability Exceeds Another in View of the Evidence of Two Samples," *Biometrika*, Vol. 25, Nos. 3–4, 1933, pp. 285–294. https://doi.org/10.1093/biomet/25.3-4.285

[14] Gardner, J. R., Kusner, M. J., and Jake, G., "Bayesian Optimization with Inequality Constraints," *31st International Conference on Machine Learning*, Beijing, China, JMLR: W&CP Volume 32, 2014, https://proceedings.mlr.press/v32/gardner14.html.

[15] Gelbart, M. A., Snoek, J., and Adams, R. P., "Bayesian Optimization with Unknown Constraints," *UAI'14: Proceedings of the 13th Conference on Uncertainty in Artificial Intelligence*, 2014, pp. 250–259. https://doi.org/10.48550/arXiv.1403.5607

[16] Hernández-Lobato, J. M., Gelbart, M. A., Adams, R. P., Hoffman, M. W., and Ghahramani, Z., "A General Framework for Constrained Bayesian Optimization Using Information-Based Search," *Journal of Machine Learning Research*, Vol. 17, No. 1, 2016, pp. 5549–5601. https://doi.org/10.48550/arXiv.1511.09422, arXiv: 1511.09422 [stat]

[17] Hernández-Lobato, J. M., Requeima, J., Pyzer-Knapp, E. O., and Aspuru-Guzik, A., "Parallel and Distributed Thompson Sampling for Large-Scale Accelerated Exploration of Chemical Space," *34th International Conference on Machine Learning*, Sydney, Australia, PMLR 70, 2017. https://doi.org/10.48550/ARXIV.1706.01825, publisher: arXiv Version Number: 1

[18] Eriksson, D., Pearce, M., Gardner, J. R., Turner, R., and Poloczek, M., "Scalable Global Optimization via Local Bayesian Optimization," *33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*, Vancouver, Canada, 2020. https://doi.org/10.48550/arXiv.1910.01739, arXiv: 1910.01739 [cs, stat].

[19] Binois, M., and Wycoff, N., "A Survey on High-Dimensional Gaussian Process Modeling with Application to Bayesian Optimization," *ACM Transactions on Evolutionary Learning and Optimization*, Vol. 2, No. 2, 2022, pp. 1–26. https://doi.org/10.1145/3545611

[20] Wang, Z., Hutter, F., Zoghi, M., Matheson, D., and de Freitas, N., "Bayesian Optimization in a Billion Dimensions via Random Embeddings," *Journal of Artificial Intelligence Research*, Vol. 55, 2016, pp. 361–387. https://doi.org/10.1613/jair.4806

[21] Raponi, E., Wang, H., Bujny, M., Boria, S., and Doerr, C., "High Dimensional Bayesian Optimization Assisted by Principal Component Analysis," *Parallel Problem Solving from Nature–PPSN XVI: 16th International Conference*, PPSN 2020, Leiden, The Netherlands, Proceedings, Part I, Sept. 2020, pp. 169–183. https://doi.org/10.1007/978-3-030-58112-1

[22] Antonov, K., Raponi, E., Wang, H., and Doerr, C., "High Dimensional Bayesian Optimization with Kernel Principal Component Analysis," *Parallel Problem Solving from Nature–PPSN XVII: 17th International Conference*, PPSN 2022, Dortmund, Germany, Proceedings, Part I, Sept. 2022, pp. 118–131. https://doi.org/10.1007/978-3-031-14714-2

[23] Santoni, M., Raponi, E., De Leone, R., and Doerr, C., "Comparison of High-Dimensional Bayesian Optimization Algorithms on BBOB," *ACM Transactions on Evolutionary Learning*, Vol. 4, No. 3, 2023, https://doi.org/arXiv:2303.00890v2.

[24] Kandasamy, K., Schneider, J., and Poczos, B., "High Dimensional Bayesian Optimisation and Bandits via Additive Models," *32nd International Conference on Machine Learning*, Lille, France, JMLR: W&CP, Vol. 37, 2015. https://doi.org/10.48550/arXiv.1503.01673, arXiv: 1503.01673 [cs, stat]

[25] Ziomek, J., and Bou-Ammar, H., "Are Random Decompositions All We Need in High Dimensional Bayesian Optimisation?" *40th International Conference on Machine Learning*, Honolulu, Hawaii, PMLR 202, 2023. https://doi.org/10.48550/arXiv.2301.12844, arXiv: 2301.12844 [cs, stat]

[26] Eriksson, D., and Poloczek, M., "Scalable Constrained Bayesian Optimization," *24th International Conference on Artificial Intelligence and Statistics (AISTATS)*, San Diego, CA, PMLR, Vol. 130, 2021. https://doi.org/10.48550/arXiv.2002.08526, arXiv: 2002.08526 [cs, stat]

[27] Higdon, D., Gattiker, J., Williams, B., and Rightley, M., "Computer Model Calibration Using High-Dimensional Output," *Journal of the American Statistical Association*, Vol. 103, No. 482, 2008, pp. 570–583. https://doi.org/10.1198/016214507000000888

[28] Jolliffe, I. T., and Cadima, J., "Principal Component Analysis: A Review and Recent Developments," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, Vol. 374, No. 2065, 2016, p. 20,150,202. https://doi.org/10.1098/rsta.2015.0202

[29] Schölkopf, B., Smola, A., and Müller, K.-R., "Nonlinear Component Analysis as a Kernel Eigenvalue Problem," *Neural Computation*, Vol. 10, No. 5, 1998, pp. 1299–1319. https://doi.org/10.1162/089976698300017467

[30] Xing, W., Shah, A., and Nair, P., "Reduced Dimensional Gaussian Process Emulators of Parametrized Partial Differential Equations Based on Isomap," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, Vol. 471, No. 2174, 2015, p. 20,140,697. https://doi.org/10.1098/rspa.2014.0697

[31] Xing, W., Triantafyllidis, V., Shah, A., Nair, P., and Zabaras, N., "Manifold Learning for the Emulation of Spatial Fields from Computational Models," *Journal of Computational Physics*, Vol. 326, Dec. 2016, pp. 666–690. https://doi.org/10.1016/j.jcp.2016.07.040

[32] Alvarez, M. A., Rosasco, L., and Lawrence, N. D., "Kernels for Vector-Valued Functions: A Review," *Foundations and Trends® in Machine Learning*, Vol. 4, No. 3, 2012, pp. 195–266. https://doi.org/10.1561/9781601985590.arXiv: 1106.6251 [cs, math, stat].

[33] Bonilla, E. V., Chai, K., and Williams, C., "Multi-Task Gaussian Process Prediction," *Advances in Neural Information Processing Systems*, edited by J. Platt, Y. Singer, D. Koller, and S. Roweis, Vol. 20, Curran Associates, Inc, Red Hook, NY, 2007, pp. 153–160, https://proceedings.neurips.cc/paper_files/paper/2007/file/66368270ffd51418ec58bd793f2d9b1b-Paper.pdf.

[34] Zhe, S., Xing, W., and Kirby, R. M., "Scalable High-Order Gaussian Process Regression," Vol. 89, 2019, pp. 2611–2620, https://proceedings.mlr.press/v89/zhe19a.html.

[35] Maddox, W. J., Balandat, M., Wilson, A. G., and Bakshy, E., "Bayesian Optimization with High-Dimensional Outputs," *35th Conference on Neural Information Processing Systems (NeurIPS 2021)*, 2021. https://doi.org/10.48550/arXiv.2106.12997, arXiv: 2106.12997 [cs, stat]

[36] Bruinsma, W. P., Perim, E., Tebbutt, W., Hosking, J. S., Solin, A., and Turner, R. E., "Scalable Exact Inference in Multi-Output Gaussian Processes," *37th International Conference on Machine Learning*, Online, PMLR 119, 2020. https://doi.org/10.48550/ARXIV.1911.06287, version Number: 3

[37] Balandat, M., Karrer, B., Jiang, D. R., Daulton, S., Letham, B., Wilson, A. G., and Bakshy, E., "BoTorch: A Framework for Efficient Monte-Carlo Bayesian Optimization," *Advances in Neural Information Processing Systems*, Vol. 33, 34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada, 2020. https://doi.org/10.48550/arXiv.1910.06403

[38] Gardner, J. R., Pleiss, G., Bindel, D., Weinberger, K. Q., and Wilson, A. G., "GPyTorch: Blackbox Matrix-Matrix Gaussian Process Inference with GPU Acceleration," *Advances in Neural Information Processing Systems*, 2018. https://doi.org/10.48550/arXiv.1809.11165

[39] Tsai, S. W., and Pagano, N. J., "Invariant Properties of Composite Materials," Air Force Materials Lab., Air Force Systems Command TR AFML-TR-67-349, Wright-Patterson Air Force Base, OH, 1968.

[40] Daniel, I. M., and Ishai, O., *Engineering Mechanics of Composite Materials*, 2nd ed., Oxford Univ. Press, New York, 2006.

[41] Maathuis, H. F., De Breuker, R., and Castro, S. G., "High-Dimensional Bayesian Optimisation with Large-Scale Constraints—An Application to Aeroelastic Tailoring," *AIAA SCITECH 2024 Forum*, AIAA Paper 2024-2012, 2024. https://doi.org/10.2514/6.2024-2012

[42] Hansen, N., "The CMA Evolution Strategy: A Comparing Review," *Towards a New Evolutionary Computation. Studies in Fuzziness and Soft Computing*, edited by J. A. Lozano, P. Larrañaga, I. Inza, and E. Bengoetxea, Vol. 192, Springer, Berlin, Heidelberg, 2006, https://github.com/CMA-ES/pycma. https://doi.org/10.1007/3-540-32494-1_4

[43] Martins, J. R. R. A., and Poon, N. M. K., "On Structural Optimization Using Constraint Aggregation," *Proceedings of the 6th World Congress on Structural and Multidisciplinary Optimization*, 2005.

[44] Letham, B., Calandra, R., Rai, A., and Bakshy, E., "Re-Examining Linear Embeddings for High-Dimensional Bayesian Optimization," 2020. https://doi.org/10.48550/arXiv.2001.11659

S. Shin
Associate Editor