Adaptive Dimensionality Reduction for Fast Sequential Optimization With Gaussian Processes

Seyede Fatemeh Ghoreishi[1]
Department of Mechanical Engineering,
Texas A&M University,
College Station, TX 77843
e-mail: f.ghoreishi88@tamu.edu

Samuel Friedman
Department of Mechanical Engineering,
Texas A&M University,
College Station, TX 77843
e-mail: samfriedman@tamu.edu

Douglas L. Allaire
Department of Mechanical Engineering,
Texas A&M University,
College Station, TX 77843
e-mail: dallaire@tamu.edu

Available computational models for many engineering design applications are both expensive and and of a black-box nature. This renders traditional optimization techniques difficult to apply, including gradient-based optimization and expensive heuristic approaches. For such situations, Bayesian global optimization approaches, that both explore and exploit a true function while building a metamodel of it, are applied. These methods often rely on a set of alternative candidate designs over which a querying policy is designed to search. For even modestly high-dimensional problems, such an alternative set approach can be computationally intractable, due to the reliance on excessive exploration of the design space. To overcome this, we have developed a framework for the optimization of expensive black-box models, which is based on active subspace exploitation and a two-step knowledge gradient policy. We demonstrate our approach on three benchmark problems and a practical aerostructural wing design problem, where our method performs well against traditional direct application of Bayesian global optimization techniques.
[DOI: 10.1115/1.4043202]

1 Introduction

For many engineering design applications, available computational models are expensive and of a black-box nature. In such situations, Bayesian global optimization techniques, such as efficient global optimization (EGO) [1], sequential Kriging optimization (SKO) [2], value-based global optimization [3], and the knowledge gradient (KG) [4–6] can be very effective. These techniques generally rely on the simultaneous learning and optimization of a metamodel or surrogate of the true model. Often, a querying policy is implemented over a set of alternatives to explore and exploit information from the true function. In many cases, the size of this alternatives set increases exponentially with the input dimension of the true function. This renders even modestly high-dimensional problems intractable and requires recourse to gradient-based Bayesian global optimization approaches, which can be susceptible to local minima. To better understand the challenges in high-dimensional problems, consider a d-dimensional design space where a simple discretization of each dimension to n points results in n^d different alternative samples for the exploration process. This number increases to n^{d+m} in a $(d+m)$-dimensional problem. This exponential growth in the alternatives set makes the optimization process slow or intractable, limiting the applicability of alternative-based Bayesian optimization techniques to problems with less than four or five dimensions.

To enable the use of alternatives based on Bayesian global optimization for the optimization of expensive black-box functions, we develop here a novel two-step knowledge gradient policy that exploits potential active subspaces of a given true function. In our approach, with previously queried data, the active subspace method [7] is used to map the problem to one of the smaller dimension based on directions of greatest variability of the true function. Exponentially fewer alternatives are required on this smaller dimensional problem and the knowledge gradient can readily be applied. The solution to the problem on the active subspace is then mapped to the original space where a second knowledge gradient is applied to a hyperplane orthogonal to the active subspace. The result is a method that enables the application of the alternatives based on knowledge gradient policy to moderately high-dimensional problems. Furthermore, due to the fact that the initial knowledge gradient step is focused in directions defined by the largest variability of the true function, our method is significantly more efficient than direct application of the knowledge gradient to the true function. That is, exploiting the active subspace, when one exists, can lead to significant gains in terms of iterations required for the convergence.

In this paper, we first present background on metamodel-based optimization in Sec. 2, which includes the class of Bayesian global optimization algorithms we generally consider here. In Sec. 3, the problem statement and the ingredients of our approach are discussed. Section 4 then presents our proposed approach. In Sec. 5, the approach is applied to three two-dimensional test functions to demonstrate how the approach works and its limitations. Section 6 then presents a practical aerostructural wing analysis problem. Finally, conclusions are drawn in Sec. 7.

2 Background

Optimization of expensive to evaluate black-box models is often made tractable by the incorporation of lower fidelity metamodels or surrogate models. These methods have seen significant application in engineering design optimization. Most of these approaches build corrections to low-fidelity information from higher fidelity sources, such as adding global response surface corrections to low-fidelity models [8,9], using low-fidelity information for coarse-grained search while using high-fidelity function values for fine-grained decisions [10,11], creating a response surface using both high- and low-fidelity results [12,13], and running higher-fidelity models when two or more lower-fidelity models disagree [14,15]. More formal multifidelity optimization frameworks use either a local approach, such as trust region model management and other surrogate management techniques [16–21], or a global approach constructed via interpolation of the high-fidelity objective function. For example, EGO, SKO, and KG optimization use a Gaussian process model to estimate the location of high-fidelity optima and guide multifidelity sampling [1,2,4,22].

Another common means of optimizing expensive to evaluate problems is the use of either local or global sensitivity analysis. In the local sensitivity analysis, gradient information is obtained

[1]Corresponding author.

Contributed by the Design Automation Committee of ASME for publication in the Journal of Mechanical Design. Manuscript received July 26, 2018; final manuscript received March 4, 2019; published online March 28, 2019. Assoc. Editor: Mian Li.

(e.g., using finite differences for black-box functions). This information can then be used to inform the search direction and plays a significant role in several of the surrogate model management frameworks noted earlier. For certain optimization methodologies, the dimension of the design vector is a key contributor to computational expense. For example, in many Bayesian global optimization techniques, such as EGO and KG, an optimization problem should be solved over the alternative set according to an acquisition function over the surrogate model. This can be computationally expensive or intractable for large alternative sets. To handle this situation, methods for reducing the dimension of the design vector in the alternative set are desired. Global sensitivity analysis provides a well-known technique for dimension reduction that maintains the original design variables. By considering total effect sensitivity indices [23–25], design variables that do not play a significant role can be fixed to a nominal value. However, in situations where effective lower dimensional approximations do not align well with the design variable directions, alternative subspace approximation techniques can be employed. It should be noted that while global or local sensitivity information can be used for dimensionality reduction, the process of obtaining this global/local information is still affected by the curse of dimensionality since it is based on the sampling of the black-box model or of a surrogate constructed from samples of it. Hence, the problems that can be solved by sensitivity analysis techniques are still limited by the size of the original design space.

Subspace approximation approaches are widely used in optimization [26], model reduction [27], optimal control [28], and other tasks. The proper orthogonal decomposition technique, also known as principal component analysis [29], has been developed to reduce the dimension of the state space of high-order systems. In this method, data are projected to a new coordinate system defined by principal components. Singular value decomposition is a common approach for the selection of dimensions using principal component analysis [30]. These techniques are used in modeling and optimization strategies to solve high-dimensional design problems with computationally expensive black-box functions and are surveyed in Refs. [31–33]. These techniques are typically used to either reduce the dimension of the output space, for example, the objectives of a multidimensional optimization or the dimension of an input space that has been conditioned by some process, for example, on the optimized samples on a pareto-front [34].

A recent approach to subspace approximation is the Active Subspace method [7,35]. In this method, the directions in which a function has the largest variability are detected to construct an approximate model in a low-dimensional subspace of the function's input design space. This is achieved using first-order derivatives. These derivatives can be computed by several techniques, such as adjoint methods [36,37] and algorithmic differentiation [38]. Upon computation of the function's gradients at a set of input design points, the active subspace technique detects the important directions, followed by rotating the input design space in the directions in which the function has the highest variability. Then, the input design space is projected to the low-dimensional subspace where the function is then approximated. Following Ref. [7], this low-dimensional subspace is called the active subspace [35]. This approach has been applied to design optimization [39,40], inverse analysis [41], spatial sensitivity analysis [42], aerospace shape optimization [34], and uncertainty quantification for multiphysics scramjet models [43].

In the following section, we state the problem we seek to solve. Our approach to solving this problem then follows. This approach brings to bear the tools and techniques of Bayesian global optimization with the active subspace method in a novel manner that enables more efficient optimization of expensive black-box models when an active subspace is present.

3 Problem Statement

We consider the problem of constrained optimization of an expensive to evaluate and black-box objective function $f(\mathbf{x})$ with design space, $\chi \subseteq \mathbb{R}^m$. The specific problem is to find a design according to

$$\mathbf{x}^* = \arg\max_{\mathbf{x}\in\chi} f(\mathbf{x}) \quad \text{s.t.} \quad c_j(\mathbf{x}) \le 0, \quad j = 1, 2, \ldots, s \tag{1}$$

where $\mathbf{x}$ is a set of design variables in the input design space χ, and $c_j(\mathbf{x})$, where $j = 1, 2, \ldots, s$, are a set of constraints that must be satisfied. These constraints are also assumed to be expensive to evaluate black-box functions.

For problems such as that given by Eq. (1), it is common to employ sequential querying policies built off of learned surrogate models as part of a Bayesian global optimization process designed to both explore and exploit based on learned function values. For this, there are two traditional techniques for choosing what to query next [44]. These are efficient global optimization [1] and its extensions, such as sequential Kriging optimization [45,46] and value-based global optimization [3], and the knowledge gradient [4,47–50]. EGO uses a Gaussian process but assumes no noise [51,52]. SKO also uses Gaussian processes but includes a tunable weighting factor to lean toward decisions with higher uncertainty [44]. KG uses Gaussian processes and differs from EGO in that the overall best objective value, as determined by a current learned surrogate is used, rather than a best-queried value. In Ref. [53], the superiority of the KG policy was demonstrated on several benchmark functions.

The KG policy, as with other Bayesian optimization techniques, performs poorly in systems with large input design spaces. This is generally due to the scaling of the alternatives set with input dimension. In other words, Bayesian optimization techniques rely on proper exploration of the design space at the beginning of the optimization process, which gets exponentially more expensive as the dimension of the input space increases. Thus, in this paper, we develop an approach based on the active subspace method and a two-step knowledge gradient technique to adaptively reduce the dimension of the input design space and boost the speed of the optimization process. In the following subsections, the key ingredients of our approach which include the active subspace, Gaussian process regression, and the knowledge gradient policy are discussed. Following this, our specific approach is described in detail in Sec. 4.

3.1 Active Subspace Method.

The active subspace method is a technique for discovering the directions of the largest variability of a function. Once discovered, an approximation of the function can then be constructed on a lower dimensional subspace defined by these directions. The result is the potential for learning a subspace with significantly lower dimension than that of the original problem [7]. This potential, when it exists, can be exploited to create significant efficiency gains in the application of knowledge gradient policies to expensive black-box optimization problems.

Following Ref. [35], let f be a scalar objective function of m input variables $\mathbf{x}$ in the input design space χ, and $\nabla_{\mathbf{x}} f$ be the column vector of gradient of f at design point $\mathbf{x}$. That is,

$$f = f(\mathbf{x}), \quad \nabla_{\mathbf{x}} f = \nabla_{\mathbf{x}} f(\mathbf{x}), \quad \mathbf{x} \in \chi \tag{2}$$

The goal is to find an n-dimensional, $n < m$, active subspace of the input variables that contains most of the variability of the objective function, and a function $g : \mathbb{R}^n \to \mathbb{R}$ that is the approximate representation of f in the active subspace.

The first step is estimating the covariance of the gradient by computing the expectation over the probability density function of $\mathbf{x}$ on χ. We note here that the use of concepts from probability is entirely for convenience. There are no requirements of stochasticity in the underlying variables or functions. Thus, the expectation is usually taken considering uniformly distributed design variables. The $m \times m$ covariance matrix $\mathbf{C}$ is defined as

$$\mathbf{C} = \mathbb{E}[\nabla_{\mathbf{x}} f(\mathbf{x}) \nabla_{\mathbf{x}} f(\mathbf{x})^{\top}] \tag{3}$$

The exact computation of the covariance matrix in Eq. (3) is not possible in most problems, especially in the case of black-box functions. However, it can be approximated using Monte Carlo methods. This can be achieved by first drawing M samples in the design space and computing the gradient values at those points (e.g., using finite differences). Then, the covariance matrix can be estimated as

$$\mathbf{C} \approx \frac{1}{M}\sum_{i=1}^{M} \nabla_{\mathbf{x}} f(\mathbf{x}_i) \nabla_{\mathbf{x}} f(\mathbf{x}_i)^{\top} \tag{4}$$

In order to identify the most effective directions of the input design space, the eigenvectors of the covariance matrix, which is a symmetric and positive semidefinite matrix, are computed. The covariance matrix can then be written based on the eigenvalue decomposition as

$$\mathbf{C} = \mathbf{W}\boldsymbol{\Lambda}\mathbf{W}^{\top} \tag{5}$$

where $\mathbf{W}$ is a $m \times m$ column matrix of eigenvectors and $\boldsymbol{\Lambda}$ is a diagonal matrix of eigenvalues. Note that here, the eigenvalues and eigenvectors are placed in descending order. The first n eigenvectors are then selected to form a reduced-order basis. This partitions the eigenvectors and eigenvalues as

$$\mathbf{W} = \begin{bmatrix} \mathbf{U} & \mathbf{V} \end{bmatrix}, \quad \boldsymbol{\Lambda} = \begin{bmatrix} \boldsymbol{\Lambda}_1 & \\ & \boldsymbol{\Lambda}_2 \end{bmatrix} \tag{6}$$

where $\mathbf{U}$ contains the first n columns of $\mathbf{W}$ and defines the active subspace of the input design space. Now, the original full space can be transferred to the active subspace as

$$\mathbf{z} = \mathbf{U}^{\top}\mathbf{x} \tag{7}$$

and the function f can be approximated in this active subspace as

$$f(\mathbf{x}) \approx g(\mathbf{U}^{\top}\mathbf{x}) = g(\mathbf{z}) \tag{8}$$

where the domain of g is

$$\mathcal{Z} = \{z = \mathbf{U}^{\top}\mathbf{x}, \quad \mathbf{x} \in \chi\} \subset \mathbb{R}^n \tag{9}$$

3.2 Gaussian Process Regression. Gaussian processes are powerful statistical tools for probabilistic modeling purposes. A Gaussian process can be thought of as a generalized version of the Gaussian distribution applied over a continuous input space. In other words, it is an infinite-dimensional normal distribution where each sample in the input space has a corresponding normal distribution that is characterized by mean and covariance functions [54]. This class of models is widely used in engineering applications due to its flexibility, the ability to incorporate prior knowledge, and the ability to work with small sample sizes.

Gaussian process regression is a nonparametric Bayesian approach that conditions a probabilistic function to training data. Following Ref. [54], Gaussian process regression is approached by conditioning a multivariate normal distribution as

$$f \sim \mathcal{N}(\mu, \Sigma) \tag{10}$$

where f is a normally distributed function with mean μ and covariance matrix Σ. Assuming that N training data are available, represented by $\mathbf{X}_N = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$ and $\mathbf{y}_N = (y_1, \ldots, y_N)$ as input and output samples, respectively, the posterior distribution of f at any design point $\mathbf{x}$ in the input design space is given as

$$f(\mathbf{x}) \mid \mathbf{X}_N, \mathbf{y}_N \sim \mathcal{N}(\mu(\mathbf{x}), \sigma^2(\mathbf{x})) \tag{11}$$

where

$$\mu(\mathbf{x}) = K(\mathbf{X}_N, \mathbf{x})^T [K(\mathbf{X}_N, \mathbf{X}_N) + \sigma_n^2 I]^{-1} \mathbf{y}_N \tag{12}$$

$$\sigma^2(\mathbf{x}) = k(\mathbf{x}, \mathbf{x}) - K(\mathbf{X}_N, \mathbf{x})^T [K(\mathbf{X}_N, \mathbf{X}_N) + \sigma_n^2 I]^{-1} K(\mathbf{X}_N, \mathbf{x}) \tag{13}$$

where k is a real-valued kernel function over the input space, $K(\mathbf{X}_N, \mathbf{X}_N)$ is the $N \times N$ matrix whose m, n entry is $k(\mathbf{x}_m, \mathbf{x}_n)$, and $K(\mathbf{X}_N, \mathbf{x})$ is the $N \times 1$ vector whose mth entry is $k(\mathbf{x}_m, \mathbf{x})$. Note that the term $\sigma_{n,i}^2$ can be used to model observation error and can also be used to guard against numerical ill-conditioning. In this paper, the following exponential kernel function has been employed

$$k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(\frac{-\|\mathbf{x} - \mathbf{x}'\|_2^2}{2l^2}\right) \tag{14}$$

where $\|.\|_2^2$ is square of the L_2-norm, σ_f^2 determines the prior variance, and l denotes the characteristic length scale. The parameters of the Gaussian process, i.e., σ_f^2, l, and σ_n^2, can be updated at each time new information is obtained by using a maximum likelihood approach or Bayesian techniques [54].

3.3 Knowledge Gradient. Given a Gaussian process representation of the underlying function to be optimized, the next requirement is the decision on where to query next. For this, we incorporate the knowledge gradient policy. Let $\{\mathbf{x}_{1:N}, y_{1:N}\}$ be the set of design points and the corresponding objective values that have been used to construct the Gaussian process of the objective function. Let $f(\mathbf{x})$ represent this posterior distribution of the function given this available information. The best expected objective value can be computed as

$$f_N^* = \max_{\mathbf{x} \in \boldsymbol{\chi}} \mathbb{E}[f(\mathbf{x}) \mid \mathbf{x}_{1:N}, y_{1:N}] \tag{15}$$

Similarly, if one additional design point can be queried to update the posterior distribution of the model, the best expected objective value would be

$$f_{N+1}^* = \max_{\mathbf{x} \in \boldsymbol{\chi}} \mathbb{E}[f(\mathbf{x}) \mid \mathbf{x}_{1:N+1}, y_{1:N+1}] \tag{16}$$

The difference $f_{N+1}^* - f_N^*$ specifies the improvement in value of the function resulting from the additional query. The idea is to select a design point to query that maximizes this improvement. Since the Gaussian process is the probabilistic representation of the objective function, when choosing $\mathbf{x}_{N+1}$, there is stochasticity in the value of the objective function, i.e., y_{N+1}, upon querying $\mathbf{x}_{N+1}$. Thus, one needs to compute the expected value of improvement using the posterior predictive distribution of the objective function.

The knowledge gradient technique [4,44,55–57] is a method for selecting the design point that maximizes the expected increase in the objective value. Letting $S^N = \mathbb{E}[f(\mathbf{x}) \mid \mathbf{x}_{1:N}, y_{1:N}]$ be the knowledge state, the value of being at state S^N is defined as $V^N(S^N) = \max_{\mathbf{x} \in \chi} S^N$. The knowledge gradient that is a measure of expected improvement, if the design point $\mathbf{x}$ would be queried at the next time step, can be defined as

$$\nu_{\mathbf{x}}^{KG,N} = \mathbb{E}[V^{N+1}(S^{N+1}(\mathbf{x})) - V^N(S^N) \mid S^N] \tag{17}$$

where the expectation is taken over the stochasticity in the posterior distribution of the objective at design point $\mathbf{x}$, i.e., $f(\mathbf{x})$.

Now, let χ_{alt} be the alternative set, which denotes a finite set of design samples.

The knowledge gradient policy for sequentially choosing the next query is then given as

$$\mathbf{x}^{KG,N} = \arg\max_{\mathbf{x} \in \boldsymbol{\chi}_{\text{alt}}} \nu_{\mathbf{x}}^{KG,N} \tag{18}$$

Calculation of the knowledge gradient, which is based on a piecewise linear function, is discussed in detail in two algorithms presented in Ref. [53].

4 Approach

Our approach to solving Eq. (1) is based on bringing together the ideas of Bayesian global optimization, via the knowledge gradient, and subspace approximation, via the active subspace. From a

general standpoint, the approach alternates between applying the knowledge gradient on a learned active subspace and applying the knowledge gradient on a hyperplane of the original design space (and orthogonal to the active subspace). The result is an efficient method for the solution of Eq. (1) when an active subspace exists. We stress here that the learned active subspace itself is adaptive in our methodology, allowing for rapid initial progress followed by better and better approximation of the true underlying subspace. The details of our approach are provided in the following paragraphs. This section concludes with an algorithm for implementing our approach and a flowchart depicting the main steps.

The approach starts by constructing Gaussian processes for the objective function $f(\mathbf{x})$ and the constraints $c_j(\mathbf{x})$, $j = 1, \ldots, s$, in the original m-dimensional input design space based on their available data. As discussed in Sec. 3.2, these Gaussian processes are given as

$$f(\mathbf{x}) \mid \mathbf{X}_N, \mathbf{y}_N \sim \mathcal{N}\left(\mu_h(\mathbf{x}), \sigma_h^2(\mathbf{x})\right) \tag{19}$$

$$c_j(\mathbf{x}) \mid \mathbf{X}_{N_j}, \mathbf{c}_{N_j} \sim \mathcal{N}\left(\mu_{c_j}(\mathbf{x}), \sigma_{c_j}^2(\mathbf{x})\right) \tag{20}$$

where μ_h and σ_h^2 are the mean and variance of the objective function, respectively, at design point $\mathbf{x}$ in the original high-dimensional space, and μ_{c_j} and $\sigma_{c_j}^2$ are those of constraint j given the available data $\mathbf{X}_{N_j}$ and $\mathbf{c}_{N_j}$.

After constructing the Gaussian processes of the objective function and constraints, the next step is to choose the next design point to query. Here, this is achieved via a two-step knowledge gradient process. Rather than directly applying the knowledge gradient on the potentially high-dimensional original space, we first map the original space to an active subspace. In this lower dimensional space, we apply the knowledge gradient policy. Since this lower dimensional space is an active subspace, useful objective function improvement is expected at significantly less computational expense. The active subspace itself is found by evaluating the gradient of the function at the N currently available samples. This can be done through finite differences with the current Gaussian process or other means that may be available for a given function. Then, the covariance matrix is estimated according to Eq. (4), and the normalized eigenvalues are computed for this matrix. The eigenvectors associated with the normalized eigenvalues larger than a user-defined threshold specify the transformation matrix $\mathbf{U}$ to find the active subspace. Notice that this threshold should take a value between 0 and 1. Setting larger values for this threshold results in more dimensionality reduction, whereas smaller values are associated with active subspaces with a number of dimensions closer to the original space. In practice, this value could be chosen adaptively during the optimization process. This could be done in such a way that larger values are selected at the beginning of the process for fast exploration, and as more queries are made and the covariance matrix of gradients is better approximated, smaller values of the threshold are selected to avoid unnecessary reduction and error in the optimization process.

After identifying the active subspace, the N available input vectors are projected onto the subspace as

$$\mathbf{Z}_N = \mathbf{U}_N^{\top}\mathbf{X}_N \tag{21}$$

where $\mathbf{Z}_N$ are the input vectors in the low-dimensional space, $\mathbf{U}_N$ is the active subspace, and $\mathbf{X}_N$ are the available input vectors in the original space. We note again that this active subspace is adaptive. In the initial stages of the approach, the covariance matrix might be estimated inaccurately due to its sensitivity to the initial quality/accuracy of the surrogate model and its correlation parameters, which can result in inaccurate detection of the active subspace in early stages. As the Gaussian process of the objective function is updated, a new active subspace is computed. Thus, as the more points are queried, the approximation of the active subspace becomes closer to the true active subspace as a result of the Gaussian process representation of the function becoming closer to the true underlying function.

After the original space has been mapped to the active subspace, we have the lower dimensional representation of the previously queried points according to Eq. (21). From these data, we construct a Gaussian process in the active subspace as

$$g(\mathbf{z}) \mid \mathbf{Z}_N, \mathbf{y}_N \sim \mathcal{N}\left(\mu_l(\mathbf{z}), \sigma_l^2(\mathbf{z})\right) \tag{22}$$

where g is the posterior distribution of the objective function in the low-dimensional space and μ_l and σ_l^2 are the mean and variance of the objective function at the design point $\mathbf{z}$ in this space, respectively. The next step of the approach is then to use the knowledge gradient to find the next best point to query in the active subspace. For this, we generate Latin Hypercube samples in the current active subspace, which we denote as the alternative set $\mathbf{Z}_f$. Among these alternatives, we select the one that leads to the maximum knowledge gradient in the current active subspace as

$$\mathbf{z}_{N+1} = \arg\max_{\mathbf{z}\in\mathbf{Z}_f} \nu_{\mathbf{z}}^{KG,N} \tag{23}$$

Upon selection of the best point in the current active subspace, this sample $\mathbf{z}_{N+1}$ needs to be mapped back to the original input space. For this, we use the following transformation:

$$\underbrace{\begin{bmatrix} z_{1N+1} \\ \vdots \\ z_{nN+1} \end{bmatrix}}_{\mathbf{z}_{N+1}} = \underbrace{\begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} & \cdots & u_{1m} \\ & & & \vdots & & \\ u_{n1} & u_{n2} & \cdots & u_{nn} & \cdots & u_{nm} \end{bmatrix}}_{\mathbf{U}_N^T} \underbrace{\begin{bmatrix} x_{1N+1} \\ \vdots \\ x_{nN+1} \\ \vdots \\ x_{mN+1} \end{bmatrix}}_{\mathbf{x}_{N+1}}$$

$$\text{s.t.} \quad \mathbf{x}_{N+1} \in \chi \tag{24}$$

where $\mathbf{x}_{N+1}$ is the solution to the above linear equations. It is easy to verify that the solution to the inverse mapping in Eq. (24) is not unique and the transformation will thus lead to an infinite set of design points in the original space. To keep the computation tractable, we propose a strategy to choose a single point $\mathbf{x}_{N+1} \in \chi$ given each $\mathbf{z}_{N+1} \in \mathcal{Z}$.

Since the design point needs to be inverse mapped from n dimensions to m dimensions ($n < m$), one needs to discretize $m-n$ arbitrary dimensions of the original space and plug in these discretized values in the equations and solve the n equations to find the remaining n values of $\mathbf{x}_{N+1}$. For simplicity and without loss of generality, N_f samples from the last $m-n$ dimensions are generated using techniques such as Latin Hypercube sampling. Then, the other dimensions of these N samples are obtained by solving Eq. (24). We note here that constraints are also checked at this point. Specifically, only those samples that satisfy $\mu_{c_j}(\mathbf{x}) - 3\sigma_{c_j}(\mathbf{x}) \leq 0$ for $j = 1, \ldots, s$ are kept. This set is denoted by $\mathbf{X}_f$, and called the inverse-mapped set. The constraint handling strategy follows that of Ref. [56], where it was found to have good performance. Since the Gaussian process variances are large in early iterations, it is easy to satisfy the probabilistic constraints, which enables good exploration potential. As the process moves forward, the variances are reduced via learning, and the probabilistic constraints converge to the true constraints given by Eq. (1). In general, a practitioner can control the level of acceptability in terms of constraint violation by modifying the coefficient of $\sigma_{c_j}(\mathbf{x})$ as deemed appropriate.

A depiction of the proposed inverse mapping process is shown in Fig. 1.

In this figure, the left plot shows the Gaussian process of an objective function in its active subspace and the right plot shows the Gaussian process in the original space. In the left plot, the circle in the horizontal axis shows the selected point, i.e. $\mathbf{z}_{N+1}$, by

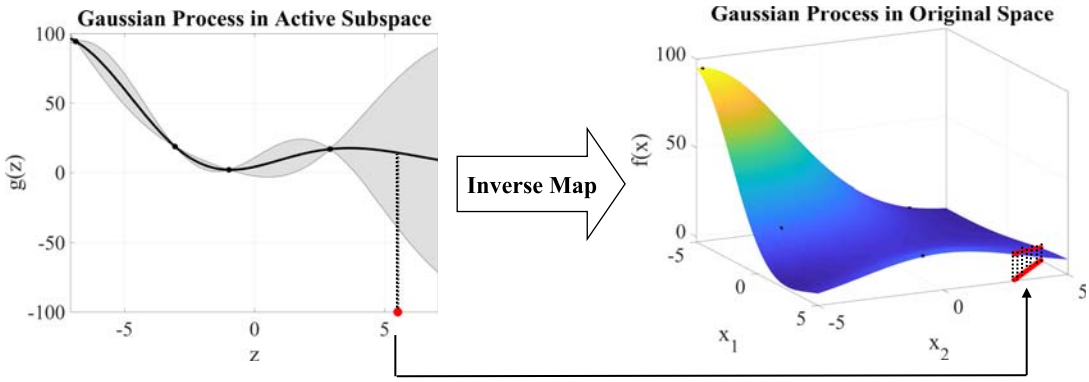

Fig. 1 A depiction of the inverse mapping process

the first-step knowledge gradient, and the samples in the right plot show the inverse-mapped set. Now, among all the samples in the inverse-mapped set, the best design point to query needs to be selected (i.e., $\mathbf{x}_{N+1} \in \mathbf{X}_f$). It is desired to select a sample that yields the highest expected improvement in the original space. We propose to select this sample by applying the knowledge gradient policy to the inverse-mapped set in the original space.

Applying the knowledge gradient in the whole original space leads to poor performance in the selection of the best design point to query. This is due to the fact that the alternative set should be enumerated in the high-dimensional original space. This number of alternatives in this set will suffer from the curse of dimensionality. Further, if there is an active subspace, then exploring certain directions of the high-dimensional original space may be inefficient due to a lack of variability along those directions. We overcome this issue in our approach by applying two knowledge gradient steps: the first on the active subspace and the second orthogonal to the active subspace in the original space. Thus, the alternative set in the original, high-dimensional space can be enumerated on a hyperplane of the original space, as shown by samples in the right plot of Fig. 1. Now, performing the knowledge gradient leads to the selection of a sample with the highest expected improvement as

$$\mathbf{x}_{N+1} = \arg\max_{\mathbf{x} \in \mathbf{X}_f} \nu_{\mathbf{x}}^{KG,N} \tag{25}$$

where $\nu_{\mathbf{x}}^{KG,N}$ is defined in Eq. (17).

After observing y_{N+1} corresponding to the selected design point $\mathbf{x}_{N+1}$, the Gaussian processes of the high-dimensional objective function and all the constraints are updated based on $\mathbf{X}_{N+1}$, $\mathbf{y}_{N+1}$, and $\mathbf{c}_{N+1_j}$. Afterward, the active subspace method is applied to find the new low-dimensional subspace according to the current updated knowledge about the objective function, and the entire process is repeated. Thus, our approach is adaptive in finding the active subspace in each iteration. The process continues by finding the next design point to query and repeats until a termination criterion, such as exhaustion of the querying budget, is met. The final solution to Eq. (1) is then found from the current Gaussian process in the original space. Our proposed approach for adaptive dimensionality reduction for fast sequential optimization with Gaussian processes is presented in Algorithm 1, and Fig. 2 presents a schematic diagram of our approach.

5 Benchmark Applications

In this section, we present the key features of our proposed sequential adaptive dimensionality reduction approach for fast optimization of expensive black-box functions. We focus here on benchmark problems designed to highlight the strengths and limitations of the approach. We begin with an analytic two-dimensional constrained optimization problem, which we use to demonstrate the effectiveness of our approach when an active subspace exists. We follow that demonstration with the application of our methodology to standard benchmark problems from the literature to reveal the behavior of our approach when an active subspace does not exist.

Algorithm 1 Adaptive dimensionality reduction for fast sequential optimization with gaussian processes

1: Construct Gaussian processes for the objective function and constraints in the original space χ.

repeat

2: Find the active subspace corresponding to the normalized eigenvalues of the covariance matrix in Eq. (4) larger than a user-defined threshold.
3: Transform the data available in the original space to the active subspace.
4: Construct the Gaussian process of the objective function in the active subspace $\mathcal{Z}$.
5: Generate Latin Hypercube samples in the active subspace $\mathcal{Z}$.
6: Apply the first-step knowledge gradient to select a design point in the active subspace.
7: Inverse map the selected design point to the original space according to Eq. (24).
8: Apply the second-step knowledge gradient to select the best design point in the original space according to Eq. (25).
9: Update the Gaussian processes of the constraints and the objective function in the original space based on the observations obtained at the selected design sample.

until termination

10: Return the feasible design point with the largest estimated objective value according to the Gaussian processes of the constraints and the objective function in the original space.

5.1 Two-Dimensional Function. We first consider the constrained maximization of an analytic two-dimensional function. The problem is given as

$$\begin{aligned} \mathbf{x}^* = \text{argmax}_{\mathbf{x} \in [-5,\,5]^2}\ & x_1^2 + x_2^2 + 2x_1x_2 \\ \text{s.t.}\quad & x_1 + x_2 < 6 \end{aligned} \tag{26}$$

The feasible optimal solution for this problem is $\mathbf{x}^* = (-5, -5)$ with the optimal objective value $f(\mathbf{x}^*) = 100$. A depiction of this example in the original two-dimensional space is shown in Fig. 3, where the vertical plane separates the feasible and infeasible regions. Figure 4 represents the transformed objective function to its active subspace, which is a one-dimensional space for this function. We note that these two figures are shown for the sake of visualization, and these are assumed to be unknown prior to applying our approach.

Figure 5 demonstrates the learned Gaussian processes for this problem both with and without use of the active subspace. Each

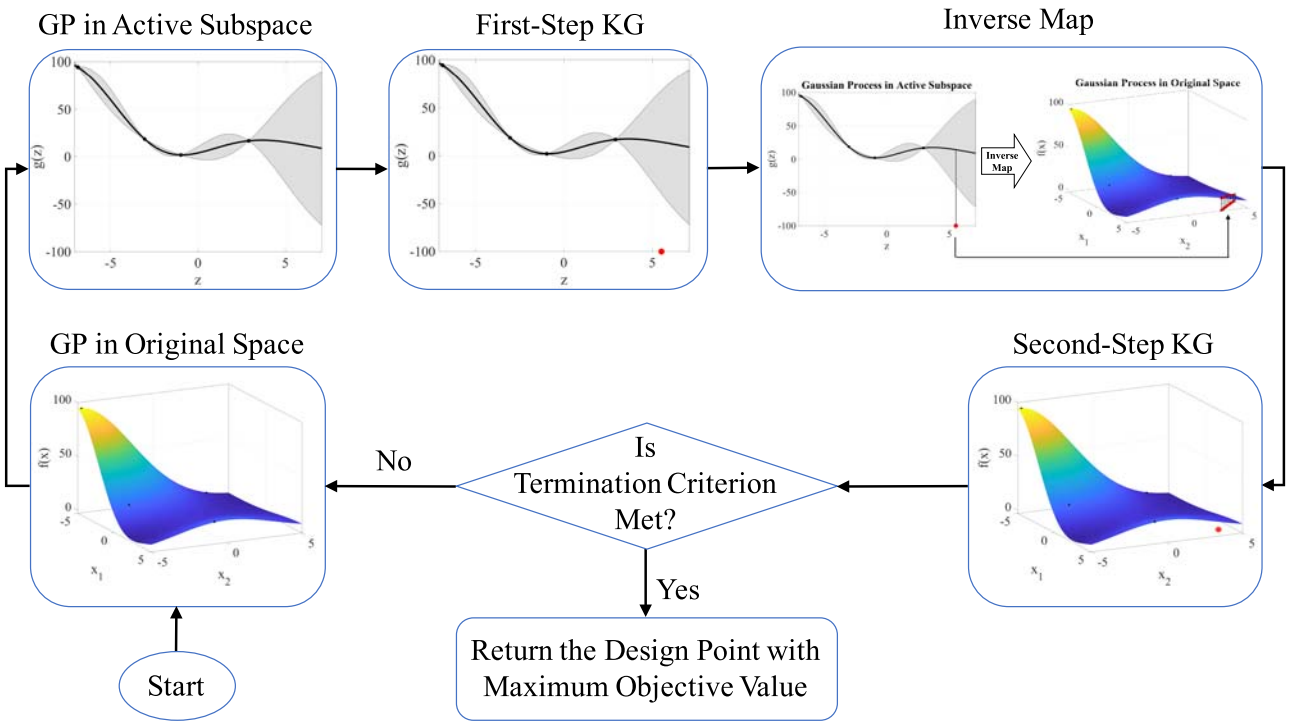

Fig. 2 A depiction of our proposed approach for fast sequential optimization. "GP" stands for Gaussian process and "KG" stands for knowledge gradient.

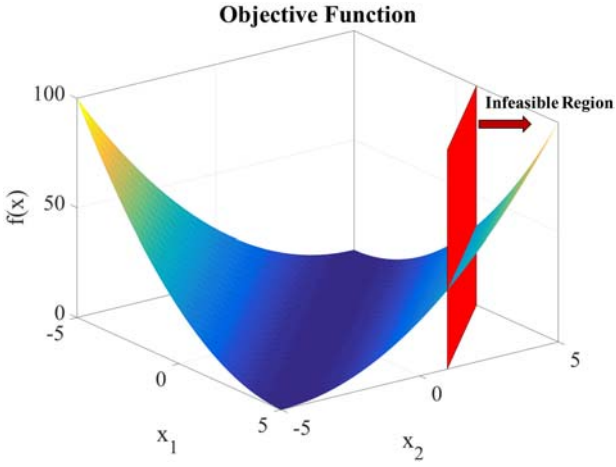

Fig. 3 A depiction of the two-dimensional function in Eq. (30)

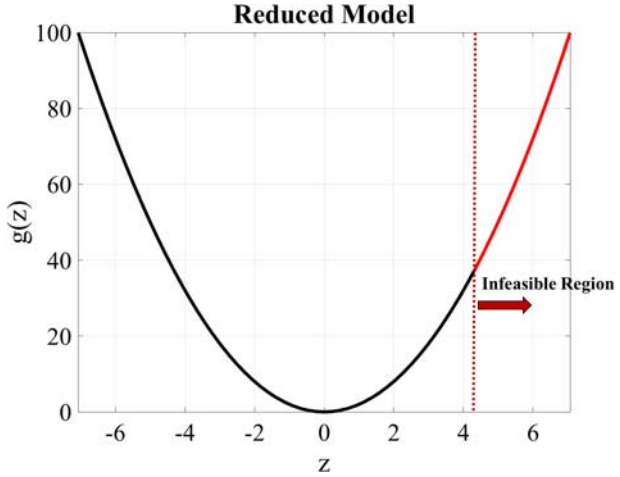

Fig. 4 A depiction of the function of Eq. (30) in its one-dimensional active subspace

column of the figure represents an iteration (here the third, sixth, and ninth iterations of the optimization process). The top row provides the Gaussian process representation in the active subspace.

The middle row shows the Gaussian process in the original space when our approach (exploiting the active subspace) is utilized. The bottom row shows the Gaussian process in the original space when the knowledge gradient is applied directly to the full model. From the figure, it is clear that the use of the active subspace significantly improves the Gaussian process representation in the original space. This is the result of more efficient use of queries to the full model, which were made possible through identification of the active subspace for this problem.

5.2 Benchmarks With and Without Active Subspaces. In this subsection, we focus on the situation where a function we seek to optimize does not have an active subspace. In most situations, given the black-box nature of the functions we seek to optimize, whether or not a given function has an active subspace will be unknown a priori. Therefore, it is necessary to understand the behavior of our approach when an active subspace does not exist for a given function. For this, we test our methodology on three unconstrained analytic problems, two of which have a negligible active subspace.

The first of the three functions studied here is the objective function given in Sec. 5.1 and will be denoted as example 1 in what follows. The second (example 2) is the negated six-hump camel-back function of Ref. [58]. This function is given as

$$f(\mathbf{x}) = -\left(\left(4 - 2.1x_1^2 + \frac{x_1^4}{3}\right)x_1^2 + x_1x_2 + \left(-4 + 4x_2^2\right)x_2^2\right) \quad (27)$$

where $x_1 \in [-3, 3]$ and $x_2 \in [-2, 2]$ and the optimal solutions are $\mathbf{x}^* = (0.0898, -0.7126)$ and $(-0.0898, 0.7126)$ with the objective

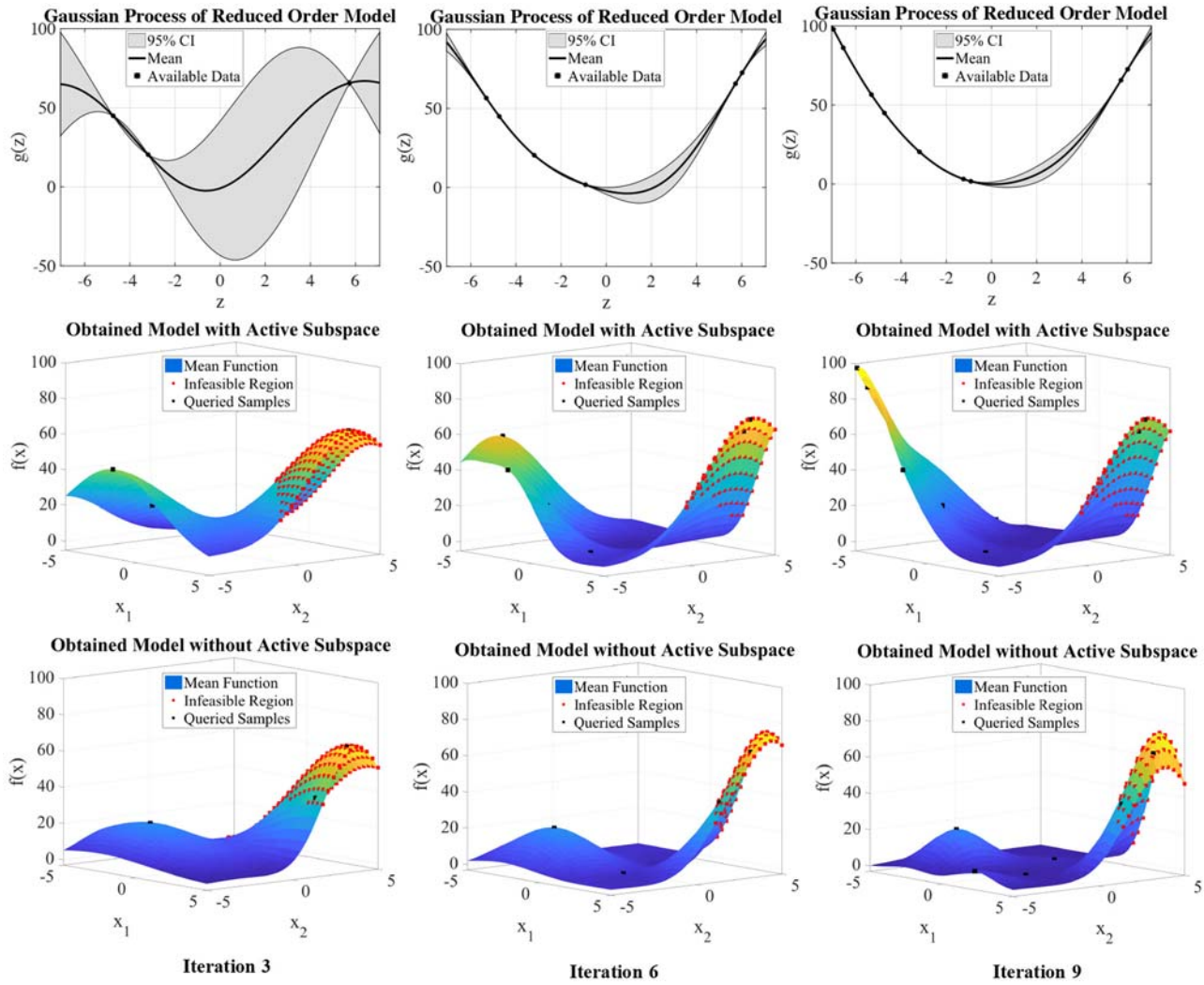

Fig. 5 Gaussian processes of the objective function in the original two-dimensional space obtained by our active subspace exploiting approach and via direct knowledge gradient application to the full model

value $f(\mathbf{x}^*) = 1.0316$. The third benchmark problem (example 3) is the negated Mishra's Bird function, which can be found in Ref. [59]. This function is given as

$$f(\mathbf{x}) = -\Big(\sin(x_2)\exp[(1-\cos x_1)^2] + \cdots \cos(x_1)\exp[(1-\sin x_2)^2] + (x_1 - x_2)^2\Big) \tag{28}$$

where $x_1 \in [-10, 0]$ and $x_2 \in [-6.5, 0]$ and the optimal solution is $\mathbf{x}^* = (-3.1302, -1.5821)$ with the objective value $f(\mathbf{x}^*) = 106.7645$.

Figure 6 shows the objective functions of examples 1, 2, and 3, as well as the mean and 95% confidence interval of the maximum objective function value obtained in each iteration of applying our approach with active subspace exploitation and performing the knowledge gradient method in the original space without the active subspace. The results are obtained over 100 independent runs. As was seen in Sec. 5.1, the first example has a one-dimensional active subspace. Our approach is able to exploit this and obtain a larger average maximum objective value in each iteration when compared with the direct application of the knowledge gradient policy to the original function. In example 2, the function gently varies in all directions, and there is no useful active subspace. As shown in the center plots of Fig. 6, our approach thus performs similarly to the direct application of the knowledge gradient. We note here, however, that in early iterations, some efficiency gains are still had through the use of our method. In the extreme case of example 3, the function varies significantly in all directions. Therefore, there is no possible reduction to a meaningful active subspace. Our approach, with the exception of the first iteration, performs exactly the same as when the knowledge gradient is applied to the function in the original space. The key takeaway here is that the worst case scenario of applying our method, that is, the situation where there is no active subspace to take advantage of, simply reverts our approach to that of direct knowledge gradient application to the original function. However, as can be seen from the left plots of Fig. 6, if there is an active subspace, our approach can exploit it for significant gains in efficiency.

5.3 Rosenbrock Function. Here, in order to assess the effectiveness of the proposed method in higher dimensions, we consider the negated 10- and 20-dimensional Rosenbrock function, which can be found in Ref. [59] and is given as

$$f(\mathbf{x}) = -\sum_{i=1}^{N-1}[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2] \tag{29}$$

where $x_i \in [-2.048, 2.048]$ and the optimal solution is $\mathbf{x}^* = (1, \ldots, 1)$ with the objective value $f(\mathbf{x}^*) = 0$. Figures 7 and 8 show the mean and 95% confidence interval of the maximum objective function value obtained in each iteration of applying our

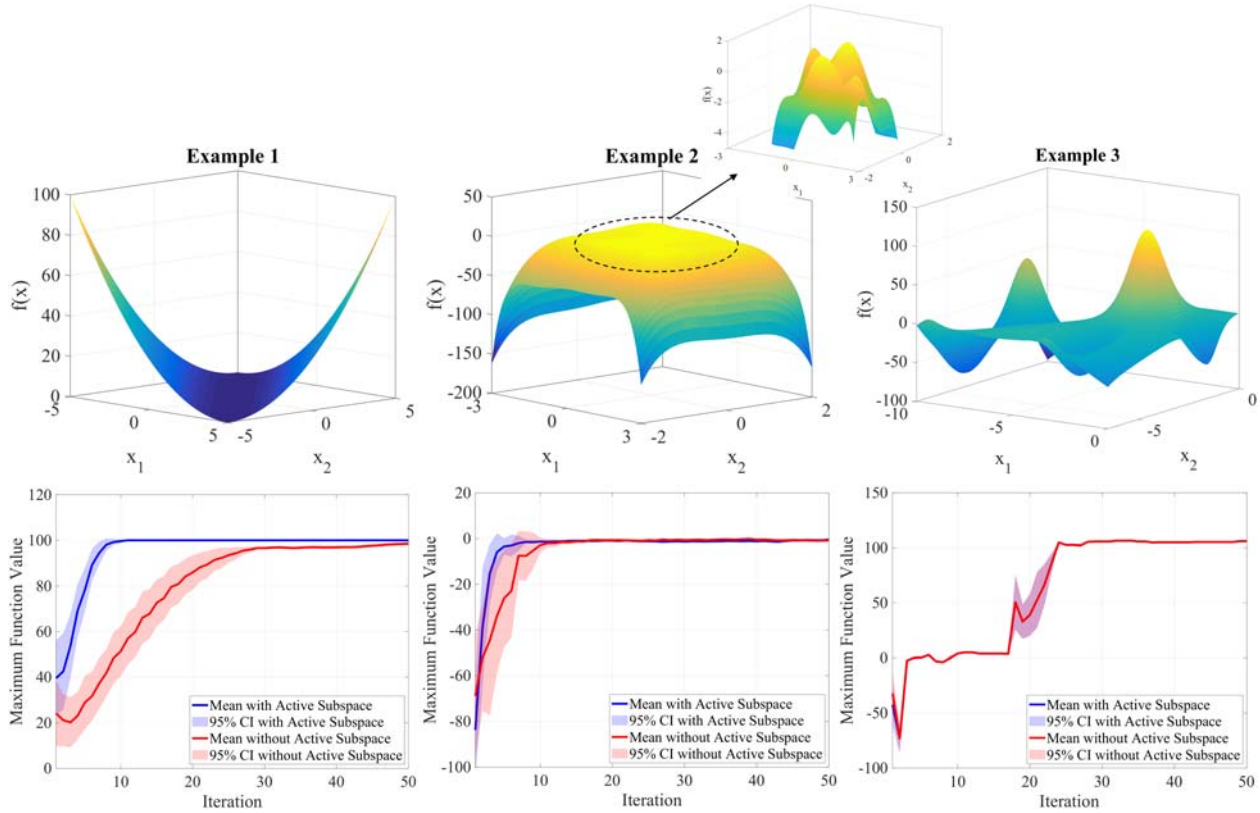

Fig. 6 The two-dimensional function examples and the mean and 95% confidence interval of the maximum function values obtained by our approach with active subspace exploitation and with direct knowledge gradient application in each iteration over 100 independent runs

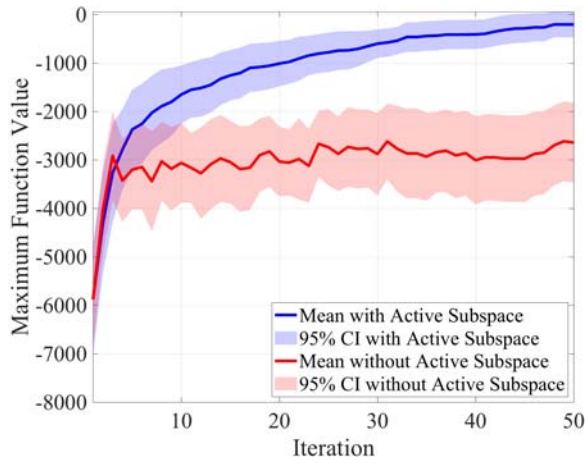

Fig. 7 The mean and 95% confidence interval of the maximum function values obtained by our approach with active subspace exploitation and without active subspace exploitation in each iteration obtained over 100 independent simulations of the 10-dimensional Rosenbrock function

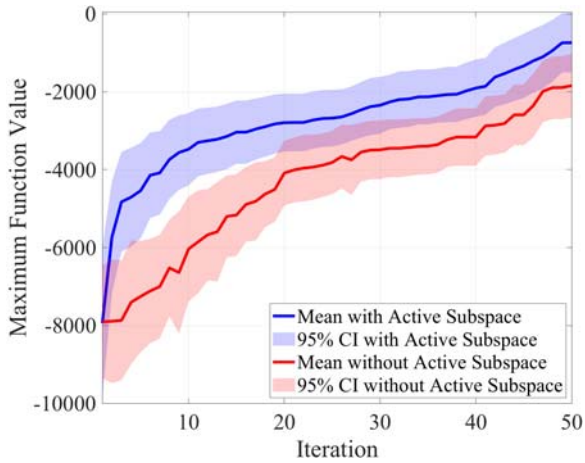

Fig. 8 The mean and 95% confidence interval of the maximum function values obtained by our approach with active subspace exploitation and without active subspace exploitation in each iteration obtained over 100 independent simulations of the 20-dimensional Rosenbrock function

approach with active subspace exploitation and performing the knowledge gradient method in the original space without the active subspace. The results are obtained over 100 independent runs. As can be seen, our approach, by exploiting the active subspace, shows significant improvements in performance versus the case without active subspace exploitation.

Figures 9 and 10 show the average dimension of the active subspace as a function of the iteration number. As can be seen, in the initial stages of the approach, the dimension is decreased substantially and as the process goes on, the dimension of the problem remains less than 5 and 9 dimensions on average for 10- and 20-dimensional Rosenbrock function, respectively.

6 Aerostructural Demonstration Problem

To demonstrate the effectiveness of our approach on a realistic problem, we consider the aerostructural design of an aircraft wing for fuel burn minimization. The aerostructural model of the wing

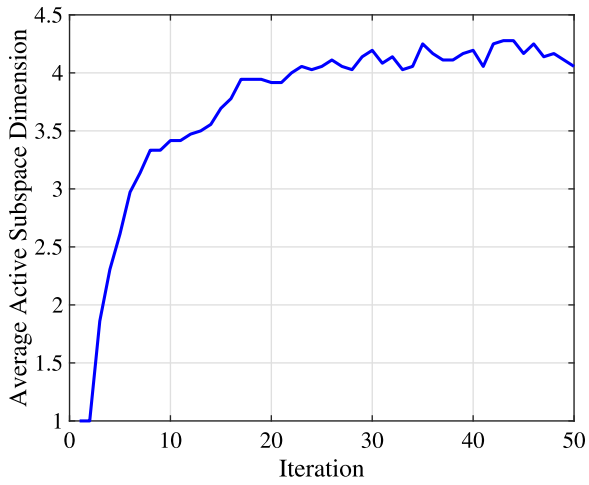

Fig. 9 Active subspace dimension in each iteration averaged over 100 independent simulations of the 10-dimensional Rosenbrock function

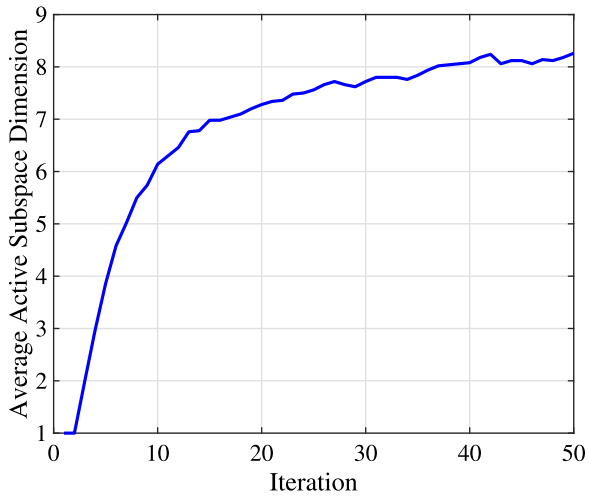

Fig. 10 Active subspace dimension in each iteration averaged over 100 independent simulations of the 20-dimensional Rosenbrock function

used for this demonstration is based on the NASA Common Research Model (CRM) [60], which is a commonly used representation of a long-range commercial airliner wing operating in transonic flight [61]. The aerostructural model consists of a three-dimensional aerodynamic and structural deformation coupled system. The aerodynamic lift is calculated using a vortex lattice method, and the structural deformation of the wing is calculated using a six-degree-of-freedom spatial beam finite element method. The coupled nature of the two subsystems is a consequence of the aerodynamic lifting force deforming the wing geometry, which in turn has a feedback effect on the aerodynamics. This is shown schematically in Fig. 11, which is adapted from Ref. [62] and uses the XDSM methodology of Ref. [63]. In the figure, y represents the fuel burn and c_i represents the constraints. A more detailed schematic of the aerostructural system is presented in Ref. [61].

The model's design variables are used to determine the undeformed wing geometry and internal spar properties. We start with the CRM wing shape and then vary its twist distribution, taper ratio, and chord ratio, and vary the spar thickness distribution. The aircraft is assumed to be in steady flight at cruising altitude, and the spar elements are assumed to be made from aluminum.

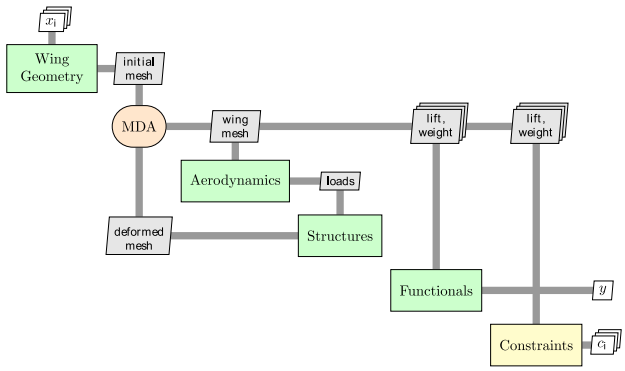

Fig. 11 A simplified depiction of the aerostructural system

The twist and thickness distribution design variables are control points used in a B-spline interpolation to vary the design along the wing's span. We used two control points each for the twist and thickness distributions. The wing's twist effectively changes the wing's angle of attack along its span. The spar thickness affects the wing's weight as well as its strength. The taper and chord ratios affect the chord length of the wing, which varies linearly along the span of the wing.

The coupled system analysis is conducted using a Gauss–Seidel iterator [64] until the aerodynamic loads and structural deformations converge to a fixed point. After the variables in the coupled system converge, the fuel range is computed along with other performance-based values and problem constraints. The model employs three constraints in the optimization: (1) it is ensured that stress on structural spars due to aerodynamic loads remain below the yield stress for the spar material properties; (2) it is ensured that the spar thickness does not exceed the wing thickness; and (3) it is assumed that the aircraft is in steady flight by enforcing the total aerodynamic lifting force be equal to the aircraft weight. In this problem, the goal is to minimize the fuel burn subject to these three constraints as

$$\begin{aligned} \mathbf{x}^* = \arg\min_{\mathbf{x}\in\chi} \ & \text{fuel burn}(\mathbf{x}) \\ \text{s.t.} \quad & \text{lift} - \text{weight constraint } (L = W) \\ & \text{structural failure constraint} \\ & \text{structural intersection constraint} \end{aligned} \tag{30}$$

where

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} \text{Twist distribution 1} \\ \text{Twist distribution 2} \\ \text{Thickness distribution 1} \\ \text{Thickness distribution 2} \\ \text{Taper ratio} \\ \text{Chord ratio} \end{bmatrix} \tag{31}$$

The boundaries of the design variables are listed in Table 1.

To identify the true optimum for this problem, we use a gradient-based nonlinear optimizer using sequential least squares programming (SLSQP) [65]. The optimal solution obtained by SLSQP is

Table 1 Design variable bounds for the aerostructural problem. LB is the lower bound, UB is the upper bound, and # is the number of variables of a given type.

Design variable	#	LB	UB
Twist distribution	2	−15	15
Thickness distribution	2	0.001	0.25
Taper ratio	1	0.2	1.5
Chord ratio	1	0.9	1.1

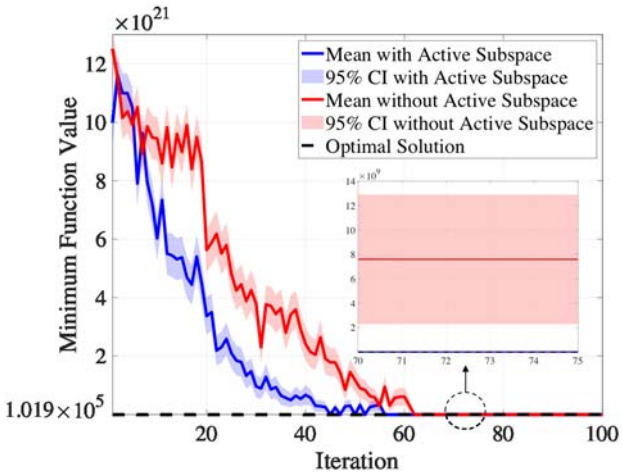

Fig. 12 The mean and 95% confidence interval of the minimum function values obtained by our approach with active subspace exploitation and without active subspace exploitation in each iteration obtained over 100 independent simulations of the aerostructural problem

$\mathbf{x}^* = [12.8037, 14.7378, 0.0377, 0.0718, 0.2, 0.9]$ with minimum fuel burn equal to 1.0190×10^5. Figure 12 shows the mean and 95% confidence interval of the minimum fuel burn obtained by our approach with active subspace exploitation and without active subspace exploitation in each iteration obtained over 100 independent simulations. Our approach, on average, obtains lower values of fuel burn in each iteration when compared with the case that does not apply the active subspace method. Figure 13 shows the mean and 95% confidence interval of the absolute difference between the optimal design variables and the design variables obtained by our approach with active subspace exploitation and without active subspace exploitation in each iteration. These results are obtained over 100 independent runs. From the figure, it is clear that the design variables obtained by our approach are closer to the optimal design variables than the case without applying the active subspace method.

Figure 14 shows the average dimension of the active subspace as a function of the iteration number. As can be seen, our approach reduces the dimension of the problem from six to less than two dimensions on average.

This reduction enables far more efficient search over alternatives when using a Bayesian global optimization search strategy, as we have done here.

It should be noted that the performance improvement by our approach with active subspace exploitation is achieved by an additional cost for surrogate model fitting for the objective function in the active subspace. In other words, the selection process by our proposed method has approximately two times more complexity than the original knowledge gradient method, due to the double selections in the active subspace and the original space. However, the most important part of the computational expense for real-world applications is often the number of black-box function evaluations, which is far less in our approach with active subspace exploitation in comparison to the case without active subspace exploitation.

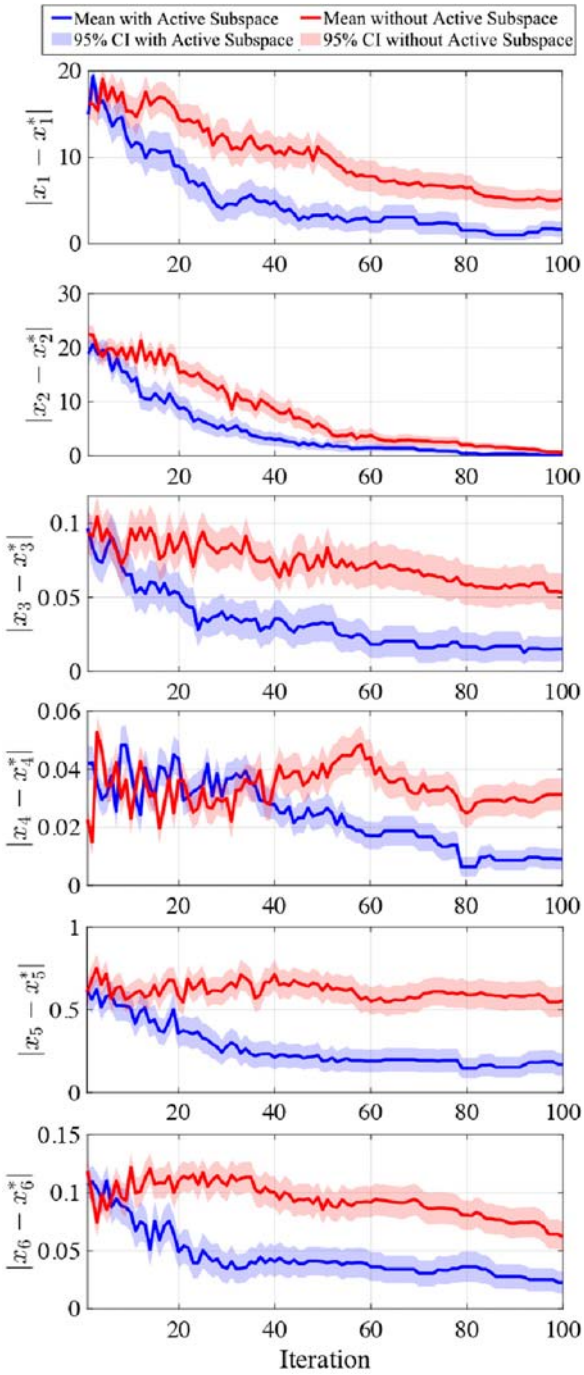

Fig. 13 The mean and 95% confidence interval of the absolute difference between the optimal design variables and the design variables obtained by our approach with active subspace exploitation and without active subspace exploitation in each iteration obtained over 100 independent simulations of the aerostructural problem

7 Conclusions

This paper has presented a sequential decision-theoretic approach for fast constrained optimization of problems with several input variables. The general use case is for expensive to evaluate black-box functions. Using the fact that certain directions in the input design space have less impact on the objective function than others, we have developed an adaptive methodology to map the high-dimensional problem to a lower dimension using the active subspace method. The Gaussian process model constructed for the objective function in the low-dimensional space is used in combination with the knowledge gradient approach to select the best sample to query in the reduced domain. Upon selection of the best design point in the low-dimensional space, an inverse mapping process to the original design space is performed. Since the inverse mapping is not one-to-one, the method selects the point which has the maximum knowledge gradient over the inverse-mapped points in the original space. We demonstrated our approach to the optimization of three two-dimensional example test problems and an aerostructural wing design problem. It has been shown that the proposed approach can

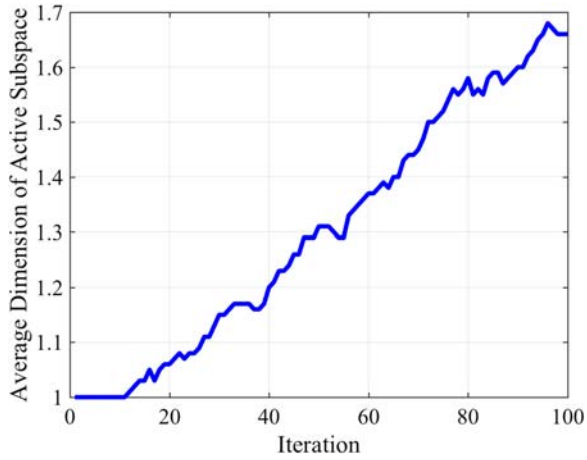

Fig. 14 Active subspace dimension in each iteration averaged over 100 independent simulations of the aerostructural problem

take advantage of existing active subspaces, leading to more efficient querying. It has also been shown that when no active subspace exists, the approach proposed here performs as well as direct application of Bayesian global optimization approaches to the full high-dimensional problem of interest. It should be noted that this framework can be generalized to any Bayesian optimization technique by only changing the acquisition function.

Acknowledgment

This work was supported by the AFOSR MURI on multi-information sources of multi-physics systems under Award Number FA9550-15-1-0038, program manager, Dr. Fariba Fahroo and by the National Science Foundation under grant no. CMMI-1663130. Opinions expressed in this paper are of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] Jones, D. R., Schonlau, M., and Welch, W. J., 1998, "Efficient Global Optimization of Expensive Black-Box Functions," J. Global Optim., **13**(4), pp. 455–492.
[2] Huang, D., Allen, T. T., Notz, W. I., and Miller, R. A., 2006, "Sequential Kriging Optimization Using Multiple-Fidelity Evaluations," Struct. Multidiscipl. Optim., **32**(5), pp. 369–382.
[3] Moore, R. A., Romero, D. A., and Paredis, C. J., 2014, "Value-Based Global Optimization," ASME J. Mech. Des., **136**(4), p. 041003.
[4] Frazier, P. I., Powell, W. B., and Dayanik, S., 2008, "A Knowledge-Gradient Policy for Sequential Information Collection," SIAM J. Control Optim., **47**(5), pp. 2410–2439.
[5] Imani, M., Ghoreishi, S. F., Allaire, D., and Braga-Neto, U., 2019, "MFBO-SSM: Multi-Fidelity Bayesian Optimization for Fast Inference in State-Space Models". AAAI.
[6] Imani, M., 2019, "Estimation, Inference and Learning in Nonlinear State-Space Models". PhD thesis, Texas A&M University, College Station, TX.
[7] Russi, T. M., 2010, *Uncertainty Quantification with Experimental Data and Complex System Models*, University of California, Berkeley.
[8] Balabanov, V., Haftka, R., Grossman, B., Mason, W., and Watson, L., 1998, "Multifidelity Response Surface Model for HSCT Wing Bending Material Weight," Seventh AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, St. Louis, MO, Sept. 2–4, AIAA 1998-4804.
[9] Simpson, T. W., Mauery, T. M., Korte, J. J., and Mistree, F., 2001, "Kriging Models for Global Approximation in Simulation-Based Multidisciplinary Design Optimization," AIAA J., **39**(12), pp. 2233–2241.
[10] Balabanov, V., and Venter, G., 2004, "Multi-Fidelity Optimization With High-Fidelity Analysis and Low-Fidelity Gradients," Tenth AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Albany, NY, August 30–Sept. 1, AIAA 2004-4459.
[11] Moore, R. A., and Paredis, C. J., 2009, "Variable Fidelity Modeling as Applied to Trajectory Optimization for a Hydraulic Backhoe," ASME 2009 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, San Diego, CA, American Society of Mechanical Engineers, New York, pp. 79–90.
[12] Keane, A. J., 2003, "Wing Optimization Using Design of Experiment, Response Surface, and Data Fusion Methods," J. Aircr., **40**(4), pp. 741–750.
[13] Chen, S., Jiang, Z., Yang, S., and Chen, W., 2016, "Multimodel Fusion Based Sequential Optimization," AIAA J., **55**(1), pp. 241–254.
[14] Choi, S., Alonso, J., and Kroo, I., 2004, "Multi-Fidelity Design Optimization of Low-Boom Supersonic Business Jets." Tenth AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Albany, NY, August 30–Sept. 1, AIAA 2004–4371.
[15] Choi, S., Alonso, J. J., and Kroo, I. M., 2009, "Two-Level Multifidelity Design Optimization Studies for Supersonic Jets," J. Aircr., **46**(3), pp. 776–790.
[16] Alexandrov, N., Dennis, J., Lewis, R., and Torczon, V., 1997, A Trust Region Framework for Managing the Use of Approximation Models in Optimization, NASA, October, Technical Report No. CR-201745.
[17] Alexandrov, N., Lewis, R., Gumbert, C., Green, L., and Newman, P., 1999, Optimization With Variable-Fidelity Models Applied to Wing Design, NASA, December, Technical Report No. CR-209826.
[18] Alexandrov, N., Lewis, R., Gumbert, C., Green, L., and Newman, P., 2001, "Approximation and Model Management in Aerodynamic Optimization with Variable-Fidelity Models," AIAA J., **38**(6), pp. 1093–1101.
[19] Talgorn, B., Le Digabel, S., and Kokkolaras, M., 2015, "Statistical Surrogate Formulations for Simulation-Based Design Optimization," ASME J. Mech. Des., **137**(2), p. 021405.
[20] Audet, C., Kokkolaras, M., Le Digabel, S., and Talgorn, B., 2018, "Order-Based Error for Managing Ensembles of Surrogates in Mesh Adaptive Direct Search," J. Global Optim., **70**(3), pp. 645–675.
[21] Talgorn, B., Audet, C., Le Digabel, S., and Kokkolaras, M., 2018, "Locally Weighted Regression Models for Surrogate-Assisted Design Optimization," Optim. Eng., **19**(1), pp. 213–238.
[22] Imani, M., Ghoreishi, S. F., and Braga-Neto, U. M., 2018, "Bayesian Control of Large MDPs With Unknown Dynamics in Data-Poor Environments," Advances in Neural Information Processing Systems, pp. 8156–8166.
[23] Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., Saisana, M., and Tarantola, S., 2008, *Global Sensitivity Analysis: The Primer*, John Wiley & Sons, New York.
[24] Sobol, I. M., 2001, "Global Sensitivity Indices for Nonlinear Mathematical Models and Their Monte Carlo Estimates," Math. Comput. Simul., **55**(1), pp. 271–280.
[25] Kucherenko, S., Rodriguez-Fernandez, M., Pantelides, C., and Shah, N., 2009, "Monte Carlo Evaluation of Derivative-Based Global Sensitivity Measures," Reliab. Eng. System Safety, **94**(7), pp. 1135–1148.
[26] Gill, P. E., Murray, W., and Wright, M. H., 1981, *Practical Optimization* Academic Press, London.
[27] Antoulas, A. C., 2005, *Approximation of Large-Scale Dynamical Systems*, SIAM.
[28] Zhou, K., Doyle, J. C., and Glover, K., 1996, *Robust and Optimal Control* Vol. 40, Prentice Hall, New Jersey.
[29] Dunteman, G. H., 1989, *Principal Components Analysis*. No. 69. Sage.
[30] Ding, C., He, X., Zha, H., and Simon, H. D., 2002, "Adaptive Dimension Reduction for Clustering High Dimensional Data," Proceedings of the 2002 IEEE International Conference on Data Mining, ICDM 2003, Maebashi City, Japan, Dec. 9–12, IEEE, New York, pp. 147–154.
[31] Shan, S., and Wang, G. G., 2010, "Survey of Modeling and Optimization Strategies to Solve High-Dimensional Design Problems with Computationally-Expensive Black-Box Functions," Struct. Multidiscipl. Optim., **41**(2), pp. 219–241.
[32] Chinesta, F., Huerta, A., Rozza, G., and Willcox, K., 2016, *Model Order Reduction: A Survey*, Wiley, New York.
[33] Benner, P., Gugercin, S., and Willcox, K., 2015, "A Survey of Projection-Based Model Reduction Methods for Parametric Dynamical Systems," SIAM Rev., **57**(4), pp. 483–531.
[34] Lukaczyk, T., Palacios, F., Alonso, J. J., and Constantine, P., 2014, "Active Subspaces for Shape Optimization," Proceedings of the 10th AIAA Multidisciplinary Design Optimization Conference, National Harbor, MD, Jan. 13–17, pp. 1–18.
[35] Constantine, P. G., Dow, E., and Wang, Q., 2014, "Active Subspace Methods in Theory and Practice: Applications to Kriging Surfaces," SIAM. J. Sci. Comput., **36**(4), pp. A1500–A1524.
[36] Bryson, A. E., 1975, *Applied Optimal Control: Optimization, Estimation and Control*, CRC Press, New York.
[37] Jameson, A., 1988, "Aerodynamic Design via Control Theory," J. Sci. Comput., **3**(3), pp. 233–260.
[38] Griewank, A., and Walther, A., 2008, *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*, SIAM.
[39] Chen, H., Wang, Q., Hu, R., and Constantine, P., 2011, "Conditional Sampling and Experiment Design for Quantifying Manufacturing Error of Transonic Airfoil," Proceedings of the 49th Aerospace Sciences Meeting.
[40] Dow, E., and Wang, Q., 2013, "Output Based Dimensionality Reduction of Geometric Variability in Compressor Blades," Proceedings of the 51st Aerospace Sciences Meeting.
[41] Constantine, P. G., Wang, Q., Doostan, A., and Iaccarino, G., 2011, "A Surrogate Accelerated Bayesian Inverse Analysis of the Hyshot II Flight Data," Proceedings of the 49th Aerospace Sciences Meeting.
[42] Constantine, P. G., Wang, Q., and Iaccarino, G., 2012, "A Method for Spatial Sensitivity Analysis". Center for Turbulence Research, Stanford, CA.
[43] Constantine, P. G., Emory, M., Larsson, J., and Iaccarino, G., 2015, "Exploiting Active Subspaces to Quantify Uncertainty in the Numerical Simulation of the Hyshot II Scramjet," J. Comput. Phys., **302**, pp. 1–20.
[44] Scott, W., Frazier, P., and Powell, W., 2011, "The Correlated Knowledge Gradient for Simulation Optimization of Continuous Parameters Using Gaussian Process Regression," SIAM. J. Optim., **21**(3), pp. 996–1026.

[45] Huang, D., Allen, T. T., Notz, W. I., and Zeng, N., 2006, "Global Optimization of Stochastic Black-Box Systems via Sequential Kriging Meta-Models," J. Global Optim., **34**(3), pp. 441–466.
[46] Ghoreishi, S. F., and Allaire, D. L., 2018, "Gaussian Process Regression for Bayesian Fusion of Multi-Fidelity Information Sources," Multidisciplinary Analysis and Optimization Conference, p. 4176.
[47] Gupta, S. S., and Miescke, K. J., 1994, "Bayesian Look Ahead one Stage Sampling Allocations for Selecting the Largest Normal Mean," Stat. Papers, **35**(1), pp. 169–177.
[48] Ghoreishi, S. F., and Allaire, D., 2018, "Multi-Information Source Constrained Bayesian Optimization," *Structural and Multidisciplinary Optimization*, Springer, New York, pp. 1–15.
[49] Gupta, S. S., and Miescke, K. J., 1996, "Bayesian Look Ahead One-Stage Sampling Allocations for Selection of the Best Population," J. Stat. Plan. Inference., **54**(2), pp. 229–244.
[50] Ghoreishi, S. F., Molkeri, A., Srivastava, A., Arroyave, R., and Allaire, D., 2018, "Multi-Information Source Fusion and Optimization to Realize ICME: Application to Dual-Phase Materials," ASME J. Mech. Des., **140**(11), p. 111409.
[51] Schonlau, M., Welch, W. J., and Jones, D., 1996, "Global Optimization with Nonparametric Function Fitting". Proceedings of the ASA, Section on Physical and Engineering Sciences, pp. 183–186.
[52] Schonlau, M., Welch, W. J., and Jones, D. R., 1998, "Global Versus Local Search in Constrained Optimization of Computer Models". Lecture Notes-Monograph Series, pp. 11–25.
[53] Frazier, P., Powell, W., and Dayanik, S., 2009, "The Knowledge-Gradient Policy for Correlated Normal Beliefs," INFORMS. J. Comput., **21**(4), pp. 599–613.
[54] Rasmussen, C. E., 2006, "Gaussian Processes for Machine Learning".
[55] Wu, J., and Frazier, P., 2016, "The Parallel Knowledge Gradient Method for Batch Bayesian Optimization". In Advances in Neural Information Processing Systems, pp. 3126–3134.
[56] Ghoreishi, S. F., and Allaire, D. L., 2018, "A Fusion-Based Multi-Information Source Optimization Approach Using Knowledge Gradient Policies". In 2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, p. 1159.
[57] Powell, W. B., and Ryzhov, I. O., 2012, *Optimal Learning Vol. 841*, John Wiley & Sons, Hoboken, NJ.
[58] Branin, F. H., Sept. 1972, "Widely Convergent Method for Finding Multiple Solutions of Simultaneous Nonlinear Equations," IBM J. Res. Dev., **16**(5), pp. 504–522.
[59] Jamil, M., and Yang, X.-S., 2013, "A Literature Survey of Benchmark Functions for Global Optimisation Problems," Int. J. Math. Model. Numer. Optim., **4**(2), pp. 150–194.
[60] Vassberg, J., Dehaan, M., Rivers, M., and Wahls, R., 2008, "Development of a Common Research Model for Applied CFD Validation Studies". In 26th AIAA Applied Aerodynamics Conference, p. 6919.
[61] Jasa, J. P., Hwang, J. T., and Martins, J. R., 2018, "Open-Source Coupled Aerostructural Optimization Using Python". Structural and Multidisciplinary Optimization, pp. 1–13.
[62] Friedman, S., Ghoreishi, S. F., and Allaire, D. L., 2017, "Quantifying the Impact of Different Model Discrepancy Formulations in Coupled Multidisciplinary Systems". In 19th AIAA Non-Deterministic Approaches Conference, AIAA SciTech Forum. American Institute of Aeronautics and Astronautics, jan.
[63] Lambe, A. B., and Martins, J. R., 2012, "Extensions to the Design Structure Matrix for the Description of Multidisciplinary Design, Analysis, and Optimization Processes," Struct. Multidiscipl. Optim., **46**(2), pp. 273–284.
[64] Yoon, S., and Jameson, A., 1988, "Lower-Upper Symmetric-Gauss-Seidel Method for the Euler and Navier-Stokes Equations," AIAA j., **26**(9), pp. 1025–1026.
[65] Kraft, D., 1988, "A Software Package for Sequential Quadratic Programming". *Forschungsbericht- Deutsche Forschungs- und Versuchsanstalt fur Luft- und Raumfahrt*.