Materials Design using an Active Subspace-based Batch Bayesian Optimization Approach

Danial Khatamsaz[*][*1], Raymundo Arróyave[†1,2,3], and Douglas Allaire[‡1]
[1]*J. Mike Walker '66 Department of Mechanical Engineering, Texas A&M University, College Station, TX, USA*
[2]*Department of Materials Science and Engineering, Texas A&M University, College Station, TX, USA*
[3]*Department of Industrial and Systems Engineering, Texas A&M University, College Station, TX, USA*

Integrated computational materials engineering (ICME) calls for integrating simulation tools and/or experiments to develop new materials and materials systems. However, implementation of ICME approaches is challenging mainly due to the considerable computational expense of such frameworks and large dimensionality of the design space. Addressing these challenges is thus critical to the success of ICME initiatives. We present here a specific Bayesian optimization framework designed to address these two challenges. In particular, we propose an active subspace batch Bayesian optimization framework. The framework makes use of dimension reduction via the active subspace method and makes use of the ability to query in parallel via the batch Bayesian optimization approach. The integration of these techniques leads to significant efficiency improvements while maintaining accuracy.

I. Introduction

In many materials design applications, existing computational models that may have a role in an ICME chain are of black-box nature and are expensive to evaluate in terms of computational runtime. Thus, the exact form of the objective function to be improved or optimized in these design applications is unknown. Available information pertaining to the objective function is often also limited to a few data points that have previously been evaluated or identified in the literature. In such situations, Bayesian optimization is considered a suitable framework for efficiently searching the design space for improved objective function values [1–4]. By employing a Bayesian optimization technique, a metamodel, or surrogate, of the objective function is built to provide a less computationally costly source to represent the true function for active learning tasks during the optimization process. However, these surrogate models are also lower in fidelity and updates to the model must to be made to ensure accuracy of the optimization process. These updates typically involve intelligently querying the ground truth function (the function we wish to optimize) and using the acquired ground truth information to update the corresponding surrogate. A querying policy, or heuristic, is implemented along with a Bayesian optimization framework to balance the exploration and exploitation of the search process based on current information about the ground truth objective function. This state of information is fully represented by the associated surrogate model of the ground truth objective function. At every stage of the design process, the querying policy is used to search the design space using surrogate models to evaluate the potential next best queries to execute from the ground truth. In this manner, the process balances needs of efficiency and needs of accuracy.

A key driver of loss of efficiency in an optimization process is the dimensionality of the design space. The more design variables, the larger the space to search. Further, the numerical aspects of many algorithms can deteriorate in higher dimensions, as is typically the case for Gaussian process surrogates in frequent use in Bayesian optimization. A common solution to this issue is dimension reduction. One such class of dimension reduction technique is subspace approximation [5]. Via a subspace approximation approach, a design problem with a high-dimensional input space is solved more efficiently by projecting the problem onto a lower-dimensional space [6–8]. One such subspace approximation approach introduced in Refs. [9–11] is the active subspace method. The intention behind using the active subspace method is to map an objective function to a lower dimensional design space based on the variability of the objective function. In particular, directions of largest variability are identified and used to construct the active subspace (which later we will search) and directions of minimal variability are ignored for the time being. This separation can be particularly effective when dealing with high-dimensional design spaces with low effective dimension. To further clarify the challenge in such design problems, assume that a querying policy uses n points per dimension of the design space, d,

[*]Ph.D. Candidate, Khatamsaz@tamu.edu, AIAA Student Member.
[†]Professor, Rarroyave@tamu.edu.
[‡]Associate Professor, Dallaire@tamu.edu, AIAA Senior Member.

to search for the next best query. This would lead to a total of n^d evaluations. Increasing the dimensionality of the design space to $d + d'$ requires $n^{d+d'}$ evaluations of the querying policy, adding exponentially to the computational cost [12]. By using the active subspace method, the problem is mapped to a lower-dimensional subspace with largest variability of the objective function. This results in either fewer points to be evaluated or a more effective use of the original planned number of queries. The active subspace method, which is similar in some respects to principal component analysis but has the key difference that it is based on the design and objective space rather than just the design (input) space, has been used in works such as design optimization [13, 14], shape optimization [15, 16], and uncertainty quantification [17, 18]. Here, we employ the active subspace method in a Bayesian optimization framework. Our approach takes two main search steps. The first step is applied over the active subspace only. Once a query candidate in the active subspace is selected, it is mapped back to the original design space. This mapping is ill-posed and leads to a second step of Bayesian optimization. In this step, the neglected space (the space orthogonal to the active subspace) is searched. This results in two smaller problems that in general are significantly more efficient than dealing with the original design space in one step. The active subspace itself is found iteratively and is updated throughout the process. That is, the active subspace can change as more is learned about the ground truth.

ICME applications often have the ability to use resources (computational typically, but also experimental) in parallel. However, typical Bayesian optimization approaches are sequential in nature and therefore cannot make use of these potential parallel capabilities. To address this, we integrate with our dimension reduction approach the batch Bayesian optimization framework. The key idea in this approach is to identify optimal batches of computational experiments that can then be run in parallel rather than typical sequential querying policies. Different strategies are recommended in Refs. [19–25] to perform optimization in a batch setting. A challenge in any Bayesian optimization framework is constructing surrogates to properly represent the true objective function. A set of hyperparameters control the behavior of these surrogates and their selection thus impacts characteristics such as smoothness and/or noise sensitivity. Since a Bayesian optimization framework is usually working with minimal data and in many cases, high-dimensional design spaces, traditional hyperparameter fitting activies (e.g., maximum likelihood, cross validation) may not yet match well the true hyperparameters of the underlying ground truth process. This increases the risk of model mismatch which can lead to longer search processes [24]. Following Ref. [24], we employ a batch Bayesian optimization approach to eliminate the hyperparameter estimation stage in constructing surrogate models. Instead, several different sets of hyperparameters are generated to construct different surrogate models, each representing the objective function differently to avoid a single representation that may not match well with the true objective function. Bayesian optimization is then carried out using all these surrogates one by one, each of which can lead to a different decision regarding the best next query. We then solve a clustering problem using the obtained set of query candidates and a batch of points is selected to query from the ground truth objective function. It has been shown in Refs. [24, 26] that batch Bayesian optimization can significantly improve the design efficiency and outperforms the pure Bayesian optimization framework as it allows more flexibility to the system in the way it models an objective function.

In this work, we develop a new Bayesian optimization framework by combining the notions of batch Bayesian optimization and the active subspace method. We seek more efficiency gains in comparison to implementing batch Bayesian optimization or the active subspace method alone by applying the active subspace method on every different representation of the true objective function obtained via constructing surrogates with different sets of hyperparameters. In other words, at every step of the framework, the active subspace of the objective function is calculated several times, each time using a different surrogate that represents the true objective function, then the two-step search process is completed for all active subspaces to obtain a set of query candidates. Finally, a clustering problem is solved over the query candidates to select a batch of designs to query the objective function and update the corresponding data set. This process is found to significantly improve on traditional sequential Bayesian optimization, as well as Bayesian optimization using the active subsapce or batch methods in isolation.

The rest of the paper proceeds as follows. First, background is provided to discuss each ingredient of the proposed framework. Gaussian process regression is introduced as a surrogate to model an objective function followed by presenting knowledge gradient as the query policy. The active subspace method and batch Bayesian optimization are then discussed in detail. We then present our approach to implement the active subspace method within a batch Bayesian optimization framework by providing an algorithm to explain every step of the process. Finally, we apply our proposed approach on benchmark problems as well as a materials design problem involving dual phase steels. We then provide concluding remarks and discuss avenues of future work.

II. Background

We consider an optimization problem as

$$\mathbf{x}^* = \arg\max_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x}) \tag{1}$$

where $f(\mathbf{x})$ is a black-box expensive to evaluate objective function defined on the design space $\mathcal{X}$.

A. Gaussian Process Regression

Gaussian process is a powerful statistical tool to model an objective function as a stochastic process by conditioning a probabilistic model to the training data. The easy manipulation and ability to incorporate prior knowledge in addition to the flexibility of a Gaussian Process, have put it among popular choices for modeling purposes in engineering applications. Using a Gaussian process to model an objective function, every point in the input space has a corresponding normal distribution in the objective space defined by mean and covariance functions [27]. Assuming there are N points available as our prior knowledge about the objective function denoted by $\{\mathbf{X}_N, \mathbf{y}_N\}$, where $\mathbf{X}_N = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$ are N input designs and $\mathbf{y}_N = (f(\mathbf{x}_1), \ldots, f(\mathbf{x}_N))$ are the corresponding objective values, the posterior distribution of the objective function at any input location $\mathbf{x}$ is

$$f_{\text{GP}}(\mathbf{x}) \mid \mathbf{X}_N, \mathbf{y}_N \sim \mathcal{N}\left(\mu(\mathbf{x}), \sigma^2_{\text{GP}}(\mathbf{x})\right) \tag{2}$$

where

$$\begin{aligned} \mu(\mathbf{x}) &= K(\mathbf{X}_N, \mathbf{x})^T \left[K(\mathbf{X}_N, \mathbf{X}_N) + \sigma_n^2 I\right]^{-1} \mathbf{y}_N \\ \sigma^2_{\text{GP}}(\mathbf{x}) &= k(\mathbf{x}, \mathbf{x}) - K(\mathbf{X}_N, \mathbf{x})^T \\ &\qquad \left[K(\mathbf{X}_N, \mathbf{X}_N) + \sigma_n^2 I\right]^{-1} K(\mathbf{X}_N, \mathbf{x}) \end{aligned} \tag{3}$$

and k is a real-valued kernel function over the input space. Then, $K(\mathbf{X}_N, \mathbf{X}_N)$ is a $N \times N$ matrix with m, n entry as $k(\mathbf{x}_m, \mathbf{x}_n)$, and $K(\mathbf{X}_N, \mathbf{x})$ is a $N \times 1$ vector with m^{th} entry as $k(\mathbf{x}_m, \mathbf{x})$. The term σ_n^2, is to incorporate model observation error based on experiments. In this work, the squared exponential function is chosen as the kernel defined by

$$k(\mathbf{x}, \mathbf{x}') = \sigma_s^2 \exp\left(-\sum_{h=1}^{d} \frac{(x_h - x'_h)^2}{2l_h^2}\right) \tag{4}$$

where d is the dimensionality of the input space, σ_s^2 is the signal variance, and l_h, where $h = 1, 2, \ldots, d$, is the characteristic length-scale to define the correlation within dimension h of the input space. The parameters σ_s^2 and l_h are the hyperparameters for a Gaussian process model which can be obtained by a maximum likelihood method [27].

B. Knowledge Gradient

An important part of any Bayesian optimization framework is the acquisition function or the heuristic to evaluate the expected gains of potential function evaluations as the next experiment to run. There have been many different acquisition functions developed for this purpose such as expected improvement (EI) [2], upper confidence bound (UCB) [28], and knowledge gradient (KG) [29–31]. In this work, knowledge gradient is used as the acquisition function.

Assume that a function is modeled by a Gaussian process and its posterior distribution at location $\mathbf{x}$, given the available data, is $f(\mathbf{x})$. Consequently, the best expected objective value is

$$f_N^* = \max_{\mathbf{x}\in\mathcal{X}} \mathbb{E}[f(\mathbf{x}) | \mathbf{x}_{1:N}, y_{1:N}] \tag{5}$$

Then, by making an additional query and updating the posterior distribution of the function, again, the best expected objective value is

$$f_{N+1}^* = \max_{\mathbf{x}\in\mathcal{X}} \mathbb{E}[f(\mathbf{x}) | \mathbf{x}_{1:N+1}, y_{1:N+1}] \tag{6}$$

and the idea is to select a query as the next experiment to maximize $f_{N+1}^* - f_N^*$ since this difference shows the improvement in the objective value. Note that since the Gaussian process is a stochastic representation of the objective function, one needs to calculate the expected value of the improvement [12].

Using knowledge gradient as the acquisition function, the value of being in a current system's knowledge state is $V^N(H^N) = \max_{\mathbf{x}\in\mathcal{X}} \mathbb{E}[f(\mathbf{x})|\mathbf{x}_{1:N}, y_{1:N}]$, and the expected improvement to the objective value if the design $\mathbf{x}$ is queried next, will be

$$\nu^{\mathrm{KG},N}(\mathbf{x}) = \mathbb{E}[V^{N+1}(H^{N+1}(\mathbf{x})) - V^N(H^N)|H^N] \tag{7}$$

and we look for a design to maximize Eq. (7) as the next query. More details on calculation of knowledge gradient are discussed in Refs. [29–31]

C. Active Subspace Method

The active subspace method (ASM) is a dimensionality reduction technique to approximate an objective function in a lower dimensional space. This lower dimensional space, known as the active subspace, is formed by linearly independent vectors in the original input space which the objective function has the largest variability along those directions. Approximating a function in a lower dimensional space reduces the computational costs regarding learning the function in machine learning objectives [9, 12, 32]. In Bayesian optimization, an acquisition function is employed to search the design space and look for the most valuable query in every stage by estimating the expected gains. Limiting the searching process to a subspace speeds up optimization tasks and reduces resource usage. Additionally, since the active subspace is formed by directions with largest variability of the function, it includes the highest amount of information about the function in comparison to any other subspaces. Eliminating other less informative directions helps to preserve computational resources and therefore, significant efficiency gains are obtained in this regard.

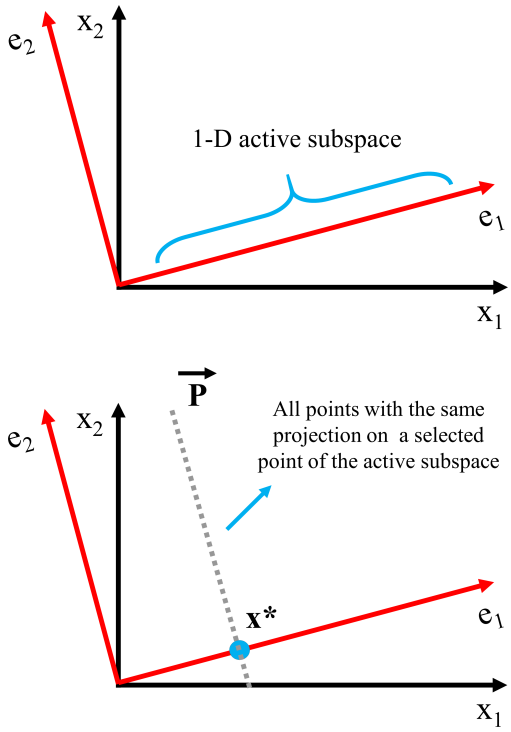

Fig 1. An example of determining the active subspace and inverse mapping problem solutions of a 2-dimensional design space.

Following the Refs. [10, 12, 33], assuming a scalar function f that takes m dimensional input $\mathbf{x}$ from the design space $\mathcal{X}$, $\nabla_x f$ represents the gradient of the objective function at location $\mathbf{x}$ in $\mathcal{X}$. Initially, we need to compute the covariance of the gradient, $\mathbf{C}$, defined as

$$\mathbf{C} = \mathbb{E}[\nabla_{\mathbf{x}} f(\mathbf{x}) \nabla_{\mathbf{x}} f(\mathbf{x})^{\mathrm{T}}] \tag{8}$$

If the objective function has a black-box nature, as it does in many engineering design problems, the gradient can be approximated by Monte Carlo approaches. Assuming there are M samples available from previous function evaluations,

the covariance matrix is formed as

$$\mathbf{C} \approx \frac{1}{M} \sum_{i=1}^{M} \nabla_{\mathbf{x}} f(\mathbf{x}_i) \nabla_{\mathbf{x}} f(\mathbf{x}_i)^{\mathrm{T}} \tag{9}$$

By computing the eigenvalues and eigen vectors of the covariance matrix, the effectiveness of directions defined by eigen vectors can be defined by the respective eigenvalues. Based on the eigenvalue decomposition, the covariance matrix can be written as

$$\mathbf{C} = \mathbf{W} \lambda \mathbf{W}^{\mathrm{T}} \tag{10}$$

where $\mathbf{W}$ is the matrix formed by eigen vectors and λ is a diagonal matrix of eigenvalues. To build an n-dimensional active subspace, the eigen vectors corresponding to the first n largest eigenvalues are selected.

$$\mathbf{W} = [\mathbf{U} \;\; \mathbf{V}], \quad \lambda = \begin{bmatrix} \lambda_1 & \\ & \lambda_2 \end{bmatrix} \tag{11}$$

The matrix $\mathbf{U}$ has n eigen vectors corresponding to the first n largest eigenvectors forming λ_1 and is called the transformation matrix. Any design point in $\mathcal{X}$ can be transformed to the active subspace using the transformation matrix:

$$\mathbf{z} = \mathbf{U}^{\mathrm{T}} \mathbf{x} \tag{12}$$

and the function g represents the original function f in the active subspace as

$$g(\mathbf{z}) = g(\mathbf{U}^{\mathrm{T}} \mathbf{x}) \approx f(\mathbf{x}) \tag{13}$$

Once the transformation matrix U is found, all evaluated design points from the objective function f are projected to the active subspace

$$\mathrm{Z}_N = \mathrm{U}^{\mathrm{T}} \mathrm{X}_N \tag{14}$$

and a new Gaussian process is built using the projected design points in the active subspace

$$g(\mathbf{z}) \mid \mathbf{Z}_N, \mathrm{y}_N \approx \mathcal{N}\left(\mu(\mathbf{z}), \sigma^2(\mathbf{z})\right) \tag{15}$$

where $g(\mathbf{z})$ is the posterior distribution of the objective function in the active subspace. Now, we seek to learn the objective function g in the active subspace instead of the original objective function f.

Employing the active subspace method requires applying the acquisition function (knowledge gradient in this work) in 2 steps. The first step is applied over the objective function g, searching the active subspace for a query candidate which maximizes the utility function defined in Eq. (7) using the knowledge gradient as the heuristic

$$\mathbf{z}^* = \arg\max_{\mathbf{z} \in \mathcal{Z}_f} \nu^{\mathrm{KG},N}(\mathbf{z}) \tag{16}$$

where $\mathcal{Z}_f$ is a set of alternative design points in the active subspace generated by Latin Hypercube space filling technique to choose the next best design to query. Next, when $\mathbf{z}^*$ is defined, it needs to be mapped back to $\mathcal{X}$, however, the challenge is there are infinite number of solutions for this mapping back problem. We suggest a method to overcome this problem by using the orthogonality of eigen vectors of a symmetric matrix. Note that we divided the eigen vectors of covariance matrix C in Eq. (10) into matrices U and V in Eq. (11). Now, the subspace formed by the linear combination of eigen vectors in V is a hyperplane orthogonal to the active subspace:

$$\overrightarrow{\mathrm{P}} = \sum_{k=1}^{m-n} a_k \overrightarrow{\mathrm{e}}_k \tag{17}$$

where a_k is a random number that for simplicity, is generated from 0 to 1 and $\overrightarrow{\mathrm{e}}_k$ is an eigen vector in V. Then, by writing the equation of vectors generated by linearly combining eigen vectors in V passing from a point in $\mathcal{X}$ with the projection $\mathbf{z}^*$ on the active subspace, we obtain a subspace of $\mathcal{X}$ that is orthogonal to the active subspace with the same projection $\mathbf{z}^*$. Using the orthogonal vector $\overrightarrow{\mathrm{P}}$ and $\mathbf{x}^*$, a design with projection $\mathbf{z}^*$ on the active subspace, the equation of this orthogonal hyperplane is given by

$$\frac{x(1) - x^*(1)}{\mathrm{P}(1)} = \frac{x(2) - x^*(2)}{\mathrm{P}(2)} = \ldots = \frac{x(m) - x^*(m)}{\mathrm{P}(m)} = t \tag{18}$$

where a design vector in $\mathcal{X}$, $\mathbf{x} = [x(1), x(2), \ldots, x(m)]^{\mathrm{T}}$, is obtained by solving m sub-equations for a given t. To ensure that the design $\mathbf{x}$ remains within the bounds of the defined design space, before calculating Eq. (18), the lower and upper bounds for t should be found by putting the design boundaries in the equation and solving for different t values. Then, a range of values for t is selected to have every design point $\mathbf{x}$ corresponding to a particular t inside $\mathcal{X}$.

D. Batch Bayesian Optimization

Let us assume we have N previously evaluated designs regarding an objective function f as the prior knowledge denoted by $\mathbb{D}_N \equiv \{\mathbf{x}_i, y_i\}_{i=1}^{N}$. To perform Bayesian optimization, a Gaussian process is fit to this data to model the objective function f, using it to search for the best next experiment to run. The obtained representation of the function or in general, any inference about the function given the data is depending on hyperparameters chosen to initialize the Gaussian process. For example, in this work, we are using the squared exponential function as the kernel equipped with a Gaussian process and the hyperparameters are signal variance and length scales as defined in Eq. (4). Length scales have the largest impact on defining the characteristics of a Gaussian process while signal variance has less impact but still important to have an acceptable value, however, it is easier to set it to an optimal value based on data and expert opinion. Correct hyperparameters to fit a Gaussian process to data can be found by a maximum likelihood method [27]. A potential problem with optimizing hyperparameters is that, most often, Bayesian optimization frameworks are employed to optimize expensive to evaluate objective function and they work with minimal data [24]. Therefore, in sparse high-dimensional input spaces, their optimized values may be depended on the limited number of data available [26].

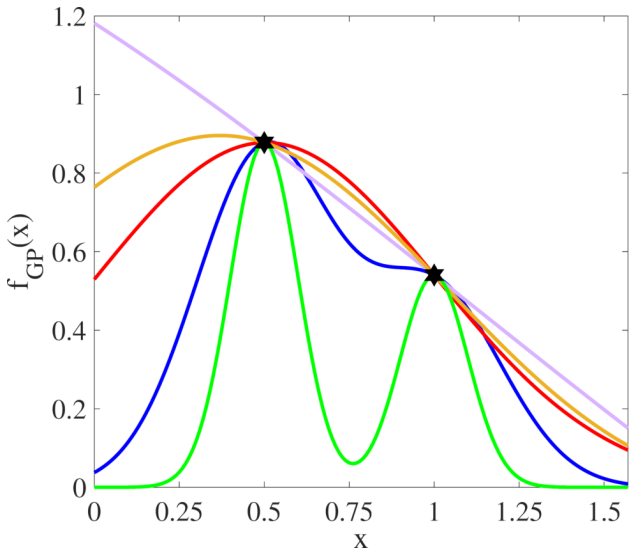

Fig 2. Different representations of an unknown function with two data points available.

In contrast to other Bayesian optimization methods that use a single model to represent the objective function, in batch Bayesian optimization, the idea is to build a variety of models by assigning different hyperparameters to these models. Thus, we are able to make different smoothness assumptions and reduce the risk of model mismatch as a results of setting improper hyperparameters [24]. Since each model makes a different inference about the objective function given data, they suggest different designs as the best next design to query when evaluating an acquisition function (knowledge gradient in this work):

$$\mathbf{x}_{1:n} = \arg\max_{\mathbf{x}\in\mathcal{X}} \nu^{\mathrm{KG},N}(\mathbf{x}|\mathrm{GP}(\mathbb{D}_N, \theta_{1:n})) \tag{19}$$

where there are n different Gaussian processes built with different set of hyperparameters, θ, given data $\mathbb{D}$. Once these query candidates are obtained, a k-medoid problem is solved to select k designs to query next. The value of k defines the number of queries we are able to make in parallel, known as the batch size. By solving the k-medoid problem, k clusters of data points are formed via choosing k points as medoids in a way to minimize the distance between data points and the medoids. In Ref [34], a batch Bayesian optimization framework is introduced that is capable of optimizing an objective function in presence of one or more models representing a quantity of interest.

In the following section, we will present a framework to employ the active subspace method and batch Bayesian optimization simultaneously to take advantage of both techniques to increase the efficiency of design optimization problems.

III. Approach

In this section, we introduce a framework to incorporate the active subspace method in batch Bayesian optimization. Both approaches are proposed separately for efficiency gain purposes in design problems, but we seek to exploit them at the same time to investigate if it is possible to further improve the design performance.

A step by step process of our proposed framework is presented in algorithm (1). Dataset $\mathbb{D}$ contains all data initially available observed from the objective function to be optimized. Number of alternative models, n, defines the number of Gaussian processes each with a different set of hyperparameters and the batch size, k, is an indication of possible parallel queries from the objective function. Once the Gaussian processes are initialized, the active subspaces according to different function representations are determined. Note that, while the data used to build the Gaussian processes is the same, the dissimilarity in hyperparameters results in different shape representation of the function. Therefore, the active subspaces that are obtained based on the function gradients are not identical. Next, dataset $\mathbb{D}$ needs to be transformed to these active subspaces and new Gaussian processes are constructed using the transformed data. An important point is, one needs to transform length-scale hyperparameters as well to comply with the dimensionality of the active subspaces. Now, we are ready to apply the first step of knowledge gradient to search for the best next experiment in each active subspace. Thus, a set of design points in each active subspace is generated. After making the decision in every active subspace, the selected designs are mapped back to the original design space. According to Eqs. (17) and (18), any design point in the subspace orthogonal to the active subspace defined by linear combination of eigen vectors in matrix $\mathbf{V}$ from Eq. (11), has the same projection on the active subspace and can be used to generate solutions to the inverse mapping problem. Accordingly, the second step of knowledge gradient is applied over the solution sets of inverse mapping problems regarding each active subspace.

Algorithm 1 : Implementing the active subspace method in batch Bayesian optimization

given Initial dataset $\mathbb{D}$, Number of alternative models n, and Batch size k.
Obtain n samples from the hyperparameter space via Latin Hypercube technique ($\theta_{1:n}$).
Initialize n Gaussian processes ($\mathrm{GP}_{1:n}$) given $\mathbb{D}$ and $\theta_{1:n}$.
repeat until termination
    Find the active subspaces corresponding to n Gaussian processes using Eqs. (9), (10), (11).
    Project the data available and length-scales in the original space $\mathcal{X}$ to the active subspaces $\mathcal{Z}_{1:n}$ using Eq. (12).
    Construct n GPs of the objective function in the active subspaces $\mathcal{Z}_{1:n}$ using projected data and length-scales.
    Generate Latin Hypercube samples in each active subspace.
    Apply the first-step knowledge gradient to select a design point in each active subspace using Eq. (16).
    Inverse map the selected design points to the original space using Eqs. (17), (18).
    Apply the second-step knowledge gradient to inverse mapped design points.
    Select the best design point in the original space regarding every different alternative model.
    Solve a k-medoid problem to select k design points from the set of candidate design points $\mathbf{x}_{1:n}$.
    Query k design points from the objective function to update the dataset $\mathbb{D}$ and $\mathrm{GP}_{1:n}$.
return The design point with the largest objective value in $\mathbb{D}$.

After finding the best query candidate in the original input space associated to every different alternative model or Gaussian process, a set of candidate design points, $\mathbf{x}_{1:n}$ is formed. Then, we need to cluster these design points where the number of clusters is defined by the batch size ,k. A k-medoid problem is solved to pick the most agreed samples and reduce the number of candidate queries to k design points, so that they can be evaluated in parallel from the objective function. In the final step of the process, the dataset $\mathbb{D}$ and all Gaussian processes are updated by adding newly observed data points. The whole process is repeated until a termination requirement such as exhausting the number of allowed function evaluations or computational time is met. The final solution to the design problem is determined by comparing objective value of all observed data points in $\mathbb{D}$ to pick the best design.

IV. Benchmark Applications

To investigate the performance of our proposed approach that combines the notions of batch Bayesian optimization and the active subspace method, it is applied on the test problems listed in Table 1 to look for the solution that maximizes each function. While Adjiman and Branin test problems provide highly smooth response surfaces, Himmelblau suggests greater variations in the objective space. Bird test problem is a more complex one to challenge the performance of our approach as it is a multi-modal function with larger function variations. Yet, all test problems are differentiable and smooth enough to make sure achieving reasonable fits in batch processing to obtain the gradients for finding the active subspaces respectively. All the test problems are defined on 2-dimensional input spaces with domains shown in Table 1.

Table 1. Test problem's input domains

Test Problem	Input Domain
Adjiman	x_1 = [-1 2] x_2 = [-1 1]
Bird	x_1 = [-5 5] x_2 = [-5 5]
Branin	x_1 = [0 15] x_2 = [0 15]
Himmelblau	x_1 = [-5 5] x_2 = [-5 5]

For comparison purposes, we have done the optimization with 4 different methods:

1) Std: standard sequential Bayesian optimization
2) ASM: sequential Bayesian optimization using the active subspace method
3) Batch: batch Bayesian optimization
4) Batch+ASM: batch Bayesian optimization using the active subspace method

A concern when comparing the batch and sequential optimization approaches is that in sequential optimization, the system has extra information achieved from every single query made in previous iterations of the process while in batch optimization, it selects a batch of design points to query and the information regarding any of these queries is only effective for the next batch of queries, not the queries in the same batch of design points. To make it clear, as an example, assuming the system is allowed to make only 2 queries from the expensive function. In sequential optimization, the system has the information obtained from the first query to make decision about the second query, however, this is not the case when querying a batch of 2 design points. On the other hand, batch Bayesian optimization has the advantage to make different inferences about the data, but it might not be as advantageous as adding data points to the system. To address this concern and balance the differences between sequential and batch optimization approaches, we define a query initialization cost to all cases. This way, no matter how many queries from the function will be made in a single iteration of the process, an initialization cost of one function evaluation will be added to the overall cost of an iteration. Therefore, every iteration of the sequential optimization costs 2 function evaluations. Followingly, if the batch size is set to k, the cost of each iteration will be $k + 1$ function evaluations. Thus, we are able to query more design points as the result of a larger batch size, but still, a proper batch size helps to balance number of function evaluations and exploit the information of previous queries.

Since the study of how changing system parameters impacts the optimization process is not in the scope of this work, we set them based on expert opinion to get nicely comparable results. For all the test problems, the batch size (k) and number of alternative models or Gaussian processes (n) are set to 3 and 60, respectively. The initial dataset ($\mathbb{D}$) for all cases contains 5 data points. Finally, the budget is limited to 100 function evaluations and all the results presented are averaged over 30 replication of simulations with different starting points.

In Fig. 3, the optimization results regarding all test problems are illustrated. The first conclusion is that using batch optimization and the active subspace method, the performance of optimization process is improved significantly in all cases except Bird test problem. That might be because its hard for the active subspace approach to identify the gradients in such multi modal functions. Although it is not possible to make a direct comparison between the batch and active subspace approaches, the results suggest smaller confidence intervals when performing batch optimization. Next, it is seen that by taking advantage of both batch optimization and the active subspace method in a single framework, the results are improved noticeably, outperforming all other methods. In general, it is shown that at any cost values, a better design suggesting greater objective value is found if both batch optimization and active subspace methods are employed.

Finally, note that in sequential optimization cases, the expensive function has been set up to evaluate 50 data points according to the total budget considered for the whole process. On the other side, in batch optimization with batch size

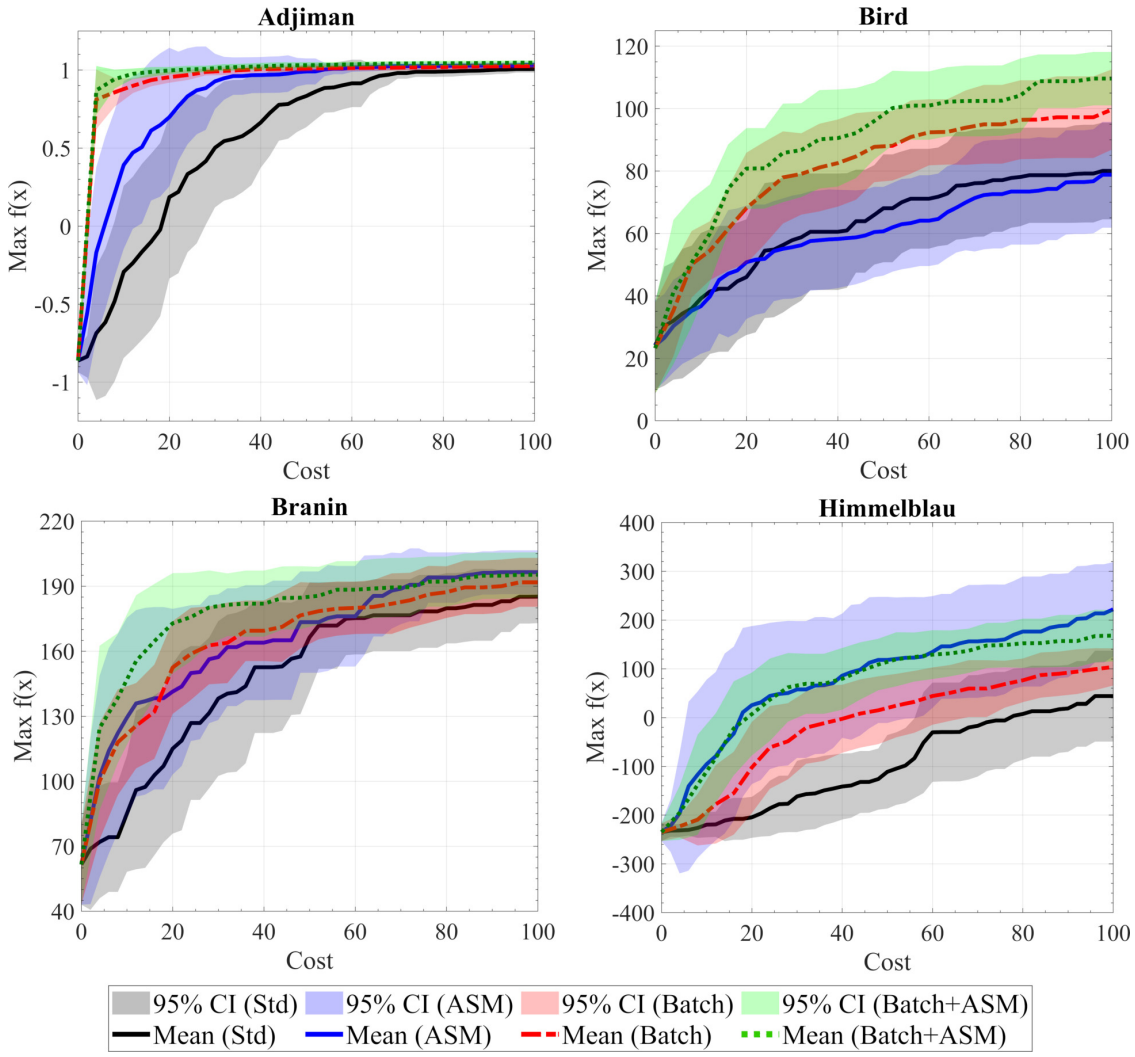

Fig 3. Results showing means and 95% confidence intervals of applying different Bayesian optimization techniques to optimize the test functions averaged over 30 replication of simulations. Std, ASM, Batch, and Batch+ASM correspond to standard sequential, active subspace method, batch, and combination of batch and active subspace method Bayesian optimization respectively.

of 3, the expensive function has been set up only 25 times and 75 data points are gathered. This setting is especially important when a real experiment needs to be done in a laboratory and setting up the process is complicated and time-consuming.

V. Demonstration Problem

In this section, we introduce a material design problem and the motivation behind using our proposed algorithm to solve the problem.

An important application of optimization algorithms in material science and engineering is to design a material to obtain a desired mechanical response. The design problem we are dealing with here is to optimize the mechanical response of dual-phase advanced high strength steels. This branch of materials are getting more attention than ever due to their applications in automotive industry [35]. Dual-phase steels consist of two phases, a soft phase and a harder martensite phase to strengthen the soft phase and define the overall characteristics of the steel. Designing a dual-phase steel to get a desired mechanical response is done by controlling the properties of different phases like each phase's volume fraction [36]. A finite element model is built to estimate the mechanical response of dual-phase

steels in different volume fractions of the hard martensite phase which is defined based on the annealing temperature and composition of alloying elements. The mechanical response in this case is Normalized Strain Hardening Rate of the material represented mathematically as $\frac{1}{\tau}(\frac{\mathrm{d}\tau}{\mathrm{d}\epsilon_{pl}})$ with τ as the stress value and ϵ as the strain level. the aim of the optimization problem here is to find the values of annealing temperature, T_{IA} and the composition of alloying elements X_C, X_{Mn} and X_{Si} to obtain the maximum normalized strain hardening rate.

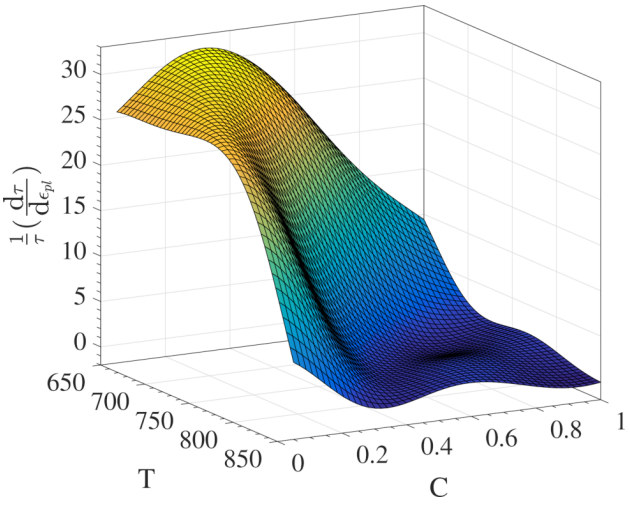

Fig 4. The response of finite element model at different annealing temperatures and Carbon weight percentages.

In Ref. [37] a sequential Bayesian optimization framework is employed to optimize the normalized strain hardening rate using all four design variables. However, for simplicity and since the goal in this work is to compare the performance of some Bayesian optimization techniques, we only seek to search in the space of temperature and carbon and other two variables are fixed. The annealing temperature ranges from 650°C to 850°C, and the weight percentage of X_C varies in the range [0.05-1]. The weight percentages of X_{Mn} and X_{Si} are set to 0.5. The response surface of the finite element model is illustrated in Fig. 4. We have got this illustration by fitting a surface to the sets of data points obtained in the works done in Refs. [36–38]. Note that in this work, we use a single model to estimate the objective value at different design points (single fidelity Bayesian optimization) while in Refs. [26, 32, 36–39] several models have been used to estimate the objective value in different accuracy (fidelity) levels and costs (multifidelity Bayesian optimization) and is shown to be a more efficient approach. Thus, extending our proposed approach to a multifidelity framework using the method used in Refs. [26, 32, 36–41] can be the subject of the future work.

Initializing the finite element model and running simulations are both very expensive procedures in matter of computational time. Therefore, efficiency gains are of interests in this regard via using batch optimization technique and employing the active subspace method. The same setting as in the test problems have been used here as well such as the batch size, number of alternative models, and number of starting data points. The budget is limited to 60 function queries, including the query initialization cost.

The results of applying all four methods, averaged over 30 replication of simulations with different starting points are shown in Fig. 5. Both approaches employing the batch optimization technique or the active subspace method are outperforming the standard sequential Bayesian optimization. Also, the large confidence intervals associated to the standard approach points to the fact that there are some simulations that the system has been struggling getting close to the optimal design region while the other approaches have quickly found the path toward the desired design region in every single simulation. These results also show that combining the batch and active subspace methods is suggesting the best outcome. This way, we have gathered the benefits of both approaches: using batch processing to reduce model mismatch risks, and employing the active subspace method to search only important regions of the design space.

Finally, the same set of experiments is run, but this time, all the initial observations are filtered to make sure they are far from the optimal design region. Basically, we are interested to see how quickly they find the optimal region and start querying the design points close to it. As seen in Fig. 6, while the standard optimization approach needs more resources to recover and get close to the optimal design region, other methods have found their way quickly from the initial iterations. Again, the combination of batch optimization and active subspace method has outperformed all other approaches, suggesting a very small uncertainty region around the mean as an indication of consistency of the approach.

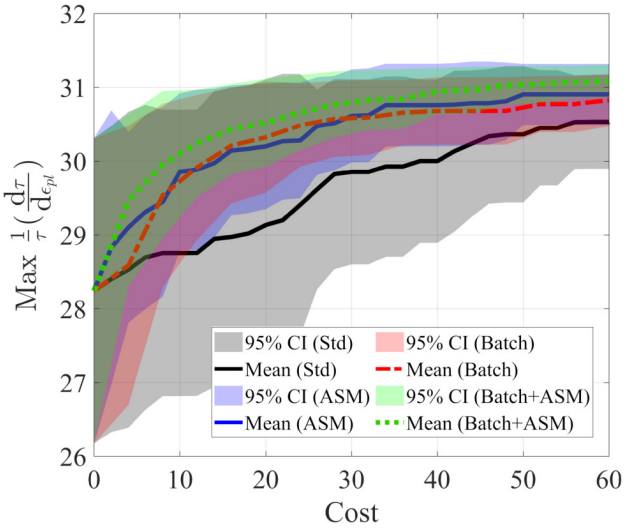

Fig 5. The means and 95% confidence intervals of largest objective values found as a function of cost, averaged over 30 replication of simulations with different starting points.

However, all approaches finally get there.

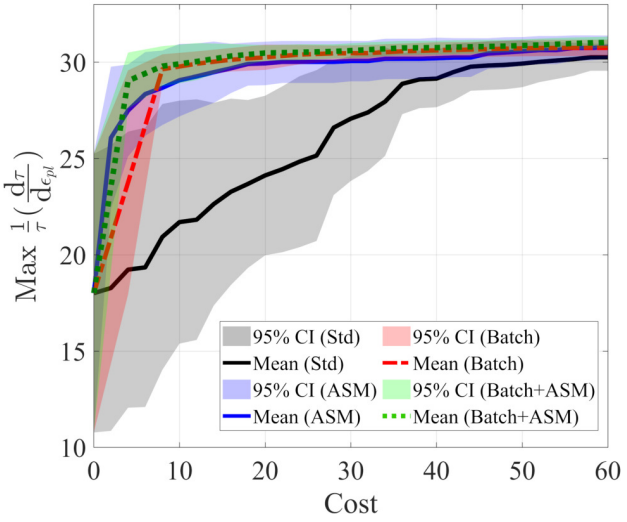

Fig 6. The largest objective values found as a function of cost, averaged over 30 replications with different starting points far from the optimal design region.

VI. Conclusions and Future Works

In this paper, we presented a new Bayesian optimization framework that implements the active subspace method and batch optimization process simultaneously. While the active subspace method and batch optimization approach have been developed separately aiming to improve the efficiency of Bayesian optimization frameworks, we have shown that taking advantage of both techniques in a single framework can further enhance the efficiency of a design process. To show the effectiveness of the proposed approach, it has been applied on different benchmark problems and the results are compared to standard Bayesian optimization, Bayesian optimization using the active subspace method and batch Bayesian optimization with the same initialization settings. It has been shown that, in addition to better performance, this new framework suggests more consistent results by comparing 95% confidence intervals around the achieved

maximum objective values at any cost value for all methods, obtained by replicating the simulations several times. Finally, the results of applying the method on a real-world engineering problem of designing a material to maximize a desired property has been presented. Since the objective function is a finite element model which is computationally expensive to evaluate, any improvements in the optimization process is demanding, and the results suggest a better design than other methods at any values of cost.

Since increasing the efficiency of design processes is desirable in engineering applications, in the future works, we aim to make more improvements to this proposed optimization framework by extending it to a multifidelity framework via incorporating multiple models estimating the same objective value in different fidelity levels. therefore, we are able to extract required information about true objective values without spending too much resources on directly evaluating the expensive objective function. Thus, better designs are expected at much lower costs.

Acknowledgments

References

[1] Močkus, J., "On Bayesian methods for seeking the extremum," *Optimization techniques IFIP technical conference*, Springer, 1975, pp. 400–404.

[2] Jones, D. R., Schonlau, M., and Welch, W. J., "Efficient global optimization of expensive black-box functions," *Journal of Global optimization*, Vol. 13, No. 4, 1998, pp. 455–492.

[3] Frazier, P. I., "A tutorial on bayesian optimization," *arXiv preprint arXiv:1807.02811*, 2018.

[4] Snoek, J., Larochelle, H., and Adams, R. P., "Practical bayesian optimization of machine learning algorithms," *Advances in neural information processing systems*, Vol. 25, 2012, pp. 2951–2959.

[5] Gill, P. E., Murray, W., and Wright, M. H., *Practical optimization*, SIAM, 2019.

[6] Shan, S., and Wang, G. G., "Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions," *Structural and multidisciplinary optimization*, Vol. 41, No. 2, 2010, pp. 219–241.

[7] Chinesta, F., Huerta, A., Rozza, G., and Willcox, K., "Model order reduction," *Encyclopedia of Computational Mechanics*, 2016.

[8] Benner, P., Gugercin, S., and Willcox, K., "A survey of projection-based model reduction methods for parametric dynamical systems," *SIAM review*, Vol. 57, No. 4, 2015, pp. 483–531.

[9] Russi, T. M., "Uncertainty quantification with experimental data and complex system models," Ph.D. thesis, UC Berkeley, 2010.

[10] Constantine, P. G., Dow, E., and Wang, Q., "Active subspace methods in theory and practice: applications to kriging surfaces," *SIAM Journal on Scientific Computing*, Vol. 36, No. 4, 2014, pp. A1500–A1524.

[11] Constantine, P. G., *Active subspaces: Emerging ideas for dimension reduction in parameter studies*, SIAM, 2015.

[12] Ghoreishi, S. F., Friedman, S., and Allaire, D. L., "Adaptive dimensionality reduction for fast sequential optimization with gaussian processes," *Journal of Mechanical Design*, Vol. 141, No. 7, 2019.

[13] Wang, Q., Chen, H., Hu, R., and Constantine, P., "Conditional sampling and experiment design for quantifying manufacturing error of transonic airfoil," *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, 2011, p. 658.

[14] Dow, E., and Wang, Q., "Output based dimensionality reduction of geometric variability in compressor blades," *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, 2013, p. 420.

[15] Lukaczyk, T. W., Constantine, P., Palacios, F., and Alonso, J. J., "Active subspaces for shape optimization," *10th AIAA multidisciplinary design optimization conference*, 2014, p. 1171.

[16] Li, J., Cai, J., and Qu, K., "Surrogate-based aerodynamic shape optimization with the active subspace method," *Structural and Multidisciplinary Optimization*, Vol. 59, No. 2, 2019, pp. 403–419.

[17] Hu, X., Parks, G. T., Chen, X., and Seshadri, P., "Discovering a one-dimensional active subspace to quantify multidisciplinary uncertainty in satellite system design," *Advances in space research*, Vol. 57, No. 5, 2016, pp. 1268–1279.

[18] Constantine, P. G., Emory, M., Larsson, J., and Iaccarino, G., "Exploiting active subspaces to quantify uncertainty in the numerical simulation of the HyShot II scramjet," *Journal of Computational Physics*, Vol. 302, 2015, pp. 1–20.

[19] Ginsbourger, D., Le Riche, R., and Carraro, L., "Kriging is well-suited to parallelize optimization," *Computational intelligence in expensive optimization problems*, Springer, 2010, pp. 131–162.

[20] Azimi, J., Jalali, A., and Fern, X., "Hybrid batch Bayesian optimization," *arXiv preprint arXiv:1202.5597*, 2012.

[21] Contal, E., Buffoni, D., Robicquet, A., and Vayatis, N., "Parallel Gaussian process optimization with upper confidence bound and pure exploration," *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Springer, 2013, pp. 225–240.

[22] Desautels, T., Krause, A., and Burdick, J. W., "Parallelizing exploration-exploitation tradeoffs in gaussian process bandit optimization," *Journal of Machine Learning Research*, Vol. 15, 2014, pp. 3873–3923.

[23] González, J., Dai, Z., Hennig, P., and Lawrence, N., "Batch bayesian optimization via local penalization," *Artificial intelligence and statistics*, 2016, pp. 648–657.

[24] Joy, T. T., Rana, S., Gupta, S., and Venkatesh, S., "Batch Bayesian optimization using multi-scale search," *Knowledge-Based Systems*, Vol. 187, 2020, p. 104818.

[25] Wu, J., and Frazier, P., "The parallel knowledge gradient method for batch bayesian optimization," *Advances in Neural Information Processing Systems*, 2016, pp. 3126–3134.

[26] Couperthwaite, R., Molkeri, A., Khatamsaz, D., Srivastava, A., Allaire, D., and Arròyave, R., "Materials Design Through Batch Bayesian Optimization with Multisource Information Fusion," *JOM*, 2020, pp. 1–13.

[27] Rasmussen, C. E., and Williams, C. K. I., *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*, The MIT Press, 2005.

[28] Srinivas, N., Krause, A., Kakade, S. M., and Seeger, M., "Gaussian process optimization in the bandit setting: No regret and experimental design," *arXiv preprint arXiv:0912.3995*, 2009.

[29] Powell, W. B., and Ryzhov, I. O., *Optimal learning*, Vol. 841, John Wiley & Sons, 2012.

[30] Frazier, P. I., Powell, W. B., and Dayanik, S., "A knowledge-gradient policy for sequential information collection," *SIAM Journal on Control and Optimization*, Vol. 47, No. 5, 2008, pp. 2410–2439.

[31] Frazier, P., Powell, W., and Dayanik, S., "The knowledge-gradient policy for correlated normal beliefs," *INFORMS journal on Computing*, Vol. 21, No. 4, 2009, pp. 599–613.

[32] Khatamsaz, D., Molkeri, A., Couperthwaite, R., James, J., Arróyave, R., Srivastava, A., and Allaire, D., "Adaptive active subspace-based efficient multifidelity materials design," *Materials & Design*, Vol. 209, 2021, p. 110001.

[33] Constantine, P., and Gleich, D., "Computing active subspaces with Monte Carlo," *arXiv preprint arXiv:1408.0545*, 2014.

[34] Couperthwaite, R., Khatamsaz, D., Molkeri, A., James, J., Srivastava, A., Allaire, D., and Arróyave, R., "The BAREFOOT Optimization Framework," *Integrating Materials and Manufacturing Innovation*, 2021, pp. 1–17.

[35] Bhattacharya, D., "Metallurgical perspectives on advanced sheet steels for automotive applications," *Advanced steels*, Springer, 2011, pp. 163–175.

[36] Ghoreishi, S. F., Molkeri, A., Arróyave, R., Allaire, D., and Srivastava, A., "Efficient use of multiple information sources in material design," *Acta Materialia*, Vol. 180, 2019, pp. 260–271.

[37] Khatamsaz, D., Molkeri, A., Couperthwaite, R., James, J., Arróyave, R., Allaire, D., and Srivastava, A., "Efficiently exploiting process-structure-property relationships in material design by multi-information source fusion," *Acta Materialia*, Vol. 206, 2021, p. 116619.

[38] Ghoreishi, S. F., Molkeri, A., Srivastava, A., Arroyave, R., and Allaire, D., "Multi-information source fusion and optimization to realize ICME: Application to dual-phase materials," *Journal of Mechanical Design*, Vol. 140, No. 11, 2018.

[39] Molkeri, A., Khatamsaz, D., Couperthwaite, R., James, J., Arróyave, R., Allaire, D., and Srivastava, A., "On the Importance of Microstructure Information in Materials Design: PSP vs PP," *Acta Materialia*, 2021, p. 117471.

[40] Ghoreishi, S. F., and Allaire, D. L., "A fusion-based multi-information source optimization approach using knowledge gradient policies," *2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2018, p. 1159.

[41] Khatamsaz, D., and Allaire, D. L., "A Comparison of Reification and Cokriging for Sequential Multi-Information Source Fusion," *AIAA Scitech 2021 Forum*, 2021, p. 1477.