JOM, Vol. 72, No. 12, 2020
https://doi.org/10.1007/s11837-020-04396-x

AUGMENTING PHYSICS-BASED MODELS IN ICME WITH MACHINE LEARNING AND UNCERTAINTY QUANTIFICATION

Materials Design Through Batch Bayesian Optimization with Multisource Information Fusion

RICHARD COUPERTHWAITE,[1,3] ABHILASH MOLKERI,[1] DANIAL KHATAMSAZ,[2] ANKIT SRIVASTAVA,[1] DOUGLAS ALLAIRE,[2] and RAYMUNDO ARRÓYAVE[1,2]

1.—Materials Science and Engineering Department, Texas A&M University, College Station, TX 77843, USA. 2.—J. Mike Walker '66 Department of Mechanical Engineering, Texas A&M University, College Station, TX 77843, USA. 3.—e-mail: richardcouperthwaite@tamu.edu

Integrated computational materials engineering (ICME) calls for the integration of simulation tools and experiments to accelerate the development of materials. ICME approaches tend to be computationally costly, and recently, Bayesian optimization (BO) has been proposed as a way to make ICME more resource efficient. Conventional BO, however, is sequential (i.e., one-at-a-time) in nature, which makes it very time-consuming when the evaluation of a materials design choice is costly. While conventional high-throughput approaches enable the synthesis and characterization (or simulation) of materials in a parallel manner, they tend to be "open loop" and are unable to provide recommendations of what to try next once the parallel experiment/simulation has been carried out and analyzed. Here, we address this problem by introducing a batch BO framework that enables the exploration of the material's design space in a parallel fashion. We augment this approach by incorporating information fusion frameworks capable of integrating multiple information sources. Demonstrating the proposed approach in the computational design of dual-phase steel, we show that batch BO can result in a significant reduction in the time and resources needed to carry out the design task. The proposed approach has wider applicability, well beyond the ICME example used to demonstrate it.

Key words: Computational materials science and engineering, knowledge gradient, reification, model fusion

INTRODUCTION

Integrated computational materials engineering (ICME)[1] calls for the integration of various computational tools (validated against experiments) to establish quantitative process–structure–property–performance (PSPP) relationships. Inverting these relationships can accelerate the design of materials—under the key assumption that simulations are faster and cheaper than experiments. However, there are still significant challenges to this approach.

(Received June 25, 2020; accepted September 18, 2020; published online October 13, 2020)

Despite the assumption that simulations are cheaper than experiments, a major drawback of ICME implementations is the considerable computational cost associated with evaluating PSPP chains. This has recently been addressed through the deployment of Bayesian optimization (BO) to efficiently balance the exploration and exploitation of materials design spaces.[2,3]

Furthermore, most ICME frameworks tend to assume that there is a single information source (i.e., model) per linkage along the PSPP chain. In recent work,[4,5] however, we showed that this is an unnecessary limitation, as the combination of multiple information sources—each containing at least some useful information about the problem

space—always results in significant improvements in the efficiency of ICME-based alloy design schemes.

A limitation shared by modern (i.e., BO-based) and more traditional ICME frameworks is that the vast majority of them query PSPP relationships in a sequential manner (i.e., one-at-a-time). In computational settings, such sequential exploration of materials spaces is far from effective, given the availability of high-performance research computing (HPRC) facilities that make high-throughput materials simulations relatively straightforward. When it comes to experimental materials science, there has been sustained growth in the number of synthesis and characterization approaches amenable to parallelization. Indeed, the Materials Genome Initiative (MGI)[6] stimulated the development of high-throughput (i.e., combinatorial) experimental[7–10] or computational[11] schemes as a way to accelerate the exploration of materials design spaces.

High-throughput experimental methods tend to involve thin-film[12] combinatorial libraries, although, very recently, additive manufacturing platforms have been used for parallel synthesis of alloys.[8] While optical and electrical properties are most easily measured in a high-throughput fashion,[8] recent approaches have shown that it is possible to rapidly measure other material properties such as composition and microstructure,[9,10] hardness,[10] and even the transformation temperature of shape-memory alloy thin films.[7] These high-throughput approaches, while highly advantageous, suffer from the fact that they tend to be open-loop, one-shot approaches, as they lack principled policies to integrate the information gained from the high-throughput exploration to decide what to do next once the first information-gathering step has been taken.

There are significant opportunities to further improve BO-based ICME approaches[4,5] by incorporating the ability to query the materials design space in a parallel fashion. This would combine the advantages of ICME (i.e., closed loops) and combinatorial materials science while addressing their common limitations (i.e., agnosticism regarding resource constraints). We note, however, that such an approach would also significantly benefit exclusively experimental combinatorial materials science efforts, including recently proposed concepts such as self-driven laboratories[13] — as well as their computational counterparts[14]—that, so far, have been implemented using sequential BO schemes.

The challenge of exploring and exploiting design spaces in a parallel and optimal fashion can be set as a general problem of Batch Bayesian Optimization (BBO). The key challenge to BBO is how to carry out such balanced exploration/exploitation in parallel while maintaining optimality throughout the process. Some common approaches to BBO include multistep look-ahead policies[15,16] where the batch is created by sequentially adding the predictions from the surrogate model and predicting a new best point. Another approach considered adding queries that maximize the variance after each sequential addition from the surrogate model.[17,18] A third approach attempts to extract multiple peaks from the same acquisition function by removing peaks that have already been identified.[19]

A more recent approach by Joy et al.[20] considers a slightly more intuitive approach: in sparsely sampled high-dimensional problem spaces, it is too risky to place too much confidence on the tuning of the hyperparameters of the surrogate model, as the latter will depend heavily on the data captured thus far. Instead, Joy et al. assume that the hyperparameters can take any possible value (within reasonably set bounds) and proceed to carry out BO over all the surrogate models that result from sampling the hyperparameter space. The predictions from this batch of BO optimizations are clustered according to the number of samples that will be evaluated in the next step, as described below.

The current work combines the approaches of Ghoreishi et al.[4,5] for multi-information-source BO and Joy et al.[20] for batch BO into a single framework. Additionally, a thermodynamic model connecting chemistry and processing conditions to microstructure phase constitution is connected to the microstructural mechanics models to establish a (chemistry) processing–structure–property chain. The framework is demonstrated using the same four micromechanical models used in Ref. 4, namely the isowork, isostress, and isostrain reduced-order models, as well as a finite element representative volume element (RVE) micromechanical model, connecting the microstructure of a dual-phase steel to its mechanical response. We start by presenting each of the elements of the framework and proceed to evaluate its performance under different policies for continuation/termination of the optimization loop.

METHODS

The design objective of the current work is the maximization of the normalized strain-hardening rate ($1/\tau(\mathrm{d}\tau/\mathrm{d}\varepsilon_{\mathrm{pl}})$) of dual-phase high-strength steel. This parameter was chosen as it is an indication of ductility and formability, with higher values indicating better ductility and formability. The current work considers the optimization of a dual phase (martensite–ferrite) composed of Fe, C, Mn, and Si. The material is considered to undergo a single-stage intercritical annealing heat treatment followed by rapid quenching. For simplification, the only parameters optimized are the concentration of C (wt.%) and the intercritical annealing temperature. The Mn and Si compositions are kept constant in the

current work. The ranges for the two parameters and the composition of Mn and Si are presented in Table I.

The first part of this section deals with descriptions of the individual computational tools used in both of these methods. These descriptions are not a detailed analysis of the methods, and interested readers are directed to the various references, should further information on the methods be required. Later in this section, we explain how these computational tools are combined into the current framework. After constructing the framework, the optimization process is tested using three different case studies, describing the parameters and termination criteria used in each of the case studies.

Computational Tools

Gaussian Process

One of the major ingredients in BO is a surrogate model capable of predicting the outcome of experiments yet to be carried out, as well as the uncertainty associated with those predictions.[21] In BO problems, such predictive models tend to be constructed out of Gaussian processes (GPs) due to their underlying mathematical properties (including smoothness, controllable modeled correlation among observed points, etc.). A GP is a nonparametric statistical model that defines a stochastic process $f(\boldsymbol{x})$, where all the finite distributions of the model are assumed to be multivariate normal. Using this definition, the joint probability distribution of the outputs from the stochastic process may be modeled as an n-dimensional multivariate normal distribution for any finite set of inputs $\mathbb{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$,

$$p(f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_n)) \sim \mathcal{N}_n(\boldsymbol{\mu}, \boldsymbol{C}), \tag{1}$$

where $\boldsymbol{\mu}$ is the mean vector and $\boldsymbol{C}$ is the covariance function. The mean and covariance are defined by a mean function $\mu(\cdot)$ and a covariance function $C(\cdot,\cdot)$ with the following properties:

$$\mu(\boldsymbol{x}_i) = \boldsymbol{\mu}_i = \mathbb{E}[f(\boldsymbol{x}_i)], \tag{2}$$

$$C(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{C}_{i,j} = cov\left[f(\boldsymbol{x}_i), f(\boldsymbol{x}_j)\right]. \tag{3}$$

Table I. The optimization approach was conducted on dual-phase steel alloyed with C, Mn, and Si. The aim is to optimize the carbon content and intercritical annealing temperature in the range shown to obtain a maximum in the normalized strain-hardening rate

T_{IA} [°C]	X_C [wt.%]	X_{Mn} [wt.%]	X_{Si} [wt.%]
650–850	0–1	0.328	0.283

From the above definitions, we formally define a Gaussian process as $f(\cdot) \sim GP(\mu, C)$. A more detailed explanation of this kind of stochastic process is provided in the work by Rasmussen and Williams.[22]

The covariance function of the GP captures the degree of correlation between two different locations in the input space. The ability to make explicit inferences (through well-defined covariances) about the degree to which observations are correlated is one of the reasons why GPs tend to be the model class of choice in BO. The assumption that current information about the state of a system can be used to infer yet-to-be-observed states is inherent to BO. Due to the difference in scales (e.g., temperature and compositions) between the inputs of the current approach, the inputs were rescaled to the interval [0, 1]. Therefore, the notion of space is abstract and the spatial dependence denotes a metric representing the distance between two points in a mathematical space. In the current work, the Matérn class of covariance functions was used, since they are generally more robust when the smoothness of the data is not known:[23]

$$C(\mathbf{x}_i, \mathbf{x}_j) = \sigma_f^2 \left[\frac{2^{1-v}}{\Gamma(v)} \left(\frac{\sqrt{2v}(x_i - x_j)}{l} \right)^v K_v \left(\frac{\sqrt{2v}(x_i - x_j)}{l} \right) \right]. \tag{4}$$

The Matérn class of covariance functions is defined by Eq. 4, where σ_f^2 is referred to as the signal variance, l is the characteristic length scale, and v is a parameter that determines the shape of the function. However, it is more common to define the function by specifying a value for v. The function has a closed-form solution for values of $v = r + 1/2$, where $r \in \mathbb{Z}^+$. One of the more common values is $v = 5/2$.[22] This choice reduces the covariance function to the form shown in Eq. 5.

$$C(\mathbf{x}_i, \mathbf{x}_j) = \sigma_f^2 \left(1 + \frac{\sqrt{5}(x_i - x_j)}{l} + \frac{5(x_i - x_j)^2}{3l^2} \right) \exp\left(-\frac{\sqrt{5}(x_i - x_j)}{l} \right). \tag{5}$$

Reification

A key ingredient of the present framework is the simultaneous consideration of multiple information sources at once and their fusion to achieve better, unbiased predictions that take advantage of all the useful information provided by each model individually.[5] Information fusion requires the quantification of the statistical correlations among the different information sources and between the sources and the ground truth. The reification

method developed by Thomison and Allaire[24] estimates the model correlations by sequentially elevating each model at a time as "truth" (i.e., the model is "reified"), followed by the computation of the statistical correlation between this reified model and the other sources.

Assuming that we have two models, $f_1(x)$ and $f_2(x)$, that can both estimate the quantity of interest (y) with some discrepancy,

$$y = f_1(x) = \bar{f}_1(x) + \delta_1(x), \tag{6}$$

$$y = f_2(x) = \bar{f}_2(x) + \delta_2(x), \tag{7}$$

where $\bar{f}_1(x)$ is the mean prediction and the model discrepancies $\delta_i(x)$ are assumed to be normally distributed with $\delta_1(x) \sim \mathcal{N}(0, \sigma_1^2)$ and $\delta_2(x) \sim \mathcal{N}(0, \sigma_2^2)$.

Using this information, we reify model 1 and then calculate the error of each model. Since model 1 has been reified, the standard deviation of model 1 ($\tilde{f}_1(x^*)$) at a single point in the design space (x^*) is defined simply by the model discrepancy as

$$\tilde{f}_1(x^*) = f_1(x^*) - \bar{f}_1(x^*) = \delta_1(x^*), \tag{8}$$

and the error for model 2, with respect to model 1, is defined by

$$\tilde{f}_2(x^*) = f_2(x^*) - \bar{f}_2(x^*), \tag{9}$$

$$= \bar{f}_1(x^*) - \bar{f}_2(x^*) + \delta_1(x^*). \tag{10}$$

To calculate the correlation it is necessary to calculate both the mean squared errors and the covariance. Using the errors above, the mean squared errors are defined by

$$\mathbb{E}[\tilde{f}_1(x^*)^2] = \mathbb{E}[\delta_1(x^*)] = \sigma_1^2, \tag{11}$$

$$\mathbb{E}[\tilde{f}_2(x^*)^2] = \mathbb{E}[(\bar{f}_1(x^*) - \bar{f}_2(x^*))^2] + \mathbb{E}[\delta_1(x^*)], \tag{12}$$

$$= (\bar{f}_1(x^*) - \bar{f}_2(x^*))^2 + \sigma_1^2, \tag{13}$$

while the covariance is given by

$$\mathbb{E}[\tilde{f}_1(x^*)\tilde{f}_2(x^*)] = \sigma_1^2. \tag{14}$$

The Pearson correlation coefficient (ρ) can then be calculated as

$$\rho_1(x^*) = \frac{\sigma_1^2}{\sigma_1\sigma_2} = \frac{\sigma_1}{\sqrt{(\bar{f}_1(x^*) - \bar{f}_2(x^*))^2 + \sigma_1^2}}, \tag{15}$$

where the subscript on the coefficient indicates which model has been reified. This process is repeated for the other model to obtain the value of $\rho_2(x^*)$. When more than two models are used, the correlation coefficients are calculated for each pair of models. The average correlation ($\bar{\rho}$) is used in the model fusion approach and is calculated as follows:

$$\bar{\rho}(x^*) = \frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2}\rho_1(x^*) + \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2}\rho_2(x^*). \tag{16}$$

Theoretically, this reification approach can be expanded to any number of models. However, there are practical limits to how many models can be considered, based on the computational resources available as well as the time necessary to compute all the relevant pairwise correlations. In most cases it is unlikely that such a computational limit can be reached, since the number of models/sources corresponding to every linkage of the PSPP chain is likely to be generally modest.

Model Fusion

Given Eqs. 6 and 7 for the two models that estimate the quantity of interest (y), the fused model can be represented by the equation

$$y = k_1(x^*)f_1(x^*) + k_2(x^*)f_2(x^*), \tag{17}$$

where $k_1(x^*)$ and $k_2(x^*)$ are real-valued scalar quantities subject to $k_1(x^*) + k_2(x^*) = 1$. By assuming that both models have a normal distribution given by $f_1(x^*) \sim \mathcal{N}(\bar{f}_1(x^*), \sigma_1^2)$ and $f_2(x^*) \sim \mathcal{N}(\bar{f}_2(x^*), \sigma_2^2)$, it is possible to solve Eq. 17 for $k_1(x^*)$ and $k_2(x^*)$ by solving the following minimization problem:

$$\min_{\mathbf{k}} \mathbf{k}^T \Sigma \mathbf{k} \text{ subject to } k_1 + k_2 = 1 \tag{18}$$

where $\mathbf{k} = [k_1, k_2]^T$ and

$$\Sigma = \begin{bmatrix} \mathbb{E}[\tilde{f}_1(x^*)^2] & \mathbb{E}[\tilde{f}_1(x^*)\tilde{f}_2(x^*)] \\ \mathbb{E}[\tilde{f}_2(x^*)\tilde{f}_1(x^*)] & \mathbb{E}[\tilde{f}_2(x^*)^2] \end{bmatrix} = \begin{bmatrix} \sigma_1^2 & \bar{\rho}\sigma_1\sigma_2 \\ \bar{\rho}\sigma_2\sigma_1 & \sigma_2^2 \end{bmatrix}. \tag{19}$$

The covariance matrix, Σ, requires the correlation coefficient, ρ. This is approximated using the reification approach outlined previously that provides $\bar{\rho}$ as an estimate of this quantity. The solution of this minimization problem defines a fused model for y that has a mean defined by

$$\mathbb{E}[y] = \frac{(\sigma_2^2 - \bar{\rho}\sigma_1\sigma_2)\bar{f}_1(x^*) + (\sigma_1^2 - \bar{\rho}\sigma_1\sigma_2)\bar{f}_2(x^*)}{\sigma_1^2 + \sigma_2^2 - 2\bar{\rho}\sigma_1\sigma_2} \tag{20}$$

and variance

$$\mathrm{Var(y)} = \frac{(1-\rho^2)\sigma_1^2\sigma_2^2}{\sigma_1^2 + \sigma_2^2 - 2\rho\sigma_1\sigma_2}. \tag{21}$$

The proof and full derivation of these equations can be found in the work by Winkler.[25] While it is not considered in the current work , it is worthwhile to note that this reification fusion approach could be used to estimate the impact of parameter

uncertainty on the BO itself by constructing multiple models with different parameters, using the reification approach to weigh the importance of each of the models, relative to the "ground truth."

Knowledge Gradient

The second ingredient[21] of any BO approach is the acquisition function or policy that is used to select the next experiment (or simulation/observation) to carry out, given the data acquired thus far, as well as the underlying model (i.e., GP with hyperparameters) used to represent the problem space. In BO, there are a large number of acquisition functions that can be used, including probability of improvement (PI),[26] expected improvement (EI),[27,28] upper confidence bound (UCB),[29] and knowledge gradient (KG).[30] In this work, we have selected the KG as it tends to be better suited to potentially noisy problem spaces,[31] although it should be pointed out that KG is considerably more expensive to compute than other acquisition functions, including EI, PI, and UCB.

For the calculation of the knowledge gradient, we define a set of M distinct alternative points in the fused model input space and evaluate the mean, μ_x^n, and variance, $(\sigma_x^n)^2$, using the posterior predictive distribution of the fused model. The superscript n denotes the iteration number. KG is then defined as

$$v^{KG} = \max_{x^n \in \{1,\ldots,M\}} \mathbb{E}_n \left[\left(\max_{x'} \mu_{x'}^{n+1} \right) - \left(\max_{x'} \mu_{x'}^{n} \right) \right], \quad (22)$$

where $\mathbb{E}_n$ is the conditional expectation with respect to what is known after the first n iterations and μ_x^{n+1} is the Bayesian look-ahead prediction of the mean at step $n+1$. The knowledge gradient approach uses a Bayesian look-ahead approach to estimate μ_x^{n+1} conditional on μ_x^n and $(\sigma_x^n)^2$. This is done by first defining the precision of the posterior predictive distribution as $\beta_x^n = (\sigma_x^n)^{-2}$. According to the work by Frazier et al.[30], the conditional variance for the look-ahead step is defined by

$$\tilde{\sigma}(\beta_x^n) = \sqrt{(\beta_x^n)^{-1} - (\beta_x^n + \beta^\epsilon)^{-1}}, \quad (23)$$

where β^ϵ is the measurement precision and is generally assumed to be constant over the entire input space. Then, the look-ahead mean is defined as

$$\mu_x^{n+1} = \mu^n + \tilde{\sigma}(\beta_x^n) Z e_x, \quad (24)$$

where Z is the standard normal distribution and e_x is a vector in $\mathbb{R}^M$ with all components zero except for component x. For a full description of the method and algorithm implemented in the current work, refer to the work by Frazier et al.[30]

Batch Bayesian Optimization

Given the formulation for the GP covariance function shown in Eq. 5, the hyperparameters are σ_f, σ_n, and l. These three hyperparameters and the available data determine the shape of the GP. The characteristic length scale, l, will possibly have the greatest effect, but the other two hyperparameters also play a role. Usually, the hyperparameters are determined by minimizing the log-marginal likelihood of the GP, given the data. This is typically done by either gradient-based optimization approaches or BO methods.[22] Unfortunately, when faced with relatively sparse high-dimensional input spaces, the optimized values of the hyperparameters may be extremely dependent on the data already available, and it is thus too risky to make such definite inferences about the covariance structure of the entire problem space, and to use this assumed correlation to evaluate the BO acquisition policy.

Joy et al.[20] propose that, under data-sparse conditions, rather than selecting single values for each of the hyperparameters, it is instead advisable to sample a wide range of hyperparameters (within reasonable bounds), thereby making no assumption with regards to the shape of the underlying objective function and on the degree of correlation between points in the design space. It follows that each set of hyperparameters sampled through this framework would result in different predictions as to the location of the next best point to query, given the current knowledge of the system and the acquisition function used:

$$\boldsymbol{x}_{1:n} = \underset{x \in \chi}{\operatorname{argmax}} \; v^{\mathrm{KG}} \left(\mathbf{x} \,|GP(\mathbf{D}_0,\ \boldsymbol{\theta}_{1:n}) \right), \quad (25)$$

where the acquisition function, in this case, is the knowledge gradient (as defined above), $\boldsymbol{D}_0$ is the data available at the start of the iteration, and n is the number of sets of hyperparameter values ($\boldsymbol{\theta}_{1:n}$).

After acquiring all of these predicted "best design points," it is then possible to cluster them into the number of clusters (K) required by the size of the batch processing step. This is done using a k-medoids approach, which clusters the samples to minimize the total distance between the samples and the selected medoids. The number of medoids (K) is defined by the size of the batch available to query the problem space. The difference between this approach and a k-means approach is that the medoids are samples in the dataset rather than the arbitrary centroids predicted by k-means that may not necessarily exist in the data acquired thus far. This clustering approach defines K points that can be queried from the information source, in parallel. For a more in-depth discussion of the technique, refer to the Electronic Supplementary Materials.

Mechanical Models

Three reduced-order models and one finite element micromechanical model are used in the current work. The reduced-order models represent different approaches to homogenize the response of a composite microstructure based on different assumptions on the nature of the interactions/coupling among the constituent phases in the composite:

1. An isostrain model, where the strain is assumed to be the same in both phases[32]
2. An isostress model, where the assumption is made that the stress is homogeneous throughout the composite[33]
3. An isowork model, where the mechanical work in the two phases is assumed to be the same[34]

The "ground truth" in the current work corresponds to the simulation of the deformation behavior of a representative volume element (RVE) representation of the dual-phase microstructure through the use of finite elements. All models include isotropic hardening that followed Ludwik's power law.[35] The strength of the two phases was dependent on the composition based on the assumption that only carbon affected the martensite strength, while manganese and silicon affected the ferrite strength. Further details of these models can be found in the Electronic Supplementary Materials.

For comparison, the output of each of the low-order models is compared with the output from the RVE model (Fig. 1). In the optimization calculations, a surrogate model was used in place of the true RVE model to speed up the calculations in the framework.

Current Approach

The previous section provides details on the methods applied in the current work. This section explains the overall algorithmic approach used in the current work. A schematic showing the general flow of the framework is shown in Fig. 2.

The first step in the current approach is to define the hyperparameter sets to be used to generate the fused model GP. These hyperparameter sets are constructed using Latin hypercube sampling in the bounds of the hyperparameter space. In the current work, this space is defined as $l_h \in [0.01, 20]$, and $\sigma_f \in [0.01, 100]$. The noise variance hyperparameter was set to a constant value of $\sigma_n^2 = 0.1$. A total of 500 different hyperparameter sets were defined, and these were kept constant throughout the optimization process. After defining the hyperparameter

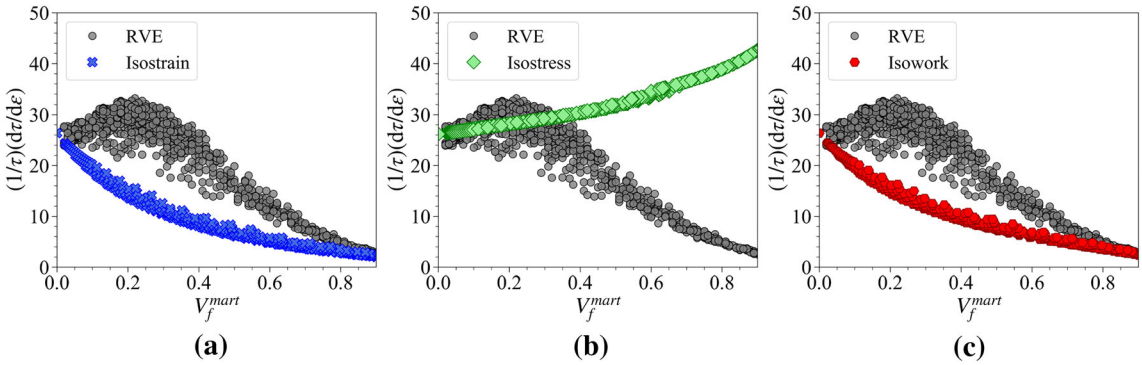

Fig. 1. Comparison of outputs of three reduced-order models and RVE finite element model: (a) isostrain, (b) isostress, and (c) isowork.

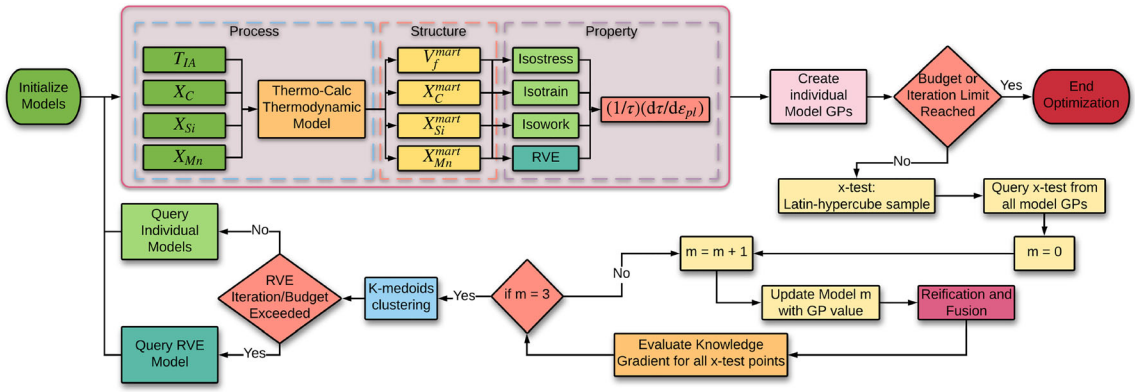

Fig. 2. Schematic overview of method applied in current work.

sets, we selected two random points within the design space. These two points are queried from all three reduced-order models and the RVE model as initial data.

In the discussion below, the reduced-order models are collectively referred to as the object ***Models***. Where each reduced-order model is indicated by an index in the set [1, 2, 3]. For example, ***Models***[***1***] refers to the isostrain reduced-order model. These model objects contain the X and Y data that has been evaluated for that particular model. Using this data, it is possible to construct a GP for that model. These GPs are denoted as ***Models***[·]***_GP***. In the current work, we assume that the hyperparameters defining the GPs of these models are known *a priori* as their extremely low computational cost makes their full evaluation over the design space highly feasible.

At the start of each iteration, a set of $\boldsymbol{x}_{\text{test}}$ vectors are defined using Latin hypercube sampling of the input space. The number of samples generated is increased after every iteration that calls the ground-truth model. This ensures that, as the optimization progresses, the process is capable of finding more finely spaced points. The design space was limited to contain only points with a volume fraction of martensite less than 0.9, and a random forest (RF) classifier was trained to remove test points that did not meet this criterion.

After defining the test points to be used, Algorithm 1 is used to calculate the knowledge gradient for each combination of hyperparameters, test point, and model. The outermost loop (line 1) runs the full set of calculations for each of the three reduced-order models, while the next loop (line 2) runs the calculation for each of the x_{test} values defined at the beginning of the iteration. The final loop (line 5) is used to obtain results using different combinations of hyperparameters. This creates a matrix of results that have the maximum knowledge gradient, reduced-order model index, ***fused_GP*** hyperparameter index, and x_{test} index.

Algorithm 1

Input: $Models, x_{test}$
Output: $\{\max KG(Models, HP, x_{test}), \arg\max KG(Models, HP, x_{test})\}$
1: **for** i=1,2,3 **do**
2: **for** j=1,2,...,length(x_{test}) **do**
3: $y = Models[\,i\,]\_GP(x_{test}[\,j\,])$
4: Update $Models[\,i\,]$ with $(x_{test}[\,j\,], y)$
5: **for** k=1,2,...,*hp_count* **do**
6: Estimate model Correlation (*Reification*)
7: Fuse $Models \rightarrow (x_{fused}, y_{fused})$
8: Build $Fused\_GP(\sigma_n^k, l_1^k, l_2^k, x_{fused}, y_{fused})$
9: Evaluate $KG(Fused\_GP(x_{test}))$
10: **end for**
11: **end for**
12: **end for**

Algorithm 2

1: $def:\ \boldsymbol{x}_{init}$
2: $calc:\ y = Models(\boldsymbol{x}_{init})$
3: $calc:\ y = RVE(\boldsymbol{x}_{init})$
4: **for** i = 1,2,...,n_{iter} **do**
5: $[v^{KG}, \arg(v^{KG}), Model]$:= Algorithm 1
6: $\boldsymbol{x}_{medoids}$ = K-Medoids Clustering $[v^{KG}, \arg(v^{KG}), Model]$
7: **if** Iteration/budget $\geqslant$ Limit **then**
8: $calc:\ y = RVE(\boldsymbol{x}_{medoids})$
9: update RVE GP
10: **else**
11: $calc:\ y = Models(\boldsymbol{x}_{medoids})$
12: update Model GPs
13: **end if**
14: **end for**

Algorithm 2 shows the entire iteration process, and as shown in line 6, the next stage in the process is to cluster the output from Algorithm 1. The clustering is done in three-dimensional space defined by the knowledge gradient value, the model index, and the index of the x_{test} value. This is done to increase the distance between the points to be queried as much as possible and to reduce the likelihood that the process will only select a single model at every iteration. The final stage of the iteration involves calling the models. At this stage, a decision is made on whether to call the RVE model or not. If the conditions for calling the RVE model are not met, the medoids are used to query the reduced-order models. Since each medoid contains a reduced-order model index and an x_{test} index, these are used to query the corresponding model and test point. If the conditions for calling the RVE model have been met, then all x_{test} points contained in the medoids are queried from the RVE model.

Each model has an individual cost (measured in computer clock units) associated with doing a single calculation, and for the current work, the cost of querying a larger batch size is considered to be the cumulative time of completing that number of calculations from that model. In other words, the current work does not consider any discount for using larger batch sizes. In many experimental or computational situations, it would be likely that there would be a discount for using a larger batch size, so this assumption is potentially a conservative one.

In addition to the model calculation costs, there is an iteration cost (again calculated as the computer clock time) associated with calculating and updating the Gaussian processes, as well as calculating the knowledge gradient. This cost and the individual model costs are used to calculate the total cost of the process. In contrast to the multiplication of the cost of model calculations, the time for the calculations is considered to be constant, no matter what the batch size. The justification behind this is that all the calculations are done in parallel. In the event of multiple models being called in a single iteration, the calculation time is considered to be the time cost of the longest-running model.

One of the challenges of using this kind of approach is that there is no single fused model to use for the predictions of the maximum normalized strain-hardening rate. Therefore, the maximum value predicted by the optimization is taken as the maximum normalized strain-hardening rate found from calculations of the RVE model.

Optimization Case Studies

We considered three optimization case studies in the present work. These case studies change the utility function used and the conditions under which the ground truth is called, as well as provide different termination criteria for the optimization process. As already mentioned, in these optimization case studies, the cost is considered to be the computation clock time of the models:

- *No-cost-constraint (NCC) optimization*: The no-cost optimization used the knowledge gradient as the acquisition function with no adjustments and all queries to the RVE model were iteration based. After 25 iterations of updating the low-order models, the next iteration would update the RVE model. This iteration limit was chosen arbitrarily, and should not be taken as an optimum setting. Finally, the optimization process was terminated after 200 iterations.
- *Cost-constrained and iteration-controlled ground-truth query (CC-IC) optimization*: In this optimization scenario, the cost (computational clock time) of the low-order models is considered when calculating the acquisition function. The cost-adjusted acquisition function is defined as

$$v_{\text{cost}}^{\text{KG}} = \frac{v^{\text{KG}}}{\text{modelcost}}, \tag{26}$$

  where v^{KG} is the knowledge gradient value and modelcost is the cost of the model in question. Additionally, a cost-based termination criterion is also included. This will stop the process from continuing once a total budget has been exceeded. This is to emulate a scenario where a project has a budget limit. The costs that contribute to this limit are both the cost of running the models as well as the cost (computational clock time) of calculating the next best points to query. This approach also queries the RVE after 25 iterations of updating the low-order models. Additionally, this approach was also run with iteration limits of 10 and 50 for the calling of the RVE. In total, 15 of the calculations were completed for this comparison.
- *Cost-constrained and cost-controlled ground-truth query (CC-CC) optimization*: The final approach uses the cost-adjusted acquisition function but considers two budget constraints. The first is that, when the cost of iterations exceeds the RVE budget amount, the RVE model will be called. After the RVE model is called, the RVE budget is replenished. The second constraint is that the process is terminated if the total cost exceeds the total budget. In both cases, the model and process costs are considered when calculating the cost of an iteration. Again, all costs are assumed to be the computational clock time.
- *Sequential Bayesian optimization* A conventional sequential Bayesian optimization of the RVE surrogate model only was conducted to enable an assessment of whether the framework in the current work is performing better. This optimization was done by initializing a GP using the same initial RVE data as used for the Batch optimization approach and then using the

knowledge gradient evaluated at 500 samples from the design space to determine the next best point to evaluate. The samples were obtained by Latin hypercube sampling, and the number of samples was incremented by 1 with each iteration. This calculation was completed for all the initial datasets that were used for the batch optimization, and the results were averaged.

RESULTS AND DISCUSSION

The current work aims to maximize the normalized strain-hardening rate of a dual-phase steel. This was done by optimizing the carbon content (wt.%) and intercritical annealing temperature. For the analysis of the results, a normalized strain-hardening rate of 30 or greater is considered an optimum result. The results show the maximum normalized strain-hardening rate found from the RVE model compared against the number of iterations, computational cost, and time for the optimization. The shaded regions of the plots indicate the 95% confidence interval calculated from the results of 20 optimization calculations for each batch size. All the batch optimization results are compared with the sequential Bayesian optimization of the RVE surrogate model only.

The first result shows the maximum normalized strain-hardening rate found from the RVE model against the number of iterations of the optimization routine (Fig. 3). In all three cases, it can be seen that the larger batch sizes (batch sizes 5 and 7) led to faster optimization of the normalized strain-hardening rate. In both cost-constrained cases, these large batch sizes ended quickly since the increased number of calls to the ground-truth function exhausted the available budget quicker. However, when considering the comparison with the sequential optimization of the RVE surrogate model, only the batch size of 7 performed as well as the sequential optimization of the RVE. While these results show a benefit for using larger batch sizes, the number of iterations for the optimization is not necessarily the most useful comparison that can be used.

When the maximum value of the RVE calculations is compared with the total cost of the optimization (Fig. 4), the downside of using the large batch sizes can be observed. In these figures, we can see that, at the beginning of the optimization, the larger batch sizes resulted in a much larger cost much more quickly, and that the lower batch sizes started optimizing at lower costs. However, considering the results after all approaches have called the RVE model at least once, the larger batch sizes were still able to achieve higher normalized strain-hardening rates at a lower computational cost. This is particularly true when comparing the results with the sequential optimization of the RVE only. This shows that, while there may be an advantage to using sequential optimization in terms of the number of iterations required, the cost of the optimization can be decreased by using the framework developed in this work.

The final consideration was how the maximum RVE value found changed with the time taken for the optimization (Fig. 5). Here we can see that the large batch sizes managed to attain higher values significantly faster in real time. This was the case, especially in the cost-constrained approaches. This could be due to the cost-constrained acquisition function favoring the cheaper (faster) models. The comparison with the sequential model shows the real significant advantage of the current approach, as it can obtain an optimum value significantly faster than the sequential optimization.

It is of interest to compare the performance of our proposed BBO approach with that of a conventional BO, carried out by exclusively querying the ground truth, i.e., the RVE-based finite element simulations. This is shown in Fig. 6, which compares a sequential approach without model fusion to our model-fusion-based approach with batch sizes of 1 and 7. As in other cases, we included the uncertainty bounds resulting from running the optimization framework over the design space, multiple times. One noticeable aspect of this figure is the extremely large variance in the performance of the sequential BO approach, even at the latest stages of the optimization approach. This implies that there is considerable risk in employing such an approach as it seems to be highly dependent on the initial conditions, i.e., data, of the optimization process. The figure also shows that our BBO approach, with a batch size of 7, results in a dramatic decrease of an order of magnitude in the time necessary to find the global optimum, with much lower levels of variance. This latter result is significant as BBO appears to be considerably less dependent on initial conditions, providing strong performance guarantees at much faster rates.

As an example of the potentially complex interplay between all the parameters of the framework, we consider the results of using different iteration limits for calling the RVE model in the CC-IC approach. The full results for all batch sizes are contained in the Electronic Supplementary Materials, but presented here is a comparison of the maximum RVE values found at the termination of the optimization procedure (Fig. 7). To obtain these results, the CC-IC approach was calculated for 15 unique initial datasets. This was repeated for each of the three iteration limits. The data in the plot shows the mean and 95% confidence interval calculated for these 15 calculations for each iteration limit.

The first trend that can be noticed is the general increase in the maximum value of the normalized strain-hardening rate correlated to the increase in batch size. This happens with all iteration limits. When we consider the final iteration number in Fig. 7a, we can see that, in general, as the batch size

was increased, the final iteration number decreased. This is a result observed previously and is linked to the increased cost of the larger batch size. What we can also observe by looking at each of the five groups is that the final iteration number is negatively correlated with the iteration limit. This result makes intuitive sense since a lower iteration limit will mean more RVE calculations, which will result in a higher cost. Thus, the budget limit will be reached faster.

However, if we consider the total cost effects in Fig. 7b, the cost of using the lower iteration limit remains fairly consistent. However, the difference between the costs of the optimizations with different iteration limits decreased as the batch size was increased. If we couple this observation with the results in Fig. 7c, we can observe that, while the cost remained fairly constant for the 10-iteration limit case, the time taken for the optimization decreased rapidly as the batch size was increased. The decrease in the time taken for the optimization is not as evident in the 25- and 50-iteration limit cases, but this could be due to the 200-iteration limit placed on the optimization.

All these results indicate that there is a fairly complex correlation between the batch size and the optimum iteration limit for calling the RVE model. However, as has been noted, it is not likely that the optimum combination can be determined when considering more costly functions, but the correlation appears to be that, for larger batch sizes, smaller iteration limits for calling the "ground truth" decrease the time taken for the optimization process. In addition, while the smaller iteration limit does increase the cost compared with a larger iteration limit, this difference decreases with batch size.

The assumption that the cost of BBO scales linearly with the batch size, relative to an exclusively sequential approach, is the most conservative one that can be made. This cost scaling is likely to operate mostly when a design space is being explored using exclusively computer simulations—the cost (in computer time) depends linearly on the number of parallel simulations. However, in real-world experimental approaches, high-throughput, or parallel, processing often has a reduced cost per experiment compared with sequential counterparts. These economies of scale arise from the

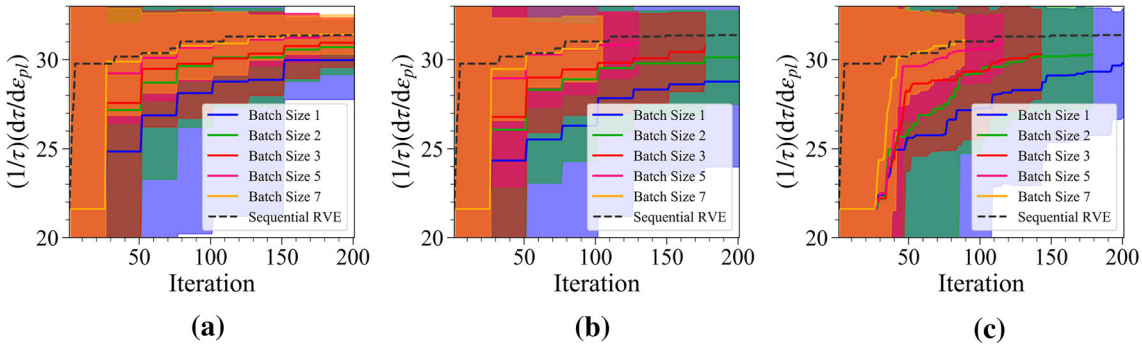

Fig. 3. Maximum normalized strain-hardening rate achieved from RVE calculations as a function of the number of iterations of the optimization process for (a) no cost constraint, (b) cost-constrained, iteration-controlled, and (c) cost-constrained, cost-controlled optimization cases. The shaded regions of the plots indicate the 95% confidence interval calculated from the results of 20 optimization calculations for each batch size, and the mean from the sequential RVE optimization calculations is also shown.

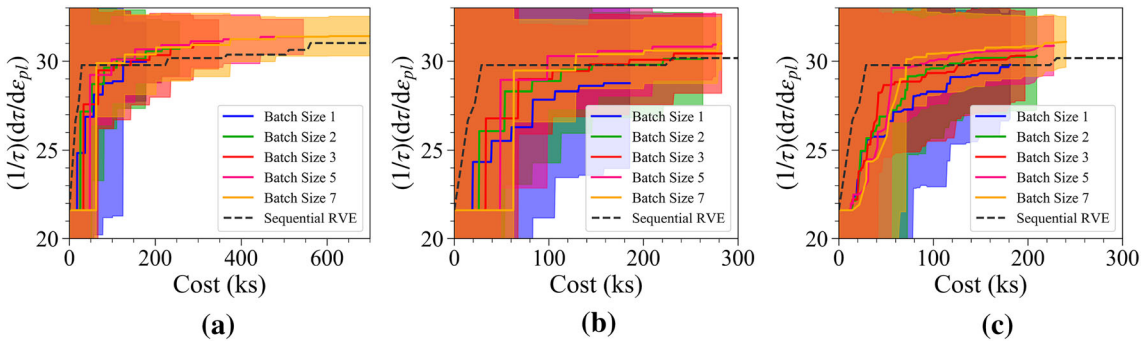

Fig. 4. Maximum normalized strain-hardening rate achieved from RVE calculations as a function of the total cost of the optimization process for (a) no cost constraint, (b) cost-constrained, iteration-controlled, and (c) cost-constrained, cost-controlled optimization cases. The shaded regions of the plots indicate the 95% confidence interval calculated from the results of 20 optimization calculations for each batch size, and the mean from the sequential RVE optimization calculations is also shown.

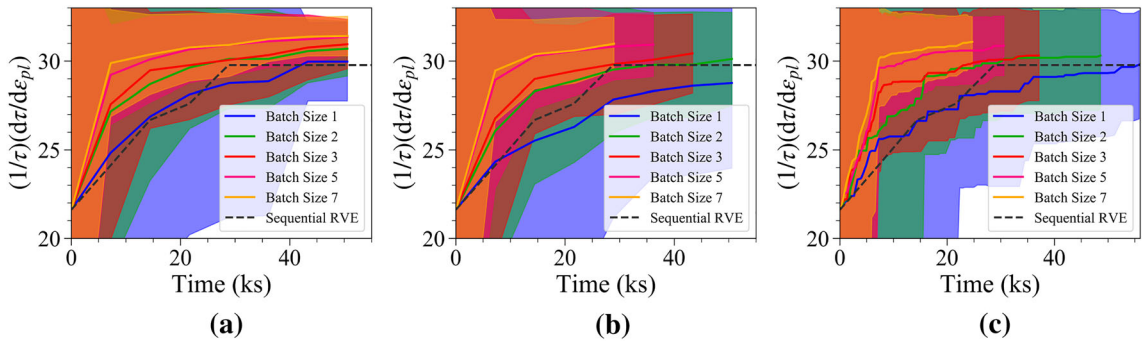

Fig. 5. Maximum normalized strain-hardening rate achieved from RVE calculations as a function of the total time of the optimization process for (a) no cost constraint, (b) cost-constrained, iteration-controlled, and (c) cost-constrained, cost-controlled optimization cases. The shaded regions of the plots indicate the 95% confidence interval calculated from the results of 20 optimization calculations for each batch size, and the mean from the sequential RVE optimization calculations is also shown.

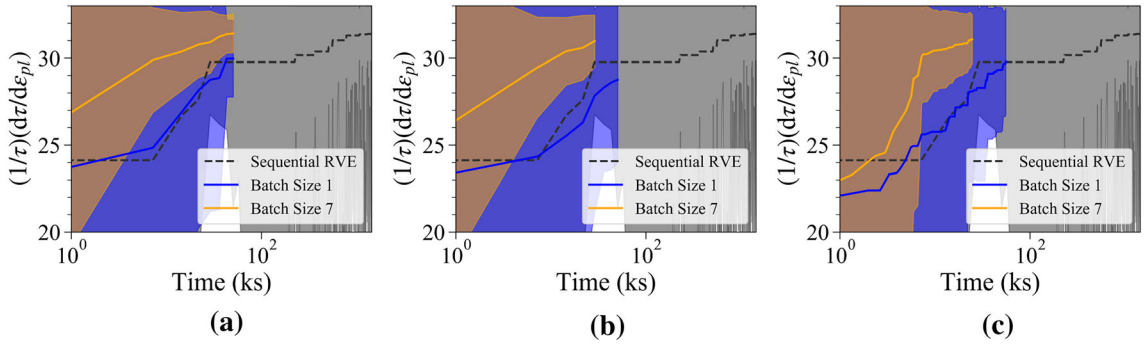

Fig. 6. Maximum normalized strain-hardening rate achieved from RVE-based sequential BO compared with BBO with batch sizes of 1 and 7, as a function of the total time of the optimization process for (a) no cost constraint, (b) cost-constrained, iteration-controlled, and (c) cost-constrained, cost-controlled optimization cases. The shaded regions of the plots indicate the 95% confidence interval calculated from the results of 20 optimization calculations for each batch size.

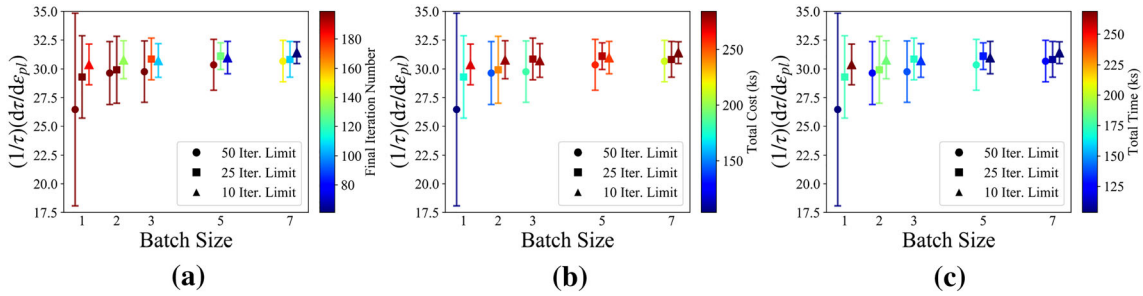

Fig. 7. Comparison of final iteration, total cost, and total time for the optimization approach in the current work when changing the iteration limit for calling the RVE model. The results shown have colors indicating (a) final iteration number, (b) total cost, and (c) total time. The results show the mean result at the end of the optimization using 15 sets of unique initial conditions, and the error bar shows the 95% confidence interval for the final prediction.

simplification of ancillary activities associated with batch experimental operations—there are many activities for which the cost is the same, regardless of the batch size; For example, we can consider the case of combinatorial synthesis through batch arc melting, in which it is necessary to load the feedstock, evacuate the chamber before melting, and wait for the samples to cool down before removal. Each of those steps will be shorter per sample, the more samples are simultaneously melted in a single run, with the ultimate cost per synthesized specimen being lower.

From the results shown above, the fact that the framework is capable of achieving better results at a lower cost under the conservative cost model that assumes no economies of scale indicates that this approach could have significant effects on the process cost for developing new materials or

optimizing existing materials. The benefits of such batch BO frameworks may indeed be much more evident in experimental settings than in computational ones. A benefit that is perhaps more difficult to quantify but that can potentially be even more significant is the beneficial impact on the net present value of any materials development campaign: the faster that one arrives at a potential optimal materials solution, the greater the value that one can extract from such a development effort. To reiterate this point: even if the total cost of a BBO-based materials development campaign were the same as that of a sequential approach, arriving at the optimal solution in much shorter times is extremely beneficial. Coupling a design framework such as that developed in the current work with high-throughput experiments has the possibility of further reducing the time and cost of materials development.

We note that, in this work, we tuned the hyperparameters for the GPs used to emulate the reduced-order models *a priori*. The cost of evaluating these models is orders of magnitude[5] lower than that of evaluating the RVE, and it is thus practical to exhaustively explore the input space of these models before the model fusion BBO is carried out. In cases where the cost of querying even the "cheap" information sources is nonnegligible, it may be necessary to modify our proposed framework. For example, we could implement the BBO routine in two stages: one (perhaps at much larger batch sizes) for fitting the reduced-order models themselves, followed by a second application of BBO for the fused model.

SUMMARY AND CONCLUDING REMARKS

The results from the current work show significant promise in the use of batch Bayesian optimization frameworks within an ICME methodology for materials design. Most notably, the results showed that using larger batch sizes resulted in the quantity of interest being optimized in a shorter time and at a lower cost than when using smaller batch sizes. This confirms an intuitive understanding that, by adding more information on each iteration, we can gain better knowledge of the system under optimization in much shorter times.

Here, we implemented a model-fusion-based BBO approach and applied it successfully to the optimization—through linked computational PSPP relationships—of the "formability" of dual-phase steels by tailoring the chemistry and processing conditions. The results indicate that using batch optimization can greatly decrease the time and cost of the process while simultaneously reducing the uncertainty of the predictions. This has important implications for ICME-enabled design of materials as well as for exclusively experimental materials discovery and design.

We provided further arguments for the benefits of this approach by pointing out that arriving at the answer faster than when using sequential approaches may supersede any consideration of the cost associated with the (computational or experimental) querying of the materials space. The reduction in time necessary to complete the alloy development process would have a very positive impact on the net present value (NPV) of the development effort, minimizing risks while maximizing the potential future benefits of deploying a material in a specific technology.

We note that there is still much work that can be done to improve the framework, particularly to make it more applicable when the hyperparameters of the reduced-order model and the fused model GPs are not known. The authors do acknowledge that the current results might not be generalizable to all applications of the framework. Therefore, work is being conducted to test the framework using standardized test functions. This will allow for full benchmarking of the results from this framework. In addition, while we have demonstrated the effect of changing the iteration limit, there are numerous other framework parameters (for example, the acquisition function, GP hyperparameter ranges, and covariance function) that have not yet been tested to ascertain their effect on the optimization process.

In fact, while in this work we have carried BBO over the hyperparameter space with a fixed covariance structure (i.e., Matérn kernel) and acquisition function (i.e., knowledge gradient), it may be possible to extend this approach over the model as well as the acquisition function space. This would follow the spirit of the current BBO approach, following the premise that, at the beginning of an optimization, it is not certain what type of covariance structure or even what type of policy is most effective for a given problem space.

ACKNOWLEDGEMENTS

The authors acknowledge Grant No. NSF-CMMI-1663130, DEMS: Multi-Information Source Value of Information Based Design of Multiphase Structural Materials and well as Grant No. NSF-S&AS-1849085. R.A. would also like to acknowledge Grant No. NSF-1835690. D.A. would also like to acknowledge Grant No. FA9550-15-1-0038. D.A. and R.A. acknowledge Grant No. NSF-DGE-1545403. Portions of this research were conducted with the advanced computing resources provided by Texas A&M High-Performance Research Computing Facility.

CONFLICT OF INTEREST

The authors declare that they have no conflicts of interest.

ELECTRONIC SUPPLEMENTARY MATERIAL

The online version of this article (https://doi.org/10.1007/s11837-020-04396-x) contains supplementary material, which is available to authorized users.

REFERENCES

1. J. Allison, D. Backman, L. Christodoulou, *JOM* 58(11), 25 (2006).
2. A. Talapatra, S. Boluki, P. Honarmandi, A. Solomou, G. Zhao, S.F. Ghoreishi, A. Molkeri, D. Allaire, A. Srivastava, X. Qian, et al., *Front. Mater.* 6, 82 (2019).
3. R. Arróyave, D.L. McDowell, Systems approaches to materials design: past, present, and future, *Annu. Rev. Mater. Res.* 49, 103 (2019).
4. S.F. Ghoreishi, A. Molkeri, A. Srivastava, R. Arroyave, D. Allaire, *J. Mech. Des.* 140, 11 (2018).
5. S.F. Ghoreishi, A. Molkeri, R. Arróyave, D. Allaire, A. Srivastava, *Acta Mater.* 180, 260 (2019).
6. J.J. de Pablo, B. Jones, C.L. Kovacs, V. Ozolins, A.P. Ramirez, *Curr. Opin. Solid State Mater. Sci.* 18, 99 (2014).
7. N.M. Al Hasan, H. Hou, S. Sarkar, S. Thienhaus, A. Mehta, A. Ludwig, I. Takeuchi, *Engineering* (2020).
8. M.A. Melia, S.R. Whetten, R. Puckett, M. Jones, M.J. Heiden, N. Argibay, A.B. Kustas, *Appl. Mater. Today* 19, 100560 (2020).
9. Y. Lyu, Y. Liu, T. Cheng, B. Guo, *J. Materiomics* 3, 221 (2017).
10. P. Liu, B. Guo, T. An, H. Fang, G. Zhu, C. Jiang, X. Jiang, *J. Materiomics* 3, 202 (2017).
11. T. Wang, Y. Xiong, Y. Wang, P. Qiu, Q. Song, K. Zhao, J. Yang, J. Xiao, X. Shi, L. Chen, *Mater. Today Phys.* 12, 100180 (2020).
12. X. Zhang, Y. Xiang, *J. Materiomics* 3, 209 (2017).
13. A. Aspuru-Guzik, K. Persson, *Materials Acceleration Platform: Accelerating Advanced Energy Materials Discovery by Integrating High-throughput Methods and Artificial Intelligence, Mission Innovation* (2018).
14. A. Talapatra, S. Boluki, T. Duong, X. Qian, E. Dougherty, R. Arróyave, Autonomous efficient experiment design for materials discovery with bayesian model averaging, *Phys. Rev. Mater.* 2, 113803 (2018).
15. D. Ginsbourger, R. Le Riche, L. Carraro, in *Computational Intelligence in Expensive Optimization Problems*, ed. by Y. Tenne, C.K. Goh (Springer Berlin Heidelberg, Berlin, Heidelberg, 2010), pp. 131–162.
16. J. Azimi, A. Jalali, X. Fern, *Hybrid Batch Bayesian Optimization* (2012).
17. E. Contal, D. Buffoni, A. Robicquet, N. Vayatis, in *Machine Learning and Knowledge Discovery in Databases*, ed. by H. Blockeel, K. Kersting, S. Nijssen, F. Železný (Springer Berlin Heidelberg, Berlin, Heidelberg, 2013), pp. 225–240.
18. T. Desautels, A. Krause, J.W. Burdick, *J. Mach. Learn. Res.* 15(119), 4053 (2014).
19. J. Gonzalez, Z. Dai, P. Hennig, N. Lawrence, in *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, *Proceedings of Machine Learning Research*, vol. 51, ed. by A. Gretton, C.C. Robert (PMLR, Cadiz, Spain, 2016), *Proceedings of Machine Learning Research*, vol. 51, pp. 648–657.
20. T.T. Joy, S. Rana, S. Gupta, S. Venkatesh, *Knowl.-Based Syst.* 187, 104818 (2020).
21. P.I. Frazier, A tutorial on bayesian optimization, arXiv preprint arXiv:1807.02811 (2018).
22. C.E. Rasmussen, C.K. Williams, *Gaussian Processes for Machine Learning* (The MIT Press, Oxford, 2006).
23. M. Stein, *Interpolation of Spatial Data* (Springer-Verlag, New York, 1999).
24. W.D. Thomison, D.L. Allaire, in *19th AIAA Non-Deterministic Approaches Conference*, AIAA SciTech Forum (American Institute of Aeronautics and Astronautics, 2017).
25. R.L. Winkler, *Manage. Sci.* 27, 479 (1981). INFORMS.
26. H.J. Kushner, *J. Basic Eng.* 86, 97 (1964).
27. J. Močkus, in *Optimization Techniques IFIP Technical Conference Novosibirsk, July 1-7, 1974*, ed. by G.I. Marchuk (Springer Berlin Heidelberg, Berlin, Heidelberg, 1975), pp. 400–404.
28. D.R. Jones, M. Schonlau, W.J. Welch, *J. Global Optim.* 13, 455 (1998).
29. N. Srinivas, A. Krause, S. Kakade, M. Seeger, in *Proceedings of the 27th ICML* (Omnipress, Madison, WI, USA, 2010), ICML'10, pp. 1015–1022. Haifa, Israel.
30. P.I. Frazier, W.B. Powell, S. Dayanik, *SIAM J. Control Optim.* 47(5), 2410 (2008).
31. W.B. Powell, I.O. Ryzhov, *Optimal Learning*, vol. 841 (Wiley, New York, 2012).
32. W. Voigt, *Ann. Phys. Chem.* 274, 573 (1889).
33. A. Reuss, *J. Appl. Math. Mech.* 9, 49 (1929).
34. O. Bouaziz, P. Buessler, *Rev. Metall.* 99, 71 (2002).
35. P. Ludwik, *Elemente der technologischen Mechanik.* 57 p. 20 illus., III fold. diagr. (J. Springer, Berlin, 1909).