IISE TRANSACTIONS
2021, VOL. 53, NO. 3, 352–364
https://doi.org/10.1080/24725854.2020.1774688

A Bayesian framework for functional calibration of expensive computational models through non-isometric matching

Babak Farmanesh, Arash Pourhabib, Balabhaskar Balasundaram, and Austin Buchanan

School of Industrial Engineering and Management, Oklahoma State University, Stillwater, OK, USA

ABSTRACT

We study statistical calibration, i.e., adjusting features of a computational model that are not observable or controllable in its associated physical system. We focus on functional calibration, which arises in many manufacturing processes where the unobservable features, called calibration variables, are a function of the input variables. A major challenge in many applications is that computational models are expensive and can only be evaluated a limited number of times. Furthermore, without making strong assumptions, the calibration variables are not identifiable. We propose Bayesian Non-isometric Matching Calibration (BNMC) that allows calibration of expensive computational models with only a limited number of samples taken from a computational model and its associated physical system. BNMC replaces the computational model with a dynamic Gaussian process whose parameters are trained in the calibration procedure. To resolve the identifiability issue, we present the calibration problem from a geometric perspective of non-isometric curve to surface matching, which enables us to take advantage of combinatorial optimization techniques to extract necessary information for constructing prior distributions. Our numerical experiments demonstrate that in terms of prediction accuracy BNMC outperforms, or is comparable to, other existing calibration frameworks.

ARTICLE HISTORY
Received 21 September 2018
Accepted 15 May 2020

KEYWORDS
Functional calibration;
Gaussian process;
Generalized minimum spanning tree

1. Introduction

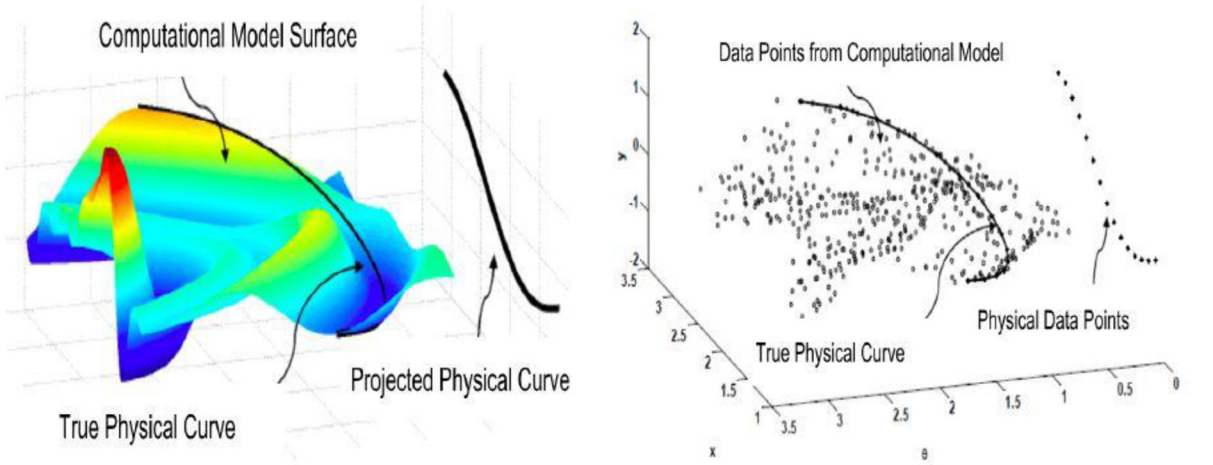
Experimenting on computational models to understand physical systems has been a popular practice ever since computers became advanced enough to handle complex mathematical models and intense computational procedures (Fang *et al.*, 2005; Santner *et al.*, 2013). This popularity is mainly due to a computational model being able to obtain outputs of an experiment in a relatively more cost-effective and timely manner compared with conducting actual experiments in a laboratory. However, one challenge in utilizing computational models is their "adjustment." In fact, computational models usually incorporate features that cannot be observed or measured in physical systems, but must be correctly specified so that the computational model can accurately represent the physical system (Kennedy and O'Hagan, 2001). We refer to these unobservable/unmeasurable features as *calibration variables*, and to the adjustment of their values as the *calibration procedure*. We call input features that are common between the computational models and the physical systems as *control variables*.

For example, in the fabrication of Poly-Vinyl Alcohol (PVA)-treated buckypaper, we are interested in understanding the relationship between the response value, which is the tensile strength, and the control variable, which is the amount of PVA (Pourhabib *et al.*, 2015). Here, the calibration variable is the percentage of PVA absorbed, which cannot be measured in the physical system, but is required in the computational model.

Past studies on the calibration problem generally assumed unique values for calibration variables, an approach referred to as *global calibration*, and used different statistical approaches to estimate these values. For instance, Craig *et al.* (2001), Kennedy and O'Hagan (2001); Higdon *et al.* (2004) Reese *et al.* (2004), Han *et al.* (2009) and Williams *et al.* (2006); Bayarri *et al.* (2007), Higdon *et al.* (2008), and Goldstein and Rougier (2009) devised various Bayesian models, whereas Loeppky *et al.* (2006) and Pratola *et al.* (2013) used maximum likelihood estimation, and Han *et al.* (2009) and Joseph and Melkote (2009) developed mixed models by combining frequentist and Bayesian methodologies. More recently Tuo and Wu (2015, 2016) developed models based on L_2 distance projection to estimate the true values of the global calibration variables.

Currently, few studies employ *functional calibration* by assuming that the values of the calibration variables *depend on the control variables*. Pourhabib *et al.* (2015) showed that, for the buckypaper fabrication problem, an approach that considers a parametric functional relationship between the amount of PVA and the percentage absorbed can outperform the global calibration approach of Kennedy and O'Hagan (2001). Similarly, Xiong *et al.* (2009) used a simple linear relationship to improve the calibration accuracy in a

CONTACT Arash Pourhabib arash.pourhabib@okstate.edu
Supplemental data for this article is available online at https://doi.org/10.1080/24725854.2020.1774688.



(a) Complete curve and surface (b) Observed curve and surface by incomplete data

Figure 1. A non-isometric curve to surface matching perspective of functional calibration: The left plot shows the complete surface and curve. In practice, we observe a scatter of data points sampled from the complete curve and surface, which is depicted in the right plot.

benchmark thermal challenge problem. Furthermore, non-parametric methods can also be utilized to model functional relationships between the calibration variables and the control variables. Such non-parametric functional relationships have been constructed using Reproducing Kernel Hilbert Spaces (Schölkopf *et al.*, 2001) and Gaussian processes (Rasmussen, 2004) by various authors Plumlee *et al.*, 2016; Brown and Atamturktur, 2018; Pourhabib *et al.*, 2018).

All the aforementioned studies in functional calibration and most studies in global calibration require computational models that are "cheaply executable." This assumption is required since computational models need to be evaluated thousands of times, either to draw samples from a posterior distribution in Bayesian approaches, or to numerically minimize a loss function in other approaches. If the computational model is "expensive," one can obtain a small number of observations from the computational model, then fit a surrogate function based on these random samples, and in the final step, replace the computational model in the calibration procedure with this new surrogate model. However, as discussed in Section 6, this poses a challenge, as "static" replacement may result in poor retrieval of the calibration variables.

Another challenge is the identifiability issue: it is difficult to solve the calibration problem in higher-dimensional spaces without making additional assumptions about the solution space (Pourhabib *et al.*, 2018). Furthermore, a good prediction performance for the response values does not necessarily imply that a method has accurately captured the functional relationship between the calibration and control variables (Tuo and Wu, 2015; Ezzat *et al.*, 2018; Plumlee and Joseph, 2018). This is a significant drawback since, in many applications, understanding the functional relationship between the calibration and control variables is as important as predicting the response values of the system under study.

In this article, we develop a new framework for the functional calibration of expensive computational models. Unlike conventional surrogate modeling, which replaces the computational model with a static, approximated surface, we employ a "dynamic" Gaussian Process (GP) over the computational model. Our GP is dynamic in the sense that the hyper-parameters of the GP's covariance function are trained during the calibration procedure. We simultaneously construct posterior distributions for the hyper-parameters of the GP's covariance function and the calibration variables associated with each of the physical control vectors. In other words, we allow the GP to tune its hyper-parameters in addition to the calibration variables such that the computational model responses become as close as possible to the physical responses.

To tackle the unidentifiability issue in higher-dimensional spaces, we use informative prior distributions. We take advantage of an alternative geometric interpretation of calibration, namely the non-isometric matching of a curve to a surface. We explain this in the case of a single control variable and a single calibration variable. From a geometric perspective, all possible values for the control variable and the physical response constitute a plane curve in the control–response space (see Figure 1(a)). By contrast, in the computational model, we can specify the values of both the control and the calibration variables. Consequently, all possible values of the control and calibration variables, and the responses of the computational model together form a surface. The plane physical curve we observe in the control–response space is a projection of a space curve in the three-dimensional control–calibration–response space. Due to the nature of projection into a lower–dimensional space, the length of the projected curve is not necessarily the same as the original curve in the three-dimensional space. The projection is therefore *non-isometric* (Bronstein *et al.*, 2003, 2005).

The geometric interpretation is due to the nature of the calibration variable in a physical process: For each value of the control variable there exists a (possibly unknown) value for the calibration variable, and these two features determine a single response. Since we do not observe the actual value of the calibration variable in the physical process, we only see a projected curve in the control–response space. Hence, calibration aims to recover the true physical curve, or in

other words, determine a non-isometric match of a curve to a surface.

The remainder of this article is organized as follows. We explain our Bayesian model for handling expensive computational models in Section 2. Section 3 contains a formal description of the calibration problem and its interpretation as a non-isometric curve to surface matching problem. Our graph-theoretic approach to utilizing this geometric perspective to construct informative prior distributions for our Bayesian model is also presented in Section 3. In Section 4, we generalize the idea of a non-isometric curve to surface matching to higher dimensions and introduce integer programming techniques to tackle the problem. The approach presented in Sections 3 and 4 to construct informative prior distributions for our Bayesian model is used to calculate posterior distributions in Section 5. Our experimental results are reported in Section 6, comparing them with previous approaches. Section 7 concludes the article and presents paths for future research.

2. General setting: A Bayesian model for calibration

Consider a physical system that operates according to a set of (possibly unknown) physical laws. In this system, there is a functional relationship between a group of features and the response (output). We call those features of the system that can be measured and specified as inputs of the physical system as *control variables*, and denote the vector of these variables by $\mathbf{x} \in \mathbb{R}^{d^x}$. We assume that we obtain data for the physical system by conducting *physical experiments*: once the control variables are set (either observed or specified) in the physical system, the physical process $\mathcal{F}^p$ generates a real-valued response y^p, that is $y^p = \mathcal{F}^p(\mathbf{x})$.

Although the response is a function of *all* features of the physical system, we write y^p explicitly as a function of $\mathbf{x}$ as the rest of the features are hard to measure or control, and hence we have no control over them in the physical system. We call such features *calibration variables* and categorize them into the following two groups: (i) *global calibration variables*, which have unique values regardless of the values of the control variables, and (ii) *functional calibration variables*, which are functions of control variables.

We denote the vector of global calibration variables by $\boldsymbol{\psi} \in \mathbb{R}^{d^\psi}$ and the vector of functional calibration variables by $\boldsymbol{\theta} \in \mathbb{R}^{d^\theta}$. We also denote the function that maps $\mathbf{x}$ to the kth element of $\boldsymbol{\theta}$, i.e., θ_k, by $\mathcal{F}^\theta_k$ and the vector of all these functions by $\mathcal{F}^\theta = [\mathcal{F}^\theta_1, ..., \mathcal{F}^\theta_{d^\theta}]^\top$. With a slight abuse of notation, we denote the vector map from $\mathbf{x}$ to $\boldsymbol{\theta}$ using the vector of functions $\mathcal{F}^\theta$ as $\boldsymbol{\theta} = \mathcal{F}^\theta(\mathbf{x})$.

Suppose we have a computational model constructed according to the laws governing the physical system. Similar to the physical system, the response of our computational model is determined by the interactions between the control and the calibration variables. However, in a computational model we can set the values of all $\mathbf{x}$, $\boldsymbol{\theta}$, and $\boldsymbol{\psi}$ arbitrarily within their respective domains. That is because, unlike physical experiments, there are no constraints on measuring or specifying control or calibration variables in a computational model. If we denote the computational process as $\mathcal{F}^s$, then the response of the computational model can be written as

$$y^s = \mathcal{F}^s(\mathbf{x}, \boldsymbol{\psi}, \boldsymbol{\theta}). \tag{1}$$

We refer to obtaining a value for y^s, given a combination of $\mathbf{x}$, $\boldsymbol{\psi}$, and $\boldsymbol{\theta}$ in the computational model, as a *computer experiment*.

The goal of *calibration* is to adjust the variables $\boldsymbol{\psi}$ and $\boldsymbol{\theta}$ such that the computational model represents the physical system in the sense that the computational model can predict the physical response at any input location $\mathbf{x}^*$.

Mathematically, calibration can be viewed as the estimation of vectors $\mathcal{F}^\theta$ and $\boldsymbol{\psi}$ such that, for any given $\mathbf{x}^*$, the function $\mathcal{F}^s : \mathbb{R}^{d^x} \times \mathbb{R}^{d^\psi} \times \mathbb{R}^{d^\theta} \to \mathbb{R}$ generates a response close to y^{p^*} up to an error ϵ^*, i.e.,

$$y^{p^*} = \mathcal{F}^s(\mathbf{x}^*, \boldsymbol{\psi}, \mathcal{F}^\theta(\mathbf{x}^*)) + \epsilon^*, \tag{2}$$

where the error ϵ^* captures the measurement error and the discrepancy between the physical and computation model.

To estimate $\boldsymbol{\psi}$ and $\mathcal{F}^\theta$ in (2) we initially obtain m responses from $\mathcal{F}^p$ at a set of physical system inputs $\{\mathbf{x}^p_1, ..., \mathbf{x}^p_m\}$ to create a dataset P corresponding to that physical system:

$$P := \left\{ p_i = \left(\mathbf{x}^p_i, y^p_i\right) | \mathbf{x}^p_i \in \mathbb{R}^{d^x}, y^p_i \in \mathbb{R}, i \in \{1, 2, ..., m\} \right\}.$$

We also create the counterpart of P in the computational model, i.e., the computational dataset as

$$S := \left\{ s_j = \left(\mathbf{x}^s_j, \boldsymbol{\psi}^s_j, \boldsymbol{\theta}^s_j, y^s_j\right) | \mathbf{x}^s_j \in \mathbb{R}^{d^x},\ \boldsymbol{\psi}^s_j \in \mathbb{R}^{d^\psi},\ \boldsymbol{\theta}^s_j \in \mathbb{R}^{d^\theta},\ j \in \{1, 2, ..., n\} \right\},$$

based on a set of computational model inputs $\{(\mathbf{x}^s_1, \boldsymbol{\psi}^s_1, \boldsymbol{\theta}^s_1), ..., (\mathbf{x}^s_n, \boldsymbol{\psi}^s_n, \boldsymbol{\theta}^s_n)\}$. We assume the sets of physical system inputs and the computational model inputs are given. For a discussion of how to select the inputs we refer the reader to the paper by Ezzat *et al.* (2018).

Let $\boldsymbol{\theta}^p_i = \mathcal{F}^\theta(\mathbf{x}^p_i)$ and $\boldsymbol{\psi}^p$ denote the true values of the calibration variables and assume that the errors are independent and have identical normal distribution with zero mean and constant variance. Therefore, we obtain the calibration model as

$$\begin{aligned} y^p_i &= \mathcal{F}^s(\mathbf{x}^p_i, \boldsymbol{\psi}^p, \boldsymbol{\theta}^p_i) + \epsilon^p_i, \ \text{ where } \epsilon^p_i \sim \mathcal{N}(0, \sigma^2), \\ &\forall i \in \{1, 2, ..., m\}. \end{aligned} \tag{3}$$

Remark 1. If we remove the functional calibration variable $\boldsymbol{\theta}^p_i$ from Equation (3), we get a simplified version of the global calibration model proposed in Kennedy and O'Hagan (2001). In fact, Kennedy and O'Hagan (2001) assume $y^p_i = \mathcal{F}^s(\mathbf{x}^p_i, \boldsymbol{\psi}^p) + \delta(\mathbf{x}^p_i) + \epsilon^p_i$, where $\delta(\cdot)$ is a GP independent from $\mathcal{F}^s$, which characterizes all the discrepancy between the computational model and the physical system due to assumptions made in building the computational model. However, as in this study we focus on computational models with a limited number of data points, we do not include a separate discrepancy term $\delta(\cdot)$, which entails introducing a set of additional parameters and would make the estimation procedure unstable. Therefore, as we utilize a dynamic GP

to minimize the overall discrepancy, we choose to use ϵ_i^p to represent not only the measurement error in the physical system but also the discrepancy between the computational model and the physical system. As such, our assumptions are similar to those made by Brown and Atamturktur (2018). In Appendix E in the supplementary material we validate these assumptions on the datasets used in this study. The reader can refer to the discussion by Tuo and Wu (2016) for a frequentist interpretation of the model proposed by Kennedy and O'Hagan (2001).

Note that applying Bayesian statistics to construct posterior distributions for parameters of model (3), i.e., $\boldsymbol{\psi}^p, \sigma^2$ and $\boldsymbol{\theta}_i^p$, requires a large number of evaluations of $\mathcal{F}^s$, which is not practical for expensive computational models. Therefore, we assume $\mathcal{F}^s$ is a GP with a constant mean (Rasmussen, 2004), i.e., $\mathcal{F}^s \sim \mathcal{GP}(0, \mathcal{K}(\cdot,\cdot))$, where $\mathcal{K}(\cdot,\cdot)$ is a covariance function. We further assume that the overall average of the responses from the computational model has been subtracted from each output, and as such we use a GP with mean zero. Here we use the squared exponential kernel function as the choice of the covariance function:

$$\mathcal{K}(\mathbf{z}, \mathbf{z}') = \gamma \exp\left(-(\mathbf{z}-\mathbf{z}')^\top \mathbf{L}(\mathbf{z}-\mathbf{z}')\right), \tag{4}$$

where γ is the magnitude parameter and $\mathbf{L}$ is a diagonal matrix of the length-scale parameters. We denote the vector of the diagonal elements of $\mathbf{L}$ by $\boldsymbol{\ell}$.

Subsequently, we can obtain the likelihood of model (3) by the GP distribution defined on $\mathcal{F}^s$ as a multivariate normal distribution:

$$\mathbf{y}^p | \mathbf{X}^p, \boldsymbol{\Theta}^p, \boldsymbol{\psi}^p, \boldsymbol{\ell}, \gamma, \sigma^2 \sim \mathcal{N}(0, \boldsymbol{\Sigma} + \sigma^2 \mathbf{I}_m), \tag{5}$$

where $\mathbf{y}^p = [y_1^p, \ldots, y_m^p]^\top$ is the vector of physical responses, $\mathbf{X}^p = [\mathbf{x}_1^p, \ldots, \mathbf{x}_m^p]^\top$ and $\boldsymbol{\Theta}^p = [\boldsymbol{\theta}_1^p, \ldots, \boldsymbol{\theta}_m^p]^\top$ are matrices of size $m \times d^x$ and $m \times d^\theta$ respectively, and $\boldsymbol{\Sigma}$ is the $m \times m$ covariance matrix whose elements are calculated by covariance function (4) with $[\mathbf{x}_i^{p^\top}, \boldsymbol{\theta}_i^{p^\top}, \boldsymbol{\psi}^{p^\top}]^\top$ as input vectors with length $(d^x + d^\theta + d^\psi)$.

Remark 2. Although in the process of deriving likelihood (5), we consider $\boldsymbol{\psi}^p$ and the columns of $\boldsymbol{\Theta}^p$ as the input variables of model (3), we do not know the values of these input variables, and we intend to estimate them. Therefore, in order to distinguish the calibration variables $\boldsymbol{\psi}$ and $\boldsymbol{\theta}$, in (1) from the parameters in model (3), we refer to $\boldsymbol{\Theta}^p$ and $\boldsymbol{\psi}^p$ as *calibration parameters*.

We can estimate the calibration parameters of model (3), the parameters of covariance function (4), and the variance of error, using Bayesian statistics with the posterior distribution:

$$\begin{aligned}\pi(\boldsymbol{\Theta}^p, \boldsymbol{\psi}^p, \boldsymbol{\ell}, \gamma, \sigma^2 | \mathbf{y}^p, \mathbf{X}^p) &\propto \pi(\mathbf{y}^p | \mathbf{X}^p, \boldsymbol{\Theta}^p, \boldsymbol{\psi}^p, \boldsymbol{\ell}, \gamma, \sigma^2) \\ &\times \pi(\boldsymbol{\Theta}^p)\pi(\boldsymbol{\psi}^p)\pi(\boldsymbol{\ell})\pi(\gamma)\pi(\sigma^2).\end{aligned} \tag{6}$$

The Bayesian model (6) would be completed by specifying prior distributions for the parameters $\pi(\boldsymbol{\Theta}^p), \pi(\boldsymbol{\psi}^p), \pi(\boldsymbol{\ell}), \pi(\gamma)$, and $\pi(\sigma^2)$. However, our model suffers from unidentifiablity in the absence of informative priors, due to the high-dimensionality of the parameter space. Therefore, in Section 5 we present graph-theoretic approaches that help construct informative priors for the calibration parameters $\boldsymbol{\Theta}^p$ and $\boldsymbol{\psi}^p$.

Note that the replacement of $\mathcal{F}^s$ by $\mathcal{GP}(0, \mathcal{K}(\cdot,\cdot))$ does not constitute a surrogate modeling approach, wherein the computational model is replaced by a *fixed* surrogate surface, which is in turn trained based on a set of limited samples drawn from the computational model prior to any calibration procedure. Our approach is fundamentally different from surrogate modeling, since building and training the GP is a part of the calibration process.

3. Calibration as non-isometric matching: A special case

We explain in this section how the calibration problem can be viewed as a non-isometric curve to surface matching problem for the special case where $\mathbf{x} \in \mathbb{R}, \boldsymbol{\theta} \in \mathbb{R}$, and $\boldsymbol{\psi} \in \emptyset$, which means both control and calibration variables are one-dimensional and no global calibration variable exists. From a geometric perspective, all the possible values for $\mathbf{x}$ and $\mathcal{F}^p(\mathbf{x})$ constitute the curve $(\mathbf{x}, \mathcal{F}^p(\mathbf{x}))$ in a two-dimensional space. In the computational model, however, we can specify the values of both $\mathbf{x}$ and $\boldsymbol{\theta}$. Consequently, all the possible values of $\mathbf{x}$, $\boldsymbol{\theta}$, and $\mathcal{F}^s(\mathbf{x}, \boldsymbol{\theta})$ together form a surface $(\mathbf{x}, \boldsymbol{\theta}, \mathcal{F}^s(\mathbf{x}, \boldsymbol{\theta}))$ in a three-dimensional space. As we noted in Section 1, the true physical curve lies on the three-dimensional computational model surface, i.e., $(\mathbf{x}, \mathcal{F}^\theta(\mathbf{x}), \mathcal{F}^p(\mathbf{x}))$. However, since we do not observe the actual values of the calibration variables in the physical process, we only see a projected curve in $\mathbf{x} - y$ space (see Figure 1(a)). Hence, the calibration problem is to recover the true physical curve, or, in other words, determine a non-isometric match of a curve to a surface.

As mentioned earlier, the non-isometry is due to the fact that the curve $(\mathbf{x}, \mathcal{F}^\theta(\mathbf{x}), \mathcal{F}^p(\mathbf{x}))$ on the three-dimensional $\mathbf{x} - \boldsymbol{\theta} - y$ space has a different length than the projected curve $(\mathbf{x}, 0, \mathcal{F}^p(\mathbf{x}))$ on a two-dimensional $\mathbf{x} - y$ space. Therefore, this is, in principle, different from isometric matching problems (Bronstein *et al.*, 2005; Gruen and Aken, 2005; Baltsavias *et al.*, 2008).

In practice we only have the finite physical system dataset P along with a finite computational model dataset S, as we do not observe a complete curve or surface. Ideally, the points in P lie on the projected curve that we observe, and the points in S lie on the computational model surface (see Figure 1(b)). Hence, what we observe is incomplete data, and we aim to match non-isometrically an incomplete curve to an incomplete surface, which is equivalent to solving the calibration problem.

This geometric perspective motivates us to view the problem through a combinatorial lens and model the problem using graph-theoretic approaches. Our graph-based solution to the non-isometric curve to surface matching problem provides us with a set of computational model data points, which carry information about the calibration parameters. We call this set of computational data points *anchor points*. These anchor points will then be used in Section 5 to construct prior distributions for our Bayesian model.

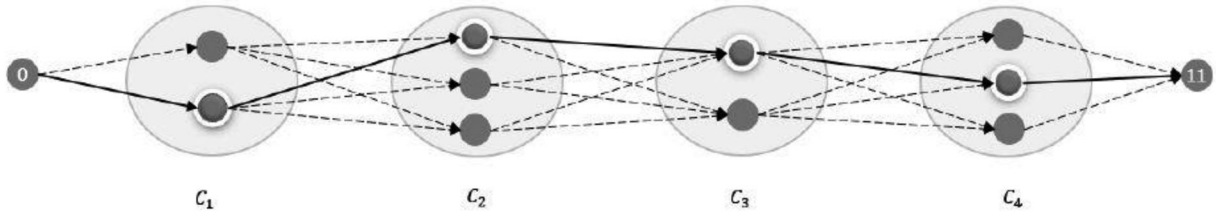


Figure 2. Illustration of the calibration digraph for the case where $m = 4$ and $n = 10$. The vertices represent data points from the computational model and the clusters C_1 through C_m correspond to physical system data points. Vertices denoted by dark circles with a white border represent the anchor points, and the solid arrows identify the edges in the shortest path found.

We seek to identify a set of anchor points among the computational data points that are "close" to the points on the true physical curve. In other words, the anchor points are positioned such that the true physical curve passes through the neighborhoods of those points. We want the anchor points to satisfy two desirable properties: (i) the computational model response should be close to the physical response for a given input $\mathbf{x}$; and (ii) the calibration parameter values for two consecutive anchor points should be close to each other. The former drives our method to identify the anchor points that have similar responses to that of the physical system, and the latter aims to encourage the smoothness of the physical curve.

Note that we are only interested in identifying these "optimal" anchor points that provide us with information about the true physical curve to be used in our prior distributions, and not the true physical curve itself. However, one could also directly use the selected anchor points to approximate the true physical curve via interpolation. Given our focus on expensive computational models wherein the number of computational model data points is limited, such an approximation of the true physical curve may not be accurate. In the next section, we formally define and address the problem of finding anchor points with the desired properties using a graph-theoretic approach for the special case when $\mathbf{x} \in \mathbb{R}, \boldsymbol{\theta} \in \mathbb{R},$ and $\boldsymbol{\psi} \in \emptyset$.

3.1. A graph-theoretic approach for finding anchor points

Without loss of generality, we assume that all the data points in the physical system and the computational model datasets are strictly ordered such that $\mathbf{x}_i^p < \mathbf{x}_{i+1}^p$, for all $i \in \{1, 2, \ldots, m-1\}$, and $\mathbf{x}_j^s < \mathbf{x}_{j+1}^s$, for all $j \in \{1, 2, \ldots, n-1\}$. We construct an edge-weighted directed graph $G = (V, E)$ with vertex set $V := \{0, 1, 2, \ldots, n+1\}$, and the edge set E described in Equation (8) below. The vertices in $V^0 := \{1, 2, \ldots, n\}$ correspond to the computational model data points in S. We refer to G as the *calibration digraph*.

Recall that, intuitively, the objective of calibration is to minimize the difference between the outputs of the physical system and the corresponding computational model. As such, the first step is to find control variables that are similar in both settings, the physical experiments and the computer experiments. Therefore, we first group the control variables in the computational model based on their distance to the control variables in the physical experiments. We partition V^0 into m clusters $C_1, \ldots, C_m$ as follows: any vertex $j \in V^0$ corresponding to data point $s_j \in S$ is assigned to a unique cluster C_i by the following formula:

$$j \in C_i \iff i = \min\left\{ \arg\min_{\ell \in \{1,2,\ldots,m\}} \{||\mathbf{x}_\ell^p - \mathbf{x}_j^s||_2\} \right\}. \tag{7}$$

If the inner minimum in (7) is not unique, then the outer minimum is used to break the tie by choosing the smallest index. As a consequence, each cluster C_i is in one-to-one correspondence with the ith physical data point. This choice of tie-breaker is easy to implement and establishes a mechanism for consistent assignment of points to a cluster. We can now describe the set of directed edges E as

$$\begin{aligned} E := &\bigcup_{i=1}^{m-1} \{(u, v) | u \in C_i, v \in C_{i+1}\} \cup \{(0, u) | u \in C_1\} \\ &\times \cup \{(u, n+1) | u \in C_m\}. \end{aligned} \tag{8}$$

This construction is illustrated in Figure 2.

The final critical step is to assign a weight w_{uv} to each edge $(u, v) \in E$. Consider two consecutive clusters C_i and C_{i+1} and vertices $u \in C_i$ and $v \in C_{i+1}$. Define w_{uv} as

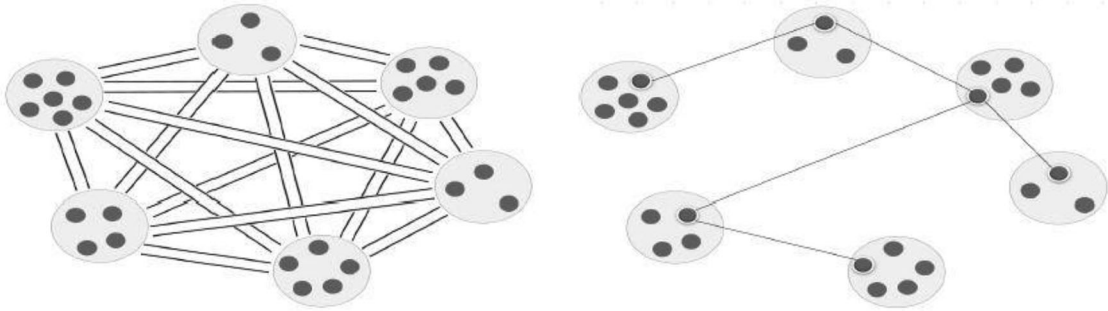
$$w_{uv} := |y_u^s - y_u^p| + \lambda ||\boldsymbol{\theta}_u^s - \boldsymbol{\theta}_v^s||_2, \tag{9}$$

where $\lambda > 0$ is a scaling parameter. The weights of edges that leave vertex 0 or enter vertex $n+1$ are identically zero. The edge-weight for any edge between two consecutive clusters i and $i+1$ consists of two parts: the first part $|y_u^s - y_u^p|$ represents the difference between the model response and physical response; the second part $||\boldsymbol{\theta}_u^s - \boldsymbol{\theta}_v^s||_2$ represents the difference between the calibration parameters of i and $i+1$. On this digraph G with the given edge-weights, we intend to solve the shortest path problem from origin vertex 0 to destination vertex $n+1$. Every path from vertex 0 to vertex $n+1$ in G has exactly $m+1$ edges by construction. Suppose $0\text{-}v_1\text{-}v_2\text{-}\cdots\text{-}v_m\text{-}(n+1)$ is the shortest path identified. Then, those points in S corresponding to $\{v_1, \ldots, v_m\}$ serve as the anchor points. The edge-weights quantify the proximity of the physical and computer experiment outputs and difference between the calibration parameters to minimize erratic changes.

Lemma 1. *Calibration digraph $G = (V, E)$ is acyclic with a topological ordering $\langle 0, 1, \ldots, n, n+1 \rangle$.*

Proof. See Appendix B for proofs. □

Since G is a directed acyclic graph, or DAG for short, we can solve the shortest path problem using an $\mathcal{O}(|E|)$ algorithm that scans outgoing edges from each vertex in the



(a) Calibration graph (b) A generalized minimum spanning tree

Figure 3. (a) A calibration graph where each black circle represents a vertex and each two parallel lines represent edges between vertices of two clusters, and (b) a generalized spanning tree in the calibration graph.

topological order and updates distance-labels as needed (Bellman, 1958; Lawler, 1976).

4. Generalization of non-isometric matching to higher dimensions

Section 3 introduced the curve to surface matching interpretation of calibration with $\mathbf{x} \in \mathbb{R}$ and $\boldsymbol{\theta} \in \mathbb{R}$. This special case allowed us to develop a graph-theoretic approach for anchor point selection that admitted a fast $\mathcal{O}(|E|)$ algorithm. The geometric perspective can be generalized to arbitrary dimensions as a hyper-curve to hyper-surface matching problem. However, in the general setting, there is no straightforward extension of the DAG model. Recall that the model hinges on the natural ordering of the computational and the physical data points on the real line, which does not exist in higher dimensions. Thus, in this section we introduce a different calibration graph model and an associated combinatorial optimization problem to find the anchor points in an arbitrary dimension. As with the special case, the anchor points will subsequently be used in Section 5 to construct prior distributions for our Bayesian model.

For the general case, we construct a *calibration graph* $G = (V, E)$ that is undirected and edge-weighted, where $V = \{1, 2, \ldots, n\}$ corresponds to the n computational data points. We partition V into m clusters, $C_1, \ldots, C_m$, in correspondence with the m physical data points and assign vertex j to a cluster C_i by the same rule in Equation (7). The graph G is a complete m-partite graph with partitions $C_1, \ldots, C_m$, i.e., distinct vertices are adjacent if and only if they belong to different partitions. The edge set can be described formally as

$$E := \bigcup_{i=1}^{m-1} \bigcup_{\ell=i+1}^{m} \{\{u, v\} | u \in C_i, v \in C_\ell\}.$$

Figure 3(a) illustrates this construction.

Finally, before defining the edge weights, we introduce two required concepts. The *calibration vector* of data point s_j is given by $[\boldsymbol{\theta}_j^{s\top}, \boldsymbol{\psi}_j^{s\top}]^\top$. We assign the weight w_e to the edge $e = \{u, v\} \in E$, where $u \in C_i$ and $v \in C_\ell$, by

$$w_e := \begin{cases} |y_u^s - y_i^p| + |y_v^s - y_\ell^p| + \lambda \| \left[\boldsymbol{\theta}_u^{s\top}, \boldsymbol{\psi}_u^{s\top}\right]^\top - \left[\boldsymbol{\theta}_v^{s\top}, \boldsymbol{\psi}_v^{s\top}\right]^\top \|_2 & \text{if } \|\mathbf{x}_u^s - \mathbf{x}_v^s\|_2 \le r \\ |y_u^s - y_i^p| + |y_v^s - y_\ell^p| + M \|\mathbf{x}_u^s - \mathbf{x}_v^s\|_2 & \text{if } \|\mathbf{x}_u^s - \mathbf{x}_v^s\|_2 > r, \end{cases} \quad (10)$$

where λ is a scaling parameter and M is a sufficiently large number used to penalize the computational data points that are far from each other. Note that the weights assigned in (10) extend the idea behind Equation (9). Here, the edge weight between vertices $u \in C_i$ and $v \in C_\ell$, where s_u and s_v are *neighbors* (that is, the Euclidean distance between their control vectors is smaller than a predefined radius r), consists of two parts, similar to (9): the first part measures the distance between each vertex's response and the physical system response associated with the cluster to which it belongs, i.e., $|y_u^s - y_i^p|$ and $|y_v^s - y_\ell^p|$; the second part measures the distance between the corresponding calibration vectors, i.e., $\| [\boldsymbol{\theta}_u^{s\top}, \boldsymbol{\psi}_u^{s\top}]^\top - [\boldsymbol{\theta}_v^{s\top}, \boldsymbol{\psi}_v^{s\top}]^\top \|_2$. Let E_1 denote the set of all edges that join vertex pairs corresponding to control vectors that are at most Euclidean distance r apart. The remainder of the edges, $E_2 = E \setminus E_1$, correspond to edges between computational data points that are not close enough, and we assign relatively large weights to these edges by setting M to a large value. Furthermore, the weight on such edges increases as the distance between the control vectors of the end points increases.

To identify the "optimal" anchor vertices from this calibration graph, we find a minimum weight tree that contains exactly one vertex from each cluster. In the optimization literature, this problem is known as the *Generalized Minimum Spanning Tree* (GMST) problem (Myung *et al.*, 1995) see Figure 3(b). By our construction of the edge weights, a GMST will tend to include edges in E_1 as they are lighter. However, if no GMST exists that only uses edges in E_1, it will be forced to include edges in E_2.

4.1. Integer programming approaches to the GMST problem

The GMST problem was introduced by Myung *et al.* (1995), who showed that it is NP-hard and does not admit a polynomial-time constant-factor approximation algorithm unless P = NP. Various authors have developed and analyzed Integer Programming (IP) formulations for this problem and the strength of the associated Linear Programming (LP) relaxations (Myung *et al.*, 1995; Feremans *et al.*, 2002; Pop, 2004; Pop *et al.*, 2006). Strong formulations, which correspond to tight LP relaxations, are desirable in a branch-and-bound algorithm, as they produce tighter bounds that can

be helpful in pruning the search tree. We employ two such strong formulations with tight LP relaxations for solving the anchor point selection problem in arbitrary dimension.

The class of formulations that were first introduced by Myung *et al.* (1995) employs exponentially many constraints and are analogous to the cutset and subtour elimination formulations of the traveling salesman problem and the minimum spanning tree problem (Bertsimas and Weismantel, 2005). Feremans *et al.* (2002) showed that strengthening a subtour elimination formulation of a more general variant of the GMST problem is among the strongest in terms of the tightness of the LP relaxation. We use this formulation, which Feremans *et al.* (2002) call the *Directed Cluster Subpacking* (DCSUB) formulation, in our computational experiments. This formulation and additional explanation are provided in Appendix C.

Due to the presence of exponentially many constraints, a direct implementation of the entire DCSUB formulation is impractical, even for small scale problems. Nonetheless, a delayed constraint generation approach could be effective in practice (Buchanan *et al.*, 2015; Lu *et al.*, 2018; Moradi and Balasundaram, 2018). This approach starts by relaxing the formulation by omitting a subset of the constraints (typically those that are exponentially many in number). During the normal progress of an LP relaxation-based branch-and-bound algorithm to solve the relaxed IP, whenever an integral solution is detected at some node of the search tree, it is necessary to verify if a constraint that violates this solution exists among those constraints that were excluded. If so, we solve the model at that node again after adding the violated constraints back; otherwise, we continue to branch as usual, thus ensuring the overall correctness of the algorithm. An effective implementation of such an algorithm is possible using the "lazy cut" feature available in most state-of-the-art IP solvers, as long as the identification of the violated constraints can be accomplished quickly.

The second type of formulation we use in our computational experiments is based on the classic Multi-Commodity network Flow (MCF) formulation (Myung *et al.*, 1995; Feremans *et al.*, 2002; Pop, 2004; Pop *et al.*, 2006). The underlying idea of this formulation is to use the flow of a (dummy) "commodity" in the network to trace a path between two vertices by designating one vertex with unit supply for that commodity and the other with unit demand. As the MCF formulation uses only polynomially many constraints and variables, it can be directly implemented and solved using most IP solvers for moderately sized instances. The MCF formulation, which also has a strong LP relaxation, is presented and discussed in greater detail in Appendix C.

Before proceeding to the next section, we point out that although GMST can be applied to any dataset (one- or multi-dimensional), the shortest path model is preferable for one-dimensional problems for the following reasons. First, the shortest path model does not require specifying parameters r and M, and second, it is solvable in time $\mathcal{O}(|V| + |E|)$. By contrast, GMST requires choosing parameters r and M, and it is NP-hard in general. For large scale one-dimensional problems, solving the GMST model may require a sophisticated IP approach and may require substantially more computational effort when compared with the shortest path alternative.

5. Prior and posterior distributions

This section describes how the information about the true physical curve carried by the anchor points, found by approaches discussed in Sections 3 and 4, can be used to construct our prior distributions for the calibration parameters. We also expand posterior distribution (6) using the priors specified in this section, and show how we can make predictions at a new control vector $\mathbf{x}^*$.

Suppose $\boldsymbol{\theta}_i^a$ and $\boldsymbol{\psi}_i^a$ are, respectively, the functional and the global calibration vectors of the anchor point associated with the ith physical data point, i.e., the anchor vertex selected from the ith cluster. Recall that the anchor points are selected by minimizing a weighted combination of two measures: (i) the difference between the model responses and physical responses, and (ii) the distance between the corresponding calibration vectors. As such we can utilize those anchor points to build priors for the calibration parameters in a Bayesian model. To account for the uncertainty associated with the selection of the anchor points, we use variance hyper-parameters as explained next. Define the matrix $\boldsymbol{\Theta}^a := [\boldsymbol{\theta}_1^a, \ldots, \boldsymbol{\theta}_m^a]^\top$ of size $m \times d^\theta$ and the mean vector $\boldsymbol{\psi}^a := \frac{1}{m}\sum_{i=1}^m \boldsymbol{\psi}_i^a$ of length d^ψ. Note that for the latter we take the average of the global calibration vectors of the anchor points, as we assume that the global calibration parameters are constant regardless of the values of the control vectors.

For each component of $\boldsymbol{\psi}^p$, the global calibration parameters, we consider a univariate normal distribution centered at the corresponding element in $\boldsymbol{\psi}^a$ with an unknown variance as the choice of the prior distribution. Therefore, we construct the prior distribution for $\boldsymbol{\psi}^p$ as

$$\boldsymbol{\psi}^p | \boldsymbol{\psi}^a, \boldsymbol{\tau}^2 \sim \mathcal{N}(\boldsymbol{\psi}^a, \mathrm{diag}(\boldsymbol{\tau}^2)), \tag{11}$$

where $\boldsymbol{\tau}^2 = [\tau_1^2, \ldots, \tau_{d^\psi}^2]^\top$ is the vector of variances of the normal distribution.

Applying the same procedure for constructing prior distributions for the functional calibration parameters increases the dimension of the parameter space, since we need to define md^θ variance parameters, which are nuisance parameters and not of interest to our model. Therefore, in order to shrink the parameter space, we use the fact that the kth column of the functional calibration parameters $\boldsymbol{\Theta}^p$, i.e. $\boldsymbol{\Theta}_k^p$, is actually a realization of the functional relationship $\mathcal{F}_k^\theta$. Therefore, the kth column of $\boldsymbol{\Theta}^a$, i.e., $\boldsymbol{\Theta}_k^a$, is a rough estimator of this realization. On this basis, we use a single variance parameter for all the elements in $\boldsymbol{\Theta}_k^p$, and construct the prior distribution for $\boldsymbol{\Theta}_k^p$ as

$$\boldsymbol{\Theta}_k^p | \boldsymbol{\Theta}_k^a, \nu_k^2 \sim \mathcal{N}(\boldsymbol{\Theta}_k^a, \nu_k^2 \mathbf{I}_m), \tag{12}$$

where ν_k^2 is the kth element of the vector of variances $\boldsymbol{\nu}^2$ with length d^θ.

The correctness of the normality assumptions in (11) and (12) is a legitimate concern, as there is no guarantee that the anchor points embrace the true physical curve, due to the limited number of observations. However, we only make the normality assumptions in (11) and (12) for constructing the prior distributions, and the Bayesian model will adjust these priors by likelihood (5).

To specify the posterior distribution, we define proper prior distributions for the rest of the parameters. As such, we get:

$$\begin{aligned}&\pi(\boldsymbol{\Theta}^p, \boldsymbol{\psi}^p, \boldsymbol{v}^2, \boldsymbol{\tau}^2, \boldsymbol{\ell}, \gamma, \sigma^2 | \mathbf{y}^p, \mathbf{X}^p, \boldsymbol{\Theta}^a, \boldsymbol{\psi}^a) \propto \\ &\pi(\mathbf{y}^p | \mathbf{X}^p, \boldsymbol{\Theta}^p, \boldsymbol{\psi}^p, \boldsymbol{\ell}, \gamma, \sigma^2) \pi(\boldsymbol{\Theta}^p | \boldsymbol{\Theta}^a, \boldsymbol{v}^2) \\ &\pi(\boldsymbol{\psi}^p | \boldsymbol{\psi}^a, \boldsymbol{\tau}^2) \pi(\boldsymbol{v}^2) \pi(\boldsymbol{\tau}^2) \pi(\boldsymbol{\ell}) \pi(\gamma) \pi(\sigma^2),\end{aligned} \tag{13}$$

where,

$$\mathbf{y}^p | \mathbf{X}^p, \boldsymbol{\Theta}^p, \boldsymbol{\psi}^p, \boldsymbol{\ell}, \gamma, \sigma^2 \sim \mathcal{N}(0, \boldsymbol{\Sigma} + \sigma^2 \mathbf{I}_m).$$

We refer the reader to Appendix A for the exact prior distributions of parameters in (13), and a discussion of how to sample from the posterior distribution.

In order to make predictions at a new control vector $\mathbf{x}^*$, we introduce variables $\boldsymbol{\psi}^p(t), \boldsymbol{\ell}(t), \gamma(t), \sigma^2(t)$, and $\boldsymbol{\Theta}^p(t)$ as tth draws from the posterior distribution (13) after some burn-in period, where $t \in \{1, ..., T\}$. The first step in the prediction of response y^* is to estimate the associated functional calibration vector of $\mathbf{x}^*$, i.e., $\boldsymbol{\theta}^* = \mathcal{F}^{\theta}(\mathbf{x}^*)$. We can estimate $\boldsymbol{\theta}^*$ based on each $\boldsymbol{\Theta}^p(t)$, where we denote the tth estimation of $\boldsymbol{\theta}^*$ based on $\boldsymbol{\Theta}^p(t)$ as $\boldsymbol{\theta}^*(t)$. To this end we note that as $\boldsymbol{\Theta}^p_k(t)$ is a vector of estimates of $\mathcal{F}^{\theta}_k$ at the design locations $\{\mathbf{x}^p_1, ..., \mathbf{x}^p_m\}$, we can write $\boldsymbol{\Theta}^p_k(t) = [\mathcal{F}^{\theta}_k(\mathbf{x}^p_1), ..., \mathcal{F}^{\theta}_k(\mathbf{x}^p_m)]^{\top} + [\epsilon^{\theta}_1, ..., \epsilon^{\theta}_m]^{\top}$, where $[\epsilon^{\theta}_1, ..., \epsilon^{\theta}_m]$ is a vector of the corresponding error terms. The error term appears because $\boldsymbol{\Theta}^p_k(t)$ does not contain exact evaluations of the function $\mathcal{F}^{\theta}_k$ but only estimations. The following proposition obtains the mean prediction of $\theta^*_k(t)$, i.e., the tth estimation of kth element of $\boldsymbol{\theta}^*$ based on $\boldsymbol{\Theta}^p_k(t)$.

Proposition 1. *Assume* $[\epsilon^{\theta}_1, ..., \epsilon^{\theta}_m]^{\top} \sim \mathcal{N}(0, \sigma^{\theta}_k \mathbf{I}_m)$ *and* $\mathcal{F}^{\theta}_k$ *is a GP with mean zero and covariance function* $\mathcal{K}$*, i.e.,* $\mathcal{F}^{\theta}_k \sim \mathcal{GP}(0, \mathcal{K}(\cdot, \cdot))$*, then*

$$\theta^*_k(t) = \boldsymbol{\Sigma}_{\mathbf{x}^* \mathbf{X}^p} (\boldsymbol{\Sigma}_{\mathbf{X}^p \mathbf{X}^p} + \sigma^{\theta}_k \mathbf{I}_m)^{-1} \boldsymbol{\Theta}^p_k(t). \tag{14}$$

To find point and interval predictions for the new response y^*, we make T predictions based on the T samples we drew from the posterior (13) and the T predictions we made for the vector $\boldsymbol{\theta}^*$ using (14). Recall from Section 2 that $\mathcal{F}^s \sim \mathcal{GP}(0, \mathcal{K}(\cdot, \cdot))$; therefore, we can use the GP predictive distribution to derive the tth prediction as

$$\begin{aligned}&\mathcal{F}^s(\mathbf{x}^*, \boldsymbol{\theta}^*(t), \boldsymbol{\psi}^p(t)) \sim \mathcal{N}\Big(\boldsymbol{\Sigma}_{\mathbf{v}^*(t)\mathbf{V}(t)} (\boldsymbol{\Sigma}_{\mathbf{V}(t)\mathbf{V}(t)} + \sigma^2(t) \mathbf{I}_m)^{-1} \mathbf{y}^p, \\ &\boldsymbol{\Sigma}_{\mathbf{v}^*(t)\mathbf{v}^*(t)} - \boldsymbol{\Sigma}_{\mathbf{v}^*(t)\mathbf{V}(t)} (\boldsymbol{\Sigma}_{\mathbf{V}(t)\mathbf{V}(t)} + \sigma^2(t) \mathbf{I}_m)^{-1} \boldsymbol{\Sigma}_{\mathbf{V}(t)\mathbf{v}^*(t)}\Big),\end{aligned} \tag{15}$$

where $\mathbf{v}^*(t) = [\mathbf{x}^{*\top}, \boldsymbol{\theta}^{*\top}(t), \boldsymbol{\psi}^{p\top}(t)]^{\top}, \mathbf{V}(t) = [\mathbf{X}^p, \boldsymbol{\Theta}^p(t), \mathbb{1}_{m \times d_{\psi}} \operatorname{diag}(\boldsymbol{\psi}^p(t))]^{\top}$, and the covariance matrices are calculated using the tth sample of the covariance parameters, namely $\boldsymbol{\ell}(t)$ and $\gamma(t)$.

Finally, we derive our prediction using distribution (15) as

$$\hat{\mu}^* = \frac{1}{T} \sum_{t=1}^{T} \Big(\boldsymbol{\Sigma}_{\mathbf{v}^*(t)\mathbf{V}(t)} \Big(\boldsymbol{\Sigma}_{\mathbf{V}(t)\mathbf{V}(t)} + \sigma^2(t) \mathbf{I}_m \Big)^{-1} \mathbf{y}^p \Big),$$

$$\hat{\sigma}^{*^2} = \frac{1}{T^2} \sum_{t=1}^{T} \Big(\boldsymbol{\Sigma}_{\mathbf{v}^*(t)\mathbf{v}^*(t)} - \boldsymbol{\Sigma}_{\mathbf{v}^*(t)\mathbf{V}(t)} \Big(\boldsymbol{\Sigma}_{\mathbf{V}(t)\mathbf{V}(t)} + \sigma^2(t) \mathbf{I}_m \Big)^{-1} \boldsymbol{\Sigma}_{\mathbf{V}(t)\mathbf{v}^*(t)} \Big).$$

6. Experimental results

In this section, we evaluate the performance of our methodology by testing it on three synthetic problems and two real problems. We use the Root Mean Squared Error (RMSE) as the measure of accuracy in prediction of responses and calibration vectors to compare the performance of the competing methodologies:

$$\begin{aligned}\text{RMSE}_y &= \sqrt{\frac{1}{n^*} \sum_{q=1}^{n^*} (\hat{y}^*_q - y^*_q)^2} \text{ and } \text{RMSE}_{\theta} \\ &= \sqrt{\frac{1}{n^*} \sum_{q=1}^{n^*} \left|\left| \left[\hat{\boldsymbol{\theta}}^{*\top}_q, \hat{\boldsymbol{\psi}}^{*\top}_q\right]^{\top} - \left[\bar{\boldsymbol{\theta}}^{*\top}_q, \bar{\boldsymbol{\psi}}^{*\top}_q\right]^{\top} \right|\right|^2_2},\end{aligned}$$

where $\hat{y}^*_q, \hat{\boldsymbol{\theta}}^*_q$, and $\hat{\boldsymbol{\psi}}^*_q$ are the predicted response and calibration parameter values for the qth test control variable.

Both the MCF and DCSUB formulations find anchor points for each dataset in less than 2 minutes on all the instances in our testbed, which shows that both formulations are fast in terms of computation time. To choose proper values of λ, r, and M for the GMST model, we use the following empirical approach. First, in order to have the same scale for the inputs and outputs in S and P, before constructing the calibration graph we standardize (divide by the range of each element) the control set $\{\mathbf{x}^p_i, \mathbf{x}^s_j | i \in \{1, 2, ..., m\}, j \in \{1, 2, ..., n\}\}$, the calibration set $\{\boldsymbol{\theta}^s_j, \boldsymbol{\psi}^s_j | j \in \{1, 2, ..., n\}\}$, and the response set $\{y^s_j, y^p_i | i \in \{1, 2, ..., m\}, j \in \{1, 2, ..., n\}\}$. We denote the standardized versions of $\mathbf{x}^p_i, \mathbf{x}^s_j, \boldsymbol{\theta}^s_j, \boldsymbol{\psi}^s_j, y^s_j$, and y^p_i by $\bar{\mathbf{x}}^p_i, \bar{\mathbf{x}}^s_j, \bar{\boldsymbol{\theta}}^s_j, \bar{\boldsymbol{\psi}}^s_j, \bar{y}^s_j$, and $\bar{y}^p_i$. Note that this standardization is only for finding the anchor points, and once the anchor points are chosen, we transform the data back to their original scales for Bayesian inference. After standardization, λ has to be less than two, otherwise the closeness of the calibration vectors is weighted as more important than closeness of the responses. We choose λ from $\{0.1, 0.5, 1, 1.5\}$ in our experiments. Moreover, for M to be a sufficiently large number, it should be larger than the sum of all the weights on the arcs that can be potentially in E_1, that is:

$$\begin{aligned}M \geq &\sum_{j \in \{1,2,...,n\}} \sum_{k \in \{1,2,...,n\}} \sum_{i \in \{1,2,...,m\}} \sum_{\ell \in \{1,2,...,m\}} |\bar{y}^s_j - \bar{y}^p_i| \\ &+ |\bar{y}^s_k - \bar{y}^p_{\ell}| + \lambda \left|\left| \left[\bar{\boldsymbol{\theta}}^{s\top}_j, \bar{\boldsymbol{\psi}}^{s\top}_j\right]^{\top} - \left[\bar{\boldsymbol{\theta}}^{s\top}_k, \bar{\boldsymbol{\psi}}^{s\top}_k\right]^{\top} \right|\right|_2.\end{aligned}$$

Finally, to choose a proper value for r, we plot the pairwise Euclidean distances between all $\bar{\mathbf{x}}^p$ vectors. Then, we use the fact that each point on this plot represents an existing distance between the centers of two clusters in the calibration

Table 1. Calibration functions defined for the three synthetic problems.

Problem	$\mathcal{F}^s(\mathbf{x},\theta,\psi)$	$\mathcal{F}^p(\mathbf{x})$	$\mathcal{F}^\theta(\mathbf{x})$	ψ
1	$\theta\exp(-0.05x^2)(\sin(x)^2+1)$	$\exp(-0.05x^2-0.05x)*(\sin(x)^2+1)$	$\exp(-0.05x)$	–
2	$0.4(x_1^2+x_2^2)\sin^2(0.7x_2)\frac{x_1+x_2}{\theta^2+1}$	$0.4(x_1^2+x_2^2)\sin^2(0.7x_2)$	$(x_1+x_2-1)^{0.5}$	–
3	$\theta+\psi x^2$	$2\sqrt{x}+2.5x^2$	$2\sqrt{x}$	2.5

graph. Therefore, an upper bound on a group of smallest distances, which are close together and disjoint from other groups of distances, can be used as a proper value for r.

6.1. Description of the calibration problems

Table 1 describes synthetic problems used in this study to test the performance of our model on different settings of the calibration problem. The first synthetic problem has one functional calibration variable and one control variable. The second problem has an additional control variable, and the third problem has an additional global calibration variable. We also evaluate the performance of our model on a high-dimensional synthetic problem (that is when d^x, the dimension of the domain of the calibration variables, is large), the results of which are presented in Appendix D. We note that since in practical settings physical observations are noise contaminated, we add a noise drawn from $\mathcal{N}(0,0.05)$ to each generated y^p.

For the first synthetic problem, we locate $m=6$ control vectors, $\mathbf{x}^p$, at locations $\{0.5,1.5,2.5,3.5,\ 4.5,5.5\}$. Then for each $\mathbf{x}^p$, we randomly sample five functional calibration vectors from the interval $[0,2]$; therefore, we have a total of $n=30$ computational data points. Finally, we sample 12 random test control vectors, $\mathbf{x}^*$, from the line segment $[0,6]$ to form a test dataset.

For the second synthetic problem, we locate $m=16$ control vectors, $\mathbf{x}^p$, uniformly on the square $[0,3.5]\times[0,3.5]$. Then for each $\mathbf{x}^p$, we sample 10 functional calibration vectors randomly from the interval $[0,5]$; therefore, we have a total of $n=160$ computational data points. Finally, we sample 10 random test control vectors, $\mathbf{x}^*$, from the same square $[0,3.5]\times[0,3.5]$ to form a test dataset.

For the third synthetic problem, we follow the setting used by Brown and Atamturktur (2018), where we choose $m=15$ control vectors for training at locations $\{0,0.05,0.10,\ 0.15,\ 0.20,\ 0.25,0.30,\ 0.35,0.40,0.70,0.75,0.80,0.85,0.90,0.95\}$, and use five physical control vectors for testing at locations $\{0.45,0.50,0.55,0.60,0.65\}$. We sample 10 functional calibration vectors for each $\mathbf{x}^p$ from the square $[0,5]\times[0,5]$; therefore, we have a total of $n=150$ computational data points.

The first real problem from a spot welding application was originally introduced by Bayarri *et al.* (2007). This problem has three control variables and one calibration variable. The dataset associated with spot welding contains 12 and 35 data points sampled from the physical experiments and the computation model, respectively. The second real problem studied by Pourhabib *et al.* (2015) has one control variable and one calibration variable, and the associated dataset contains 11 and 150 data points sampled from the physical experiments and the computation model, respectively. This instance arises from a PVA-treated buckypaper fabrication process.

For the real problems, we partition the sets of physical data points using four-fold cross validation to form training and test datasets. Therefore, for each iteration of cross validation for the spot welding dataset, we have eight physical data points in the training set and four physical data points in the test set. Similarly, for the PVA dataset we have eight to nine physical data points in the training set, and two to three data points in the test set in each iteration of cross validation. We note that the cross validation does not affect the size of the computational datasets, i.e., $n=35$ for spot welding and $n=150$ for the PVA dataset.

6.2. Results

We compare the results of our proposed methods with competing functional calibration methodologies. These include Non-Parametric Functional Calibration (NFC) (Pourhabib *et al.*, 2018), Parametric Functional Calibration (PFC) (Pourhabib *et al.*, 2015), and Non-Parametric Bayesian Calibration (NBC) (Brown and Atamturktur, 2018), which all require surrogate modeling for handling expensive computational models. When we use the generalized minimum spanning tree model on a calibration digraph to find a set of anchor points, we refer to our approach as BMNC. If we use the shortest path model on a calibration digraph, we call the approach BMNC-DAG.

For each of the aforementioned problems, we choose the values of λ, r, and M following the empirical approach explained in Section 6 (see Table 2). Tables 3 and 4 compare the performance of BNMC and BNMC-DAG in terms of RMSE_θ, RMSE_y, and computation time with the other competing methodologies.

For the first synthetic problem, the second and third columns of Table 3 show that BNMC and BNMC-DAG both perform more accurately than the other methodologies in terms of RMSE_y, and they have the same order of accuracy in terms of RMSE_θ. Moreover, Table 4 shows that BNMC-DAG performs slightly faster than BNMC. However, because we only have $n=30$ computational and $m=6$ physical data points, this difference is not very large. To better compare the computational costs of the approach using the shortest path and the GMST models, we choose values of ns from the set $\{100,200,300,400,500\}$ and run both BNMC and BNMC-DAG. As expected, the shortest path model in BNMC-DAG finds the anchor points for all the values of n in less than a second, whereas GMST requires 3.6, 21.2, 65.3, 144.7, and 323.5 seconds for $n\in\{100,200,300,400,500\}$, respectively.

For the second synthetic problem, because the dimension of $\mathbf{x}^p$ is greater than one, we cannot apply BNMC-DAG.

Table 2. The calibration graph parameters for the five calibration problems.

Parameter	*First synthetic problem*	*Second synthetic problem*	*Third synthetic problem*	*PVA*	*Spot Welding*
λ	0.5	0.5	0.5	0.5	0.5
r	1.5	0.9	0.16	0.5	4
M	10^5	10^5	10^5	10^5	10^5

Table 3. RMSE_θ and RMSE_y of different methodologies for the five calibration problems.

	First synthetic problem		*Second synthetic problem*		*Third synthetic problem*		*PVA*	*Spot Welding*
Methodology	RMSE_y	RMSE_θ	RMSE_y	RMSE_θ	RMSE_y	RMSE_θ	RMSE_y	RMSE_y
NFC	0.184	0.254	0.143	0.202	–	–	0.379	0.683
PFC	0.226	0.244	0.296	0.390	–	–	0.450	1.115
NBC	0.162	0.356	0.132	0.627	0.172	0.426	0.281	0.516
BNMC	0.098	0.271	0.076	0.356	0.063	0.354	0.296	0.409
BNMC-DAG	0.098	0.265	–	–	–	–	0.288	–

Table 4. The computation times (in seconds) for the five calibration problems.

Methodology	*First synthetic problem*	*Second synthetic problem*	*Third synthetic problem*	*PVA*	*Spot Welding*
NFC	2.88	8.02	–	3.29	0.77
PFC	18.04	253.08	–	176.95	19.06
NBC	45.26	307.31	230.19	180.92	68.04
BNMC	125.15	169.23	147.01	159.84	117.53
BNMC-DAG	123.91	–	–	144.54	–

However, the fourth and fifth columns of Table 3 show that BNMC outperforms the other methodologies in terms of RMSE_y and has the second best accuracy in terms of RMSE_θ.

For the third synthetic problem, we only compare the results of NBC and BNMC, since the codes for NFC and PFC are written only for univariate calibration problems. The sixth and the seventh columns of Table 3 show that BNMC outperforms NBC both in terms of RMSE_y and RMSE_θ. We note that the reported RMSE_y for NBC in Brown and Atamturktur (2018) under the cheap computational code assumption is 0.0538, which is a better accuracy compared with that of BNMC; however, here BNMC is superior when NBC uses surrogate modeling.

Since the true values of the calibration parameters are unknown for the real problems, we compare the results only in terms of RMSE_y. The eighth column (PVA) of Table 3 shows that BNMC, BNMC-DAG, and NBC have the same order of accuracy and perform better than NFC and PFC. Finally, we observe in the last column of Table 3 that for the Spot Welding problem BNMC outperforms the other competing methodologies by a large margin. We attribute this performance to the capability of BNMC in handling expensive computational models with a small number of computational data points.

Figure 4 shows the 95% confidence interval predictions for the responses and the functional calibration parameter for the test datasets of the synthetic problems. Note that since $\mathbf{x}^p \in \mathbb{R}^2$ for the second synthetic problem, we plot the predicted values against their indices in Figures 4(c) and 4(d), and connect the data points to each other for better visualization. Moreover, although we added white noise to response variables to mimic real-world processes, we show the denoised responses for clearer illustration.

As noted in Section 2, due to the limited number of samples we collect from the computational model, we cannot accurately recover $\mathcal{F}^\theta$, but the way we train the hyperparameters of the GP aims to compensate for this limitation. We can observe this in Figure 4, where the predictions of the response values have better accuracy and tighter confidence intervals compared to those of functional calibration parameter values.

For illustration, Figure 5 shows the 95% confidence interval predictions for one of the test datasets created in the cross validation process for each of the spot welding and the PVA problems. Since we do not know the true functional calibration parameter values, we cannot provide a similar plot for the calibration predictions. Similar to Figures 4(c) and 4(d), we plot the predicted values against their indices in Figure 5(a) for better visualization.

We refer the reader to Appendix E for an analysis of the residuals to validate the assumptions made in our proposed approach, especially those made in Equations (2) and (3).

7. Concluding remarks

We proposed a Bayesian non-isometric matching calibration model for expensive computational models. A limited budget to evaluate computational models led to the use of a GP, which was trained *during* the calibration procedure. We used Bayesian statistics to simultaneously train the hyperparameters of the GP's covariance function and make inferences on the calibration parameters associated with the physical data points. To construct informative prior distributions for our new approach, we used a geometric interpretation of calibration based on non-isometric curve to surface matching. This point of view enabled us to develop graph-theoretic approaches to address the problem of finding a set of anchor points used in constructing informative prior distributions. For the special case of a single control and calibration variable, we introduced a shortest path model on a directed acyclic calibration graph to tackle the problem of

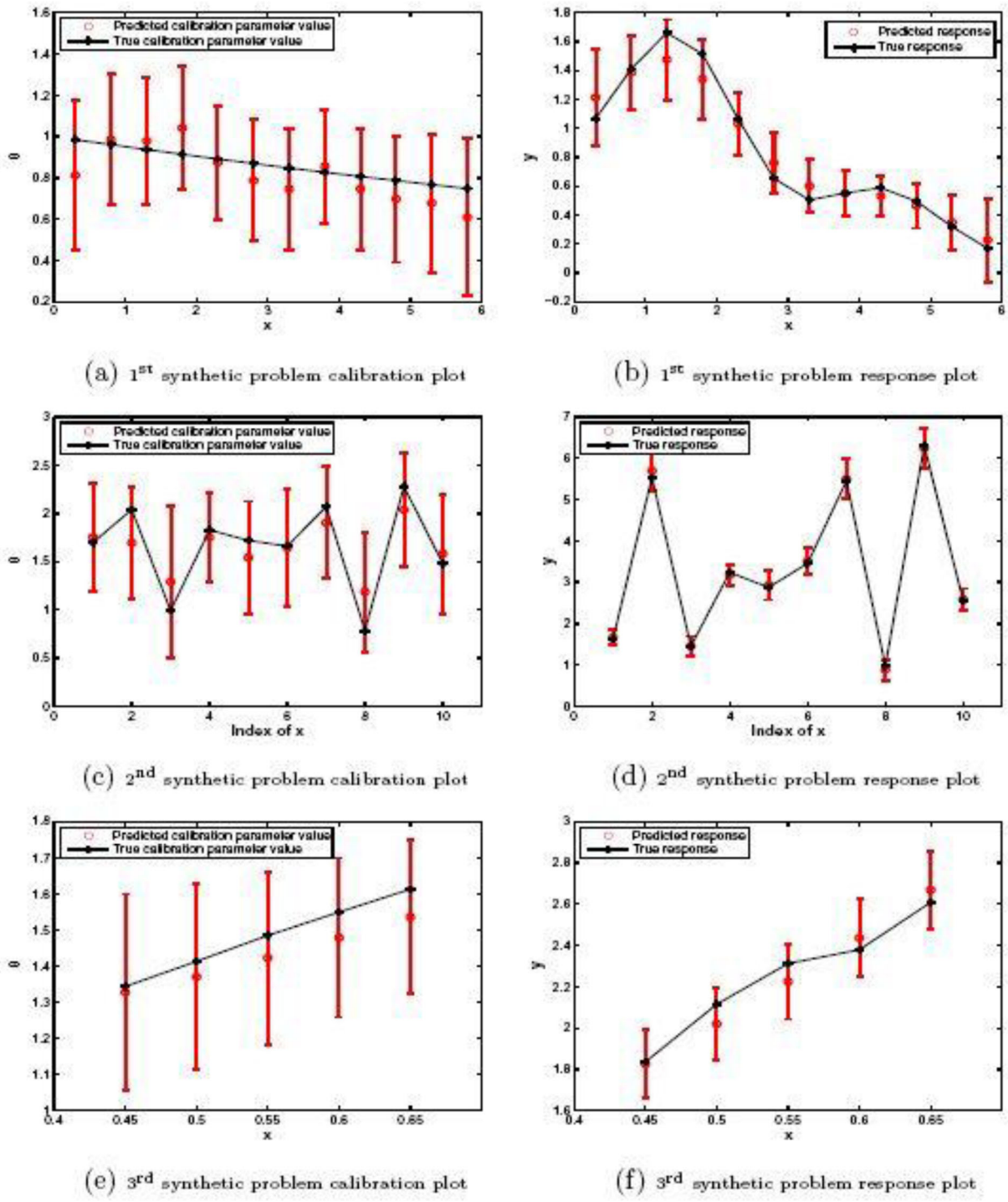


Figure 4. The 95% confidence interval predictions for functional calibration parameter values and responses for the test datasets of the synthetic problems.

finding anchor points, while for the general case, we introduced the generalized minimum spanning tree model. Our numerical experiments conducted on four benchmark calibration problems showed that our approach outperformed the existing calibration models under the assumption of expensive computational models.

The framework developed in this article could be extended in several ways. We only considered a single computational data point to construct a prior distribution for each calibration parameter; however, information on multiple computational data points could be taken into account. An implementation of this idea, of course, requires developing new combinatorial optimization techniques capable of choosing an appropriate number of computational data points. Another interesting research path would be to consider data uncertainty formally in the calibration graph model instead of using a deterministic calibration graph model, and then using Bayesian inference to deal with the uncertainty in the data. Moreover, one may consider including an independent discrepancy function in Equation (3), as

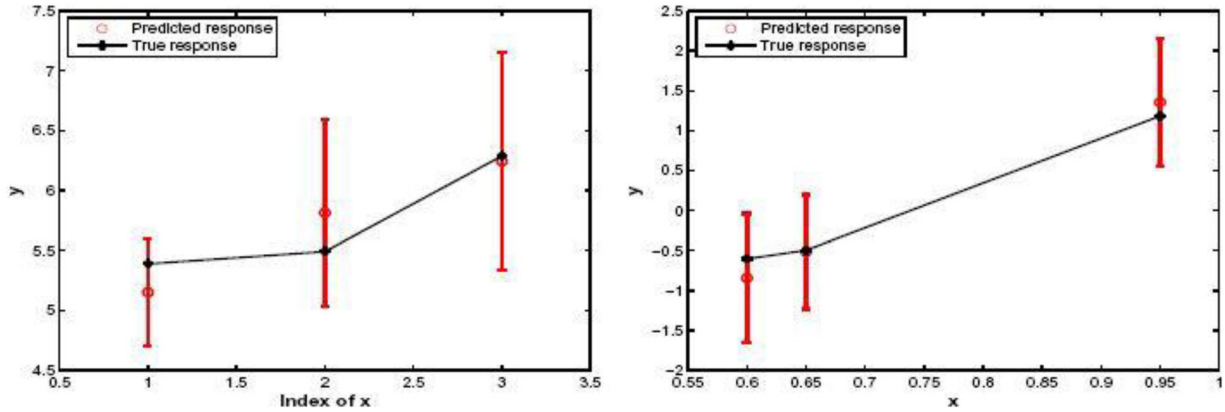


Figure 5. The 95% confidence interval predictions for one of the test datasets created in the cross validation process for each of the spot welding and the PVA problems.

noted in Section 2. Finally, the proposed approach would potentially benefit from using cross validation for the selection of the tuning parameters such as λ, r, and M.

Funding

Austin Buchanan is supported by the National Science Foundation under Grant No. 1662757. This work was completed utilizing the High Performance Computing Center facilities of Oklahoma State University at Stillwater.

Acknowledgments

The authors acknowledge Chuck Zhang from Georgia Institute of Technology for providing the authors with the PVA-treated buckypaper fabrication data.

Notes on contributors

Babak Farmanesh received his BS in industrial engineering from Sharif University of Technology in 2014, and his PhD from the School of Industrial Engineering and Management at Oklahoma State University in 2018. He is currently working as a Data Scientist at DELL Technologies. His research interest includes statistical machine learning, optimization, and operations research.

Arash Pourhabib received his BS in industrial engineering from Sharif University of Technology in 2008, and his PhD from the Department of Industrial and Systems Engineering at Texas A&M University in 2014. He is currently an adjunct assistant professor at the School of Industrial Engineering and Management at Oklahoma State University and a Data Scientist at Google. His research interests are in the areas of system informatics and statistical machine learning. He is a member of INFORMS and IISE.

Austin Buchanan is an assistant professor in the School of Industrial Engineering & Management at Oklahoma State University. In 2015, he received his PhD in industrial and systems engineering at Texas A&M University. His research interests are in network optimization and integer programming. He is a member of INFORMS and was recently awarded an NSF CAREER award.

Balabhaskar Balasundaram is an associate professor and the holder of Wilson Bentley Professorship in the School of Industrial Engineering and Management at Oklahoma State University. He received his B.Tech in mechanical engineering from the Indian Institute of Technology-Madras in 2002, and his PhD in industrial engineering from Texas A&M University in 2007. His research interests include combinatorial optimization and integer linear programming, specifically graph-theoretic models and their applications in social and biological network analysis. He is a member of IISE, INFORMS, SIAM, and MOS.

ORCID

Arash Pourhabib http://orcid.org/0000-0002-9394-8196
Balabhaskar Balasundaram http://orcid.org/0000-0002-3490-4257
Austin Buchanan http://orcid.org/0000-0003-2999-9666

References

Baltsavias, E., Gruen, A., Eisenbeiss, H. Zhang, L.and Waser, W. (2008) High-quality image matching and automated generation of 3D tree models. *International Journal of Remote Sensing*, **29**(5), 1243–1259.

Bayarri, M.J., Berger, J.O., Paulo, R., Sacks, J., Cafeo, J.A., Cavendish, J., Lin, C.-H. and Tu, J. (2007) A framework for validation of computer models. *Technometrics*, **49**(2), 138–154.

Bellman, R. (1958) On a routing problem. *Quarterly of Applied Mathematics*, **16**(1), 87–90.

Bertsimas, D. and Weismantel, R. (2005) *Optimization Over Integers*, Dynamic Ideas, Belmont, MA.

Bronstein, A.M., Bronstein, M.M. and Kimmel, R. (2003) Expression-invariant 3d face recognition, in *International Conference on Audio- and Video-based Biometric Person Authentication*, Springer, Berlin, Heidelberg, pp. 62–70.

Bronstein, A.M., Bronstein, M.M. and Kimmel, R. (2005) Three-dimensional face recognition. *International Journal of Computer Vision*, **64**(1), 5–30.

Brown, D.A. and Atamturktur, S. (2018) Nonparametric functional calibration of computer models. *Statistica Sinica*, **28**, 721–742.

Buchanan, A., Sung, J.S. Butenko, S. and Pasiliao, E.L. (2015) An integer programming approach for fault-tolerant connected dominating sets. *INFORMS Journal on Computing*, **27**(1), 178–188.

Craig, P.S., Goldstein, M. Rougier, J.C. and Seheult, A.H. (2001) Bayesian forecasting for complex systems using computer simulators. *Journal of the American Statistical Association*, **96**(454), 717–729.

Ezzat, A.A., Pourhabib, A. and Ding, Y. (2018) Sequential design for functional calibration of computer models. *Technometrics*, **60**(3), 286–296.

Fang, K.-T., Li, R. and Sudjianto, A. (2005) *Design and Modeling for Computer Experiments*. CRC Press, Boca Raton, FL.

Feremans, C., Labbé, M. and Laporte, G. (2002) A comparative analysis of several formulations for the generalized minimum spanning tree problem. *Networks*, **39**(1), 29–34.

Goldstein, M. and Rougier, J. (2009) Reified Bayesian modelling and inference for physical systems. *Journal of Statistical Planning and Inference*, **139**(3), 1221–1239.

Gruen, A. and Akca, D. (2005) Least squares 3D surface and curve matching. *ISPRS Journal of Photogrammetry and Remote Sensing*, **59**(3), 151–174.

Han, G.,T. Santner, J. and Rawlinson, J.J. (2009) Simultaneous determination of tuning and calibration parameters for computer experiments. *Technometrics*, **51**(4), 464–474.

Higdon, D., Gattiker, J., Williams, B. and Rightley, M. (2008) Computer model calibration using high-dimensional output. *Journal of the American Statistical Association*, **103**(482), 570–583.

Higdon, D., Kennedy, M., Cavendish, J.C., Cafeo, J.A. and Ryne, R.D. (2004) Combining field data and computer simulations for calibration and prediction. *SIAM Journal on Scientific Computing*, **26**(2), 448–466.

Joseph, V.R. and Melkote, S.N. (2009) Statistical adjustments to engineering models. *Journal of Quality Technology*, **41**(4), 362–375.

Kennedy, M.C. and O'Hagan, A. (2001) Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **63**(3), 425–464.

Lawler, E. (1976) *Combinatorial Optimization: Networks and Matroids*, Holt, Rinehart, and Winston, New York, NY.

Loeppky, J., Bingham, D. and Welch, W. (2006) Computer model calibration or tuning in practice. Technical report, University of British Columbia, Vancouver, BC, Canada.

Lu, Y., Moradi, E. and Balasundaram, B. (2018) Correction to: Finding a maximum *k*-club using the *k*-clique formulation and canonical hypercube cuts. *Optimization Letters*, **12**(8), 1959–1969.

Moradi, E. and Balasundaram, B. (2018) Finding a maximum *k*-club using the *k*-clique formulation and canonical hypercube cuts. *Optimization Letters*, **12**(8), 1947–1957.

Myung, Y.-S., Lee, C.-H. and Tcha, D.-W. (1995) On the generalized minimum spanning tree problem. *Networks*, **26**(4), 231–241.

Plumlee, M. and Joseph, V.R. (2018) Orthogonal Gaussian process models. *Statistica Sinica*, **28**(2), 601–619.

Plumlee, M., Joseph, V.R. and Yang, H. (2016) Calibrating functional parameters in the ion channel models of cardiac cells. *Journal of the American Statistical Association*, **111**(514), 500–509.

Pop, P.C. (2004) New models of the generalized minimum spanning tree problem. *Journal of Mathematical Modelling and Algorithms*, **3**(2), 153–166.

Pop, P.C., Kern, W. and Still, G. (2006) A new relaxation method for the generalized minimum spanning tree problem. *European Journal of Operational Research*, **170**(3), 900–908.

Pourhabib, A., Huang, J.Z., Wang, K., Zhang, C., Wang, B. and Ding, Y. (2015) Modulus prediction of buckypaper based on multi-fidelity analysis involving latent variables. *IIE Transactions*, **47**(2), 141–152.

Pourhabib, A., Tuo, R. He, S., Huang, J.Z., and Ding, Y. (2018) Functional calibration of computer models. Manuscript.

Pratola, M.T., Sain, S.R., Bingham, D., Wiltberger, M. and Rigler, E.J. (2013) Fast sequential computer model calibration of large nonstationary spatial-temporal processes. *Technometrics*, **55**(2), 232–242.

Rasmussen, C.E. (2004) Gaussian processes for machine learning, in *Advanced Lectures on Machine Learning*, Springer, Cambridge, MA, pp. 63–71.

Reese, C.S., Wilson, A.G. Hamada, M. Martz, H.F. and Ryan, K.J. (2004) Integrated analysis of computer and physical experiments. *Technometrics*, **46**(2), 153–164.

Santner, T.J., Williams, B.J. and Notz, W.I. (2013) *The Design and Analysis of Computer Experiments*, Springer, New York, NY.

Schölkopf, B., Herbrich, R. and Smola, A.J. (2001) A generalized representer theorem, in *International Conference on Computational Learning Theory*, Springer, Berlin, Heidelberg, pp. 416–426.

Tuo, R. and Wu, C.F.J. (2015) Efficient calibration for imperfect computer models. *The Annals of Statistics*, **43**(6), 2331–2352.

Tuo, R. and Wu, C.F.J. (2016) A theoretical framework for calibration in computer models: Parametrization, estimation and convergence properties. *SIAM/ASA Journal on Uncertainty Quantification*, **4**(1), 767–795.

Williams, B., Higdon, D., Gattiker, J., Moore, L., McKay, M., Keller-McNulty S. (2006). Combining experimental data and computer simulations, with an application to flyer plate experiments. *Bayesian Analysis*, **1**(4), 765–792.

Xiong, Y., Chen, W., Tsui, K.-L., and Apley, D.W. (2009) A better understanding of model updating strategies in validating engineering models. *Computer Methods in Applied Mechanics and Engineering*, **198**(15), 1327–1337.