



Design optimization by integrating limited simulation data and shape engineering knowledge with Bayesian optimization (BO-DK4DO)

Jia Hao¹ · Mengying Zhou¹ · Guoxin Wang¹ · Liangyue Jia¹ · Yan Yan¹

Received: 18 April 2019 / Accepted: 17 February 2020 / Published online: 2 March 2020
© Springer Science+Business Media, LLC, part of Springer Nature 2020

Abstract

Surrogate models have been widely studied for optimization tasks in the domain of engineering design. However, the expensive and time-consuming simulation cycles needed for complex products always result in limited simulation data, which brings a challenge for building high accuracy surrogate models because of the incomplete information contained in the limited simulation data. Therefore, a method that builds a surrogate model and conducts design optimization by integrating limited simulation data and engineering knowledge through Bayesian optimization (BO-DK4DO) is presented. In this method, the shape engineering knowledge is considered and used as derivative information which is integrated with the limited simulation data with a Gaussian process (GP). Then the GP is updated sequentially by sampling new simulation data and the optimal design solutions are found by maximizing the GP. The aim of BO-DK4DO is to significantly reduce the required number of computer simulations for finding optimal design solutions. The BO-DK4DO is verified by using benchmark functions and an engineering design problem: hot rod rolling. In all scenarios, the BO-DK4DO shows faster convergence rate than the general Bayesian optimization without integrating engineering knowledge, which means the required amount of data is decreased.

Keywords Bayesian optimization · Limited simulation data · Engineering knowledge · Surrogate model · Design optimization

Introduction

Design optimization is very important for complex product design process, which allows designers to locate optimal solutions within design space. Recently, the utilization of optimization has increased dramatically in product design. The nature of complex products includes complex structure, intensive technology, long development cycle, etc. The design process involves the integration of multidisciplinary knowledge and the coordination of multiple professionals,

which makes the computer simulations of complex products are extremely time-consuming and the optimizations are quite difficult. Optimization in design has the following characteristics (Forrester et al. 2008; Jones et al. 1998; Shahriari et al. 2016): High-dimensional, Black-box and Expensive in terms of computer simulations (HBE). These characteristics lead to a situation where a huge design space must be explored, which means that designers must expend considerable time and cost for computer simulations to find an optimal design solution.

Most engineering designs require a large number of experiments and/or computer simulations to evaluate design solutions (Liu et al. 2016; Sahnoun et al. 2016). A single computer simulation of complex product can take many hours, days, or even months to complete. One of the effective ways to solve this issue is to build a surrogate model to replace the computer simulations, and then use this to obtain an optimal solution. A surrogate model is a low-cost mathematical model used to replace the time-consuming computer simulation (Vidal and Archer 2016). Surrogate models have been used for tasks such as optimization, design space exploration, and sensitivity analysis (Gorissen and Dhaene 2010) due to their low computational expense. When enough sim-

✉ Jia Hao
Haojia632@bit.edu.cn
Mengying Zhou
zmygz@163.com
Guoxin Wang
wangguoxin@bit.edu.cn
Liangyue Jia
719834513@qq.com
Yan Yan
yanyan331@bit.edu.cn

¹ Beijing Institute of Technology, No. 5 South Street, Zhongguancun, Haidian District, Beijing, China

ulation data are available, surrogate models can be used to support designers in finding an optimal solution since building a high-accuracy surrogate model is easy. However, when only limited simulation data are available, building a high-accuracy surrogate model is quite difficult. Therefore, developing new method for building surrogate models and conducting optimization with limited simulation data is a critical and challenging research problem which requires more research focus.

Integrating engineering knowledge and limited simulation data provides a new way to build a surrogate model and conducting design optimization when only limited simulation data are available. Existing hybrid methods try to extract synergistic rules from domain theory (Monisha and Peter 2017; Towell and Shavlik 1994) or knowledge from historical datasets (Yu et al. 2008; Zhang et al. 2019) to optimize the settings of the neural network structure. Although these methods can be used to build surrogate models and integrate the extra information, the performances of the surrogate model are heavily dependent on the algorithms used to extract knowledge. For integrating synergistic rules, it also relies on rules-to-network algorithms. Besides, these methods cannot integrate the mapping knowledge directly that designers hold, which contains useful information that is not contained in the limited simulation data. In this work, we develop a method which is capable of building surrogate model and conducting design optimization synchronously by integrating limited simulation data and shape knowledge, which is a specific kind of engineering knowledge. Our method realizes the direct integration of the engineering knowledge owned by designers. Besides, we use GP as a surrogate model which requires less data to train comparing with neural networks.

The rest of the paper is structured as follows. In section 2 there is an evaluation of some related research works and the research gaps are also pointed out. In section three the details of the proposed method are described. In section four several experiments are conducted to verify this method. In section five the empirical results are analyzed and discussed while in section six there is a summary of this work and some contributions are presented.

Related work

Optimization in engineering design

The use of optimization techniques in engineering design has been on the rise steadily as the computational capabilities of the computers are increasing (Liu et al. 2018). Optimization has a wide range of applications in engineering design, such as structural design [e.g., pressure vessel design (Fatemeh et al. 2019), welded beam design (Du et al. 2018)], shape optimization (Daróczy et al. 2018; Fengjie

and Lahmer 2018), topology optimization (e.g., airfoil (Bhattacharyya et al. 2019), fluid (Yoshimura et al. 2017), inverse optimization [Johnson–Cook model (Ning et al. 2018)], processing planning (Pratap et al. 2018; Zhang et al. 2018), product designs (Du et al. 2019), etc. Routine tasks such as design optimization, sensitivity analysis and design space exploration usually require thousands or even millions of simulation evaluations (Sefat et al. 2012). For many engineering design problems, however, a single simulation is extremely time-consuming which is a main obstacle to implement design optimization.

One way of alleviating this burden is by exploiting surrogate models for optimization. Yoshimura et al. (Yoshimura et al. 2017) proposes a non-gradient-based approach applied to fluid problems. This paper uses genetic algorithm for topology optimization assisted by the Kriging surrogate model. Song et al. (Song et al. 2016) conducts a sensitivity analysis and carries out a reliability based design optimization by taking into account the uncertainties in the TWB configuration. Wang et al. (2017) proposes a novel surrogate-assisted particle swarm optimization combining uncertainty and performance based criteria for expensive problems.

Among the optimization techniques based on surrogate models, Bayesian optimization is a state-of-the-art global optimization method with two attractive advantages. One is that it can be applied to problems where it is vital to optimize a performance criterion while keeping the number of evaluations small (Calandra et al. 2016). Another is that it considers the uncertainty which is ubiquitous in engineering design in conducting the sampling (updating) process. While Bayesian optimization is a popular probabilistic approach when limited data are available, as indicated in Trucano et al. (2006), the prior surrogate model are often difficult to specify due to the lack of prior knowledge. Subjectively assigned prior distributions may yield unstable posterior distributions (Aughenbaugh and Herrmann 2007), which undermines the advantage of Bayesian updating. Therefore, our research focuses on the integration of engineering knowledge and limited simulation data into the prior surrogate model to improve the accuracy of it. The model itself is refined and its demand for sampling is further reduced in the updating process, which is more suitable for limited data problems.

Actually, the performance of surrogate model depends on two key factors: (1) the type of surrogate models, such as neural network, Kriging, response surface model and so on, and (2) the information contained in training data. When only limited data are available, it's challenging to build a high-accurate surrogate model. The following two subsections summarize the ways of building surrogate models with limited data from two perspectives: data augmentation and knowledge integration.

Building surrogate models by data augmentation

Limited simulation data brings serious challenges for building surrogate models in the domain of engineering design (Dougherty et al. 2015). The main difficulty is that information contained in limited data is often insufficient. Currently, two types of methods from artificial intelligent provide ideas and ways to build surrogate models with limited simulation data. One such method is data augmentation, and the other is knowledge integration.

Data augmentation refers to the generation of new data records based on existing information. It can be implemented in many ways, for example, interpolation (Amsallem et al. 2009; Zhao et al. 2017), noise injection (Bella et al. 2007; Fortuna et al. 2009), data sampling (Gorissen and Dhaene 2010), and virtual sample generation (Kang et al. 2019; Li et al. 2013, 2018a; Tsai and Li 2008). For these methods, virtual samples can also be generated based on prior knowledge (Li et al. 2003, 2012). Although these methods can improve accuracy, they also have limitations. For data augmentation, new errors will inevitably be introduced. Besides, the information contained in new data is still similar with existing data or knowledge.

Building Surrogate Models by Integrating Knowledge

Knowledge integration is used to add some restrictions to the model. There are two ways to be addressed separately, including regularization and transfer learning (Min et al. 2017). For the model regularization, only very simple knowledge can be integrated when defining regularization terms while with transfer learning it is difficult to interpret and integrate different data.

The main way is integrating more complex knowledge into surrogate models. In this way, currently, there are only a few related studies about integrating engineering knowledge and limited simulation data in the domain of engineering design (Kotlowski and Slowinski 2013). Hence, we also review papers in the domains of artificial intelligence and machine learning. The existing methods can be classified from two perspectives: the types of knowledge studied and the types of models (Gaussian process, neural network) integrating knowledge and data. Different authors use various designations for knowledge, such as knowledge (Nagarajan et al. 2018), prior knowledge (Aguirre and Furtado 2007; Parrado-Hernández et al. 2012), constraints (Wang and Welch 2018; Wang and Berger 2016), hints (Abu-Mostafa 1990; Sill 1998; Sill and Abu-Mostafa 1997) etc.

Gaussian processes and neural networks are the two main methods for integrating knowledge. There are many works for integrating knowledge and data using neural networks. The main mechanism is usually realized by putting con-

straints on the parameters of a neural network. Abu-Mostafa (1990) propose a method using neural networks to learn from hints, as expressed by a set of data. Sill (1998) and Sill and Abu-Mostafa (1997) further develops this work, and proposes a general method for incorporating monotonicity information into neural networks. Based on this work, Daniels and Velikova (2010) presents a monotonic neural network, and Gupta et al. (2018) presented a neural network with shape constraints. Different from these methods, our previous work (Hao et al. 2018) proposes the integration of limited simulation data and engineering knowledge through an evolutionary neural network.

Another mechanism builds a hybrid learning network which use both knowledge and data during learning through connectionist machine learning. Towell and Shavlik (1994) describes a hybrid learning system knowledge-based artificial neural networks (KBANN) which uses both synergistic rules and classified training examples. An extension of this approach (Shavlik 1994) initializes an RNN by synergistic knowledge expressed as a Finite State Automaton (FSA). Hybrid methods have been used in product design. Monisha and Peter (2017) uses a knowledge based neural network (KBNN) for designing of triple band planar inverted-F antenna (PIFA). Prior knowledge is used in the form of trained radial basis function network (RBFN) in addition with back propagation network (BPN). Yu et al. (2008) creates a learning-based hybrid method named KBANN-DT, which combines knowledge-based artificial neural network (KBANN) and CART decision tree (DT). The KBANN is applied to realizing the mapping between customer needs and product specifications which integrates both two knowledge resources, i.e., domain theory and historical database. Zhang et al. (2019) proposes a knowledge-based artificial neural network (ANN) to model the relationship between customer requirement attributes and product service system (PSS) base types. The initial structure of ANN is defined by domain knowledge which is extracted from the data of historical configuration instance data sets using a priori algorithm.

Although, it is possible to integrate the data and knowledge using the neural network, there are limitations. The first is that these methods require deliberately specifying the structure of the neural networks and/or the normalization terms, which is an experience-based process and is also very difficult. The second is that neural networks requires more data to train compared with other models. The third limitation is that it's very hard to form a universal method for integrating different kinds of knowledge into neural networks, which means for different knowledge totally different structure and normalization terms must be designed.

Gaussian process (GP) is a popular tool for function estimation which allows a straightforward combination of function and derivative observations in a prior model. Wu and Feb (2017) exploited exact derivative values in GP which

are hard to acquire in real-world problems. Some types of prior knowledge have been successfully incorporated including monotonicity (Riihimäki and Vehtari 2010; Wang and Welch 2018; Wang and Berger 2016), monotonic convex or concave (Lenk and Choi 2017; Wang and Berger 2016), unimodality (Andersen et al. 2017; Jauch and Peña 2016; Li et al. 2018b). The method of Wang and Welch (2018) is designed for a dominant function which is composed of functions with monotonicity constraints. Preliminary work (Riihimäki and Vehtari 2010) also study Gaussian processes with monotonicity information, which paves the way for much following research. For unimodality work, unfortunately, (Andersen et al. 2017) has to resort to approximate inference to compute intractable posterior distributions and focused on univariate problems to provide proof of concept. As the dimensions get larger and larger, the reliability of Jauch and Peña (2016) will be reduced and it eventually fails. The method proposed by Li et al. (2018b) is only limited for the problems of producing something with a target value. However, for many engineering design problems, the performances of products are supposed to be larger or smaller as possible. Therefore, we can safely conclude that none of the existing works solves our problem with multiple knowledge including monotonicity, unimodality, multimodal and any shape that can be decomposed as several monotonic intervals, thus the problem remains open.

The above two methods are used for integrating data and knowledge. However, they are mainly applied to tuning the hyperparameters of machine learning models and thus rarely are used to build surrogate models in the domain of engineering design. In this paper we develop a method for building surrogate models and conducting design optimization by integrating shape engineering knowledge and limited simulation data through Bayesian optimization (BO-DK4DO).

Method

Limited simulation data and engineering knowledge

(1) Limited simulation data

Limited simulation data means that the data are inadequate to describe the design space for building an accurate surrogate model. This data commonly includes design variables and their corresponding performances. For the purposes of this work, a limited simulation dataset has the following characteristics:

- The amount of data is limited

If a solution and its performances are regarded as a piece of data, the dataset only contains a few such pairs.

However, there is no predetermined threshold value that determines whether a dataset is limited simulation data or not, this depends on the accuracy we want to achieve. For a 1-dimension *Lipschitz-continuous* function, which satisfies Eq. (1), the amount of data must reach $\left(\frac{C}{2\epsilon}\right)^2$ to get a model that has the prediction error ϵ (Brochu et al. 2010).

$$\|f(x_1) - f(x_2)\| \leq C\|x_1 - x_2\| \quad (1)$$

If we have a 1-dimension function and want to get a model has prediction error 0.05 (assuming C is 1), then we need 100 data records. From the above analysis, it can be said that all datasets are limited simulation data to some extent, since we always require higher accuracy.

- The information is incomplete

A dataset must contain information about the mapping between design solutions and its performances. If a dataset contains a substantial number of data records that cover the entire design space, then it contains almost all information about the mapping. In contrast, if a dataset contains data records that only cover a portion of the design space, then it contains incomplete information about the mapping. When the amount of data is limited, the information contained is incomplete. When the amount of data is large, but the information is repeated, it will also lead to incomplete information.

In this work, we assume a zero-mean Gaussian process prior

$$p(f|X) = N(f|0, K(X, X)) \quad (2)$$

where f is a vector of the real performances of a solution X .

Limited simulation data can be regarded as noisy observations y of the real values f of the performance because a simulation model will never accurately model all the details. The relationships between f and y follows a Gaussian distribution as shown in Eq. (3).





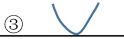
$$p(y|f) = N(y|f, \sigma^2 I) \quad (3)$$

where σ is the variance of noise, I is the identity matrix, and $y = f + \vartheta$, $\vartheta \sim N(0, \sigma^2)$.

(2) Shape engineering knowledge

Engineering knowledge plays a critical role in building surrogate models, especially when only limited simulation data are available. The knowledge is assumed to be correct in this paper, which is summarized by designers in long-term working. The definition of shape knowledge in this

Table 1 The definition of monotonicity and shape engineering knowledge

Type	Monotonicity	Shape
Formula	$y = a \times (x - b)^2 + c$ $x \in [A, B]$ $A > \text{or } B < b$	$x \in [A_1, B_1] \cup [A_2, B_2] \dots \cup [A_n, B_n]$ ($n \geq 1$) For p ($1 \leq p \leq n$) intervals, x conforms to monotonicity formula
Description	① IF x INCREASES THEN y INCREASES ② IF x INCREASES THEN y DECREASES	① IF x INCREASES THEN y INCREASES in the domain A_2 and B_2 ; Otherwise, IF x INCREASES THEN y DECREASES ② IF x INCREASES THEN y INCREASES in the domain A_3 and B_3 ; IF x INCREASES THEN y DECREASES in the domain A_1 and B_1 ③ IF x INCREASES THEN y INCREASES in the domain A_2 and B_2 ; IF x INCREASES THEN y DECREASES in the domain A_1 and B_1
Graph	①  ② 	①  ②  ③ 

paper is the combination of monotonicity knowledge, which is monotonic in one or more piecewise domains. Monotonicity knowledge means that $f(x)$ is monotonically decreasing or increasing against x over a specific range (Sill 1998). Table 1 shows more details about the monotonicity and shape engineering knowledge in formula, description and graph. In formula, we can see that monotonicity is a special case of shape when $n = 1$ and x conforms to monotonicity formula. In this paper, knowledge can be described in the form of rule. All the descriptions of monotonicity knowledge are listed in Table 1. Due to the diversity of shape engineering knowledge, we only list some of them.

In our experience we have found that shape engineering knowledge are more common than other kinds in the domain of engineering design. Also, it is easy for designers to give this type of knowledge articulately. Thus, this paper focuses on shape engineering knowledge.

Framework of BO-DK4DO

BO-DK4DO is used to build high accuracy surrogate models and find the optimal solution when only limited simulation data are available by integrating engineering knowledge. The underlying idea of this method is integration while updating limited simulation data are modeled by a Gaussian process and used as prior process, while the shape engineering knowledge by another Gaussian process and uses it as a likelihood process. To calculate the posterior process, the information from both of the prior process and likelihood process are integrated through Bayes rule. The updating is used to guide the process of collecting new data points automatically, which

means the posterior process determines where to sample a new data point. The whole framework is shown in Fig. 1.

As shown in Fig. 1, the following steps are contained in the proposed method.

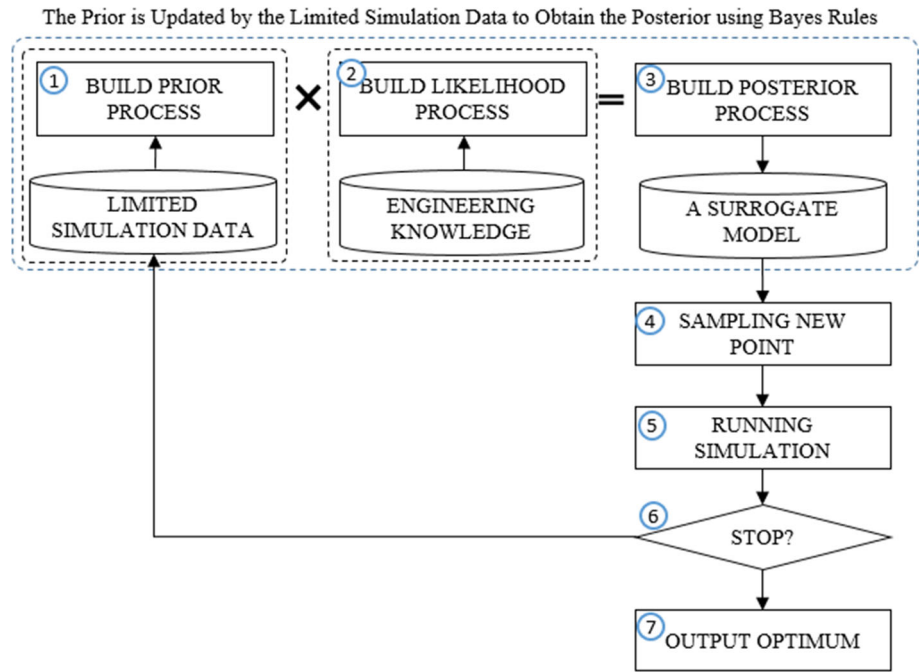
- Step 1:** Modeling limited simulation data by a Gaussian process;
- Step 2:** Modeling engineering knowledge by a Gaussian process;
- Step 3:** Calculating a posterior Gaussian process using Bayes Rule;
- Step 4:** Sampling new data from the posterior Gaussian process;
- Step 5:** Run computer simulations to obtain the performance of the newly sampled data;
- Step 6:** If conditions (number of iterations or required maximum value) are meet, go to Step 7 else go to Step 1 for next iteration;
- Step 7:** Obtain the optimal value based on the posterior Gaussian process.

The first three steps are to integrate limited simulation data and engineering knowledge and they are illustrated in “[Integrating knowledge and data \(integration\)](#)” section. The step 4 is the updating process, and it is illustrated in “[Sampling new data point \(updating\)](#)” section.

Integrating knowledge and data (integration)

To model engineering knowledge and limited simulation data, we first assume a zero-mean Gaussian process (prior

Fig. 1 The general framework of the proposed method (BO-DK4DO)



GP) $\mathcal{GP}(0, \mathbf{K}_{prior})$ and choose a squared exponential (SE) function as the covariance function [Eq. (4)].

$$\mathbf{K}_{prior}(\mathbf{x}^i, \mathbf{x}^j) = \varepsilon e^{-\frac{\|\mathbf{x}^i - \mathbf{x}^j\|^2}{2l^2}} \tag{4}$$

where ε controls the output variance. l is length scale which is a free parameter to control the strength of correlation.

Essentially, in engineering design, data and knowledge can be regarded as the mapping relationships between design variables and performances. Therefore, surrogate model is used to reflect this mapping information. Besides, in engineering design, the obtained data often contain noise and uncertainty. To deal with uncertainty, Gaussian distribution is commonly used in engineering background because of simplicity. A Gaussian Process is a collection of random variables, any finite number of these random variables have joint Gaussian distribution (Rasmussen et al. 2006). In complex engineering design, GP do a better job of accurately representing the portion of the design space that is of interest to the engineer (Wang et al. 2005). Therefore, we use Gaussian process to model data and knowledge in this work. Besides, the mean of the GP does not influence the learning process, therefore, we set the mean to 0.

Based on the above assumption, single monotonicity engineering knowledge is first integrated into the prior GP. Because the monotonicity engineering knowledge can be expressed by partial derivatives easily, we build a joint Gaussian distribution to model the relationship between the function values and the partial derivatives using the following steps.

Step 1: a finite number N of locations X are sampled, and the observation values and function values are indicated by y and f respectively. $D = [X, y]$ indicates the limited simulation data.

All function values f follows Gaussian process (GP) $P(f) = \mathcal{GP}(f, \mathbf{K}_{X,X})$ and the $\mathbf{K}_{X,X}$ can be calculated using Eq. (4). With this model, the function value f^0 of a new point \mathbf{x}^0 can be calculated using Eqs. (5) and (6) based on marginal distribution when only X, y are known. However, these two equations do not contain the information from monotonicity engineering knowledge.

$$E[f^0 | \mathbf{x}^0, X, y] = \mathbf{K}(\mathbf{x}^0, X) (\mathbf{K}(X, X) + \sigma^2 \mathbf{I})^{-1} y \tag{5}$$

$$\begin{aligned} Var[f^0 | \mathbf{x}^0, X, y] &= \mathbf{K}(\mathbf{x}^0, \mathbf{x}^0) - \mathbf{K}(\mathbf{x}^0, X) \\ &\quad \times (\mathbf{K}(X, X) + \sigma^2 \mathbf{I})^{-1} \mathbf{K}(X, \mathbf{x}^0) \end{aligned} \tag{6}$$

Step 2: a finite number M of locations, \bar{X} , where the function values are samples which are known to be monotonic against the d^{th} variable, and the values and signs of the partial derivatives are indicated by f' and m respectively.

To express the monotonicity, the probit likelihood $P(m|f')$ is assumed to follow a distribution as shown in Eqs. (7) and (8) based on existing works (Riihimäki and Vehtari 2010).

$$P(m|f') = \prod_{i=1}^M \Phi\left(\frac{f'_i}{v}\right) \tag{7}$$

$$\Phi(x) = \int_{-\infty}^x N(t|0, 1)dt \tag{8}$$

where v controls the strictness of the monotonicity engineering knowledge and it was fixed to $v = 10^{-2}$ in all experiments;

Step 3: the f and f' are also assumed to follow a joint GP $P(f, f') = \mathcal{GP}(\boldsymbol{\mu}_{joint}, \mathbf{K}_{joint})$, where

$$\boldsymbol{\mu}_{joint} = \begin{bmatrix} f \\ f' \end{bmatrix}, \mathbf{K}_{joint} = \begin{bmatrix} \mathbf{K}_{X,X} & \mathbf{K}_{X,\bar{X}} \\ \mathbf{K}_{\bar{X},X} & \mathbf{K}_{\bar{X},\bar{X}} \end{bmatrix} \tag{9}$$

According to Bayes rule, this joint GP can be updated using Eq. (10) to derive the joint posterior GP when some observation values \mathbf{y} and the signs of partial derivatives \mathbf{m} are obtained.

$$\begin{aligned} P(f, f'|y, m) &= \frac{P(y, m|f, f')P(f, f')}{P(y, m)} \\ &= \frac{P(y|f)P(m|f')P(f, f')}{P(y, m)} \\ &= \frac{1}{Z}P(f, f')P(y|f)P(m|f') \end{aligned} \tag{10}$$

where Z is a normalization term, $P(f, f')$ is the joint Gaussian distribution, $P(y|f)$ is the observation model introduced in “[Limited simulation data and engineering knowledge](#)” section [Eq. (3)].

Since $P(m|f')$ is not a GP, which makes the calculation of the posterior difficult. Therefore, this work adopts an expectation propagation algorithm to calculate the posterior.

The EP algorithm approximates the posterior distribution in Eq. (10) with

$$\begin{aligned} q(f, f'|y, m) &= \frac{1}{Z_{EP}}P(f, f'|X, \bar{X})P(y|f) \\ &\times \prod_{i=1}^M t_i(\bar{Z}_i, \bar{\mu}_i, \bar{\sigma}_i^2) \end{aligned} \tag{11}$$

where $t_i(\bar{Z}_i, \bar{\mu}_i, \bar{\sigma}_i^2) = \bar{Z}_i N(f'_i|\bar{\mu}_i, \bar{\sigma}_i^2)$ are local likelihood approximations with site parameters $\bar{Z}_i, \bar{\mu}_i$ and $\bar{\sigma}_i^2$. The posterior is a product of Gaussian distributions, and can be simplified to

$$q(f, f'|y, m) = N([X, \bar{X}]|\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}}) \tag{12}$$

The posterior mean is $\hat{\boldsymbol{\mu}} = \hat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}$ and the covariance $\hat{\boldsymbol{\Sigma}} = (\mathbf{K}_{joint}^{-1} + \boldsymbol{\Sigma}^{-1})^{-1}$, where

$$\boldsymbol{\mu} = \begin{bmatrix} \mathbf{y} \\ \bar{\boldsymbol{\mu}} \end{bmatrix}, \boldsymbol{\Sigma} = \begin{bmatrix} \sigma^2\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \bar{\boldsymbol{\Sigma}} \end{bmatrix} \tag{13}$$

$\bar{\boldsymbol{\mu}}$ is the vector of site means $\bar{\mu}_i$, and $\bar{\boldsymbol{\Sigma}}$ is a diagonal matrix with site variances $\bar{\sigma}_i^2$ on the diagonal.

With the above model, the function value f^0 of a new point \mathbf{x}^0 can be calculated by Eq. (14) and (15) based on the marginal distribution when $X, \mathbf{y}, \bar{X}, \mathbf{m}$ are given.

$$E[f^0|\mathbf{x}^0, X, \mathbf{y}, \bar{X}, \mathbf{m}] = \mathbf{K}(\mathbf{x}^0, [X, \bar{X}]) (\mathbf{K}_{joint} + \sigma^2\mathbf{I})^{-1} \boldsymbol{\mu} \tag{14}$$

$$\begin{aligned} Var[f^0|\mathbf{x}^0, X, \mathbf{y}, \bar{X}, \mathbf{m}] &= \mathbf{K}(\mathbf{x}^0, \mathbf{x}^0) - \mathbf{K}(\mathbf{x}^0, [X, \bar{X}]) \\ &\quad (\mathbf{K}_{joint} + \sigma^2\mathbf{I})^{-1} \mathbf{K}([X, \bar{X}], \mathbf{x}^0) \end{aligned} \tag{15}$$

With the above methods, monotonicity engineering knowledge is represented as derivatives signs and can be integrated with limited simulation data. However, to integrate shape engineering knowledge, two issues should be addressed further. The first is how to represent shape engineering knowledge while the second is what mathematical operations can be adopted for integrating this knowledge. For the first issue, we divided the original domain of shape engineering knowledge into piecewise domains and in one or more domains the shape engineering knowledge is monotonic. Based on that, we randomly sample some data points from the domain where we know the latent function are monotonic for different variables. Each point is expressed as $[x, +d]$, where d represents the d th variable and the $+/-$ indicates the latent function is positive or negative with the d th variable. For the second issue, we revise Eqs. (7)–(16) which takes the variables into consideration.

$$P(m|f') = \prod_{i=1}^M \Phi\left(\frac{\partial f_i}{\partial x_d} \frac{1}{v}\right) \tag{16}$$

Sampling new data point (updating)

The joint posterior GP shown in Eq. (10) is a surrogate model. With this surrogate model, the value of each point becomes a Gaussian distribution, and the mean and variance can be calculated using Eqs. (14) and (15). Therefore, we need to balance the mean and variance to find the optimal value for the next iteration. The mean μ indicates the value that the objective function is expected to be, while variance σ represents the uncertainty of the value. The value of this point may fall in the interval $(\mu - \sigma, \mu + \sigma)$. When looking for the maximum value, the latent value of a point with a small mean (Point B in Fig. 2) but a high uncertainty may be larger than the point (Point A in Fig. 2) with a larger mean but low uncertainty (Fig. 2). If we only consider the mean and select A, we will lose the chance of obtaining a higher value by using a point B.

To address this issue, an acquisition function is adopted which can balance the mean and variance. The new sam-

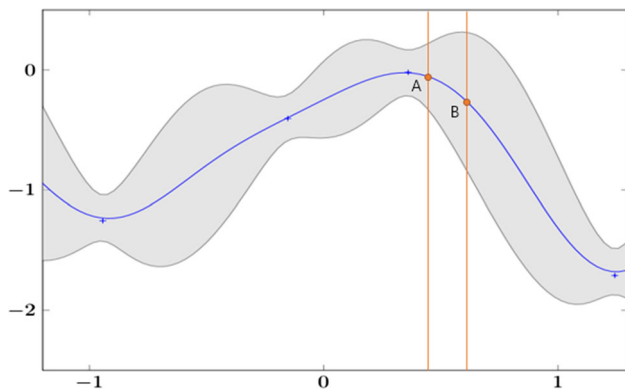


Fig. 2 One dimensional Gaussian process inference of the posterior mean (blue line) and posterior deviation (half of the height of the gray envelope). The latent value of point B may be higher than point A (Color figure online)

point x_{n+1} can be determined by maximizing the acquisition function as shown in Eq. (17)

$$x_{n+1} = \operatorname{argmax} \alpha(x|D_n) \quad (17)$$

where D_n represent the limited simulation data at the n th iteration and α indicates the acquisition function. As shown in Fig. 1, after x_{n+1} is determined and the simulation data is obtained, the data will be further added into the limited simulation data to form a new dataset D_{n+1} for the next iteration. Therefore, we find that the acquisition function plays a very critical role in the updating process. Currently, several different acquisition functions are proposed and selected, such as the probability of improvement (PI) (Kushner 1964), Expected Improvement (EI) (Calvin et al. 2012; Srinivas et al. 2010), Gaussian process upper confidence bound (GP-UCB) (Srinivas et al. 2010), Thompson sampling (Russo and Van Roy 2014) and entropy search (Hennig and Schuler 2012). Considering the cost and time of objective function evaluation, we would like to make the convergence rate of the algorithm fast, which means to get close to the optimum as quickly as possible. Therefore, we use GP-UCB as the acquisition function in this work, Eq. (18).

$$\alpha_n(x) = \mu_{n-1}(x) + \sqrt{\beta_n \sigma_{n-1}(x)} \quad (18)$$

where β_n denotes a positive trade-off parameter and was fixed to 0.1 in this paper. $\mu_{n-1}(x)$ is the mean and $\sigma_{n-1}(x)$ is the variance.

This function has been studied extensively in the literature (Schneider 2015; Smola 2012). It was first proposed and analyzed by Srinivas et al. (2010) in the noisy setting and extended to the noiseless case by Smola (2012). Srinivas et al. (2010) found that GP-UCB acquisition function is currently known to has the fastest convergence rate for GP global optimization. Srinivas theoretically proved

that (1) the asymptotic property of BO with GP-UCB is to be no-regret ($\lim_{T \rightarrow \infty} \frac{R_T}{T} = 0$) and (2) the cumulative regret R_T increases sub-linearly with the growth of T rounds ($R_T = \sum_{t=1}^T r_t$). This work provides the theoretical foundation why we use this acquisition function.

Experiments with BO-DK4DO

The proposed method is verified by two groups of experiments: including benchmark functions and an engineering problem.

Benchmark functions

In this paper, six commonly used single-objective benchmark functions¹ are adopted, and they are (1) Matyas, (2) Rastrigin, (3) Sphere (5 dimensions), (4) Styblinski-Tang (5 dimensions), (5) Sphere (7 dimensions), (6) Styblinski-Tang (7 dimensions). These functions are named B1, B2, B3, B4, B5 and B6 in the following sections, and all these functions have a maximum value. The limited simulation data is sampled by Latin hypercube sampling from these functions and the engineering knowledge is obtained by analyzing these functions, as shown in Table 2. In this table, the optimum is the extreme value of the corresponding function while samples are the initial number of data points used to build a surrogate model.

The proposed method (BO-DK4DO) is compared with general BO (GBO) which does not integrate engineering knowledge. Each experiment is run 20 times and all empirical results are the average value of the 20 runs. The empirical results are shown in “Benchmark functions” section.

Engineering optimization problem

As an engineering material, steel is very popular in industry. The material processing process is very complex due to many variables and constraints must be considered. To meet the desired properties, many trials are required to get an optimal solution, which is an expensive and time-consuming process. Luckily, designers have accumulated some pieces of shape knowledge through previous trials. Typically, our method is useful for dealing with such problem. Figure 3 shows the hot rod rolling (HRR) process for creating a round rod from a lab of steel, which forms the input material for automotive steel gear production. The final properties of steel depend heavily on the microstructure generated after cooling.

Tensile strength (TS) is a key property of steel, which indicates the resistance to break under the tensile load. In this

¹ https://en.wikipedia.org/wiki/Test_functions_for_optimization.

Table 2 The benchmark functions and the engineering knowledge

	Domain	Optimum	Samples	Engineering knowledge
B1	$x_1 \in [-3, 3]$ $x_2 \in [-5, 0]$	16.04	3	① WHEN x_2 is EQUAL to 0, IF x_1 INCREASES THEN y INCREASES in the domain 0 and 3; IF x_1 INCREASES THEN y DECREASES in the domain -3 and 0 ② IF x_2 INCREASES THEN y DECREASES WHEN x_1 is between 0 and 3
B2	$x_1, x_2 \in [-4, 6]$	40.50	3	① IF x_1 INCREASES THEN y INCREASES in the domain 0 and 5; Otherwise, IF x_1 INCREASES THEN y DECREASES ② IF x_2 INCREASES THEN y INCREASES in the domain 0 and 5; Otherwise, IF x_2 INCREASES THEN y DECREASE
B3	$x_1 \sim x_5 \in [0, 10]$	25.00	6	① IF x_1 INCREASES THEN y INCREASES in the domain 5 and 10; IF x_1 INCREASES THEN y DECREASES in the domain 0 and 5 ② IF x_1 INCREASES THEN y INCREASES in the domain 5 and 10; IF x_1 INCREASES THEN y DECREASES in the domain 0 and 5 ③ IF x_1 INCREASES THEN y INCREASES in the domain 5 and 10; IF x_1 INCREASES THEN y DECREASES in the domain 0 and 5
B4	$x_1 \sim x_5 \in [-5, 5]$	12.50	6	① IF x_1 INCREASES THEN y INCREASES in the domain 3 and 5; IF x_1 INCREASES THEN y DECREASES in the domain -5 and -3 ② IF x_1 INCREASES THEN y INCREASES in the domain 3 and 5; IF x_1 INCREASES THEN y DECREASES in the domain -5 and -3 ③ IF x_1 INCREASES THEN y INCREASES in the domain 3 and 5; IF x_1 INCREASES THEN y DECREASES in the domain -5 and -3
B5	$x_1 \sim x_7 \in [0, 10]$	35	8	① IF x_1 INCREASES THEN y INCREASES in the domain 5 and 10; IF x_1 INCREASES THEN y DECREASES in the domain 0 and 5 ② IF x_1 INCREASES THEN y INCREASES in the domain 5 and 10; IF x_1 INCREASES THEN y DECREASES in the domain 0 and 5 ③ IF x_1 INCREASES THEN y INCREASES in the domain 5 and 10; IF x_1 INCREASES THEN y DECREASES in the domain 0 and 5

Table 2 continued

	Domain	Optimum	Samples	Engineering knowledge
B6	$x_1 \sim x_7 \in [-5, 5]$	17.5	8	① IF x_1 INCREASES THEN y INCREASES in the domain 3 and 5; IF x_1 INCREASES THEN y DECREASES in the domain - 5 and - 3 ② IF x_1 INCREASES THEN y INCREASES in the domain 3 and 5; IF x_1 INCREASES THEN y DECREASES in the domain - 5 and - 3 ③ IF x_1 INCREASES THEN y INCREASES in the domain 3 and 5; IF x_1 INCREASES THEN y DECREASES in the domain - 5 and - 3

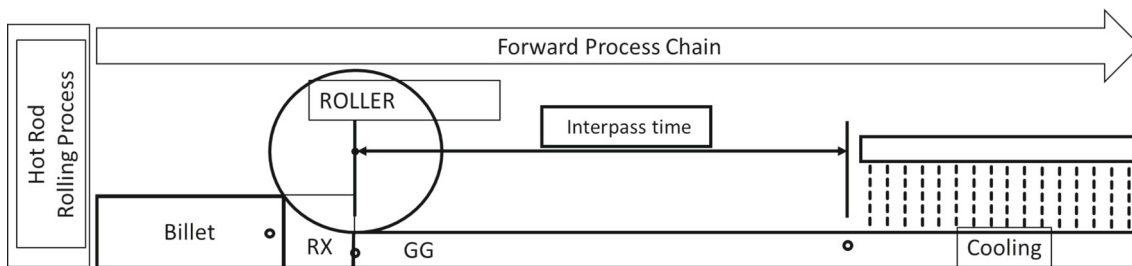


Fig. 3 Hot rod rolling (HRR) process chain

Table 3 The variables and performance for HRR

Type	Name	Range	Symbol
Design variable	The composition of Si	0.18–0.3%	x_1
	The composition of N	0.007–0.009%	x_2
	Ferrite grain size (D_α)	8–25 μm	x_3
	The pearlite interlamellar spacing (S_0)	0.15–0.25 μm	x_4
	The phase fractions of ferrite (X_f)	0.1–0.9	x_5
Performance	Tensile strength (TS)		y_1

Table 4 The engineering knowledge of HRR

ID	Knowledge
1	IF x_1 INCREASES THEN y_1 INCREASES in the domain 0.23 and 0.3; IF x_1 INCREASES THEN y_1 DECREASES in the domain 0.18 and 0.23
2	IF x_2 INCREASES THEN y_1 INCREASES
3	IF x_3 INCREASES THEN y_1 INCREASES in the domain 19 and 25 IF x_3 INCREASES THEN y_1 DECREASES in the domain 8 and 16
4	IF x_4 INCREASES THEN y_1 INCREASES
5	IF x_5 INCREASES THEN y_1 DECREASES

experiment, we try to build a surrogate model which learns the mapping between TS and a group of variables, including the composition of Si ([Si]), the composition of N ([N]), ferrite grain size (D_α), the pearlite interlamellar spacing (S_0) and the phase fractions of ferrite (X_f). Table 3 shows the details of the variables and performance while Table 4 shows the related engineering knowledge. To verify the proposed method, the HRR problem is run for 20 times. For each run, we first collect a dataset with 6 data points and the method iterate 11 times to find an optimal solution.

Results and discussion

Benchmark functions

The empirical results of benchmark functions are listed in Table 5. In this table, the iteration indicates the number of iterations (Step 1 to Step 6 shown in Fig. 1) the algorithms run, and the corresponding empirical results are obtained at this stage. This paper compares the proposed BO-DK4DO and general Bayesian optimization (GBO), and in this table “BO-DK4DO-N” uses N pieces of engineering knowledge which are integrated with limited simulation data.

Table 5 The empirical results of benchmark functions

	Optimum	Iteration	Method	Rank-AM	AM	Rank-SSE	SSE
B1	16.04	4	GBO	3	14.35	3	0.39
			BO-DK4DO-1	2	15.28	2	0.30
			BO-DK4DO-2	1	15.93	1	0.08
		8	GBO	1	16.04	1	0.00
			BO-DK4DO-1	1	16.04	1	0.00
			BO-DK4DO-2	1	16.04	1	0.00
B2	40.50	4	GBO	3	34.25	3	1.18
			BO-DK4DO-1	2	38.66	2	0.26
			BO-DK4DO-2	1	39.91	1	0.15
		8	GBO	3	39.05	3	0.29
			BO-DK4DO-1	2	39.70	2	0.23
			BO-DK4DO-2	1	40.30	1	0.04
B3	25.00	5	GBO	4	16.78	4	1.02
			BO-DK4DO-1	3	24.47	3	0.25
			BO-DK4DO-2	1	25.00	1	0.00
		10	BO-DK4DO-3	1	25.00	1	0.00
			GBO	4	23.00	4	0.64
			BO-DK4DO-1	3	24.79	3	0.12
		15	BO-DK4DO-2	1	25.00	1	0.00
			BO-DK4DO-3	1	25.00	1	0.00
			GBO	1	25.00	1	0.00
			BO-DK4DO-1	4	24.91	4	0.06
			BO-DK4DO-2	1	25.00	1	0.00
			BO-DK4DO-3	1	25.00	1	0.00
B4	12.50	5	GBO	4	2.05	3	0.42
			BO-DK4DO-1	3	9.40	4	0.44
			BO-DK4DO-2	2	10.80	2	0.19
		10	BO-DK4DO-3	1	11.24	1	0.10
			GBO	4	3.30	4	0.66
			BO-DK4DO-1	3	10.73	3	0.25
		15	BO-DK4DO-2	1	11.53	1	0.08
			BO-DK4DO-3	2	11.52	1	0.08
			GBO	4	4.65	4	0.96
			BO-DK4DO-1	3	11.58	3	0.14
			BO-DK4DO-2	2	11.76	2	0.08
			BO-DK4DO-3	1	11.81	1	0.07
B5	35	6	GBO	4	28.00	4	1.50
			BO-DK4DO-1	3	34.56	3	0.04
			BO-DK4DO-2	2	34.87	2	0.01
		12	BO-DK4DO-3	1	34.95	1	0.00
			GBO	4	33.57	4	0.65
			BO-DK4DO-1	3	34.71	3	0.03
		18	BO-DK4DO-2	2	34.94	1	0.00
			BO-DK4DO-3	1	34.96	1	0.00
			GBO	3	34.88	3	0.02
			BO-DK4DO-1	4	34.74	4	0.03
			BO-DK4DO-2	2	34.96	1	0.00
			BO-DK4DO-3	1	34.98	1	0.00

Table 5 continued

	Optimum	Iteration	Method	Rank-AM	AM	Rank-SSE	SSE	
B6	17.5	6	GBO	4	1.21	3	0.30	
			BO-DK4DO-1	3	9.21	4	1.33	
			BO-DK4DO-2	2	14.67	2	0.19	
			BO-DK4DO-3	1	15.30	1	0.11	
		12		GBO	4	3.23	4	0.98
				BO-DK4DO-1	3	12.65	3	0.97
				BO-DK4DO-2	2	15.05	2	0.19
				BO-DK4DO-3	1	15.58	1	0.10
		18		GBO	4	5.33	4	1.31
				BO-DK4DO-1	3	13.15	3	0.91
				BO-DK4DO-2	2	15.35	2	0.20
				BO-DK4DO-3	1	15.90	1	0.09

The bold means where the performance of GBO is better than that of BO-DK4DO-1

After N iterations, the methods can find the maximum value. This paper adopts the following six metrics to measure the performance. All the experiments are run on ThinkPad E470c with Intel(R) Core(TM) i5-6200U CPU @ 2.3 GHz 2.4 GHz and 8 GB RAM.

- (1) The average means (AM) indicates the average current maximum value of the 20 runs of the experiments for a certain iteration. A higher AM value indicates a greater rate of convergence to the real maximum.

$$AM_i = \frac{1}{n} \sum_{p=1}^n x_{pi} \tag{19}$$

where n is the times of experiments. x_{pi} represents the value of the i th iteration in the p th experiment. Accordingly, AM_i is the mean of i th iteration.

- (2) The Rank-AM is the order of these methods in terms of the AM value. If the order is 1, the corresponding method has the maximal AM.
- (3) The scaled standard error (SSE) measures the stability of the 20 maximums found after N running. A lower SSE implies a higher stability and robust capability to find the real maximum.

$$SSE_i = \left(\frac{\frac{1}{n} \sum_{p=1}^n (x_{pi} - AM_i)^2}{n} \right)^{\frac{1}{2}} \tag{20}$$

Similarly, SSE_i is the standard error of i th iteration. The numerator is the standard error formula, which we scaled for the showing on the figure.

- (4) The Rank- SSE is the order of these methods in terms of the SSE value. If the order is 1, the corresponding method has the minimal SSE.

- (5) The relative difference (RD) is the proximity of real maximum and AM found at certain iteration. The smaller the value, the closer of real maximum and AM. The real maximum is titled as “optimum” in Table 5.

$$RD = \frac{\text{optimum} - AM}{\text{optimum}} \tag{21}$$

- (6) The relative time (RT) indicator calculated by Eq. (22).

$$RT = \frac{T_1 - T_2}{T_2} \tag{22}$$

Where T_1 and T_2 represent the time of two methods, respectively.

From Table 5, several results can be seen. The first and most important finding is that the integration of engineering knowledge (no matter single or multiple pieces of engineering knowledge) with limited simulation data brings a higher AM and a lower SSE. This implies that the integration of engineering knowledge tends to accelerate the speed of convergence with higher robustness.

The second finding is that when multiple pieces of engineering knowledge are integrated, the AM is further increased while the SSE is decreased. We can see from Table 5 that at last iteration of BO-DK4DO-3 has the maximal AM and minimal SSE, which implies it’s the best method. BO-DK4DO-2 has a relatively small AM and bigger SSE, but the performance is obviously better than BO-DK4DO-1 and GBO.

The third finding is that although in many situations the performance of BO-DK4DO-1 is better than that of GBO, there are indeed some situations where GBO outperformed BO-DK4DO-1, which have an outline border in Table 5. From the corresponding values of AM and SSE, we find the differences are very small. This finding again tells us that it is

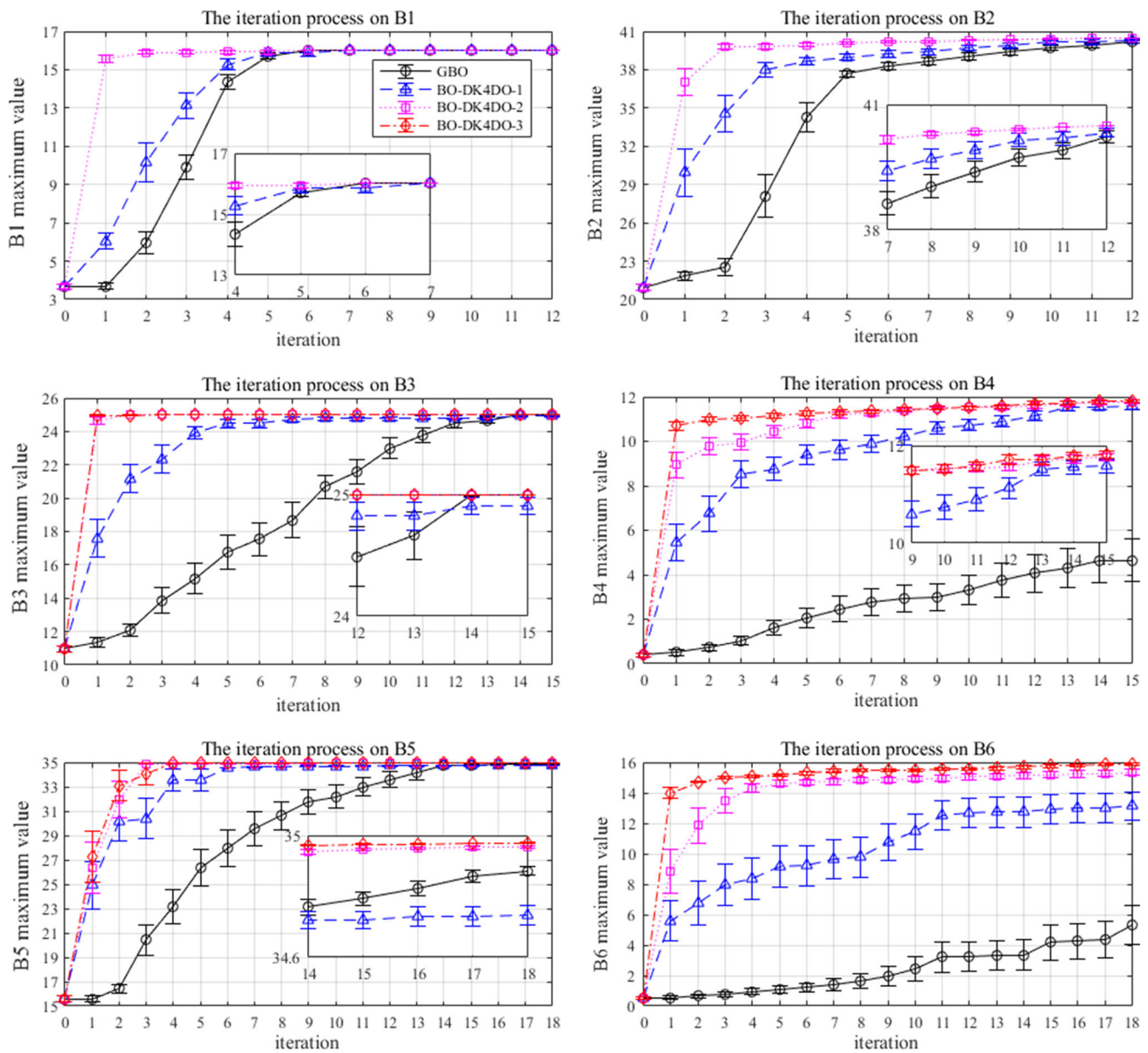


Fig. 4 The iteration process of GBO and BO-DK4DO on benchmark functions

better to integrate multiple pieces of engineering knowledge with limited simulation data to obtain an obvious improvement in terms of the performance.

Figure 4 shows the AM and SSE for every iteration, from which we obtain another observation. This observation can be expressed as the most obvious difference between GBO and BO-DK4DO occurs in the first few iterations, where BO-DK4DO has obvious advantages. When many iterations (more than 10) are finished, the performances of BO-DK4DO and GBO tend to become closer. Because most computer simulations are expensive and time-consuming, the performance in the first few iterations influences the applicability of the

methods significantly. From this point of view, the proposed BO-DK4DO is far better than GBO.

To further discuss the efficiency of our method quantitatively, we use RD to compare BO-DK4DO-N with GBO shown in Table 6. Meanwhile, due to the time required for each iteration is almost the same, we show and compare the total time of a run in Table 7. The RT shown in the table is also the average RT of each iteration.

Table 6 indicates that BO-DK4DO has a better performance in terms of convergence when the initial samples are relatively small (3, 6 and 8 shown in Table 2). As the complexity of the problem increases, BO-DK4DO becomes more competitive in terms of RD. The RD of GBO can be reduced

Table 6 Comparative results between GBO and BO-DK4DO-N of benchmark functions

	Optimum	Iteration	Method	AM	RD (%)	
B1	16.04	4	GBO	14.35	10.56	
			BO-DK4DO-1	15.28	4.74	
			BO-DK4DO-2	15.93	0.69	
		8	GBO	16.04	0	
			BO-DK4DO-1	16.04	0	
			BO-DK4DO-2	16.04	0	
	B2	40.50	4	GBO	34.25	15.42
				BO-DK4DO-1	38.66	4.54
				BO-DK4DO-2	39.91	1.45
		8	GBO	39.05	3.60	
			BO-DK4DO-1	39.70	1.97	
			BO-DK4DO-2	40.30	0.50	
B3		25.00	5	GBO	16.78	32.87
				BO-DK4DO-1	24.47	2.12
				BO-DK4DO-2	25.00	0.01
		10	GBO	23.00	8.03	
			BO-DK4DO-1	24.79	0.86	
			BO-DK4DO-2	25.00	0.01	
		15	GBO	25.00	0.02	
			BO-DK4DO-1	24.91	0.37	
			BO-DK4DO-2	25.00	0	
B4	12.50	5	GBO	2.05	83.59	
			BO-DK4DO-1	9.40	24.80	
			BO-DK4DO-2	10.80	13.63	
		10	BO-DK4DO-3	11.24	10.09	
			GBO	3.30	73.58	
			BO-DK4DO-1	10.73	14.17	
		15	BO-DK4DO-2	11.53	7.79	
			BO-DK4DO-3	11.52	7.84	
			GBO	4.65	62.83	
	6	BO-DK4DO-1	11.58	7.38		
		BO-DK4DO-2	11.76	5.92		
		BO-DK4DO-3	11.81	5.51		
B5	35	6	GBO	28.00	20.02	
			BO-DK4DO-1	34.56	1.25	
			BO-DK4DO-2	34.87	0.37	
		12	BO-DK4DO-3	34.95	0.13	
			GBO	33.57	4.10	
			BO-DK4DO-1	34.71	0.82	
			BO-DK4DO-2	34.94	0.18	
			BO-DK4DO-3	34.96	0.10	

Table 6 continued

	Optimum	Iteration	Method	AM	RD (%)
B6	17.5	6	GBO	1.21	93.11
			BO-DK4DO-1	9.21	47.37
			BO-DK4DO-2	14.67	16.13
		12	BO-DK4DO-3	15.30	12.58
			GBO	3.23	81.57
			BO-DK4DO-1	12.65	27.73
		18	BO-DK4DO-2	15.05	13.98
			BO-DK4DO-3	15.58	10.99
			GBO	5.33	69.56
		BO-DK4DO-1	13.15	24.84	
		BO-DK4DO-2	15.35	12.30	
		BO-DK4DO-3	15.90	9.16	

Table 7 Comparison of relative program run time of GBO and BO-DK4DO for benchmark functions

	Method	Total time (s)	RT	
B1	GBO	20.36	0	
	BO-DK4DO-1	74.47	2.66	
	BO-DK4DO-2	83.68	3.11	
B2	GBO	20.13	0	
	BO-DK4DO-1	82.07	3.08	
	BO-DK4DO-2	98.18	3.88	
B3	GBO	44.58	0	
	BO-DK4DO-1	190.46	3.27	
	BO-DK4DO-2	222.27	3.99	
	BO-DK4DO-3	291.30	5.53	
	B4	GBO	42.26	0
	BO-DK4DO-1	163.07	2.86	
	BO-DK4DO-2	182.28	3.31	
	BO-DK4DO-3	203.65	3.82	
	B5	GBO	59.67	0
	BO-DK4DO-1	287.78	3.82	
	BO-DK4DO-2	525.52	7.81	
	BO-DK4DO-3	1051.26	16.62	
B6	GBO	55.80	0	
	BO-DK4DO-1	236.03	3.23	
	BO-DK4DO-2	289.76	4.19	
	BO-DK4DO-3	382.70	5.86	

by increasing the number of iterations. However, it is not guaranteed to achieve the performance of BO-DK4DO.

There are two phenomena need to be addressed. The first is that for B3, the RD of BO-DK4DO-1 is higher than GBO in the last two iterations, with a final difference of 0.35%. However, in the tenth iteration, BO-DK4DO-1 has already reached 0.86%, while GBO is still up to 8.03%. The second is that for B5, the RD of BO-DK4DO-1 is higher than GBO in the last two iterations, with a final difference of 0.42%. However, in the twelfth iteration, BO-DK4DO-1 has already reached 0.82%, while GBO is still up to 4.10%. The above analysis implies that the proposed method is able of reducing the interactions without sacrificing of the solution.

As shown in Table 7, we can see that when more pieces of engineering knowledge are integrated, the computational time also increases. Meanwhile, the convergence speed is also increasing rapidly inferred from Table 6. Take B4 and B6 as examples, GBO's RD on B4 is 83.59% in the 5th iteration and GBO's RD on B6 is 93.11% in the 6th iteration, while our method is only 24.80%, 13.63%, 10.09% for B4 and 47.37%, 16.13%, 12.58% for B6 respectively. Considering here are situations where the time required for a single simulation is often hours, days or even more, our proposed method is effective which is often cheaper than performing additional costly simulations of the unknown function.

For the functions B3 and B5 with errors highlighted in Table 5 Line Rank-AM, correspondingly, compared with B4 and B6 shown in Table 7, the RT increases faster as multiple knowledge integrated. To be noted, in the 5th iteration, for B3 and B5, the RD for GBO is 32.87% and 20.02%, while our method is only 2.12%, 0.01%, 0% for B3 and 1.25%, 0.37%, 0.13% for B5. To meet the same requirements for optimal solution, our method is able to save up many iterations.

As shown in Table 7, For B4 and B6 (Their name is Styblinski-Tang), as the dimension increases, the RT of all methods only increase slightly, which means if we have a big number of design variables, the RT will not increase obvious. Therefore, the proposed method is capable for complex product designs with many variables.

Engineering optimization problems

In this paper, we use BO-DK4DO and GBO to maximize the tensile strength of the rolling steel. Four methods are compared including GBO, BO-DK4DO-1, BO-DK4DO-3 and BO-DK4DO-5. The empirical results are shown in Table 8.

From Table 8, we have similar findings with the benchmark experiments and the control system experiments. The first is that the BO-DK4DO is efficient of integrating engineering knowledge with limited simulation data and this integration is helpful for finding the optimal solution efficiently. The second is that the integration of engineering knowledge decreases the SSE, which implies that it increases

Table 8 The empirical results of HRR

Methods	Iterations	AM	SSE
GBO	5	748.34	2.57
BO-DK4DO-1		723.67	11.13
BO-DK4DO-3		755.72	3.42
BO-DK4DO-5	7	763.73	1.40
GBO		752.70	2.22
BO-DK4DO-1		732.31	9.95
BO-DK4DO-3	11	762.35	2.09
BO-DK4DO-5		767.76	0.97
GBO		760.82	1.72
BO-DK4DO-1		756.09	2.37
BO-DK4DO-3		765.16	1.55
BO-DK4DO-5		770.16	0.00

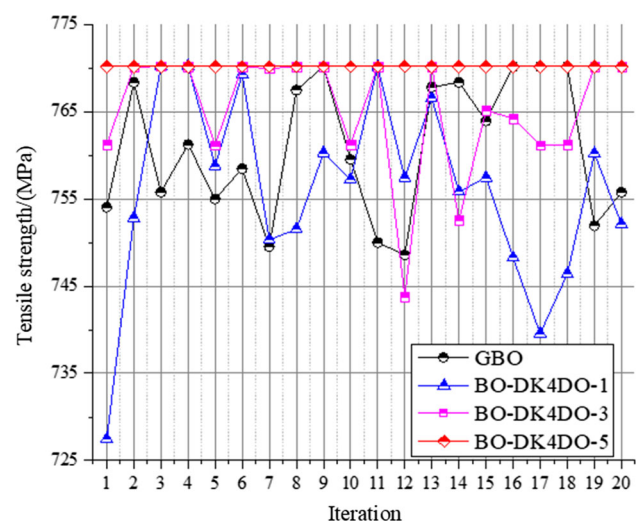


Fig. 5 The predicted values of the last iteration of the 20 runs

the robustness of the methods. In other words, BO-DK4DO is less affected by the distribution of the initial points. It can be seen more clearly in Fig. 5, we take the last iteration of 20 runs as an example.

In this experiment, one thing is different from the benchmark experiments where the performance is increasing as long as engineering knowledge is integrated. When adding the first piece of engineering knowledge, GBO has a better performance compare with BO-DK4DO-1, and this can be seen more specific in Fig. 6. However, when adding more than one (three or five) pieces of engineering knowledge, the method gains obvious advantages compare with GBO. This phenomenon occurs because GP with engineering knowledge tends to favor smooth functions, but the real function is not smooth. Figure 7 shows the range of the function within the domain 0.18 to 0.3, which is also the domain of the first piece of engineering knowledge. The fixed value of x_2 , x_3 , x_4 and x_5 are 0.009, 8, 0.25 and 0.1 respectively.

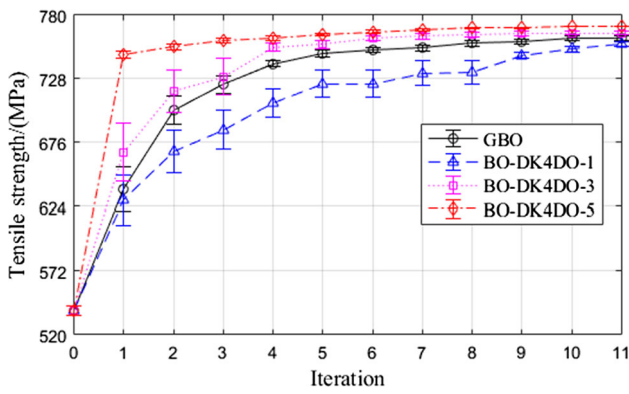


Fig. 6 The iteration process of GBO and BO-DK4DO on HRR

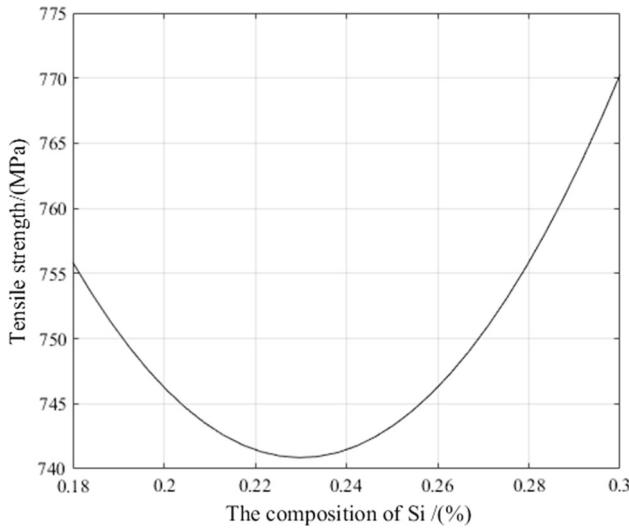


Fig. 7 The relationship between x_1 and y_1 of the HRR

To compare the efficiency of these methods, Table 9 show the average runtime at each iteration. The computational time of each iteration in both methods is nearly the same.

As shown in Table 8, though the latent function of HRR is non-smooth, both BO-DK4DO-3 and BO-DK4DO-5 do better than GBO, with 765.16 MPa and 770.16 MPa in the last iteration, while BO-DK4DO-1 is 756.09 finally. Meanwhile, form Table 10, as the RT of GBO is set to 0, the RT of BO-DK4DO-1, BO-DK4DO-3, BO-DK4DO-5 for the HRR are 2.61, 3.64 and 4.83.

Note that the reduction of the number of experiments is also significant, as shown in Fig. 6 We only show the available value against iteration in this figure, the difference of real cost is much larger. In this case, the target value is over 755 MPa. BO-DK4DO-5 and BO-DK4DO-3 takes 3 and 5 iterations respectively to reach the target while the BO-DK4DO-1 and GBO takes 11 and 8 iterations. To apply the method to real-world problems, the computation time should include not only the program run time which is consumed to find the optimal solution but also the simulation time used for sam-

Table 9 The average program run time at each iteration for the HRR

Iterations	Methods			
	GBO	BO-DK4DO-1	BO-DK4DO-3	BO-DK4DO-5
1	4.78	17.33	22.88	28.78
2	4.60	17.39	22.34	28.14
3	4.80	17.96	22.43	28.14
4	4.62	16.82	22.78	28.72
5	4.86	17.20	22.10	28.07
6	4.49	17.29	22.15	26.78
7	4.62	17.64	22.54	28.15
8	5.33	18.33	22.59	28.67
9	5.04	16.71	22.90	28.51
10	5.19	18.22	22.82	28.45
11	4.95	17.43	22.10	28.32

Table 10 Comparison of relative program run time (11 iterations) for the HRR

Methods	Program run time (s)	RT
GBO	53.29	0
BO-DK4DO-1	192.34	2.61
BO-DK4DO-3	247.63	3.64
BO-DK4DO-5	310.74	4.83

pling. Therefore, the total computation time is defined as follows in this paper.

$$T = T_p + T_s \tag{23}$$

where T_p is the program run time to obtain the target value and T_s is the simulation time.

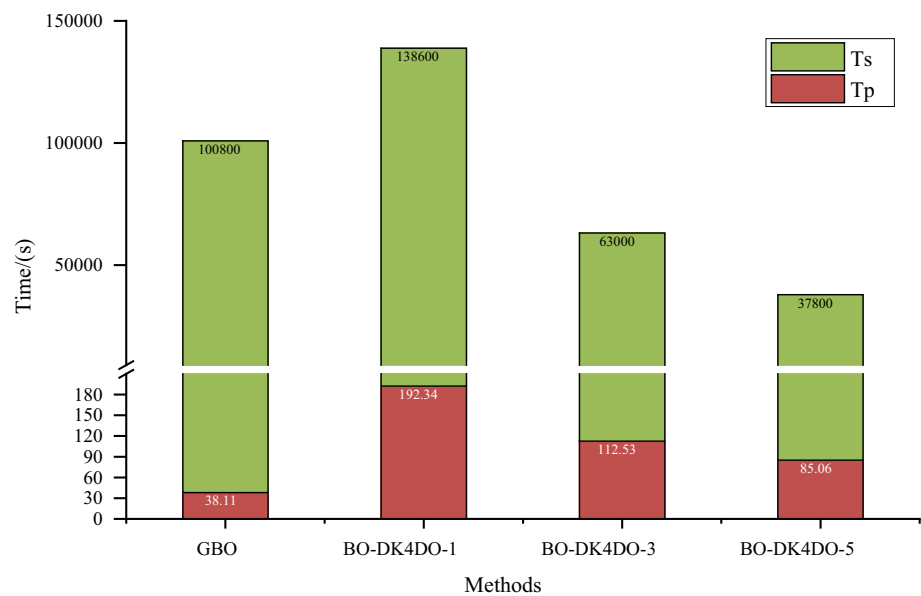
In the HRR problem, the average sampling time for generating a data point in the simulation is about 3.5 h (12600 s), which is at least 2 orders of magnitude larger than the total program run time for all methods.

Figure 8 show the T_p and T_s for all methods. Mapping to the real time, BO-DK4DO-3 and BO-DK4DO-5 are able to save up to 37.41% and 62.43% of total computation time T , which firmly establishes the utility of using multiple engineering knowledge through our proposed method. Actually, we can ignore the T_p in Eq. (23) due to the huge magnitude difference. For many complex engineering problems, the computation time is the equivalent of the simulation time.

Closure

Surrogate models are widely used in simulation-based design, and many engineering problems are solved with

Fig. 8 Comparison of T_p and T_s of GBO and BO-DK4DO for the HRR



the support of surrogate models. However, enough data are required to build a highly accurate surrogate model, which are not available for many real-world engineering problems. Therefore, developing methods for building high-accuracy surrogate models and implement optimization designs are still a challenge. In this paper, we present a method to integrate limited simulation data and shape engineering knowledge to build surrogate models and conduct design optimizations based on Bayesian optimization, which is called BO-DK4DO. The proposed method is verified using 6 benchmark functions and HRR problem. By the analysis of the empirical results, we find (1) the proposed method is capable of integrating shape engineering knowledge and limited simulation data efficiently; (2) the integration of engineering knowledge brings satisfactory improvement in terms of finding optimum values; (3) when more pieces of engineering knowledge are integrated, the performance of the methods are further improved.

By this paper, three-fold contributions are made. The first is that the notion of integration engineering knowledge and available limited data to build surrogate models and implementing design optimization. The notion expands the application of surrogate models to scenarios where only limited simulation data are available. The second is the adoption of Bayesian optimization and a probabilistic model (Gaussian process) to model the engineering knowledge and limited simulation data. This natural adoption of uncertainty of the proposed method provides a new way and tools for further research about the uncertainty of engineering knowledge and limited simulation data. The last contribution is that a computational method to accomplish the above idea is implemented, and the proposed method is demonstrated through 6 benchmark functions and a real-world engineering

design problem. We suggest that this method is foundational for further development.

Acknowledgements The authors would like to thank Professor Janet K. Allen and Professor Farrokh Mistree from the University of Oklahoma for their valuable comments. The author also appreciates the strong support provided by the National Natural Science Foundation of China (NSFC 51505032), the Beijing Natural Science Foundation (BJNSF 3172028).

References

- Abu-Mostafa, Y. S. (1990). Learning from hints in neural networks. *Journal of Complexity*, 6(2), 192–198.
- Aguirre, L. A., & Furtado, E. C. (2007). Building dynamical models from data and prior knowledge: The case of the first period-doubling bifurcation. *Physical Review E*, 76(4), 046219.
- Amsallem, D., Cortial, J., Carlberg, K., & Farhat, C. (2009). A method for interpolating on manifolds structural dynamics reduced-order models. *International Journal for Numerical Methods in Engineering*, 80(9), 1241–1258.
- Andersen, M. R., Siivola, E., & Vehtari, A. (2017). Bayesian Optimization of Unimodal Functions. In *31st Conference on Neural Information Processing Systems (NIPS 2017)*.
- Aughenbaugh, J., & Herrmann, J. (2007). Updating uncertainty assessments: A comparison of statistical approaches. In *ASME 2007 international design engineering technical conferences and computers and information in engineering conference* (pp. 1195–1209). <https://doi.org/10.1115/detc2007-35158>.
- Bhattacharyya, A., Conlan-Smith, C., & James, K. A. (2019). Design of a bi-stable airfoil with tailored snap-through response using topology optimization. *Computer-Aided Design*, 108, 42–55. <https://doi.org/10.1016/j.cad.2018.11.001>.
- Brochu, E., Cora, V. M., & de Freitas, N. (2010). A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *ArXiv*, 49. doi:10.1007/9783642532580?COVERIMAGEURL=HTTPS://STATICCONTENT.SPRINGER.COM/COVER/BOOK/9783642532580.JPG.

- Calandra, R., Seyfarth, A., Peters, J., & Deisenroth, M. P. (2016). Bayesian optimization for learning gaits under uncertainty. *Annals of Mathematics and Artificial Intelligence*, 76(1), 5–23. <https://doi.org/10.1007/s10472-015-9463-9>.
- Calvin, J. M., Chen, Y., & Žilinskas, A. (2012). An adaptive univariate global optimization algorithm and its convergence rate for twice continuously differentiable functions. *Journal of Optimization Theory and Applications*, 155(2), 628–636.
- Daniels, H., & Velikova, M. (2010). Monotone and partially monotone neural networks. *IEEE Transactions on Neural Networks*, 21(6), 906–917.
- Daróczy, L., Janiga, G., & Thévenin, D. (2018). Computational fluid dynamics based shape optimization of airfoil geometry for an H-rotor using a genetic algorithm. *Engineering Optimization*, 50(9), 1483–1499. <https://doi.org/10.1080/0305215X.2017.1409350>.
- Di Bella, A., Fortuna, L., Grazianil, S., Napoli, G., Xibilia, M. G., & Doria, V. A. (2007). Development of a soft sensor for a thermal cracking unit using a small experimental data set. In *2007 IEEE international symposium on intelligent signal processing*.
- Dougherty, E. R., Dalton, L. A., & Alexander, F. J. (2015). Small data is the problem. In *2015 49th Asilomar conference on signals, systems and computers* (pp. 418–422). IEEE.
- Du, T.-S., Ke, X.-T., Liao, J.-G., & Shen, Y.-J. (2018). DSLC-FOA: Improved fruit fly optimization algorithm for application to structural engineering design optimization problems. *Applied Mathematical Modelling*, 55, 314–339. <https://doi.org/10.1016/j.apm.2017.08.013>.
- Du, G., Xia, Y., Jiao, R. J., & Liu, X. (2019). Leader-follower joint optimization problems in product family design. *Journal of Intelligent Manufacturing*, 30(3), 1387–1405. <https://doi.org/10.1007/s10845-017-1332-4>.
- Fatemeh, D. B., Loo, C. K., & Kanagaraj, G. (2019). Shuffled Complex Evolution based Quantum Particle Swarm Optimization algorithm for mechanical design optimization problems. *Journal of Modern Manufacturing Systems and Technology*, 02, 23–32.
- Fengjie, T., & Lahmer, T. (2018). Shape optimization based design of arch-type dams under uncertainties. *Engineering Optimization*, 50(9), 1470–1482. <https://doi.org/10.1080/0305215X.2017.1409348>.
- Forrester, A. I. J., Sbester, A., & Keane, A. J. (2008). *Engineering design via surrogate modelling*. Chichester: Wiley.
- Fortuna, L., Graziani, S., & Xibilia, M. G. (2009). Comparison of soft-sensor design methods for industrial plants using small data sets. *IEEE Transactions on Instrumentation and Measurement*, 58(8), 2444–2451.
- Gorissen, D., & Dhaene, T. (2010). A surrogate modeling and adaptive sampling toolbox for computer based design. *Journal of Machine Learning Research*, 11(1), 2051–2055.
- Gupta, M., Bahri, D., Cotter, A., & Canini, K. (2018). Diminishing returns shape constraints for interpretability and regularization. In *Advances in neural information processing systems* (Vol. 31, pp. 6834–6844). Curran Associates, Inc. <http://papers.nips.cc/paper/7916-diminishing-returns-shape-constraints-for-interpretability-and-regularization.pdf>.
- Hao, J., Ye, W., Wang, G., Jia, L., & Wang, Y. (2018). Evolutionary neural network-based method for constructing surrogate model with small scattered dataset and monotonicity experience. In *Proceedings of the 2018 soft computing & machine intelligence*, Nairobi, Kenya.
- Hennig, P., & Schuler, C. J. (2012). Entropy search for information-efficient global optimization. *The Journal of Machine Learning Research*, 13(1), 1809–1837.
- Jauch, M., & Peña, V. (2016). Bayesian optimization with shape constraints. In *29th conference on neural information processing systems*, Barcelona, Spain.
- Jones, D. R., Schonlau, M., & William, J. (1998). Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4), 455–492.
- Kang, G., Wu, L., Guan, Y., & Peng, Z. (2019). A virtual sample generation method based on differential evolution algorithm for overall trend of small sample data: Used for lithium-ion battery capacity degradation data. *IEEE Access*, 7, 123255–123267. <https://doi.org/10.1109/ACCESS.2019.2937550>.
- Kotlowski, W., & Slowinski, R. (2013). On nonparametric ordinal classification with monotonicity constraints. *IEEE Transactions on Knowledge and Data Engineering*, 25(11), 2576–2589.
- Kushner, H. J. (1964). A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. *Journal of Basic Engineering*, 86(1), 97–106.
- Lenk, P. J., & Choi, T. (2017). Bayesian analysis of shape-restricted functions using Gaussian process priors. *Statistica Sinica*, 27, 43–70.
- Li, D.-C., Chang, C.-C., Liu, C.-W., & Chen, W.-C. (2013). A new approach for manufacturing forecast problems with insufficient data: The case of TFT-LCDs. *Journal of Intelligent Manufacturing*, 24(2), 225–233.
- Li, D.-C., Chen, L.-S., & Lin, Y.-S. (2003). Using functional virtual population as assistance to learn scheduling knowledge in dynamic manufacturing environments. *International Journal of Production Research*, 41(17), 4011–4024.
- Li, D.-C., Chen, H.-Y., & Shi, Q.-S. (2018a). Learning from small datasets containing nominal attributes. *Neurocomputing*, 291, 226–236. <https://doi.org/10.1016/j.neucom.2018.02.069>.
- Li, D.-C., Fang, Y.-H., Liu, C.-W., & Juang, C. (2012). Using past manufacturing experience to assist building the yield forecast model for new manufacturing processes. *Journal of Intelligent Manufacturing*, 21(4), 1–12.
- Li, C., Santu, R., Gupta, S., Nguyen, V., Venkatesh, S., Sutti, A., et al. (2018b). Accelerating experimental design by incorporating experimenter hunches. In *2018 IEEE international conference on data mining (ICDM)* (pp. 257–266). <https://doi.org/10.1109/icdm.2018.00041>.
- Liu, B., Koziel, S., & Zhang, Q. (2016). A multi-fidelity surrogate-model-assisted evolutionary algorithm for computationally expensive optimization problems. *Journal of Computational Science*, 12, 28–37. <https://doi.org/10.1016/j.jocs.2015.11.004>.
- Liu, H., Ong, Y.-S., & Cai, J. (2018). A survey of adaptive sampling for global metamodeling in support of simulation-based complex engineering design. *Structural and Multidisciplinary Optimization*, 57(1), 393–416. <https://doi.org/10.1007/s00158-017-1739-8>.
- Min, A. T. W., Sagarna, R., Gupta, A., Ong, Y., & Goh, C. K. (2017). Knowledge transfer through machine learning in aircraft design. *IEEE Computational Intelligence Magazine*, 12(4), 48–60.
- Monisha, R., & Peter, D. J. (2017). Triple band PIFA antenna using knowledge based neural networks. *Asian Journal of Applied Science and Technology (AJAST)*, 1(3), 271–274.
- Nagarajan, H. P. N., Mokhtarian, H., Jafarian, H., Dimassi, S., Bakrani-Balani, S., Hamed, A., et al. (2018). Knowledge-based design of artificial neural network topology for additive manufacturing process modeling: A new approach and case study for fused deposition modeling. *Journal of Mechanical Design*, 141(2), 1. <https://doi.org/10.1115/1.4042084>.
- Ning, J., Nguyen, V., Huang, Y., Hartwig, K. T., & Liang, S. Y. (2018). Inverse determination of Johnson–Cook model constants of ultra-fine-grained titanium based on chip formation model and iterative gradient search. *The International Journal of Advanced Manufacturing Technology*, 99(5), 1131–1140. <https://doi.org/10.1007/s00170-018-2508-6>.

- Parrado-Hernández, E., Ambroladze, A., Shawe-Taylor, J., & Sun, S. (2012). PAC-bayes bounds with data dependent priors. *Journal of Machine Learning Research*, 13(1), 3507–3531.
- Pratap, S., Daultani, Y., Tiwari, M. K., & Mahanty, B. (2018). Rule based optimization for a bulk handling port operations. *Journal of Intelligent Manufacturing*, 29(2), 287–311. <https://doi.org/10.1007/s10845-015-1108-7>.
- Rasmussen, C. E., & Williams, C. K. I. (2006). *Gaussian processes for machine learning*. Cambridge, Massachusetts, London: The MIT Press.
- Riihimäki, J., & Vehtari, A. (2010). Gaussian processes with monotonicity information. *Journal of Machine Learning Research*, 9, 645–652.
- Russo, D., & Van Roy, B. (2014). Learning to optimize via posterior sampling. *Mathematics of Operations Research*, 39(4), 949–1348.
- Sahnoun, M., Bettayeb, B., Bassetto, S.-J., & Tollenaere, M. (2016). Simulation-based optimization of sampling plans to reduce inspections while mastering the risk exposure in semiconductor manufacturing. *Journal of Intelligent Manufacturing*, 27(6), 1335–1349. <https://doi.org/10.1007/s10845-014-0956-x>.
- Schneider, J. (2015). High dimensional bayesian optimisation and bandits via additive models. In *ICML'15 Proceedings of the 32nd international conference on international conference on machine learning* (Vol. 37, pp. 295–304).
- Sefat, M., Salahshoor, K., Jamialahmadi, M., & Moradi, B. (2012). A new approach for the development of fast-analysis proxies for petroleum reservoir simulation. *Petroleum Science and Technology*, 30(18), 1920–1930. <https://doi.org/10.1080/10916466.2010.512885>.
- Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., & de Freitas, N. (2016). Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1), 148–175.
- Shavlik, J. W. (1994). Combining symbolic and neural learning. *Machine Learning*, 14(3), 321–331. <https://doi.org/10.1007/BF0093982>.
- Sill, J. (1998). Monotonic networks. In *Proceedings of the 1997 conference on advances in neural information processing systems* (Vol. 10, pp. 661–667). Cambridge: MIT Press.
- Sill, J., & Abu-Mostafa, Y. S. (1997). Monotonicity hints. In M. C. Mozer, M. I. Jordan, & T. Petsche (Eds.), *Advances in neural information processing systems* (Vol. 6, pp. 634–640). Cambridge: MIT Press.
- Smola, A. J. (2012). Exponential regret bounds for gaussian process bandits with deterministic observations. In *ICML'12 Proceedings of the 29th international conference on international conference on machine learning* (pp. 955–962).
- Song, X., Sun, G., & Li, Q. (2016). Sensitivity analysis and reliability based design optimization for high-strength steel tailor welded thin-walled structures under crashworthiness. *Thin-Walled Structures*, 109, 132–142. <https://doi.org/10.1016/j.tws.2016.09.003>.
- Srinivas, N., Krause, A., Kakade, S., & Seeger, M. (2010). Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proceedings of the 27th international conference on international conference on machine learning* (pp. 1015–1022). USA: Omnipress.
- Towell, G. G., & Shavlik, J. W. (1994). Knowledge-based artificial neural networks. *Artificial Intelligence*, 70(1), 119–165. [https://doi.org/10.1016/0004-3702\(94\)90105-8](https://doi.org/10.1016/0004-3702(94)90105-8).
- Trucano, T. G., Swiler, L. P., Igusa, T., Oberkampf, W. L., & Pilch, M. (2006). Calibration, validation, and sensitivity analysis: What's what. *Reliability Engineering & System Safety*, 91(10), 1331–1357. <https://doi.org/10.1016/j.res.2005.11.031>.
- Tsai, T., & Li, D. (2008). Utilize bootstrap in small data set learning for pilot run modeling of manufacturing systems. *Expert Systems with Applications*, 35(3), 1293–1300.
- Vidal, A., & Archer, R. (2016). Calibration of a geothermal reservoir model using a global method based on surrogate modeling. In *41st workshop on geothermal reservoir engineering* (pp. 1–8). Stanford University.
- Wang, X., & Berger, J. O. (2016). Estimating shape constrained functions using Gaussian processes. *SIAM/ASA Journal on Uncertainty Quantification*, 4(1), 1–25.
- Wang, L., Beeson, D., Akkaram, S., & Wiggs, G. (2005). Gaussian process metamodels for efficient probabilistic design in complex engineering design spaces. In *ASME 2005 international design engineering technical conferences and computers and information in engineering conference* (pp. 785–798). <https://doi.org/10.1115/DETC2005-85406>.
- Wang, H., Jin, Y., & Doherty, J. (2017). Committee-based active learning for surrogate-assisted particle swarm optimization of expensive problems. *IEEE Transactions on Cybernetics*, 47(9), 2664–2677. <https://doi.org/10.1109/TCYB.2017.2710978>.
- Wang, W., & Welch, W. J. (2018). Bayesian optimization using monotonicity information and its application in machine learning hyperparameter tuning. In *Proceedings of AutoML 2018 @ ICML/IJCAI-ECAI* (pp. 1–13).
- Wu, J., & Feb, M. L. (2017). Bayesian optimization with gradients. In *31st conference on neural information processing systems (NIPS 2017)* (pp. 1–17).
- Yoshimura, M., Shimoyama, K., Misaka, T., & Obayashi, S. (2017). Topology optimization of fluid problems using genetic algorithm assisted by the Kriging model. *International Journal for Numerical Methods in Engineering*, 109(4), 514–532. <https://doi.org/10.1002/nme>.
- Yu, L., Wang, L., & Yu, J. (2008). Identification of product definition patterns in mass customization using a learning-based hybrid approach. *The International Journal of Advanced Manufacturing Technology*, 38(11), 1061–1074. <https://doi.org/10.1007/s00170-007-1152-3>.
- Zhang, Z., Chai, N., Liu, Y., & Xia, B. (2019). Base types selection of PSS based on a priori algorithm and knowledge-based ANN. *IET Collaborative Intelligent Manufacturing*, 1(2), 29–38. <https://doi.org/10.1049/iet-cim.2018.0003>.
- Zhang, X., Wang, S., Yi, L., Xue, H., & Yang, S. (2018). An integrated ant colony optimization algorithm to solve job allocating and tool scheduling problem. *Journal of Engineering Manufacture*, 232(1), 172–182. <https://doi.org/10.1177/0954405416636038>.
- Zhao, T., Montoya-Noguera, S., Phoon, K.-K., & Wang, Y. (2017). Interpolating spatially varying soil property values from sparse data for facilitating characteristic value selection. *Canadian Geotechnical Journal*, 55(2), 171–181. <https://doi.org/10.1139/cgj-2017-0219>.