Metamodel-based Optimization of Stochastic Computer Models for Engineering Design under Uncertain Objective Function

Article *in* IIE Transactions · July 2018

CITATIONS
0

READS
43

3 authors, including:

Guilin Li
National University of Singapore
4 PUBLICATIONS **0** CITATIONS

SEE PROFILE

Matthias H. Y. Tan
City University of Hong Kong
25 PUBLICATIONS **127** CITATIONS

SEE PROFILE

Metamodel-based Optimization of Stochastic Computer Models for Engineering Design under Uncertain Objective Function

Guilin Li[†], Matthias Hwai-yong Tan[†], Szu Hui Ng[‡]
[†]School of Data Science, City University of Hong Kong
[‡]Department of Industrial Systems Engineering & Management, National University of Singapore

Abstract

Stochastic computer models are prevailingly used to help the design engineer to understand and optimize analytically intractable systems. A frequently encountered but often ignored problem is that the objective function representing system performance may contain some uncertain parameters. Due to lack of computationally efficient tools, rational procedures for dealing with the problem such as finding multiple Pareto optimal solutions or conducting sensitivity analysis on the uncertain parameters require the stochastic computer model to be optimized many times, which would incur extensive computational burden. In this work, we provide a computationally efficient metamodel-based solution to capture this uncertainty. This solution first constructs a Cartesian product design over the space of both design variables and uncertain parameters. Thereafter, a radial basis function metamodel is used to provide a smooth prediction surface of the objective value over the space of both design variables and uncertain parameters. Based on the Cartesian product design structure, a fast fitting algorithm is also derived for fitting the metamodel. To illustrate the effectiveness of the developed tools in solving practical problems, they are applied to seek a robust optimal solution to a drug delivery system with uncertain desirability function parameters based on a criterion that we propose.
Keywords: stochastic computer models, Cartesian product design, objective function uncertainty, radial basis function, sensitivity analysis

1. Introduction

With the constantly upgraded computing power, stochastic computer models are becoming important tools for understanding and optimizing engineering systems that are analytically intractable and subject to random fluctuations. Typically, a stochastic computer model is constructed based on technical knowledge of how the engineering system operates. An important purpose of constructing a stochastic computer model is to utilize it to guide engineering design, which aims at choosing the settings for the design variables of the system such that the system performance is optimized. Some examples of engineering design applications of stochastic computer models are the integrated photonic filters in electrical engineering ([50]), aerospike nozzle in aerospace engineering([37]), and microelectromechanical

systems (MEMS) devices ([26]).

An ideal case would be such that the design engineer or the investigator has complete knowledge for determining the objective functions for optimizing the system performance. However, this may not always be true especially when analyzing complex engineering systems. In many areas of applications, there are multiple objectives or difficult-to-evaluate parameters in the objective functions (for example [39, 33, 5]). When multiple objectives need to be optimized simultaneously, the most popular method is to form a composite objective function through weighted sum ([29]) or desirability functions ([51, 35]), where a weight or preference parameter proportional to the user preference is assigned to a particular objective. However, the determination of these preference parameters is usually highly subjective and not straightforward. It requires an investigation of the qualitative and experience-driven information to determine the quantitative preference parameter values. Without the possible trade-off optimal solutions in hand, this is an even more challenging task. Besides these uncertain preference parameters, the uncertainty in an objective function may also come from the difficult-to-evaluate parameters. For example, when designing a nuclear power plant building, the cost incurred by varying rates of leakage of radioactive material can be hard to quantify ([25]). Moreover, a change in the potential cost will possibly result in a totally different optimal solution. This objective function uncertainty problem appears in various areas of application and brings many challenges in both concept and computation.

When dealing with this uncertainty problem, the current development of stochastic computer models mainly focuses on replacing the uncertain parameters by some estimates and assuming the objective function is precisely determined, for example, see [39, 24, 50]. By such a choice, the uncertainty is pushed out of sight through approximating the uncertain parameters by some particular estimates based on the decision maker's preference information, expert judgment, historical records, or a sample mean. Meanwhile, the inferential procedure for other choices of the uncertain parameters is simply ignored. However, unless reliable preference information or an accurate estimate is available, the optimal solution obtained by such methods would be highly subjective to the specific investigator or largely sensitive to the choice of the estimates ([52]).

In the optimization literature, a commonly employed procedure for a multi-objective problem is the *posteriori approach* ([7]), in which a set of different trade-off optimal solutions is first obtained and then a multiple-criteria decision-making technique (such as using high-level information) ([10]) is used to analyze the solutions to choose a most preferred solution. This is because any two different optimal solutions (such as Pareto-points) represent different trade-offs among the objectives, and the decision maker would be in a good position to balance the risk when such choices are presented. Likewise, when the objective function contains some difficult-to-evaluate parameters, a rational procedure is to conduct *sensitivity analysis* (post-optimality analysis) ([47]). For example, when the cost of failure is uncertain, it is important for the investigator to know how profit would be affected by a change in the potential failure

cost. The results of a sensitivity analysis provide information on how sensitive the optimal solution is to a change in the uncertain parameters and establish the upper and lower bounds for the uncertain parameters within which they can vary without causing violent changes in the current optimal solution.

As pointed out by [7], the classical methods use a different philosophy such that the objective function uncertainty is artificially forced out of sight, mainly due to a lack of suitable optimization tools to obtain many optimal solutions efficiently. To properly capture this uncertainty, the most straightforward method should be to optimize the system many times with various settings of the uncertain parameters. However, for a time-consuming stochastic computer model, this would usually incur extensive computational burden. In the context of multi-objective optimization, an alternative solution is the *interactive approach* ([9, 4]), in which the decision maker's responses to specific questions are used iteratively to guide the solution towards the preferred part of the Pareto-optimal region. Although this method reduces the computational burden of optimizing the system repeatedly for many different uncertain parameter choices, it requires a lot of cognitive effort on the part of the decision maker. Moreover, when the modeler and the decision maker is not the same person, this method can be infeasible or very inefficient.

In short, it is important to capture the uncertainty in the objective function to offer the decision maker desired flexibility in making a more informed and rational decision. However, capturing this uncertainty means the investigator has to optimize the computer model repeatedly for many different choices of the uncertain parameters, which is very time-consuming. In this work, we propose a computationally efficient solution to this problem from an experimental design and metamodeling point of view. This solution first constructs a Cartesian product design over the space of both design variables and uncertain parameters. Thereafter, a radial basis function metamodel is used to provide a smooth prediction surface of the objective value over the design region for the design variables and uncertain parameters. In addition, based on the Cartesian product design structure, a fast fitting algorithm is also derived for fitting the metamodel. To demonstrate the computational efficiency of the proposed method, we utilize it to optimize an intuitively appealing but computationally intensive criterion that we propose for choosing a robust optimal solution in optimizing a desirability function with uncertain parameters. To illustrate the effectiveness of the proposed method for solving practical problems, we use the optimization of drug release from a polymer matrix ([41]) as a test example, where the developed tools are used to facilitate a robust selection of the parameters in the desirability function. The idea of using nonparametric emulators like the radial basis function metamodel for time consuming stochastic computer models has been investigated by other scholars. In the operations research literature, [46] and [2] are among the early articles that promote the use of Gaussian process emulators for stochastic computer models. Huang et al. [20] develop an expected improvement criterion for optimization of a stochastic computer model. Gaussian process emulators have been widely employed as surrogates for time consuming computer models for the physical sciences [40]

but support vector regression has been used as well ([48]). Some problems solved using these surrogates are reliability optimization [16] and calibration of computer model parameters [49]. Recently, [18], [30], [36], [44], and [19] utilize the Gaussian process emulator and other regression models to build surrogates for stochastic computer models for the physical sciences. However, none of these works deal with optimization of time consuming stochastic simulation models *with explicit and systematic consideration of objective function uncertainty.*

The remainder of the paper is organized as follows: Section 2 introduces some examples of uncertain parameters in general classes of objective functions encountered in practical applications and proposes a criterion for choosing a robust optimal solution for a desirability function with uncertain parameters. Section 3 first introduces the approximate interpolation of an objective function estimate with fixed objective function parameters with an RBF metamodel, and then proposes design and modeling techniques for conveniently capturing the uncertainty in the objective function. In Section 4, the developed tools are used to solve a practical problem and demonstrate how a more informed and rational decision can be made. Finally, Section 5 concludes the paper.

2. Objective functions with uncertain parameters

In this section, to demonstrate the practical scenarios where the proposed framework is helpful, we present some of the major cases involving stochastic computer models where there are uncertain parameters in the objective functions. Following each specific case, the corresponding post-optimal decision-making tools are also discussed. Most of these tools require optimizing the objective function under many different choices of the uncertain parameters, which motivates the development of computationally efficient tools for conducting such post-optimal analysis for time-consuming stochastic computer models.

2.1. Uncertain failure cost in structural design

In structural design, stochastic computer models are usually used to predict the strength of the structure or the probability of failure (see [34, 21]). Such models accept as inputs a vector of design/decision variables $\boldsymbol{x}$ and a random vector $\boldsymbol{\xi}$, and returns a random output vector of damage level η. Using the *cost-benefit analysis* tools ([23]), the optimal structural design is determined, where the 'cost' is the price of increasing safety and the 'benefit' is the reduced risk in terms of expected failure cost. An example of the total cost C_T defined by [23] is

$$C_T = C_I + C_F \int_{\eta_0}^{1} \frac{\eta - \eta_0}{1 - \eta_0} p(\eta) d\eta \tag{1}$$

where C_I is the initial cost (construction cost), $\eta \in [\eta_0, 1]$ is the damage level, η_0 is the initial damage level, and C_F is the failure cost at ultimate limit state $\eta = 1$. The quantity $p(\eta)$ is the probability density function for the occurrence of damage level η, which is estimated based on the output of the stochastic computer model. Note that C_I and $p(\eta)$ are functions of the vector of design variables $\boldsymbol{x}$. In this objective function, the ultimate limit state failure cost C_F is an uncertain parameter which varies from structure to structure (e.g., building to building) and contains some difficult-to-evaluate parts such as casualties caused by the collapse of buildings or serious failure of nuclear power plants. Note that in this example, although the stochastic computer model may have captured the random fluctuations in loads and structural resistance, the failure cost parameter C_F is not part of the stochastic computer model and does not affect the computer model output and the estimate of $p(\eta)$ based on the output. Instead, it is an uncertain parameter in the objective function.

The simplest solution is to choose C_F based on subjective judgments or historical records. However, the optimal solution could be very sensitive to the choice of C_F and hence result in very unreliable and subjective decisions. A more rational decision can be made if the structural designer can conduct a *sensitivity analysis* (post-optimality analysis) ([47]) of the optimal design solution with respect to the uncertainty in C_F. This involves re-optimizing the system under alternative choices of C_F to determine how sensitive the optimal solution is to a change in C_F. If significant changes of the optimal solution are identified and associated with the changes in C_F, more attention and effort should, therefore, be made to select the value of C_F in order to increase the robustness of the decision.

When there are difficult-to-evaluate parameters in the objective function, conducting *sensitivity analysis* can also enhance the communication from modeler to decision makers (e.g. by making recommendations more informative, credible or understandable). For example, the result of *sensitivity analysis* would provide information to the manager on how profit would be affected by a change in the potential failure cost. If the results are insensitive to changes in uncertain parameters, the manager can be quite confident that the decision made is good. Note that to efficiently conduct *sensitivity analysis* for time-consuming stochastic computer models, a procedure for optimizing the system with a low computing budget is desired.

2.2. Pareto-optimal front

Suppose there are S objective functions, denoted as $R_s(\boldsymbol{x}), s = 1, \cdots, S$, where $\boldsymbol{x}$ is the vector of decision variables (also called design or control variables in this paper). In the stochastic simulation context considered in this paper, $R_s(\boldsymbol{x}) = E_{\boldsymbol{Y}}\{J_s[\boldsymbol{Y}(\boldsymbol{x}, \boldsymbol{\xi}), \boldsymbol{x}]\}$, where $\boldsymbol{Y}(\boldsymbol{x}, \boldsymbol{\xi})$ is the output of the stochastic simulation model that accepts $\boldsymbol{x}$ and a random vector $\boldsymbol{\xi}$ as inputs (the damage level η in Section 2.1), J_s is a loss/utility function that depends on $\boldsymbol{Y}$ and $\boldsymbol{x}$, and $E_{\boldsymbol{Y}}$ is the expectation with respect to $\boldsymbol{Y}$. For this multi-objective optimization problem, an ideal procedure is first to find multiple trade-off solutions with a wide range of values (Pareto-optimal front), and then a multiple-

criteria decision-making technique (such as using high-level information) ([10]) is used to analyze the solutions to choose a most preferred solution. Specifically, the Pareto-optimal front consists of solutions with the property that none of the objective values can be improved without degrading some of the other objective values. The simplest way to obtain the Pareto-optimal front is the weighted sum method, where the solution $\boldsymbol{x}^*$ optimizing the objective $R(\boldsymbol{x}) = \sum_{s=1}^{S} w_s R_s(\boldsymbol{x})$ is Pareto-optimal if $w_s > 0, s = 1, \cdots, S$ and $\sum_{s=1}^{S} w_s = 1$. Every different choice of the weight parameters under the constraints $w_s > 0, s = 1, \cdots, S$ and $\sum_{s=1}^{S} w_s = 1$ would produce different points in the Pareto-optimal front ([31]). In this case, the weights $w_s > 0, s = 1, \cdots, S$ are uncertain parameters in the objective function $R(\boldsymbol{x})$, and a procedure allowing convenient optimization of $R(\boldsymbol{x})$ under different weight values would save considerable computational effort.

2.3. Desirability function

The desirability function ([17, 11]) is one of the most widely used methods for the optimization of multiple responses or multiple objective engineering systems. This approach first transforms the different objectives into a common scale $[0, 1]$ and then combines them into an overall objective using the geometric mean. Suppose there are S scalar performance criteria, denoted as $K_s(\boldsymbol{Y}, \boldsymbol{x}), s = 1, \cdots, S$ that each depends on the output $\boldsymbol{Y}$ of the stochastic computer model and the vector $\boldsymbol{x}$ of decision variables, as in Section 2.2. For simplicity of notation, the argument $(\boldsymbol{x}, \boldsymbol{\xi})$ of $\boldsymbol{Y}$ is omitted and we shall abbreviate $K_s(\boldsymbol{Y}, \boldsymbol{x})$ with K_s. For each of the S performance criteria, a corresponding desirability score function is constructed which is high when K_s is at a desirable level (such as minimum, maximum, or target) and low when K_s is at an undesirable level. For example, for smaller-the-better K_s, [11] construct the desirability function as

$$J_s^{min}(\boldsymbol{Y}, \boldsymbol{x}, \boldsymbol{\theta}_s) = \begin{cases} 0 & \text{if } K_s > B \\ [\frac{K_s - B}{A - B}]^C & \text{if } A \leqslant K_s \leqslant B \\ 1 & K_s < A \end{cases} \tag{2}$$

where the preference parameter vector $\boldsymbol{\theta}_s = (A, B, C)$ is chosen by the investigator. For target-the-best K_s, the desirability function is

$$J_s^{target}(\boldsymbol{Y}, \boldsymbol{x}, \boldsymbol{\theta}_s) = \begin{cases} [\frac{K_s - A}{T_0 - A}]^C & \text{if } A \leqslant J_s \leqslant T_0 \\ [\frac{K_s - B}{T_0 - B}]^D & \text{if } T_0 < J_s \leqslant B \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where T_0 is the target value chosen by the investigator, and we call $\boldsymbol{\theta}_s = (A, B, T_0, C, D)$ a preference parameter vector. For larger-the-better K_s, the desirability function is given in [51].

Given that the S desirability functions $J_1, \cdots, J_S$ are on the same scale $[0, 1]$, they can be combined to produce an overall desirability score function through the geometric mean:

$$J(\boldsymbol{Y}, \boldsymbol{x}, \boldsymbol{\theta}) = [\prod_{s=1}^{S} J_s(\boldsymbol{Y}, \boldsymbol{x}, \boldsymbol{\theta}_s)]^{1/S} \tag{4}$$

where $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_S)$. Taking expectation of the overall desirability score gives an objective function $R(\boldsymbol{x}, \boldsymbol{\theta}) = E_{\boldsymbol{Y}}\{J(\boldsymbol{Y}, \boldsymbol{x}, \boldsymbol{\theta})\}$, which we call the expected desirability score. Note that the preference parameter vector $\boldsymbol{\theta}$ has to be chosen such that the resulting desirability score function gives values that are sensible to the decision maker. For example, choosing $C > 1$ in (3) would place more reward on being close to the target value, and choosing $0 < C < 1$ would make this less important. We point out that in many existing works involving multiresponse physical experiments (e.g., [22]), the performance criteria/responses $K_s, s = 1, \cdots, S$ are replaced with their expected values, which leads to a deterministic overall desirability score. This yields an objective function different from one obtained by taking the expectation of the overall desirability score, as given by $R(\boldsymbol{x}, \boldsymbol{\theta})$ above. However, the approach of replacing $K_s, s = 1, \cdots, S$ with their expected values seems to be motivated by computational convenience and the fact that the variations in physical experiments are often different from the variation that is observed in actual system operation. Since our interest is in optimizing stochastic simulation models, which are intended to simulate variation in actual system operation, the use of $R(\boldsymbol{x}, \boldsymbol{\theta})$ as an objective function is sensible. The expected desirability score is also employed as an objective function in several papers by Heike Trautmann (e.g., [45]).

The desirability-based method is easy to understand and is available in many data analysis software packages. It has been extensively used in optimization of industrial problems, such as [13, 54, 53]. However, to use this method, the investigator needs to choose values of the preference parameters to represent the trade-off preferences. This is not a trivial task and the choices may possibly be made without awareness of the fact that different values of these preference parameters would possibly produce very different optimal solutions. Other than specifying these preference parameters subjectively, [22] propose an *interactive approach* to select the values for these parameters. They propose to first initialize the preference parameter and obtain an initial optimal solution, then interact with the decision maker to find his reaction to this solution and thereafter adjust the preference parameter value until some stopping criterion is satisfied. Although this method reduces the computational burden of optimizing the system repeatedly for many different preference parameter settings, it requires a lot of cognitive effort on the part of the decision maker.

An alternative solution to this uncertain preference parameter problem is to select the most robust choice of the

preference parameter, as we now explain. Specifically, for a particular choice of the preference parameter $\boldsymbol{\theta}$, one would obtain the corresponding optimal solution depending on $\boldsymbol{\theta}$ as $\boldsymbol{x}^*(\boldsymbol{\theta}) := \text{argmax}_{\boldsymbol{x}} R(\boldsymbol{x}, \boldsymbol{\theta})$. It makes sense to restrict the choice of $\boldsymbol{x}$ to the set $\{\boldsymbol{x}^*(\boldsymbol{\theta}) : \boldsymbol{\theta} \in \Theta\}$ of optimal solutions. To select the most robust $\boldsymbol{x}^*(\boldsymbol{\theta})$ or equivalently the most robust $\boldsymbol{\theta}$, a reasonable criterion to use should measure how $\boldsymbol{x}^*(\boldsymbol{\theta})$ would perform under all the other possible choices of the preference parameter in Θ. Suppose the 'true' preference parameter value is $\boldsymbol{\vartheta}$. Then, the maximum expected desirability score would be achieved at point $\boldsymbol{x} = \boldsymbol{x}^*(\boldsymbol{\vartheta})$ with value $R[\boldsymbol{x}^*(\boldsymbol{\vartheta}), \boldsymbol{\vartheta}]$ while the expected desirability score of setting $\boldsymbol{x} = \boldsymbol{x}^*(\boldsymbol{\theta})$ would be $R[\boldsymbol{x}^*(\boldsymbol{\theta}), \boldsymbol{\vartheta}]$. Hence, the ratio $R[\boldsymbol{x}^*(\boldsymbol{\theta}), \boldsymbol{\vartheta}]/R[\boldsymbol{x}^*(\boldsymbol{\vartheta}), \boldsymbol{\vartheta}]$, which is between zero and one, is a measure of the closeness of $\boldsymbol{x} = \boldsymbol{x}^*(\boldsymbol{\theta})$ to the optimal solution when the 'true' preference parameter value is $\boldsymbol{\vartheta}$. A larger ratio implies that $\boldsymbol{x}^*(\boldsymbol{\theta})$ is closer in desirability to the optimal solution $\boldsymbol{x}^*(\boldsymbol{\vartheta})$. We define the robust preference parameter value as the value $\boldsymbol{\theta}$ of the preference parameter that maximizes the average ratio over Θ, i.e., $Q(\boldsymbol{\theta}) = \int_{\boldsymbol{\vartheta}\in\Theta}\{R[\boldsymbol{x}^*(\boldsymbol{\theta}), \boldsymbol{\vartheta}]/R[\boldsymbol{x}^*(\boldsymbol{\vartheta}), \boldsymbol{\vartheta}]\}d\boldsymbol{\vartheta} / \int_{\boldsymbol{\vartheta}\in\Theta} d\boldsymbol{\vartheta}$, which is an intuitively reasonable definition for cases where all values of the preference parameter in Θ are approximately equally likely to be the true value. Hence, we propose the following definition to select a robust preference parameter value for a desirability function.

Definition 1 (robust preference parameter value for desirability functions). *For the maximization of a expected desirability score $R(\boldsymbol{x}, \boldsymbol{\theta})$ with design variable $\boldsymbol{x} \in \mathcal{X}$ and uncertain preference parameter $\boldsymbol{\theta} \in \Theta$, denote the optimal solution for a specific choice of $\boldsymbol{\theta}$ as $\boldsymbol{x}^*(\boldsymbol{\theta}) := argmax_{\boldsymbol{x}} R(\boldsymbol{x}, \boldsymbol{\theta})$. A preference parameter choice $\boldsymbol{\theta}^*$ is called the robust preference parameter value of desirability function $R(\boldsymbol{x}; \boldsymbol{\theta})$, if it is the global maximum solution of the following problem:*

$$\begin{aligned} &\textit{Maximize} \quad Q(\boldsymbol{\theta}) = \int_{\boldsymbol{\vartheta}\in\Theta} \frac{R[\boldsymbol{x}^*(\boldsymbol{\theta}), \boldsymbol{\vartheta}]}{R[\boldsymbol{x}^*(\boldsymbol{\vartheta}), \boldsymbol{\vartheta}]} d\boldsymbol{\vartheta} / \int_{\boldsymbol{\vartheta}\in\Theta} d\boldsymbol{\vartheta} \\ &\textit{subject to} \quad \boldsymbol{\theta} \in \Theta \end{aligned}$$

The corresponding solution $\boldsymbol{x}^*(\boldsymbol{\theta}^*)$ is called the robust optimal solution.

Remark: For details on computation of the robust preference parameter value, see Appendix A

By using this robustness definition, one can select a preference parameter and obtain the corresponding robust optimal solution without depending on subjective and qualitative information, except for the specification of Θ. Nonetheless, to obtain the robust choice of the preference parameter value, we need to optimize the computer model under a large number of preference parameter choices $\boldsymbol{\theta} \in \Theta$. In such cases, an integrated and efficient solution to optimizing the time-consuming computer model repeatedly with many different preference parameter settings would be very helpful. Such a solution would make the desirability function method more appealing to the practitioners who have trouble selecting the preference parameter value based on qualitative information.

3. Methodology

In this section, we first formally define the computer model based engineering design optimization problem with uncertain parameters in the objective function. Thereafter, we introduce a metamodeling method for optimizing stochastic computer models when the uncertain parameters in the objective function are fixed. However, this method would require one to refit the metamodel every time the values of the uncertain parameters are changed, which is time-consuming and unstable. Following that, we propose an alternative solution that provides a smooth prediction of the objective function surface over both the design variables space and uncertain parameters space.

3.1. Engineering design optimization under uncertain objective function

In engineering design problems, stochastic computer models are often constructed based on technical knowledge of how the engineering system operates. We consider a stochastic computer model that accepts a fixed design variable vector $\boldsymbol{x} = (x_1, \cdots, x_k)^T \in \mathcal{X} \subset \mathbb{R}^k$, a random system fluctuation vector $\boldsymbol{\xi}$, and returns a random output vector $\boldsymbol{Y}(\boldsymbol{x}, \boldsymbol{\xi}) \in \mathbb{R}^p$. For optimization purpose, the objective function with uncertain parameter $\boldsymbol{\theta}$ can be defined as

$$R(\boldsymbol{x}, \boldsymbol{\theta}) = E_{\boldsymbol{Y}}\{J[\boldsymbol{Y}(\boldsymbol{x}, \boldsymbol{\xi}), \boldsymbol{x}, \boldsymbol{\theta}]\} \tag{5}$$

where $\boldsymbol{\theta} \in \Theta \subset \mathbb{R}^q$, $J : \mathbb{R}^p \times \mathcal{X} \times \Theta \to \mathbb{R}$ is a real valued continuous function that measures the system performance and $E_{\boldsymbol{Y}}$ is the expectation operator with respect to the randomness in $\boldsymbol{Y}(\boldsymbol{x}, \boldsymbol{\xi})$ (induced by randomness in $\boldsymbol{\xi}$). In this paper, we make the nonrestrictive assumptions that J is a bounded function (e.g., the overall desirability score (4)) and the cumulative distribution function of $\boldsymbol{Y}$ is a continuous function of $\boldsymbol{x}$. This assumption and the dominated convergence theorem implies that $R(\boldsymbol{x}, \boldsymbol{\theta})$ *is a continuous function of* $(\boldsymbol{x}, \boldsymbol{\theta})$. The optimization problem (for simplicity, maximization is taken to be the standard) depending on $\boldsymbol{\theta}$ is then

$$\text{maximize } R(\boldsymbol{x}, \boldsymbol{\theta}) \text{ over all } \boldsymbol{x} \in \mathcal{X} \tag{6}$$

When dealing with the objective function uncertainty problem, the current development of methodologies for stochastic computer models mainly focuses on pushing the uncertainty out of sight through replacing the uncertain parameter $\boldsymbol{\theta}$ by some particular estimate $\hat{\boldsymbol{\theta}}$, and solving a single optimization problem

$$\text{maximize } R(\boldsymbol{x}, \hat{\boldsymbol{\theta}}) \text{ over all } \boldsymbol{x} \in \mathcal{X} \tag{7}$$

By doing this, the inferential procedure for other choices of $\boldsymbol{\theta} \in \Theta$ is simply disregarded.

As argued in Section 2, capturing the uncertainty in $\boldsymbol{\theta}$ is of great importance to make a rational and reliable decision. Therefore, in this work we would regard the optimization of the computer model as a family of optimization problems where for each different $\boldsymbol{\theta} \in \Theta$, it yields an optimal value

$$\boldsymbol{x}^*(\boldsymbol{\theta}) := \text{argmax}_{\boldsymbol{x}} R(\boldsymbol{x}, \boldsymbol{\theta}) \tag{8}$$

which needs to be analyzed in its dependence on $\boldsymbol{\theta}$.

3.2. Metamodel-based optimization under a fixed objective function

Frequently, computer codes that simulate an engineering system are very time-consuming to run. Consequently, a practically appealing approach is to approximate the computer model by a more computationally efficient metamodel. A metamodel-based optimization strategy requires one to first identify a metamodel form, then design an experiment to collect data by running the expensive computer code, and finally fit and optimize the metamodel ([3]). Metamodels can be built using many types of regression models with a variety of prediction power. For example, linear regression models are easy to built and have been widely used. Although simple, this type of models lacks the ability to model complicated surfaces. By using more sophisticated methods such as Gaussian process models or radial basis function (RBF) models, one can achieve better prediction. RBF models using compactly supported RBFs can be fitted efficiently to very large data sets. RBF models are also applicable to problems with high dimensional design variable spaces since generally few restrictions are imposed on the location of sample points. In this section, we focus on using the RBF metamodel to predict and optimize the stochastic computer model under a fixed objective function.

Suppose an experiment with n design points $\mathcal{D} = \{\boldsymbol{x}^1, \cdots, \boldsymbol{x}^n\}$ is chosen, and M independent replicates have been obtained from running the computer model at each design point. Denote the data from the experiment by $S_n = \{\boldsymbol{Y}_i = [\boldsymbol{Y}_1(\boldsymbol{x}^i, \boldsymbol{\xi}), \cdots, \boldsymbol{Y}_M(\boldsymbol{x}^i, \boldsymbol{\xi})]'\}_{i=1}^n$. A standard approach to optimizing the objective function $R(\boldsymbol{x}, \boldsymbol{\theta})$ (5) for predetermined $\boldsymbol{\theta}$ is to first estimate $R(\boldsymbol{x}^i, \boldsymbol{\theta})$ with output observations $\boldsymbol{Y}_i$ using some estimator $\widetilde{R}(\boldsymbol{x}^i, \boldsymbol{\theta}) = g(\boldsymbol{Y}_i)$, then fit an RBF regression model with the control-response data pairs $\{(\boldsymbol{x}^1, \widetilde{R}_1), \cdots, (\boldsymbol{x}^n, \widetilde{R}_n)\}$. Note that for simplicity of notation, we use $\widetilde{R}(\boldsymbol{x}^i, \boldsymbol{\theta})$ and $\widetilde{R}_i$ interchangeably. Thereafter, the fitted RBF metamodel can be used to predict $R(\boldsymbol{x}, \boldsymbol{\theta})$ for all $\boldsymbol{x} \in \mathcal{X}$ for a predetermined $\boldsymbol{\theta}$, and the problem of optimizing the computer model is transformed to optimizing the RBF metamodel. A straightforward estimator of $R(\boldsymbol{x}^i, \boldsymbol{\theta})$ would be

$$\widetilde{R}(\boldsymbol{x}^i, \boldsymbol{\theta}) = \sum_{j=1}^{M} J[\boldsymbol{Y}_j(\boldsymbol{x}^i, \boldsymbol{\xi}), \boldsymbol{x}^i, \boldsymbol{\theta}]/M \tag{9}$$

Although averaging over the replicated observations $\boldsymbol{Y}_j(\boldsymbol{x}^i, \boldsymbol{\xi}), j = 1, \ldots, M$ reduces the random error in $\widetilde{R}(\boldsymbol{x}^i, \boldsymbol{\theta})$, it is still a noisy version of the true value of $R(\boldsymbol{x}^i, \boldsymbol{\theta})$. Therefore, we include a nugget parameter in the RBF metamodel to mitigate overfitting and the bootstrap method is used for performing statistical inference. An advantage of such a treatment is that the methodology is not restricted to situations where random errors associated with $\widetilde{R}_i, i = 1, \cdots, n$ are independent and identically distributed.

Now, we introduce the RBF metamodel which is extensively studied by [6]. This RBF model approximates the unknown function with a linear combination of positive definite kernels and takes the forms of

$$\hat{R}(\boldsymbol{x}, \boldsymbol{\theta}) = \mu + \sum_{i=1}^{n} \beta_i \phi(\boldsymbol{x} - \boldsymbol{x}^i) \tag{10}$$

where $\hat{R}(\boldsymbol{x}, \boldsymbol{\theta})$ is the predicted objective value for any $\boldsymbol{x} \in \mathcal{X}$ and $\phi(\cdot)$ is a kernel basis function. Many choices of $\phi(\cdot)$ are available, and examples include multiquadrics, thin plate splines, cubic splines, Gaussian, and inverse multiquadrics. A difficulty with the widely used Gaussian basis function is that the kernel/correlation matrix can often be close to singularity, which induces a lot of computational problems. Hence, in this article, we adopt the basis function

$$\phi(\boldsymbol{x} - \boldsymbol{z}) = \prod_{j=1}^{k} \exp[-|x_j - z_j|/\gamma_j](|x_j - z_j|/\gamma_j + 1), \tag{11}$$

which is a member of the class of Matern correlation functions ([40]). In (11), the γ_j's are positive parameters to be estimated. Note that some authors ([14], page 41) reserve the term RBF for basis functions ϕ that depend on $(\boldsymbol{x} - \boldsymbol{z})$ only through $||\boldsymbol{x} - \boldsymbol{z}||$, and use the term kernel for more general bases that depend on the translates $(\boldsymbol{x} - \boldsymbol{z})$. However, we follow the convention of [6] (page 4) and call (11) a RBF as well. This basis function avoids numerical difficulties commonly encountered with the Gaussian RBF. It is twice continuously differentiable, but not three times continuously differentiable. Let $\boldsymbol{\beta} = (\beta_1, \cdots, \beta_n)'$, $\boldsymbol{\gamma} = (\gamma_1, \cdots, \gamma_k)'$, $\boldsymbol{B}$ be an $n \times n$ matrix with elements $(\boldsymbol{B})_{ij} = \phi(\boldsymbol{x}^i - \boldsymbol{x}^j)$, and $\widetilde{\boldsymbol{R}} = (\widetilde{R}_1, \cdots, \widetilde{R}_n)'$. Then, given $\mu, \boldsymbol{\gamma}$, and the nugget $\lambda > 0$, the vector $\boldsymbol{\beta}$ is determined by solving the linear equations

$$(\boldsymbol{B} + \lambda \boldsymbol{I})\boldsymbol{\beta} = \widetilde{\boldsymbol{R}} - \mu \mathbf{1} \tag{12}$$

where $\boldsymbol{I}$ is an $n \times n$ identity matrix and $\mathbf{1}$ is an $n \times 1$ vector of 1's.

This RBF model is mathematically equivalent to the posterior mean of a Gaussian process (GP) with correlation function (11). Therefore, the correlation length parameters $\boldsymbol{\gamma}$ determine the degree of influence of each observation on the prediction $\hat{R}(\boldsymbol{x}, \boldsymbol{\theta})$: with short correlation length, the prediction at $\boldsymbol{x}$ depends more strongly on nearby observations

and weakly on far away observations. The above system of equations (12) for determining $\boldsymbol{\beta}$ with a nugget parameter can be motivated by considering a GP model with additive independent and identically distributed normal errors with unknown variance. Note that if $\lambda = 0$, the coefficients $\boldsymbol{\beta}$ given by (12) will make (10) interpolate the data $\{(\boldsymbol{x}^1, \tilde{R}_1), \cdots, (\boldsymbol{x}^n, \tilde{R}_n)\}$, but if $\lambda > 0$, the model will not interpolate the data.

To determine $\mu, \lambda, \boldsymbol{\gamma}$, a common way is to set $\mu = \sum_{i=1}^{n} \tilde{R}_i/n$ and choose $\boldsymbol{\gamma}, \lambda$ using leave-one-out (LOO) crossed validation. Specifically, let $\tilde{\boldsymbol{R}}_{-i} = (\tilde{R}_1, \cdots, \tilde{R}_{i-1}, \tilde{R}_{i+1}, \cdots, \tilde{R}_n)'$. Then, the predicted value of the response variable at $\boldsymbol{x}^i$ using the LOO data $\tilde{\boldsymbol{R}}_{-i}$ is $\hat{R}(\boldsymbol{x}^i, \boldsymbol{\theta})|\tilde{\boldsymbol{R}}_{-i}$, with prediction error $e_i = \tilde{R}_i - \hat{R}(\boldsymbol{x}^i, \boldsymbol{\theta})|\tilde{\boldsymbol{R}}_{-i}$. Then the optimal value of λ and $\boldsymbol{\gamma}$ can be obtained by minimizing the mean square prediction error:

$$E_{LOO} = \sum_{i=1}^{n} e_i^2/n \tag{13}$$

Aside from being an intuitively reasonable method, an advantage of estimating the parameters $\boldsymbol{\gamma}, \lambda$ for the RBF model by minimizing (13) is that the error vector $\boldsymbol{e} = (e_1, \cdots, e_n)'$ can be calculated analytically as (see online supplement of [43] for a proof)

$$\boldsymbol{e} = \text{diag}\{(\boldsymbol{B} + \lambda \boldsymbol{I})^{-1}\}^{-1}(\boldsymbol{B} + \lambda \boldsymbol{I})^{-1}(\tilde{\boldsymbol{R}} - \mu \boldsymbol{1}) \tag{14}$$

This shortcut formula saves substantial computation time over a brute force approach to compute the error vector $\boldsymbol{e}$.

As mentioned, the decision made by using only one predetermined estimate of $\boldsymbol{\theta}$ would result in unforeseeable risk; hence, solving the optimization problem under a set of candidate values $\boldsymbol{\theta}^1, \cdots, \boldsymbol{\theta}^m$ would be desired. In such cases, one has to refit the RBF metamodel (10) m times. The calculation of $(\boldsymbol{B} + \lambda \boldsymbol{I})^{-1}$ in (14) would take $O(n^3)$ arithmetic operations (see [32] Page 55), and hence the fitting of each $\hat{R}(\cdot, \boldsymbol{\theta}^i)$ would take $O(n^3)$ arithmetic operations. Thus, fitting/estimating all $\hat{R}(\cdot, \boldsymbol{\theta}^1), \cdots, \hat{R}(\cdot, \boldsymbol{\theta}^m)$ would take $O(mn^3)$ arithmetic operations. For purposes such as calculating the Pareto front, conducting sensitivity analysis or obtaining the robust preference parameter using Definition 1, the required RBF model fitting times could be quite large because m can be very large. Moreover, even if the investigator can afford the required computational budget to fit the m RBF models, the predicted value $\hat{R}(\bar{\boldsymbol{x}}, \boldsymbol{\theta})$ at a fixed decision point $\bar{\boldsymbol{x}}$ could possibly be a discontinuous function of $\boldsymbol{\theta}$. One reason is the existence of many possible local optimizers of the objective function (13) when estimating λ and $\boldsymbol{\gamma}$, which is an inevitable problem for fitting RBF and GP models with the current development of these models.

3.3. Metamodel-based optimization under uncertain objective function

To capture the uncertainty in $\boldsymbol{\theta}$ in a more efficient way, we now propose a computationally efficient solution based on the unique structure of the RBF metamodel. The fact that for fixed μ, λ, and $\boldsymbol{\gamma}$, the coefficient $\boldsymbol{\beta}$ of the RBF metamodel is linear in the vector of response data (see (12)) allows us to use the short-cut formula (14) to calculate the leave-one-out crossed validation errors. Furthermore, the model form of the RBF metamodel is characterized by *separability*: the RBF (11) is a product of functions that each depends on an input variable. This *separability* structure enables the use of a fast matrix inversion formula for calculating $(\boldsymbol{B} + \lambda \boldsymbol{I})^{-1}$ if the design has a Cartesian product structure. This fact motivates us to collect data using a Cartesian product of a design in $\mathcal{X}$ and a design in Θ. The obtained data can then be used to fit an RBF model for predicting the objective function value over both the space of $\boldsymbol{x}$ and the space of $\boldsymbol{\theta}$. In this way, we can not only reduce the computational burden of fitting the RBF model, but also provide a smooth prediction surface of $\hat{R}(\bar{\boldsymbol{x}}, \boldsymbol{\theta})$ over the space of $\boldsymbol{\theta}$ for a fixed decision vector point $\bar{\boldsymbol{x}}$.

We now describe the design and model fitting strategies in detail. Suppose we use a Cartesian product design

$$\widetilde{\mathcal{D}} = \{\boldsymbol{x}^1, \cdots, \boldsymbol{x}^n\} \times \{\boldsymbol{\theta}^1, \cdots, \boldsymbol{\theta}^m\} = \{(\boldsymbol{x}^1, \boldsymbol{\theta}^1), \cdots, (\boldsymbol{x}^1, \boldsymbol{\theta}^m), \cdots, (\boldsymbol{x}^n, \boldsymbol{\theta}^1), \cdots, (\boldsymbol{x}^n, \boldsymbol{\theta}^m)\} := \{\boldsymbol{d}^1, \cdots, \boldsymbol{d}^N\}, \quad (15)$$

where $\boldsymbol{d} = (\boldsymbol{x}, \boldsymbol{\theta}) = (x_1, \cdots, x_k, \theta_1, \cdots, \theta_q) = (d_1, \cdots, d_{k+q})$ and $N = n \times m$. Given the data $S_n = \{\boldsymbol{Y}_v = [\boldsymbol{Y}_1(\boldsymbol{x}^v, \boldsymbol{\xi}), \cdots, \boldsymbol{Y}_M(\boldsymbol{x}^v, \boldsymbol{\xi})]'\}_{v=1}^n$ obtained from running the stochastic simulator, we can obtain estimates of $R(\boldsymbol{d}^i)$ for all $\boldsymbol{d}^i \in \widetilde{\mathcal{D}}, i = 1, \cdots, N$ using

$$\widetilde{R}(\boldsymbol{d}^i) = \widetilde{R}(\boldsymbol{x}^v, \boldsymbol{\theta}^l) = \sum_{j=1}^{M} J[\boldsymbol{Y}_j(\boldsymbol{x}^v, \boldsymbol{\xi}), \boldsymbol{x}^v, \boldsymbol{\theta}^l]/M \quad \text{for } v = 1, \cdots, n, l = 1, \cdots, m \quad (16)$$

We emphasize that this estimator is based on the same data as the estimator in (9), i.e., no additional computer model runs are needed to estimate the objective function as its parameter vector $\boldsymbol{\theta}$ varies. Then, the RBF model

$$\begin{aligned}\hat{R}(\boldsymbol{x}, \boldsymbol{\theta}) = \hat{R}(\boldsymbol{d}) &= \mu + \sum_{i=1}^{N} \beta_i \prod_{e=1}^{k+q} \exp[-|d_e - d_e^i|/\gamma_e](|d_e - d_e^i|/\gamma_e + 1) \\ &= \mu + \sum_{i=1}^{N} \beta_i \prod_{v=1}^{k} \exp[-|x_v - x_v^i|/\gamma_v](|x_v - x_v^i|/\gamma_v + 1) \prod_{l=1}^{q} \exp[-|\theta_l - \theta_l^i|/\gamma_{k+l}](|\theta_l - \theta_l^i|/\gamma_{k+l} + 1) \quad (17)\end{aligned}$$

is used to construct a predictor of $R(\boldsymbol{x}, \boldsymbol{\theta})$ for all $(\boldsymbol{x}, \boldsymbol{\theta}) \in \mathcal{X} \times \Theta$, , where x_v^i is the v_{th} component of $\boldsymbol{x}^i$ and θ_l^i is the l_{th} component of $\boldsymbol{\theta}^i$ (similarly for d_e^i).

Since $\widetilde{\mathcal{D}}$ in (15) is a Cartesian product design, an algorithm that reduces the number of arithmetic operations to

fit the RBF model (17) compared to a direct approach is available. Specifically, for design (15), one can verify that

$$\boldsymbol{B} + \lambda \boldsymbol{I}_N = \boldsymbol{B}_1 \otimes \boldsymbol{B}_2 + \lambda \boldsymbol{I}_N \tag{18}$$

where $\boldsymbol{I}_N$ is an $N \times N$ identity matrix, $(\boldsymbol{B})_{ij} = \phi(\boldsymbol{d}^i - \boldsymbol{d}^j)$, $(\boldsymbol{B}_1)_{ij} = \phi(\boldsymbol{x}^i - \boldsymbol{x}^j)$, $(\boldsymbol{B}_2)_{ij} = \phi(\boldsymbol{\theta}^i - \boldsymbol{\theta}^j)$, and ϕ is the RBF given in (11).

Based on (18), we derive a computationally convenient formula for $(\boldsymbol{B} + \lambda \boldsymbol{I}_N)^{-1}$ used in (12) and (14) when fitting the RBF model. Since $\boldsymbol{B}_1$ and $\boldsymbol{B}_2$ are positive definite matrices, their eigendecompositions can be denoted by $\boldsymbol{E}_1 \boldsymbol{L}_1 \boldsymbol{E}_1^T$ and $\boldsymbol{E}_2 \boldsymbol{L}_2 \boldsymbol{E}_2^T$ respectively, where $\boldsymbol{E}_1^{-1} = \boldsymbol{E}_1^T$ and $\boldsymbol{E}_2^{-1} = \boldsymbol{E}_2^T$. Here, $\boldsymbol{E}_1, \boldsymbol{E}_2$ are matrices with orthonormal eigenvectors as columns and $\boldsymbol{L}_1, \boldsymbol{L}_2$ are the diagonal matrices of the corresponding eigenvalues. Note that

$$\begin{aligned}
\boldsymbol{B}_1 \otimes \boldsymbol{B}_2 + \lambda \boldsymbol{I}_N &= (\boldsymbol{E}_1 \boldsymbol{L}_1 \boldsymbol{E}_1^T) \otimes (\boldsymbol{E}_2 \boldsymbol{L}_2 \boldsymbol{E}_2^T) + \lambda \boldsymbol{E}_1 \boldsymbol{E}_1^T \otimes \boldsymbol{E}_2 \boldsymbol{E}_2^T && (19)\\
&= (\boldsymbol{E}_1 \otimes \boldsymbol{E}_2)(\boldsymbol{L}_1 \otimes \boldsymbol{L}_2)(\boldsymbol{E}_1^T \otimes \boldsymbol{E}_2^T) + \lambda (\boldsymbol{E}_1 \otimes \boldsymbol{E}_2)(\boldsymbol{E}_1^T \otimes \boldsymbol{E}_2^T) && (20)\\
&= (\boldsymbol{E}_1 \otimes \boldsymbol{E}_2)(\boldsymbol{L}_1 \otimes \boldsymbol{L}_2 + \lambda \boldsymbol{I}_N)(\boldsymbol{E}_1^T \otimes \boldsymbol{E}_2^T) && (21)
\end{aligned}$$

Since $(\boldsymbol{E}_1 \otimes \boldsymbol{E}_2)^{-1} = (\boldsymbol{E}_1^T \otimes \boldsymbol{E}_2^T)$, it is seen that $(\boldsymbol{B} + \lambda \boldsymbol{I}_N)^{-1}$ is given by the formula

$$(\boldsymbol{B} + \lambda \boldsymbol{I}_N)^{-1} = (\boldsymbol{E}_1 \otimes \boldsymbol{E}_2)(\boldsymbol{L}_1 \otimes \boldsymbol{L}_2 + \lambda \boldsymbol{I}_N)^{-1}(\boldsymbol{E}_1^T \otimes \boldsymbol{E}_2^T) \tag{22}$$

Denote $\boldsymbol{L}_1 = \mathrm{diag}\{L_{11}, \cdots, L_{1n}\}$ and $\boldsymbol{L}_2 = \mathrm{diag}\{L_{21}, \cdots, L_{2m}\}$. Then note that

$$(\boldsymbol{L}_1 \otimes \boldsymbol{L}_2 + \lambda \boldsymbol{I}_N)^{-1} = \mathrm{diag}\{(\lambda + L_{11}L_{21})^{-1}, \cdots, (\lambda + L_{1n}L_{2m})^{-1}\} \tag{23}$$

Thus, if the number of arithmetic operations to obtain the eigendecompositions of $\boldsymbol{B}_1$ and $\boldsymbol{B}_2$ are $O(n^3)$ and $O(m^3)$ respectively (see [32] Page 70), by using the derived formula (22), we can compute $(\boldsymbol{B} + \lambda \boldsymbol{I}_N)^{-1}(\widetilde{\boldsymbol{R}} - \mu \boldsymbol{1})$ and $\mathrm{diag}\{(\boldsymbol{B}+\lambda \boldsymbol{I})^{-1}\}^{-1}$ with $O(n^3+m^3+n^2m^2)$ arithmetic operations. On the contrary, we would need $O(N^3) = O(n^3m^3)$ arithmetic operations to compute $(\boldsymbol{B} + \lambda \boldsymbol{I}_N)^{-1}(\widetilde{\boldsymbol{R}} - \mu \boldsymbol{1})$ directly.

To summarize, we give the proposed algorithm for fitting the RBF metamodel in the following table.

Computationally efficient fitting procedure for RBF model

Input: $\widetilde{R}(\boldsymbol{d}^i), i = 1, \cdots, N$

Procedure:

Step 1. Set $\mu = \sum_{i=1}^{N} \widetilde{R}_i/N$ (see (16)).

Step 2. Choose $\boldsymbol{\gamma}, \lambda$ by minimizing the mean square LOO prediction error as in (13), with the error vector calculated by (14) and (22).

Step 3. Calculate $\boldsymbol{\beta} = (\boldsymbol{B} + \lambda \boldsymbol{I})^{-1}(\widetilde{\boldsymbol{R}} - \mu \boldsymbol{1})$, where $\widetilde{\boldsymbol{R}} = (\widetilde{R}_1, ..., \widetilde{R}_N)'$ and $(\boldsymbol{B} + \lambda \boldsymbol{I})^{-1}$ is as given in (22).

After fitting the RBF model, predictions of $R(\boldsymbol{x}, \boldsymbol{\theta})$ for all $(\boldsymbol{x}, \boldsymbol{\theta}) \in \mathcal{X} \times \Theta$ can be made through equation (17). Based on the predictions $\hat{R}(\boldsymbol{x}, \boldsymbol{\theta})$, the optimal solution $\boldsymbol{x}^*(\boldsymbol{\theta})$ optimizing $\hat{R}(\boldsymbol{x}, \boldsymbol{\theta})$ over all $\boldsymbol{x} \in \mathcal{X}$ for any fixed $\boldsymbol{\theta} \in \Theta$ can be obtained.

Due to randomness in the computer model output, constructing confidence intervals for estimates obtained with the proposed RBF metamodel is important for assessing the amount of uncertainty in the estimates. Denote the scalar population quantity of interest as Υ (for example, $R(\boldsymbol{x}, \boldsymbol{\theta})$ in (5) or the components of $\boldsymbol{x}^*(\boldsymbol{\theta})$ in equation (8)) and its estimator as $\hat{\Upsilon}$. To measure the uncertainty associated with $\hat{\Upsilon}$, approximate confidence intervals can be constructed via the bootstrap procedure in Appendix B.

4. Case study: design of drug delivery system

4.1. A computer model of drug release from polymer matrix devices

With the constantly upgraded computing power, computer models have recently been used extensively to understand the drug release process from polymeric devices or drug-eluting stents, see [33, 15, 5]. Optimizing the drug release process generally involves many conflicting goals: maximizing the drug therapy effect, minimizing the side effect and minimizing the cost of the drug, etc. For example, [33, 5] reported their pioneering attempts at including multiple design objectives for optimizing the design of drug-eluting stents.

To demonstrate the application of our approach, we use the design and modeling of drug release from a polymer matrix (PM) ([41] page 340) as a test example. An illustration of the geometry of the PM, where the PM radius is x (cm) and the PM length is z_L (cm), and boundary conditions for the mathematical model for drug release is given in Figure (1). A certain amount of drug is initially administered in the PM, and the modeling problem is to determine how fast the drug will leave the PM and enter the surrounding tissue treated by the drug. Within the PM, the movement of the drug is modeled by diffusion equations, and the transfer rate to the surrounding tissue is described by a

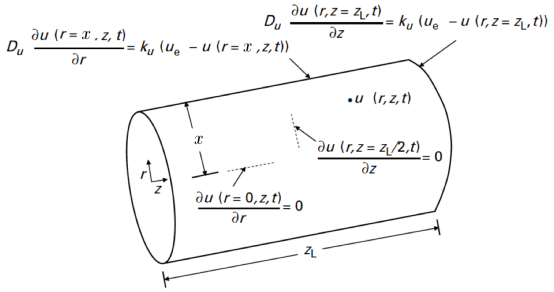


Figure 1: Diagram of a drug diffusion system

mass transfer coefficient. The partial differential equation describing the diffusion in cylindrical coordinates (r, z, φ), where r is the radial distance, z is the height, and φ is the angular coordinate, is given by

$$\frac{\partial u}{\partial t} = D_u\left(\frac{\partial^2 u}{\partial r^2} + \frac{1}{r}\frac{\partial u}{\partial r} + \frac{\partial^2 u}{\partial z^2}\right), \tag{24}$$

where t denotes time, D_u is the drug diffusivity, which is a specified constant, and u is the drug concentration. Note that because the external drug concentration is assumed to be uniform, the system in (24) is independent of the angular coordinate φ. Thus, the solution of (24) is a function of (r, z, t), i.e., $u = u(r, z, t)$, which gives the drug concentration at location (r, z) after a release time of t. Equation (24) is first order in t and second order in r and z. Thus, one initial condition (IC) in t and two boundary conditions (BCs) in r and z are required. The IC is:

$$u(r, z, t = 0) = u_0 \tag{25}$$

where u_0 is a specified constant and represents the initial value of u.

The homogeneous Neumann BC ([41] page 113) for equation (24) at $r = 0$ is used to specify symmetry in r:

$$\frac{\partial u(r = 0, z, t)}{\partial r} = 0 \tag{26}$$

and the BC at the exterior surface $r = x$ is based on a mass transfer coefficient k_u:

$$D_u \frac{\partial u(r = x, z, t)}{\partial r} = k_u(u_e - u(r = x, z, t)) \tag{27}$$

where u_e is the external concentration of drug. This equation equate the mass fluxes at the polymer surface $r = x$ to the fluxes due to concentration differences at $r = x$. Since the external biological environment varies from patient to patient, both u_e and k_u are subject to random fluctuations. Similarly, the BC at $z = z_L/2$ reflects symmetry in z and

the BC at $z = z_L$ specify the fluxes at the bottom of the polymer:

$$\frac{\partial u(r, z = z_L/2, t)}{\partial z} = 0, \qquad D_u \frac{\partial u(r, z = z_L, t)}{\partial z} = k_u(u_e - u(r, z = z_L, t)) \tag{28}$$

Equations (24) to (28) constitute equations solved by the computer model for the system illustrated in Figure (1). The partial differential equation (PDE) (24) with initial and boundary conditions given in (25) to (28) is solved by the method of lines (MOL), which proceeds by first discretizing the spatial derivatives and leaving the time variable continuous. This leads to a system of ordinary differential equations (ODEs) to which a numerical method can be applied. In this problem, the PDE is solved by discretizing r to 11 grid points over the range $0 \leqslant r \leqslant x$ and z to 11 grid points over the range $z_L/2 \leqslant z \leqslant z_L$. In other words, a system of 11×11 ODEs that approximates the PDE is used. More detailed description of the computer model can be found in Chapter 7 of [41].

To conclude the description of this stochastic computer model, we summarize the model design variable, output variable, random fluctuation variables, and the constants in Table (1). The value of these variables and parameters are specified following Section 7.1.1 of [41]. Note that values for concentrations are normalized, i.e., $0 \leqslant u_0, u_e \leqslant 1$, which facilitates their interpretation (e.g., their departure from one). The variable u_e is the exterior drug concentration value which represents the remaining drug concentration from the previously administered dose. It is set to be random due to differences in individual patient metabolism. In addition, the mass transfer coefficient k_u depends in part on the characteristics of the patient's tissue and in vivo environment surrounding the PM. Thus, we also model k_u as a random variable.

Table 1: Summary of the variables and parameters of the drug release stochastic computer model

Variable, parameter	Interpretation	Type
$u(r, z, t) \in [0, 1]$	concentration of drug at time t and location (r, z)	output variable
$x \in [0.5, 1.5]$	PM radius (cm)	design variable
$u_e \sim$ uniform $(0, 0.5)$	exterior drug concentration value (normalized)	random variable
$z_L = 2$	PM length (cm)	constant
$u_0 = 1$	initial drug concentration value (normalized)	constant
$D_u = 1.0e-06$	diffusivity (cm^2/ s)	constant
$k_u \sim$ uniform $(0.05, 0.15)$	mass transfer coefficient (cm/s)	random variable

4.2. Objective function for drug design

In this example, we focus on minimizing the cost of drug while keeping the effect of drug therapy close to the desired effect. This joint consideration is typical in drug design (see [28]). The cost of drug is measured by the amount of drug initially administered to the system $K_1(x) = \pi x^2 z_L u_0$. On the other hand, the effect of drug therapy is measured by the difference between the desired (target) drug release profile and the actual release profile after $t = 24$

and $t = 48$ hours of the initial drug administration, where the actual total amount of the drug that has been released from the PM at time t is (see [41] page 359).

$$Y_t(x, \boldsymbol{\xi}) = \pi x^2 z_L u_0 - 2 \times 2\pi \int_{z_L/2}^{z_L} \int_0^x u(r, z, t) r dr dz \tag{29}$$

Note that $Y_t(x, \boldsymbol{\xi})$ depends on x and $\boldsymbol{\xi} = (u_e, k_u)'$ through the PDE solution $u(r, z, t)$. The output of the stochastic computer model is $\boldsymbol{Y}(x, \boldsymbol{\xi}) = (Y_{24}(x, \boldsymbol{\xi}), Y_{48}(x, \boldsymbol{\xi}))'$. Translating these goals to desirability functions, a smaller-the-better function (2) is used for the cost measure with values $A = 0.1$ and $B = 15$, i.e.,

$$J_1^{min}(\boldsymbol{Y}, x, \theta_1) = \begin{cases} 0 & \text{if } K_1(x) > 15 \\ [\frac{K_1(x)-15}{0.1-15}]^{\theta_1} & \text{if } 0.1 \leqslant K_1(x) \leqslant 15 \\ 1 & K_1(x) < 0.1 \end{cases} \tag{30}$$

for the cost $K_1(x)$, where $\theta_1 > 0$ is an uncertain parameter.

Next, target oriented desirability function (3) is used for $Y_{24}(x, \boldsymbol{\xi})$ with $A = \theta_2$, $T_0 = 3.5$, $B = 12$, and $Y_{48}(x, \boldsymbol{\xi})$ with $A = 1.2$, $T_0 = 7$, $B = 13$, i.e.,

$$J_2^{target}(\boldsymbol{Y}, x, \theta_2, \theta_{21}, \theta_{22}) = \begin{cases} [\frac{Y_{24}-\theta_2}{3.5-\theta_2}]^{\theta_{21}} & \text{if } \theta_2 \leqslant Y_{24} \leqslant 3.5 \\ [\frac{Y_{24}-12}{3.5-12}]^{\theta_{22}} & \text{if } 3.5 \leqslant Y_{24} \leqslant 12 \\ 0 & \text{otherwise} \end{cases} \tag{31}$$

$$J_3^{target}(\boldsymbol{Y}, x, \theta_{31}, \theta_{32}) = \begin{cases} [\frac{Y_{48}-1.2}{7-1.2}]^{\theta_{31}} & \text{if } 1.2 \leqslant Y_{48} \leqslant 7 \\ [\frac{Y_{48}-13}{7-13}]^{\theta_{32}} & \text{if } 7 \leqslant Y_{48} \leqslant 13 \\ 0 & \text{otherwise} \end{cases} \tag{32}$$

where the θ's are uncertain parameters. Thus, we can write the overall desirability score as

$$J(\boldsymbol{Y}, x, \theta_1, \theta_2, \theta_{21}, \theta_{22}, \theta_{31}, \theta_{32}) = [J_1^{min}(\boldsymbol{Y}, x, \theta_1) \times J_2^{target}(\boldsymbol{Y}, x, \theta_2, \theta_{21}, \theta_{22}) \times J_3^{target}(\boldsymbol{Y}, x, \theta_{31}, \theta_{32})]^{1/3} \tag{33}$$

Appropriate values of the preference parameters $\theta_1, \theta_2, \theta_{21}, \theta_{22}, \theta_{31}, \theta_{32}$ are often hard to determine, which makes them uncertain parameters .

4.3. Optimization results and discussion

In this section, we describe the computer model and design optimization based on the proposed RBF metamodel. The control variable is the PM radius x(cm) restricted to the interval $[0.5, 1.5]$, and the vector of model random fluctuation variables is $\boldsymbol{\xi} = (u_e, k_u)'$, where u_e is the normalized exterior drug concentration value and k_u is the mass transfer coefficient, which are assumed to be uniformly distributed on $[0, 0.5]$ and $[0.05, 0.15]$ respectively. The MOL numerical method used to solve the PDE gives the drug concentration $u(r, z, t)$ at time $t = 24, 48$h and at positions (r, z) on the 11×11 uniform grid on $\{(r, z) : 0 \leqslant r \leqslant x, z_L/2 \leqslant z \leqslant z_L\}$. These values are then employed to compute the computer model outputs Y_{24} and Y_{48} via a quadrature scheme based on approximating the restriction of the integrand to each cell (smallest rectangles with grid points as corners) as a function of the form $\alpha_0 + \alpha_1 r + \alpha_2 z + \alpha_{12} rz$ with coefficients $\alpha_0, \alpha_1, \alpha_2, \alpha_{12}$ chosen so that it interpolates the integrand values at the corners of the cell. For the objective function described in (33), we assume that $\theta_{21} = 2, \theta_{22} = 1, \theta_{31} = 2, \theta_{32} = 1$ are fixed parameters.

We perform two numerical simulation studies to illustrate the proposed methodology. In the first case, we first fix $\theta_2 = 1$ and treat θ_1 as an uncertain parameter to be adjusted within the range $[0.5, 3]$. We use a 5×21 -point uniform grid Cartesian product design $\widetilde{\mathcal{D}}$ given by

$$\widetilde{\mathcal{D}} = \{0.5, 0.75, \ldots, 1.5\} \times \{0.5, 0.625, \ldots, 3\} \tag{34}$$

over the space of control variable x and uncertain parameter θ_1. At each design point $x \in \{0.5, 0.75, 1, 1.25, 1.5\}$, $M = 5$ computer model replications are conducted and the stochastic output $\boldsymbol{Y}(x, \boldsymbol{\xi}) = (Y_{24}(x, \boldsymbol{\xi}), Y_{48}(x, \boldsymbol{\xi}))'$ is obtained for each replication. The sample average of the desirability score (33) is then computed for each design point $(x^v, \theta_1^l) \in \widetilde{\mathcal{D}}$ as $\widetilde{R}(x^v, \theta_1^l) = \sum_{j=1}^{M} J(\boldsymbol{Y}_j(x^v, \boldsymbol{\xi}), x^v, \theta_1^l)/M$. Recall that $\boldsymbol{Y}_j(x^v, \boldsymbol{\xi})$ is the observed computer output in the jth replication at input point x^v and we have dropped the fixed θ's from the argument of the overall desirability score J given by (33). We use the RBF model in (17) to predict $R(x, \theta_1)$ at all values of $x \in [0.5, 1.5]$ and $\theta_1 \in [0.5, 3]$.

The desirability scores $J(\boldsymbol{Y}_j(x^v, \boldsymbol{\xi}), x^v, \theta_1^l)$ for the five replicates $j = 1, \ldots, 5$ at each design point are plotted and connected with a vertical line in Figure (2). It is seen that there are likely no outliers in our data but in general, outliers, which indicate possible instability of the computer model, can be identified by various methods. However, the variability is not negligible and prediction intervals constructed using the bootstrap procedure in Appendix B can help assess the uncertainty in the predictions $\hat{R}(x, \theta_1)$ of the RBF model due to variability in the computer output. Note that the term 'prediction' is used herein to refer to estimation of the expected desirability score at a given design point instead of prediction of a single desirability value computed from one replication of the computer model output at a design point, which is an uninteresting problem to the decision maker. To check the accuracy of the RBF model's

predictions and coverage of the prediction intervals given by the bootstrapping procedure, we compare the model predictions in a large test set, i.e., a 21×21 uniform grid over the design space $[0.5, 1.5] \times [0.5, 3]$, with the 'true value'. Specifically, the 'true value' is an accurate estimate of the expected desirability score based on the average of $M = 100$ computer model replications. Figure (3) plots the predicted desirability values and the 'true value' versus the grid point label, where the label is obtained by numbering the grid points according to the order it is encountered as we move from point to point, each time taking a step to an adjacent grid point. As we can see, the curve of predicted desirability score is very close to the plotted accurate estimates (or 'true value') of the expected desirability score, suggesting that the RBF regression model has very good prediction ability. The plotted 98% prediction intervals are narrow and contain all the 'true values'.

We compare the accuracy of the proposed RBF model with the support vector regression (SVR) method introduced

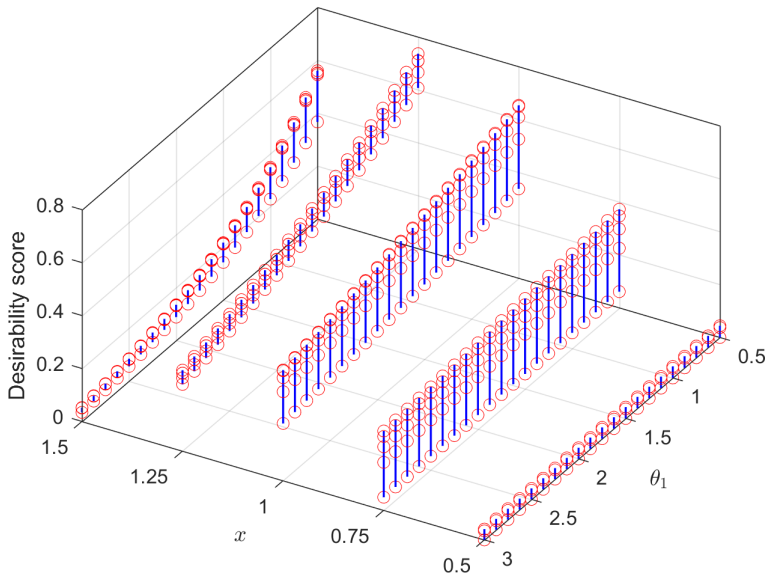


Figure 2: Plot of desirability score for each replicate versus (x, θ_1); vertical lines connect the five desirability scores (circles) for each (x, θ_1) point

in [12]. The polynomial kernel introduced in [12] with input vector (x, θ_1) standardized so that it takes on value in $[0, 1]^2$ is employed. Matlab 2017b built-in function fitrsvm is used to fit the SVR model. Given the fixed form of the kernel, the hyperparameters 'BoxConstraint', 'KernelScale', and 'Epsilon' are optimized while the polynomial degree is fixed at three. Matlab uses five-fold cross-validation to optimize the hyperparameters, where the experiment data is randomly partitioned into five equal-sized subsamples. Due to variability in the partitioning of the data and the Bayesian search method for optimizing the hyperparameters, we observe large variability in the optimized

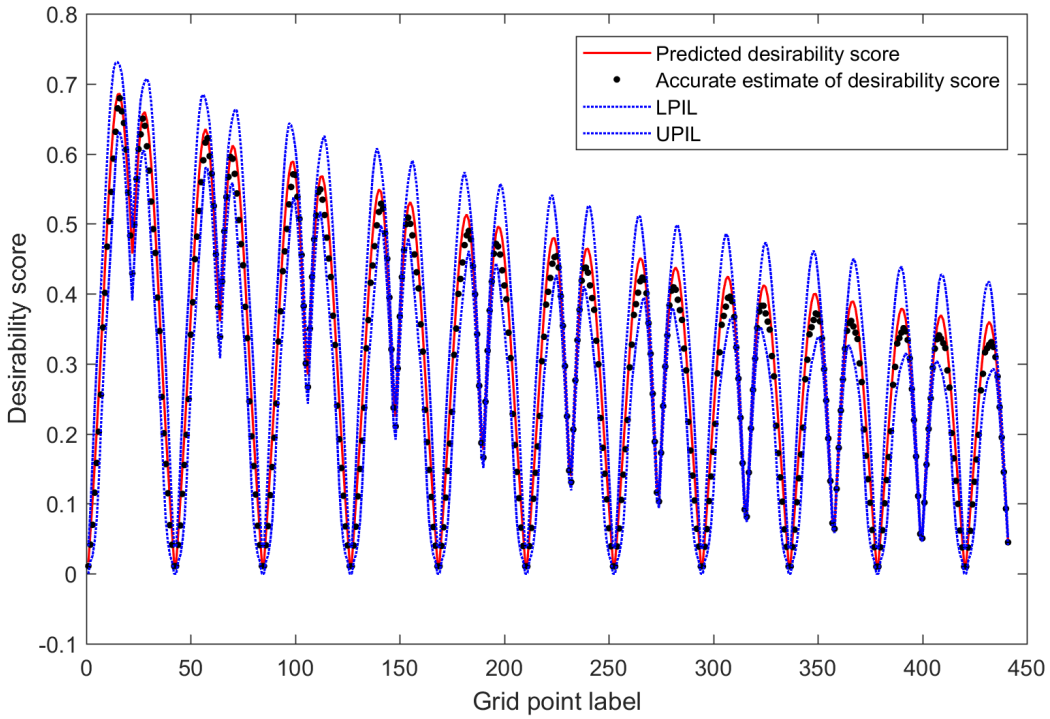
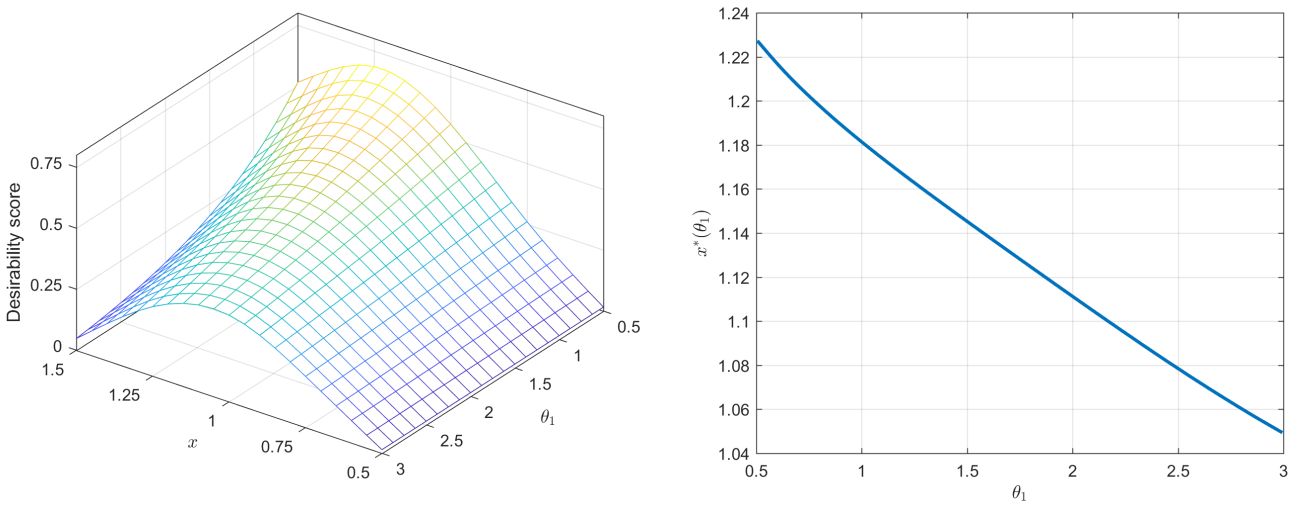


Figure 3: Point predictions and 98% prediction intervals of desirability score given by RBF model, and accurate estimates of desirability score (based on 100 replicates) versus grid point label. Upper and lower prediction interval limits are labeled as UPIL and LPIL respectively.

hyperparameters and resulting model predictions. Thus, we repeated the model fitting 10 times, giving 10 SVR models. The prediction mean squared error (average of the squared difference between the prediction minus the 'true value') of the 10 models on the 21×21 uniform grid test set range from 0.00110 to 0.02627 with a mean of 0.00793. In contrast, the prediction mean squared error of the proposed RBF model is only 0.000931. Moreover, the total time taken to fit the 10 SVR models and computing their predictions on the test set on a HP Z640 workstation with 64GB RAM and Intel Xeon E5-2650 v4 @ 2.2GHz processor with 12 cores is 661 seconds. In contrast, the time taken to fit the proposed RBF model and computing its predictions on the test set is only 0.67 seconds. Note that as the proposed RBF model is the posterior mean of a GP model with correlation function (11) (see [40]), it will tend to work very well when $R(\boldsymbol{x}, \boldsymbol{\theta})$ can be modeled as a realization of a Gaussian process. This and other theoretical support (e.g., error bounds in Chapter 5 of [6]) give us reassurance that the model can work well in a wide variety of problems.

We plot the predicted desirability score surface in Figure (4a). As can be seen from this plot, using the proposed Cartesian product design and the RBF metamodel, we can predict the expected desirability score as a smooth function of both the control variable x and the uncertain preference parameter θ_1. Based on this prediction, we can conveniently obtain the optimal control variable setting $x^*(\theta_1)$ (see 8) given any fixed $\theta_1 \in [0.5, 3]$, while avoiding the burden of refitting the RBF metamodel every time θ_1 changes. In Figure (4b), we plot $x^*(\theta_1)$ versus $\theta_1 \in [0.5, 3]$. In this plot, the



(a) (b)

Figure 4: (a) Point prediction of expected desirability score given by RBF model; (b) estimated optimal solution versus θ_1

optimal setting $x^*(\theta_1)$ changes significantly with θ_1, which supports the argument that it is necessary to analyze the dependency of the optimal solution on the uncertain objective function parameters. In this drug delivery example, θ_1 determines the impact of the cost of the drug on the desirability score. When multiple optimal solutions are available for different values of θ_1, high-level information can be used to choose one of the optimal solutions. For example, if the drug is expensive, the investigator may want to choose a large value for θ_1 to heavily punish high amounts of initial drug $K_1(x)$. The calculation shows that all the optimal solution $x^*(\theta_1)$ fall into the range $[1.05, 1.23]$. If based on experience, x values in the range $[1.05, 1.23]$ are generally acceptable, one can comfortably select a large value of θ_1 without worrying about the potential unsatisfactory treatment effect.

In cases where the qualitative/higher-level information may not be available to the investigator, an attractive method is to select the preference parameters using a quantitative procedure. In Definition 1, we proposed such a solution to select the most 'robust' optimal setting $x^*(\theta_1)$. Now we illustrate the usefulness of this solution in this practical problem. In Figure 5, we plot the robustness measure $Q(\theta_1) = \int_{\vartheta\in[0.5,3]} \frac{\hat{R}[x^*(\theta_1),\vartheta]}{\hat{R}[x^*(\vartheta),\vartheta]} d\vartheta / \int_{\vartheta\in[0.5,3]} d\vartheta$ versus θ_1, where $\hat{R}(x,\vartheta)$ is the RBF model prediction. It can be seen that the robustness of $x^*(\theta_1)$ reaches its peak at $\theta_1 = 1.765$ with $x^*(1.765) = 1.127$ and the average ratio of the expected desirability score of this control variable setting to the optimum expected desirability score is a very high value of 0.9875. Therefore, choosing $\theta_1 = 1.765$ as the preference parameter value would produce an optimal solution that performs well under other choices of θ_1. This robust optimal solution can also serve as a compromised solution when there are many different decision makers/patients, and different preference parameter should be used to represent their specific trade-off preference.

In the second case, we treat $\boldsymbol{\theta} = (\theta_1, \theta_2)$ as an uncertain parameter vector to be adjusted within the region

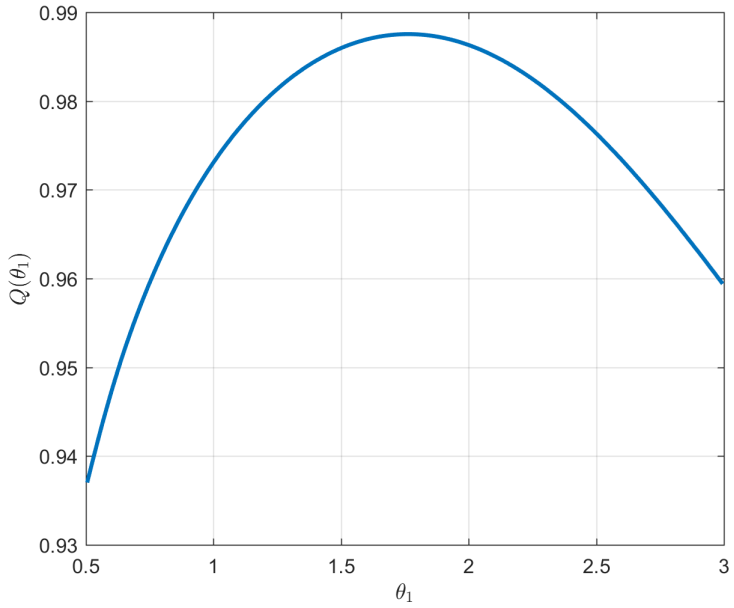
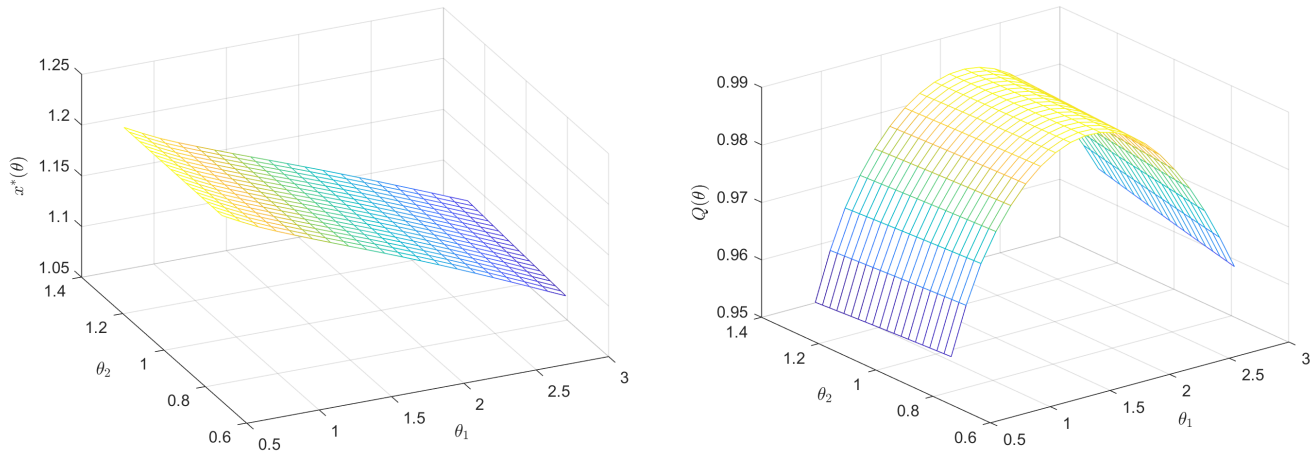


Figure 5: Plot of robustness measure $Q(\theta_1)$ versus θ_1

$[0.5, 3] \times [0.75, 1.25]$. We use a $5 \times 11 \times 6$ -point Cartesian product design $\widetilde{\mathcal{D}}$ over the space of control variable x and uncertain parameter $\boldsymbol{\theta}$, i.e.,

$$\widetilde{\mathcal{D}} = \{0.5, 0.75, \ldots, 1.5\} \times \{0.5, 0.75, \ldots, 3\} \times \{0.75, 0.85, \ldots, 1.25\} \tag{35}$$

Note that (22) can be generalized to exploit the fact that $\widetilde{\mathcal{D}}$ can be written as a Cartesian product of three sets (which can further reduce the computational cost in fitting the RBF model). At each design point $x \in \{0.5, 0.75, 1, 1.25, 1.5\}$, $M = 10$ computer model replications are conducted. We use the RBF model in (17) to predict $R(x, \boldsymbol{\theta})$ at all values of $x \in [0.5, 1.5]$ and $\boldsymbol{\theta} \in [0.5, 3] \times [0.75, 1.25]$. We plot $x^*(\boldsymbol{\theta})$ versus $\boldsymbol{\theta}$ in Figure (6a) and the robustness measure $Q(\boldsymbol{\theta})$ versus $\boldsymbol{\theta}$ in Figure (6b). It can be seen that the optimal setting $x^*(\boldsymbol{\theta})$ changes significantly with θ_1 but not θ_2. Similarly, $Q(\boldsymbol{\theta})$ changes significantly with θ_1 but not θ_2. Thus, θ_1 has a major impact on the optimal solution and selection of robust optimal solution while uncertainty in θ_2 can be ignored. There are some noticeable but small discrepancies between Figure (4b) and the values given by Figure (6a) for $\theta_2 = 1$, and also between Figure (5) and the values given by Figure (6b) for $\theta_2 = 1$. This is due to variability in the computer model output.



(a) (b)

Figure 6: (a) Estimated optimal solution versus $\boldsymbol{\theta}$; (b) plot of robustness measure $Q(\boldsymbol{\theta})$ versus $\boldsymbol{\theta}$

5. Conclusion

In this work, we explored metamodel-based optimization of stochastic computer models subject to an objective function with uncertain parameters. We presented typical scenarios where the objective function is uncertain and provided the corresponding uncertainty quantification techniques. We developed an RBF metamodel based optimization method through predicting the objective function as a function of both the design variables and the uncertain parameters in the objective function. By using a design that is a Cartesian product of points in the design variable space and the uncertain parameter space, we developed a fast fitting algorithm to construct the RBF metamodel. These tools provide the system designers more flexibility in making an informed and rational decision than the traditional choice of replacing the uncertain parameter by some estimates.

Our proposed methodology is general. It can be applied to any time consuming stochastic simulation model with multiple outputs to be optimized via the desirability function approach or other approaches for multiobjective optimization. In particular, it can be applied to simulation models based on PDEs with randomly drawn inputs, with the drug delivery system design case study given in this paper being one example. In this type of problems, a PDE model for a physical system is solved numerically with the finite element or other methods implemented with a computer code. Due to uncertainties in some inputs to the model, they are modeled as random variables and their values are randomly chosen in the simulations. Thus, the outputs of the computer code are random, which necessitates the use of replicated simulations to quantify the distribution of the outputs. This type of models is widely used in mechanical and civil engineering design to account for input uncertainty. It has been implemented in the popular ANSYS software for

finite element simulation and engineers are often interested in using such software for optimizing multiple quantifiable characteristics of an engineering design under input uncertainty [38]. According to [38], ANSYS has built-in capability to construct polynomial emulators (called response surface method) to help reduce simulation time. Recently, other researchers such as [42] have considered the use of Gaussian process emulators. However, it appears that objective function uncertainty has not been systematically considered by these engineering design researchers. Our proposed method can be a useful tool for these researchers to optimize multiple characteristics of an engineering design under both input and objective function uncertainty. This includes design of manufacturing processes using finite element simulation models [27]. In addition, our proposed approach can also be applied to optimize discrete event stochastic simulation models in operations research ([1]; [24]).

Finally, although the proposed methodology is for predicting one overall objective function with uncertain parameters, it can be easily extended to predict multiple objective functions. For example, in a multi-objective optimization problem, if the manager needs the individual objective function values to guide decision-making, independent metamodels can be used to predict each of the objective functions.

Acknowledgments: The authors thank two referees for comments that helped improve the article. Guilin Li's research was supported by project T32-101/15-R funded by the Research Grants Council of Hong Kong. Matthias Tan's research was supported by General Research Fund projects 11226716 and 11201117 funded by the Research Grants Council of Hong Kong.

Appendix A. Computation of robust preference parameter

We consider the common case where Θ is a hypercube, i.e., $\Theta = \prod_{i=1}^{q}[l_i, u_i]$. We construct a grid on Θ given by

$$\mathcal{G} = \prod_{i=1}^{q}\{l_i + \Delta_i/2, l_i + 3\Delta_i/2, \ldots, u_i - \Delta_i/2\}, \tag{A.1}$$

where $u_i - \Delta_i/2 = l_i + (n_i - 1/2)\Delta_i$ and n_i is some positive integer. The grid points in $\mathcal{G}$ are the centroids in a partition of Θ into hypercubes/cells of side lengths $\Delta_1, \Delta_2, \ldots, \Delta_q$ and the total number of grid points is $\prod_{i=1}^{q} n_i$. For each $\boldsymbol{\vartheta} \in \mathcal{G}$, we use a numerical optimization routine in Matlab (patternsearch to be specific) to optimize $R(\boldsymbol{x}, \boldsymbol{\vartheta})$ (or more precisely, the RBF model (17) of $R(\boldsymbol{x}, \boldsymbol{\vartheta})$) to find $\boldsymbol{x}^*(\boldsymbol{\vartheta})$ and the optimal desirability score $R[\boldsymbol{x}^*(\boldsymbol{\vartheta}), \boldsymbol{\vartheta}]$. Then, for each $\boldsymbol{\theta} \in \mathcal{G}$, we estimate $Q(\boldsymbol{\theta})$ with

$$\hat{Q}(\boldsymbol{\theta}) = \sum_{\boldsymbol{\vartheta} \in \mathcal{G}} \frac{R[\boldsymbol{x}^*(\boldsymbol{\theta}), \boldsymbol{\vartheta}]}{R[\boldsymbol{x}^*(\boldsymbol{\vartheta}), \boldsymbol{\vartheta}]} \Big/ \prod_{i=1}^{q} n_i. \tag{A.2}$$

This corresponds to using the product midpoint rule ([8], page 591) to estimate the multidimensional integral in the definition of $Q(\boldsymbol{\theta})$. The value of $\boldsymbol{\theta} \in \mathcal{G}$ that maximizes $\hat{Q}(\boldsymbol{\theta})$ is taken to be the robust preference parameter value.

Appendix B. Constructing confidence intervals via the bootstrap method

---

Input: $S_n = \{\boldsymbol{Y}_i = [\boldsymbol{Y}_1(\boldsymbol{x}^i, \boldsymbol{\xi}), \cdots, \boldsymbol{Y}_M(\boldsymbol{x}^i, \boldsymbol{\xi})]'\}_{i=1}^n$

Procedure:

Step 1. Compute the estimator $\hat{\Upsilon}$ based on S_n.

Step 2. Generate an empirical bootstrap sample, by taking random samples with replacement from S_n :

$$S_n^* = \{\boldsymbol{Y}_i^* = [\boldsymbol{Y}_1^*(\boldsymbol{x}^i, \boldsymbol{\xi}), \cdots, \boldsymbol{Y}_M^*(\boldsymbol{x}^i, \boldsymbol{\xi})]'\}_{i=1}^n,$$

where $\boldsymbol{Y}_j^*(\boldsymbol{x}^i, \boldsymbol{\xi}) \in \{\boldsymbol{Y}_1(\boldsymbol{x}^i, \boldsymbol{\xi}), \cdots, \boldsymbol{Y}_M(\boldsymbol{x}^i, \boldsymbol{\xi})\}$.

Step 3. Estimate Υ based on S_n^*, and denote the estimator as $\hat{\Upsilon}^*$.

Repeat *Step 2* and *Step 3* B^* times.

Step 4. Compute the bootstrap differences $\delta^* = \hat{\Upsilon}^* - \hat{\Upsilon}$. Sort these B^* values in increasing order and pick out the 0.975 and 0.025 quantiles. Set these values as $\delta^*_{.975}$ and $\delta^*_{.025}$.

Step 5. The estimated 95% bootstrap confidence interval for Υ is: $[\hat{\Upsilon} + \delta^*_{.025}, \hat{\Upsilon} + \delta^*_{.975}]$

References

[1] Alberto, I., Azcárate, C., Mallor, F., and Mateo, P. M. (2002). Optimization with simulation and multiobjective analysis in industrial decision-making: A case study. *European Journal of Operational Research*, 140(2):373–383.

[2] Ankenman, B., Nelson, B. L., and Staum, J. (2010). Stochastic kriging for simulation metamodeling. *Operations research*, 58(2):371–382.

[3] Barton, R. R. and Meckesheimer, M. (2006). Metamodel-based simulation optimization. *Handbooks in operations research and management science*, 13:535–574.

[4] Boyle, C. R. and Shin, W. S. (1996). An interactive multiple-response simulation optimization method. *IIE transactions*, 28(6):453–462.

[5] Bozsak, F., Gonzalez-Rodriguez, D., Sternberger, Z., Belitz, P., Bewley, T., Chomaz, J.-M., and Barakat, A. I. (2015). Optimization of drug delivery by drug-eluting stents. *PloS one*, 10(6):e0130182.

[6] Buhmann, M. D. (2003). *Radial basis functions: theory and implementations*, volume 12. Cambridge university press.

[7] Burke, E. K. and Kendall, G. (2013). *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques (page 403)*. Springer Publishing Company, Incorporated, 2nd edition.

[8] Dahlquist, G. and Björck, Å. (2008). Numerical methods in scientific computing, volume i. *Society for Industrial and Applied Mathematics*.

[9] Deb, K., Sinha, A., Korhonen, P. J., and Wallenius, J. (2010). An interactive evolutionary multiobjective optimization method based on progressively approximated value functions. *IEEE Transactions on Evolutionary Computation*, 14(5):723–739.

[10] Deb, K. and Sundar, J. (2006). Reference point based multi-objective optimization using evolutionary algorithms. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 635–642. ACM.

[11] Derringer, G. (1980). Simultaneous optimization of several response variables. *Journal of quality technology*, 12(4):214–219.

[12] Drucker, H., Burges, C. J., Kaufman, L., Smola, A. J., and Vapnik, V. (1997). Support vector regression machines. In *Advances in neural information processing systems*, pages 155–161.

[13] Elsayed, K. and Lacor, C. (2013). Cfd modeling and multi-objective optimization of cyclone geometry using desirability function, artificial neural networks and genetic algorithms. *Applied Mathematical Modelling*, 37(8):5680–5704.

[14] Fasshauer, G. and McCourt, M. (2015). *Kernel-based approximation methods using Matlab*, volume 19. World Scientific Publishing Co Inc.

[15] Ferreira, J. A., de Oliveira, P., and da Silva, P. (2012). Analytics and numerics of drug release tracking. *Journal of Computational and Applied Mathematics*, 236(15):3572–3583.

[16] Ghasemi, H., Rafiee, R., Zhuang, X., Muthu, J., and Rabczuk, T. (2014). Uncertainties propagation in metamodel-based probabilistic optimization of cnt/polymer composite structure using stochastic multi-scale modeling. *Computational Materials Science*, 85:295–305.

[17] Harrington, E. (1965). The desirability function. *Industrial quality control*, 21(10):494–498.

[18] Henderson, D. A., Boys, R. J., Krishnan, K. J., Lawless, C., and Wilkinson, D. J. (2009). Bayesian emulation and calibration of a stochastic computer model of mitochondrial dna deletions in substantia nigra neurons. *Journal of the American Statistical Association*, 104(485):76–87.

[19] Hernandez, A. F. and Grover, M. A. (2010). Comparison of sampling strategies for gaussian process models, with application to nanoparticle dynamics. *Industrial & Engineering Chemistry Research*, 50(3):1379–1388.

[20] Huang, D., Allen, T. T., Notz, W. I., and Zeng, N. (2006). Global optimization of stochastic black-box systems via sequential kriging meta-models. *Journal of global optimization*, 34(3):441–466.

[21] Jeong, H. and Shenoi, R. (2000). Probabilistic strength analysis of rectangular frp plates using monte carlo simulation. *Computers & Structures*, 76(1):219–235.

[22] Jeong, I.-J. and Kim, K.-J. (2009). An interactive desirability function method to multiresponse optimization. *European Journal of Operational Research*, 195(2):412–426.

[23] Kanda, J. and Shah, H. (1997). Engineering role in failure cost evaluation for buildings. *Structural Safety*, 19(1):79–90.

[24] Kleijnen, J. P. (2014). Simulation-optimization via kriging and bootstrapping: a survey. *Journal of Simulation*, 8(4):241–250.

[25] Korsakissok, I., Mathieu, A., and Didier, D. (2013). Atmospheric dispersion and ground deposition induced by the fukushima nuclear power plant accident: A local-scale simulation and sensitivity study. *Atmospheric environment*, 70:267–279.

[26] Liu, H., Wang, M., Wang, J., Zhang, G., Liao, H., Huang, R., and Zhang, X. (2007). Monte carlo simulations of gas flow and heat transfer in vacuum packaged mems devices. *Applied thermal engineering*, 27(2):323–329.

[27] Liu, S. C. and Hu, S. J. (1997). Variation simulation for deformable sheet metal assemblies using finite element methods. *Journal of manufacturing science and engineering*, 119(3):368–374.

[28] Lu, S., Fred Ramirez, W., and Anseth, K. S. (1998). Modeling and optimization of drug release from laminated polymer matrix devices. *AIChE journal*, 44(7):1689–1696.

[29] Marler, R. T. and Arora, J. S. (2010). The weighted sum method for multi-objective optimization: new insights. *Structural and multidisciplinary optimization*, 41(6):853–862.

[30] Marrel, A., Iooss, B., Da Veiga, S., and Ribatet, M. (2012). Global sensitivity analysis of stochastic computer models with joint metamodels. *Statistics and Computing*, 22(3):833–847.

[31] Miettinen, K. (2012). *Nonlinear multiobjective optimization*, volume 12. Springer Science & Business Media.

[32] O'Leary, D. P. (2009). *Scientific computing with case studies*. SIAM.

[33] Pant, S., Limbert, G., Curzen, N. P., and Bressloff, N. W. (2011). Multiobjective design optimisation of coronary stents. *Biomaterials*, 32(31):7755–7773.

[34] Papadrakakis, M. and Lagaros, N. D. (2002). Reliability-based structural optimization using neural networks and monte carlo simulation. *Computer methods in applied mechanics and engineering*, 191(32):3491–3507.

[35] Park, K. S. and Kim, K.-J. (2005). Optimizing multi-response surface problems: how to use multi-objective optimization techniques. *IIE Transactions*, 37(6):523–532.

[36] Plumlee, M. and Tuo, R. (2014). Building accurate emulators for stochastic simulations via quantile kriging. *Technometrics*, 56(4):466–473.

[37] Rajan, A., Kuang, Y. C., Ooi, M. P.-L., and Demidenko, S. N. (2016). Moment-based measurement uncertainty evaluation for reliability analysis in design optimization. In *Instrumentation and Measurement Technology Conference Proceedings (I2MTC), 2016 IEEE International*, pages 1–6. IEEE.

[38] Reh, S., Beley, J.-D., Mukherjee, S., and Khor, E. H. (2006). Probabilistic finite element analysis using ansys. *Structural Safety*, 28(1):17–43.

[39] Ristow, A., Begović, M., and Rohatgi, A. (2005). Modeling the effects of uncertainty and reliability on the cost of energy from pv systems. Georgia Institute of Technology.

[40] Santner, T. J., Williams, B. J., and Notz, W. I. (2013). *The design and analysis of computer experiments*. Springer Science & Business Media.

[41] Schiesser, W. E. (2012). *Partial differential equation analysis in biomedical engineering: case studies with MATLAB*. Cambridge University Press.

[42] Su, G., Peng, L., and Hu, L. (2017). A gaussian process-based dynamic surrogate model for complex engineering structural reliability analysis. *Structural Safety*, 68:97–109.

[43] Tan, M. H. Y. (2015). Stochastic polynomial interpolation for uncertainty quantification with computer experiments. *Technometrics*, 57(4):457–467.

[44] Tan, M. H. Y. (2016). Monotonic quantile regression with bernstein polynomials for stochastic simulation. *Technometrics*, 58(2):180–190.

[45] Trautmann, H. and Mehnen, J. (2009). Preference-based pareto optimization in certain and noisy environments. *Engineering Optimization*, 41(1):23–38.

[46] Van Beers, W. C. and Kleijnen, J. P. C. (2003). Kriging for interpolation in random simulation. *Journal of the Operational Research Society*, 54(3):255–262.

[47] Wallace, S. W. (2000). Decision making under uncertainty: Is sensitivity analysis of any use? *Operations Research*, 48(1):20–25.

[48] Wang, H., Li, E., and Li, G. (2011). Probability-based least square support vector regression metamodeling technique for crashworthiness optimization problems. *Computational Mechanics*, 47(3):251–263.

[49] Wang, H., Zeng, Y., Yu, X., Li, G., and Li, E. (2016). Surrogate-assisted bayesian inference inverse material identification method and application to advanced high strength steel. *Inverse Problems in Science and Engineering*, 24(7):1133–1161.

[50] Weng, T.-W., Melati, D., Melloni, A., and Daniel, L. (2017). Stochastic simulation and robust design optimization of integrated photonic filters. *Nanophotonics*, 6(1):299–308.

[51] Wu, F.-C. (2004). Optimization of correlated multiple quality characteristics using desirability function. *Quality engineering*, 17(1):119–126.

[52] Wurl, R. and Albin, S. (1999). A comparison of multiresponse optimization sensitivity to parameter selection. *Quality Engineering*, 11(3):405–415.

[53] Yalcınkaya, Ö. and Bayhan, G. M. (2009). Modelling and optimization of average travel time for a metro line by simulation and response surface methodology. *European Journal of Operational Research*, 196(1):225–233.

[54] Zhang, H., Jiang, Z., and Guo, C. (2009). Simulation-based optimization of dispatching rules for semiconductor wafer fabrication system scheduling by the response surface methodology. *The International Journal of Advanced Manufacturing Technology*, 41(1-2):110–121.

Guilin Li is currently a postdoctoral research fellow in the School of Data Science at City University of Hong Kong. She received a B.S. degree in Statistics from Nankai University, China, and a Ph.D. degree in Industrial Systems Engineering and Management from National University of Singapore. Her research interests, which lie broadly in the area of engineering statistics, include regression analysis, experimental design and uncertainty quantification.

Matthias Hwai Yong Tan is an assistant professor in the School of Data Science at City University of Hong Kong. He received his B.Eng. degree in Mechanical-Industrial Engineering from the Universiti Teknologi Malaysia, an M.Eng. degree in Industrial and Systems Engineering from the National University of Singapore and a Ph.D. degree in Industrial and Systems Engineering from Georgia Institute of Technology. His research interests include uncertainty quantification, design and analysis of computer experiments, and applied statistics.

Szu Hui Ng is an associate professor in the Department of Industrial Systems Engineering & Management at the National University of Singapore. She holds B.S., M.S. and Ph.D. degrees in Industrial and Operations Engineering from the University of Michigan. Her research interests include computer simulation modeling and analysis, design of experiments and quality and reliability engineering. She is a member of IEEE and INFORMS, and a senior member of IIE.