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

Guilin Li[†], Matthias Hwai-yong Tan[‡], Szu Hui Ng[†]

[†]Department of Industrial & Systems Engineering, National University of Singapore

[‡]Department of Systems Engineering and Engineering Management, City University of Hong Kong

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

Abstract

In the design of complex engineering systems, 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, such as uncertain trade-offs (weights) among conflicting goals or difficult-to-evaluate parameters. Due to lack of computationally efficient tools, the current solution to this problem is to replace the uncertain parameters by some best guess. More rational procedures include finding multiple trade-off optimal solutions such as Pareto-points or conducting sensitivity analysis on the difficult-to-evaluate parameters. However, adopting these procedures means that the stochastic computer model has to be optimized many times, which would incur extensive computational burden. In this work, we provide a computationally efficient solution to capture this uncertainty 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 variables and uncertain parameters spaces. 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 optimize a drug delivery system with conflicting goals.

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. The purpose of engineering design is to choose a set of settings for the design variables of the system such that the system performance is optimized. These tools have been recently successfully applied to many engineering design applications including the integrated photonic filters design in electrical engineering ([35]), aerospike nozzle design in aerospace engineering([32]), continuous stirred-tank (CSTR) reactor design in chemical engineering ([11]), etc.

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 [29, 26, 3]). When multiple objectives need to be optimized simultaneously, the most popular method is to form a composite objective function through weighted sum ([23]) or desirability functions ([36, 28]), where a weight or scale 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 ([21]). And 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 [17, 6, 29, 20, 35]. By such a choice, the uncertainty is pushed out of sight through approximating the uncertain parameters by some particular estimates, such as 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 a reliable preference or 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([37]).

In the optimization literature, an ideal procedure for a multi-objective problem is the *posteriori approach* ([5]), in which a set of different trade-off optimal solutions is first obtained and then a multiple-criterion decision-making technique (such as using high-level information) ([8]) is used to analyze the solutions to choose a most preferred solutions. 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 better position to balance 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) ([34]). 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 *sensitivity analysis* provide how sensitive the optimal solution is to the 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 by [5], 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* ([7, 2]), in which the decision maker's responses to specific questions were 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 efforts 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 proposed 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 variables and uncertain parameters spaces. In addition, based on the Cartesian product design structure, a fast fitting algorithm is also derived for fitting the metamodel. To illustrate the effectiveness of the proposed method for solving practical problems, we use the optimization of drug release from a polymer matrix ([31]) as a test example, where the developed tools are used to facilitate a robust selection of the scale parameters in the desirability function.

The remainder of the paper is organized as follows: Section 2 introduces some examples of uncertainty parameters in general classes of objective functions and stresses its importance in practical applications. Section 3 first introduces the 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 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 degree of deformation or the probability of damage (see [27, 9]). Using the *cost-benefit analysis* tools ([19]), 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 [19] is

$$C_T = C_I + C_F \int_0^1 \frac{\eta - \eta_0}{1 - \eta_0} P(\eta) d\eta \tag{1}$$

where C_I is the initial cost, $\eta \in [0, 1]$ 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 for different damage levels to happen which is the output of the computer model. In this objective function, the ultimate limit state failure cost C_F is an uncertain parameter which varies from 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 do not affect the computer model output $P(\eta)$. 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) ([34]) 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 decide how sensitive the optimal solution is to the change in the 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 $J_s(x), s = 1, \cdots, S$. For multi-objective optimization problems, 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) ([8]) 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 $x*$ optimizing the objective $J(x) = \sum_{s=1}^{S} w_s J_s(x)$ is Pareto-optimal if $w_s > 0, s = 1, \cdots, S$ and $\sum_{s=1}^{S} w_s = 1$. Under some assumptions, 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 ([24]). In this case, the weights $w_s > 0, s = 1, \cdots, S$ are uncertain parameters in the objective function $J(x)$, and a procedure allowing convenient optimization of $J(x)$ under different weight values would save considerable computational effort.

2.3. desirability function

The desirability function ([16, 10]) 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 objective functions, denoted as $J_s(\boldsymbol{x}), s = 1, \cdots, S$. For each of the S functions, a corresponding desirability score function is constructed which is high when $J_s(x)$ is at a desirable level (such as minimum, maximum, or target) and low when

$J_s(\boldsymbol{x})$ is at an undesirable level. For example, for minimization of $J_s(\boldsymbol{x})$, [10] construct the desirability function as

$$D_s^{min}(\boldsymbol{x};\theta_s) = \begin{cases} 0 & \text{if } J_s(\boldsymbol{x}) > B \\ [\frac{J_s(\boldsymbol{x})-B}{A-B}]^{\theta_s} & \text{if } A \leqslant J_s(\boldsymbol{x}) \leqslant B \\ 1 & J_s(\boldsymbol{x}) < A \end{cases} \tag{2}$$

where A, B, and the scale parameters θ_s are chosen by the investigator. For target-the-best situations, the desirability function is

$$D_s^{target}(\boldsymbol{x};\theta_s) = \begin{cases} [\frac{J_s(\boldsymbol{x})-A}{T_0-A}]^{\theta_{s1}} & \text{if } A \leqslant J_s(\boldsymbol{x}) \leqslant T_0 \\ [\frac{J_s(\boldsymbol{x})-B}{T_0-B}]^{\theta_{s2}} & \text{if } T_0 < J_s(\boldsymbol{x}) \leqslant B \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where $\theta_s = (\theta_{s1}, \theta_{s2})$ and T_0 is the target value chosen by the investigator. And for maximization, the desirability function is given in [36].

Given that the S desirability functions $D_1, \cdots, D_S$ are on the same scale($[0,1]$), they can be combined to produce an overall desirability function through the geometric mean:

$$D(\boldsymbol{x};\boldsymbol{\theta}) = [\prod_{s=1}^{S} D_s(\boldsymbol{x};\theta_s)]^{1/S} \tag{4}$$

where $\boldsymbol{\theta} = (\theta_1, \cdots, \theta_S)$. In this overall objective function $D(\boldsymbol{x};\boldsymbol{\theta})$, the scale parameters $\boldsymbol{\theta}$ have to be chosen such that the desirability measure is easier or harder to satisfy. For example, choosing $\theta_{s1} > 1$ in (3) would place more rewards on being close to the target value, and choosing $0 < \theta_{s1} < 1$ would make this less important.

This 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 [12, 39, 38]. However, to use this method, the investigator needs to choose values of the shape 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 scale parameters would possibly produce very different optimal solutions. Other than specifying these scale parameters subjectively, [18] proposed an *interactive approach* to select the values for these parameters. They proposed to first initialize the scale parameter and obtain an initial optimal solution, then interact with the decision maker to find his reaction to this solution and thereafter adjust the scale parameter value until some stopping criterion is satisfied. Although this method reduced the computational burden of optimizing the system repeatedly for many different scale parameter settings, it requires a lot of cognitive efforts of the decision maker.

An alternative solution to this uncertain scale parameter problem is to select the most 'robust' choice of the scale parameter. Specifically, for a particular choice of the scale parameter, one would obtain the corresponding optimal solution depending on $\boldsymbol{\theta}$ as $\boldsymbol{x}^*(\boldsymbol{\theta}) := \text{argmax}_{\boldsymbol{x}} D(\boldsymbol{x};\boldsymbol{\theta})$. To select the most 'robust' $\boldsymbol{\theta}$ from the many choices, or equivalently, to choose the most 'robust' corresponding $\boldsymbol{x}^*(\boldsymbol{\theta})$, a reasonable criterion is to measure how $\boldsymbol{x}^*(\boldsymbol{\theta})$ would perform under all the other possible choices of the scale parameter $\boldsymbol{\vartheta} \in \Theta$. Suppose the 'true' scale parameter value is $\boldsymbol{\vartheta}$, the maximum desirability score would be achieved at point $\boldsymbol{x} = \boldsymbol{x}^*(\boldsymbol{\vartheta})$ with value $D[\boldsymbol{x}^*(\boldsymbol{\vartheta});\boldsymbol{\vartheta}]$; while the desirability score of setting $\boldsymbol{x} = \boldsymbol{x}^*(\boldsymbol{\theta})$ would be $D[\boldsymbol{x}^*(\boldsymbol{\theta});\boldsymbol{\vartheta}]$. Hence, $\frac{D[\boldsymbol{x}^*(\boldsymbol{\theta});\boldsymbol{\vartheta}]}{D[\boldsymbol{x}^*(\boldsymbol{\vartheta});\boldsymbol{\vartheta}]}$ could be used to represent the efficiency of setting $\boldsymbol{x} = \boldsymbol{x}^*(\boldsymbol{\theta})$ when the 'true' scale parameter value is $\boldsymbol{\vartheta}$. By integrating this efficiency over all the possible values of $\boldsymbol{\vartheta} \in \Theta$, one can obtain the overall efficiency score of setting $\boldsymbol{x} = \boldsymbol{x}^*(\boldsymbol{\theta})$. Hence, we propose the following definition to select the robust scale parameter value for a desirability function.

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

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

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

By using this robustness definition, one can select a scale parameter and obtain the corresponding robust optimal solution without depending on subjective and qualitative information. Nonetheless, to obtain the robust choice of the scale parameter value, we need to optimize the computer model under every scale parameter choice $\boldsymbol{\theta} \in \Theta$. In such cases, an integrated and efficient solution to optimizing the time-consuming computer model repeatedly with many different scale 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 scale 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 and uncertain parameters spaces.

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 objective function, which 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 $\boldsymbol{\xi}$). The optimization problem(for simplicity, minimization is taken to be the standard) depending on $\boldsymbol{\theta}$ is then

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

When dealing with this uncertainty problem, the current development of 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

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

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 this as a family of problems where for each different $\boldsymbol{\theta} \in \Theta$, it yields an optimal value

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

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

3.2. Metamodel-based optimization under a fixed objective function

Frequently, computer codes of simulating 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 ([1]). Metamodels can be built using many regression models with a variety of prediction power. For example, simple linear regression is easy to built and has been used widely. Although simple, this model 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 simple spline functions can be fit efficiently. They are also applicable to problems with high dimensional design variable space 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 $R(\boldsymbol{x}_i, \boldsymbol{\theta})$ (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 $\{(\boldsymbol{x}_1, \widetilde{R}_1), \cdots, (\boldsymbol{x}_n, \widetilde{R}_n)\}$. 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

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

For simplicity of notation, we use $\widetilde{R}(\boldsymbol{x}_i, \boldsymbol{\theta})$ and $\widetilde{R}_i$ interchangeably. Although averaging over the replicated observations $\boldsymbol{y}_j(\boldsymbol{x}_i, \boldsymbol{\xi})$ 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. An advantage of such a treatment is that the regression form 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 form which was extensively studied by [4]. 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 singular, which induces a lot of computational problems. Hence, in this article, we adopt

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

which is a member of the class of Matern correlation functions([30]). Note that some authors ([13], 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 [4] (page 4) and call (11) a radial basis function 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. Define $\boldsymbol{\beta} = (\beta_1, \cdots, \beta_n)'$, $\boldsymbol{\gamma} = (\gamma_1, \cdots, \gamma_k)'$, $(\boldsymbol{B})_{ij} = \phi(\boldsymbol{x}_i - \boldsymbol{x}_j)$, 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 a $n \times n$ identity matrix.

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. And the above system of equations (12) for determining $\boldsymbol{\beta}$ with a nugget parameter is similar in spirit with the GP with 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 $\{(\boldsymbol{x}_1, \widetilde{R}_1), \cdots, (\boldsymbol{x}_n, \widetilde{R}_n)\}$. .

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

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

The advantage of LOO for linear models such as RBF regression is that the error vector $\boldsymbol{e} = (e_1, \cdots, e_n)'$ can be calculated analytically as (see [33] for proof)

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

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 $\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_K$ would be desired. In such cases, one has to refit the RBF metamodel (10) K times. The calculation of $(\boldsymbol{B} + \lambda \boldsymbol{I})^{-1}$ in (14) would take $O(n^3)$ arithmetic operations (see [25] Page 70), and hence the fitting of each $\hat{R}(\cdot, \boldsymbol{\theta}_i)$ would take $O(n^3)$ arithmetic operations. Thus, fitting/estimation of all $\hat{R}(\cdot, \boldsymbol{\theta}_1), \cdots, \hat{R}(\cdot, \boldsymbol{\theta}_K)$ would take $O(Kn^3)$ arithmetic operations. For purposes such as calculating the Pareto front, conducting sensitivity analysis or obtaining the robust scale parameter using Definition 1, the required RBF model fitting times could be quite large because K can be very large. Moreover, even if the investigator has the required computational budgets to fit the K RBF models, the predicted value of $\hat{R}(\bar{\boldsymbol{x}}, \boldsymbol{\theta})$ at a fixed design point $\bar{\boldsymbol{x}}$ could possibly be discontinuous over $\boldsymbol{\theta}$. One reason is the existence of many possible local optimizers of function (13) when estimating λ and $\boldsymbol{\gamma}$, which is an inevitable problem for fitting RBF and GP model 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 fast fitting solution based on the unique model structure of RBF. The fact that the RBF metamodel is linear in the vector of response data allows us to use the short-cut formula (14) to calculate the leave-one-out crossed validation error. Furthermore, the model form of RBF metamodel is characterized by *separability*: the RBF 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 space is a Cartesian product. This fact motivates us to collect data using a Cartesian product design in the $(\boldsymbol{x}, \boldsymbol{\theta})$ space: $\varOmega = \{(\boldsymbol{x}_1, \boldsymbol{\theta}_1), \cdots, (\boldsymbol{x}_N, \boldsymbol{\theta}_N)\} \subset \mathcal{X} \times \Theta$. 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 many times, but also provide a smooth prediction surface of $\hat{R}(\bar{\boldsymbol{x}}, \boldsymbol{\theta})$ over the space of $\boldsymbol{\theta}$ for a fixed design point $\bar{\boldsymbol{x}}$.

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

$$\widetilde{\mathcal{D}} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n\} \times \{\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_m\} = \{(\boldsymbol{x}_1, \boldsymbol{\theta}_1), \cdots, (\boldsymbol{x}_1, \boldsymbol{\theta}_m), \cdots, (\boldsymbol{x}_n, \boldsymbol{\theta}_1), \cdots, (\boldsymbol{x}_n, \boldsymbol{\theta}_m)\} := \{\boldsymbol{d}_1, \cdots, \boldsymbol{d}_N\} \tag{15}$$

with $\boldsymbol{d} = (\boldsymbol{x}, \boldsymbol{\theta}) = (x_1, \cdots, x_k, \theta_1, \cdots, \theta_q) = (d_1, \cdots, d_{k+q})$ and $N = n \times m$. Assuming that computer model output data $S_n = \{\boldsymbol{y} = [\boldsymbol{y}_1(\boldsymbol{x}_v, \boldsymbol{\xi}), \cdots, \boldsymbol{y}_M(\boldsymbol{x}_v, \boldsymbol{\xi})]'\}_{v=1}^n$ is collected 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

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

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

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

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_{iv} is the v_{th} component of $\boldsymbol{x}_i$ and θ_{il} is the l_{th} component of $\boldsymbol{\theta}_i$.

Since $\widetilde{\mathcal{D}}$ in (15) is of a Cartesian product design, an algorithm reducing the number of arithmetic operations to fit the RBF model (17) can be obtained. Specifically, for design (15), it is easy to 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 $N \times N$ identity matrix, $(\boldsymbol{B})_{ij} = \phi(\boldsymbol{d}_i - \boldsymbol{d}_j)$, $(\boldsymbol{B}_1)_{ij} = \phi(\boldsymbol{x}_i - \boldsymbol{x}_j)$, and $(\boldsymbol{B}_2)_{ij} = \phi(\boldsymbol{\theta}_i - \boldsymbol{\theta}_j)$.

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 semi-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 the matrices of orthogonal eigenvectors and $\boldsymbol{L}_1, \boldsymbol{L}_2$ are the diagonal matrices of the corresponding eigenvalues. Note that

$$\boldsymbol{B}_1 \otimes \boldsymbol{B}_2 + \lambda \boldsymbol{I}_N = (\boldsymbol{E}_1 \boldsymbol{L}_1 \boldsymbol{E}_1^T) \otimes (\boldsymbol{E}_2 \boldsymbol{L}_2 \boldsymbol{E}_2^T) + \lambda \boldsymbol{E}_1 \boldsymbol{E}_1^T \otimes \boldsymbol{E}_2 \boldsymbol{E}_2^T \tag{19}$$

$$= (\boldsymbol{E}_1 \otimes \boldsymbol{E}_2)(\boldsymbol{L}_1 \otimes \boldsymbol{L}_2)(\boldsymbol{E}_1^T \otimes \boldsymbol{E}_2^T) + \lambda (\boldsymbol{E}_1 \otimes \boldsymbol{E}_2)(\boldsymbol{E}_1^T \otimes \boldsymbol{E}_2^T) \tag{20}$$

$$= (\boldsymbol{E}_1 \otimes \boldsymbol{E}_2)(\boldsymbol{L}_1 \otimes \boldsymbol{L}_2 + \lambda \boldsymbol{I}_N)(\boldsymbol{E}_1^T \otimes \boldsymbol{E}_2^T) \tag{21}$$

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

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

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 [25] Page 70), by using the derived formula (22), we can compute $(\boldsymbol{B} + \lambda \boldsymbol{I}_N)^{-1}(\widetilde{\boldsymbol{R}} - \mu \boldsymbol{1})$ 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.

As a summary, we list the fitting procedures of RBF models in the following table

---

Fast fitting of RBF models

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

Procedure:

Step 1. Setting $\mu = \sum_{i=1}^{n} \widetilde{R}_i / N$.

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

Step 3. Calculate $\boldsymbol{\beta}$ from $\boldsymbol{\beta} = (\boldsymbol{B} + \lambda \boldsymbol{I})^{-1}(\widetilde{\boldsymbol{R}} - \mu \boldsymbol{1})$, where calculation of $(\boldsymbol{B} + \lambda \boldsymbol{I})^{-1}$ follows (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). And 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 $\boldsymbol{\theta} \in \Theta$ can be obtained through equation (8).

Denote the scalar statistics of interest as $\hat{\Upsilon}$ (for example, $\hat{R}(\boldsymbol{x}, \boldsymbol{\theta})$ in (17) or the components of $\boldsymbol{x}^*(\boldsymbol{\theta})$ in equation (8)). To measure the uncertainty associated with $\hat{\Upsilon}$, approximate confidence intervals can be constructed via the following bootstrap procedure:

Constructing confidence intervals via Bootstrap

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

Procedure:

Step 1. Estimate $\hat{\Upsilon}$ based on S_n.

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

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

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 $\delta^* = \hat{\Upsilon}^* - \hat{\Upsilon}$. Put these B^* values in order and pick out the 0.975 and 0.025 critical values. Set these values as $\delta^*_{.975}$ and $\delta^*_{.025}$.

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

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 [26, 14, 15, 3]. 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, [26, 3] reported their pioneering attempts at including multiple design objectives for optimizing the design of Drug-eluting stents, where a sufficiently high drug concentrations in smooth muscle cells and low drug concentration at the endothelial cell surface are to be achieved simultaneously.

To demonstrate the application of our approach, we use the design and modeling of drug release from a polymer matrix ([31] page 340) as a test example. The schematic of the drug release stochastic computer model is represented in Figure (1) as a polymer matrix (PM), where the PM radius is r_0 (cm), and PM length is z_L (cm). A certain amount of drug is initially administered in the PM, the modeling problem is then 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 mass transfer coefficient. The partial differential equation describing the diffusion in cylindrical coordinates (r, z, φ) is given by

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

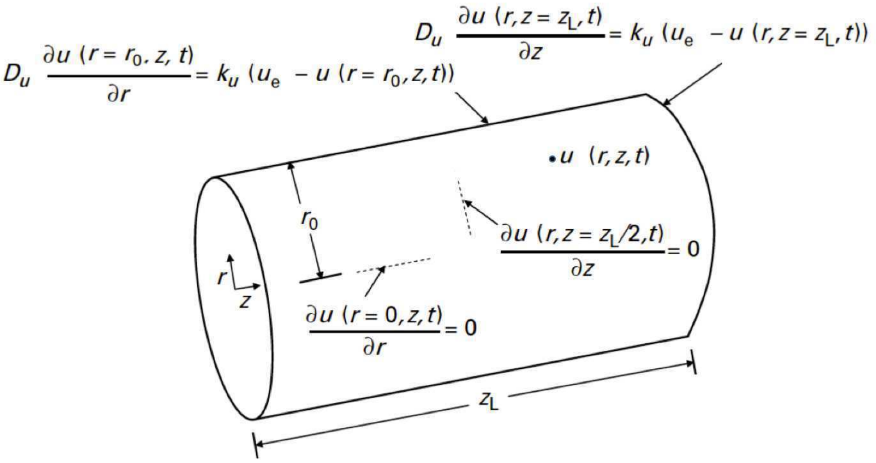
$D_u \frac{\partial u\ (r = r_0 .\ z,\ t)}{\partial r} = k_u\ (u_e\ - u\ (r = r_0, z, t))$
$D_u \frac{\partial u\ (r, z = z_L, t)}{\partial z} = k_u\ (u_e\ - u\ (r, z = z_L, t))$
$\bullet u\ (r, z, t)$
r_0
$\frac{\partial u\ (r, z = z_L/2, t)}{\partial z} = 0$
r
z
$\frac{\partial u\ (r = 0, z, t)}{\partial r} = 0$
z_L

Figure 1: Diagram of a drug diffusion system

Here, $u = u(r, z, \varphi, t)$ is the drug concentration at location (r, z, φ) after a release time of t, and D_u is the drug diffusivity which is a specified constant.

Assuming a uniform external drug concentration, the system in (24) is symmetric in φ, so the angular term in (24) is dropped. This gives $u = u(r, z, t)$ and

$$\frac{\partial u}{\partial t} = D_u(\frac{\partial^2 u}{\partial r^2} + \frac{1}{r}\frac{\partial u}{\partial r} + \frac{\partial^2 u}{\partial z^2}) \tag{25}$$

Equation (25) 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:

$$u(r, z, t = 0) = u_0 \tag{26}$$

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

The homogeneous Neumann BC ([31] page 113) for equation (25) at $r = 0$ is used to specify symmetry in r:

$$\frac{\partial u(r = 0, z, t)}{\partial r} = 0 \tag{27}$$

and the BC at the exterior surface $r = r_0$ is based on mass transfer coefficients k_u:

$$D_u \frac{\partial u(r = r_0, z, t)}{\partial r} = k_u(u_e - u(r = r_0, z, t)) \tag{28}$$

where u_e is the external concentration of drug subject to random fluctuation. This equation equate the mass fluxes at the polymer surface $r = r_0$ to the fluxes due to concentration differences at $r = r_0$. Similarly, the BC reflect symmetry in z and the BC at $z = z_L$ specify the fluxes in the bottom of the polymer:

$$D_u \frac{\partial u(r, z = z_L/2, t)}{\partial z} = 0, \qquad D_u \frac{\partial u(r, z = z_L, t)}{\partial z} = k_u(u_e - u(r, z = z_L, t)) \tag{29}$$

Equations (25) to (29) constitute equations of the computer model in the system of Figure (1). These partial differential equations(PDEs) are 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 PDEs are solved by discretizing r to 11 grid points over the value $0 \leqslant r \leqslant r_0$ and z to 11 grid points over the value $0 \leqslant z \leqslant z_L$. In other words, a system of 11×11 ODEs that approximates the PDE are used. More detailed description of the computer model can be found in [31] page 340.

To conclude the description of this stochastic computer model, we summarize the model design variable, output variable, random fluctuation variable, and the constants in Table (1). The value of these variables and parameters are specified following [31] page 340. Note that for concentrations, normalized values $0 \leqslant u(r, z, t), u_0, u_e \leqslant 1$ are used to 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.

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
$r_0 \in [0.5, 1.5]$	PM radius (cm)	control 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
$k_u = 1.0e - 01$	mass transfer coefficient (cm/s)	constant

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 [22]). First, the cost of drug is measured by the amount of drug initially administered to the system with $J_1(r_0) = \pi r_0^2 z_L u_0$. Second, 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 released from the PM at t is (see [31] page 340).

$$Q_t(r_0) = \pi r_0^2 z_L u_0 - 2 * 2\pi \int_{z_L/2}^{z_L} \int_0^{r_0} u(r,z,t) r dr dz \tag{30}$$

Translating these goals to desirability functions, a smaller-the-better function (2) is used for the cost measure with values $A = 0.1$ and $B = 15$

$$d_1^{min}(r_0) = \begin{cases} 0 & \text{if } J_1(r_0) > 15 \\ [\frac{J_1(r_0)-15}{0.1-15}]^{\theta} & \text{if } 0.1 \leqslant J_1(r_0) \leqslant 15 \\ 1 & J_1(r_0) < 0.1 \end{cases} \tag{31}$$

for the cost $J_1(r_0)$.

Next, target oriented desirability function (3) is used for $Q_{24}(r_0)$ with $A = 1$, $T_0 = 3.5$, $B = 12$, and $Q_{48}(r_0)$ with $A = 1.2$, $T_0 = 9.45$, $B = 13$ as

$$d_2^{target}(r_0) = \begin{cases} [\frac{Q_{24}(r_0)-1}{3.5-1}]^{\theta_{21}} & \text{if } 1 \leqslant Q_{24}(r_0) \leqslant 3.5 \\ [\frac{Q_{24}(r_0)-12}{3.5-12}]^{\theta_{22}} & \text{if } 3.5 \leqslant Q_{24}(r_0) \leqslant 12 \\ 0 & \text{otherwise} \end{cases} \tag{32}$$

$$d_3^{target}(r_0) = \begin{cases} [\frac{Q_{48}(r_0)-1.2}{9.45-1.2}]^{\theta_{31}} & \text{if } 1.2 \leqslant Q_{48}(r_0) \leqslant 9.45 \\ [\frac{Q_{48}(r_0)-13}{9.45-13}]^{\theta_{32}} & \text{if } 9.45 \leqslant Q_{48}(r_0) \leqslant 13 \\ 0 & \text{otherwise} \end{cases} \tag{33}$$

we can write the overall objective function as

$$J(r_0; \theta, \theta_{21}, \theta_{22}, \theta_{31}, \theta_{32}) = [d_1^{min}(r_0) * d_2^{target}(r_0) * d_3^{target}(r_0)]^{1/3} \tag{34}$$

The scale parameters $\theta, \theta_{21}, \theta_{22}, \theta_{31}, \theta_{32}$ are typically uncertain in the above objective function.

4.3. Optimization results and discussion

In this section, we describe the computer model and design optimization based on RBF metamodel in detail. The control variable $\boldsymbol{x}$ is the PM radius r_0(cm) in a design space $[0.5, 1.5]$, the model random fluctuation variable $\boldsymbol{\xi}$ is the normalized exterior drug concentration value u_e, which is assumed to be uniformly distributed in $[0, 0.2]$. The model output is the drug concentration $u(r, z, 24)$ and $u(r, z, 48)$ at time $t = 24, 48$h and at position (r, z). For the objective function described in (34), we fix $\theta_{21} = 2, \theta_{22} = 1, \theta_{31} = 2, \theta_{32} = 1$ and only treat θ as an uncertain parameter to be adjusted for illustrative purpose. Suppose a 6×6 points Cartesian product design is used over the space of model control variable r_0 and the uncertain parameter θ as

$$\widetilde{\mathcal{D}} = \{0.5, 0.7, 0.9, 1.1, 1.3, 1.5\} \times \{0, 0.6, 1.2, 1.8, 2.4, 3\} \tag{35}$$

At each design point of $r_0^v \in \{0.5, 0.7, 0.9, 1.1, 1.3, 1.5\}$, $M = 30$ computer model replications are conducted and the outputs $\boldsymbol{y}_j(r_0^v) = (u_j(r, z, 24), u_j(r, z, 48))|r_0^v, j = 1, \cdots, M$ are obtained for each replication. The sample average of the objective value from the replicates is then computed for each design point $(r_0^v, \theta^l) \in \widetilde{\mathcal{D}}$ as $\widetilde{R}(r_0^v, \theta^l) = \sum_{j=1}^{M} J(\boldsymbol{y}_j(r_0^v), r_0^v, \theta^l)/M$ Using the RBF regression in (17) to interpolate the surface of $\widetilde{R}(r_0, \theta)$, we obtain the prediction of $\widetilde{R}(r_0, \theta)$ at all values of $r_0 \in [0.5, 1.5]$ and $\theta \in [0, 3]$.

To check the model fitting efficiency, we compare the model predictions in a larger test design set with the "true value". Specifically, we estimate the "true value" accurately by using the average of $M = 100$ replicates. Figure (2) plots the predicted desirability values and the "true value" versus the grid point number for a 21×21 grid point Cartesian product design over the design space $[0.5, 1.5] \times [0, 3]$. As we can see, the curve of predicted and true value of the desirability is very close to each other, suggesting that the RBF regression model has very good prediction ability.

Then we plot the predicted desirability value surface in Figure (3) (left). As can be seen from this plot, using the proposed Cartesian product design and the RBF metamodel, we can predict the smooth surface of the desirability function score in its dependency on both the control variable r_0 and the uncertain scale parameter θ. Based on this prediction, we can conveniently obtain $r_0^*(\theta)$ for any choice of $\theta \in [0, 3]$ through 8, while avoiding the burden of refitting the RBF metamodel every time θ changes. In Figure (3) (right), we plot $r_0^*(\theta)$ for different choices of $\theta \in [0, 3]$, where the black curve is a smoothly changed optimal value curve under different settings of the scale parameter θ. In this plot, the optimal setting r_0^* changes significantly with different values of θ, which supports the argument that it is necessary to analyze the optimal solutions in their dependence on the uncertain objective parameters. In this drug delivery example, θ determines the impact of the cost of the drug on the overall desirability. When multiple optimal

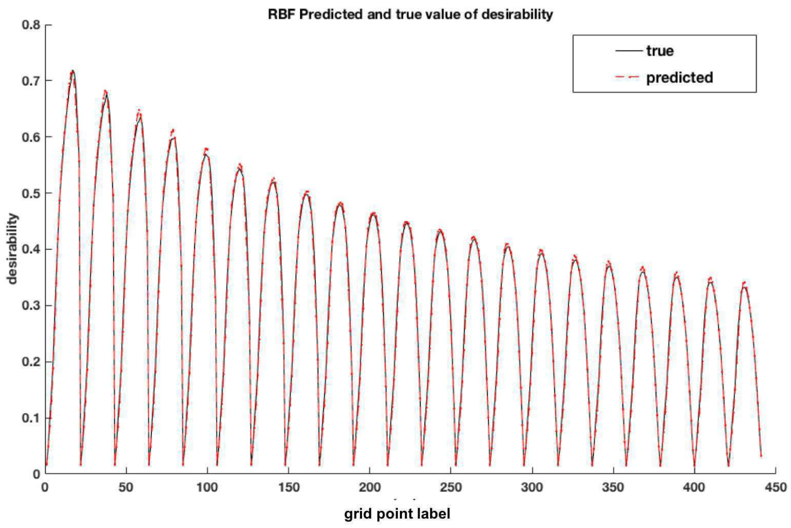

Figure 2: RBF predicted versus true desirability value

solutions are available for different values of θ, high-level information can be used to choose from one of the optimal solutions. For example, if the drug is expensive, the investigator may want to choose a large value for θ to heavily punish high amounts of initial drug $J_1(r_0)$. The calculation shows that all the optimal solution r_0^* fall into the range $[0.99, 1.27]$. If based on experience, r_0 values in the range $[0.9, 1.3]$ are generally acceptable, one can comfortably select a large value of θ without worrying about the potential unsatisfactory treatment effect.

In cases where the qualitative/higher-level information may not be approachable to the investigator, an attractive method is to select the scale parameters using a quantitative procedure. In Definition 1, we proposed such a solution to select the most "robust" optimal setting r_0^*. Now we illustrate the usefulness of this solution in this practical problem. In figure 4, we plot the robust measure $\int_{\boldsymbol{\vartheta}\in\Theta} \frac{D[\boldsymbol{x}^*(\boldsymbol{\theta});\boldsymbol{\vartheta}]}{D[\boldsymbol{x}^*(\boldsymbol{\vartheta});\boldsymbol{\vartheta}]} d\boldsymbol{\vartheta}$ for each optimal setting $r_0^*(\theta)$. The robustness of $r_0^*(\theta)$ reaches its peak at $r_0^*(\theta^*) = 1.1200$ with $\theta^* = 1.52$. Therefore, choosing $\theta^* = 1.52$ as the scale parameter value would produce a robust optimal solution that performs well under other choices of θ. This robust optimal solution can also serve as a compromised solution when there are many different decision makers/patients, and different scale parameter should be used to represent their specific trade-off preference.

5. Discussion

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 optimiza-

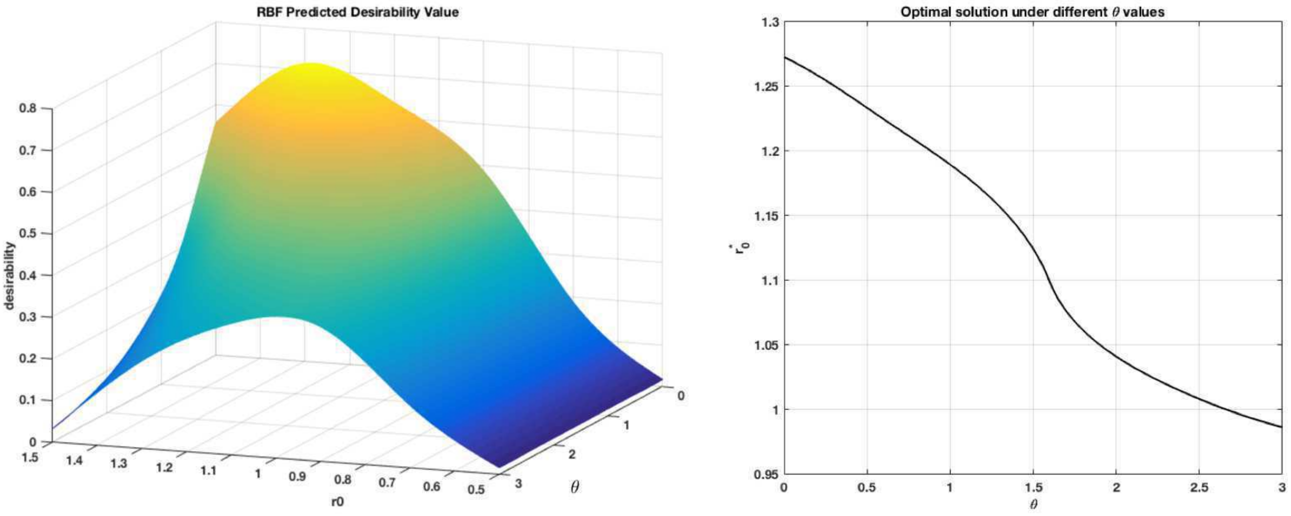

Figure 3: Desirability surface and optimal solutions

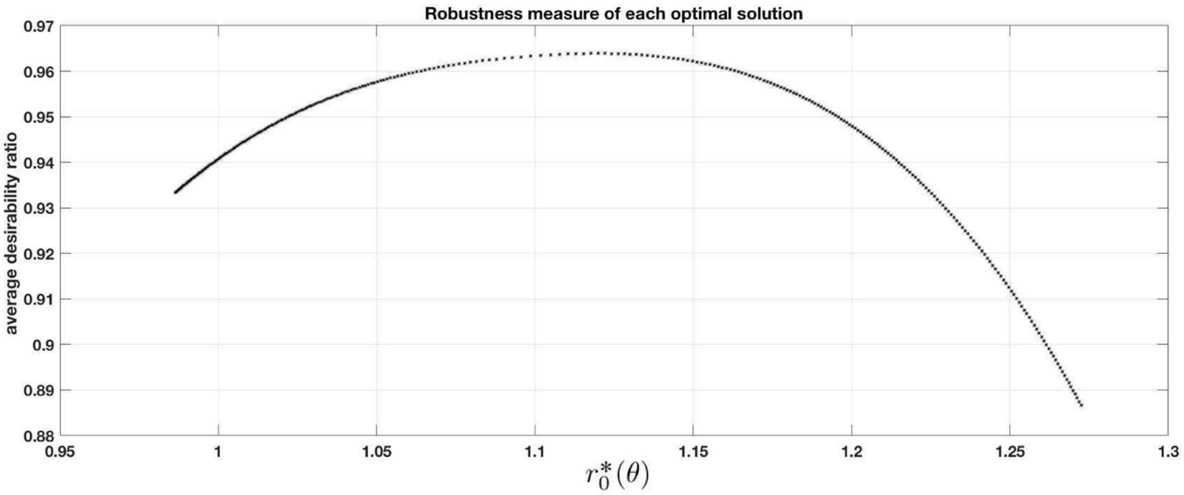

Figure 4: Robust measure plot for different r_0^*

tion method through predicting the objective function as a function of both the design variables and the objective function uncertain parameters. 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.

Although the developed RBF focused on the case where one overall objective function with uncertain parameters is of interest, it can be easily extended for fitting 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.

References

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

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

[3] 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.

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

[5] 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.

[6] Cao, Y., Li, H., and Petzold, L. (2004). Efficient formulation of the stochastic simulation algorithm for chemically reacting systems. *The journal of chemical physics*, 121(9):4059–4067.

[7] 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.

[8] 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.

[9] Deng, D., Luo, Y., Serizawa, H., Shibahara, M., and Murakawa, H. (2003). Numerical simulation of residual stress and deformation considering phase transformation effects (mechanics, strength & structural design). *Transactions of JWRI*, 32(2):325–333.

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

[11] Ding, J., Wang, X., Zhou, X.-F., Ren, N.-Q., and Guo, W.-Q. (2010). Cfd optimization of continuous stirred-tank (cstr) reactor for biohydrogen production. *Bioresource technology*, 101(18):7005–7013.

[12] 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.

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

[14] 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.

[15] Groh, C. M., Hubbard, M. E., Jones, P. F., Loadman, P. M., Periasamy, N., Sleeman, B. D., Smye, S. W., Twelves, C. J., and Phillips, R. M. (2014). Mathematical and computational models of drug transport in tumours. *Journal of The Royal Society Interface*, 11(94):20131173.

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

[17] Humphrey, D. G. and Wilson, J. R. (2000). A revised simplex search procedure for stochastic simulation response surface optimization. *INFORMS Journal on Computing*, 12(4):272–283.

[18] 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.

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

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

[21] 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.

[22] 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.

[23] 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.

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

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

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

[27] 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.

[28] 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.

[29] 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.

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

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

[32] Stevens, B. and Branam, R. (2015). 3-dimensional aerospike nozzle design. In *APS Division of Fluid Dynamics Meeting Abstracts.*

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

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

[35] 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.

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

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

[38] 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.

[39] 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.