Proceedings of the 2024 Winter Simulation Conference
H. Lam, E. Azar, D. Batur, S. Gao, W. Xie, S. R. Hunter, and M. D. Rossetti, eds.

INTRODUCTORY TUTORIAL: SIMULATION OPTIMIZATION UNDER INPUT UNCERTAINTY

Linyun He[1] and Eunhye Song[1]

[1]School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, USA

ABSTRACT

Input uncertainty in the simulation output is caused by the estimation error in the input models of the simulator due to finiteness of the data from which they are estimated. Ignoring input uncertainty when formulating and solving a simulation optimization problem may lead to a solution with poor system performance. This tutorial discusses how to incorporate input uncertainty in simulation optimization to avoid such risk. We first categorize the problems into three groups based on their contexts: fixed batch data, streaming data, and active input data collection problems. Input and simulation output response modeling frameworks that can be adopted in all three categories are discussed. Then, we provide a high-level overview of simulation optimization problem formulations and algorithmic approaches to tackle problems in each group. Some thoughts on future research directions are shared.

1 INTRODUCTION

Simulation is a popular tool to support policy, design, or operational decision making thanks to its modeling flexibility for representing a real-world system's stochastic behavior in several application areas including healthcare, manufacturing, finance, marketing and more. In stochastic simulation, what drives randomness in its output is the collection of input random variables generated from distribution functions that mimic the randomness in the real-world system. Often, the true, real-world distributions of such inputs are unknown and must be estimated from a set of data or postulated based on some prior knowledge in the absence of data. If the target system to simulate already exists, then it is sensible to estimate the input distributions from data collected from the target system. The resulting estimators are referred to as input models.

Since the data is always finite, the input models are subject to estimation error. When simulation inputs are generated from such erroneous input models, their estimation error is propagated to the simulation output causing what is known as *input uncertainty*. Input uncertainty must be differentiated from the inherent stochastic uncertainty in the simulation output. The latter is the randomness we like to model to mimic the real-world stochasticity of the system, whereas the former is an error we wish to reduce.

When a performance measure of the system is estimated from simulation outputs, both stochastic and input uncertainties introduce error in the estimate. The estimation error caused by the stochastic uncertainty can be reduced by improving the simulation experiment design, e.g., increasing the number of replications or length of the simulation run. However, input uncertainty in the performance measure estimate persists even if infinite simulation effort is spent unless the input models are improved by collecting additional data, which may or may not be feasible depending on the problem context; see Section 3.

In simulation optimization, the objective function/constraints are defined as statistics (e.g., expectations) of simulation outputs. In general, these do not have analytical expressions, and must be estimated by running simulations at a feasible solution. A classical simulation optimization algorithm controls the level of error in these estimates caused by stochastic uncertainty so that the solution returned by the algorithm is indeed optimal at a desired level of statistical guarantee. However, when the estimated objective function values are subject to input uncertainty, there is risk of selecting a suboptimal solution if a classical algorithm is applied without accounting for the effect of input uncertainty, known as the *input model risk*. This tutorial

discusses *how input model risk imposes challenges in the simulation optimization problems, and introduces mathematical models and algorithmic methodologies to address such challenges.*

Input model risk issues can be found in practical simulation optimization studies. One example is the Vehicle Content Optimization program at General Motors (GM), a simulation-based decision support system that helps optimize a vehicle content portfolio to improve GM's market performance (Song et al. 2020). One of the key inputs in their simulator is a consumer's utility parameter vector whose elements represent the consumer's preferences for attribute levels of various features of a vehicle relative to its price. The distribution of utility parameters is estimated from GM's conjoint analysis data obtained by surveying a stratified sample of consumers. Song et al. (2020) report that the estimated utility parameters are the most significant source of uncertainty in the predicted market performance (e.g., market share), which can potentially negatively impact their content optimization decisions.

There are several tutorials introducing the input uncertainty quantification problem (Song et al. 2014; Lam 2016; Corlu et al. 2020; Barton et al. 2022). These are excellent resources to learn the basic premise of the input uncertainty problem in simulation as well as different simulation experiment design/estimation methods to quantify input uncertainty. Their common goal is to make a correct inference on the true performance measure of a single system design or policy even if the true input distributions are unknown. Estimating the true performance measure with high accuracy can be challenging, if there is large input uncertainty and additional data collection is difficult. Therefore, the focus tends to be on determining the minimum simulation effort such that the inference can be made as precisely as possible.

On the other hand, simulation optimization focuses on finding the best solution among the feasible candidates, which naturally involves *comparisons* among solutions. This typically alleviates the precision challenge in the quantification problem. Intuitively speaking, as long as the ordering among the solutions are correct, the optimal solution can be selected correctly even if the performance measure estimate has an error. We further expand on this point in Section 4.3. On the contrary, the goal of a simulation optimization problem is not as straightforward to state as that of the quantification problem because the problem formulation closely depends on the context—feasibility of additional data collection and frequency at which the decision is required at the system—and the decision-maker's attitude towards risk among others. This tutorial discusses what problem contexts one should consider when tackling a simulation optimization problem under input uncertainty. We then introduce modeling choices to represent input uncertainty's effect on the simulation output, simulation optimization problem formulation, and solution method (algorithm), all of which depend on the problem contexts.

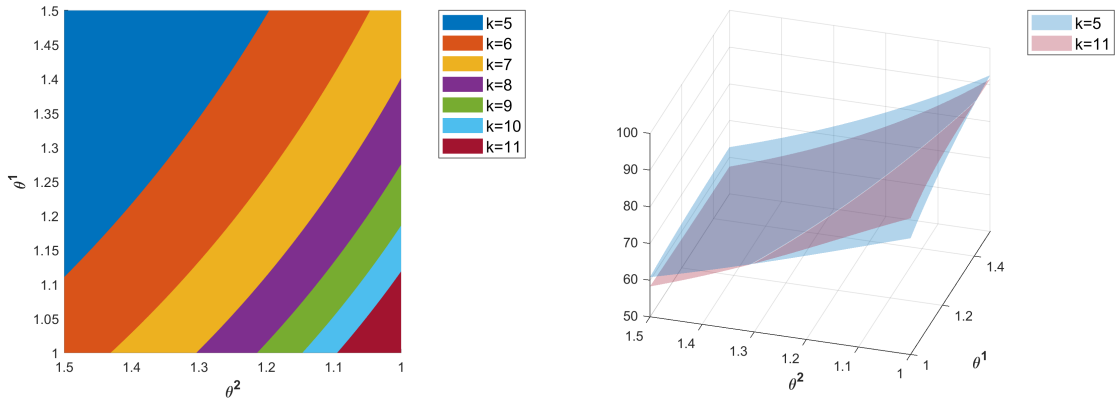
The rest of the tutorial is organized as follows. Section 2 provides a concrete example that highlights input model risk in simulation optimization. In Section 3, we discuss the problem contexts one should consider before formulating the simulation optimization problem. We introduce different modeling techniques to represent the estimation error in the input models as well as how such error affects the simulation output in Section 4. In Section 5, we introduce different problem formulations suitable for each problem context discussed in Section 3, and corresponding solution methods. Some thoughts on future research are shared in Section 6.

2 INPUT MODEL RISK IN SIMULATION OPTIMIZATION

Consider the following representation of a simulation output random variable:

$$Y(\mathbf{x};F) = \mathbb{E}[Y(\mathbf{x};F)|F] + \varepsilon(\mathbf{x};F) \triangleq \eta(\mathbf{x};F) + \varepsilon(\mathbf{x};F), \tag{1}$$

where F is the input model adopted to run the simulator. We allow F to be a random quantity in this tutorial as F is estimated from input data. Additionally, $\mathbf{x}$ represents the decision variable (solution) of the simulation optimization problem. Depending on the purpose, $\mathbf{x}$ may be an operational policy or a design for the target system. The run-to-run variation (i.e., simulation error) of $Y(\mathbf{x};F)$ is represented by $\varepsilon(\mathbf{x};F)$, which has mean 0 conditional on F and we further assume its conditional variance, $\mathbb{V}(Y(\mathbf{x};F)|F)$,



(a) Top view; each area of (θ_1, θ_2) is colored to show the conditional-mean-profit-maximizing k.

(b) Conditional mean profits given (θ_1, θ_2) for $k = 5$ and 11.

Figure 1: Conditional means of an hourly profit for a steady-state M/M/1/k queue for different values of k within the ranges of arrival rate $\theta_1 \in [1, 1.5]$ and mean service time $\theta_2 \in [1, 1.5]$.

is nonzero and finite for all F. Moreover, $\eta(\mathbf{x}; F)$ is the notation we adopt for the conditional mean of the simulation output given F, i.e., $\eta(\mathbf{x}; F) = \mathbb{E}[Y(\mathbf{x}; F)|F]$.

Throughout the tutorial, we denote the true input distribution function that generates the data in the target system by F^c, where c stands for "correct." We assume that we can collect independent and identically distributed (i.i.d.) observations from F^c. Furthermore, F^c can be a collection of several independent input distributions if there are more than one data sources in the system. Additional notation is specified in Section 5.3 when we discuss active input data collection from multiple data sources.

The following is a generic simulation optimization problem parameterized by input model F:

$$\mathbf{x}^*(F) = \arg\max_{\mathbf{x} \in \mathscr{X}} \eta(\mathbf{x}; F), \qquad \text{Opt}(F)$$

where $\mathscr{X}$ represents the feasible solution set. For simplicity, let us assume there exists a unique optimal solution, $\mathbf{x}^*(F^c)$. Note that $\mathbf{x}^*(F^c)$ as well as $\eta(\mathbf{x}; F^c)$ for each $\mathbf{x} \in \mathscr{X}$ are deterministic. Indeed, several algorithms have been proposed to solve Opt(F^c) assuming F^c is known, however, they provide little insights for finding $\mathbf{x}^*(F^c)$ when F^c is unknown. The most common *ad hoc* approach is to compute estimator $\hat{F}$ of F^c, then solve Opt($\hat{F}$) hoping that $\mathbf{x}^*(\hat{F}) = \mathbf{x}^*(F^c)$. In general, this statement is not true; $\mathbf{x}^*(\hat{F})$ is random because $\hat{F}$ depends on the randomly observed data, whereas $\mathbf{x}^*(F^c)$ is deterministic.

Figure 1 illustrates the challenge more concretely with a system design example, where the objective is to find the system capacity, k, that maximizes the expected profit of an M/M/1/k queueing system. Here, the revenue is generated by charging each served customer a price of service, whereas there is cost per waiting time of each entering customer. If the system is full (i.e., k customers in the system), then no more arrivals can be accepted. In this example, F consists of the inter-arrival and service time distributions, which are known to follow exponential distributions. Thus, F can be characterized by the parameter vector $\boldsymbol{\theta}$ consisting of the arrival rate, θ_1, and the mean service time, θ_2. Therefore, inferring F^c boils down to estimating the true parameter vector, $\boldsymbol{\theta}^c = (\theta_1^c, \theta_2^c)$.

Suppose the decision variable $\mathbf{x}$ for this problem is the capacity of the system, k, where the feasible solution set is $k \in \{5, 6, 7, 8, 9, 10, 11\}$. The objective function value is then the mean profit, $\eta(k; \boldsymbol{\theta}) \triangleq \mathbb{E}[Y(k; F(\boldsymbol{\theta})|\boldsymbol{\theta}]$, which depends on both k and $\boldsymbol{\theta}$. Larger k increases the revenue by letting more customers into the system, but it also incurs higher cost as the customers admitted to the queue would wait longer on average. The trade-off between the revenue and cost depends on the arrival and the service rates, which potentially makes different ks be optimal for different $\boldsymbol{\theta}$ values. Such dependence is captured in Figure 1b, which

plots $\eta(5;\boldsymbol{\theta})$ and $\eta(11;\boldsymbol{\theta})$ in the range of $\boldsymbol{\theta} \in [1,1.5]^2$. Notice that for lower values of θ_1 and θ_2, $k=11$ outperforms $k=5$ as the service is fast enough so that the waiting cost does not dominate. In Figure 1a, we plot $\eta(k;\boldsymbol{\theta})$ for all feasible values of k projected onto the domain of $\boldsymbol{\theta}$. Each colored area indicates the set of $\boldsymbol{\theta}$ values that makes the corresponding k be optimal. Suppose the unknown true parameter values are $(\theta_1^c,\theta_2^c)=(1.1,1.4)$, which implies that the optimal capacity level for the system is $k=6$. If $\hat{\boldsymbol{\theta}}$ estimated from data belongs in the area of $k=6$, then solving Opt$(F(\hat{\boldsymbol{\theta}}))$ correctly leads us to the true optimum, $k=6$. On the other hand, when $\hat{\boldsymbol{\theta}}$ does not belong in this area, then $\mathbf{x}^*(F(\hat{\boldsymbol{\theta}})) \neq 6$.

3 PROBLEM CONTEXTS

As emphasized earlier, how to tackle Opt(F^c) when F^c is unknown depends on the problem contexts. The key features to consider are 1) whether it is possible to collect additional input data from the system; and 2) the timeline at which the decision is required for the target system. In this section, we categorize simulation optimization problems under input uncertainty in three groups depending on their problem contexts.

The first group is what we referred to as *fixed batch data* problems. In these cases, a decision-maker is given a set of input data, however, additional data collection is infeasible or too costly to consider. As such, the focus is on making statistical inference on the identity of $\mathbf{x}^*(F^c)$ among the feasible solutions in $\mathscr{X}$ given the batch data. Here, we use the term "inference" because it is often difficult to find $\mathbf{x}^*(F^c)$ with high statistical confidence with a fixed batch input data (Song et al. 2015). Instead, the problem can be relaxed to find a set of solutions that contain $\mathbf{x}^*(F^c)$, for instance. Or one may formulate a problem alternative to Opt(F^c) to guard against the risk of not knowing F^c. In both cases, the corresponding solution methods control stochastic error in simulation, not input uncertainty. Since additional input data is not collected, the problem can be solved whenever the decision is required.

The second group falls under the category of *streaming input data* problems, where the system generates additional data and the decision-maker can passively collect them. Here, by passively we mean that the input data generating process cannot be actively controlled. The data may be streamed in continuously or in batches at discrete time points. In either case, for the convenience of analysis, one may discretize the time into periods and assume that a batch of data is collected at the beginning of each period.

There are two possible decision time lines in this context. First, the system may require a single solution (e.g., system design) to be implemented and the decision-maker needs to determine when to stop collecting the streaming data and return a solution. In this case, it is sensible to have a fixed precision requirement that measures how closely the returned solution approximates $\mathbf{x}^*(F^c)$. Second, the system may require a solution at every period (e.g., operational policy). Then, a feasible solution needs to be returned at the end of each period. In both cases, the stochastic error can be controlled, not input uncertainty.

In the second case, it is important to mention that the decision can possibly alter the input data generating process. For instance, if a seller adjust the price of their product at the end of a period, then the sales in the next period may decrease. However, in this tutorial we focus on the case that the decision is decoupled from the input-generating process. For instance, the standby location of an ambulance in between its operation does not affect the distribution of time and location of the emergency incidents.

The third group, *active input data collection* problems, include the scenarios where the decision-maker can actively choose the input data sources to collect more data from and how much at a cost. Here, the cost may be associated with the time or price of obtaining a new data point (e.g., collecting a survey response, time to process raw data). Therefore, unlike the first two groups, the solution methods for the third group control both stochastic error and input uncertainty. Since the simulation also costs time and computing resources, one can consider a trade-off between the input data collection and simulation replication so that the resulting solution can be found efficiently.

In Section 5, we discuss simulation optimization problem formulations for each of the three categories defined here and provide pointers to corresponding algorithms to solve them.

4 MODELING

In this section, we present a brief overview on modeling techniques to characterize input uncertainty in simulation optimization. In Section 4.1, we introduce different input models and how their estimation errors can be quantified. Section 4.2 discusses how $\eta(\mathbf{x}, F)$ can be modeled as a function of both $\mathbf{x}$ and F (Section 4.2) to improve efficiency in inferring the objective function values at solutions. In Section 4.3, we further discuss improving the statistical efficiency of comparison between two solutions using a model introduced in Section 4.2.

4.1 Input Modeling

To set up, let $\mathscr{Z} = \{Z_1, \ldots, Z_m\}$ denote the input data set of size m, where $Z_i \overset{i.i.d.}{\sim} F^c$ represents each observation within the sample. For simplicity, we assume Z_i is a scalar random variable unless otherwise mentioned, however, all subsequent discussions can be extended to include vector-valued Z_i. To infer F^c from $\mathscr{Z}$, there are two competing philosophies of statistical modeling methods: frequentist and Bayesian.

4.1.1 Frequentist Approach

Frequentists begin with the assumption that there exists a fixed (but unknown) F^c. We call it parametric input modeling when F^c is assumed to have to a known parametric distribution function such as exponential, Gamma, normal, etc. Then, estimating F^c boils down to estimating its parameter vector $\boldsymbol{\theta}^c$. That is, $F^c(\cdot) = F(\cdot|\boldsymbol{\theta}^c)$. Let $\Theta \subset \mathbb{R}^d$ denote the feasible parameter space for the distribution family of F^c. Then, the estimator of $\boldsymbol{\theta}^c$, $\hat{\boldsymbol{\theta}} \in \Theta$, can be computed from $\mathscr{Z}$ using various methods, such as M- or Z-estimators, method-of-moment estimators, and more.

Because $\hat{\boldsymbol{\theta}}$ is computed from a finite batch of data, it is random. The sampling distribution of $\hat{\boldsymbol{\theta}}$ characterizes how $\hat{\boldsymbol{\theta}}$ would be distributed if we are allowed to sample another size-m data set under the assumption that the distribution function imposed for F^c is correct. Unfortunately, it is difficult to derive such a sampling distribution in general when m is finite. This motivates us to turn to the asymptotic sampling distribution of $\hat{\boldsymbol{\theta}}$, which stipulates how $\hat{\boldsymbol{\theta}}$ behaves in distribution when m increases to infinity.

Under certain regularity conditions, these estimators are consistent for $\boldsymbol{\theta}^c$ and have statistical convergence rates associated with the sample size m. For example, when $\hat{\boldsymbol{\theta}}$ is the maximum likelihood estimator (MLE) of $\boldsymbol{\theta}^c$, under some conditions, $\hat{\boldsymbol{\theta}}$ satisfies the following asymptotic normality (Van der Vaart 1998, Section 5.5):

$$\sqrt{m}(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^c) \xrightarrow{D} \mathscr{N}(\mathbf{0}, \Sigma(\boldsymbol{\theta}^c)), \tag{2}$$

where, $\xrightarrow{D}$ denotes convergence in distribution, $\mathbf{0}$ is a zero vector, $\Sigma(\cdot) \triangleq I^{-1}(\cdot)$, and $I(\boldsymbol{\theta}^c)$ is the Fisher information matrix of $F(\cdot|\boldsymbol{\theta})$ at θ^c. Although, $I(\boldsymbol{\theta}^c)$ is unknown, one can plug in $I(\hat{\boldsymbol{\theta}})$ to approximate (2).

In reality, the distribution family of F^c is likely unknown. A popular heuristic to tackle this issue is to fit several different distribution functions and run hypothesis tests to find a statistically valid model. However, the resulting model is not consistent to F^c unless F^c truly belongs in the chosen distribution family. Even if m increases to infinity, the approximation error of $F(\cdot|\hat{\boldsymbol{\theta}})$ cannot be made arbitrarily small.

Instead, one can opt for nonparametric input modeling by approximating F^c with an empirical cumulative distribution function (ecdf) constructed from $\mathscr{Z}$, $\hat{F}(z) = \frac{1}{m}\sum_{i=1}^{m} \mathbf{1}\{Z_i \leq z\}$, where $\mathbf{1}\{\cdot\}$ is the indicator function. The Glivenko-Cantelli theorem shows that $\hat{F}$ is uniformly strongly consistent to F^c (Van der Vaart 1998, Section 19), i.e., $\hat{F}$ becomes arbitrarily close to F^c over the entire support as m increases, a strong basis for one to adopt $\hat{F}$ as an estimator for F^c as it does not require the distribution function of F^c to be known.

Much like $\hat{\boldsymbol{\theta}}$, one can define the sampling distribution of $\hat{F}$ from that $\mathscr{Z}$ is a size-m i.i.d. sample from F^c. This again is difficult to characterize for finite m and thus, an asymptotic approximation is often adopted. A popular approximation method is known as *bootstrapping*. The main idea of bootstrap is to regard $\hat{F}$ as F^c and construct a bootstrap sample $\mathscr{Z}^\star = \{Z_1^\star, \ldots, Z_m^\star\}$, where $Z_i^\star \sim \hat{F}$. The ecdf constructed

from $\mathscr{Z}^\star$, $\hat{F}^\star$, is a bootstrap estimator of $\hat{F}$. Bootstrap theory relies on that as m increases $\hat{F} \overset{D}{\approx} F^c$, and thus $\hat{F}^\star \overset{D}{\approx} \hat{F}$, where $\overset{D}{\approx}$ means that the two distributions are approximately equal (loosely speaking). Since $\hat{F}$ is known, one can generate multiple sets of $\mathscr{Z}^\star$ and corresponding $\hat{F}^\star$, and approximate the sampling distribution of $\hat{F}$ with that of $\hat{F}^\star$.

4.1.2 Bayesian Approach

Alternative to the frequentist standpoint, the (parametric) Bayesian regards the unknown parameter as random vector $\boldsymbol{\theta}$ rather than fixed $\boldsymbol{\theta}^c$. Before considering the data, they impose prior distribution $\pi(\boldsymbol{\theta})$ on $\boldsymbol{\theta}$ to represent its uncertainty. The prior is then updated to the posterior distribution function, $p(\boldsymbol{\theta}|\mathscr{Z})$, reflecting $\mathscr{Z}$ via Bayes' rule:

$$p(\boldsymbol{\theta}|\mathscr{Z}) = \frac{L(\mathscr{Z}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})}{\int_{\boldsymbol{\theta}\in\Theta} L(\mathscr{Z}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta}}, \tag{3}$$

where the likelihood function of $\boldsymbol{\theta}$ is defined as $L(\mathscr{Z}|\boldsymbol{\theta}) = \prod_{i=1}^{m} f(Z_i|\boldsymbol{\theta})$ and $f(Z_i|\boldsymbol{\theta})$ the probability distribution function of $F(\cdot|\boldsymbol{\theta})$ evaluated at each Z_i. In words, $p(\boldsymbol{\theta}|\mathscr{Z})$ is the posterior belief about $\boldsymbol{\theta}$ after reflecting the data, $\mathscr{Z}$. Contrary to frequentist methods where the sampling distributions of the input models need to be approximated in general, the Bayesian posterior gives the exact probability distribution of the parameter $\boldsymbol{\theta}$ given the evidence of $\mathscr{Z}$ and prior belief π.

There are nonparametric Bayesian methods that remove the assumptions that π and f belong to known distribution families, and allow them to be infinite-dimensional models (e.g. an infinite-capacity mixture model; see Gershman and Blei (2012) for instance). However, we have found these models yet to be utilized in the input uncertainty literature.

In practice, there are a few factors beyond statistical philosophies that determine the input modeling approach. The parametric frequentist methods offer the advantage of simplicity in implementation and computational ease, especially when the input data size is big. The parametric Bayesian methods may reflect the prior knowledge about $\boldsymbol{\theta}$ as it can be reflected in the prior, π. However, unless there is conjugacy between π and L, the posterior distribution in (3) may not be analytically computed. In these cases, one can still obtain an approximately i.i.d. sample from the posterior via the Markov chain Monte Carlo (MCMC) method, which tends to be computationally demanding (Asmussen and Glynn 2007). Nonparametric approaches are free of the bias introduced by the parametric family assumption, however, analyzing input uncertainty's effect on the simulation output means can be more mathematically and computationally challenging. Lastly, Bayesian approaches are well-suited for the active input data collection problems as the posterior update is not affected by whether the additional input data collection decision is based on the current data set (Kim and Song 2022), whereas, in the frequentist case, the analysis may become more complex once the i.i.d. assumption about the input data is violated.

4.2 Simulation Response Modeling

In the design and analysis of computer experiment literature, the simulation conditional mean function, $\eta(\mathbf{x};F)$, is often referred to as the "response" function (Santner et al. 2003). We adopt this terminology.

In simulation optimization, response modeling is exploited to make inference on the solution's response more efficiently by imposing some assumption—statistical or functional—about the relationship between $\eta(\mathbf{x};F)$ and $\mathbf{x}$ so that $\eta(\mathbf{x};F)$ can be learned by the simulation outputs collected from all solutions, not only $\mathbf{x}$. In the papers incorporating input uncertainty, it is useful to model the response as a function of both $\mathbf{x}$ and F since the dependence on the solution's performance on the input model, F, is an important factor to consider. Below, we introduce some popular models in the literature.

For all models discussed in this section and thereafter, we assume that all solutions share the same input model; see Song et al. (2015) for a discussion on a more general case. In the M/M/1/k example discussed in Section 2, all system capacity levels share the same inter-arrival and service time distributions.

Suppose the decision-maker adopts a parametric model so that $F^c(\cdot) = F(\cdot|\boldsymbol{\theta}^c)$. Then, the following Taylor series approximation can be adopted to represent $\eta(\mathbf{x};\boldsymbol{\theta})$ for each $\mathbf{x}$ under mild conditions:

$$\eta(\mathbf{x};\boldsymbol{\theta}) \approx \eta(\mathbf{x};\boldsymbol{\theta}^c) + \nabla_{\boldsymbol{\theta}}\eta(\mathbf{x};\boldsymbol{\theta}^c)^\top(\boldsymbol{\theta}-\boldsymbol{\theta}^c) \text{ for } \boldsymbol{\theta} \in \Theta. \tag{4}$$

Note that $\nabla_{\boldsymbol{\theta}}$ denotes the partial derivative operator with respect to $\boldsymbol{\theta}$. Namely, (4) decomposes $\eta(\mathbf{x};\boldsymbol{\theta})$ into the true mean of solution, $\eta(\mathbf{x};\boldsymbol{\theta}^c)$, and the linear effect of $\boldsymbol{\theta} \neq \boldsymbol{\theta}^c$ on η. When $\boldsymbol{\theta}^c$ is estimated by its MLE $\hat{\boldsymbol{\theta}}$, Model (4) can be combined with (2) to characterize the sampling distribution of $\eta(\mathbf{x};\hat{\boldsymbol{\theta}})$:

$$\sqrt{m}(\eta(\mathbf{x};\hat{\boldsymbol{\theta}}) - \eta(\mathbf{x};\boldsymbol{\theta}^c)) \xrightarrow{D} \mathcal{N}(\mathbf{0}, \nabla_{\boldsymbol{\theta}}\eta(\mathbf{x};\boldsymbol{\theta}^c)^\top \Sigma(\boldsymbol{\theta}^c)\nabla_{\boldsymbol{\theta}}\eta(\mathbf{x};\boldsymbol{\theta}^c)) \tag{5}$$

as m increases to infinity. Indeed, (5) is extremely handy as it completely characterizes the effect of input uncertainty on the simulation output provided that (4) is assumed to hold. The gradient, $\nabla_{\boldsymbol{\theta}}\eta(\mathbf{x};\boldsymbol{\theta}^c)$, is unknown, but can be estimated via regression (Song and Nelson 2019).

Clearly, Model (4) has its limitations. First, it leaves out higher-order effects of $\boldsymbol{\theta}$. However, the fidelity of the model improves when m increases as $\hat{\boldsymbol{\theta}}$ tends to $\boldsymbol{\theta}^c$ and ultimately (4) fits well in the neighborhood of $\boldsymbol{\theta}^c$ given that $\eta(\mathbf{x};\boldsymbol{\theta}^c)$ is smooth at $\boldsymbol{\theta}^c$. Second, (4) is defined for each $\mathbf{x}$, which is perfectly reasonable when each $\mathbf{x}$ is categorical (e.g., different queueing priority rules). However, when $\mathscr{X}$ can be embedded in a metric space such as the Euclidean space, it is sensible to consider a model that takes both $(\mathbf{x},\boldsymbol{\theta}) \in \mathscr{X} \times \Theta$ as inputs to achieve a better prediction of the response.

One way to achieve such a model is to assume $\eta(\mathbf{x},\boldsymbol{\theta})$ is a realization of a stochastic process that maps $(\mathbf{x},\boldsymbol{\theta}) \in \mathscr{X} \times \Theta$ to $\eta(\mathbf{x};\boldsymbol{\theta}) \in \mathbb{R}$. Additionally, we assume $\mathscr{X} \in \mathbb{R}^p$ for simplicity in the following discussion. A popular choice of the model is Gaussian process (GP) regression that imposes the following prior to η before simulating any $(\mathbf{x},\boldsymbol{\theta})$:

$$\eta(\mathbf{x};\boldsymbol{\theta}) \sim GP(\mu(\mathbf{x};\boldsymbol{\theta}), K(\mathbf{x},\boldsymbol{\theta};\mathbf{x}',\boldsymbol{\theta}')), \tag{6}$$

where μ is the mean function and $K(\mathbf{x},\boldsymbol{\theta};\mathbf{x}',\boldsymbol{\theta}')$ is the covariance kernel, which determines the covariance between $\eta(\mathbf{x};\boldsymbol{\theta})$ and $\eta(\mathbf{x}';\boldsymbol{\theta}')$ for any $(\mathbf{x},\boldsymbol{\theta}),(\mathbf{x}',\boldsymbol{\theta}') \in \mathscr{X} \times \Theta$. Although more general models are possible, the following form of product kernel is a popular choice for incorporating input uncertainty in the simulation optimization literature (Ungredda et al. 2022): $K(\mathbf{x},\boldsymbol{\theta};\mathbf{x}',\boldsymbol{\theta}') = K_{\mathscr{X}}(\mathbf{x},\mathbf{x}')K_{\Theta}(\boldsymbol{\theta},\boldsymbol{\theta}')$, where $K_{\mathscr{X}}(\mathbf{x},\mathbf{x}')$ and $K_{\Theta}(\boldsymbol{\theta},\boldsymbol{\theta}')$ are the covariance kernels defined on $\mathscr{X}$ and Θ, respectively. Moreover, a kernel is said to be stationary if it only depends on the difference between the two inputs. For instance, the following Gaussian kernel is a popular stationary kernel:

$$K_{\mathscr{X}}(\mathbf{x},\mathbf{x}') = \tau^2_{\mathscr{X}} \exp\left(-||\mathbf{x}-\mathbf{x}'||^2/2\lambda^2\right), \tag{7}$$

where $\tau_{\mathscr{X}}$ and λ are hyperparameters of the kernel. Observe that (7) always returns a positive covariance. Loosely speaking, (7) models covariance between $\eta(\mathbf{x};\boldsymbol{\theta})$ and $\eta(\mathbf{x}';\boldsymbol{\theta})$ assuming that the function values are more similar to each other (higher covariance) when $\mathbf{x}$ and $\mathbf{x}'$ are closer. A similar choice can be made for $K_{\Theta}(\boldsymbol{\theta},\boldsymbol{\theta}')$ to model the similarity between $\eta(\mathbf{x};\boldsymbol{\theta})$ and $\eta(\mathbf{x};\boldsymbol{\theta}')$.

After a set of $(\mathbf{x},\boldsymbol{\theta})$ are simulated, (6) can be updated to the posterior conditional on the simulation history. By assuming the simulation error, $\varepsilon(\mathbf{x};\boldsymbol{\theta})$, to be normally distributed, the posterior of $\eta(\mathbf{x},\boldsymbol{\theta})$ still remains to be a GP. Moreover, the vector of η at any set of $(\mathbf{x},\boldsymbol{\theta})$s has a multivariate normal distribution according to its posterior, which can be completely characterized by a mean vector and a variance-covariance matrix. See Ungredda et al. (2022) for the mathematical details about the posterior update.

One of the advantages of adopting the GP model is that it is conducive to a sequential algorithm; one can use the current posterior mean and variance at candidate $(\mathbf{x},\boldsymbol{\theta})$s to decide which pair to simulate next. Once a new pair is simulated the GP can be updated easily to the posterior.

The two models discussed above require parametric input models. Response surface modeling for nonparametric input models has been explored less in the simulation optimization literature. Although (4)

is written in the parametric version, the nonparametric functional Taylor series expansion can be adopted to expand $\eta(\mathbf{x};\hat{F})$ as a functional of $\hat{F}$; see Section 17.3.2 of Barton et al. (2022). A nonparametric input model can be incorporated in the GP model as well by adopting a kernel that measures the similarity between two empirical distributions (Xie et al. 2021).

The two models reviewed above can be applied to accommodate a continuous parameter space. However, sometimes Θ can be finite, or continuous Θ may be approximated by a finite set. For instance, if there is no conjugacy to exploit when updating the posterior on the input model parameter, $p(\boldsymbol{\theta}|\mathscr{Z})$, then an ecdf constructed from an MCMC sample can be adopted as an estimate for the posterior. In this case, the support of the ecdf is a finite set of sampled $\boldsymbol{\theta}$. If Θ is a subset of a continuous parameter space within which the discussed models can be defined, then the same models can be utilized to make predictions at discrete θs.

4.3 Effect of Common Input Data on Comparison of Solutions

In the simulation optimization context, it is often of interest to compare the means of two candidate solutions $\mathbf{x}$ and $\mathbf{x}'$; e.g., for a maximization problem, we prefer the solution with the larger mean. In this section, we discuss a modeling technique, the *common-input-data (CID) effect*, that improves the statistical efficiency of the comparison between two solutions when they share the common input model.

First discussed in Song and Nelson (2019), the CID effect models how $\eta(\mathbf{x};F)$ and $\eta(\mathbf{x}';F)$ are affected differently by the common estimated input model, F. Suppose the decision-maker has adopted a parametric input model. Taking the linear model as an example, consider the difference,

$$\eta(\mathbf{x};\boldsymbol{\theta})-\eta(\mathbf{x}';\boldsymbol{\theta})\approx\eta(\mathbf{x};\boldsymbol{\theta}^c)-\eta(\mathbf{x}';\boldsymbol{\theta}^c)+\left\{\nabla_{\boldsymbol{\theta}}\eta(\mathbf{x};\boldsymbol{\theta})-\nabla_{\boldsymbol{\theta}}\eta(\mathbf{x}';\boldsymbol{\theta})\right\}^\top(\boldsymbol{\theta}-\boldsymbol{\theta}^c).$$

The true difference, $\eta(\mathbf{x};\boldsymbol{\theta}^c)-\eta(\mathbf{x};\boldsymbol{\theta}^c)$, is what we want to make inference on to correctly compare $\mathbf{x}$ and $\mathbf{x}'$. The additional term, $\{\nabla_{\boldsymbol{\theta}}\eta(\mathbf{x};\boldsymbol{\theta})-\nabla_{\boldsymbol{\theta}}\eta(\mathbf{x}';\boldsymbol{\theta})\}^\top(\boldsymbol{\theta}-\boldsymbol{\theta}^c)$, characterizes the difference in the CID effects at $\mathbf{x}$ and $\mathbf{x}'$. If $\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}$ admits (2), then the difference in the CID effects can be shown to have the following normal distribution as the input data sample size increases:

$$\sqrt{m}(\eta(\mathbf{x};\boldsymbol{\theta})-\eta(\mathbf{x};\boldsymbol{\theta}))\xrightarrow{D}\mathcal{N}\left(\mathbf{0},\left\{\nabla_{\boldsymbol{\theta}}\eta(\mathbf{x};\boldsymbol{\theta})-\nabla_{\boldsymbol{\theta}}\eta(\mathbf{x}';\boldsymbol{\theta})\right\}^\top\Sigma(\boldsymbol{\theta}^c)\left\{\nabla_{\boldsymbol{\theta}}\eta(\mathbf{x};\boldsymbol{\theta})-\nabla_{\boldsymbol{\theta}}\eta(\mathbf{x}';\boldsymbol{\theta})\right\}\right) \quad (8)$$

Notice that the variance of (8) depends on the difference in the gradients of $\eta(\cdot;\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$. Consider the simplest case when the gradients at all solutions are equal. Then, the variance term vanishes to 0 and the problem can be solved as if there is no input uncertainty.

In general, $\nabla_{\boldsymbol{\theta}}\eta(\mathbf{x};\boldsymbol{\theta})\neq\nabla_{\boldsymbol{\theta}}\eta(\mathbf{x}';\boldsymbol{\theta})$ for $\mathbf{x}\neq\mathbf{x}'$ for non-trivial problems. Nevertheless, if the variance of (8) is smaller, then the inference on the true mean difference, $\eta(\mathbf{x};\boldsymbol{\theta}^c)-\eta(\mathbf{x}';\boldsymbol{\theta}^c)$, can be made more precisely. If $\eta(\mathbf{x};\boldsymbol{\theta})$ is sufficiently smooth in $\mathbf{x}$, then the gradient difference at the same $\boldsymbol{\theta}$ tends to be small in the neighborhood of a good solution, e.g., $\mathbf{x}^*(F(\boldsymbol{\theta}^c))$, making the local comparison easier. This links back to the comment made in Section 1 such that the comparison of solutions may be made more precisely than inferring a single solution's true mean under input uncertainty. Several simulation optimization approaches introduced in Section 5 exploit (8) to design algorithms more robust to input model risk.

5 PROBLEM FORMULATIONS AND SOLUTION METHODS

In this section, we discuss problem formulations and solution methods for each of the three problem contexts discussed in Section 3. We first provide minimal background on classical simulation optimization, where input uncertainty is not considered, to set up the discussion; F is adopted to denote a generic input model. We refer the readers to Fu (2015) for a comprehensive overview.

A simulation optimization algorithm allocates simulation replications to a set of solutions to learn the unknown responses until it either runs out of the simulation budget (fixed budget) or achieves a statistical guarantee that the algorithm is designed to provide (fixed precision). Let the solution returned by an

algorithm after solving Opt(F) be $\hat{\mathbf{x}}(F)$. Because $\hat{\mathbf{x}}(F)$ is dependent on the simulation sample paths generated within the algorithm run, it is difficult to provide a deterministic guarantee for the optimality of $\hat{\mathbf{x}}(F)$. Instead, several statistical guarantees have been considered in the literature. For instance, the suboptimality of $\hat{\mathbf{x}}(F)$ can be bounded by some tolerable error $\varepsilon > 0$ with at least $1-\alpha$ confidence: $\mathbb{P}\{\eta(\mathbf{x}^*(F);F) - \eta(\hat{\mathbf{x}}(F);F) \leq \varepsilon\} \geq 1-\alpha$, where $\alpha \in (0,1)$ and the probability is taken with respect to the simulation sample path generated by the algorithm. Different algorithms provide different guarantees, which is an important factor for the decision-maker to consider when selecting an algorithm.

When $\mathscr{X}$ is countable and $|\mathscr{X}| = k$ is a relatively small integer, the problem is often referred to as ranking and selection (R&S). R&S can accommodate the case when the solutions are categorical, e.g., different priority policies for a queue. When discussing R&S, we replace $\mathbf{x}$ with index i of the solution to write $\eta(i;F)$ and $i^*(F)$ instead of $\eta(\mathbf{x};F)$ and $\mathbf{x}^*(F)$, respectively. Similarly, $\hat{i}(F)$ denotes the solution returned by an R&S procedure. A popular statistical guarantee for an R&S algorithm is the probability of correct selection, PCS $\triangleq \mathbb{P}\{\hat{i}(F) = i^*(F)\}$.

The subsections are organized as follows: Section 5.1 discusses the fixed batch data problems while Sections 5.2 and 5.3 respectively concern the streaming data and active data collection problems.

5.1 Fixed Batch Data

When the decision-maker is given a fixed batch of input data, input uncertainty of the problem cannot be further reduced. As mentioned in Section 3, there are two streams of research that account for the input uncertainty in this context: one aims to make inference on the identity of $\mathbf{x}^*(F^c)$, while the other focuses on reformulating Opt(F^c) to hedge against the risk of not knowing F^c. Both approaches only control simulation error, not input uncertainty. In Section 5.1.1, we introduce the inference approach and Section 5.1.2 discusses the reformulation approach.

5.1.1 Inference Approach

Song et al. (2015) show that one may not find $\mathbf{x}^*(F^c)$ with the desired probability guarantee (e.g., 95%) in general when the input data size is finite. Intuitively, when the best solution ($\mathbf{x}^*$) and a close contender ($\mathbf{x}'$) have very similar mean performances under $\boldsymbol{\theta}^c$, then large input uncertainty makes it difficult to determine the sign of $\eta(\mathbf{x}^*;F^c) - \eta(\mathbf{x}';F^c)$ with high probability. This can be easily seen from (8). Moreover, recall that $\eta(\mathbf{x};F)$ must be estimated by running simulations at $(\mathbf{x},F)$, which makes it even harder to infer the true mean difference when the simulation error in the estimator is convoluted with input uncertainty.

One way to handle this challenge is to modify the objective from finding $\mathbf{x}^*(F^c)$ to returning a set of solutions that includes $\mathbf{x}^*(F^c)$ with high probability. For the latter, any target probability $1-\alpha$ is achievable by simply returning the entire feasible solution space, $\mathscr{X}$. Of course, this is extremely naïve. The key is to design a procedure that returns the smallest set of solutions that are statistically indistinguishable from $\mathbf{x}^*(F^c)$ given the finite input data.

To achieve this in the R&S context, Song and Nelson (2019) adopt (8) to explicitly model the difference in the CID effects. They extend the classical multiple comparisons with the best (MCB) algorithm to incorporate both input uncertainty and simulation error. The goal of the classical MCB is to find k simultaneous confidence intervals (CIs) $[L_i, U_i], 1 \leq i \leq k$, that satisfy

$$\mathbb{P}\{\eta(i;F^c) - \max_{\ell \neq i} \eta(\ell;F^c) \in [L_i, U_i], 1 \leq i \leq k\} \geq 1-\alpha. \tag{9}$$

In words, $[L_i, U_i]$ covers the difference between the ith solution's mean performance and the best of the rest's. Consequently, the set of solutions that have positive MCB upper bounds, $\mathscr{I} = \{1 \leq i \leq k : U_i > 0\}$, is guaranteed to contain the best solution with probability $\geq 1-\alpha$. Chang and Hsu (1992) show that $\{L_i, U_i\}_{i=1}^k$ satisfying (9) can be obtained from $\{w_{i\ell}, i \neq \ell\}$ satisfying the following one-sided simultaneous CIs for all $1 \leq i \leq k$:

$$\mathbb{P}\{\eta(i;F^c) - \eta(\ell;F^c) \geq w_{i\ell}, \forall \ell \neq i\} \geq 1-\alpha. \tag{10}$$

Indeed, the same approach can be adopted to incorporate input uncertainty if one can find $\{w_{i\ell}, i \neq \ell\}$ that reflect input uncertainty to provide $1-\alpha$ coverage probability.

Suppose a frequentist parametric input model is adopted to estimate $\boldsymbol{\theta}^c$ with $\hat{\boldsymbol{\theta}}$. Running r replications of all k solutions adopting $F(\hat{\boldsymbol{\theta}})$ as the common input model, one can obtain the sample average, $\bar{Y}(i;\hat{\boldsymbol{\theta}}) = \sum_{j=1}^{r} Y_j(i;\hat{\boldsymbol{\theta}})/r$, for all $1 \leq i \leq k$, where $Y_j(i;\hat{\boldsymbol{\theta}})$ is the simulation output from the jth replication at $(i,\hat{\boldsymbol{\theta}})$. Consider a pair of solutions $i \neq \ell$. Then, their sample mean difference can be decomposed as

$$\begin{aligned}\bar{Y}(i;\hat{\boldsymbol{\theta}}) - \bar{Y}(\ell;\hat{\boldsymbol{\theta}}) =& \eta(i;\hat{\boldsymbol{\theta}}) - \eta(\ell;\hat{\boldsymbol{\theta}}) + \bar{\varepsilon}(i;\hat{\boldsymbol{\theta}}) - \bar{\varepsilon}(\ell;\hat{\boldsymbol{\theta}}) \\ \approx& \eta(i;\boldsymbol{\theta}^c) - \eta(\ell;\boldsymbol{\theta}^c) + \{\nabla_{\boldsymbol{\theta}}\eta(i;\boldsymbol{\theta}^c) - \nabla_{\boldsymbol{\theta}}\eta(\ell;\boldsymbol{\theta}^c)\}^\top(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^c) + \bar{\varepsilon}(i;\hat{\boldsymbol{\theta}}) - \bar{\varepsilon}(\ell;\hat{\boldsymbol{\theta}}), \quad (11)\end{aligned}$$

where $\bar{\varepsilon}(i;\hat{\boldsymbol{\theta}}) \triangleq \bar{Y}(i;\hat{\boldsymbol{\theta}}) - \eta(i;\hat{\boldsymbol{\theta}})$ and the approximation in the second line is from (4). Therefore, by deriving the joint distribution of $\{\nabla_{\boldsymbol{\theta}}\eta(i;\boldsymbol{\theta}^c) - \nabla_{\boldsymbol{\theta}}\eta(\ell;\boldsymbol{\theta}^c)\}^\top(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^c) + \bar{\varepsilon}(i;\hat{\boldsymbol{\theta}}) - \bar{\varepsilon}(\ell;\hat{\boldsymbol{\theta}})$ for all $\ell \neq i$, one can find $\{w_{i\ell}, \ell \neq i\}$ satisfying (10).

Finding the joint distribution of $\bar{\varepsilon}(i;\hat{\boldsymbol{\theta}}) - \bar{\varepsilon}(\ell;\hat{\boldsymbol{\theta}})$ at all $i \neq \ell$ has long been studied in the classical simulation optimization literature. However, the challenge in (11) is that, the simulation error difference is convoluted with the difference of the CID effects. Song and Nelson (2019) address this by decomposing the CI bounds as $w_{i\ell} = w^1_{i\ell} + w^2_{i\ell}$ for $\ell \neq i$, where $\{w^1_{i\ell}, \ell \neq i\}$ and $\{w^2_{i\ell}, \ell \neq i\}$ satisfy $\mathbb{P}\{\{\nabla_{\boldsymbol{\theta}}\eta(i;\boldsymbol{\theta}^c) - \nabla_{\boldsymbol{\theta}}\eta(\ell;\boldsymbol{\theta}^c)\}^\top(\boldsymbol{\theta} - \boldsymbol{\theta}^c) \geq w^1_{i\ell}, \forall \ell \neq i\} \geq 1-\alpha_1$ and $\mathbb{P}\{\bar{\varepsilon}(i;\hat{\boldsymbol{\theta}}) - \bar{\varepsilon}(\ell;\hat{\boldsymbol{\theta}}) \geq w^2_{i\ell}, \forall \ell \neq i\} \geq 1-\alpha_2$, respectively. Namely, the former and the latter respectively provide the simultaneous CI widths for the CID effects and the simulation errors. Choosing $1-\alpha = (1-\alpha_1)(1-\alpha_2)$, (10) can be guaranteed asymptotically as m increases, which justifies adopting $w_{i\ell} = w^1_{i\ell} + w^2_{i\ell}$ to construct the MCB CIs and find $\mathscr{I}$. Here, note that the asymptotic framework of $m \to \infty$ does not imply that the input data size increases; it is to provide an assurance that for sufficiently large m, the asymptotic scheme can provide an effective approximation. If the returned $\mathscr{I}$ contains a single solution, then the solution has a high probability of being $i^*(\boldsymbol{\theta}^c)$ even in the presence of input uncertainty.

Corlu and Biller (2013) take a similar modeling approach as above to return a set of solutions containing the true best. Instead of the MCB, they extend the classical subset selection algorithm.

A clear downside of the inference approach is its inability to guarantee $|\mathscr{I}| = 1$. If input uncertainty is large, then $\mathscr{I}$ is likely to contain many solutions, which may not be too useful for the decision-maker. Nevertheless, a large $|\mathscr{I}|$ suggests that the input uncertainty is significant for the problem at hand and the decision-maker should try to collect more data if it is feasible at all to reduce the input uncertainty, or choose a solution robust against input uncertainty as discussed in the following section.

5.1.2 Reformulation Approach

The second approach reformulates Opt(F^c) to account for that F^c is estimated based on finite data. Problem formulations fall under this category reflect uncertainty about F^c in the objective function so that the solution to the modified problems can be found by algorithms designed to control the simulation error.

Assuming a Bayesian parametric input model, Wu et al. (2018) connect several reformulations proposed in the literature, which we explore below. Consider the following problem formulation:

$$\mathbf{x}^*_\rho = \arg\max_{\mathbf{x}\in\mathscr{X}} \rho_{\boldsymbol{\theta}}[\eta(\mathbf{x};\boldsymbol{\theta})], \quad (12)$$

where $\rho_{\boldsymbol{\theta}}$ denotes a statistical measure taken with respect to the distribution of $\boldsymbol{\theta}$. The decision-maker's willingness to take the risk of not knowing $\boldsymbol{\theta}^c$ when choosing a solution to implement in the system can be coded into $\rho_{\boldsymbol{\theta}}$. For example, a risk-neutral decision-maker can take $\rho_{\boldsymbol{\theta}}$ as the expectation,

$$\rho_{\boldsymbol{\theta}}[\eta(\mathbf{x};\boldsymbol{\theta})] = \mathbb{E}_{\boldsymbol{\theta}}[\eta(\mathbf{x};\boldsymbol{\theta})] = \int \eta(\mathbf{x};\boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathscr{Z}) \mathrm{d}\boldsymbol{\theta}, \quad (13)$$

where $p(\boldsymbol{\theta}|\mathscr{Z})$ is the posterior distribution of $\boldsymbol{\theta}$ defined in (3). Adopting (13) as the objective function of (12) implies that $\eta(\mathbf{x};\boldsymbol{\theta}^c)$ is replaced with the average of $\eta(\mathbf{x};\boldsymbol{\theta})$ over all possible realization of $\boldsymbol{\theta}$.

However, this may expose the decision-maker to input model risk if there is larger uncertainty about $\boldsymbol{\theta}$ and each solution's performance varies significantly at different values of $\boldsymbol{\theta}$; recall the example in Figure 1.

To make the problem robust to input uncertainty, one can adopt a risk measure for $\rho_{\boldsymbol{\theta}}$. For example, given $\alpha \in (0,1)$, one can choose α-quantile of $\eta(\mathbf{x};\theta)$, $\rho_{\boldsymbol{\theta}}[\eta(\mathbf{x};\boldsymbol{\theta})] = q_\alpha(\eta(\mathbf{x};\boldsymbol{\theta}))$. For small α, the quantile provides a robust assessment (pessimistic) on the performance of $\mathbf{x}$ with respect to uncertainty about $\boldsymbol{\theta}$. A criticism for the quantile function, however, is that it cannot detect when $\eta(\mathbf{x};\boldsymbol{\theta})$ below the α-quantile is extremely poor. Alternatively, the conditional expectation, $\rho_{\boldsymbol{\theta}}[\eta(\mathbf{x};\boldsymbol{\theta})] = \mathbb{E}_{\boldsymbol{\theta}}[\eta(\mathbf{x};\boldsymbol{\theta})|\eta(\mathbf{x};\boldsymbol{\theta}) \leq q_\alpha(\eta(\mathbf{x};\boldsymbol{\theta}))]$, can be adopted to compare the average performance on the lower extreme tail. If we take $\alpha = 0$ for either choice, $\rho_{\boldsymbol{\theta}}[\eta(\mathbf{x};\boldsymbol{\theta})]$ is equivalent to $\min_{\boldsymbol{\theta}\in\Theta} \eta(\mathbf{x};\boldsymbol{\theta})$ and (12) reduces to

$$\mathbf{x}^*_\rho = \arg\max_{\mathbf{x}\in\mathscr{X}} \min_{\boldsymbol{\theta}\in\Theta} \eta(\mathbf{x};\boldsymbol{\theta}), \tag{14}$$

which can be linked to the distributionally robust optimization (DRO) framework (Delage and Ye 2010). In DRO, Θ is referred to as the ambiguity set as it contains the values that the uncertain $\boldsymbol{\theta}^c$ can take. Clearly, (14) is more conservative as it evaluates each $\mathbf{x}$ based on its worst-case performance among $\boldsymbol{\theta} \in \Theta$.

As mentioned in the beginning of this section, a benefit of the reformulation approach is that once the choice of $\rho_{\boldsymbol{\theta}}$ is made, only simulation error needs to be controlled to solve (12) as input uncertainty is already reflected in the objective function by $\rho_{\boldsymbol{\theta}}$. For instance, focusing on the R&S context with finite ambiguity set $\mathscr{F}$, Gao et al. (2017) propose an R&S algorithm to solve (14). They redefine PCS to evaluate if the solution returned from the algorithm, $\hat{i}_\rho$, matches $i^*_\rho \triangleq \arg\max_{1\leq i\leq k} \min_{F\in\mathscr{F}} \eta(i;F)$. Equivalently, $\text{PCS} = \mathbb{P}\left\{\min_{F\in\mathscr{F}} \eta(\hat{i}_\rho;F) \geq \max_{1\leq i\leq k} \min_{F\in\mathscr{F}} \eta(i;F)\right\}$. They formulate a budget allocation problem to determine fixed sampling ratios for all (i,F) pairs to maximize the convergence rate of the PCS when the simulation budget is infinite, and solve it with some approximations. The resulting sampling ratios are then utilized to design a sequential sampling algorithm when the simulation budget is finite.

There are several frameworks in the literature that adopt the risk-neutral measure for reformulation. Corlu and Biller (2015) propose a subset selection algorithm to return a set containing i^*_ρ when $\rho_{\boldsymbol{\theta}}$ is the expectation. With the same choice for $\rho_{\boldsymbol{\theta}}$, Pearce and Branke (2017) and Wang et al. (2020) extend the Bayesian optimization framework to incorporate input uncertainty. Meanwhile, Fan et al. (2020) consider the same problem setting as Gao et al. (2017) and provide a fixed precision R&S algorithm to find i^*_ρ.

5.2 Streaming Data

In the streaming data environment, input uncertainty can be reduced when additional data is gathered over time. In this section, we focus on the case where the additional data are made available periodically and a system decision can be updated at the end of each period reflecting the changes due to the additional data. As the data streaming process is beyond the decision-maker's control, the algorithms designed to solve these problems can only control the simulation error. Thus, the focus is on incorporating simulation and optimization methods to account for progressively decreasing input uncertainty and allocating the simulation budget efficiently to update each period's decision.

To make the discussion concrete, consider a time horizon with discrete time points that define periods, $p = 1,2,\ldots,P$. In period p, additional input data of size Δm_p are gathered at the beginning of the period, and the updated decision $\mathbf{x}_p$ is required by the end of the period. Suppose the decision-maker adopts a parametric frequentist input model and estimates $\boldsymbol{\theta}^c$ by its MLE. As new data comes in at each period, it is sensible to update the MLE so that the simulator can be run with the most up-to-date input model. Namely, at the beginning of the pth period, using all cumulative data, $m_p \triangleq \sum_{i=1}^{p} \Delta m_i$, the MLE, $\hat{\boldsymbol{\theta}}_p$, can be updated. Since the simulator's input model is parameterized with the updated $\hat{\boldsymbol{\theta}}_p$, then the corresponding simulator optimization problem is also updated to $\text{Opt}(\hat{\boldsymbol{\theta}}_p)$. One can expect its optimal solution $\mathbf{x}^*(\hat{\boldsymbol{\theta}}_p)$ to be closer to the true optimum, $\mathbf{x}^*(\boldsymbol{\theta}^c)$ as p increases because $\hat{\boldsymbol{\theta}}_p$ approaches $\boldsymbol{\theta}^c$. Of course, no simulation optimization algorithm can solve $\text{Opt}(\hat{\boldsymbol{\theta}}_p)$ to optimality in finite time due to the simulation error. Instead, it returns estimator $\mathbf{x}_p$ of $\mathbf{x}^*(\boldsymbol{\theta}_p)$ after spending a finite simulation budget. Thus, if $\mathbf{x}_p$ is a poor estimator

of $\mathbf{x}^*(\boldsymbol{\theta}_p)$, we may not be able to achieve convergence of $\mathbf{x}_p$ to $\mathbf{x}^*(\boldsymbol{\theta}^c)$. To design a good algorithm, a criterion is needed to determine how good $\mathbf{x}_p$ returned at the end of each period p is.

Song and Shanbhag (2019) suggest that since $\mathbf{x}_p$ is adopted in the system whose stochasticity is characterized by $F(\boldsymbol{\theta}^c)$, an appropriate metric is the cumulative expected suboptimality of the sequence, $\{\mathbf{x}_i\}_{i=1}^p$, for Opt($\boldsymbol{\theta}^c$): $\sum_{i=1}^p \mathbb{E}[\eta(\mathbf{x}^*(\boldsymbol{\theta}^c);\boldsymbol{\theta}^c) - \eta(\mathbf{x}_i;\boldsymbol{\theta}^c)]$. The ith summand evaluates how much loss in the objective function value there is if $\mathbf{x}_i$ is adopted in the system instead of the true optimum, $\mathbf{x}^*(\boldsymbol{\theta}^c)$, where the expectation is with respect to the simulation sample path generated by the algorithm as well as the sampling distribution of $\hat{\boldsymbol{\theta}}_p$. Intuitively, the suboptimality may be caused by the mismatch between (i) Opt($\hat{\boldsymbol{\theta}}_p$) and Opt($\boldsymbol{\theta}^c$); and (ii) the stochastic error in solving the former. Although (i) cannot be controlled in the streaming data environment, (ii) can be reduced by controlling the algorithm that solves Opt($\hat{\boldsymbol{\theta}}_p$).

To control (ii), consider the following decomposition of the suboptimality at the pth period:

$$\begin{aligned}|\eta(\mathbf{x}_p;\boldsymbol{\theta}^c) - \eta(\mathbf{x}^*(\boldsymbol{\theta}^c);\boldsymbol{\theta}^c)| \le &\underbrace{|\eta(\mathbf{x}_p;\boldsymbol{\theta}^c) - \eta(\mathbf{x}_p;\hat{\boldsymbol{\theta}}_p)|}_{(a)} + \underbrace{|\eta(\mathbf{x}_p;\hat{\boldsymbol{\theta}}_p) - \eta(\mathbf{x}^*(\hat{\boldsymbol{\theta}}_p);\hat{\boldsymbol{\theta}}_p)|}_{(b)} \\ &+ \underbrace{|\eta(\mathbf{x}^*(\hat{\boldsymbol{\theta}}_p);\hat{\boldsymbol{\theta}}_p) - \eta(\mathbf{x}^*(\boldsymbol{\theta}^c);\hat{\boldsymbol{\theta}}_p)|}_{(c)} + \underbrace{|\eta(\mathbf{x}^*(\boldsymbol{\theta}^c);\hat{\boldsymbol{\theta}}_p) - \eta(\mathbf{x}^*(\boldsymbol{\theta}^c);\boldsymbol{\theta}^c)|}_{(d)}.\end{aligned}$$

Here, (a) represents Solution $\mathbf{x}_p$'s performance gap for Opt($\boldsymbol{\theta}^c$) and Opt($\hat{\boldsymbol{\theta}}_p$); and (d) shows the same difference for $\mathbf{x}^*(\boldsymbol{\theta}^c)$. Term (b) is the suboptimality of $\mathbf{x}_p$ for Opt($\hat{\boldsymbol{\theta}}_p$); and (c) represents the performance gap between $\mathbf{x}^*(\boldsymbol{\theta}^c)$ and $\mathbf{x}^*(\hat{\boldsymbol{\theta}}_p)$ for Opt($\hat{\boldsymbol{\theta}}_p$). Among these, only (b) is controllable by adopting a suitable simulation optimization algorithm while the other three terms are caused by the estimation error in $\hat{\boldsymbol{\theta}}_p$ and can only be diminished by collecting more input data. Nevertheless, the decomposition above provides a useful insight on how to determine the precision at which Opt($\boldsymbol{\theta}_p$) should be solved. Namely, spending a large simulation budget to close the suboptimality in (b) is wasteful if all other parts are dominant due to large remaining input uncertainty at period p.

Focusing on continuous $\mathscr{X}$, He et al. (2024) design an efficient algorithm to solve the streaming data problem by deriving asymptotic upper bounds for parts (a)–(d) under some smoothness assumptions on η,

$$\mathbb{E}[\eta(\mathbf{x}_p;\boldsymbol{\theta}^c) - \eta(\mathbf{x}^*(\boldsymbol{\theta}^c);\boldsymbol{\theta}^c)] \le \gamma_1 \mathbb{E}[\|\mathbf{x}_p - \mathbf{x}^*(\hat{\boldsymbol{\theta}}_p)\|^2] + \gamma_2 \mathbb{E}[\|\hat{\boldsymbol{\theta}}_p - \boldsymbol{\theta}^c\|^2], \tag{15}$$

where γ_1 and γ_2 are constants associated with the smoothness conditions. The solution approach they adopt is stochastic approximation (SA), which takes a editsstochastic gradient ascent step at each iteration, where the gradient is estimated via simulations (Shapiro et al. 2021). Namely, the first term of the upper bound in (15) is derived from (b). Indeed, $\mathbb{E}[\|\mathbf{x}_p - \mathbf{x}^*(\hat{\boldsymbol{\theta}}_p)\|^2]$ can be controlled by choosing the numper of SA iterations, n_p, taken in the pth period. On the other hand, the second term in (15) has the asymptotic convergence rate of $\mathscr{O}(m_p^{-1})$ from (2). This bound implies that choosing large n_p for earlier period p may be unnecessary—even with $n_p = \infty$, the upper bound in (15) may be large if $\hat{\boldsymbol{\theta}}_p$ has large error. Conversely, when $\hat{\boldsymbol{\theta}}_p$ is more precise, larger n_p is favored to obtain a tighter upper bound. Under some regularity conditions, the first term can be shown to diminish in $\mathscr{O}(1/n_p)$ implying that matching n_p with m_p attain the best convergence rate of the expected suboptimality while minimizing the computational cost.

Liu et al. (2024) study a similar algorithm to solve a simulation optimization problem with continuous $\mathscr{X}$ while adopting a Bayesian input model and (12) as the objective, where $\rho_{\boldsymbol{\theta}}$ is the risk-neutral measure (mean). The streaming data problems have also been considered in the R&S context. Wang and Zhou (2022) propose an optimal computing budget allocation procedure, where simulation size is decided in each period to maximize the convergence rate of the probability of false selection. Wu et al. (2024) design two sequential eliminating algorithms with confidence bands on the solution accounting for the decreasing input uncertainty.

5.3 Active Input Data Collection

If collecting additional data is feasible for the system in consideration, then the decision-maker can choose to acquire additional input data at a cost should it help finding $\mathbf{x}^*(F^c)$. Here, the aim is to find $\mathbf{x}^*(\boldsymbol{\theta}^c)$ by actively collecting input data to improve the simulation model. In this case, both simulation error and input uncertainty can be controlled.

To facilitate the discussion, we assume that there are L independent input sources in the system and corresponding L input distributions. We explicitly define F as a collection of L input models, $F = \{F_1, F_2, \ldots, F_L\}$, where each F_ℓ is estimated from m_ℓ i.i.d. observations from the ℓth true input distribution, F_ℓ^c. Furthermore, we modify the definition of m as $m = \sum_{\ell=1}^{L} m_\ell$. In the parametric case, for each ℓ, $F_\ell^c = F(\vartheta_\ell^c)$ and $F_\ell = F(\hat{\vartheta}_\ell)$, where ϑ_ℓ^c and $\hat{\vartheta}_\ell$ are the true and the estimated parameter vectors, respectively. We still adopt $\boldsymbol{\theta} = (\vartheta_1, \vartheta_2, \ldots, \vartheta_L)$ to represent the parameter vector for all L input models.

Kim and Song (2022) consider this problem by adopting a Bayesian parametric input model in the R&S context. Given the posterior, $p(\boldsymbol{\theta}|\mathscr{Z})$, Kim et al. (2021) define the following posterior probability each Solution i being optimal as the *posterior preference* for Solution i:

$$P_{i,m} \triangleq \int_{\boldsymbol{\theta}\in\Theta} \mathbf{1}\left(i = i^*(\boldsymbol{\theta})\right) p(\boldsymbol{\theta}|\mathscr{Z}) d\boldsymbol{\theta}. \tag{16}$$

From the posterior preferences of all solutions, the maximum a posteriori estimator (MAP) of $\mathbf{x}^*(\boldsymbol{\theta}^c)$ can be found as $\tilde{i}_m \triangleq \arg\max_{1\le i\le k} P_{i,m}$. Kim et al. (2021) refer to $\tilde{i}_m$ as the *most probable best* (MPB). Although suppressed from notation, $P_{i,m}$ and i_m depend on $\mathscr{Z}$.

The MPB has several features suitable for designing an active input data collection algorithm. First, when all L input processes data are collected infinitely many times, the posterior preference of the MPB converges to one. Since $p(\boldsymbol{\theta}|\mathscr{Z})$ concentrates the probability mass at $\boldsymbol{\theta}^c$, this implies that the MPB converges to $i^*(\boldsymbol{\theta}^c)$ with probability one. More interestingly, the convergence rate of the posterior preference of the MPB can be characterized as a function of fractional sample sizes $\beta_\ell \triangleq m_\ell/m$ for $1 \le \ell \le L$, if each β_ℓ converges to a constant when m increases:

$$\lim_{m\to\infty} -\frac{1}{m}\log(1 - P_{\tilde{i}_m,m}) = \inf_{\boldsymbol{\theta}\notin\Theta_{i^*(\boldsymbol{\theta}^c)}} \sum_{\ell=1}^{L} \beta_\ell \mathrm{KL}(\vartheta_\ell^c||\vartheta_\ell) \text{ almost surely}, \tag{17}$$

where $\Theta_{i^*(\boldsymbol{\theta}^c)} \triangleq \{\boldsymbol{\theta}|i^*(\boldsymbol{\theta}) = i^*(\boldsymbol{\theta}^c)\}$ and $\mathrm{KL}(\vartheta_\ell^c||\vartheta_\ell)$ is the Kullback-Liebler (KL) divergence between $F_\ell(\vartheta_\ell^c)$ and $F_\ell(\vartheta_\ell)$, which measures how much the two distributions differ. By definition, $\boldsymbol{\theta}^c \in \Theta_{i^*(\boldsymbol{\theta}^c)}$. Thus, (17) characterizes the logarithmic convergence rate of the MPB's preference probability to one.

To gain the insight, recall that $1 - P_{\tilde{i}_m,m}$ is the probability assigned to the event of $\boldsymbol{\theta}$ ending up outside of $\Theta_{i^*(\boldsymbol{\theta}^c)}$, which becomes a rare event as $p(\boldsymbol{\theta}|\mathscr{Z})$ concentrates at $\boldsymbol{\theta}^c$. By the large-deviation theory, the convergence rate of the probability of a union of rare events is determined by the most likely event among them (Hollander 2000). Loosely speaking, out of all $\boldsymbol{\theta} \notin \Theta_{i^*(\boldsymbol{\theta}^c)}$, the $\boldsymbol{\theta}$ with the minimal weighted KL divergence in (17) is the $\boldsymbol{\theta}$ assigned with the largest probability mass in its neighborhood by $p(\boldsymbol{\theta}|\mathscr{Z})$ as m increases (i.e., most likely $\boldsymbol{\theta}$), if the input data from L sources are collected according to the sampling ratios, $\{\beta_\ell\}_{\ell=1}^L$.

Hence, by finding $\{\beta_\ell\}_{\ell=1}^L$ that maximize the right-hand side of (17), the input data collection can be optimized to achieve the fastest convergence rate of the MPB's posterior preference. In general, this maximization problem is challenging to solve when Θ is continuous. Moreover, one must know $\boldsymbol{\theta}^c$ and $\Theta_{i^*(\boldsymbol{\theta}^c)}$ to be able to solve the problem. For the former, a natural plug-in estimator is the MAP of $\boldsymbol{\theta}^c$ given $p(\boldsymbol{\theta}|\mathscr{Z})$. The latter must be learned by estimating $\eta(i;\boldsymbol{\theta})$ at each $(i,\boldsymbol{\theta})$ via simulations. For the case when Θ is finite, Kim and Song (2022) present a sequential sampling algorithm that incorporates both simulation sampling and input data collection.

Wang and Zhou (2023) also considers a R&S problem, where additional data collection scheme is developed based on asymptotic normality of the point estimator (cf. (2)) assuming a frequentist parametric

input model. The Bayesian optimization algorithm proposed by Ungredda et al. (2022) is more flexible than the previous two as it can incorporate continuous $\mathscr{X}$ and Θ.

6 REMAINING RESEARCH QUESTIONS

Although significant attention has been brought to input uncertainty quantification in the literature, simulation optimization under input uncertainty is explored much less. In particular, there are relatively fewer algorithms to solve streaming input data or active input data collection problems. Moreover, there is lack of frameworks to accommodate dependent input data such as time series. All works reviewed in this tutorial assume i.i.d. input data. Also, majority of them adopts parametric input models by assuming their distribution families are known. Although it makes theoretical analyses more convenient, it is difficult to hold in practice.

Depending on the system, there may be feedback between the solution implemented by the decision-maker and the input-generating process, which was not discussed in this tutorial. Incorporating such dependence in the active data collection problem is an open question to the best of our knowledge. Also, input uncertainty may not be the only source of model risk. Logical error in the simulation model can be a significant source of model risk that cannot be reduced by methods discussed in this tutorial.

Lastly, we believe that there are significant opportunities to apply simulation optimization methods to solve streaming data and active input data collection problems in various applications where input data are continuously collected and updated via interconnected smart devices (e.g. the network of the internet of things or the digital twin applications). Developing theories to design simulation optimization algorithms that incorporate the domain-specific characteristics of the input data (e.g., consumer survey data in the GM example in Section 1) would enhance the practical impact of this research stream.

ACKNOWLEDGMENTS

This work is supported by National Science Foundation Grant CAREER CMMI-2045400.

REFERENCES

Asmussen, S. and P. W. Glynn. 2007. *Stochastic Simulation: Algorithms and Analysis*. New York: Springer.

Barton, R. R., H. Lam, and E. Song. 2022. "Input Uncertainty in Stochastic Simulation". In *The Palgrave Handbook of Operations Research*, edited by S. Salhi and J. Boylan, 573–620. Cham: Springer International Publishing.

Chang, J. Y. and J. C. Hsu. 1992. "Optimal Designs for Multiple Comparisons with the Best". *Journal of Statistical Planning and Inference* 30(1):45–62.

Corlu, C. G., A. Akcay, and W. Xie. 2020. "Stochastic Simulation under Input Uncertainty: A Review". *Operations Research Perspectives* 7:100162.

Corlu, C. G. and B. Biller. 2013. "A Subset Selection Procedure under Input Parameter Uncertainty". In *2013 Winter Simulations Conference (WSC)*, 463–473 https://doi.org/10.1109/WSC.2013.6721442.

Corlu, C. G. and B. Biller. 2015. "Subset Selection for Simulations Accounting for Input Uncertainty". In *2015 Winter Simulation Conference (WSC)*, 437–446 https://doi.org/10.1109/WSC.2015.7408185.

Delage, E. and Y. Ye. 2010. "Distributionally Robust Optimization under Moment Uncertainty with Application to Data-driven Problems". *Operations Research* 58(3):595–612.

Fan, W., L. J. Hong, and X. Zhang. 2020. "Distributionally Robust Selection of the Best". *Management Science* 66(1):190–208.

Fu, M. C. (Ed.) 2015. *Handbook of Simulation Optimization*, Volume 216. New York: Springer.

Gao, S., H. Xiao, E. Zhou, and W. Chen. 2017. "Robust Ranking and Selection with Optimal Computing Budget Allocation". *Automatica* 81:30–36.

Gershman, S. J. and D. M. Blei. 2012. "A Tutorial on Bayesian Nonparametric Models". *Journal of Mathematical Psychology* 56(1):1–12.

He, L., U. V. Shanbhag, and E. Song. 2024. "Stochastic Approximation for Multi-period Simulation Optimization with Streaming Input Data". *ACM Transactions on Modeling and Computer Simulation* 34(2):1–27.

Hollander, F. 2000. *Large Deviations*. American Mathematical Society.

Kim, K.-K., T. Kim, and E. Song. 2021. "Selection of the Most Probable Best under Input Uncertainty". In *2021 Winter Simulation Conference (WSC)*, 1–12 https://doi.org/10.1109/WSC52266.2021.9715474.

Kim, T. and E. Song. 2022. "Optimizing Input Data Acquisition for Ranking and Selection: A View Through the Most Probable Best". In *2022 Winter Simulation Conference (WSC)*, 2258–2269 https://doi.org/10.1109/WSC57314.2022.10015453.

Lam, H. 2016. "Advanced Tutorial: Input Uncertainty and Robust Analysis in Stochastic Simulation". In *2016 Winter Simulation Conference (WSC)*, 178–192 https://doi.org/10.1109/WSC.2016.7822088.

Liu, T., Y. Lin, and E. Zhou. 2024. "Bayesian Stochastic Gradient Descent for Stochastic Optimization with Streaming Input Data". *SIAM Journal on Optimization* 34(1):389–418.

Pearce, M. and J. Branke. 2017. "Bayesian Simulation Optimization with Input Uncertainty". In *2017 Winter Simulation Conference (WSC)*, 2268–2278 https://doi.org/10.1109/WSC.2017.8247958.

Santner, T., B. Williams, B. Williams, and W. Notz. 2003. *The Design and Analysis of Computer Experiments*. New York: Springer.

Shapiro, A., D. Dentcheva, and A. Ruszczynski. 2021. *Lectures on Stochastic Programming: Modeling and Theory*. SIAM.

Song, E. and B. L. Nelson. 2019. "Input–output Uncertainty Comparisons for Discrete Optimization via Simulation". *Operations Research* 67(2):562–576.

Song, E., B. L. Nelson, and L. J. Hong. 2015. "Input Uncertainty and Indifference-Zone Ranking & Selection". In *2015 Winter Simulation Conference (WSC)*, 414–424 https://doi.org/10.1109/WSC.2015.7408183.

Song, E., B. L. Nelson, and C. D. Pegden. 2014. "Advanced Tutorial: Input Uncertainty Quantification". In *2014 Winter Simulation Conference (WSC)*, 162–176 https://doi.org/10.1109/WSC.2014.7019886.

Song, E. and U. V. Shanbhag. 2019. "Stochastic Approximation for Simulation Optimization under Input Uncertainty with Streaming Data". In *2019 Winter Simulation Conference (WSC)*, 3597–3608 https://doi.org/10.1109/WSC40007.2019.9004677.

Song, E., P. Wu-Smith, and B. L. Nelson. 2020. "Uncertainty Quantification in Vehicle Content Optimization for General Motors". *INFORMS Journal on Applied Analytics* 50(4):225–238.

Ungredda, J., M. Pearce, and J. Branke. 2022. "Bayesian Optimisation vs. Input Uncertainty Reduction". *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 32(3):1–26.

Van der Vaart, A. W. 1998. *Asymptotic Statistics*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press.

Wang, H., J. Yuan, and S. H. Ng. 2020. "Gaussian Process Based Optimization Algorithms with Input Uncertainty". *IISE Transactions* 52(4):377–393.

Wang, Y. and E. Zhou. 2022. "Fixed Budget Ranking and Selection with Streaming Input Data". In *2022 Winter Simulation Conference (WSC)*, 3027–3038 https://doi.org/10.1109/WSC57314.2022.10015327.

Wang, Y. and E. Zhou. 2023. "Input Data Collection Versus Simulation: Simultaneous Resource Allocation". In *2023 Winter Simulation Conference (WSC)*, 3657–3668 https://doi.org/10.1109/WSC60868.2023.10408130.

Wu, D., Y. Wang, and E. Zhou. 2024. "Data-driven Ranking and Selection under Input Uncertainty". *Operations Research* 72(2):781–795.

Wu, D., H. Zhu, and E. Zhou. 2018. "A Bayesian Risk Approach to Data-driven Stochastic Optimization: Formulations and Asymptotics". *SIAM Journal on Optimization* 28(2):1588–1612.

Xie, W., C. Li, Y. Wu, and P. Zhang. 2021. "A Nonparametric Bayesian Framework for Uncertainty Quantification in Stochastic Simulation". *SIAM/ASA Journal on Uncertainty Quantification* 9(4):1527–1552.

AUTHOR BIOGRAPHIES

LINYUN HE is a Ph.D. candidate in the School of Industrial and Systems Engineering at Georgia Institute of Technology. His research interests include simulation optimization, stochastic optimization, non-parametric methods and high-dimensional statistics. His email address is lhe85@gatech.edu and his website is https://he-linyun.github.io.

EUNHYE SONG is a Coca-Cola Foundation Early Career Professor and Assistant Professor in the H. Milton Stewart School of Industrial and Systems Engineering at Georgia Institute of Technology. She earned her PhD degree in Industrial Engineering and Management Sciences at Northwestern University. Her research interests include simulation design of experiments, uncertainty and risk quantification, and simulation optimization. Her email address is eunhye.song@isye.gatech.edu. Her website is http://eunhyesong.info.