Proceedings of the 2025 Winter Simulation Conference
E. Azar, A. Djanatliev, A. Harper, C. Kogler, V. Ramamohan, A. Anagnostou, and S. J. E. Taylor, eds.

GENERAL-PURPOSE RANKING AND SELECTION FOR STOCHASTIC SIMULATION WITH STREAMING INPUT DATA

Jaime Gonzalez-Hodar[1] and Eunhye Song[1]

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

ABSTRACT

We study ranking and selection (R&S) where the simulator's input models are increasingly more precisely estimated from the streaming data obtained from the system. The goal is to decide when to stop updating the model and return the estimated optimum with a probability of good selection (PGS) guarantee. We extend the general-purpose R&S procedure by Lee and Nelson by integrating a metamodel that represents the input uncertainty effect on the simulation output performance measure. The algorithm stops when the estimated PGS is no less than $1-\alpha$ accounting for both prediction error in the metamodel and input uncertainty. We then propose an alternative procedure that terminates significantly earlier while still providing the same (approximate) PGS guarantee by allowing the performance measures of inferior solutions to be estimated with lower precision than those of good solutions. Both algorithms can accommodate nonparametric input models and/or performance measures other than the means (e.g., quantiles).

1 INTRODUCTION

Simulation is an essential computational tool for evaluating the performances of complex stochastic systems. These simulation models mimic system stochasticity by generating random variates from input distributions and feeding them through the simulation logic to produce outputs. However, the input distributions are typically unknown and must be estimated from the observations from the system. Due to the finiteness of the data, the input models of the simulator are subject to estimation error, which is then propagated to cause additional uncertainty in the simulation output – such additional uncertainty is referred to as input uncertainty. In this paper, we investigate finding the optimum for an R&S problem for the target system when there is input uncertainty.

The incorporation of input uncertainty in simulation optimization has been studied actively in recent years. He and Song (2024) categorize the literature into three problem contexts depending on the availability of input data: fixed batch data, streaming data, and active input data collection problems. Our problem belongs in the second category. We study the case where the input data streams in from the target system and the simulation model can be periodically updated incorporating the latest batch of the data while a one-time static decision is needed for the system. In this problem context, the decision boils down to choose when to stop updating the simulation model and return the estimated optimum.

The existing literature on input uncertainty in the streaming data environment is relatively new. Liu et al. (2021) take a Bayesian approach in where they update the posterior distribution of the input parameter with each batch of incoming data. Wu et al. (2024) develop an algorithm based on a sequential elimination framework, where they consider a streaming data to estimate the parameters of the input distribution with a moving average estimator. He et al. (2024) study a multi-period continuous simulation optimization problem, where in each period they get a fixed number of additional observations where they can re-estimate the input parameters of the simulation model. Wang and Zhou (2025) propose an optimal computing budget allocation scheme that sequentially allocates the simulation budget to solve an R&S problem while periodically updating the input models with the streaming data.

This work investigates the streaming data environment in a multi-period setting, in which we update our estimation of the input distribution in each period and we decide when to stop collecting data. The R&S framework we propose utilizes the general-purpose R&S procedure developed by Lee and Nelson (2016) for a classical R&S problem. The authors do not impose any parametric assumption on the simulation output distributions of the systems. Instead, their work obtains the statistical guarantee for finding the optimum via bootstrapping and shows the procedure is asymptotically consistent as more simulation results accumulate.

The main contribution of this work is three-fold. First, to the best of our knowledge, our framework is the first to tackle R&S under input uncertainty completely nonparametrically, which does not make parametric assumptions on the input models nor the simulation output distributions. Second, our framework can accommodate general performance measures other than the mean such as a quantile or conditional value at risk. Lastly, our procedure provides an asymptotic probability of selecting a solution whose performance measure is within a user-defined threshold from the best, which is known as the PGS in the literature. Again, to the best of our knowledge, we are the first to provide a PGS guarantee for R&S under input uncertainty.

We extend Lee and Nelson (2016) by incorporating input uncertainty into the general-purpose R&S procedure. The procedure selects the best solution for the target system based on the current estimate of the input models. Then, it estimates the probability that the selected solution is indeed the best accounting for both stochastic error in simulation as well as input uncertainty. If there is enough statistical evidence of optimality, the procedure returns the estimated best solution.

The paper is structured as follows. Section 2 defines the streaming data environment and the problem. In Section 3, we introduce the metamodel to predict the performance measure values at different sets of input models and use it to set up an R&S procedure that stops with a PGS guarantee. The procedure is further improved in Section 4, which is designed to require fewer data points and number of simulation replications before stopping. The results from several numerical experiments are shared in Section 5 to demonstrate the performances of the algorithms. We conclude in Section 6.

2 PROBLEM DEFINITION

Consider an R&S problem with k systems, where the goal is to find the one with the best performance measure. Each system's simulation is driven by a set of common input distributions $\mathbf{F}^c$, which consists of L independent input functions $\{F_1^c, \ldots, F_L^c\}$, where the c stands for *correct*. Let $Y_i(\mathbf{F})$ represent the simulation output of system i when the inputs are generated from $\mathbf{F}$ and $\eta_i(\mathbf{F})$ be the simulation output performance measure of system i. For example, if $\eta_i(\mathbf{F})$ represents the mean of the simulation output, then $\eta_i(\mathbf{F}) = \mathrm{E}_{\mathbf{F}}[Y_i(\mathbf{F})]$, where $\mathrm{E}_{\mathbf{F}}[\cdot]$ represents that the expectation is taken with respect to $\mathbf{F}$. Another example for $\eta_i(\mathbf{F})$ is to be the α quantile of the simulation output, i.e., $\eta_i(\mathbf{F}) = q_{\alpha,\mathbf{F}}(Y_i(\mathbf{F}))$. The objective is then to find the system, i^c, that maximizes the performance measure under the correct input distributions:

$$i^c \triangleq \arg\max_{i\in\{1,\ldots,k\}} \eta_i(\mathbf{F}^c).$$

In practice, $\mathbf{F}^c$ is unknown and must be estimated from data collected from the system. We assume that for each $p \in \{1,\ldots,L\}$, a streaming process generates independent and identically distributed (i.i.d.) data, which are collected in batches at discrete time points referred to as periods. To make the problem simpler, we assume that in each period, $\Delta n_p \geq 1$ additional observations are collected for $p \in \{1,\ldots,L\}$; our framework is flexible enough to allow the value of Δn_p varies at each period. We adopt n_p to denote the cumulative data size collected from the pth process. At the end of each period, we update the estimator, $\hat{\mathbf{F}} \triangleq \{\hat{F}_1, \ldots, \hat{F}_L\}$, of $\mathbf{F}^c$ using all $\{n_1, \ldots, n_L\}$ data. If the parametric families of $\mathbf{F}^c$ are assumed known, then estimating $\hat{\mathbf{F}}$ boils down to estimating the unknown parameter vector of $\mathbf{F}^c$. Alternatively, $\hat{\mathbf{F}}$ can be estimated nonparametrically by an empirical distribution function constructed from data. Our framework accommodates both nonparametric and parametric input models.

In general, $\eta_i(\mathbf{F}^c) \neq \eta_i(\hat{\mathbf{F}})$ and the latter is random as $\hat{\mathbf{F}}$ depends on the finite data. Applying a traditional R&S procedure taking $\hat{\mathbf{F}}$ as the true input distribution leads to finding the conditional best given $\hat{\mathbf{F}}$:

$$\arg\max_{i\in\{1,\ldots,k\}} \eta_i(\hat{\mathbf{F}}). \tag{1}$$

Meanwhile, $\eta_i(\hat{\mathbf{F}}), i \in \{1,\ldots,k\}$, must be estimated by running simulations. Suppose we denote the corresponding estimate of (1) by $\hat{i}$. A natural question is, if $\hat{i}$ and i^c are the same? In other words, we are interested in the probability of correct selection,

$$\text{PCS} \triangleq \Pr\{\hat{i} = i^c\},$$

where the probability is taken with respect to both the simulation error in estimating $\eta_i(\hat{\mathbf{F}})$ and the sampling distribution of $\hat{\mathbf{F}}$. Since the streaming data size increases as the periods continue, $\hat{\mathbf{F}}$ converges to $\mathbf{F}^c$ and we expect that the PCS converges to one provided that the simulation error in estimating η_i is reduced appropriately. Thus, the problem boils down to deciding when to stop collecting the streaming input data and return $\hat{i}$ with a PCS guarantee.

However, providing the exact PCS guarantee is difficult without making some assumption about the separability between the performance measures at i^c and the rest of the systems even for a classical R&S without input uncertainty. Instead, we consider a modified statistical guarantee, the PGS, given user-specified threshold, $\delta > 0$:

$$\text{PGS} \triangleq \Pr\{\eta_{i^c}(\mathbf{F}^c) - \eta_{\hat{i}}(\mathbf{F}^c) \leq \delta\}.$$

Namely, PGS is the probability that the estimated best solution's performance measure is within δ from the true best's, and refer to these solutions as *good solutions*. In the next section, we introduce an R&S procedure that returns $\hat{i}$ when the estimated PGS exceeds $1-\alpha$ for some user-chosen $\alpha > 0$.

3 EXTENSION OF THE GENERAL-PURPOSE R&S

In this section, we present an R&S procedure that estimates i^c with a PGS guarantee. To design an efficient procedure, we adopt a metamodel to predict η_i at arbitrary input model $\mathbf{F}$ without having to run simulations at $\mathbf{F}$. Exploiting the metamodel, we extend the general-purpose R&S procedure by Lee and Nelson (2016) to account for both input uncertainty and stochastic error in metamodeling. We introduce the metamodel in Section 3.1 and the R&S procedure in Section 3.2.

3.1 Metamodeling

To address the effect of input uncertainty in R&S, we need to explicitly model the dependence of $\eta_i(\hat{\mathbf{F}})$ on $\hat{\mathbf{F}}$ and estimate its effect on determining i^c. While more sophisticated models are available, we adopt a simple linear regression in this work.

For arbitrary input model $\mathbf{F}$ and each $i \in \{1,\ldots,k\}$, we model the performance measure, $\eta_i(\mathbf{F})$, as a function of the moments of $\mathbf{F}$, e.g., means, variances, pairwise covariances of L input models, etc. We denote the moment vector by $\boldsymbol{\theta} \triangleq (\theta_1,\ldots,\theta_D)^\top$, where θ_1 is set to one, and impose a linear model in $\boldsymbol{\theta}$, $\psi_i(\boldsymbol{\theta}) \triangleq \boldsymbol{\theta}^\top \boldsymbol{\beta}^i$. Here, $\boldsymbol{\beta}^i \triangleq (\beta_1^i,\ldots,\beta_D^i)^\top$ is the vector of regression coefficients and β_1^i is the intercept term. Song and Nelson (2015) adopt a similar model in the context of input uncertainty quantification.

The least squares method lets us estimate the value of $\boldsymbol{\beta}^i$ by fitting the model with simulation outputs run at $B \geq D+1$ input models. To create the design matrix to fit the metamodel, we adopt bootstrap to generate B sets of input models. Let $\mathbf{X}_p^* \triangleq \{X_{p1}^*,\ldots,X_{pm_p}^*\}$ denote a size-m_p bootstrap sample drawn from $\hat{F}_p$ for each $p \in \{1,\ldots,L\}$ and $\hat{\mathbf{F}}^* \triangleq \{\hat{F}_1^*,\ldots,\hat{F}_L^*\}$ be the set of input distributions constructed from the bootstrap samples, $\{\mathbf{X}_1^*,\ldots,\mathbf{X}_L^*\}$. Here, m_p may or may not equal n_p, which will be specified later in the algorithms.

Suppose we bootstrap $\hat{\mathbf{F}}$ B times to generate $\hat{\mathbf{F}}^{*(b)}, b \in \{1,\ldots,B\}$, run $R_0 \geq 1$ replications at each b, and compute the estimate of $\eta_i(\hat{\mathbf{F}}^{*(b)})$, $\hat{\eta}_i(\hat{\mathbf{F}}^{*(b)})$, from the R_0 replications. For instance, if η_i is the mean

function, then $\hat{\eta}_i(\hat{\mathbf{F}}^{*(b)})$ is the sample average of the R_0 replications. Moreover, we assume all simulations are run independently across k systems, i.e., no common random numbers. We define Θ to be a $B \times D$ matrix whose bth row corresponds to the vector of moments of $\hat{\mathbf{F}}^{*(b)}$, $\hat{\boldsymbol{\theta}}^{*(b)}$, and $\mathbf{y}_i$ be a B-dimensional vector whose bth entry is $\hat{\eta}_i(\hat{\mathbf{F}}^{*(b)})$. Then, the least squares estimator of $\boldsymbol{\beta}^i$ and the fitted linear model for arbitrary $\boldsymbol{\theta}$ are $\hat{\boldsymbol{\beta}}^i \triangleq (\Theta^\top\Theta)^{-1}\Theta^\top\mathbf{y}_i$ and $\hat{\psi}_i(\boldsymbol{\theta}) \triangleq \boldsymbol{\theta}^\top\hat{\boldsymbol{\beta}}^i$.

We can further characterize the prediction error of $\hat{\psi}_i(\boldsymbol{\theta})$ under some simplifying assumptions. Suppose that the estimation error of $\hat{\eta}_i(\hat{\mathbf{F}}^{*(b)})$ is i.i.d. $\mathrm{N}(0,\sigma_i^2)$ for all $b \in \{1,\ldots,B\}$. Let $\mathbf{e}_i$ be the vector of estimation errors, i.e., $\mathbf{e}_i \triangleq \left(\hat{\eta}_i(\hat{\mathbf{F}}^{*(1)}) - \eta_i(\hat{\mathbf{F}}^{*(1)}),\ldots,\hat{\eta}_i(\hat{\mathbf{F}}^{*(B)}) - \eta_i(\hat{\mathbf{F}}^{*(B)})\right)^\top$. Then, given Θ,

$$\hat{\psi}_i(\boldsymbol{\theta}) = \boldsymbol{\theta}^\top\left(\boldsymbol{\beta}^i + (\Theta^\top\Theta)^{-1}\Theta^\top\mathbf{e}_i\right) \sim \mathrm{N}\left(\boldsymbol{\theta}^\top\boldsymbol{\beta}^i, \sigma_i^2\boldsymbol{\theta}^\top(\Theta^\top\Theta)^{-1}\boldsymbol{\theta}\right). \tag{2}$$

Clearly, the normality assumption we make here is unlikely to hold in general, but (2) provides an easy way to quantify the prediction error, which proves to be useful in designing the R&S procedure in Section 3.2 robust to the prediction error of the metamodel.

The algorithms we propose in section 3.2 and section 4 require the predictions from the metamodel to evaluate the stopping criteria at each period. While it is possible to fit the metamodel only once at the beginning of the algorithm and continue to use it throughout the algorithm, this may bring up two issues. The first is that we represent $\eta_i(\mathbf{F})$ as a linear function of the moments of $\mathbf{F}$, which may approximate η_i well locally, but may not have global fidelity. Our goal is to find i^c. Therefore, metamodels that lead to ranking i^c as the best near $\mathbf{F}^c$ may be good enough since $\hat{\mathbf{F}}$ becomes closer to $\mathbf{F}^c$ as n_p increases for all $p \in \{1,\ldots,L\}$. However, if we do not update the metamodels, then there is no guarantee that $\hat{\psi}_i(\cdot)$ would approximate $\eta_i(\cdot)$ well near $\mathbf{F}^c$ since all B design points of the regression are computed from the bootstrap samples generated from $\hat{\mathbf{F}}$. Secondly, even if $\eta_i(\cdot) = \psi_i(\cdot)$, i.e., the performance measure is indeed a linear function of $\boldsymbol{\theta}$, the fitted metamodel parameters, $\hat{\boldsymbol{\beta}}^i$, are subject to the estimation error as long as B is finite, which translates to the prediction error in $\hat{\psi}_i(\cdot)$.

The first issue may be addressed by refitting the metamodel in each period so that the design points can be generated from the most up-to-date $\hat{\mathbf{F}}$. On the other hand, the second issue may be addressed by increasing the number of design points to fit the metamodels, which favors including the simulation outputs from the previous periods. To address both problems, we introduce parameter $\gamma \in (0,1]$ to control the fraction of periods whose bootstrapped design points and simulation outputs that are included in the regression. At the tth data collection period, we include the last $\lceil \gamma t \rceil$ periods' design points and performance estimators to refit the metamodel. The ceiling function ensures that we always get an integer number and at least one period, particularly when t is low. A similar idea has been adopted in Wu et al. (2024). The bias-variance trade-off occurs when choosing γ, which we will investigate further in our future work.

3.2 R&S Procedure

In this section, we introduce Algorithm 1 that utilizes the metamodel in Section 3.1 to estimate i^c with a PGS guarantee given $\delta > 0$. We start by providing a brief background on the general-purpose R&S in Lee and Nelson (2016) for classical R&S problems with known $\mathbf{F}^c$ and discuss the modification we make to reflect input uncertainty.

Let $\hat{\eta}_i(\mathbf{F}^c)$ be an arbitrary estimator of $\eta_i(\mathbf{F}^c)$. Hsu (1996) shows that, if we have

$$\Pr\left\{\hat{\eta}_i(\mathbf{F}^c) - \hat{\eta}_j(\mathbf{F}^c) - \left(\eta_i(\mathbf{F}^c) - \eta_j(\mathbf{F}^c)\right) \le \delta,\ \forall i \ne j\right\} \ge 1-\alpha, \tag{3}$$

then, for each $i \in \{1,\ldots,k\}$,

$$\Pr\left\{\eta_i(\mathbf{F}^c) - \max_{j\ne i}\eta_j(\mathbf{F}^c) \in \left[\hat{\eta}_i(\mathbf{F}^c) - \max_{j\ne i}\hat{\eta}_j(\mathbf{F}^c) - \delta,\ \hat{\eta}_i(\mathbf{F}^c) - \max_{j\ne i}\hat{\eta}_j(\mathbf{F}^c) + \delta\right]\right\} \ge 1-\alpha. \tag{4}$$

Namely, if we have $\pm\delta$ CIs for all pairwise differences $\eta_i(\mathbf{F}^c) - \eta_j(\mathbf{F}^c), i \neq j$, with simultaneous coverage probability $1-\alpha$, then for each $i \in \{1,\ldots,k\}$ we can build a $\pm\delta$ CI of the performance measure difference from the best of the rest, $\eta_i(\mathbf{F}^c) - \max_{j\neq i} \eta_j(\mathbf{F}^c)$. Suppose we choose $\hat{i} = \arg\max_{i\in\{1,\ldots,k\}} \hat{\eta}_i(\mathbf{F}^c)$. Then, from (4), with probability $\geq 1-\alpha$, we have

$$\eta_{\hat{i}}(\mathbf{F}^c) - \max_{j\neq\hat{i}} \eta_j(\mathbf{F}^c) \geq \hat{\eta}_{\hat{i}}(\mathbf{F}^c) - \max_{j\neq\hat{i}} \hat{\eta}_j(\mathbf{F}^c) - \delta \geq -\delta,$$

since $\hat{\eta}_{\hat{i}}(\mathbf{F}^c) - \max_{j\neq\hat{i}} \hat{\eta}_j(\mathbf{F}^c) \geq 0$. This result implies that $\hat{i}$ is the best or within δ from the best with probability $1-\alpha$, i.e., the PGS is guaranteed to be $\geq 1-\alpha$. Exploiting this idea, the general-purpose R&S procedure in Lee and Nelson (2016) keeps simulating all k solutions at $\mathbf{F}^c$ until a bootstrap version of (3) is confirmed.

In our problem, we cannot directly simulate with $\mathbf{F}^c$ as it is unknown. Instead, we take $\hat{\psi}_i(\hat{\boldsymbol{\theta}})$, the prediction of $\eta_i(\hat{F})$ from the metamodel, as the estimate of $\eta_i(\mathbf{F}^c)$ since $\hat{\mathbf{F}}$ best approximates $\mathbf{F}^c$ given the current set of data. That is, the probability in (3) can be rewritten as

$$P \triangleq \Pr\Big\{ \hat{\psi}_i(\hat{\boldsymbol{\theta}}) - \hat{\psi}_j(\hat{\boldsymbol{\theta}}) - \big(\eta_i(\mathbf{F}^c) - \eta_j(\mathbf{F}^c)\big) \leq \delta,\ \forall i \neq j \Big\},$$

where the probability is taken with respect to the sampling distribution of $\hat{\mathbf{F}}$ as well as the prediction error of the metamodel. We design our Algorithm 1 to stop when P exceeds $1-\alpha$. Because the sampling distribution of $\hat{\mathbf{F}}$ is generally unknown, P cannot be directly evaluated. Instead, we adopt bootstrapping to approximate its value as in Lee and Nelson (2016).

The bootstrap estimator of P is defined as

$$P^* \triangleq \Pr\Big\{ \hat{\psi}_i(\hat{\boldsymbol{\theta}}^*) - \hat{\psi}_j(\hat{\boldsymbol{\theta}}^*) - \big(\hat{\psi}_i(\hat{\boldsymbol{\theta}}) - \hat{\psi}_j(\hat{\boldsymbol{\theta}})\big) \leq \delta,\ \forall i \neq j \,\Big|\, \hat{\mathbf{F}} \Big\}, \tag{5}$$

where the conditional probability is taken with respect to the bootstrap sampling distribution of $\hat{\mathbf{F}}^*$ and the prediction error of the metamodel given $\hat{\mathbf{F}}$. Suppose we adopt the same normality assumption as in (2). Then, (5) can be rewritten as

$$P^* = \Pr\Big\{ (\hat{\boldsymbol{\theta}}^* - \hat{\boldsymbol{\theta}})^\top (\boldsymbol{\beta}^i - \boldsymbol{\beta}^j) + (\hat{\boldsymbol{\theta}}^* - \hat{\boldsymbol{\theta}})^\top (\xi_i - \xi_j) \leq \delta,\ \forall i \neq j \,\Big|\, \hat{\mathbf{F}} \Big\}, \tag{6}$$

where $\xi_i \sim \mathrm{N}\left(0, \sigma_i^2 (\Theta^\top\Theta)^{-1}\right)$ for $i \in \{1,\ldots,k\}$, which characterizes the prediction error in the metamodel. The exact computation of P^* is difficult in general, but bootstrap sampling distribution of $\hat{\mathbf{F}}^*$ can be empirically approximated while ξ_i may be sampled from $\mathrm{N}\left(0, \sigma_i^2 (\Theta^\top\Theta)^{-1}\right)$ given σ_i^2. Although σ_i^2 is unknown, it can be estimated from the regression model. For each $i \in \{1,\ldots,k\}$, let $\mathbf{r}_i \triangleq \mathbf{y}_i - \Theta\hat{\boldsymbol{\beta}}_i = (r_{i1},\ldots,r_{iB})^\top$ be the vector of residuals of the regression. Then, $\hat{\sigma}_i^2 = \frac{1}{B-D}\sum_{b=1}^{B} r_{ib}^2$ can be used as a point estimate of σ_i^2.

Algorithm 1 starts by initializing the parameters including $\alpha, \delta, \gamma, B$, and R_0 and setting the values of n_p^0, the initial sample size from the pth input process, and Δn_p for all $p \in \{1,\ldots,L\}$. In Steps 4-14, Algorithm 1 updates the bootstrapped input distributions, $\hat{\mathbf{F}}^{*(1)},\ldots,\hat{\mathbf{F}}^{*(B)}$, runs simulations with them to obtain new performance measure estimates and updates the metamodel. In Step 15, P^* is estimated by

$$\hat{P}^* \triangleq \frac{1}{B}\sum_{b=1}^{B} \mathbf{1}\Big\{ (\hat{\boldsymbol{\theta}}^{*(b)} - \hat{\boldsymbol{\theta}})^\top (\hat{\boldsymbol{\beta}}^i - \hat{\boldsymbol{\beta}}^j) + (\hat{\boldsymbol{\theta}}^{*(b)} - \hat{\boldsymbol{\theta}})^\top (\xi_i^{(b)} - \xi_j^{(b)}) \leq \delta,\ \forall i \neq j \Big\}, \tag{7}$$

where $\xi_i^{(b)}$ is sampled from $\mathrm{N}\left(0, \hat{\sigma}_i^2 (\Theta^\top\Theta)^{-1}\right)$ for each $b \in \{1,\ldots,B\}$. We stop the procedure if $\hat{P}^* \geq 1-\alpha$. Otherwise, it proceeds to the next period to collect more input data and to reduce the prediction error of the metamodels.

Let T_1 denote the stopping period of Algorithm 1. Then, it runs $R_0 B T_1 k$ simulation replications in total. In addition, for each system we run one regression in each iteration to update the metamodel, making a total of $T_1 k$ linear regressions.

Algorithm 1 General Purpose R&S with Input Uncertainty

1: Choose $0 < \alpha < 1 - 1/k, \delta > 0, 0 < \gamma \leq 1, B \geq D+1$, and $R_0 \geq 1$. Set the values of $n_p^0 \geq 1$, and $\Delta n_p \geq 1$ for all $p \in \{1,\ldots,L\}$. Set $t = 1$.
2: For each $p \in \{1,\ldots,L\}$, let $\mathbf{X}_p$ to be the vector of n_p^0 observations and $n_p = n_p^0$.
3: Update $\hat{\mathbf{F}}$ and $\hat{\boldsymbol{\theta}}$ from $\{\mathbf{X}_1,\ldots,\mathbf{X}_L\}$.
4: **for** $b = 1$ to B **do**
5: For each $p \in \{1,\ldots,L\}$, draw size-n_p bootstrap sample $\mathbf{X}_p^{*(b)}$ from $\mathbf{X}_p$.
6: Update $\hat{\mathbf{F}}^{*(b)}$ and $\hat{\boldsymbol{\theta}}^{*(b)}$ from $\{\mathbf{X}_1^{*(b)},\ldots,\mathbf{X}_L^{*(b)}\}$.
7: Run R_0 replications to obtain $\hat{\eta}_i(\hat{\mathbf{F}}^{*(b)})$ for each $i \in \{1,\ldots,k\}$.
8: **end for**
9: Set $\Theta_t = (\hat{\boldsymbol{\theta}}^{*(1)} \ldots \hat{\boldsymbol{\theta}}^{*(B)})^\top$ and update Θ to include the bootstrapped design points from the last $\lceil \gamma t \rceil$ iterations, i.e., $\Theta = (\Theta_{t-\lceil \gamma t \rceil+1}^\top \ldots \Theta_t^\top)^\top$.
10: **for** $i = 1$ to k **do**
11: Set $\mathbf{y}_{it} = (\hat{\eta}_i(\hat{\mathbf{F}}_t^{*(1)}) \ldots \hat{\eta}_i(\hat{\mathbf{F}}_t^{*(B)}))^\top$ and update $\mathbf{y}_i = (\mathbf{y}_{i,t-\lceil \gamma t \rceil+1}^\top \ldots \mathbf{y}_{i,t}^\top)^\top$.
12: Compute $\hat{\boldsymbol{\beta}}_i = (\Theta^\top\Theta)^{-1}\Theta^\top\mathbf{y}_i$ and $\hat{\sigma}_i^2 = \frac{1}{B-D}\sum_{b=1}^B r_{ib}^2$ from $\mathbf{r}_i = \mathbf{y}_i - \Theta\hat{\boldsymbol{\beta}}_i$.
13: For each $b \in \{1\ldots,B\}$, sample $\xi_i^{(b)} \sim \mathrm{N}\left(0, \hat{\sigma}_i^2(\Theta^\top\Theta)^{-1}\right)$.
14: **end for**
15: Compute $\hat{P}^*$ in (7).
16: **if** $\hat{P}^* \geq 1-\alpha$ **then**
17: Return $\hat{i} = \arg\max_{i\in\{1,\ldots,k\}} \hat{\psi}_i(\hat{\boldsymbol{\theta}})$ as an estimate of i^c.
18: **else**
19: Obtain Δn_p additional input data and update $\mathbf{X}_p$ for all $p \in \{1,\ldots,L\}$; set $t = t+1$, and $n_p = n_p + \Delta n_p$; and go to Step 3.
20: **end if**

4 IMPROVING THE SAMPLE-SIZE EFFICIENCY

Recall that Algorithm 1 provides the PGS guarantee based on the simultaneous CIs for all pairwise comparisons of the systems. However, estimating the performance differences precisely for all $\binom{k}{2}$ pairs is considerably more demanding than the good selection event because even for a system whose performance is significantly worse than the best, Algorithm 1 still requires its difference from all other solutions to be estimated at the same level of precision. This indeed makes Algorithm 1 conservative as can be seen from the empirical results in Section 5; the empirical PGS of the algorithm tends to be much higher than the target, $1-\alpha$, implying that the procedure could have stopped earlier. This observation has motivated Algorithm 2 introduced in this section, which improves the efficiency of Algorithm 1 by relaxing the pairwise comparison requirement; this is inspired by Lee and Nelson's unpublished follow-up research to improve the computational efficiency of the general-purpose R&S procedure who generously shared it with the authors.

We start by establishing a less conservative probability bound to guarantee the PGS. Nelson and Banerjee (2001) show that, if

$$\Pr\left\{\hat{\eta}_i(\mathbf{F}^c) - \hat{\eta}_{i^c}(\mathbf{F}^c) - \left(\eta_i(\mathbf{F}^c) - \eta_{i^c}(\mathbf{F}^c)\right) \leq \max\{\delta, \eta_{i^c}(\mathbf{F}^c) - \eta_i(\mathbf{F}^c)\},\ \forall i \neq i^c\right\} \geq 1-\alpha, \tag{8}$$

then we have $\Pr\left\{\eta_{i^c}(\mathbf{F}^c) - \eta_{\hat{i}}(\mathbf{F}^c) \leq \delta\right\} \geq 1-\alpha$. Hence, if we find estimators $\hat{\eta}_1(\mathbf{F}^c),\ldots,\hat{\eta}_k(\mathbf{F}^c)$ that satisfy (8), then the PGS is guaranteed with probability at least $1-\alpha$.

There are two benefits to this approach that enhance the efficiency of the algorithm. First, the number of comparisons is reduced from $k(k-1)$ in (3) to $k-1$ in (8) by focusing on the comparisons between each solution with i^c. Second, the comparisons between i^c and considerably inferior systems are allowed

Algorithm 2 Efficient General Purpose R&S with Input Uncertainty

1: Choose $0 < \alpha < 1 - 1/k, \delta > 0, 0 < \gamma \leq 1, 0 < \varepsilon < 1, B \geq D+1$, and $R_0 \geq 1$. Set the values of $n_p^0 \geq 1$, and $\Delta n_p \geq 1$ for all $p \in \{1,\ldots,L\}$. Set $t = 1$.
2: For each $p \in \{1,\ldots,L\}$, let $\mathbf{X}_p$ to be the vector of n_p^0 observations and $n_p = n_p^0$.
3: Update $\hat{\mathbf{F}}$ and $\hat{\boldsymbol{\theta}}$ from $\{\mathbf{X}_1,\ldots,\mathbf{X}_L\}$.
4: For each $p \in \{1,\ldots,L\}$, let $m_p = \lceil n_p^{1-\varepsilon} \rceil$.
5: **for** $b = 1$ to B **do**
6: For each $p \in \{1,\ldots,L\}$, draw size-m_p bootstrap sample $\mathbf{X}_p^{*(b)}$ from $\mathbf{X}_p$.
7: Update $\hat{\mathbf{F}}^{*(b)}$ and $\hat{\boldsymbol{\theta}}^{*(b)}$ from $\{\mathbf{X}_1^{*(b)},\ldots,\mathbf{X}_L^{*(b)}\}$.
8: Run R_0 replications to obtain $\hat{\eta}_i(\hat{\mathbf{F}}^{*(b)})$ for each $i \in \{1,\ldots,k\}$.
9: **end for**
10: Set $\Theta_t = (\hat{\boldsymbol{\theta}}^{*(1)} \ldots \hat{\boldsymbol{\theta}}^{*(B)})^\top$ and update Θ to include the bootstrapped design points from the last $\lceil \gamma t \rceil$ iterations, i.e., $\Theta = (\Theta_{t-\lceil \gamma t \rceil+1}^\top \ldots \Theta_t^\top)^\top$.
11: **for** $i = 1$ to k **do**
12: Set $\mathbf{y}_{it} = (\hat{\eta}_i(\hat{\mathbf{F}}_t^{*(1)}) \ldots \hat{\eta}_i(\hat{\mathbf{F}}_t^{*(B)}))^\top$ and update $\mathbf{y}_i = (\mathbf{y}_{i,t-\lceil \gamma t \rceil+1}^\top \ldots \mathbf{y}_{i,t}^\top)^\top$.
13: Compute $\hat{\boldsymbol{\beta}}_i = (\Theta^\top\Theta)^{-1}\Theta^\top \mathbf{y}_i$ and $\hat{\sigma}_i^2 = \frac{1}{B-D}\sum_{b=1}^B r_{ib}^2$ from $\mathbf{r}_i = \mathbf{y}_i - \Theta\hat{\boldsymbol{\beta}}_i$.
14: For each $b \in \{1\ldots,B\}$, sample $\xi_i^{(b)} \sim \mathrm{N}\left(0, \hat{\sigma}_i^2(\Theta^\top\Theta)^{-1}\right)$.
15: **end for**
16: Set $\hat{i} = \arg\max_{i\in\{1,\ldots,k\}} \hat{\psi}_i(\hat{\boldsymbol{\theta}})$.
17: Compute $\hat{P}'^*$ in (11).
18: **if** $\hat{P}'^* \geq 1-\alpha$ **then**
19: Return $\hat{i}$ as an estimate of i^c.
20: **else**
21: Obtain Δn_p additional input data and update $\mathbf{X}_p$ for all $p \in \{1,\ldots,L\}$; set $t = t+1$ and $n_p = n_p + \Delta n_p$; and go to Step 3.
22: **end if**

to be less precise. For the good systems, (8) prescribes the CI upper bound to be δ. For the others, it allows greater upper bounds.

Using the same metamodels defined in Section 3.1, the probability in (8) can be rewritten as

$$P' \triangleq \Pr\left\{\hat{\psi}_i(\hat{\boldsymbol{\theta}}) - \hat{\psi}_{i^c}(\hat{\boldsymbol{\theta}}) - \left(\eta_i(\mathbf{F}^c) - \eta_{i^c}(\mathbf{F}^c)\right) \leq \max\{\delta, \eta_{i^c}(\mathbf{F}^c) - \eta_i(\mathbf{F}^c)\},\ \forall i \neq i^c\right\}, \tag{9}$$

where the probability is taken with respect to the sampling distribution of $\hat{\mathbf{F}}$ as well as the prediction error of the metamodels. We design Algorithm 2 to stop when $P' \geq 1-\alpha$.

Since P' cannot be directly evaluated, we utilize its bootstrap estimator, P'^*. Notice that (9) requires the identity of i^c; the equivalent quantity under the bootstrap sampling distribution given $\hat{F}$ is $\arg\max_{i\in\{1,\ldots,k\}} \psi_i(\hat{\boldsymbol{\theta}})$, which we approximate with $\hat{i} = \arg\max_{i\in\{1,\ldots,k\}} \hat{\psi}_i(\hat{\boldsymbol{\theta}})$ as an estimate of i^c. Adopting the same normality assumption as in (2), we define

$$\begin{aligned} P'^* &\triangleq \Pr\left\{\hat{\psi}_i(\hat{\boldsymbol{\theta}}^*) - \hat{\psi}_{\hat{i}}(\hat{\boldsymbol{\theta}}^*) - \left(\hat{\psi}_i(\hat{\boldsymbol{\theta}}) - \hat{\psi}_{\hat{i}}(\hat{\boldsymbol{\theta}})\right) \leq \max\{\delta, \hat{\psi}_{\hat{i}}(\hat{\boldsymbol{\theta}}) - \hat{\psi}_i(\hat{\boldsymbol{\theta}})\},\ \forall i \neq \hat{i} \,\middle|\, \hat{\mathbf{F}}\right\} \\ &= \Pr\left\{(\hat{\boldsymbol{\theta}}^* - \hat{\boldsymbol{\theta}})^\top(\boldsymbol{\beta}^i - \boldsymbol{\beta}^{\hat{i}}) + (\hat{\boldsymbol{\theta}}^* - \hat{\boldsymbol{\theta}})^\top(\xi_i - \xi_{\hat{i}}) \leq \max\{\delta, \hat{\boldsymbol{\theta}}^\top(\boldsymbol{\beta}^{\hat{i}} - \boldsymbol{\beta}^i) + \hat{\boldsymbol{\theta}}^\top(\xi_{\hat{i}} - \xi_i)\},\ \forall i \neq \hat{i} \,\middle|\, \hat{\mathbf{F}}\right\}, \end{aligned} \tag{10}$$

where the conditional probability is taken with respect to the bootstrap sampling distribution of $\hat{\mathbf{F}}^*$ and the prediction error of the metamodel given $\hat{\mathbf{F}}$.

Recall that in (6), the right-hand side threshold of the probability statement is δ, which is fixed and known. On the other hand, the threshold in (10) is $\max\{\delta, \hat{\psi}_{\hat{i}}(\hat{\boldsymbol{\theta}}) - \hat{\psi}_i(\hat{\boldsymbol{\theta}})\}$, which contains estimates.

Although we do not present here, some asymptotic analyses have revealed that to enjoy consistency of P'^* in an asymptotic regime, $\psi_{\hat{i}}(\hat{\boldsymbol{\theta}}) - \psi_i(\hat{\boldsymbol{\theta}}), \forall i \neq \hat{i}$, must converge faster to $\psi_{\hat{i}}(\boldsymbol{\theta}^c) - \psi_i(\boldsymbol{\theta}^c)$ than the rate at which the bootstrap distribution of $\psi_i(\hat{\boldsymbol{\theta}}^*) - \psi_{\hat{i}}(\hat{\boldsymbol{\theta}}^*)$ converges to $\psi_i(\hat{\boldsymbol{\theta}}) - \psi_{\hat{i}}(\hat{\boldsymbol{\theta}})$ given $\hat{\mathbf{F}}$. Motivated by this preliminary analysis, we choose $m_p \neq n_p$ in Algorithm 2; we impose m_p to grow at a slower rate than n_p by setting $m_p = \lceil n_p^{1-\varepsilon} \rceil$, for some $\varepsilon \in (0,1)$. Hence, Algorithm 2 constructs $\hat{\mathbf{F}}^{*(1)}, \ldots, \hat{\mathbf{F}}^{*(B)}$ by sampling m_p observations from $\hat{F}_p$ for each $p \in \{1, \ldots, L\}$.

Algorithm 2 is similar to Algorithm 1; we only highlight the differences below. Algorithm 2 requires the value of ε in Step 1 as described above. In Step 17, we estimate P'^* with

$$\hat{P}'^* \triangleq \frac{1}{B}\sum_{b=1}^{B} \mathbf{1}\Big\{(\hat{\boldsymbol{\theta}}^{*(b)} - \hat{\boldsymbol{\theta}})^\top (\hat{\boldsymbol{\beta}}^i - \hat{\boldsymbol{\beta}}^{\hat{i}}) + (\hat{\boldsymbol{\theta}}^{*(b)} - \hat{\boldsymbol{\theta}})^\top (\xi_i^{(b)} - \xi_{\hat{i}}^{(b)}) \leq \max\{\delta, \hat{\boldsymbol{\theta}}^\top (\hat{\boldsymbol{\beta}}^{\hat{i}} - \hat{\boldsymbol{\beta}}^i) + \hat{\boldsymbol{\theta}}^\top (\xi_{\hat{i}}^{(b)} - \xi_i^{(b)})\}, \forall i \neq \hat{i}\Big\}. \tag{11}$$

Let T_2 denote the stopping period of Algorithm 2. Algorithm 2 runs a total of $R_0 B T_2 k$ simulation replications and $T_2 k$ linear regressions. These are identical to Algorithm 1 if we replace T_2 with T_1, however, we expect $T_2 < T_1$ as can be confirmed by the experiments in Section 5.

5 EMPIRICAL STUDY

In this section, we evaluate the proposed algorithms with some numerical examples. In the first example, the performance measure is the simulation output mean and a linear function in $\boldsymbol{\theta}$, i.e., $\eta_i = \psi_i$. We consider a quadratic performance measure in $\boldsymbol{\theta}$ in the second example while ψ_i still assumes a linear model to test robustness. In the first two examples, we add homoscedastic simulation output noises to make them stochastic. The last example is a $M/M/1/c$ queueing system, where the goal is to find the system capacity, c, that minimizes the expectation of a cost function. All three examples have analytical expressions for the performance measures facilitating the empirical evaluations of the algorithms.

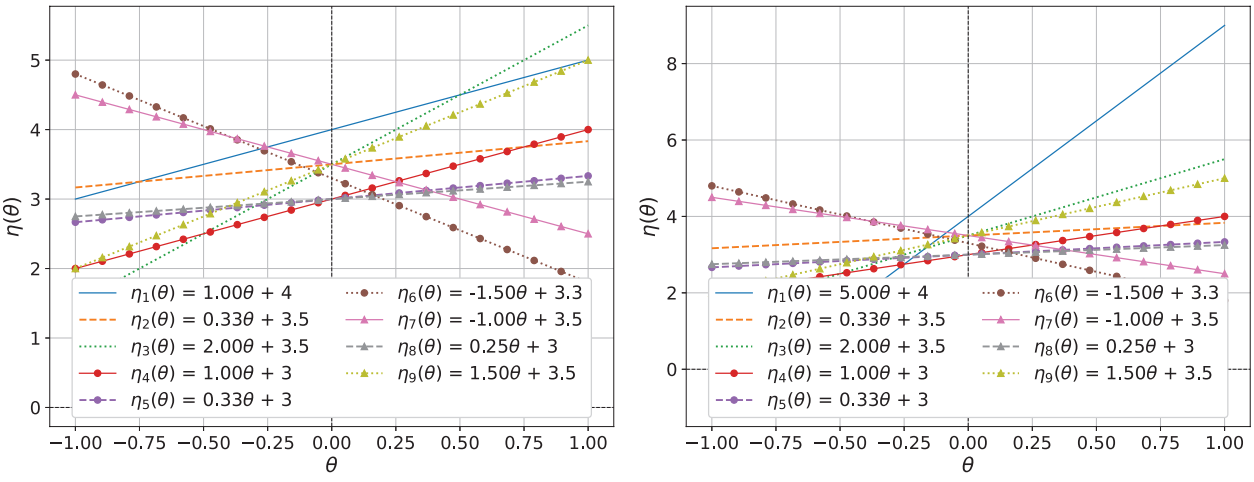
In all examples we consider $k = 9$ systems, the target $1-\alpha$ is set to 0.9, and the number of bootstrap iterations $B = 500$. For Algorithm 2, we set $\varepsilon = 0.1$. All experiments are run for 1000 macroreplications for performance evaluations of the algorithms; we consider four metrics. First, the coverage probability of the joint CIs that implies PGS is empirically evaluated. For Algorithm 1, this probability corresponds to (3); for Algorithm 2, it is (8). The second and third metrics are the empirical PCS and PGS, respectively. Both algorithms aim to provide the PGS $\geq 1-\alpha$, but do not guarantee the PCS. The last metric is the number of input data collected before stopping; the lower, the more efficient.

5.1 Linear Response

In this example. we choose $\mathbf{F}^c = \mathrm{N}(0,1)$ as our single uni-variate input model, which we pretend to be unknown. We adopt $\eta_i(\theta) = a_i + b_i\theta$ as the simulation output mean of System i where θ is the mean of $\mathbf{F}$.

The values of $(a_i, b_i), i \in \{1, \ldots, k\}$, we adopt are presented in Figure 1. We consider two parameter settings for the best system, $i^c = 1$. Figure 1 plots $\eta_i(\theta)$ of all 9 systems in $\theta \in [-1,1]$. In the first case plotted in Figure 1(a), $b_1 = 1$. In the second case in Figure 1(b), we increase b_1 to 5. This change makes System 1 more sensitive to the variation in θ near the true mean, which makes it more difficult to correctly find $i^c = 1$ in the second case. The simulation output is generated by adding a normal noise to η_i: $Y_i(\theta) = \eta_i(\theta) + \sigma Z$, where $Z \sim \mathrm{N}(0,1)$ and $\sigma = 20$.

The algorithmic parameters used for this problem are $n^0 = 50, \Delta n = 1, R_0 = 1$ and $\gamma = 0.8$. In addition, we set $\delta = 0.5$. In both test cases, this choice makes System 1 the only good system, i.e., PCS = PGS. For Algorithm 2, PCS and PGS are also equal to the coverage in this case. To observe this, note that $\max\{\delta, \eta_{i^c}(\mathbf{F}^c) - \eta_i(\mathbf{F}^c)\} = \eta_{i^c}(\mathbf{F}^c) - \eta_i(\mathbf{F}^c)$ for all $i \neq i^c$. Hence, we write the event in (8) as $\{\hat{\psi}_{i^c}(\hat{\boldsymbol{\theta}}) - \hat{\psi}_i(\hat{\boldsymbol{\theta}}) \geq 0, i \neq i^c\}$, which is true if and only if $\hat{i} = i^c$.



(a) (b)

Figure 1: Plots of $\eta_i(\theta) = a_i + b_i\theta, i \in \{1,\ldots,9\}$, for two test cases. In Case (a), $b_1 = 1$, whereas in Case (b), $b_1 = 5$. All other systems' performance measures are the same for both cases.

Table 1: Empirical coverage, PCS, PGS and input data size (at stopping) for the examples in Figure 1 computed from 1000 macroruns.

Algorithm	**Case (a)**				**Case (b)**			
	Coverage	PCS = PGS	Input Data Size Avg	Input Data Size Std Dev	Coverage	PCS = PGS	Input Data Size Avg	Input Data Size Std Dev
Algorithm 1	0.913	0.995	227	52	0.887	0.957	445	106
Algorithm 2	0.936	0.936	93	21	0.864	0.864	208	133

The left panel of Table 1 contains the empirical results for Case (a). Both algorithms have coverages greater than the target, 0.9. Algorithm 1 clearly overshoots the PGS (=PCS) compared to Algorithm 2. The average input data sizes are 227 and 93 for Algorithms 1 and 2, respectively. Clearly, Algorithm 2 is more efficient because it reaches the desired PGS earlier. The difference between both algorithms indicates the conservatism in Algorithm 1, since it mandates precise pairwise comparisons of all systems, which requires a higher number of observations to reach the coverage target.

The right side of Table 1 contains the empirical results for Case (b). Both algorithms have coverages lower than the target, 0.9. Case (b) shows to be more difficult for both algorithms, since the coverage and the PGS are lower compared to Case (a). The PGS is still greater than the target for Algorithm 1 but it is below for Algorithm 2. This difference is produced because the conservatism in Algorithm 1 is still present; the average input data sizes are 445 and 208 for Algorithms 1 and 2, respectively.

5.2 Quadratic Response

In this example, we consider $\eta_i(\theta) = a_i(\theta - b_i)^2 + c_i$ as the simulation output mean of System i. As in the previous example, we choose $\mathbf{F}^c = \mathrm{N}(0,1)$, which we pretend to be unknown. Figure 2 plots $\eta_i(\theta)$ of all 9 systems in $\theta \in [-1,1]$. Observe that System 1 performs the best when $\theta \in [-0.115, 0.1]$. We consider the simulation output $Y_i(\theta) = \eta_i(\theta) + \sigma Z$, where $Z \sim \mathrm{N}(0,1)$ and $\sigma = 4$. The algorithmic parameters chosen

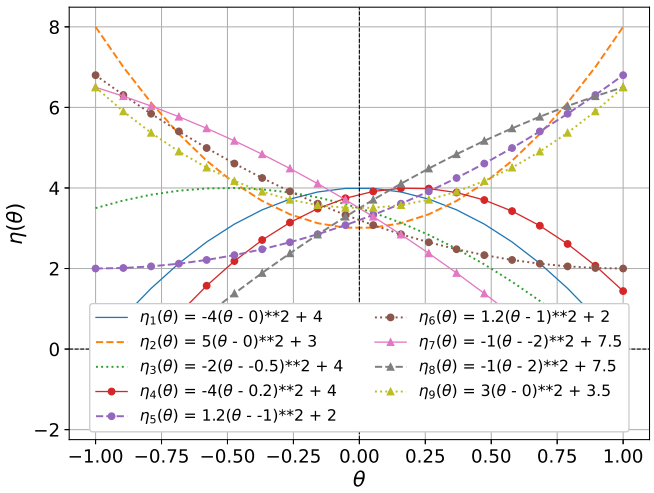


Figure 2: Plots of $\eta_i(\theta) = a_i(\theta - b_i)^2 + c_i, i \in \{1, \ldots, 9\}$.

Table 2: Empirical coverage, PCS, PGS and input data size (at stopping) for the examples in Figure 2 . Results are computed from 1000 macroruns.

Algorithm	Coverage	PCS	PGS	Input Data Size	
				Avg	Std Dev
Algorithm 1	0.893	0.983	1.000	652	113
Algorithm 2	0.907	0.823	0.935	248	63

for this example are $n^0 = 100, \Delta n = 10, R_0 = 1$ and $\gamma = 0.8$. In addition, we set $\delta = 0.5$, which makes System 4 a good system since $\eta_1(0) - \eta_4(0) < \delta$. Therefore, PGS $\neq$ PCS in this example.

Table 2 contains the empirical results. Both algorithms have coverages close to the target: 0.893 for Algorithm 1 and 0.907 for Algorithm 2. The PCS is lower than the target, 0.9, for Algorithm 2. However, the goal of both algorithms is to provide the PGS, which is achieved by both. The PGS is overshot by Algorithm 1 compared to Algorithm 2. The average input data size are 652 and 248 for Algorithms 1 and 2, respectively, which suggests that Algorithm 1 is conservative for this example, too.

5.3 Queuing System Simulation

Consider now a $M/M/1/c$ queuing system with Poisson arrival process with rate 0.9 and only one server with exponentially distributed service times with mean 1.21, where the capacity of the system is c (one in the server and $c-1$ in the queue). We define a cost function for a customer as $C = 1 \cdot W + 15 \cdot Z + 0.01 \cdot c^{3/2}$, where W is the waiting time and Z is a variable that takes the value 1 if the customer could not enter the system because it was full and 0 otherwise. Lower values of c yield low values of W but a higher likelihood that Z is equal to 1. In contrast, higher values of c decrease the probability of finding the system full, but increase the average waiting time, since we can accept more customers.

The objective is to find the value of c that minimizes the expected cost of a customer in steady state, considering that W and Z are random quantities. We assume that $3 \leq c \leq 11$, giving a total of 9 systems

Table 3: Expected Cost for Different System Capacities in the $M/M/1/c$ queuing system.

System Capacity	3	4	5	6	7	8	9	10	11
Expected Cost	2.895	2.449	2.263	2.211	2.229	2.285	2.360	2.444	2.532

Table 4: Empirical coverage, PCS, PGS and input data size (at stopping) for the $M/M/1/c$ queueing system. Results are computed from 1000 macroruns.

Algorithm	Coverage	PCS	PGS	Input Data Size	
				Avg	Std Dev
Algorithm 1	0.886	0.957	1.000	3146	1100
Algorithm 2	0.894	0.751	0.999	243	100

to evaluate. Since this problem has been studied exactly in the literature, we know the exact expression of the expected cost. Table 3 contains the expected cost for each system, making $c = 6$ the optimal solution.

We pretend that the arrival rate is unknown to us and must be estimated from observations. In this example, $\mathbf{F}^c$ represents the distribution of the interarrival times, which is exponential with rate 0.9. We assume that we know the distribution family of $\mathbf{F}^c$. Therefore, we only need to estimate the parameter of the arrival process. The algorithmic parameters used for this problem are $n^0 = 50, \Delta n = 5, R_0 = 2$, and $\gamma = 0.5$. In addition, we set the indifference zone parameter $\delta = 0.06$, which allows c to be 5, 6 or 7 to be considered a good system.

Table 4 contains the results of both algorithms. Algorithms 1 and 2 show coverages of 0.886 and 0.894, respectively, suggesting that the bootstrap estimations closely approximate the target, 0.9. Algorithm 1 shows high empirical PCS and PGS, as well as the average input data size, confirming the conservatism of the procedure – in line with what was observed in the other examples. Note that the PCS and average input data size are considerably lower for Algorithm 2 than for Algorithm 1. Recall that neither algorithm provides a PCS guarantee and it is rather difficult to tell apart $c = 6$ from two other good systems as all three systems perform similarly. Therefore, we attribute the low PCS for Algorithm 2 to the difficulty of the problem. On the other hand, because Algorithm 1 is more conservative, it samples more observations, which explains the higher PCS. Finally, both algorithms overshoot the target PGS. Recall that Algorithm 2 still applies lower bounds to guarantee the PGS albeit less conservation than Algorithm 1.

6 CONCLUSION

In this article, we study the R&S problem with streaming input data. We propose two sequential algorithms that determine when to stop receiving the streaming input data and return the estimated optimum with the PGS guarantees. Both procedures adopt a metamodel to represent the effect of input uncertainty on the system performance measures and evaluate the probabilities of the events that imply PGS via bootstrap. The former relies on precise pairwise comparisons while the latter relaxes this requirement allowing the procedure to terminate much earlier.

Several questions remain to be studied in the future work. First, we will provide theoretical guarantees of the procedures under an asymptotic regime. Furthermore, some algorithmic parameters are not trivial to set up such as the parameter γ, that controls the number of design points to refit the metamodel with or the bootstrap sample size, $m_p, p \in \{1, \ldots, L\}$, in Algorithm 2. The choices for these parameters need to be investigated both empirically and theoretically in an asymptotic regime.

ACKNOWLEDGMENTS

This research has been funded by the National Science Foundation grants CMMI-2045400 and CMMI-2417616. The authors thank Barry Nelson for sharing his insights for the research.

REFERENCES

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.

He, L., and E. Song. 2024. "Introductory Tutorial: Simulation Optimization Under Input Uncertainty". In *2024 Winter Simulation Conference (WSC)*, 1338–1352 https://doi.org/10.1109/WSC63780.2024.10838862.

Hsu, J. C. 1996. *Multiple Comparisons: Theory and Methods*. 1st ed. London: Chapman & Hall/CRC.

Lee, S., and B. L. Nelson. 2016. "General-Purpose Ranking and Selection for Computer Simulation". *IIE transactions* 48(6):555–564.

Liu, T., Y. Lin, and E. Zhou. 2021. "A Bayesian Approach to Online Simulation Optimization with Streaming Input Data". In *2021 Winter Simulation Conference (WSC)*, 1–12 https://doi.org/https://doi.org/10.1109/WSC52266.2021.9715392.

Nelson, B. L., and S. Banerjee. 2001. "Selecting a Good System: Procedures and Inference". *IIE Transactions* 33(3):149–166.

Song, E., and B. L. Nelson. 2015. "Quickly Assessing Contributions to Input Uncertainty". *IIE Transactions* 47(9):893–909.

Wang, Y., and E. Zhou. 2025. "Optimal Computing Budget Allocation for Data-Driven Ranking and Selection". *INFORMS Journal on Optimization* 7(1):1–19.

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

AUTHOR BIOGRAPHIES

JAIME GONZALEZ-HODAR is a Ph.D. student in the H. Milton Stewart School of Industrial and Systems Engineering at Georgia Institute of Technology. His research interests are input uncertainty quantification and simulation optimization. He has experience with applied projects across various domains, including healthcare, supply chain, and mining. His e-mail address is jhodar3@gatech.edu. His website is https://sites.google.com/view/jaime-gonzalez-hodar/.

EUNHYE SONG is an Associate Professor in the H. Milton Stewart School of Industrial and Systems Engineering at Georgia Institute of Technology. Her research interests include simulation model validation/calibration, uncertainty and risk quantification, and simulation optimization. She has several past and current industry collaborations on manufacturing digital twins, energy sustainability, product portfolio optimization and more. Her email address is eunhye.song@isye.gatech.edu. Her website is http://eunhyesong.info.