Proceedings of the 2022 Winter Simulation Conference
B. Feng, G. Pedrielli, Y. Peng, S. Shashaani, E. Song, C.G. Corlu, L.H. Lee, E.P. Chew, T. Roeder, and P. Lendermann, eds.

A CLASSIFICATION METHOD FOR RANKING AND SELECTION WITH COVARIATES

Gregory Keslin
Barry L. Nelson
Matthew Plumlee

Department of Industrial Engineering & Management Sciences
Northwestern University
2145 Sheridan Road
Evanston, Illinois USA

Bernardo K. Pagnoncelli

SKEMA Business School
Université Côte d'Azur
Avenue Willy Brandt - 59777 EURALILLE
Lille FRANCE

Hamed Rahimian

Department of Industrial Engineering
Clemson University
211 Fernow Street
Clemson, South Carolina USA

ABSTRACT

Ranking & selection (R&S) procedures are simulation-optimization algorithms for making one-time decisions among a finite set of alternative system designs or feasible solutions with a statistical assurance of a good selection. R&S with covariates (R&S+C) extends the paradigm to allow the optimal selection to depend on contextual information that is obtained just prior to the need for a decision. The dominant approach for solving such problems is to employ offline simulation to create metamodels that predict the performance of each system or feasible solution as a function of the covariate. This paper introduces a fundamentally different approach that solves individual R&S problems offline for various values of the covariate, and then treats the real-time decision as a classification problem: given the covariate information, which system is a good solution? Our approach exploits the availability of efficient R&S procedures, requires milder assumptions than the metamodeling paradigm to provide strong guarantees, and can be more efficient.

1 INTRODUCTION

Simulation optimization (SO) is typically an offline exercise: Construct a simulation model of a real or conceptual system; optimize the expected value of one or more performance measures over controllable decision variables; and implement the resulting optimal decision in the real world. Recently there has been interest in settings where the real or conceptual system can be simulated, but the decision may be deferred until some real-world context or covariate information is available. Later in this paper we will consider an inventory purchasing problem for multiple products where product demand in one period is strongly correlated with demand in a previous period. By deferring the ordering decision until the previous period's demand is realized, better purchasing decisions are possible by refining the demand distribution with this "covariate" information.

When the number of feasible solutions or systems is finite and not too large, ranking & selection (R&S) procedures are the go-to SO tool. R&S procedures simulate all systems to estimate their performance and

select a system to implement with some statistical guarantee of goodness. In this paper we focus on the statistical guarantee of probability of good selection (PGS). PGS refers to the probability of the procedure yielding a system whose performance is within a user-specified optimality gap, δ, of the best. We will denote the desired or nominal PGS by $1-\alpha$.

An intuitive way to extend R&S procedures to problems with covariate information is to defer applying the procedure until the covariate is observed. We denote the covariate by a random vector $\boldsymbol{X} \in \mathscr{X}$ and a particular realized value of it by $\boldsymbol{x}_0$. Upon observing $\boldsymbol{X} = \boldsymbol{x}_0$, a R&S procedure could be applied to select a good system given the covariate. Thus, the procedure is effectively online since the simulations rely on the observed value of the covariate, $\boldsymbol{x}_0$. Of course, if the simulations are computationally expensive then there could be a significant delay between the observation of $\boldsymbol{X}$ and when a system is chosen by the procedure. This delay may be too long for certain problems, requiring a different approach that we refer to as R&S with covariates (R&S+C).

R&S+C is a generalization of R&S that accounts for the covariate by executing the simulations necessary to support a selection offline, prior to the observation of the covariate's value. Because the simulations are executed offline, the delay between the observation of $\boldsymbol{X} = \boldsymbol{x}_0$ and selecting a system does not depend on the computational expense of the simulations. Another advantage of running simulations offline is that the results can be reused. This advantage is relevant in problems when we repeatedly observe different values of the covariate and want to make custom selections each time. For example, in inventory purchasing one may need to select a product mix repeatedly, say for different periods or for different stores. Even though the observed value of $\boldsymbol{X}$, say the previous period's demand, may change each time a product mix needs to be selected, the simulation results can be reused because they were obtained offline.

The incumbent approach in the literature on R&S+C is to form metamodels that characterize each system's performance as a function of the covariate. Specifically, simulation output is used to construct a metamodel, $\widehat{\mu}_j(\boldsymbol{x})$, for $j = 1,2,\ldots,p$, where p denotes the number of distinct systems. Then given $\boldsymbol{X} = \boldsymbol{x}_0$, the selected system is $\operatorname{argmax}_j \widehat{\mu}_j(\boldsymbol{x}_0)$. Examples include Shen et al. (2021) and Hu and Ludkovski (2017). We refer to this approach as "weight then optimize" (WtO), since the metamodels can often be viewed as weighted averages of the simulation outputs and the optimization happens after the weighting.

In this paper, we propose an alternative to WtO that exploits established R&S procedures that provide a PGS guarantee. Consider choosing an experimental design of covariate values in $\mathscr{X}$, denoted by $\mathscr{D}_m = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m\}$, and then executing a R&S procedure at each covariate value in the design. This process creates a database that assigns to each covariate value in $\mathscr{D}_m$ a corresponding good system that was selected by the R&S procedure. The database can then be used to calibrate a *classifier*, where here "classification" means assigning a good system to any given covariate value. Various classification tools can be used, but we employ K-nearest neighbors (KNN, James et al. 2013, Chapter 4) due to its relative simplicity for this initial study. We refer to our approach of constructing a database through the use of R&S and then calibrating a classifier as "optimize then weight" (OtW), since the classification can often be viewed as a weighted average of the existing classes and the optimization happens before the weighting.

There are two potential advantages of OtW as an alternative to WtO. The first is the probabilistic characteristics of OtW. The statistical guarantees of WtO typically rely on specific and restrictive metamodeling choices and assumptions. One example is Shen et al. (2021), where the metamodel is assumed to be a linear function of the covariate. In contrast, OtW methods inherit their statistical guarantees from R&S procedures that have been refined through decades of study. A second potential advantage that OtW provides over WtO is a better paradigm for allocating replications to each candidate system. WtO metamodels are often built for predictive accuracy, which means suboptimal systems may still require many simulations to construct metamodels with a high degree of predictive accuracy. On the other hand, OtW with modern R&S procedures like KN (Kim and Nelson 2001) can lead to fewer simulations of systems that are clearly suboptimal.

Following a brief literature review in the following section, we introduce our OtW approach to R&S+C and prove a basic consistency result in Section 3. We then provide empirical comparisons relative to the WtO approach of Shen et al. (2021) in Section 4.

2 LITERATURE

Real-time decisions based on observed covariate information has recently received considerable academic attention. For example, Bertsimas and Kallus (2020) adopt a WtO approach to minimize the conditional expected cost by solving static, continuous optimization problems in the areas of operations research and operations management. Their approach constructs an approximation that employs various classes of nonparametric learning methods, including KNN, Classification and Regression Trees (CART), random forests, and kernels (James et al. 2013), to create metamodels. Using a set of historical data on the observed system performance and covariate information, each data point is weighted according to some measure of proximity to the current observed value of the covariate, and the optimization happens after the weighting. Another recent paper that adopts a WtO approach to responsive decisions is Kannan et al. (2022), where the authors focus on parametric learning metamodels in the same vein as those in Shen et al. (2021). In this larger view of decisions, Ban and Rudin (2019) propose several ways to handle covariates, including an empirical risk minimization approach that seems to fit under an OtW philosophy: Using a set of historical data on the observed performance measure and covariate information, they solve an optimization problem once to learn an optimal mapping from the covariate space, $\mathscr{X}$, to the decision space (i.e., a policy). Then when covariate information is observed, a decision for such a realization is readily available.

Although the literature on R&S+C is relatively young, there have been papers that deploy various choices of metamodels, and also others that opt not to use a metamodel at all. Gao et al. (2019) and Jin et al. (2019) assume that the covariate space, $\mathscr{X}$, is finite and small enough that every system and covariate combination can be simulated; therefore, no metamodel is required. They derive experiment designs to efficiently allocate a finite simulation budget to maximize a measure of probability of correct selection (PCS) across all combinations. Of course, when no metamodel is constructed, no information can be gained for covariate values that were not simulated, which will always be the case when $\mathscr{X}$ is infinite.

Many R&S+C methods apply Bayesian metamodels by leveraging their posterior distribution. A workhorse Bayesian metamodel is Gaussian process regression. To design the simulation experiment, these methods typically employ acquisition functions, which are myopic approximations of the optimal allocation of simulation effort to facilitate sequential design. Examples of acquisition functions include expected improvement (Jones et al. 1998) and the knowledge gradient (Ryzhov et al. 2012). Although Cakmak et al. (2022) assume $\mathscr{X}$ is finite, they use a Gaussian process metamodel to more efficiently maximize the posterior PCS (PPCS).

In a setting where $\mathscr{X}$ is infinite and it is therefore not possible to simulate every covariate value, a "design" consists of which covariate values in $\mathscr{X}$ to simulate and the number of simulation replications to allocate to them. Hu and Ludkovski (2017), Pearce and Branke (2018), and Ding et al. (2022) all derive sequential designs for a Gaussian process regression to maximize the posterior PGS (PPGS) for a fixed simulation budget. The differences among these papers stem from the choice of acquisition function.

Instead of using a Gaussian process metamodel, Li et al. (2022) and Li et al. (2020) use a Gaussian mixture model. Under their framework, the covariate space, $\mathscr{X}$, and the systems $\{1,2,\ldots,p\}$, are grouped into clusters so that the systems' performances are similarly distributed within each cluster. This clustering aids in the allocation of simulation effort to attain the best possible worst-case PPCS.

In contrast to the Bayesian approaches, Shen et al. (2021) consider R&S+C from a frequentist perspective. For a fixed covariate design, they derive two-stage simulation-replication algorithms that provide a PGS guarantee when averaged across the covariate space. To attain this guarantee, Shen et al. (2021) assume that each system's mean is a linear function of the covariate, and that the real-world distribution of the covariates is known. Two algorithms for setting the number of replications are given: One applies when the simulation output variance does not depend on the value of the covariate, while the second allows for

a non-constant variance across $\mathscr{X}$. Li et al. (2018) extend this paradigm to the case when $\mathscr{X}$ is high dimensional. Specifically, they use LASSO (James et al. 2013) to estimate a reduced set of parameters in the linear metamodel. They also consider a larger class of basis functions than linear, which generalizes the approach of Shen et al. (2021). Their focus is on maximizing the minimum PCS across $\mathscr{X}$.

In this paper we desire the sort of frequentist guarantees obtained by Shen et al. (2021) but with weaker assumptions. We do not employ a metamodel for each system's mean, but instead construct an overall classification model that predicts a good system at any covariate value.

3 R&S+C AS CLASSIFICATION

In this section we formally introduce our R&S+C classifier. Recall that $\boldsymbol{X}$ is the covariate, which is treated as a random vector prior to realization, and there are p competing systems. Let $Y_{j,\ell}(\boldsymbol{x})$, be the output of the ℓth replication from system $j \in \{1,2,\ldots,p\}$, conditional on $\boldsymbol{X} = \boldsymbol{x}$. For any system j we assume the $Y_{j,\ell}(\boldsymbol{x})$ are i.i.d. $N(\mu_j(\boldsymbol{x}), \sigma_j^2(\boldsymbol{x}))$, $\ell = 1,2,\ldots$, with neither the mean nor variance known. We further assume that $Y_{j,\ell}(\boldsymbol{x})$ and $Y_{m,n}(\boldsymbol{x}')$ are simulated independently, when $\boldsymbol{x} \neq \boldsymbol{x}'$. However, for fixed replication ℓ and covariate $\boldsymbol{x}$, $(Y_{1,\ell}(\boldsymbol{x}), Y_{2,\ell}(\boldsymbol{x}), \ldots, Y_{p,\ell}(\boldsymbol{x}))$ may be dependent due to the use of common random numbers. When $\boldsymbol{X} = \boldsymbol{x}$, let $\mu^\star(\boldsymbol{x}) = \max_j \mu_j(\boldsymbol{x})$ be the performance of the optimal selection, which is denoted by $j^\star(\boldsymbol{x}) = \operatorname{argmax}_j \mu_j(\boldsymbol{x})$, with ties broken arbitrarily.

Revisiting the R&S+C setting, upon observing $\boldsymbol{X} = \boldsymbol{x}_0$, we are interested in predicting $j^\star(\boldsymbol{x}_0)$, or a system whose performance does not differ from it by much, without additional simulation.

3.1 R&S+C Classifier Without Noise

To simplify the presentation and provide intuition, suppose that prior to being given a realized value of the covariate, $\boldsymbol{X}$, we can freely query $j^\star(\boldsymbol{x})$ for any $\boldsymbol{x}$. However, when given $\boldsymbol{X} = \boldsymbol{x}_0$, no further queries are possible. One strategy is to query $j^\star(\cdot)$ for some collection of covariate values, $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m\}$, and then to use these data to build a classifier for any $\boldsymbol{x}$; we denote the classifier by $\widehat{j^\star}(\boldsymbol{x})$. Therefore, when given $\boldsymbol{X} = \boldsymbol{x}_0$, we choose $\widehat{j^\star}(\boldsymbol{x}_0)$.

Since we are treating R&S+C as a classification problem, many classifiers could be employed. For this paper we use KNN. To build the KNN classifier, we construct a database of the optimal system selections at different covariate values. Specifically, this database consists of a set of m design points, $\mathscr{D}_m = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m\} \subset \mathscr{X}$, and a corresponding vector of classifications, $\mathscr{R}_m = \{j^\star(\boldsymbol{x}_1), j^\star(\boldsymbol{x}_2), \ldots, j^\star(\boldsymbol{x}_m)\}$. Let $d(\boldsymbol{x}, \boldsymbol{x}')$ be a metric in $\mathscr{X}$, and let $\mathrm{T}(\boldsymbol{x}, \mathscr{D}_m, k)$ be the set of the k closest design points to $\boldsymbol{x}$ in $\mathscr{D}_m$ using this metric, with ties broken arbitrarily. Then the KNN classifier at an $\boldsymbol{x}$ is

$$\mathrm{KNN}_{\mathrm{OtW}}(\boldsymbol{x}, \mathscr{D}_m, \mathscr{R}_m, k) = \operatorname*{argmax}_{j=1,\ldots,p} \sum_{i=1}^{m} I(j^\star(\boldsymbol{x}_i) = j) I(\boldsymbol{x}_i \in \mathrm{T}(\boldsymbol{x}, \mathscr{D}_m, k)), \tag{1}$$

with ties broken arbitrarily. This is precisely the OtW approach described in Section 1: We first *optimize* by obtaining $j^\star(\boldsymbol{x}_i), i = 1,2,\ldots,m$, and then *weight* each selection with $I(\boldsymbol{x}_i \in \mathrm{T}(\boldsymbol{x}, \mathscr{D}_m, k))$ to classify.

To illustrate the differences between OtW and WtO, consider the corresponding WtO approach. In the noiseless case this means one can query $\mu_j(\boldsymbol{x})$ for any system j and covariate value $\boldsymbol{x}$ to build a metamodel $\widehat{\mu}_j(\boldsymbol{x})$. To provide a parallel with our classification method, let KNN regression be the choice of metamodel. While distinct covariate designs could be specified for each system, to make the explanation clearer assume a single design, $\mathscr{D}_m = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m\}$ and a corresponding collection of response values for each system j, $\mathscr{R}_{m,j} = \{\mu_j(\boldsymbol{x}_1), \mu_j(\boldsymbol{x}_2), \ldots, \mu_j(\boldsymbol{x}_m)\}$. Then for system j, the KNN regression metamodel is

$$\widehat{\mu}_j(\boldsymbol{x}) = \frac{1}{k} \sum_{i=1}^{m} \mu_j(\boldsymbol{x}_i) I(\boldsymbol{x}_i \in \mathrm{T}(\boldsymbol{x}, \mathscr{D}_m, k)).$$

This metamodel is constructed by weighting each performance value $\mu_j(\boldsymbol{x}_i)$ by $I(\boldsymbol{x}_i \in \mathrm{T}(\boldsymbol{x}, \mathscr{D}_m, k))/k$. Then conditional on $\boldsymbol{X} = \boldsymbol{x}$, one can optimize by selecting

$$\mathrm{KNN}_{\mathrm{WtO}}(\boldsymbol{x}, \mathscr{D}_m, \{\mathscr{R}_{m,j}\}_{j=1}^m, k) = \underset{j=1,\ldots,p}{\operatorname{argmax}}\ \widehat{\mu}_j(\boldsymbol{x}) = \underset{j=1,\ldots,p}{\operatorname{argmax}} \sum_{i=1}^m \mu_j(\boldsymbol{x}_i) I(\boldsymbol{x}_i \in \mathrm{T}(\boldsymbol{x}, \mathscr{D}_m, k)).$$

That is, we "weight then optimize," which is the most common approach described in Section 2. The primary difference in the approaches is the information being used: an OtW approach uses the queries of the optimal system while the WtO approach uses the actual responses.

3.2 R&S+C Classifier With Noise

In a stochastic simulation we cannot query $j^\star(\boldsymbol{x})$. However, R&S procedures provide a proxy for $j^\star(\boldsymbol{x})$. Given a covariate value $\boldsymbol{x}$, a R&S procedure can be executed to obtain replications $Y_{j,\ell}(\boldsymbol{x})$ for each system $j = 1, 2, \ldots, p$ as required by the procedure. A R&S procedure executed at $\boldsymbol{x}$ is a function, $R(\boldsymbol{x}, \alpha, \delta)$, where $1-\alpha$ is a desired PGS and $\delta > 0$ is the allowable optimality gap. Notice that this function is itself a random variable, with outputs taking values in $\{1, 2, \ldots, p\}$ and (we assume) it satisfies $\Pr\left\{\mu^\star(\boldsymbol{x}) - \mu_{R(\boldsymbol{x},\alpha,\delta)}(\boldsymbol{x}) \le \delta | \boldsymbol{X} = \boldsymbol{x}\right\} \ge 1-\alpha$ for each $\boldsymbol{x} \in \mathscr{X}$.

For a design consisting of covariate values $\mathscr{D}_m = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m\}$, a desired PGS level of $1-\alpha$, and an optimality gap of δ, let $\mathscr{R}_m = \{R(\boldsymbol{x}_1, \alpha, \delta), R(\boldsymbol{x}_2, \alpha, \delta), \ldots, R(\boldsymbol{x}_m, \alpha, \delta)\}$, where $R(\boldsymbol{x}_i, \alpha, \delta)$ is the value returned by the R&S procedure at covariate $\boldsymbol{x}_i$. Analogous to Equation (1), we propose the R&S+C classifier

$$\mathrm{KNN}(\boldsymbol{x}, \mathscr{D}_m, \mathscr{R}_m, k) = \underset{j=1,\ldots,p}{\operatorname{argmax}} \sum_{i=1}^m I(R(\boldsymbol{x}_i, \alpha, \delta) = j) I(\boldsymbol{x}_i \in \mathrm{T}(\boldsymbol{x}, \mathscr{D}_m, k)), \tag{2}$$

with ties broken arbitrarily; see Algorithm 1. We discuss the choice of design $\mathscr{D}_m$ and the number of nearest neighbors k later.

Algorithm 1 R&S+C Classifier with KNN

Input: a R&S procedure, $R(\cdot, \alpha, \delta)$, and design, $\mathscr{D}_m$
initialize $\mathscr{R}_m = \emptyset$
for each design point, $\boldsymbol{x}_i \in \mathscr{D}_m$, $i = 1, 2, \ldots, m$ **do**
  $\mathscr{R}_m \leftarrow \mathscr{R}_m \bigcup R(\boldsymbol{x}_i, \alpha, \delta)$
end for
return $\mathrm{KNN}(\cdot, \mathscr{D}_m, \mathscr{R}_m, k)$

In Equation (2), the only change from Equation (1) in Section 3.1 is that the true optimal system at a design point is replaced by the R&S outcome. This substitution produces two important differences. The first is that at a design point $\boldsymbol{x}_i$, the R&S procedure only promises a good selection with probability $1-\alpha$, so there is some chance of returning a system whose mean is far from that of $j^\star(\boldsymbol{x}_i)$. The second is that the promise is of a *good* selection, not the best, where "good" is a system whose mean performance at $\boldsymbol{x}_i$ is within δ of the best system. Therefore, even when the procedure returns a good selection, there is some chance that $R(\boldsymbol{x}_i, \alpha, \delta) \neq j^\star(\boldsymbol{x}_i)$.

By employing a R&S procedure at each design point, our approach makes no assumptions regarding the form of $\mu_i(\cdot)$ or the relationships among the systems. Furthermore, it gives a probabilistic guarantee on the selected system at each design point independent of our ultimate choice of classification metamodel. Recall that to provide a PGS guarantee in the WtO procedure of Shen et al. (2021) they make a strong assumption that each system's mean performance is linear in the covariate.

3.3 R&S+C Guarantees

As is standard in R&S, we would like to assess the statistical guarantees that Algorithm 1 provides. Suppose that we desire PGS $\geq 1-\alpha$ with optimality gap δ. To be meaningful, the "PGS" should have some interpretation across the entire covariate space. There are at least two versions, the first being a *pointwise PGS guarantee*:

$$\Pr\left\{\mu^\star(\boldsymbol{x}_0) - \mu_{\mathrm{KNN}(\boldsymbol{x}_0,\mathscr{D}_m,\mathscr{R}_m,k)}(\boldsymbol{x}_0) \leq \delta | \boldsymbol{X} = \boldsymbol{x}_0\right\} \geq 1-\alpha \text{ for each } \boldsymbol{x}_0 \in \mathscr{X}. \tag{3}$$

This is a marginal guarantee on every possible covariate value $\boldsymbol{x}_0$ individually, and therefore very strong. We note that the only random variables in the pointwise PGS guarantee if $\mathscr{D}_m$ is fixed are the R&S results $\mathscr{R}_m$. The second guarantee is the *expected PGS guarantee*, defined as:

$$\Pr\left\{\mu^\star(\boldsymbol{X}) - \mu_{\mathrm{KNN}(\boldsymbol{X},\mathscr{D}_m,\mathscr{R}_m,k)}(\boldsymbol{X}) \leq \delta\right\} \geq 1-\alpha. \tag{4}$$

Clearly, if a procedure attains the pointwise PGS guarantee then it attains the expected PGS guarantee as well, but the reverse is not true. Shen et al. (2021) focus on expected PGS.

One advantage of our classification approach is the ability to prove results such as (3) and (4) under certain conditions using well-established properties of the R&S procedures themselves; we leave those for a future paper. Here we establish a third guarantee, *asymptotic pointwise consistency* as the number of design points m and nearest neighbors k increase. This result provides assurance that the classification performance gets better and better as we expend more and more simulation effort, so that any desired level of PGS can be achieved. Theorem 1 below will employ the following assumptions:

(A1): $\mathscr{X} \subseteq \Re^d$
(A2): $\boldsymbol{X} \sim F$ with support $\mathscr{X}$
(A3): $\mu_i(\cdot),\ i = 1,2,\ldots,p$ are continuous on $\mathscr{X}$
(A4): For any $\boldsymbol{x} \in \mathscr{X}$, $\Pr\left\{\mu^\star(\boldsymbol{x}) - \mu_{R(\boldsymbol{x},\alpha,\delta)}(\boldsymbol{x}) \leq \delta | \boldsymbol{X} = \boldsymbol{x}\right\} \geq 1-\alpha$
(A5): $\mathscr{D}_m = \{\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_m\}$, where $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots$ are i.i.d. F.
(A6): R&S procedures executed at different design points in $\mathscr{D}_m$ are independent of each other.
(A7): The KNN metric is the Euclidean distance.

Theorem 1 If assumptions (A1)–(A7) hold, $\delta > 0$, $\alpha < 1/p$, and $m \to \infty$ implies $k \to \infty$ with $k/m \to 0$, then for all $\boldsymbol{x}_0 \in \mathscr{X}$ such that for any $\lambda > 0$, $\Pr\{\boldsymbol{X} \in B(\boldsymbol{x}_0, \lambda)\} > 0$, where $B(\boldsymbol{x}, \lambda)$ is a Euclidean ball with radius λ centered at $\boldsymbol{x}$, we have

$$\lim_{m\to\infty} \Pr\left\{\mu^\star(\boldsymbol{x}_0) - \mu_{\mathrm{KNN}(\boldsymbol{x}_0,\mathscr{D}_m,\mathscr{R}_m,k)}(\boldsymbol{x}_0) \leq \delta | \boldsymbol{X} = \boldsymbol{x}_0\right\} = 1.$$

Some additional notation and two basic results will be useful in the proof. For fixed $\boldsymbol{x}_0$, let $\mathbf{X}_{j:m}$ be the jth closest design point in $\mathscr{D}_m$ to $\boldsymbol{x}_0$ in Euclidean distance; therefore $\{\mathbf{X}_{1:m}, \mathbf{X}_{2:m}, \ldots, \mathbf{X}_{k:m}\}$ are the k nearest neighbors to $\boldsymbol{x}_0$ in $\mathscr{D}_m$. Furthermore, let

$$A_{j:m} = I\left(\mu^\star(\boldsymbol{x}_0) - \mu_{R(\mathbf{X}_{j:m},\alpha,\delta)}(\boldsymbol{x}_0) \leq \delta\right)$$

be an indicator of whether the system selected via R&S for design point $\mathbf{X}_{j:m}$ would be a good selection for covariate value $\boldsymbol{x}_0$. Finally, let $R(\boldsymbol{x}) = R(\boldsymbol{x}, \alpha, \delta)$ for simplicity. The following elementary results will be employed:

1. If $B_m \sim \mathrm{Binomial}(m, q)$ with $0 \leq q \leq 1$ and $k \leq mq$ then from Hoeffding's inequality

$$\Pr\{B_m \leq k\} \leq \exp\left\{-2m\left(q - \frac{k}{m}\right)^2\right\}. \tag{5}$$

2. For any $\boldsymbol{x} \in \mathscr{X}$, the random variable $I\left(\mu^\star(\boldsymbol{x}) - \mu_{R(\boldsymbol{x},\alpha,\delta)}(\boldsymbol{x}) \le \delta\right)$ is stochastically larger than $C \sim \text{Bernoulli}(1-\alpha)$ by properties of the R&S procedure in (A4).

Proof. The key insight is that a good selection for $\boldsymbol{x}_0$ must occur if

$$k - \sum_{j=1}^{k} A_{j:m} < \frac{1}{p-1} \sum_{j=1}^{k} A_{j:m}.$$

This can be seen by noting that the left-hand side is the number of non-good selections among the KNN, while the right-hand side is the number of good selections, but spread out evenly over the maximum number of possible "good" systems without all of them being good. We will show that for arbitrary $\varepsilon > 0$ there is an m_ε large enough that for all $m \ge m_\varepsilon$

$$\Pr\left\{ k - \sum_{j=1}^{k} A_{j:m} \ge \frac{1}{p-1} \sum_{j=1}^{k} A_{j:m} \right\} \le \varepsilon.$$

Define $\mathscr{G}(\boldsymbol{x}) = \{j\colon \mu^\star(\boldsymbol{x}) - \mu_j(\boldsymbol{x}) \le \delta\}$ to be the set of good systems for covariate value $\boldsymbol{x}$, and $\mathscr{G}^c(\boldsymbol{x})$ its complement. By the continuity assumption (A3), there exists a $\lambda^\star > 0$ but small enough that if $j \in \mathscr{G}^c(\boldsymbol{x}_0)$ then $j \in \mathscr{G}^c(\boldsymbol{x})$ for $\boldsymbol{x} \in B(\boldsymbol{x}_0, \lambda^\star)$; that is, if system j is not a good selection for $\boldsymbol{x}_0$ then it is also not a good selection for any $\boldsymbol{x} \in B(\boldsymbol{x}_0, \lambda^\star)$.

Let $q(\boldsymbol{x}_0) = \Pr\{\boldsymbol{X} \in B(\boldsymbol{x}_0, \lambda^\star)\}$, which is greater than zero by assumption on $\boldsymbol{x}_0$, and

$$N_{k,m} = \#\left\{\mathbf{X}_{1:m}, \mathbf{X}_{2:m}, \ldots, \mathbf{X}_{k:m} \in B(\boldsymbol{x}_0, \lambda^\star)\right\}.$$

To state the result entirely in terms of the number of design points m, let $k = k(m)$ with the only restrictions that $k(m)$ is nondecreasing in m, $k(m) \to \infty$ and $k(m)/m \to 0$ as $m \to \infty$. Then using (5) we can show that with $(k(m)-1) \le mq(\boldsymbol{x}_0)$ we have

$$\Pr\{N_{k,m} < k\} = \Pr\{B_m \le k-1\} \le \exp\left\{ -2m\left(q(\boldsymbol{x}_0) - \frac{k-1}{m} \right)^2 \right\} \to 0$$

as $m \to \infty$ and $k(m)/m \to 0$, where $B_m \sim \text{Binomial}(m, q(\boldsymbol{x}_0))$. Thus, for $\varepsilon_1 = \varepsilon/2$ there exists an m_1 such that $\Pr\{N_{k,m} < k\} \le \varepsilon_1$ for all $m \ge m_1$.

Next, when $N_{k,m} = k$, a good selection for any of $\{\mathbf{X}_{1:m}, \mathbf{X}_{2:m}, \ldots, \mathbf{X}_{k:m}\}$ is also a good selection for $\boldsymbol{x}_0$ because $N_{k,m} = k$ implies there are no covariates leading to bad selections at $\boldsymbol{x}_0$ in this set. Therefore,

$$\begin{aligned} \Pr\left\{ k - \sum_{j=1}^{k} A_{j:m} \ge \frac{1}{p-1} \sum_{j=1}^{k} A_{j:m} \middle| N_{k,m} = k \right\} &= \Pr\left\{ \frac{1}{k} \sum_{j=1}^{k} A_{j:m} \le 1 - \frac{1}{p} \middle| N_{k,m} = k \right\} \\ &\le \Pr\left\{ \frac{1}{k} \sum_{j=1}^{k} C_j \le 1 - \frac{1}{p} \right\} \end{aligned} \tag{6}$$

where $C_1, C_2, \ldots$ are i.i.d. Bernoulli$(1-\alpha)$, by the stochastic ordering noted above. Then since $1 - 1/p < 1 - \alpha$, the weak law of large numbers implies that there is an m_2 such that (6) $\le \varepsilon_2 = \varepsilon/2$ for all $m \ge m_2$ (notice that here m_2 ensures that k is large enough). Thus, by the law of total probability

$$\begin{aligned} &\Pr\left\{ k - \sum_{j=1}^{k} A_{j:m} \ge \frac{1}{p-1} \sum_{j=1}^{k} A_{j:m} \right\} \\ &= \Pr\left\{ \frac{1}{k} \sum_{j=1}^{k} A_{j:m} \le 1 - \frac{1}{p} \middle| N_{k,m} = k \right\} \Pr\{N_{k,m} = k\} + \Pr\left\{ \frac{1}{k} \sum_{j=1}^{k} A_{j:m} \le 1 - \frac{1}{p} \middle| N_{k,m} < k \right\} \Pr\{N_{k,m} < k\} \\ &\le \Pr\left\{ \frac{1}{k} \sum_{j=1}^{k} A_{j:m} \le 1 - \frac{1}{p} \middle| N_{k,m} = k \right\} + \Pr\{N_{k,m} < k\} \le \varepsilon_1 + \varepsilon_2 = \varepsilon \end{aligned}$$

for all $m \geq m_\varepsilon = \max\{m_1, m_2\}$. □

Remark 1 Theorem 1 holds for $\boldsymbol{x}_0$ such that the probability of design points falling in a neighborhood around $\boldsymbol{x}_0$ is positive. Without this condition, design points may never be close enough to $\boldsymbol{x}_0$ to discover the good systems at $\boldsymbol{x}_0$. This mild condition is easily satisfied. For example, any interior point of $\mathscr{X}$ satisfies this condition if the density of F is positive and continuous at $\boldsymbol{x}_0$.

4 EMPIRICAL EVALUATION

In this section we empirically evaluate our R&S+C classifier (RSCC) using two separate problems: the first being the benchmark problems in Shen et al. (2021), and the second a two-product single-period inventory problem.

The RSCC procedure requires the following parameters: the R&S procedure to use, the PGS, $1-\alpha$, the optimality gap, δ, the initial number of replications for the procedure, n_0, the covariate design, $\mathscr{D}_m$, and the choice of classifier. Here we employ KN (Kim and Nelson 2001) for R&S, $1-\alpha = 0.95$ and KNN classifier with $k = 1$. KN is widely believed to provide a PGS guarantee, although this has not been proven. The initial number of replications, optimality gap, and design depend on the problem.

We compare RSCC to the two procedures of Shen et al. (2021), TS and TS$^+$. Given procedure parameters $\mathscr{D}_m$, n_0, δ, and α, both procedures provide an expected PGS guarantee of $1-\alpha$ under the assumption that each $\mu_j(\cdot)$ is a linear function of the covariate. TS further assumes that the output variance is a constant across the covariate space, while TS$^+$ allows for unequal variances. The benchmark problems satisfy the assumptions of either TS or TS$^+$, but the inventory problem has neither a linear response nor equal variances.

Two metrics are employed to evaluate the procedures: the expected PGS with respect to the covariate distribution and the expected number of replications consumed by the procedures. As both test problems are tractable, whether or not a good selection is made can be evaluated. A single fixed set of 10,000 points is generated from the covariate distribution. Then each procedure is applied to each problem 1,000 times (macroreplications) building a classifier (RSCC) or linear models (TS and TS$^+$) in each case. The metrics are then evaluated using the set of test points.

For each of the procedures, let $\widehat{j}^\star(\boldsymbol{x})$ represent the selected system at covariate $\boldsymbol{x}$. Define $\boldsymbol{x}_s$ as the sth point in the test set. The expected PGS (EPGS) is estimated by

$$\widehat{\text{EPGS}} = \frac{1}{1{,}000 \cdot 10{,}000} \sum_{r=1}^{1{,}000} \sum_{s=1}^{10{,}000} I(\mu^\star(\boldsymbol{x}_s) - \mu_{\widehat{j}^\star(\boldsymbol{x}_s)}(\boldsymbol{x}_s) < \delta)_r,$$

where $I(\mu^\star(\boldsymbol{x}_s) - \mu_{\widehat{j}^\star(\boldsymbol{x}_s)}(\boldsymbol{x}_s) < \delta)_r$ is the indicator of whether the procedure made a good selection at $\boldsymbol{x}_s$ for macroreplication r. Let $N(\boldsymbol{x}_s)_r$ be the number of replications the procedure consumed at $\boldsymbol{x}_s$ for replication r. The expected number of replications, (ENR), is estimated by

$$\widehat{\text{ENR}} = \frac{1}{1{,}000 \cdot 10{,}000} \sum_{r=1}^{1{,}000} \sum_{i=1}^{10{,}000} N(\boldsymbol{x}_s)_r.$$

4.1 Benchmark Problems

We first reproduce the benchmark problem suite of Shen et al. (2021), which consists of a baseline problem and variations on it. In the baseline problem, $Y_{j,\ell}(\boldsymbol{x}) = \mu_j(\boldsymbol{x}) + \varepsilon_{\ell,j} = \beta_{0j} + \boldsymbol{x}^\top \beta + \varepsilon_{\ell,j}$, where $\boldsymbol{x}$ and β are $d \times 1$ with $d = 3$, and $\beta_{01} - \delta = \beta_{0j} = 0,\ j = 2, 3, \ldots, p$, with $p = 5$ and $\delta = 1$. The $\varepsilon_{j,\ell}$ are i.i.d. $N(0, \sigma^2)$. In words, the mean responses of the systems as functions of the covariate are parallel planes; Shen et al. (2021) establish that this is the least favorable configuration in terms of EPGS for their procedure. The elements of the covariate $\boldsymbol{X} = (X_1, X_2, \ldots, X_d)^\top$ are i.i.d. Uniform$(0,1)$. In addition, $\alpha = 0.05$ and $\mathscr{D}_m = \{0, 0.5\}^d$. The variations include changing the number of systems, allowing the variances to be unequal across systems,

Table 1: Results for the benchmark problem suite.

	$\widehat{\text{EPGS}}$			$\widehat{\text{ENR}}$		
Procedure	RSCC	TS	TS$^+$	RSCC	TS	TS$^+$
Baseline	0.96	0.96	0.98	21,982	48,276	67,222
$p = 2$	0.95	0.95	0.97	4,957	9,446	13,026
$p = 8$	0.96	0.97	0.99	39,877	95,707	133,034
Increasing σ	0.96	0.96	0.98	19,630	54,265	75,511
Decreasing σ	0.97	0.96	0.98	28,763	54,223	75,341
Heteroscedastic σ	0.97	0.94	0.99	31,591	60,242	83,994
$d = 1$	0.95	0.97	0.97	5,511	22,545	25,716
$d = 5$	0.96	0.96	0.99	88,014	72,821	117,234
Normal Covariates	0.96	0.96	0.98	22,066	47,477	65,886

changing the dimension of the covariate, and changing the distribution of the covariate. See Shen et al. (2021) for complete details. Notice that since Shen et al. (2021) count the intercept term as a covariate their values of d are one greater than ours, but these are only notational differences.

Although the benchmark problems are the least favorable for TS and TS$^+$, they are favorable for RSCC because a correct selection at one design point is a correct selection everywhere. Thus, our focus is on comparing the number of replications consumed. Notice that we treat system 1 as the single good system, even though in our assumption (A4) all systems would be considered "good."

The empirical results are found in Table 1. We observe that the results for TS and TS$^+$ are consistent with the results in Shen et al. (2021). Because RSCC employs KN at each design point, and KN accommodates unequal variances, the configuration of the variances does not affect the PGS guarantees of RSCC. Thus, both RSCC and TS$^+$ achieve the nominal 95% PGS for every set of problem parameters. TS might not achieve the specified PGS when the variances are heteroscedastic, but it is close. The RSCC procedure requires substantially fewer replications for every problem except for the case when $d = 5$. We attribute this to the large size of the design in the higher dimension, which is much larger than RSCC actually needs to achieve satisfactory EPGS on this particular problem. Nonetheless, this shows that in difficult settings for TS and TS$^+$, RSCC can perform quite well.

4.2 Two-product, Single-period Inventory Problem

We next compare TS$^+$ and RSCC on a two-product, single-period inventory problem, primarily to evaluate the impact of number of design points, m, as well as the type of design.

In the classical single-period inventory problem, a one-time order must be placed at the beginning of the selling period to accommodate stochastic demand given the per-item purchase cost and sale price to maximize the expected profit. In our example there are two products, and although their demands are independent, the total order quantity of both products is constrained, and each product can only be ordered in certain multiples. The feasible order quantities implied by these constraints define the set of competing systems. Further, there are two selling periods (but with no carry over). The demand for each product is identically distributed across both periods, but strongly correlated. Specifically, the demand for product i in period 1 (denoted $D_{1,i}$) and period 2 (denoted $D_{2,i}$) are bivariate normally distributed. Therefore, after observing $D_{1,i}$, the demand for product i in period 2 can be refined to make a better decision. Thus, $D_{1,i}$ can be employed as a covariate. Specifically, the demand in period 2 of product i, given covariate $D_{1,i}$, has distribution

$$D_{2,i} \mid D_{1,i} \sim \mathrm{N}\left(\eta + \rho(D_{1,i} - \eta_i),\ (1-\rho^2)\varsigma^2\right)$$

where η and ς^2 are the unconditional mean and variance of demand, respectively, and ρ is the correlation between demands in periods 1 and 2. An analogous model holds for both products. The distribution parameters are chosen to be the same for both products to simplify the analysis of the empirical evaluation.

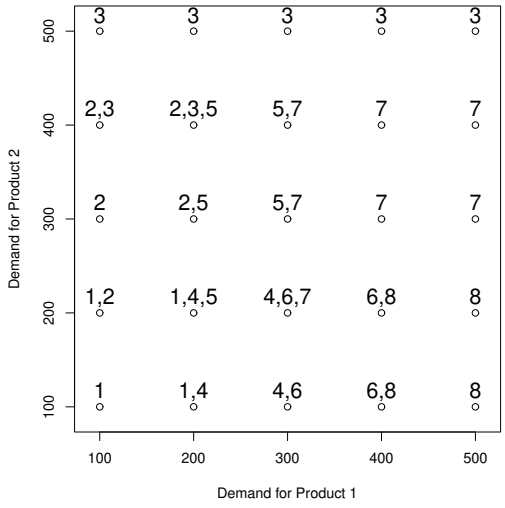

Figure 1: Good systems for a grid of covariate pairs for the two-product, single-period inventory problem.

To compare RSCC and TS$^+$, the distribution parameters of the inventory problem are specified as follows: $\varsigma^2 = 40$, $\eta = 195$, $\rho = 0.9$. For product 1, the per-item selling price and the per-item cost of purchase are \$10 and \$6, respectively. For product 2, the per-item selling price and the per-item cost of purchase are \$15 and \$7, respectively. The total order constraint is set at 600. The multiples of product 1 that can be purchased are 100 with a minimum purchase amount of 100 and the multiples of product 2 that can be purchased are 150 with a minimum purchase amount of 150. Therefore, the different order quantities of both products are, $(100,150),(100,300),(100,450),(200,150),(200,300),(300,150),(300,300),(400,150)$, which we denote as systems $1,2,\ldots,8$, respectively. Figure 1 lists the good systems at a grid of covariate pairs $(D_{1,1},D_{1,2})$ to give a sense of the problem.

The procedure parameters that are varied are the first-stage sample size n_0 and the number of design points m. The value of n_0 is specified to be a function of δ and the average variance of the profits. Specifically, $n_0 = (\psi/\delta)^2\bar{V}$. To approximate $\bar{V}$, we use Monte Carlo simulation. Two different values of ψ are used: 4 and 5. These values of ψ result in values of 6 and 9 for n_0, respectively. Recall that the main purpose of the inventory problem is to investigate how EPGS and number of replications change as the number of design points m changes. To that end, $m = 5,10,15,20,25,30$ are tested.

We set $\alpha = 0.05$. The remaining experiment parameters are δ and the choice of design points. To ensure that δ is large enough that there is more than one good selection at most covariate values, while ensuring that δ is small enough that at most covariate values not all the systems are good selections, we choose δ such that $\Pr(\mu^\star(D_1) - \max_{j\neq i^\star}\mu_j(D_1) \leq \delta) = 0.9$. Thus, δ is the 90th percentile difference between the mean profit of the optimal system and the mean profit of the second-best system at D_1, where D_1 follows the covariate distribution described above. This approach gave $\delta = \$363$ using Monte Carlo.

For the experiment design we use Latin hypercube samples of increasing size. For a design of size m, a random Latin hypercube sample of size m is generated taking values in $[0,1]$. The inverse cdf of the covariate distribution then returns quantiles in the covariate space based on the uniform quantiles generated from the Latin hypercube sample. The returned quantiles are the design. Since TS$^+$ assumes a linear model, we also apply it with the minimal full factorial design. If π is an extreme uniform quantile, a factorial design of $\{\pi, 1-\pi\}^2$ is generated. The inverse cdf of the covariate distribution again returns the quantiles in the covariate space for the factorial design. We use two factorial designs, $\pi = 0.05$ and $\pi = 0.01$.

Table 2: Results for the two-product single-period inventory problem with LHS design.

		Size of Design					
Procedure	Result	5	10	15	20	25	30
RSCC	$\widehat{\text{EPGS}}$	0.95	0.98	0.99	0.99	0.99	1
$\psi = 5$	$\widehat{\text{ENR}}$	430	860	1,293	1,724	2,154	2,582
RSCC	$\widehat{\text{EPGS}}$	0.95	0.98	0.99	0.99	0.99	1
$\psi = 4$	$\widehat{\text{ENR}}$	422	860	1,277	1,707	2,140	2,582
TS^+	$\widehat{\text{EPGS}}$	1	1	1	1	1	1
$\psi = 5$	$\widehat{\text{ENR}}$	2,070	1,368	1,568	1,815	2,106	2,404
TS^+	$\widehat{\text{EPGS}}$	1	1	1	1	1	1
$\psi = 4$	$\widehat{\text{ENR}}$	3,336	2,315	2,587	2,836	3,136	3,416

Table 3: Results for TS^+ for the two-product single-period inventory problem with factorial design.

π	ψ	$\widehat{\text{EPGS}}$	$\widehat{\text{ENR}}$
$\pi = 0.05$	$\psi = 4$	1	746
	$\psi = 5$	1	469
$\pi = 0.01$	$\psi = 4$	0.95	591
	$\psi = 5$	0.94	384

We ran 1,000 macro replications for each case. The $\widehat{\text{EPGS}}$ and $\widehat{\text{ENR}}$ results for the Latin hypercube designs are found in Table 2. In all cases, both RSCC and TS^+ achieve a PGS of at least 0.95. However, both tend to overachieve the nominal PGS, in some cases substantially. RSCC consumes fewer replications than TS^+ until the design size grows above $m = 25$. Furthermore, the expected number of replications consumed by TS^+ appears not to be very sensitive to the design size. Since TS^+ assumes a linear model, replications and design points are essentially interchangeable. The replications consumed by RSCC are directly related to the design size since it executes an independent R&S procedures at each design point. Fortunately, RSCC achieves the desired PGS with a small design.

The factorial results for procedure TS^+ are found in Table 3. Because the mean response functions are not linear, the placement of the factorial points impacts the results. In principle, if the mean performance functions were truly linear, then the more extreme the factorial design points are, the better. However, in this example the more extreme design $\pi = 0.01$ yields lower EPGS than the less extreme design $\pi = 0.05$. Considering $\widehat{\text{ENR}}$ of TS^+, the factorial designs give similar $\widehat{\text{ENR}}$ to RSCC and far fewer replications than the Latin hypercube design. However, a trade-off between a factorial and Latin hypercube design for TS^+ exists. In the inventory problem, while fewer replications are needed for the factorial design, if the design points are not appropriately placed, then the linear model may not well approximate the nonlinear surface and the specified EPGS may not be achieved.

Overall, our empirical evaluation demonstrates that a classification approach, built on top of workhorse R&S procedures, can be an effective and efficient solution to R&S+C problems without strong assumptions or prior knowledge. Experiment design for RSCC is the key outstanding research problem.

ACKNOWLEDGEMENTS

This research was partially supported by NSF Grant Nos. DMS-1854562 and DMS-1953111.

REFERENCES

Ban, G.-Y., and C. Rudin. 2019. "The Big Data Newsvendor: Practical Insights from Machine Learning". *Operations Research* 67(1):90–108.

Bertsimas, D., and N. Kallus. 2020. "From Predictive to Prescriptive Analytics". *Management Science* 66(3):1025–1044.

Cakmak, S., S. Gao, and E. Zhou. 2022. "Contextual Ranking and Selection with Gaussian Processes". https://arxiv.org/abs/2201.07782, accessed 27 September.

Ding, L., L. J. Hong, H. Shen, and X. Zhang. 2022. "Knowledge Gradient for Selection with Covariates: Consistency and Computation". *Naval Research Logistics* 69(3):496–507.

Gao, S., J. Du, and C.-H. Chen. 2019. "Selecting the Optimal System Design under Covariates". In *Proceedings of the 15th International Conference on Automation Science and Engineering*, 547–552. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Hu, R., and M. Ludkovski. 2017. "Sequential Design for Ranking Response Surfaces". *SIAM/ASA Journal on Uncertainty Quantification* 5(1):212–239.

James, G., D. Witten, T. Hastie, and R. Tibshirani. 2013. *An Introduction to Statistical Learning*. New York: Springer.

Jin, X., H. Li, and L. H. Lee. 2019. "Optimal Budget Allocation in Simulation Analytics". In *Proceedings of the 15th International Conference on Automation Science and Engineering*, 178–182. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Jones, D. R., M. Schonlau, and W. J. Welch. 1998. "Efficient Global Optimization of Expensive Black-Box Functions". *Journal of Global Optimization* 13(4):455–492.

Kannan, R., G. Bayraksan, and J. R. Luedtke. 2022. "Data-driven Sample Average Approximation with Covariate Information". *Optimization Online*. http://www.optimization-online.org/DBHTML/2020/07/7932.html, accessed 27 September.

Kim, S.-H., and B. L. Nelson. 2001. "A Fully Sequential Procedure for Indifference-Zone Selection in Simulation". *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 11(3):251–273.

Li, H., H. Lam, Z. Liang, and Y. Peng. 2020. "Context-Dependent Ranking and Selection Under a Bayesian Framework". In *Proceedings of the 2020 Winter Simulation Conference*, edited by K.-H. G. Bae, B. Feng, S. Kim, S. Lazarova-Molnar, Z. Zheng, T. Roeder, and R. Thiesing, 2060–2070. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Li, H., H. Lam, and Y. Peng. 2022. "Efficient Learning for Clustering and Optimizing Context-Dependent Designs". *Operations Research*. https://doi.org/10.1287/opre.2022.2368, accessed 27 September.

Li, X., X. Zhang, and Z. Zheng. 2018. "Data-Driven Ranking and Selection: High-Dimensional Covariates and General Dependence". In *Proceedings of the 2018 Winter Simulation Conference*, edited by M. Rabe, A. A. Juan, N. Mustafee, A. Skoogh, S. Jian, and B. Johansson, 1933–1944. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Pearce, M., and J. Branke. 2018. "Continuous Multi-Task Bayesian Optimisation with Correlation". *European Journal of Operational Research* 270(3):1074–1085.

Ryzhov, I. O., W. B. Powell, and P. I. Frazier. 2012. "The Knowledge Gradient Algorithm for a General Class of Online Learning Problems". *Operations Research* 60(1):180–195.

Shen, H., L. J. Hong, and X. Zhang. 2021. "Ranking and Selection with Covariates for Personalized Decision Making". *INFORMS Journal on Computing* 33(4):1500–1519.

AUTHOR BIOGRAPHIES

GREGORY KESLIN is a PhD student in the Department of Industrial Engineering and Management Sciences at Northwestern University. His research interests are simulation optimization and statistics. His e-mail address is gregorykeslin2025@u.northwestern.edu

BARRY L. NELSON is the Walter P. Murphy Professor in the Department of Industrial Engineering and Management Sciences at Northwestern University. He is a Fellow of INFORMS and IISE. His e-mail address is nelsonb@northwestern.edu.

BERNARDO K. PAGNONCELLI is an Associate Professor at SKEMA Business School, in Lille, France. His research focuses on theory and applications of stochastic optimization. His e-mail address is bernardo.pagnoncelli@skema.edu.

MATTHEW PLUMLEE is an Assistant Professor in the Department of Industrial Engineering and Management Sciences at Northwestern University. His e-mail address is mplumlee@northwestern.edu.

HAMED RAHIMIAN is an Assistant Professor in the Department of Industrial Engineering at Clemson University. His research focuses on stochastic and robust optimization. His e-mail address is hrahimi@clemson.edu.