Proceedings of the 2021 Winter Simulation Conference
S. Kim, B. Feng, K. Smith, S. Masoud, Z. Zheng, C. Szabo, and M. Loper, eds.

A UNIFIED OFFLINE-ONLINE LEARNING PARADIGM VIA SIMULATION FOR SCENARIO-DEPENDENT SELECTION

Haitao Liu
Xiao Jin
Haobin Li
Loo Hay Lee
Ek Peng Chew

Department of Industrial Systems Engineering and Management
National University of Singapore
1 Engineering Drive 2
Singapore, 117576, SINGAPORE

ABSTRACT

Simulation has primarily been used for offline static system design problems, and the simulation-based online decision making has been a weakness as the online decision epoch is tight. This work extends the scenario-dependent ranking and selection model by considering online scenario and budget. We propose a unified offline-online learning (UOOL) paradigm via simulation to find the best alternative conditional on the online scenario. The idea is to offline learn the relationship between scenarios and mean performance, and then dynamically allocates the online simulation budget based on the learned predictive model and online scenario information. The superior performance of UOOL paradigm is validated on four test functions by comparing it with artificial neural networks and decision tree.

1 INTRODUCTION

There are many systems in which their performances can only be estimated via simulation, and decisions should be periodically made based on the *scenario* observed at that time (Hong and Jiang 2019; Pedrielli et al. 2019). As the scenarios in practice are unpredictable, we thus refer them to as *online scenarios*. The best alternative with respect to some performance criterion generally is not universal but depends on the scenario. As depicted in Figure 1, after an online scenario is revealed, the available decision epoch (e.g., one hour) before the decision deadline typically is tight such that we cannot conduct many simulation replications online, which actually poses great challenges to the *scenario-dependent selection* of best alternatives. We can view it as an online decision-making problem aiming to find the best alternative conditioning on the online scenario.

Recently, a few works have considered the scenario-dependent ranking and selection (R&S) problems, where the scenarios are also known as the covariates, contexts, or side information. Gao et al. (2019) considered a discrete scenario space and proposed an optimal computing budget allocation (OCBA) rule to maximize the rate function of probability of false selection. Under a robust perspective, Li et al. (2020) also considered a number of finite and discrete scenarios and developed a dynamic sampling policy to optimize the worst-case probability of correct selection (PCS) of all scenarios. Ding et al. (2019) considered a continuous scenario space and developed the integrated knowledge gradient (KG) policy to dynamically maximize the expected increment in the maximum over scenario space. Under the indifference-zone formulation, Shen et al. (2021) developed two-stage procedures to achieve the targeted PCS which takes the average over the continuous scenario space. These works mainly emphasize offline simulation budget

978-1-6654-3311-2/21/$31.00 ©2021 IEEE

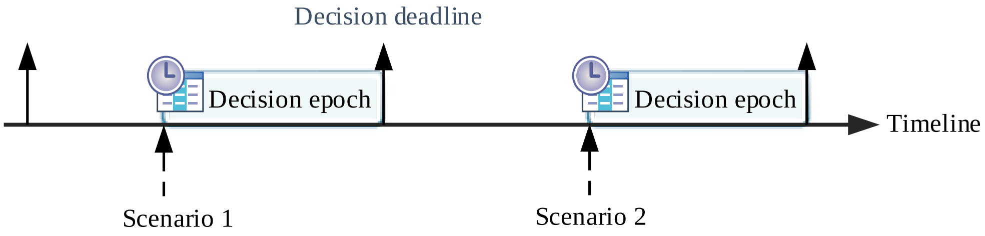


Figure 1: Online scenarios and decision epoch.

allocation to maximize unconditional PCS or expected opportunity cost (EOC). However, we are interested in selecting the best alternative conditional on online scenario, by bridging offline and online simulation and utilizing online scenario information.

To our knowledge, existing works in fact assume that simulation is performed offline, and then offline learn parametric (Brantley et al. 2013; Xiao et al. 2015; Xiao et al. 2021; Shen et al. 2021) or non-parametric (Scott et al. 2011; Ding et al. 2019) regression models predicting the mean performance of each alternative. As a result, the learned model will be deployed to inform the scenario-dependent selection. Since the online budget typically is insignificant relative to offline simulation budget, there is little study considering online simulation. Nevertheless, the value of online scenario and budget may be consistently underestimated and overlooked, which motivates us to propose a unified offline-online learning paradigm and utilize the online information to make better scenario-dependent selection.

In this paper, we consider a continuous scenario space, which implies that it is impossible to exhaustively simulate all the scenarios even if the simulation is performed offline, and during online phase there will be many new scenarios we never encounter. In this case, the scenario-dependent selection will benefit greatly from the online budget and scenario information. During offline phase, we can model the mean performance of each alternative as a function of scenarios taking a nonparametric form, and model the randomness in performance as additive Gaussian noise. We deploy offline simulation to generate samples and learn the mean functions. Upon observing an online scenario, we run online simulation to obtain additional samples. Then, we retrain the model based on offline and online samples and accordingly update the mean functions. As a result, the learned functions can be deployed to select the best alternative conditional on the online scenario. We formalize these procedures in a so-called *unified offline-online learning* paradigm. Notice that the UOOL paradigm is dynamic, that is, we can keep it learning in practice by consecutively feeding new samples.

This paper makes the following contributions. First, we extend the scenario-dependent R&S model by considering online scenarios and budget, and bridge offline and online simulation by proposing a generic UOOL framework. Second, we place a Gaussian process (GP) prior over the mean of alternatives and model it as a function of scenarios using GP regression with a squared exponential covariance kernel, in which we are able to model our uncertainty about the mean function and the noise in alternative performance simultaneously. Third, we develop an online budget allocation policy to dynamically determine the scenarios and alternatives should be sampled by utilizing the information of online scenario. Given any online scenario, we show that it is better to sample the location where the online scenario lies at each time step. Finally, we demonstrate the superior performance of UOOL paradigm on four test functions in comparison with artificial neural networks and decision tree.

The rest of the paper is organized as follows. Section 2 introduces the problem formulation and the method of learning unknown mean performance. Then, we define the value of new information in Section

3 and propose our UOOL framework in Section 4. Section 5 provides numerical experiments on synthetic test functions. The final section offers conclusions.

2 PROBLEM FORMULATION

Suppose that there are K competing alternatives, and the performance Y_a of each alternative $a = 1,\ldots,K$ depends on the scenario $\boldsymbol{x} := (x_1,\ldots,x_d)^\top \in \mathscr{X} \subset \mathbb{R}^d$ characterized by a d-dimensional vector, where $\mathscr{X}$ is the collection of scenarios, and the total number of scenarios may be infinite. For each alternative, we assume that the conditional performance (i.e., $Y_a \mid \boldsymbol{x}$) at different scenarios follows a normal distribution

$$Y_a \mid \boldsymbol{x} \sim \mathscr{N}\left(\mu_a(\boldsymbol{x}), \sigma_a^2\right),\ a = 1,\ldots,K,$$

where the mean $\mu_a(\boldsymbol{x})$ and variance σ_a^2 are unknown but can be estimated via sampling. The scenario-dependent best alternative is defined as

$$a^*(\boldsymbol{x}) := \underset{a=1,\ldots,K}{\arg\max}\, \mu_a(\boldsymbol{x}).$$

Without loss of generality, we assume the number of offline observations for each alternative is equal, and define the offline dataset as

$$\mathscr{D}^0 = \{(\boldsymbol{x}_a^1, y_a^1),\ldots,(\boldsymbol{x}_a^{n_0}, y_a^{n_0}), a = 1,\ldots,K\}.$$

After an online scenario is revealed, we have a short time to run a small number of simulation replications. During the online phase, we need to make a sequence of sampling decisions

$$\{a^t, \boldsymbol{x}^t : t = 1,\ldots,T\},$$

which indicates that the t-th replication is allocated to some alternative $a^t \in \{1,\ldots,K\}$ and scenario $\boldsymbol{x}^t \in \mathscr{X}$. Denote y_a^t as the resulting observation for $(a^t, \boldsymbol{x}^t)$, drawn independently from the normal distribution

$$y_a^t \mid \boldsymbol{x}^t, a^t \sim \mathscr{N}\left(\mu_{a^t}(\boldsymbol{x}^t), \sigma_{a^t}^2\right).$$

Denote $\mathscr{D}^t$ as the information set including all the offline data and online observations collected up to time t, i.e., $\mathscr{D}^t = \mathscr{D}^{t-1} \cup \{(\boldsymbol{x}_a^t, y_a^t)\}$ for $t = 1,\ldots,T$, where $\boldsymbol{x}_a^t$ denotes the sampling decision $(a^t, \boldsymbol{x}^t)$ for notational convenience. Denote $\boldsymbol{x}_s \in \mathscr{X}$ as the online scenario. Then, we can sequentially learn the truth $\mu_a(\boldsymbol{x}_s)$ for $a = 1,\ldots,K$ at each time t based on the available samples $\mathscr{D}^t$, and estimate the best alternative $a^*(\boldsymbol{x}_s)$ accordingly.

2.1 Learn Unknown Mean Performance Via Gaussian Process

Taking the Bayesian viewpoint, we treat the unknown mean $\mu_a(\boldsymbol{x})$ as a random variable for each alternative, and assign a Gaussian process prior over the truth $\mu_a(\boldsymbol{x})$. Under the GP prior, every finite collection of $\{\mu_a(\boldsymbol{x}) : \boldsymbol{x} \in \mathscr{X}\}$ is a Gaussian process with mean function $\mu_a^0(\boldsymbol{x}) := \mathbb{E}[\mu_a(\boldsymbol{x})]$ and covariance function $k_a^0(\boldsymbol{x}, \boldsymbol{x}') := \mathrm{Cov}[\mu_a(\boldsymbol{x}), \mu_a(\boldsymbol{x}')]$, where $\boldsymbol{x}' \in \mathscr{X}$. Consequently, we express the GP priori over $\mu_a(\boldsymbol{x})$ as

$$\mu_a(\boldsymbol{x}) \sim \mathscr{GP}\left(\mu_a^0(\boldsymbol{x}), k_a^0(\boldsymbol{x}, \boldsymbol{x}')\right),\ a = 1,\ldots,K. \tag{1}$$

Let $\mathbf{X}_a^t$ denote the set of scenarios sampled up to time t for alternative a, i.e.,

$$\mathbf{X}_a^t := \{\boldsymbol{x}_a^1,\ldots,\boldsymbol{x}_a^{n_0}\} \cup \{\boldsymbol{x}^\ell : \boldsymbol{x}^\ell = a, \ell = 1,\ldots,t\},$$

and define $\mathbf{y}_a^t := \{y_a^1,\ldots,y_a^{n_0}\} \cup \{y^\ell : \boldsymbol{x}^\ell = a, \ell = 1,\ldots,t\}$ likewise. With slight abuse of notation, we treat $\mathbf{X}_a^t$ as a matrix wherein each row represents a scenario and arranged in the order of appearance, and $\mathbf{y}_a^t$ as the corresponding column vector. Then, we define the estimated mean at time t as

$$\mu_a^t(\boldsymbol{x}) := \mathbb{E}\left[\mu_a(\boldsymbol{x}) | \mathscr{D}^t\right],\ a = 1,\ldots,K.$$

For ease of notation, we use $\mathbb{E}^t[\cdot] = \mathbb{E}[\cdot|\mathscr{D}^t]$ and define $\mathrm{Cov}^t[\cdot]$ likewise. Consequently, the covariance function at time t can be expressed as

$$k_a^t(\boldsymbol{x},\boldsymbol{x}') := \mathrm{Cov}^t[\mu_a(\boldsymbol{x}),\mu_a(\boldsymbol{x}')]. \tag{2}$$

The covariance matrix at time t can be defined as $\mathbf{K}_a^t = \mathrm{Cov}^t[\mathbf{X}_a^t]$ with entries calculated via (2). A convention is to place a zero-mean GP prior over the function value (Williams and Rasmussen 2006), and the covariance function in essence is the crucial ingredient in a GP predictor, because it encodes our assumptions about the function which we wish to learn.

The fundamental assumption under a GP is that the close locations (i.e., $\boldsymbol{x}$) are likely to have similar outputs, and sampled scenarios should be informative about the prediction at new scenarios. In fact, the covariance function defines the nearness or similarity, and we adopt the squared exponential (SE) kernel that is infinitely differentiable, expressed by

$$k(\boldsymbol{x},\boldsymbol{x}') = \beta^2 \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{x}')^\top \mathbf{L}(\boldsymbol{x}-\boldsymbol{x}')\right), \tag{3}$$

where $\mathbf{L} = l^{-2}\mathbb{I}$ with $\mathbb{I}$ being an identify matrix, β is referred to as the signal amplitude controlling the uncertainty of our belief about the function $\mu_a(\boldsymbol{x})$, and l is the length scale representing how smooth $\mu_a^0(\boldsymbol{x})$ at each dimension of $\boldsymbol{x}$. Without loss of generality, the GP prior for all alternatives is the SE kernel (3).

We can see that the covariance of two variables defined by (3) increases as the distance between them decreases, thus resulting in much larger correlation. Moreover, the covariance function by definition is positive, i.e., $k(\boldsymbol{x},\boldsymbol{x}') > 0, \forall \boldsymbol{x},\boldsymbol{x}' \in \mathscr{X}$, implying that any scenario is positively correlated with other scenarios. In addition, the covariance function can be parametrized by hyper-parameters β and l. Typically, we have only rather vague information about β and l. Thus in order to learn the truth $\mu_a(\boldsymbol{x})$ efficiently, it is also essential to learn the hyper-parameters β and l for each alternative. Note that although the functional form of kernel for all alternatives is same, the hyper-parameters are trained individually and thus are different.

In fact, we are learning parameters $\mu_a(\boldsymbol{x}_s)$ and $\boldsymbol{\theta}_a := (\beta^2, l^2, \sigma_a^2)^\top$ simultaneously for $a = 1,\ldots,K$ based on offline and online simulation data. At the end of online simulation, the estimated best for online scenario $\boldsymbol{x}_s \in \mathscr{X}$ is given by

$$\hat{a}(\boldsymbol{x}_s) := \underset{a=1,\ldots,K}{\arg\max}\, \mu_a^T(\boldsymbol{x}_s), \tag{4}$$

which means that we will select the alternative that appears to be the best.

2.2 Updating Equations

After the first t online sampling decisions, we obtain the training observations $\{\mathbf{X}_a^t, \mathbf{y}_a^t\}$ for each alternative a. Under the GP prior (1), the observed outputs $\mathbf{y}_a^t$ and the function value $\mu_a^t(\boldsymbol{x})$ are jointly Gaussian distributed as

$$\begin{bmatrix} \mathbf{y}_a^t \\ \mu_a(\boldsymbol{x}) \end{bmatrix} \sim \mathscr{N}\left(\begin{bmatrix} \boldsymbol{\mu}_a^0 \\ \mu_a^0(\boldsymbol{x}) \end{bmatrix}, \begin{bmatrix} \mathbf{K}_a^0 + \sigma_a^2 \mathbf{I} & \mathbf{k}_a \\ \mathbf{k}_a^\top & k_a^0(\boldsymbol{x},\boldsymbol{x}') \end{bmatrix} \right), \quad \forall \boldsymbol{x} \in \mathscr{X},\ a = 1,\ldots,K, \tag{5}$$

where $\mathbf{K}_a^0$ is the covariance matrix with entries calculated by $k_a^0(\boldsymbol{x},\boldsymbol{x}')$, and $\mathbf{I}$ is the identity matrix, $\mathbf{k}_a$ is a covariance vector between the test scenario x and training scenarios $\mathbf{X}_a^t$ calculated by $k_a^0(\boldsymbol{x},\boldsymbol{x}')$ with $\boldsymbol{x}' \in \mathbf{X}_a^t$. Then, the conditional distribution of $\mu_a^t(\boldsymbol{x})$ can be explicitly derived as

$$\mu_a(\boldsymbol{x}) \mid \mathbf{X}_a^t, \mathbf{y}_a^t, \theta_a, \sigma_a \ \sim\ \mathscr{N}\left(\mu_a^t(\boldsymbol{x}), k_a^t(\boldsymbol{x},\boldsymbol{x})\right),\ a = 1,\ldots,K, \tag{6}$$

where

$$\mu_a^t(\boldsymbol{x}) = \mu_a^0(\boldsymbol{x}) + \mathbf{k}_a^\top \left[\mathbf{K}_a^0 + \sigma_a^2 \mathbf{I}\right]^{-1} (\mathbf{y}_a^t - \boldsymbol{\mu}_a^0) \tag{7}$$

$$k_a^t(\boldsymbol{x},\boldsymbol{x}') = k_a^0(\boldsymbol{x},\boldsymbol{x}') - \mathbf{k}_a^\top \left[\mathbf{K}_a^0 + \sigma_a^2 \mathbf{I}\right]^{-1} \mathbf{k}_a. \tag{8}$$

The readers can refer to (Williams and Rasmussen 2006) for more details on the computation of (6)-(8).

Equation (7) implies that the mean prediction is a linear combination of observations $\mathbf{y}_a^t$, while the variance $k_a^t(\boldsymbol{x},\boldsymbol{x})$ in (8) depends on the inputs $\mathbf{X}_a^t$ instead of $\mathbf{y}_a^t$, because $k_a^t(\boldsymbol{x},\boldsymbol{x})$ models our uncertainty about the mean $\mu_a(\boldsymbol{x})$ rather than the output $Y_a(\boldsymbol{x})$. As the second term in the right-hand side of (8) representing the information the observations give about $\mu_a(\boldsymbol{x})$, is positive, the posterior variance in (8) is thus smaller than the prior, that is, the uncertainty on the unknown function $\mu_a(\boldsymbol{x})$ is reduced. As a result, we can plug in the online scenario $\boldsymbol{x}_s$ and dynamically learn the truth $\{\mu_a(\boldsymbol{x}_s)\}_{a=1}^K$ via (7)-(8).

We are now in a position to compute the predictive distribution of next observation y^{t+1}, as the distribution of y^{t+1} represents our belief about the next observation given the sampling decision $(a^t,\boldsymbol{x}^t)$. Under the GP model the prior on the unknown function is normal, and the performance of each alternative is assumed to be normal. As a result, the conditional distribution of y^{t+1} is also normally distributed as

$$y^{t+1} \mid a^t,\boldsymbol{x}^t,\mathscr{D}^t \sim \mathcal{N}\left(\mu_{a^t}^t(\boldsymbol{x}^t), k_{a^t}^t(\boldsymbol{x}^t,\boldsymbol{x}^t)+\sigma_{a^t}^2\right). \tag{9}$$

After we have made the sampling decision $(a^t,\boldsymbol{x}^t)$ but before the next observation y^{t+1} is available, $\mu_a^{t+1}(\boldsymbol{x})$ is also normally distributed. Then, we refer to (Frazier et al. 2009; Scott et al. 2011) to derive the recursions which express $\mu_a^{t+1}(\boldsymbol{x})$ and $k_a^{t+1}(\boldsymbol{x},\boldsymbol{x}')$ as functions of $\mu_a^t(\boldsymbol{x})$, $k_a^t(\boldsymbol{x},\boldsymbol{x}')$, a^t, $\boldsymbol{x}^t$, and y^t, given by

$$\mu_a^{t+1}(\boldsymbol{x}) = \mu_a^t(\boldsymbol{x}) + \tilde{\sigma}(\boldsymbol{x},\boldsymbol{x}^t,a^t)\frac{y^{t+1}-\mu_{a^t}^t(\boldsymbol{x})}{\sqrt{\sigma_{a^t}^2+k_{a^t}^t(x^t,\boldsymbol{x}^t)}}\cdot \mathbb{1}\{a^t=a\} \tag{10}$$

$$k_a^{t+1}(\boldsymbol{x},\boldsymbol{x}') = k_a^t(\boldsymbol{x},\boldsymbol{x}') - \tilde{\sigma}(\boldsymbol{x},\boldsymbol{x}^t,a^t)\tilde{\sigma}^t(\boldsymbol{x}',\boldsymbol{x}^t,a^t)\cdot \mathbb{1}\{a^t=a\}, \tag{11}$$

where $\mathbb{1}\{\cdot\}$ is an indicator function, and $\tilde{\sigma}^t$ is a function defined as

$$\tilde{\sigma}^t(\boldsymbol{x},\boldsymbol{x}',a) = \frac{k_a^t(\boldsymbol{x},\boldsymbol{x}')}{\sqrt{\sigma_a^2+k_a^t(\boldsymbol{x}',\boldsymbol{x}')}}. \tag{12}$$

Define $Z^{t+1} = \dfrac{y^{t+1}-\mu_{a^t}^t(\boldsymbol{x})}{\sqrt{\sigma_{a^t}^2+k_{a^t}^t(\boldsymbol{x}^t,\boldsymbol{x}^t)}}$. It can be shown that $Z^{t+1}\sim\mathcal{N}(0,1)$ by (9). Then, we substitute the complex term in (7) with a standard normal variable Z^{t+1} and rewrite (7) as

$$\mu_a^{t+1}(\boldsymbol{x}) = \mu_a^t(\boldsymbol{x}) + \tilde{\sigma}^t(\boldsymbol{x},\boldsymbol{x}^t,a^t)Z^{t+1}\cdot \mathbb{1}\{a^t=a\}. \tag{13}$$

3 VALUE OF INFORMATION

One crucial class of the approximate sampling policies for sequential learning problems is developed based on the concept of expected improvement criterion (Jones et al. 1998), in which the value of information is measured by the expected single-period improvement in the objective value for some sampling decision before the next observation occurs. Although we do not know exactly how a new observation y^{t+1} will change our beliefs about the unknown functions $\{\mu_a(\boldsymbol{x})\}_{a=1}^K$, we can compute the expected difference over the predictive distribution of y^{t+1} in (9), and this quantity is referred to as knowledge gradient in (Williams and Rasmussen 2006). Given a sampling decision $(\boldsymbol{x}^t,a^t)$ at time t, the value of new information for our problem is defined as

$$v^t(\boldsymbol{x},a;\boldsymbol{x}_s) := \mathbb{E}^t\left[\max_{i=1,\dots,K}\mu_i^{t+1}(\boldsymbol{x}_s)\;\middle|\;\boldsymbol{x}^t=\boldsymbol{x},a^t=a\right] - \max_{i=1,\dots,K}\mu_i^t(\boldsymbol{x}_s), \tag{14}$$

where the conditional expectation is taken with respect to the distribution of our belief on y^{t+1}.

Note that the predictive distribution of $\mu_a^{t+1}(\boldsymbol{x})$ is characterised by (10)-(13). We can rewrite (14) as

$$v^t(\boldsymbol{x},a;\boldsymbol{x}_s) = \mathbb{E}^t\left[\max_{i=1,\ldots,K}\left(\mu_i^t(\boldsymbol{x}_s) + \tilde{\sigma}^t(\boldsymbol{x}_s,\boldsymbol{x}^t,a^t)Z^{t+1}\cdot\mathbb{1}\{a^t=i\}\right) \,\middle|\, \boldsymbol{x}^t=\boldsymbol{x}, a^t=a\right] - \max_{i=1,\ldots,K}\mu_i^t(\boldsymbol{x}_s).$$

Then, the policy in (15) chooses the sampling decision at time t by maximizing the expected value of information:

$$(\boldsymbol{x}^t,a^t) \in \mathop{\arg\max}_{\boldsymbol{x}\in\mathscr{X},\ a=1,\ldots,K} v^t(\boldsymbol{x},a;\boldsymbol{x}_s). \tag{15}$$

Define a function $\zeta^t : \mathbb{R}^d \times \mathbb{R} \mapsto (-\infty,0]$ as

$$\zeta^t(\boldsymbol{x},a;\boldsymbol{x}_s) := -\left|\frac{\mu_a^t(\boldsymbol{x}_s) - \max_{i\neq a}\mu_i^t(\boldsymbol{x}_s)}{\tilde{\sigma}^t(\boldsymbol{x}_s,\boldsymbol{x},a)}\right|, \tag{16}$$

and then define $\Psi : \mathbb{R} \mapsto \mathbb{R}$ as

$$\Psi(\zeta) := \zeta\Phi(\zeta) + \phi(\zeta),$$

where $\Phi(\cdot)$ is the normal cumulative distribution function, and $\phi(\cdot)$ is the normal probability density. As a result, given $\boldsymbol{x}_s \in \mathscr{X}$, we can further write (14) as

$$v^t(\boldsymbol{x},a;\boldsymbol{x}_s) = \tilde{\sigma}^t(\boldsymbol{x}_s,\boldsymbol{x},a)\Psi\left(\zeta^t(\boldsymbol{x},a;\boldsymbol{x}_s)\right). \tag{17}$$

Theorem 1 For any $\boldsymbol{x}_s \in \mathscr{X}$ and $a \in \{1,\ldots,K\}$, we have

$$\left.\frac{\partial}{\partial \boldsymbol{x}} v^t(\boldsymbol{x},a;\boldsymbol{x}_s)\right|_{\boldsymbol{x}=\boldsymbol{x}_s} = \boldsymbol{0},\ \forall\ \boldsymbol{x}_s \in \mathscr{X},\ a = 1\ldots,K, \tag{18}$$

and the solution to $\frac{\partial}{\partial \boldsymbol{x}} v^t(\boldsymbol{x},a;\boldsymbol{x}_s) = \boldsymbol{0}$ is unique.

Theorem 1 is fundamental to computing the sampling decision $(\boldsymbol{x}^t,a^t)$, because it implies that the online budget should be always allocated to scenario $\boldsymbol{x}_s$ no matter which alternative is sampled. As a result, we only need to solve the following problem

$$a^t \in \mathop{\arg\max}_{a=1,\ldots,K} \tilde{\sigma}^t(\boldsymbol{x}_s,\boldsymbol{x}_s,a)\Psi(\zeta^t(\boldsymbol{x}_s,a,\boldsymbol{x}_s)). \tag{19}$$

4 UNIFIED OFFLINE-ONLINE LEARNING PARADIGM

In this section, we formally present the unified offline-online learning paradigm via simulation, which provides an elegant way to simultaneously utilize the offline-online simulation data and online scenario information, to learn the best decision for the revealed online scenario. The underlying belief in UOOL paradigm is that the mean performances at different locations $\boldsymbol{x} \in \mathscr{X}$ have correlations, and thus historical data can provide information on inferring the mean performance at new scenario. It is known that GPs in fact work mainly depending on the correlation (i.e., covariance kernels). Therefore, we propose deploying a GP to model the true performance means of alternatives, and the unknown variances are learned by maximizing the log-likelihood under GP prior. To generate more informative online simulation data, we should measure the location where the online scenario lies in. Then, we deploy the offline and online data to train GP models, and leverage the trained GP models to learn the mean $\mu_a(\boldsymbol{x}_s)$ at online scenario $\boldsymbol{x}_s$. Based on above generic ideas, we develop an Algorithm 1 containing all procedures explicitly.

First note that we do not focus on offline simulation budget allocation, but think of offline data as a fixed setting, thus generating offline data $\mathscr{D}^0$ randomly. Then, $\boldsymbol{\theta}_a$ is updated by maximizing the marginal

Algorithm 1 Unified Offline-Online Learning Paradigm via Simulation

```
Input: online scenario $\boldsymbol{x}_s \in \mathscr{X}$, covariance kernel $k(\cdot,\cdot;\beta,l)$, number of alternative $K$, number of offline samples $n_0$, online budget $T$, parameters $\boldsymbol{\theta}_a$, $t \leftarrow 0$.
Randomly sample $n_0$ locations for each alternative $a$ and run simulation to observe corresponding outputs, which constitutes the offline data $\mathscr{D}^0 = \{(\boldsymbol{x}_a^1, y_a^1), \ldots, (\boldsymbol{x}_a^{n_0}, y_a^{n_0}), a = 1, \ldots, K\}$.
Initialize $\{\boldsymbol{\theta}_a\}_{a=1}^K$ based on $\mathscr{D}^0$ by maximizing the log marginal likelihood in (20).
Update the mean function $\mu_a^t(\boldsymbol{x})$ and covariance function $k_a^t(\boldsymbol{x}',\boldsymbol{x};\boldsymbol{\theta}_a)$ for $a = 1,\ldots,K$ via (7), (8), (12) and (16).
Increase $t \leftarrow t+1$.
if $t \le T$ then
    Obtain the sampling decision $(\boldsymbol{x}^t, a^t)$ by setting $\boldsymbol{x}^t = \boldsymbol{x}_s$ and solving (19) to get $a^t$.
    Run one simulation replication at location $\boldsymbol{x}^t$ for alternative $a^t$, and get new observation $(\boldsymbol{x}_a^t, y_a^t)$.
    Update information set $\mathscr{D}^t$, $\mathbf{y}_a^t$, $\mathbf{X}_a^t$, and then update $\boldsymbol{\theta}_{a^t}$ by solving (20).
    Update $\mu_a^t$, $k_a^t$, $\tilde{\sigma}^t$ and $\zeta^t$ via (7), (8), (12) and (16).
    Increase $t \leftarrow t+1$.
end if
Return the best design $\hat{a}(\boldsymbol{x}_s)$ by solving (2).
```

likelihood (or evidence) $p(\mathbf{y}_a^t \mid \mathbf{X}_a^t, \boldsymbol{\theta}_a)$ which is the integral of the likelihood times the GP prior. By observing that $\mathbf{y}_a^t \sim \mathscr{N}\left(\boldsymbol{\mu}_a^0, \mathbf{K}_a^0 + \sigma_a^2 \mathbf{I}\right)$ from (5), we can directly obtain the log marginal likelihood

$$\log\ p\left(\mathbf{y}_a^t \mid \mathbf{X}_a^t, \boldsymbol{\theta}_a\right) = -\frac{1}{2}\mathbf{y}_a^{t\top}\left(\mathbf{K}_a^0 + \sigma_a^2\mathbf{I}\right)^{-1}\mathbf{y}_a^t - \frac{1}{2}\log\det\left(\mathbf{K}_a^0 + \sigma_a^2\mathbf{I}\right) - \frac{|\mathbf{y}_a^t|}{2}\log 2\pi, \tag{20}$$

where $\det(\cdot)$ stands for the determinant of a matrix, and $|\cdot|$ returns the number of elements in a vector. Now it is evident that Algorithm 1 in fact is an adaptive learning framework, because $\boldsymbol{\theta}_a$ is updated dynamically.

5 NUMERICAL EXPERIMENTS

In this section, we assess the performance of UOOL paradigm on several test functions by comparing it with artificial neural networks (ANNs) and decision tree. We consider a two-layer ANN with each layer of five nodes, and the training of both ANNs and decision tree is performed by using MATLAB built-in functions with default settings.

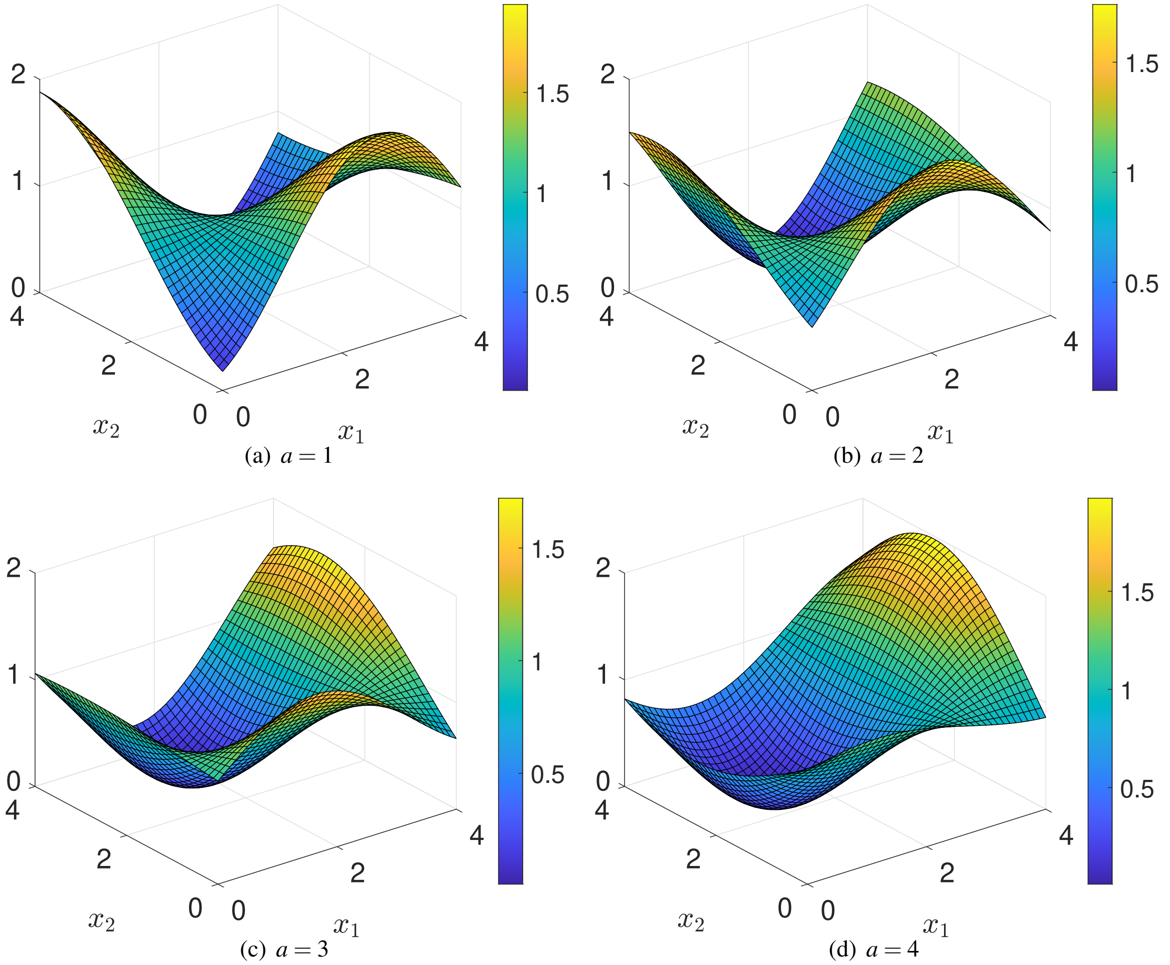
5.1 Experiment Settings

We choose the Griewank function which has many widespread local minima, and modify it by adding an additional term $a/2$ at each dimension of $\boldsymbol{x}$, i.e.,

$$\mu_a(\boldsymbol{x}) = \sum_{i=1}^{d}\frac{(x_i + a/2)^2}{4000} - \prod_{i=1}^{d}\cos\left(\frac{x_i + a/2}{\sqrt{i}}\right) + 1,\ a = 1,\ldots,K. \tag{21}$$

To better understand the modified Griewank function, we plot the surface of $\mu_a(\boldsymbol{x})$ for $a = 1,\ldots,K$ and $d = 2$ in Figures 2-3. We can see that the best alternative $a^*(\boldsymbol{x})$ is not universal but changes with scenarios. In this paper, we consider $d = 1,2,3,4$. The modified function (21) is deployed as the true mean of alternatives, and a normally distributed observation noise with a mean of zero and standard deviation of one is imposed on the function (21), i.e., $\sigma_a^2 = 1, a = 1,\ldots,K$.

For numerical experiments, we consider the scenario space $\mathscr{X} = [0,4]^d$ and $K = 8$. The offline data for each alternative is generated by randomly sampling $n_0 = 3000$ locations from $\mathscr{X}$ and obtaining corresponding observations. For performance evaluation, we randomly generate $N_s = 40$ online scenarios.



(a) $a = 1$

(b) $a = 2$

(c) $a = 3$

(d) $a = 4$

Figure 2: Surfaces of modified Griewank functions $a = 1, 2, 3, 4$.

In order to achieve fair comparison, all algorithms used the same offline data and online scenarios. For performance evaluation, we define the empirical opportunity cost as

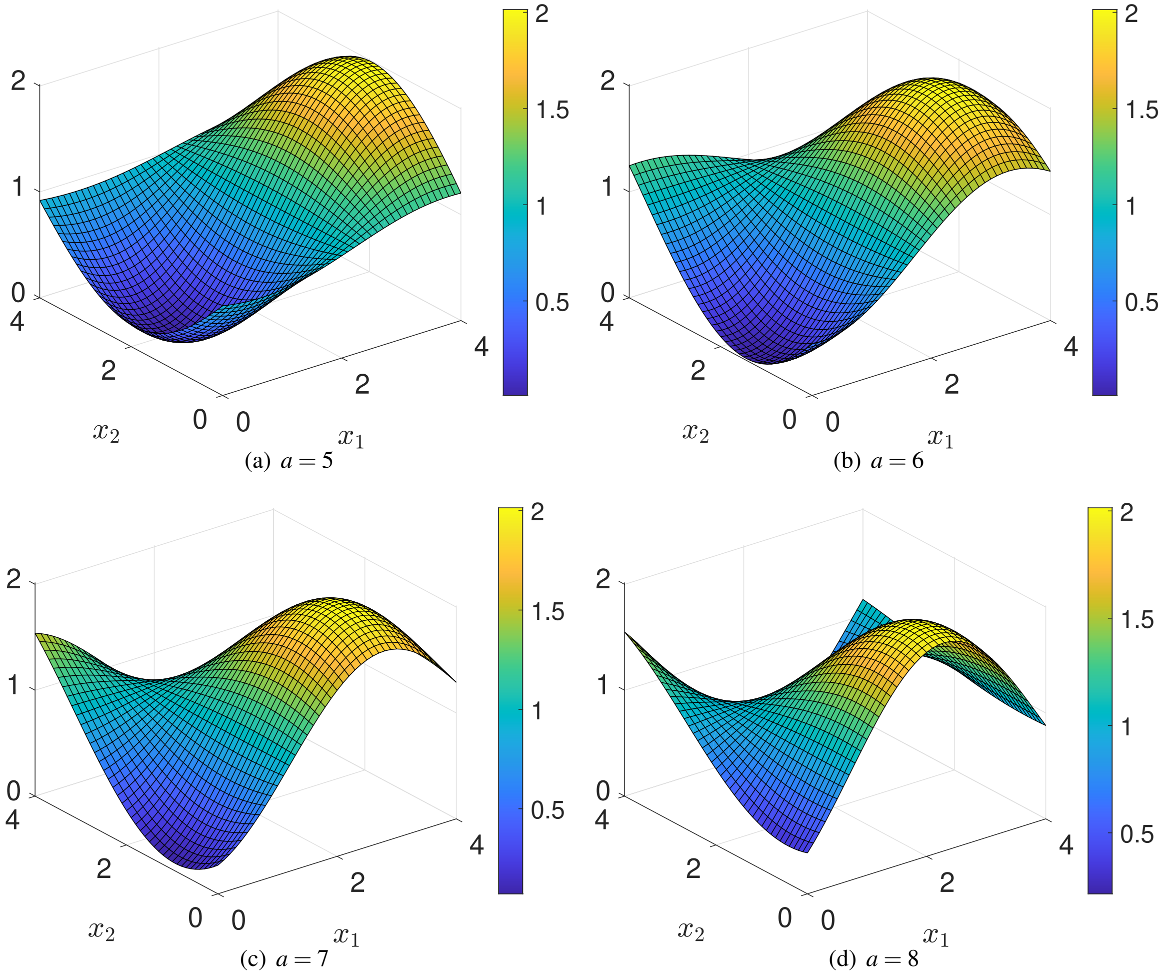
$$\mathrm{Eoc}(\boldsymbol{x}_s) = {N_r}^{-1} \sum_{r=1}^{N_r} \left(\mu_{a^*(\boldsymbol{x}_s)}(\boldsymbol{x}_s) - \mu_{\hat{a}(\boldsymbol{x}_s;\omega_r)}(\boldsymbol{x}_s) \right), \tag{22}$$

and the average opportunity cost by averaging over scenarios as

$$\mathrm{Aoc} = {N_s}^{-1} \sum_{s=1}^{N_s} \mathrm{Eoc}(\boldsymbol{x}_s), \tag{23}$$

where $\hat{a}(\boldsymbol{x}_s;\omega_r) \in \arg\max_a \mu_a^T(\boldsymbol{x}_s;\omega_r)$ denotes the estimated best alternative under sample path ω_r in replication r, and N_r and N_s denote the number of replications and online scenarios, respectively. Note that we calculate (23) in such a way that given each online scenario we independently run N_r replications for every algorithm, rather than we choose N_r scenarios after each simulation replication is finished. There are substantial differences between the two ways. As a by-product of $\mathrm{Eoc}(\boldsymbol{x}_s)$, we define

$$\mathrm{Eoc}^{max} := \max_{1 \le s \le N_s} \mathrm{Eoc}(\boldsymbol{x}_s)$$



(a) $a = 5$

(b) $a = 6$

(c) $a = 7$

(d) $a = 8$

Figure 3: Surfaces of modified Griewank functions for $a = 5, 6, 7, 8$.

and

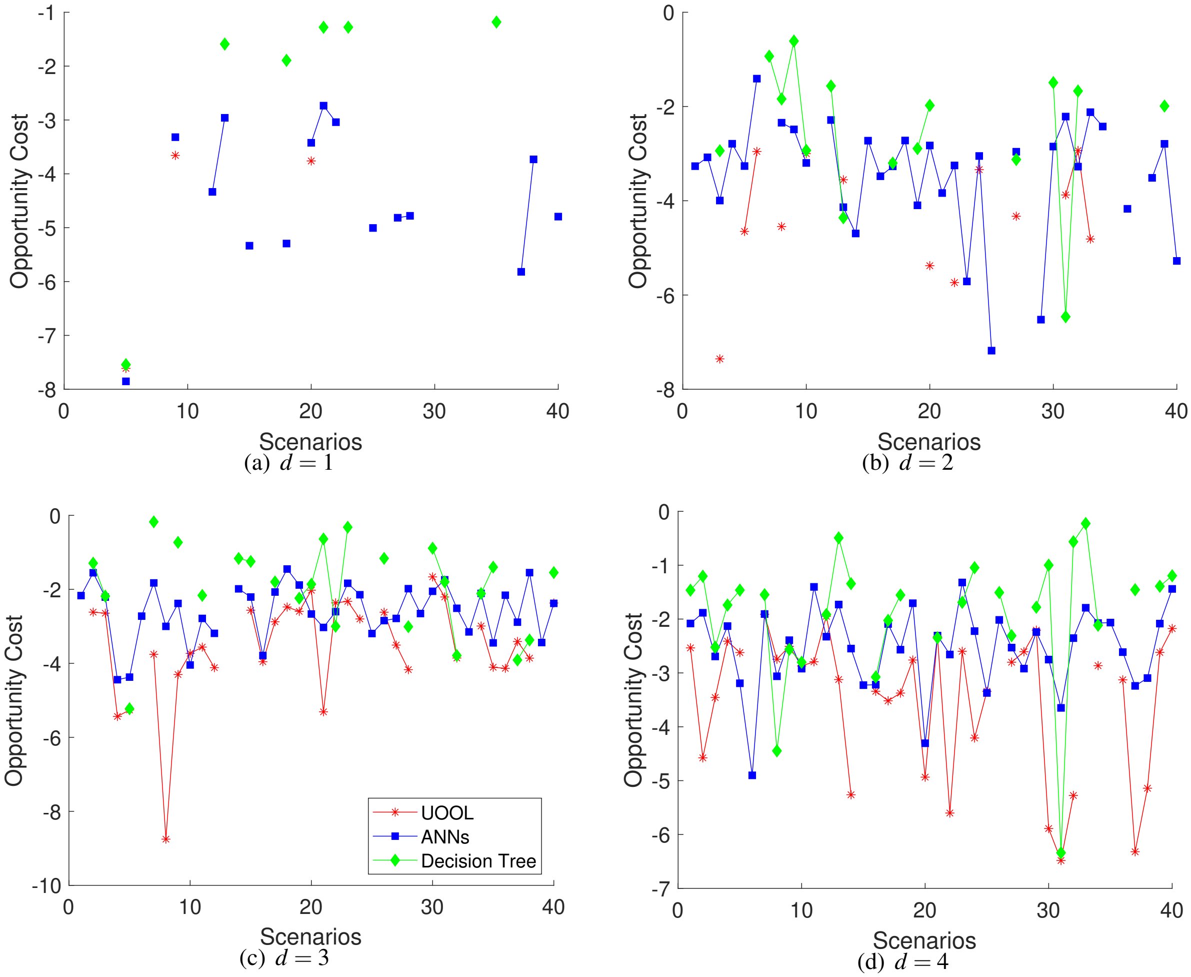
$$\text{Eoc}^{min} := \min_{1 \le s \le N_s} \text{Eoc}(\boldsymbol{x}_s),$$

where Eoc^{max} and Eoc^{min} respectively represent the worst and best scenario cases, and we can assess the robustness of algorithms from the measure Eoc^{max}. Throughout the paper, we set $N_r = 30$ and $T = 100$.

5.2 Numerical Results

As the empirical opportunity costs obtained by the three algorithms are near or equal to zero, to better observe the differences among them we take the natural logarithm of Eoc. Then, we plot the logarithm of empirical opportunity cost over 40 scenarios in Figure 4. Note that when the empirical opportunity cost is equal to zero, the log of zero is infinity, resulting in missing points and disconnected lines in Figure 4. Therefore, less points and smaller values indicate a better performance.

We can see that the UOOL algorithm achieves overall better performance compared to ANNs and decision tree over the four test functions. As the dimension of scenarios increases, the problem of finding the scenario-dependent best alternative also gradually becomes more difficult. When the dimension of Griewank function is equal to one or two, our algorithm can find the best alternative for most of scenarios at each simulation replication. In addition, Figure 4 shows that the performance of algorithms is significantly affected by online scenarios, which validates the necessity of considering online scenario and budget for dealing with the online decision-making problems.



(a) $d = 1$

(b) $d = 2$

(c) $d = 3$

(d) $d = 4$

Figure 4: Empirical opportunity cost against scenarios.

In addition, we summarize the numerical results over the 40 scenarios in Table 1. We can explicitly observe that UOOL algorithm has achieved superior performance in terms of the average opportunity cost and worst-case Eoc, because the Aoc and Eoc^{max} values obtained by UOOL algorithm are smaller than that achieved by ANNs and decision tree. Moreover, the three algorithms are able to find the best alternative for some scenarios due to the fact that Eoc$^{min} = 0$.

Table 1: Opportunity cost obtained by different algorithms.

		UOOL	ANNs	Decision Tree			UOOL	ANNs	Decision Tree
Griewank	Aoc	0.0012	0.0079	0.0305	Griewank	Aoc	0.0083	0.0440	0.0563
$d = 1$	Eoc^{max}	0.0258	0.0649	0.3068	$d = 2$	Eoc^{max}	0.0532	0.2444	0.5426
	Eoc^{min}	0	0	0		Eoc^{min}	0	0	0
Griewank	Aoc	0.0407	0.0915	0.1439	Griewank	Aoc	0.0458	0.1000	0.1638
$d = 3$	Eoc^{max}	0.1896	0.2344	0.8387	$d = 4$	Eoc^{max}	0.1478	0.2669	0.7977
	Eoc^{min}	0	0	0		Eoc^{min}	0	0	0

6 CONCLUSIONS

The era of big data has created new opportunities for simulation-based online decision making. Instead of performing offline simulation to only estimate the static system performance measures, nowadays we can store all the offline simulation data with a low cost and perform a number of online simulation replications efficiently, and then utilize the offline and online simulation data jointly for informing the online decision making. This work is the first step toward unifying offline and online simulation to address online decision-making problems. In comparison with the well-known ANNs and decision tree in machine learning, our UOOL algorithm achieves superior performance.

ACKNOWLEDGEMENTS

This paper is supported by the Centre of Excellence in Modelling and Simulation for Next Generation Ports (C4NGP).

REFERENCES

Brantley, M. W., L. H. Lee, C.-H. Chen, and A. Chen. 2013. “Efficient simulation budget allocation with regression”. *IISE Transactions* 45(3):291–308.

Ding, L., L. J. Hong, H. Shen, and X. Zhang. 2019. “Knowledge gradient for selection with covariates: Consistency and computation”. *arXiv preprint arXiv:1906.05098*.

Frazier, P., W. Powell, and S. Dayanik. 2009. “The knowledge-gradient policy for correlated normal beliefs”. *INFORMS Journal on Computing* 21(4):599–613.

Gao, S., J. Du, and C.-H. Chen. 2019. “Selecting the optimal system design under covariates”. In *2019 IEEE 15th International Conference on Automation Science and Engineering*, 547–552. Institute of Electrical and Electronics Engineers, Inc.

Hong, L. J., and G. Jiang. 2019. “Offline simulation online application: A new framework of simulation-based decision making”. *Asia-Pacific Journal of Operational Research* 36(06):1940015.

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.

Li, H., H. Lam, Z. Liang, and Y. Peng. 2020. “Context-dependent ranking and selection under a bayesian framework”. In *2020 Winter Simulation Conference*, 2060–2070: Institute of Electrical and Electronics Engineers, Inc.

Pedrielli, G., K. Selcuk Candan, X. Chen, L. Mathesen, A. Inanalouganji, J. Xu, C.-H. Chen, and L. H. Lee. 2019. “Generalized Ordinal Learning Framework (GOLF) for Decision Making with Future Simulated Data”. *Asia-Pacific Journal of Operational Research* 36(06):1940011.

Scott, W., P. Frazier, and W. Powell. 2011. “The correlated knowledge gradient for simulation optimization of continuous parameters using gaussian process regression”. *SIAM Journal on Optimization* 21(3):996–1026.

Shen, H., L. J. Hong, and X. Zhang. 2021. “Ranking and selection with covariates for personalized decision making”. *INFORMS Journal on Computing*.

Williams, C. K., and C. E. Rasmussen. 2006. *Gaussian processes for machine learning*. Cambridge: Massachusetts Institute of Technology press.

Xiao, H., L. H. Lee, and C.-H. Chen. 2015. “Optimal budget allocation rule for simulation optimization using quadratic regression in partitioned domains”. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 45(7):1047–1062.

Xiao, H., L. H. Lee, D. Morrice, C.-H. Chen, and X. Hu. 2021. “Ranking and selection for terminating simulation under sequential sampling”. *IISE Transactions* 53(7):735–750.

AUTHOR BIOGRAPHIES

HAITAO LIU is a Ph.D. candidate in the Department of Industrial Systems Engineering and Management at National University of Singapore. His research interests include simulation optimization and statistical learning. His email address is haitao_liu@u.nus.edu.

XIAO JIN is a Research Fellow in the Centre of Excellence in Modelling and Simulation for Next Generation Ports (C4NGP). He received his Ph.D. degree in Department of Industrial Systems Engineering and Management from National University of Singapore. His research interest includes simulation optimization and artificial intelligence with applications on logistics and maritime studies. His email address is isejinx@nus.edu.sg.

HAOBIN LI is a Senior Lecturer in the Department of Industrial Systems Engineering and Management at National University of Singapore. He received his Ph.D. degree in Department of Industrial Systems Engineering and Management from National University of Singapore. His research interests are in operations research, simulation optimization and designing high performance optimization tools with application on logistics and maritime studies. He is the designer of O2DES.Net framework for simulation integrated optimization. His email address is li_haobin@nus.edu.sg.

LOO HAY LEE is currently Professor in the Department of Industrial Systems Engineering and Management at National University of Singapore. He received his Ph.D. in engineering science from Harvard University, USA. His research interests include logistics, vehicle routing, supply chain modeling, and simulation-based optimization. His email address is iseleelh@nus.edu.sg.

EK PENG CHEW is currently Professor in the Department of Industrial Systems Engineering and Management at National University of Singapore. He received his Ph.D. in Industrial Engineering from Georgia Institute of Technology, USA. His current research areas are in port logistics and maritime transportation, simulation optimization and inventory management. His email address is isecep@nus.edu.sg.