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

CONTEXTUAL RANKING AND SELECTION WITH GAUSSIAN PROCESSES

Sait Cakmak
Enlu Zhou

H. Milton Stewart School of
Industrial & Systems Engineering
Georgia Institute of Technology
755 Ferst Drive NW
Atlanta, GA 30332, USA

Siyang Gao

Department of Systems Engineering
and Engineering Management
City University of Hong Kong
83 Tat Chee Avenue
Kowloon, Hong Kong, CHINA

ABSTRACT

In many real world problems, we are faced with the problem of selecting the best among a finite number of alternatives, where the best alternative is determined based on context specific information. In this work, we study the contextual Ranking and Selection problem under a finite arm - finite context setting, where we aim to find the best alternative for each context. We use a separate Gaussian process to model the reward for each arm, derive the large deviations rate function for both the expected and worst-case contextual probability of correct selection, and propose an iterative algorithm for maximizing the rate function. Numerical experiments show that our algorithm is highly competitive in terms of sampling efficiency, while having significantly smaller computational overhead.

1 INTRODUCTION

Ranking & Selection (R&S) studies the problem of identifying the best among a finite number of alternatives (arms), where the true performance of each alternative is only observed through noisy evaluations. The settings of R&S can be typically categorized into fixed confidence and fixed budget. In the fixed-confidence setting, the goal is to achieve a target probability of correct selection (*PCS*) of the best alternative using as few evaluations as possible, while in the fixed-budget setting one aims to achieve a *PCS* as high as possible with the given sampling budget. The R&S problem has been studied extensively over past few decades, and we refer the reader to Kim and Nelson (2007) and Chen et al. (2015) for an overview.

In certain applications, the best alternative may not be the same across the board, and may depend on the underlying *context*. The benefit of making context-dependent decisions is easily seen by a simple application of Jensen's inequality: $\mathbb{E}_c[\max_k f(k;c)] \geq \max_k \mathbb{E}_c[f(k;c)]$, where $f(k;c)$ represents the reward of selecting alternative k for the context c, and $\mathbb{E}_c[\cdot]$ denotes the expectation with respect to (w.r.t.) c. Examples of context-dependent decision making include personalized medicine, where the best drug and dose may depend on the patient's age, gender, and medical history; and recommender systems, where personalized decisions have been the focus of study for over a decade (Nunes and Hu 2012). Context-dependent decision making also arises in R&S. For example, based on a set of forecasted market conditions (contexts), we can identify a set of alternative configurations (arms) of a complex manufacturing system, which can be simulated under any given context to determine the most profitable configuration to use when the market conditions are realized.

In this work, we study the contextual R&S problem, in which the rewards are a function of the context. Our goal is to identify the best alternative for each context under a fixed budget. Much like the classical R&S problem, at each iteration, the decision maker selects an arm-context pair to evaluate and observes a noisy evaluation of the true reward. With a finite sampling budget and noisy observations, it is not

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

possible to identify the best arm with certainty, and we need to design a sampling policy, which takes in the current estimate of rewards and outputs the next arm-context pair to sample, in order to achieve the highest possible "aggregated" *PCS* when the budget is exhausted. The aggregation is needed because in the contextual R&S, for any sampling policy, *PCS* is also context dependent, i.e., for each context c there is a $PCS(c)$. This defines multiple objectives to consider when designing the sampling policy. In this work, we consider two approaches to aggregate $PCS(c)$'s to a scalar objective. The first one is the expected *PCS* (Gao et al. 2019; Shen et al. 2021), denoted PCS_E, which is the expectation or the weighted average of $PCS(c)$ given a set of normalized weights, and the second one is the worst-case *PCS* (Li et al. 2020), denoted PCS_M, which is the minimum $PCS(c)$ obtained across all contexts.

The contextual R&S problem has seen an increasing interest in past few years. Notable works that study this problem under finite arm-context setting include but not limited to Gao et al. (2019), Jin et al. (2019), Li et al. (2020) and Shen et al. (2021). Li et al. (2020) focus on worst-case *PCS*, use independent normal random variables to model rewards, and propose a one-step look-ahead policy with an efficient value function approximation scheme to maximize PCS_M. Shen et al. (2021) assume that the reward for each arm is a linear function of the context and propose a two-stage algorithm based on the indifference zone formulation for optimizing the expected *PCS*. The most closely related work to ours is Gao et al. (2019). They model the rewards using independent normal random variables, extend the analysis in Glynn and Juneja (2004) to derive the large deviations rate function for the contextual *PCS*, and propose an algorithm that obtains the asymptotically optimal allocation ratio for both the worst-case and expected *PCS*. Jin et al. (2019) also follow a large deviations approach similar to that of Gao et al. (2019), however, their algorithm does not perform well when only the observed data is used to make decisions.

In this work, we use a separate Gaussian process (GP) to model the reward function for each arm. By leveraging the hidden correlation structure within the reward function, GPs offer significant improvements in posterior inference over independent normal random variables, which are commonly used in the R&S literature. Due to the finite solution space we focus on, when compared to a simpler multi-variate Gaussian prior, GPs may appear to complicate things by introducing kernels, which are typically used in continuous spaces. We prefer GPs since they have hyper-parameters that can be trained to better fit the observations as the optimization progresses. In contrast, a multi-variate Gaussian prior is a static object that needs to be specified beforehand based on limited domain knowledge. Using the posterior mean of the GP as the predictor of the true rewards, we derive the large deviations rate function for the contextual PCS, and show that it is identical for both PCS_E and PCS_M. We propose a sequential sampling policy that aims to maximize the rate function, and uses the GP posterior mean and variance to select the next arm-context to sample. Our sampling policy, GP-C-OCBA, is based on the same idealized policy as the C-OCBA policy by Gao et al. (2019), and mainly differs in the statistical model and the predictors used. We show that our algorithm achieves significantly improved sampling efficiency when compared with C-OCBA and DSCO (Li et al. 2020), and is highly competitive against the integrated knowledge gradient (IKG) algorithm (Pearce and Branke 2018), which uses the same GP model but requires significantly larger computational effort to decide on the next point to sample.

2 PROBLEM FORMULATION

We consider a finite set of arms (alternatives) $k \in \mathscr{K}$ and a finite set of contexts that are represented by vectors $c \in \mathscr{C}$. We assume that $\mathscr{K}$ is a set of categorical inputs, i.e., that there is no metric defined over $\mathscr{K}$. On the other hand, $\mathscr{C}$ is assumed to be subset of a known metric space, such as the Euclidean space of the corresponding dimension.

At each iteration, the decision maker selects an arm-context pair (k,c) to evaluate, and observes $y^n(k,c) = \mu^c(k,c) + \varepsilon^n(k,c)$, where $\mu^c(k,c)$ is the true (unknown) performance of arm k under context c (with $\cdot^c$ standing in for correct) and $\{\varepsilon^n(k,c)\}_n$ is zero-mean i.i.d. Gaussian noise with known variance $\sigma^2(k,c)$ (In implementation, $\sigma^2(k,c)$ is unknown and is substituted with a plug-in estimate). The decision maker aims to find the best alternative for each context, i.e., identify $\pi^*(c) = \arg\max_k \mu^c(k,c)$, with a given

sampling budget B. Given a finite sampling budget and noisy observations, $\mu^c(k,c)$ is not known exactly and we cannot solve for $\pi^*(c)$.

Let $\mu_n(k,c)$ denote our estimate of $\mu^c(k,c)$ at n-th iteration, and let $\pi^n(c) = \arg\max_k \mu_n(k,c)$ denote the predicted best arm for context c. Prior to iteration n, $\pi^n(c)$ is not realized and can be viewed as a random variable defined in an appropriate probability space. Suppose that the observations and the resulting $\pi^B(c)$ are generated following a given sampling policy. For any context c, we can measure the quality of this sampling policy with its probability of correct selection

$$PCS(c) = P(\pi^B(c) = \pi^*(c)).$$

As discussed in the introduction, $PCS(c)$ for all contexts define multiple objectives to consider while designing a sampling policy, with each *PCS* becoming larger as we allocate more samples to the corresponding c. Since we have a total sampling budget B rather than individual budgets for each context, it makes sense to work with a scalar objective instead. In the literature, there are two common approaches for constructing a scalar objective from $[PCS(c)]_{c\in\mathscr{C}}$. The first approach assumes that we are given a set of normalized weights $w(c)$ for each $c \in \mathscr{C}$ or the context variable follows a probability distribution $\{w(c), c \in \mathscr{C}\}$, and uses the expected *PCS* (Gao et al. 2019)

$$PCS_E = \mathbb{E}_{c\sim w(c)}[PCS(c)]$$

as the objective to be maximized. The other alternative takes a worst-case approach and aims to maximize the worst-case *PCS* (Li et al. 2020)

$$PCS_M = \min_{c\in\mathscr{C}} PCS(c).$$

We refer to either of PCS_E and PCS_M as the contextual *PCS*, and aim to design a sampling policy that maximizes the contextual *PCS* with the given sampling budget B. We propose an iterative approach that repeats the following steps at each iteration until the sampling budget is exhausted.

- Use the available data to update the statistical model of the reward function.
- With the objective of maximizing the large deviations rate function, use the sampling policy to decide on next arm-context, from which to sample one more observation.

In the following sections, we introduce our statistical model, which is a Gaussian process (GP) model that leverages the hidden correlation structure in the reward function for more efficient posterior inference, derive the large deviations rate function using the posterior mean of the GP as the predictor of the rewards, and introduce our sampling policy, which aims to maximize the large deviations rate function.

3 STATISTICAL MODEL

Gaussian processes are a class of Bayesian non-parametric models that are highly flexible for modeling continuous functions. By restricting to a discrete subset of the solution space, they also provide a powerful alternative to a multi-variate Gaussian prior for modeling a discrete set of correlated designs. Given the history of designs evaluated up to time n and the corresponding observations, $\mathscr{F}_n = \{D_{1:n}, O_{1:n}\}$, and a set of hyper-parameters θ, the GP implies a multi-variate Gaussian posterior distribution on any finite set of designs D_*, given by:

$$f(D_*) \mid \mathscr{F}_n, \theta \sim \mathscr{N}(\mu_n(D_*), \Sigma_n(D_*, D_*));$$

where $\mu_n(D_*)$ and $\Sigma_n(D_*, D_*)$ are the posterior mean vector and covariance matrix which are given by

$$\mu_n(D_*) = \mu_0(D_*) + \Sigma_0(D_*, D_{1:n}) A_n^{-1} (O_{1:n} - \mu_0(D_{1:n}))^\top,$$

$$\Sigma_n(D_*, D_*) = \Sigma_0(D_*, D_*) - \Sigma_0(D_*, D_{1:n}) A_n^{-1} \Sigma_0(D_{1:n}, D_*),$$

with $A_n = \Sigma_0(D_{1:n}, D_{1:n}) + \text{diag}(\sigma^2(D_{1:n}))$, where $\sigma^2(\cdot)$ denotes the standard deviation of the Gaussian observation noise. The prior mean, μ_0, is commonly set to a constant, e.g., $\mu_0(\cdot) = 0$, and the prior covariance, Σ_0, is commonly chosen from popular covariance kernels such as squared exponential, $\Sigma_0(D, D'; \theta) = \theta_0 \exp\left(-\frac{1}{2}\sum_{i=1}^{d} \theta_i (D_i - D'_i)^2\right)$, and Matèrn,

$$\Sigma_0(D, D'; \theta) = \theta_0 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu} d\right)^{\nu} K_\nu \left(\sqrt{2\nu} d\right),$$

where $d = \sqrt{\sum_i \theta_i (D_i - D'_i)^2}$, $\Gamma(\cdot)$ is the gamma function, $K_\nu(\cdot)$ is the modified Bessel function of second kind, ν is a shape parameter that is commonly set to $\nu = 5/2$, and θ denotes the output-scale and the length-scale parameters. The observation noise level, $\sigma^2(\cdot)$, is commonly unknown and gets replaced with a plug-in estimate, which is optimized jointly with the hyper-parameters θ, using maximum likelihood or maximum a-posteriori estimation.

We model the rewards for each arm using an independent GP model, which is defined over the context space $\mathscr{C}$ and trained using only the observations corresponding to that arm. There are a couple of reasons for using an independent GP model for each arm.

- With $\mathscr{K}$ as a set of categorical inputs, we do not have a metric defined over $\mathscr{K}$. Thus, we cannot use a covariance kernel with the categorical arm values as the inputs. It is possible to define a latent embedding of $\mathscr{K}$ into a Euclidean space and apply a covariance kernel in the embedded space (see, e.g., Guo and Berkhahn 2016; Feng et al. 2020), however, this introduces many additional hyper-parameters to the model, resulting in a non-convex optimization problem with many local optima for training the model. We found the predictive performance of such models to be highly sensitive to initial values of these hyper-parameters.
- The complexity of the GP inference is dominated by the inversion of matrix A_n, which has a $\mathscr{O}(n^3)$ cost using standard techniques. When using $K := |\mathscr{K}|$ independent GP models, each with n_k training inputs, we have K matrices $A^k_{n_k}$, with $\sum_k n_k = n$, where $A^k_{n_k}$ corresponds to k-th arm. The GP inference with independent models has a total cost of $\mathscr{O}(\sum_k n_k^3)$. If we assume that the samples are evenly distributed across arms, i.e., $n_k = n/K$, this results in a $\mathscr{O}(n^3/K^2)$ cost of inference for K independent models, which is much cheaper than $\mathscr{O}(n^3)$ for a single GP model.

On the other hand, if the set of arms belongs to a metric space, using a single GP model with a well defined covariance kernel over arms could lead to better sampling efficiency, and may be preferable when the samples are expensive or the sampling budget is severely limited. Although our derivation utilizes the independence of the models across different arms, the resulting GP-C-OCBA algorithm presented in this work is agnostic to the specifics of the GP model, and it can be used with either a single GP defined over the arm-context space or a GP model with the latent embedding as discussed in the first point, whenever such models are found to be appropriate.

For the rest of the paper, we use $\mu_n(k, c)$ and $\Sigma_n(k, c)$ to denote the posterior mean and posterior variance for context c under the GP model corresponding to k-th arm, and use $\Sigma_n(c, c'; k)$ to denote the covariance between two contexts c and c' for the k-th arm.

4 DERIVATION OF LARGE DEVIATIONS RATE FUNCTION

In this section, we derive the large deviations rate function for the contextual *PCS* measures. Our derivation follows the ideas presented in Glynn and Juneja (2004), with modifications to accommodate the use of posterior mean $\mu_n(k, c)$ instead of the sample mean. On related work, the ideas in Glynn and Juneja (2004) have been extended to the contextual setting by Gao et al. (2019), with the significant difference being their use of independent Gaussian random variables to model the rewards and the sample mean as the predictor versus our use of Gaussian processes to model rewards and $\mu_n(k, c)$ as the predictor.

Let $p(k,c)$ and $N(k,c) = np(k,c)$ denote the fraction and the total number, respectively, of samples allocated to (k,c). Both $p(k,c)$ and $N(k,c)$ are determined by the sampling policy and are assumed to be strictly positive. We ignore the technicalities arising from $N(k,c)$ not being an integer. For sake of simplicity, we leave implicit the dependency of the probabilities and other quantities on the sampling policy. For a given context c, the probability of false selection $PFS(c) = 1 - PCS(c)$ is given by

$$PFS(c) = P(\mu_n(\pi^*(c),c) < \mu_n(k,c), \exists k \neq \pi^*(c)).$$

We can lower and upper bound this respectively by

$$\max_{k\neq\pi^*(c)} P(\mu_n(\pi^*(c),c) < \mu_n(k,c)) \text{ and } (K-1)\max_{k\neq\pi^*(c)} P(\mu_n(\pi^*(c),c) < \mu_n(k,c)).$$

If for $k \neq \pi^*(c)$, $\lim_{n\to\infty} \frac{1}{n}\log P(\mu_n(\pi^*(c),c) < \mu_n(k,c)) = -G_{(k,c)}(p(\pi^*(c),c), p(k,c))$ for some rate function $G_{(k,c)}$, then $\lim_{n\to\infty}\frac{1}{n}\log PFS(c) = -\min_{k\neq\pi^*(c)} G_{(k,c)}(p(\pi^*(c),c),p(k,c))$. Similarly, the $PFS_\sim$ corresponding to both PCS_E and PCS_M can be lower and upper bounded by $\max_c PFS(c)$ and $(K-1)\max_c PFS(c)$ (or with an additional constant factor $\max_c w(c)/\min_c w(c)$ if $w(c)$ are not uniform), respectively. Thus, we can extend this to write the rate function of contextual *PCS* as

$$\lim_{n\to\infty}\frac{1}{n}\log PFS_\sim = -\min_{c\in\mathscr{C}}\min_{k\neq\pi^*(c)} G_{(k,c)}(p(\pi^*(c),c),p(k,c)). \tag{1}$$

To make use of (1), we need to find $G_{(k,c)}(p(\pi^*(c),c),p(k,c))$. We will follow the analysis of Glynn and Juneja (2004) and use the Gartner-Ellis Theorem (Dembo and Zeitouni 1998) to find $G_{(k,c)}(p(\pi^*(c),c),p(k,c))$, which requires understanding the distributional behavior of $\mu_n(k,c)$. In particular, we will need to calculate a certain limit of the log moment generating function (MGF): $\Lambda_n(\lambda;k,c) = \log\mathbb{E}[\exp(\lambda\mu_n(k,c))]$.

Let us focus on a fixed arm k for simplicity. Using the conjugacy property of GPs (under the assumption of Gaussian observation noise with known variance) and updating the posterior using samples from one context at a time, we can decompose $\mu_n(k,c)$ as

$$\mu_n(k,c) = \mu_0(k,c) + \sum_{i=1}^{|\mathscr{C}|}(\Sigma^{i-1}(c,c_i;k)]_{1\times N(k,c_i)}(A^i)^{-1}[Y^i - [\mu^{i-1}(k,c_i)]_{N(k,c_i)\times 1}),$$

where $\mu^{i-1}(k,c_i)$ is defined in the same way except with the summation being from 1 to $i-1$ with $\mu^0(\cdot,\cdot) = \mu_0(\cdot,\cdot)$, $[\alpha]_{n\times m}$ denotes the $n\times m$ matrix where each element is α, $A^i = [\Sigma^{i-1}(c_i,c_i;k)]_{N(k,c_i)\times N(k,c_i)} + diag_{N(k,c_i)}(\sigma^2(k,c_i))$, with $diag_N(\beta)$ denoting the diagonal matrix of size $N\times N$ with diagonals β, Y^i denotes the $N(k,c_i)\times 1$ matrix of observations corresponding to c_i, and

$$\Sigma^i(c,c';k) = \Sigma^{i-1}(c,c';k) - [\Sigma^{i-1}(c,c_i;k)]_{1\times N(k,c_i)}(A^i)^{-1}[\Sigma^{i-1}(c_i,c';k)]_{N(k,c_i)\times 1},$$

with $\Sigma^0(\cdot,\cdot;k) = \Sigma_0(\cdot,\cdot;k)$. The inverse of A^i can be calculated in closed form using the Sherman-Morrison formula (Meyer 2000). After some algebra, we can rewrite $\mu_n(k,c)$ as follows:

$$\mu_n(k,c) = \mu_0(k,c) + \sum_{i=1}^{|\mathscr{C}|}\frac{N(k,c_i)(\overline{Y^i - [\mu^{i-1}(k,c_i)]_{N(k,c_i)\times 1}})\Sigma^{i-1}(c,c_i;k)}{\sigma^2(k,c_i) + N(k,c_i)\Sigma^{i-1}(c_i,c_i;k)},$$

where $\overline{Y^i - [\mu^{i-1}(k,c_i)]_{N(k,c_i)\times 1}}$ denotes the average of the vector. Similarly, we can rewrite $\Sigma^i(c,c';k)$ as:

$$\Sigma^i(c,c';k) = \Sigma^{i-1}(c,c';k) - \frac{N(k,c_i)\Sigma^{i-1}(c,c_i;k)\Sigma^{i-1}(c_i,c';k)}{\sigma^2(k,c_i) + N(k,c_i)\Sigma^{i-1}(c_i,c_i;k)}.$$

For a Gaussian random variable $\mathcal{N}(\tilde{\mu},\tilde{\sigma}^2)$, the log-MGF is given by $\tilde{\mu}\lambda+\tilde{\sigma}^2\lambda^2/2$. Since the true distribution of samples is $y(k,c)\sim\mathcal{N}(\mu^c(k,c),\sigma^2(k,c))$ and the samples are independent of each other, we can view $\mu_n(\cdot,\cdot)$ as a linear combination of independent Gaussian random variables and write the log-MGF

$$\Lambda_n(\lambda;k,c)=\mu_0(k,c)\lambda+\sum_{i=1}^{|\mathscr{C}|}\left[(\mu^c(k,c_i)-\mu^{i-1}(k,c_i))C(k,c,i)\lambda+\frac{\sigma^2(k,c_i)C(k,c,i)^2\lambda^2}{2N(k,c_i)}\right],$$

where $C(k,c,i)=\frac{N(k,c_i)\Sigma^{i-1}(c,c_i;k)}{\sigma^2(k,c_i)+N(k,c_i)\Sigma^{i-1}(c_i,c_i;k)}$. Due to our use of $\mu_n(k,c)$ rather than the sample mean, part of the analysis in Glynn and Juneja (2004), particularly Lemma 1, cannot be used as is and needs to be re-established. For a given c and for $k\neq\pi^*(c)$, let $\Lambda_n(\lambda_{\pi^*(c)},\lambda_k;c)$ denote the log-MGF of $Z_n=(\mu_n(\pi^*(c),c),\mu_n(k,c))$. In order to use the Gartner-Ellis Theorem, we need to establish the limiting behavior of $\frac{1}{n}\Lambda_n(n\lambda_{\pi^*(c)},n\lambda_k;c)$.

$$\lim_{n\to\infty}\frac{1}{n}\Lambda_n(n\lambda_{\pi^*(c)},n\lambda_k;c)=\sum_{\kappa\in(\pi^*(c),k)}\mu^c(\kappa,c)\lambda_\kappa+\lim_{n\to\infty}\frac{nVar(\mu_n(\kappa,c))\lambda_\kappa^2}{2}. \tag{2}$$

The convergence of $\mathbb{E}[\mu_n(k,c)]\to\mu^c(k,c)$, which we substituted above, should be evident from the analysis of variance below. Here and in the remainder of this section, $\to$ denotes the limit as $n\to\infty$. The next step is to find the limiting behavior of $Var(\mu_n(\pi^*(c),c))$. To do so, note that $Var(\mu_n(k,c))=\sum_{i=1}^{|\mathscr{C}|}\frac{\sigma^2(k,c_i)C(k,c,i)^2}{N(k,c_i)}$. Due to conjugacy of GPs, we can choose to process the summation in any order, as long as we follow the same order for updating $\Sigma^i(\cdot,\cdot;k)$. Let us analyze $Var(\mu_n(\pi^*(c),c))$, for a given c, with the summation and the update processed starting from c, i.e., using $c_1=c$ with an appropriate re-ordering of $\mathscr{C}$. Note that

$$\Sigma^i(c',c'';k)\to\Sigma^{i-1}(c',c'';k)-\frac{\Sigma^{i-1}(c',c_i;k)\Sigma^{i-1}(c_i,c'';k)}{\Sigma^{i-1}(c_i,c_i;k)},$$

which implies that $\Sigma^i(\cdot,c_1;k)\to 0, i\geq 1$, and $C(k,c',i)\to\frac{\Sigma_0(c',c_i;k)}{\Sigma_0(c_i,c_i;k)}$ if $i=1$ and $C(k,c',i)\to 0$ otherwise. Thus, we can ignore the rest of the terms in the summation and write $Var(\mu_n(k,c))\xrightarrow{\approx}\frac{\sigma^2(k,c)}{N(k,c)}=\frac{\sigma^2(k,c)}{np(k,c)}$, where $\xrightarrow{\approx}$ denotes equivalence in the limit. We are now ready to continue from (2). Let $\Lambda^t(\lambda_k;k,c)$ denote the log-MGF of the observation $y(k,c)\sim\mathcal{N}(\mu^c(k,c),\sigma^2(k,c))$.

$$\begin{aligned}\lim_{n\to\infty}\frac{1}{n}\Lambda_n(n\lambda_{\pi^*(c)},n\lambda_k)&=\sum_{\kappa\in(\pi^*(c),k)}\mu^c(\kappa,c)\lambda_\kappa+\frac{\sigma^2(\kappa,c)\lambda_\kappa^2}{2p(\kappa,c)}\\&=\sum_{\kappa\in(\pi^*(c),k)}p(\kappa,c)\left(\frac{\mu^c(\kappa,c)\lambda_\kappa}{p(\kappa,c)}+\frac{\sigma^2(\kappa,c)\lambda_\kappa^2}{2p(\kappa,c)^2}\right)=\sum_{\kappa\in(\pi^*(c),k)}p(\kappa,c)\Lambda^t(\lambda_\kappa/p(\kappa,c);\kappa,c),\end{aligned}$$

which is the exact term in Lemma 1 of Glynn and Juneja (2004). Following the steps therein, we find that the rate function of Z_n is given by $I(x_{\pi^*(c)},x_k)=p(\pi^*(c),c)I^t(x_{\pi^*(c)};\pi^*(c),c)+p(k,c)I^t(x_k;k,c)$, where $I^t(x_k;k,c)=\frac{(x_k-\mu^c(k,c))^2}{2\sigma^2(k,c)}$ is the Fenchel-Legendre transform of $\Lambda^t(\lambda_k/p(k,c);k,c)$. With the rate function of Z_n established, Glynn and Juneja (2004) show that

$$G_{(k,c)}(p(\pi^*(c),c),p(k,c))=\inf_{x_{\pi^*(c)}\geq x_k}\left[p(\pi^*(c),c)I^t(x_{\pi^*(c)};\pi^*(c),c)+p(k,c)I^t(x_k;k,c)\right],$$

where the infimum can be calculated via differentiation (Gao et al. 2019), giving us

$$G_{(k,c)}(p(\pi^*(c),c),p(k,c))=\frac{(\mu^c(\pi^*(c),c)-\mu^c(k,c))^2}{2(\sigma^2(\pi^*(c),c)/p(\pi^*(c),c)+\sigma^2(k,c)/p(k,c))}.$$

Putting it all together, we summarize the result in the following theorem.

Theorem 1 Suppose that the observations are given as $y^n(k,c) = \mu^c(k,c) + \varepsilon_k^n(c)$ where $\varepsilon_k^n(c) \sim \mathcal{N}(0, \sigma^2(k,c))$ and $\varepsilon_k^n(c)$ are independent across n,k,c; and the best arm, $\pi^*(c)$, is unique for all $c \in \mathscr{C}$. Using $\mu_n(k,c)$ to predict the rewards, the large deviations rate function for both PCS_E and PCS_M is given by

$$\lim_{n\to\infty} \frac{1}{n} \log PFS_{\sim} = \min_{c\in\mathscr{C}} \min_{k\neq\pi^*(c)} \frac{(\mu^c(\pi^*(c),c) - \mu^c(k,c))^2}{2(\sigma^2(\pi^*(c),c)/p(\pi^*(c),c) + \sigma^2(k,c)/p(k,c))}. \tag{3}$$

Remark 1 The large deviations rate function presented in Theorem 1 is identical to the rate function derived in Gao et al. (2019), which is the same as the rate function originally derived in Glynn and Juneja (2004) with an additional minimum over the contexts. The main contribution of our analysis is to show that the same rate function is still applicable when the independent Gaussian model is replaced with a Gaussian process. This enables efficient inference by learning the correlations, resulting in significantly improved performance with small sampling budgets, while retaining similar asymptotical properties.

5 SAMPLING POLICY

In this section, we introduce the GP-C-OCBA policy, which aims to maximize the rate function presented in Theorem 1, as well as the IKG policy from the literature as it applies to our problem setting, and present a comparison of the computational cost of the two policies.

5.1 GP-C-OCBA

In classical R&S literature, optimal computing budget allocation (OCBA, Chen, Chick, Lee, and Pujowidianto 2015) is a popular approach for maximizing the *PCS* asymptotically. For the contextual R&S problem, Gao et al. (2019) derive the Karush-Kuhn-Tucker (KKT) conditions for maximizing (3), and propose an idealized sampling policy that iteratively realizes the KKT conditions.

The idealized policy derived by Gao et al. (2019) (see Section III.C of the paper for derivation) relies on $\mu^c(k,c)$, $\sigma^2(k,c)$, and $\pi^*(c)$, which are not known in practice, as well as $\hat{p}(k,c)$, which denotes the fraction of total samples allocated to (k,c) so far, which is different than $p(k,c)$ used in the derivation to denote the idealized asymptotical allocation rate. For a practical algorithm, Gao et al. (2019) replace $\mu^c(k,c)$ and $\sigma^2(k,c)$ with the sample mean and variance, respectively, and $\pi^*(c)$ with the corresponding estimate to define the C-OCBA policy.

Using the GP model, we take a similar approach and use the posterior mean $\mu_n(k,c)$ and the posterior variance $\Sigma_n(k,c)$, which are our estimates at time n, in place of $\mu^c(k,c)$ and $\sigma^2(k,c)/\hat{p}(k,c)$ in the idealized policy. The resulting GP-C-OCBA policy is presented in Algorithm 1.

Our experiments (in Section 6) show that using the GP model with the GP-C-OCBA sampling strategy leads to significantly higher contextual *PCS* using the same sampling budget, thanks to the improvements in the posterior inference from using a statistical model that leverages the hidden correlation structure in the reward function. An additional benefit of our approach over Gao et al. (2019) is in its applicability when the initial sampling budget is too small to draw multiple samples from each arm. Using normal random variables to model each arm-context pair requires a small number of samples from each pair for the initial estimate of the variance, which may limit the applicability of the algorithm when the sampling budget is limited. The GP prior, on the other hand, can be trained using very few samples for each arm, rather than each arm-context pair, thus the modified algorithm can be used even with a limited sampling budget.

5.2 Integrated Knowledge Gradient

On a related note, another applicable method for the contextual R&S problem is the integrated knowledge gradient (IKG) algorithm, which has been developed for the closely related problem of contextual Bayesian optimization. In this section, we introduce the IKG algorithm as it applies to our setting, and compare it

with GP-C-OCBA. IKG offers a strong benchmark for our method, since it is based on the same GP model and has demonstrated superior sampling efficiency in prior work.

Knowledge Gradient (KG) (Frazier et al. 2009) is a value-of-information type policy that was originally proposed for the R&S problem and later expanded to global optimization of black-box functions. It is well known for its superior sampling efficiency, which comes at a significant computational cost. For a given context c', we can write the KG factor, which measures the expected improvement in value of the maximizer for context c' from adding an additional sample at (k,c), as

$$\mathrm{KG}(k,c;c') = \mathbb{E}_n[\max_{k'\in\mathscr{K}} \mu_{n+1}(k',c') \mid (k^{n+1},c^{n+1}) = (k,c)] - \max_{k'\in\mathscr{K}} \mu_n(k',c').$$

In the classical R&S setting, where c and c' are redundant (i.e., there is only a single context), the KG policy operates by evaluating the arm $k^* = \arg\max_k \mathrm{KG}(k,c;c)$. To extend this to the contextual Bayesian optimization problem, Pearce and Branke (2018), Ding et al. (2020) and Pearce et al. (2020) each study an integrated (or summed) version of KG, under slightly different problem settings, where either the context space or both arm-context spaces are continuous. The main differences between these three works are in how they approximate and optimize the acquisition function in their respective problem settings. For our problem setting, these approaches are equivalent, and we refer to the sampling policy as IKG. We use IKG as a benchmark to evaluate the sampling efficiency of our proposed algorithm.

The IKG factor is simply a weighted sum of KG factors corresponding to each context. It measures the weighted sum of the improvement in value of maximizers, and is written as $\mathrm{IKG}(k,c) = \sum_{c'\in\mathscr{C}} \mathrm{KG}(k,c;c')w(c')$. At each iteration, the IKG policy samples the arm-context pair that maximizes the IKG factor, $(\tilde{k}^*,\tilde{c}^*) = \arg\max_{k\in\mathscr{K},c\in\mathscr{C}} \mathrm{IKG}(k,c)$. The IKG policy is proven to be consistent and its superior sampling efficiency has been demonstrated in numerical experiments.

The main difficulty with using the IKG policy is its computational cost. In the finite arm-context setting that we are working with, the $\mathrm{KG}(k;c)$ can be computed exactly using Algorithm 1 from Pearce and Branke (2018), which has a cost of $\mathscr{O}(K\log K)$ for any pair (k,c), given $\mu_n(\cdot,\cdot)$ and $\Sigma_n(\cdot,\cdot)$. This translates to an $\mathscr{O}(|\mathscr{C}|K\log K)$ cost for calculating the IKG factor for a given (k,c). In total, to find the next pair to sample using IKG costs $\mathscr{O}(|\mathscr{C}|^2K^2\log K)$ for calculating the IKG factors, and an additional $\mathscr{O}(\sum_k[n_k^3+|\mathscr{C}|^2n_k+|\mathscr{C}|n_k^2])$ to calculate posterior mean and covariance matrices, where n_k denotes the total number of samples allocated to arm k.

Algorithm 1 GP-C-OCBA for contextual R&S

1: Use a pre-determined rule to allocate initial samples to each arm.
2: **for** n = 1, ..., B **do**
3: Update the GP model with the available observations, calculate $\mu_n(k,c)$ and $\Sigma_n(k,c)$.
4: For all $c\in\mathscr{C}$ and $k\neq\pi^n(c)$, calculate

$$\zeta(k,c) = \frac{(\mu_n(\pi^n(c),c)-\mu_n(k,c))^2}{\Sigma_n(\pi^n(c),c)+\Sigma_n(k,c)};$$

and set

$$\psi^{(1)}(c) = \frac{\hat{p}(\pi^n(c),c)}{\Sigma_n(\pi^n(c),c)}; \qquad \psi^{(2)}(c) = \sum_{k\neq\pi^n(c)} \frac{\hat{p}(k,c)}{\Sigma_n(k,c)}.$$

5: Solve for $(\tilde{k}^*,\tilde{c}^*) = \arg\min_{k\neq\pi^n(c),c\in\mathscr{C}}\zeta(k,c)$, and draw an additional sample from $(\pi^n(\tilde{c}^*),\tilde{c}^*)$, if $\psi^{(1)}(\tilde{c}^*) < \psi^{(2)}(\tilde{c}^*)$, and draw an additional sample from $(\tilde{k}^*,\tilde{c}^*)$ otherwise.
6: **end for**
7: **Return:** $\pi^B(c) = \arg\max_k \mu_B(k,c), c\in\mathscr{C}$ as the set of predicted best arms.

On the other hand, the cost of GP-C-OCBA is dominated by the cost of calculating the posterior mean and variance for each arm-context pair, which has a total cost of $\mathcal{O}(\sum_k [n_k^3 + |\mathscr{C}| n_k^2])$. Note that we avoid the $|\mathscr{C}|^2 n_k$ term since our algorithm only requires the posterior variance, as well as the $\mathcal{O}(|\mathscr{C}|^2 K^2 \log K)$ cost of IKG calculations. This puts GP-C-OCBA at a significant advantage in terms of computational complexity.

6 NUMERICAL EXPERIMENTS

In this section, we demonstrate the performance of our algorithm on a set of synthetic benchmark problems. We compare our algorithm with the algorithms by Li et al. (2020) (DSCO), Gao et al. (2019) (C-OCBA), and with the IKG algorithm as described in Section 5.2. We chose these benchmarks since DSCO and C-OCBA were both proposed for the contextual R&S with the finite arm-context setting that is studied in this paper and has demonstrated superior performance in experiments; and IKG was chosen since KG type algorithms, including variants of IKG, have consistently demonstrated superior sampling efficiency under various problem settings.

We implemented the experiments in Python, and used the GP models from the BoTorch package (Balandat et al. 2020) with the default priors. The GP hyper-parameters are re-trained every 10 iterations, and we use the Matern $5/2$ kernel. The code will be made available at https://github.com/saitcakmak/contextual_rs upon publication.

6.1 Test Functions

For the experiments, we generate the true rewards, $\mu^c(k,c)$, by evaluating common global optimization test functions on randomly drawn points from the function domain. We use the first dimension of the function input for the arms, i.e. each arm corresponds to a fixed value of x_1, and spread the arms evenly across the corresponding domain. The remaining input dimensions are used for the contexts, thus, contexts are $d-1$ dimensional vectors for a d dimensional test function. Put together, this corresponds to $\mu^c(k,c) = f(x_k, x_c)$ where x_k and x_c are fixed realizations of 1 and $d-1$ dimensional uniform random variables, respectively. The rewards are observed with additive Gaussian noise with standard deviation set as $\frac{f_{max}-f_{min}}{100/3}$, where f_{max} and f_{min} are estimated using 1000 samples drawn uniformly at random from the function domain. We use the following functions in our experiments:

- The 2D Branin function, evaluated on $[-5,10] \times [0,10]$:

$$f(x) = -(x_2 - bx_1^2 + cx_1 - r)^2 - 10(1-t)cos(x_1) - 10,$$

  where $b = 5.1/(4\pi^2)$, $c = 5/\pi$, $r = 6$ and $t = 1/(8\pi)$. We run two experiments using the Branin function, both with 10 arms and 10 contexts. The first objective is the expected *PCS* with weights set arbitrarily as $[0.03, 0.07, 0.2, 0.1, 0.15, 0.2, 0.02, 0.08, 0.1, 0.05]$, and the second objective is the worst-case *PCS*. We draw 2 samples from each arm-context pair for the initialization phase.
- The 2D Griewank function, evaluated on $[-10,10]^2$:

$$f(x) = -\sum_{i=1}^{d} \frac{x_i^2}{4000} + \prod_{i=1}^{d} \cos\left(\frac{x_i}{\sqrt{i}}\right) - 1.$$

  We run two experiments with the Greiwank function, using 10 arms and 20 contexts. We use the expected *PCS* with uniform weights and the worst-case *PCS*, and initialize with 2 samples from each arm-context pair.
- The 3D Hartmann function, evaluated on $[0,1]^3$:

$$f(x) = \sum_{i=1}^{4} \alpha_i \exp\left(-\sum_{j=1}^{3} A_{ij}(x_j - P_i j)^2\right),$$

where the constants α, A, and P are given in Surjanovic and Bingham (2013). We run a single experiment with 20 arms and 20 contexts, using the expected *PCS* with uniform weights. Since the number of arm-context pairs is quite large in this experiment, we select only 6 contexts for each arm, uniformly at random, and draw a single sample from these contexts for the initial stage. Due to insufficient initial sampling budget, DSCO and C-OCBA are not applicable here, and we only run the GP based algorithms for this experiment.

- The 8D Cosine8 function, evaluated on $[-1,1]^8$:

$$f(x) = 0.1\sum_{i=1}^{8} cos(5\pi x_i) - \sum_{i=1}^{8} x_i^2.$$

We run a single experiment with the mean *PCS* objective with uniform weights. We use 20 arms and 40 contexts. For the initial stage, we randomly select 16 contexts for each arm, and draw a single sample from these contexts, which is again due to the large number of arm-context pairs in this experiment. Similar to the previous experiment, DSCO and C-OCBA are not applicable here, we only run the GP based algorithms for this experiment.

6.2 Results

The experiment results are plotted in Figure 1. We ran each experiment for 2000 iterations, except for IKG in Hartmann and Cosine8 functions, which were run for 1000 iterations due to their excessive cost. The plots show the empirical contextual *PCS*, estimated using 100 replications. The first 4 plots compare all algorithms, however, the last 2 only compare GP-C-OCBA and IKG, due to small initial budget preventing drawing of multiple samples from each arm-context pair, which is necessary to form an initial estimate of sample mean and variance that is used by DSCO and C-OCBA. In all 4 of the experiments comparing all algorithms, we see that the two algorithms using the GP models achieve significantly higher contextual *PCS* compared to the algorithms using independent normal random variables to model rewards, which clearly demonstrates the benefit of using a statistical model that leverages the hidden correlation structure in the reward function. In particular, with the Griewank function, we see that the DSCO and C-OCBA have worst-case *PCS* very close to 0, which is explained by a total of 2400 samples being far from sufficient to form reliable estimates for 200 arm-context pairs using an independent statistical model.

Although it is slightly trailing behind in some experiments, we see that GP-C-OCBA is highly competitive against IKG, while having significantly smaller computational complexity. The wall-clock times for the experiments are reported in Table 1. We see that even in the smallest experiments, the IKG algorithm takes about 5 times as long to run, with the ratio increasing significantly to about 75 times as we move to larger experiments. The reported run times are for 1000 iterations of a full experiment, and include the cost of fitting the GP model, which is identical for both algorithms. It is worth noting that cost of evaluating the test functions is negligible. As the cost of function evaluation increases, the results would look nicer for IKG, though it would still remain the more expensive alternative.

Table 1: Comparison of computational cost of IKG and GP-C-OCBA. We report the average wall-clock time, in seconds, for running 1000 iterations of the given experiment. The experiments were run on a shared cluster using 4 cores of the allocated CPU. To save on space, we report the average run-time of the two objectives for Branin and Griewank.

Algorithm	Branin PCS_E / PCS_M	Griewank PCS_E / PCS_M	Hartmann	Cosine8
IKG	1218	4135	11096	43224
GP-C-OCBA	249	296	441	544

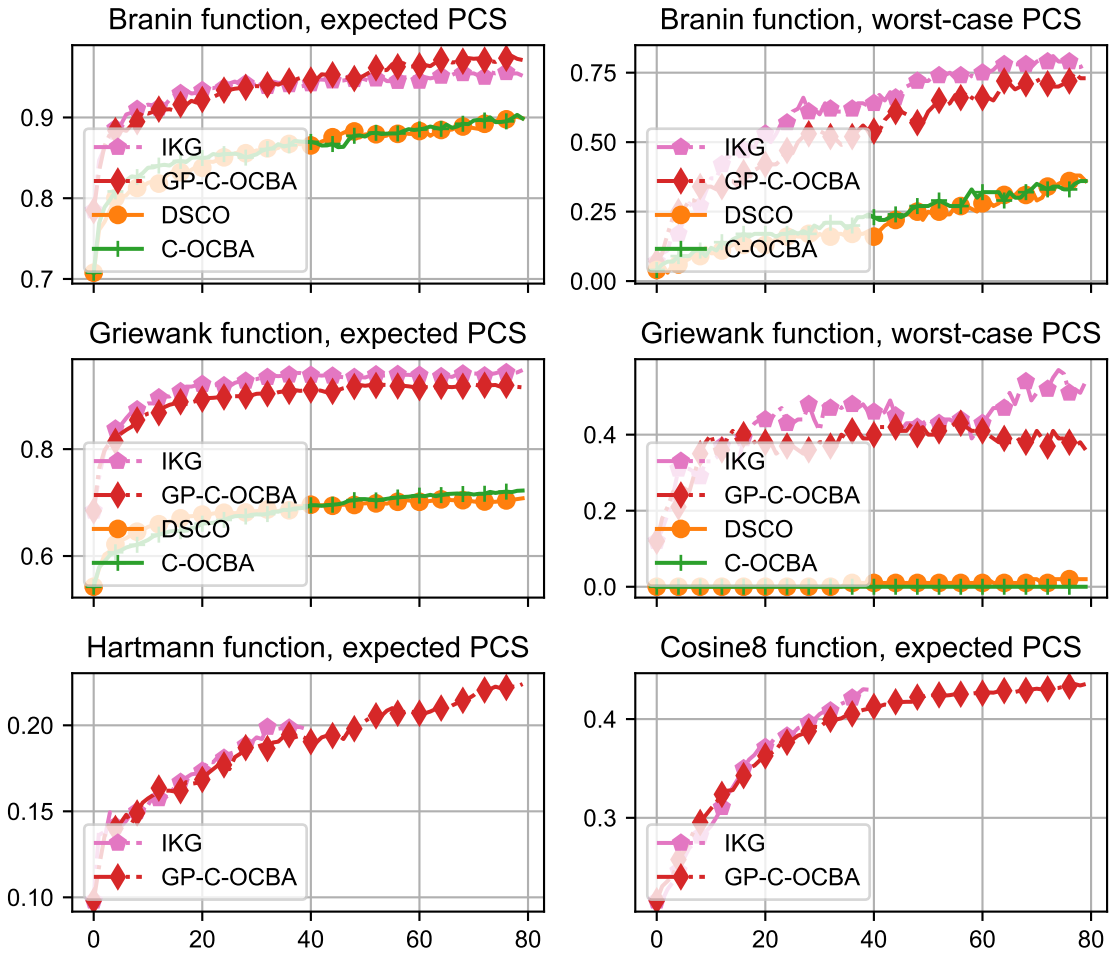

Figure 1: Experiments using Branin function with PCS_E and PCS_M, Griewank function with PCS_E and PCS_M, Hartmann function with PCS_E, and Cosine8 with PCS_E. The plots show the empirical contextual *PCS* on the y-axis, and the number of iterations/samples (post-initialization) on the x-axis.

Overall, the experiments show that GP-C-OCBA is highly competitive in terms of sampling efficiency, while being significantly cheaper than other high performing benchmarks. We believe that this makes GP-C-OCBA an attractive option for any practitioner that is faced with a contextual R&S problem.

7 CONCLUSION

We studied the contextual R&S problem under finite arm-context setting, using a separate GP to model the reward for each arm. We derived the large deviations rate functions for the contextual *PCS*, and proposed the GP-C-OCBA algorithm that aims to maximize the rate function using the information available in the GP posterior. GP-C-OCBA is shown to achieve significant sampling efficiency, while having a significantly smaller computational overhead compared to other competitive alternatives.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the support by the Air Force Office of Scientific Research under Grant FA9550-19-1-0283.

REFERENCES

Balandat, M., B. Karrer, D. R. Jiang, S. Daulton, B. Letham, A. G. Wilson, and E. Bakshy. 2020. "BoTorch: A Framework for Efficient Monte-Carlo Bayesian Optimization". In *Advances in Neural Information Processing Systems 33*, edited by H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, 21524–21538. Red Hook, NY: Curran Associates, Inc.

Chen, C.-H., S. E. Chick, L. H. Lee, and N. A. Pujowidianto. 2015. "Ranking and Selection: Efficient Simulation Budget Allocation". In *Handbook of Simulation Optimization*, edited by M. C. Fu, 45–80. New York, NY: Springer-Verlag.

Dembo, A., and O. Zeitouni. 1998. *Large Deviations Techniques and Applications*. 2nd ed. Berlin: Springer-Verlag.

Ding, L., L. J. Hong, H. Shen, and X. Zhang. 2020. "Knowledge Gradient for Selection with Covariates: Consistency and Computation". http://arxiv.org/abs/1906.05098 Accessed 31st March 2021.

Feng, Q., B. Letham, H. Mao, and E. Bakshy. 2020. "High-Dimensional Contextual Policy Search with Unknown Context Rewards using Bayesian Optimization". In *Advances in Neural Information Processing Systems 33*, edited by H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, 22032–22044. Red Hook, NY: Curran Associates, Inc.

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. Chen. 2019. "Selecting the Optimal System Design under Covariates". In *15th International Conference on Automation Science and Engineering*. August 22nd-26th, Vancouver, BC, Canada, 547-552.

Glynn, P., and S. Juneja. 2004. "A Large Deviations Perspective on Ordinal Optimization". In *Proceedings of the 2004 Winter Simulation Conference*, edited by R. G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, 577–585. Piscataway, NJ: Institute of Electrical and Electronics Engineers, Inc.

Guo, C., and F. Berkhahn. 2016. "Entity Embeddings of Categorical Variables". http://arxiv.org/abs/1604.06737 Accessed 31st March 2021.

Jin, X., H. Li, and L. H. Lee. 2019. "Optimal Budget Allocation in Simulation Analytics*". In *15th International Conference on Automation Science and Engineering*. August 22nd-26th, Vancouver, BC, Canada, 178-182.

Kim, S.-H., and B. L. Nelson. 2007. "Recent Advances in Ranking and Selection". In *Proceedings of the 2007 Winter Simulation Conference*, edited by S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, 162–172. Piscataway, NJ: Institute of Electrical and Electronics Engineers, Inc.

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. Bae, B. Feng, S. Kim, S. Lazarova-Molnar, Z. Zheng, T. Roeder, and R. Thiesing, 2060–2070. Piscataway, NJ: Institute of Electrical and Electronics Engineers, Inc.

Meyer, C. D. 2000. *Matrix Analysis and Applied Linear Algebra*. Philadelphia, PA: Society for Industrial and Applied Mathematics.

Nunes, M. A. S., and R. Hu. 2012. "Personality-Based Recommender Systems: An Overview". In *Proceedings of the Sixth ACM Conference on Recommender Systems*. September 9th-13th, Dublin, Ireland, 5-6.

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

Pearce, M., J. Klaise, and M. Groves. 2020. "Practical Bayesian Optimization of Objectives with Conditioning Variables". http://arxiv.org/abs/2002.09996 Accessed 31st March 2021.

Shen, H., L. J. Hong, and X. Zhang. 2021. "Ranking and Selection with Covariates for Personalized Decision Making". *INFORMS Journal on Computing*. Articles in advance, published online.

Surjanovic, S., and D. Bingham. 2013. "Hartmann 3-Dimensional Function". In *Virtual Library of Simulation Experiments*. https://www.sfu.ca/~ssurjano/hart3.html Accessed 31st March 2021.

AUTHOR BIOGRAPHIES

SAIT CAKMAK is a PhD student in Operations Research at School of Industrial & Systems Engineering at Georgia Institute of Technology. His research focuses on black-box optimization and simulation optimization. His e-mail address is scakmak3@gatech.edu.

SIYANG GAO is an Associate Professor of the Department of Systems Engineering and Engineering Management at the City University of Hong Kong. He holds a PhD in Industrial Engineering from University of Wisconsin-Madison. His research interests include simulation optimization and its application to healthcare. His e-mail address is siyangao@cityu.edu.hk.

ENLU ZHOU is an Associate Professor in the School of Industrial and Systems Engineering at Georgia Institute of Technology. Her research interests include control, simulation, and stochastic optimization. Her email address is enlu.zhou@isye.gatech.edu.