



Contextual Ranking and Selection with Gaussian Processes and Optimal Computing Budget Allocation

SAIT CAKMAK and YUHAO WANG, Georgia Institute of Technology, USA

SIYANG GAO, City University of Hong Kong, CHINA

ENLU ZHOU, Georgia Institute of Technology, USA

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-alternative-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 alternative and derive the large deviations rate function for both the expected and worst-case contextual probability of correct selection. We propose the GP-C-OCBA sampling policy, which uses the Gaussian process posterior to iteratively allocate observations to maximize the rate function. We prove its consistency and show that it achieves the optimal convergence rate under the assumption of a non-informative prior. Numerical experiments show that our algorithm is highly competitive in terms of sampling efficiency, while having significantly smaller computational overhead.

CCS Concepts: • Computing methodologies → Simulation theory;

Additional Key Words and Phrases: Ranking and selection, Gaussian process, optimal computing budget allocation

ACM Reference format:

Sait Cakmak, Yuhao Wang, Siyang Gao, and Enlu Zhou. 2024. Contextual Ranking and Selection with Gaussian Processes and Optimal Computing Budget Allocation. *ACM Trans. Model. Comput. Simul.* 34, 2, Article 8 (April 2024), 24 pages.

<https://doi.org/10.1145/3633456>

8

1 INTRODUCTION

Ranking and Selection (R&S) studies the problem of identifying the best among a finite number of alternatives, 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

The first, second, and fourth authors gratefully acknowledge the support by Air Force Office of Scientific Research under Grants No. FA9550-19-1-0283 (<https://www.afrl.af.mil/AFOSR/>) and No. FA9550-22-1-0244 (<https://www.afrl.af.mil/AFOSR/>) and the support by National Science Foundation under Grant No. DMS2053489 (<https://www.nsf.gov>).

Authors' addresses: S. Cakmak, Y. Wang, and E. Zhou, Georgia Institute of Technology, School of Industrial and Systems Engineering, 755 Ferst Drive NW, Atlanta, Georgia, USA, 30332; e-mails: {scakmak3, yuhaowang}@gatech.edu, enlu.zhou@isye.gatech.edu; S. Gao, City University of Hong Kong, Department of Advanced Design and Systems Engineering, 83 Tat Chee Avenue, Kowloon, Hong Kong, CHINA; e-mail: siyangao@cityu.edu.hk.



This work is licensed under a Creative Commons Attribution International 4.0 License.

© 2024 Copyright held by the owner/author(s).

1049-3301/2024/04-ART8

<https://doi.org/10.1145/3633456>

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 References [2, 20] 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 [24]. 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 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 alternative-context pair to evaluate and observes a noisy evaluation of the true reward. With a finite sampling budget and noisy observations, it is not possible to identify the best alternative with certainty, and we need to design a sampling policy, which takes in the current estimate of rewards and outputs the next alternative-context pair to sample, 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* [9, 29], denoted as 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* [21], denoted as 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 alternative-context setting include but not limited to References [13, 16, 19, 21, 28, 29]. Reference [21] focuses on worst-case *PCS*, uses independent normal random variables to model rewards, and proposes a one-step look-ahead policy with an efficient value function approximation scheme to maximize PCS_M . Reference [29] assumes that the reward for each alternative is a linear function of the context and proposes a two-stage algorithm based on the indifference zone formulation for optimizing the expected *PCS*. Reference [28] considers an online framework, where the contexts arrive sequentially and real-time decision is needed for each arriving context. Reference [16] also considers real-time decision making but further integrates machine learning predictions for the performance estimates. The most closely related work to ours is Reference [9]. They model the rewards using independent normal random variables, extend the analysis in Reference [15] 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*. In contrast, we use a Gaussian process, model correlations between contexts, and extend the large deviations approach to leverage the efficient inference brought in by our statistical model. Reference [19] also follows a large deviations approach similar to that of Reference [9]; 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 alternative. 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 can be characterized by a small number of hyper-parameters with the introduction of the kernel function. In contrast, a multi-variate Gaussian prior needs specification of its covariance matrix entry by entry, which is difficult based on limited domain knowledge. The contributions of our article are summarized as follows:

- Using the posterior mean of the GP as the predictor of the true rewards and leveraging a novel decomposition of the GP update formulas, we derive the large deviations rate function for the contextual PCS, and show that it is identical for both $PCSE$ 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 alternative-context to sample. Our sampling policy, GP-C-OCBA, is based on the same idealized policy as the C-OCBA policy [9], and mainly differs in the statistical model and the predictors used.
- Under a set of mild assumptions, we show that GP-C-OCBA almost surely identifies the best alternative for each context and the sampling allocations produced by GP-C-OCBA converge to the optimal static allocation ratio as the sampling budget goes to infinity.
- We build on this to establish, to the best of our knowledge, the first convergence rate result for a contextual R&S procedure in the literature. We show that the resulting contextual PCS converges to 1 at the optimal exponential rate.
- Finally, we demonstrate the performance of GP-C-OCBA in numerical experiments. We show that GP-C-OCBA achieves significantly improved sampling efficiency when compared with C-OCBA [9], DSCO [21], TS, and TS+ [29]; and is highly competitive against the **integrated knowledge gradient (IKG)** algorithm [26], which uses the same GP model but requires significantly larger computational effort to decide on the next alternative-context to evaluate.

2 PROBLEM FORMULATION

We consider a finite set of alternatives $\{k \in \mathcal{K}\}$ and a finite set of contexts $\{c \in C \subset \mathbb{R}^d\}$, where d is the dimension of the context variable. The assumption of finite context set is common in the literature of contextual R&S (for example, see References [9, 19, 21, 29]). A finite set can also be used to represent general (continuous) context spaces, where a fixed set of contexts are sampled to discretize the space. We assume that \mathcal{K} is a set of categorical inputs, i.e., that there is no metric defined over \mathcal{K} .

The observations of the reward function, $F(k, c)$, is subject to noise, and the decision maker is given a total budget of B function evaluations. The aim is to identify the true best alternative for each context,

$$\pi^*(c) := \arg \max_{k \in \mathcal{K}} F(k, c),$$

so that a decision can be made immediately once the final context is revealed. Since $F(k, c)$ is unknown and the observations are noisy, $\pi^*(c)$ cannot be identified with certainty using a finite budget. If we let $\mu_B(k, c)$ denote the estimate of $F(k, c)$ obtained using B function evaluations, then we can write the corresponding estimated best alternative as

$$\pi^B(c) := \arg \max_{k \in \mathcal{K}} \mu_B(k, c).$$

Note that $\pi^B(c)$ is determined by the outcome of the B function evaluations, as well as the sampling policy used to allocate those B evaluations. Thus, for any given sampling policy, $\pi^B(c)$ is a random variable. We can measure the quality of the given sampling policy, for any context c , by

the corresponding probability of correct selection (PCS) after exhausting the budget B :

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

As discussed in the introduction, $PCS^B(c)$ 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 $[PCS^B(c)]_{c \in C}$. The first approach assumes that we are given a set of normalized weights $w(c)$ for each $c \in C$ or the context variable follows a probability distribution $\{w(c), c \in C\}$, and uses the expected PCS [9]

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

as the objective to be maximized. The other alternative takes a worst-case approach and aims to maximize the worst-case PCS [21]

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

We refer to either of PCS_E^B and PCS_M^B as the contextual PCS . We 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 reward observations collected so far 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 alternative-context, from which to sample one more observation.

In the following sections, we introduce our statistical model, which builds on a GP 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 non-parametric Bayesian 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 designs. In this work, we use Gaussian processes to model the reward function. We use a separate GP for each alternative, which allows modeling the correlations between the reward function corresponding to each context for that alternative. As discussed at the beginning of the Section 2, the fixed context set can be regarded as a discretized approximation of the original context space. Although we restrict R&S only on the discretized context set, we can still estimate the reward function $F(k, c')$ with c' possibly outside the finite set C , using the same GP on the whole context space. Under a mild assumption that the original context space is compact and the reward function $F(k, c)$ is continuous in terms of c , the error caused by such discretization will diminish as the set C becomes more dense in the original context space.

Let $\mathcal{F}_n(k) = \{D_n(k), Y_n(k)\}$ denote the history of observations corresponding to alternative k up to time n (i.e., n total observations across all alternatives), where $D_n(k)$ and $Y_n(k)$ denote the set of contexts that have been evaluated and the corresponding observations, respectively. Given the history $\mathcal{F}_n(k)$ and a set of hyper-parameters θ , the GP implies a multi-variate Gaussian posterior

distribution on the reward function, given by

$$F(k, C) \mid \mathcal{F}_n, \theta \sim \mathcal{N}(\mu_n(k, C), \Sigma_n(C, C; k)),$$

where $\mu_n(k, C)$ and $\Sigma_n(C, C; k)$ are the posterior mean vector and covariance matrix, which, assuming Gaussian observation noise with known standard deviation $\sigma^2(\cdot, \cdot)$, can be computed in closed form as

$$\begin{aligned}\mu_n(k, C) &= \mu_0(k, C) + \Sigma_0(C, D_n(k); k) A_n^{-1}(k) (Y_n(k) - \mu_0(D_n(k)))^\top, \\ \Sigma_n(C, C; k) &= \Sigma_0(C, C; k) - \Sigma_0(C, D_n(k); k) A_n^{-1}(k) \Sigma_0(D_n(k), C; k),\end{aligned}$$

with $A_n(k) = \Sigma_0(D_n(k), D_n(k); k) + \text{diag}(\sigma^2(k, D_n(k)))$. The prior mean, μ_0 , is commonly set to a constant, e.g., $\mu_0(\cdot, \cdot) = 0$, and the prior covariance, Σ_0 , is commonly chosen from popular covariance kernels such as squared exponential, $\Sigma_0(c, c'; k \mid \theta) = \theta_0 \exp(-\frac{1}{2} \text{dist}^2)$, and Matérn,

$$\Sigma_0(c, c'; k \mid \theta) = \theta_0 \frac{2^{1-\nu}}{\Gamma(\nu)} (\sqrt{2\nu} \text{dist})^\nu K_\nu(\sqrt{2\nu} \text{dist}),$$

where $\text{dist} = \sqrt{\langle \theta_{1:d}(c - c'), c - c' \rangle}$ is the coordinate-wise scaled Euclidean distance, $\Gamma(\cdot)$ 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 $\nu = 5/2$. θ_0 and $\theta_{1:d}$ denote the output-scale and the length-scale parameters, respectively, and they are collectively referred to as the hyper-parameters of the GP. Throughout, we keep the dependence on the hyper-parameters $\theta = \{\theta_0, \theta_{1:d}\}$ implicit in the notation. The observation noise level, $\sigma^2(\cdot, \cdot)$, 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. In addition to $\mu_n(k, c)$ and $\Sigma_n(c, c'; k)$ to denote the posterior mean and covariance, we use $\Sigma_n(k, c) := \Sigma_n(c, c; k)$ to denote the posterior variance.

In this work, we model the rewards for each alternative using an independent GP model, which is defined over the context space C and trained using only the observations corresponding to that alternative. The literature has examples of papers using both independent GP models (e.g., References [8, 25]) as well as a single GP defined over the whole design space (i.e., the product space of alternative and contexts) (e.g., References [10, 27]). In any application, problem specifics should be taken into account while choosing the model. In the setting of this article, there are a couple of reasons for using an independent GP model for each alternative.

- With \mathcal{K} as a set of categorical inputs, we do not have a metric defined over \mathcal{K} . Thus, we cannot use a covariance kernel with the categorical alternative values as the inputs. It is possible to define a latent embedding of \mathcal{K} into a Euclidean space and apply a covariance kernel in the embedded space (see, e.g., References [10, 18]). However, this introduces 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 $\mathcal{O}(n^3)$ cost for an $n \times n$ matrix using standard techniques. If we use a single GP fit on all n observations, then this results in an $n \times n$ matrix A_n . In contrast, when using $K := |\mathcal{K}|$ independent GP models, each with $N_n(k)$ training inputs, we have K matrices $A_n(k)$, each of size $N_n(k) \times N_n(k)$, with $\sum_k N_n(k) = n$. The GP inference with independent models has a total cost of $\mathcal{O}(\sum_k n_k^3)$. If we assume that the samples are evenly distributed across alternatives, i.e., $N_n(k) = n/K$, then this results in a $\mathcal{O}(n^3/K^2)$ cost of inference for K independent models, which is much cheaper than $\mathcal{O}(n^3)$ for a single GP model.

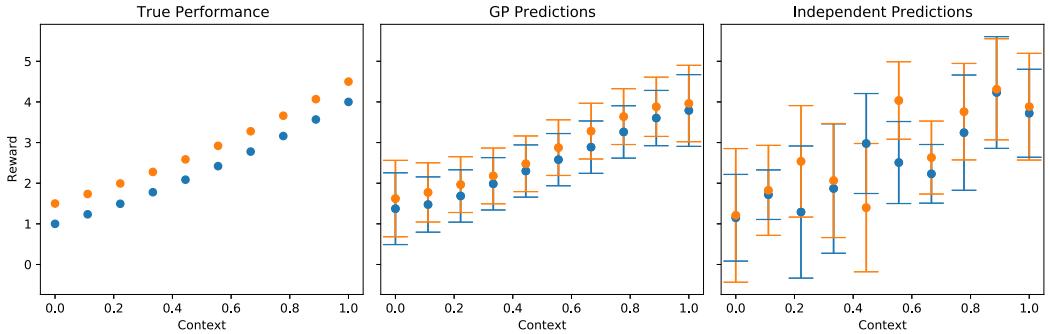


Fig. 1. Comparison of the GP predictions with those of an independent frequentist model. Left to right, the plots show the true performances of the two alternatives (y-axis) as a function of the contexts (x-axis), the GP predictions and the independent predictions after sampling each alternative-context pair 10 times. The error bars denote the two standard deviations of the predictive variance. It is seen that the GP model smooths out the errors and leads to much more reliable predictions and lower predictive variance.

However, if the set of alternatives belongs to a metric space, then using a single GP model with a well defined covariance kernel over alternatives 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 alternatives, 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 alternative-context space or a GP model with the latent embedding as discussed in the first point, whenever such models are found to be appropriate.

In Figure 1, we see a comparison of the predictions of the GP model used in this article, and the independent model used by Reference [9] where the performance of each alternative-context pair is modeled by a normal random variable. Both models are fitted on identical data, and the posterior mean is plotted along with error bars denoting two standard deviations of the predictive variance. The GP model smooths out the errors in the observations and leads to significantly more accurate predictions. In addition, we see that the error bars for the GP predictions are significantly smaller, which is attributed to the model using neighboring observations to lower the predictive uncertainty.

4 THE LARGE DEVIATIONS RATE FUNCTION

In this section, we derive the large deviations rate function corresponding to the contextual PCS measures. In particular, we calculate the rate at which the probability of false selection,

$$PFS_E^n = 1 - PCS_E^n \text{ or } PFS_M^n = 1 - PCS_M^n,$$

approaches zero as the number of samples $n \rightarrow \infty$. The main result is summarized in the following theorem.

THEOREM 4.1. *Suppose that the observations are given as $y^n(k, c) = F(k, c) + \epsilon^n(k, c)$ where $\epsilon^n(k, c) \sim \mathcal{N}(0, \sigma^2(k, c))$ and $\epsilon^n(k, c)$ are independent across n, k, c ; and the best alternative, $\pi^*(c)$, is unique for all $c \in C$. Using $\mu_n(k, c)$ to predict the rewards, the large deviations rate function for both PCS_E^n and PCS_M^n is given by*

$$\lim_{n \rightarrow \infty} \frac{1}{n} \log PFS_{\sim}^n = - \min_{c \in C} \min_{k \neq \pi^*(c)} \frac{(F(\pi^*(c), c) - F(k, c))^2}{2(\sigma^2(\pi^*(c), c)/p(\pi^*(c), c) + \sigma^2(k, c)/p(k, c))}, \quad (1)$$

where $PFS_{\mathcal{C}}^n$ is either of PFS_E^n or PFS_M^n , and $p(k, c)$ denotes the fraction of samples allocated to k, c .

The derivation of the result follows the ideas originating in Reference [15], with modifications to accommodate the use of the posterior mean $\mu_n(k, c)$ instead of the sample mean. We first show that the $PFS^n(c)$ can be upper and lower bounded by a constant multiple of $\max_{k \neq \pi^*(c)} P(\mu_n(\pi^*(c), c) < \mu_n(k, c))$. Then, if we can identify a rate function for $P(\mu_n(\pi^*(c), c) < \mu_n(k, c))$, then we can extend this with a minimum over the contexts and alternatives to find the rate function of the contextual PFS . The rest of the derivation is focused on analyzing the log of the moment generating function of $\mu_n(k, c)$, which requires a novel decomposition of the GP update formulas, and showing that it is asymptotically equivalent to that of the sample mean. This connects us back to the derivation in Reference [15], and we follow the steps therein to obtain the final result. The full proof is presented in the Appendix.

It is worth remarking that the large deviations rate function presented in Theorem 4.1 is identical to the rate function derived in Reference [9] using the sample mean estimator, which is the same as the rate function originally derived in Reference [15] with an additional minimum over the contexts. Our analysis shows that the same rate function is still applicable when the independent Gaussian distribution is replaced with a Gaussian process. This is not that surprising, since, as we see in the analysis, the effects of correlations in the model disappear as more observations are added, and two estimators are asymptotically equivalent. As a result, using a GP enables efficient inference by learning the similarities between contexts, 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 4.1, adapt the IKG policy from the literature 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)** [4] is a popular approach for maximizing the *PCS* asymptotically with independent Gaussian noise. Reference [9] later extended the OCBA approach [4, 5] to the contextual R&S problem, where the author derives the **Karush-Kuhn-Tucker (KKT)** conditions for maximizing the rate function (Equation (1)), and proposes an idealized sampling policy that iteratively realizes the KKT conditions. The idealized policy derived by Reference [9] (see Section III C of the article for derivation) relies on $F(k, c)$, $\sigma^2(k, c)$, 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 and is different than $p(k, c)$ used in the derivation to denote the idealized asymptotic allocation rate. For a practical algorithm, Reference [9] replaces $F(k, c)$ 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.

As shown in Theorem 4.1, using the GP model yields the same rate function as using the sample mean predictors in Reference [9], so 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 based on n observations collected so far, in place of $F(k, c)$ 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 7) 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, i.e., correlation between reward functions under

ALGORITHM 1: GP-C-OCBA for Contextual R&S

- 1: Use a pre-determined rule to allocate initial samples to each alternative. Let N_0 be the total number of initial observations used.
- 2: **for** $n = N_0, \dots, B - 1$ **do**
- 3: Update the GP model with the available observations, calculate $\mu_n(k, c), \Sigma_n(k, c), \pi^n(c)$.
- 4: For all $c \in C$ 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

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

- 5: Solve for $(\tilde{k}^*, \tilde{c}^*) = \arg \min_{k \neq \pi^n(c), c \in C} \zeta(k, c)$, 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:** $\pi^B(c) = \arg \max_k \mu_B(k, c), c \in C$ as the set of predicted best alternatives.

different contexts. An additional benefit of our approach over Reference [9] is in its applicability when the initial sampling budget is too small to draw multiple samples from each alternative. Using normal random variables to model each alternative-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, however, can be trained using very few samples for each alternative, rather than each alternative-context pair, thus the modified algorithm can be used even with a limited sampling budget.

A final remark about the GP-C-OCBA is that in the extreme case where there is only a single context, the sampling policy is equivalent to the OCBA-2 algorithm presented in Reference [22], which has been shown to achieve the optimal convergence rate in the standard R&S setting. Thus, GP-C-OCBA can be viewed as a principled extension of a rate optimal R&S algorithm to the contextual R&S setting.

5.2 Integrated Knowledge Gradient

On a related note, another applicable method for the contextual R&S problem is the IKG algorithm, which has been developed for the closely related problem of contextual Bayesian optimization. In this section, we adapt the IKG algorithm 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 [26, 27].

Knowledge Gradient (KG) [11] 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

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

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 alternative $k^* = \arg \max_k KG(k, c; c)$. To extend this to the contextual Bayesian optimization problem, References [8, 26, 27] each study an integrated (or

summed) version of KG, under slightly different problem settings, where either the context space or both alternative-context spaces are continuous. The main differences between these three works are in how they approximate and optimize the integrated KG factor 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

$$\text{IKG}(k, c) = \sum_{c' \in C} \text{KG}(k, c; c') w(c').$$

At each iteration, the IKG policy samples the alternative-context pair that maximizes the IKG factor, $(\tilde{k}^*, \tilde{c}^*) = \arg \max_{k \in \mathcal{K}, c \in C} \text{IKG}(k, c)$. The IKG policy is summarized in Algorithm 2.

ALGORITHM 2: IKG for Contextual R&S

- 1: Use a pre-determined rule to allocate initial samples to each alternative. Let N_0 be the total number of initial observations used.
- 2: **for** $n = N_0, \dots, B - 1$ **do**
- 3: Update the GP model with the available observations, calculate $\mu_n(k, c)$ and $\Sigma_n(c, c'; k)$.
- 4: For all $c, c' \in C$ and $k \in \mathcal{K}$, use Algorithm 1 of Reference [26] to calculate $\text{KG}(k, c; c')$ and set
- $$\text{IKG}(k, c) = \sum_{c' \in C} \text{KG}(k, c; c') w(c').$$
- 5: Solve for $(\tilde{k}^*, \tilde{c}^*) = \arg \max_{k \in \mathcal{K}, c \in C} \text{IKG}(k, c)$, and draw an additional sample from $(\tilde{k}^*, \tilde{c}^*)$.
- 6: **end for**
- 7: **Return:** $\pi^B(c) = \arg \max_k \mu_B(k, c), c \in C$ as the set of predicted best alternatives.

The main difficulty with using the IKG policy is its computational cost. In the finite alternative-context setting that we are working with, the $\text{KG}(k; c)$ can be computed exactly using Algorithm 1 from [11], which has a cost of $\mathcal{O}(K \log K)$ for any pair (k, c) , given $\mu_n(\cdot, \cdot)$ and $\Sigma_n(\cdot, \cdot)$. This translates to an $\mathcal{O}(|C|K \log K)$ cost for calculating the IKG factor for a given (k, c) . In total, to find the next pair to sample using IKG costs $\mathcal{O}(|C|^2 K^2 \log K)$ for calculating the IKG factors, and an additional $\mathcal{O}(n_k^3 + |C|^2 n_k + |C|n_k^2)$ (when the unchanged posterior terms are re-used between iterations) to calculate posterior mean and covariance matrices, where n_k denotes the total number of samples allocated to alternative k .

However, the cost of GP-C-OCBA is dominated by the cost of calculating the posterior mean and variance for each alternative-context pair, which has a total cost of $\mathcal{O}(n_k^3 + |C|n_k^2)$ (when the unchanged posterior terms are re-used between iterations). Note that we avoid the $|C|^2 n_k$ term, since our algorithm only requires the posterior variance, as well as the $\mathcal{O}(|C|^2 K^2 \log K)$ cost of IKG calculations. This puts GP-C-OCBA at a significant advantage in terms of computational complexity.

6 CONVERGENCE ANALYSIS

In this section, we analyze the convergence properties of the GP-C-OCBA algorithm, and show that it identifies $\pi^*(c)$ almost surely as the simulation budget $B \rightarrow \infty$. In addition, under the assumption of an independent prior distribution, we show that the allocation ratio produced by GP-C-OCBA converges almost surely to the optimal allocation ratio obtained by maximizing the rate function

in Theorem 4.1. Finally, we build on this result to show that the contextual PFS converges to zero at the optimal exponential rate.

6.1 Consistency of GP-C-OCBA

We prove the result under a set of mild assumptions.

ASSUMPTION 6.1.

- (1) *The best alternative, $\pi^*(c)$, is unique for all $c \in C$.*
- (2) *The observation noise is normally distributed with known variance, i.e., $\epsilon^n(k, c) \sim \mathcal{N}(0, \sigma^2(k, c))$ with known $\sigma^2(k, c)$, and $\epsilon^n(k, c)$ are independent across n, k, c .*
- (3) *The GP prior is fixed across iterations.*
- (4) *The prior correlation coefficient between any two contexts for any given alternative, $\text{Corr}_0(c, c'; k) := \frac{\Sigma_0(c, c'; k)}{\sqrt{\Sigma_0(k, c)\Sigma_0(k, c')}}$, is strictly between $[-1, 1]$, i.e., $-1 < \text{Corr}_0(c, c'; k) < 1$. In other words, $\Sigma_0(C, C; k)$ is a strictly positive definite matrix.*

The first three points are common assumptions from the literature that simplify the analysis. The known observation noise level is needed for exact GP inference, and the assumption of a fixed prior eliminates the need to study the hyper-parameters in the analysis. In practice, the form of the GP prior is kept fixed, however, the hyper-parameters of the GP are periodically updated to better fit the data, which is typically done via maximum likelihood estimation. Though showing this rigorously for an adaptive sampling policy is a daunting task, the hyper-parameters of the GP prior typically converge to some final value and remain fixed after that. The last point is merely technical, since in practice it would not make sense to consider two contexts that are assumed to be perfectly correlated. The following lemma shows that this implies that $\Sigma_n(C, C; k)$ must remain strictly positive definite. Note that since the posterior variance is a deterministic function of the sampling decisions (i.e., it is independent of the observations) the following statement holds for any sequence of sampling decisions.

LEMMA 6.2. *If $-1 < \text{Corr}_0(c, c'; k) < 1$, then $-1 < \text{Corr}_n(c, c'; k) < 1, \forall n \geq 0$.*

The proofs of this and the following statements are included in the online supplement. The posterior variance of a GP model at a given point decreases monotonically as we add more samples to the model. Thus, for a given alternative-context, the posterior variance in the limit is 0 if that pair gets sampled infinitely often, and is a strictly positive value otherwise. The lemma below shows this rigorously.

LEMMA 6.3. *Under Assumption 6.1, $\Sigma_n(k, c) \rightarrow 0$ almost surely if and only if $N_n(k, c) \rightarrow \infty$.*

The intuitive interpretation of Lemma 6.3 is that, despite modeling correlations between contexts, we cannot drive the uncertainty about the reward function of an alternative-context to zero, unless we collect infinitely many observations for that pair. This rules out the pathological cases where the statistical model treats a reward estimate as certain purely based on observations of other alternative-context pairs.

We will build on these results to show that GP-C-OCBA policy samples each alternative-context pair infinitely often. We will first establish that if a given context gets sampled infinitely often, all alternative-context pairs corresponding to that context must get sampled infinitely often.

LEMMA 6.4. *Under Assumption 6.1, using the GP-C-OCBA policy, for any $c \in C$,*

$$\lim_{n \rightarrow \infty} \sum_{k \in \mathcal{K}} N_n(k, c) = \infty \implies \lim_{n \rightarrow \infty} N_n(k, c) = \infty, \forall k \in \mathcal{K}, \text{ almost surely.}$$

If we only had a single context, then Lemma 6.4 would be sufficient to prove the consistency of the algorithm. However, with multiple contexts, we need to ensure that a strict subset of contexts cannot consume all observations after a certain iteration. The following theorem extends Lemma 6.4 to show that, as the budget goes to infinity, all alternative-context pairs are sampled infinitely often by GP-C-OCBA.

THEOREM 6.5. *Under Assumption 6.1, the GP-C-OCBA policy samples each alternative and each context infinitely often, i.e.,*

$$\lim_{n \rightarrow \infty} N_n(k, c) = \infty, \forall k \in \mathcal{K}, c \in \mathcal{C}, \text{ almost surely.} \quad (2)$$

Allocating infinitely many samples to each alternative-context pair would have been sufficient to prove consistency if we did not model any correlation between contexts. However, we are not aware of any general consistency results regarding the GP posterior mean, so we need to ensure that the predictions converge to the true reward function. The following lemma establishes this result.

LEMMA 6.6. *Under Assumption 6.1, if $\lim_{n \rightarrow \infty} N_n(k, c) = \infty$, then $\mu_n(k, c) \rightarrow F(k, c)$ almost surely.*

Coupled with Theorem 6.5, Lemma 6.6 shows that using GP-C-OCBA to allocate the observation budget, the posterior mean converges almost surely to the true reward for all alternative-context pairs. Thus, it follows that the predicted best alternative must almost surely converge to the correct best alternative for all contexts. The following corollary summarizes the result.

COROLLARY 6.7. *Under conditions of Theorem 6.5, GP-C-OCBA policy is strongly consistent, i.e., $\pi^B(c) \rightarrow \pi^*(c), \forall c$ almost surely as the observation budget $B \rightarrow \infty$.*

6.2 Convergence to the Optimal Allocation Ratio and the Convergence Rate

The results of the previous subsection show that, given enough samples, GP-C-OCBA is guaranteed to identify the optimal policy, $\pi^*(c)$. However, these are asymptotic results, and they do not provide any insight into how fast this convergence occurs. Under certain conditions on the prior covariance matrix, we can show that the allocation ratio produced by GP-C-OCBA converges to the optimal static allocation ratio, which is obtained by maximizing the rate function given in Theorem 4.1.

Let

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

and

$$\hat{\eta}_n(k, c) = \frac{(F(\pi^*(c), c) - F(k, c))^2}{\sigma^2(\pi^*(c), c)/\hat{p}_n(\pi^*(c), c) + \sigma^2(k, c)/\hat{p}_n(k, c)}.$$

The work of Reference [9] shows that the optimal static allocation ratio, i.e., the allocation ratio maximizing the large deviations rate function, satisfies the following conditions.

LEMMA 6.8. *(Lemma 1 of Reference [9]) The optimal static allocation ratio, $p^*(k, c)$, satisfies the following:*

$$\frac{p^*(\pi^*(c), c)^2}{\sigma^2(\pi^*(c), c)} = \sum_{k \neq \pi^*(c)} \frac{p^*(k, c)^2}{\sigma^2(k, c)}, \forall c \in \mathcal{C},$$

$$\eta(k, c) = \eta(k', c), \forall c \in \mathcal{C}, k \neq k' \neq \pi^*(c),$$

$$\eta(k, c) = \eta(k', c'), \forall c \neq c', k \neq \pi^*(c), k' \neq \pi^*(c').$$

The following result states that the allocation ratio produced by GP-C-OCBA converges to $p^*(k, c)$, i.e., that GP-C-OCBA asymptotically obtains the optimal static allocation ratio.

THEOREM 6.9. Suppose that Assumption 6.1 holds. Then,

$$\begin{aligned} \frac{\hat{p}_n(\pi^*(c), c)^2}{\sigma^2(\pi^*(c), c)} - \sum_{k \neq \pi^*(c)} \frac{\hat{p}_n(k, c)^2}{\sigma^2(k, c)} &\rightarrow 0, \forall c \in C, \\ \hat{\eta}_n(k, c) - \hat{\eta}_n(k', c) &\rightarrow 0, \forall c \in C, k \neq k' \neq \pi^*(c), \\ \hat{\eta}_n(k, c) - \hat{\eta}_n(k', c') &\rightarrow 0, \forall c \neq c', k \neq \pi^*(c), k' \neq \pi^*(c'), \end{aligned}$$

almost surely as $n \rightarrow \infty$. In other words, $\hat{p}_n(k, c)$ satisfies the conditions of Lemma 6.8 and converges to $p^*(k, c)$ as $n \rightarrow \infty$.

The proof is borrowed from the proof of Theorem 4 in Reference [9], where the authors proposed a similar algorithm as GP-C-OCBA. The difference between the two algorithms is that in Reference [9] they used sample mean and sample variance to estimate the true expected performance and observation noise variance, whereas we replace the sample mean with the posterior mean $\mu_n(k, c)$ and replace the sample variance with $N_n(k, c)\Sigma_n(k, c)$ for alternative k and context c . The validity of the proof relies on the convergence of the two posterior estimators. While the Gaussian posterior mean $\mu_n(k, c)$ is guaranteed to converge to the true value $F(k, c)$ almost surely as long as $N_n(k, c)$ goes to infinity (see Reference [11]), the convergence of $\Sigma_n(k, c)$ is more subtle, since the different components converge at different rates. This may be of independent interest for sequential sampling with GP model. We summarize the convergence result of posterior covariance matrix in Lemma 6.10.

LEMMA 6.10. Suppose that Assumption 6.1 holds. Then, $\Sigma_n(k, c) = \frac{\sigma(k, c)^2}{N_n(k, c)} + O(\frac{1}{N_n(k, c)}) \forall k \in \mathcal{K}, c \in C$, almost surely.

Lemma 6.10 is a stronger result of Lemma 6.3, since as $N_n(k, c)$ tends to infinity, Lemma 6.10 tells us that $\Sigma_n(k, c)$ converges to 0 at the rate $\frac{1}{N_n(k, c)}$ as well as the limit it converges to after dividing the rate. With the convergence of posterior estimators, we have the following Lemma 6.11, which guarantees that all the budget ratio assigned to any context-alternative $\hat{p}_n(k, c)$ is positive in the long run.

LEMMA 6.11. Suppose that Assumption 6.1 holds. Then, $\liminf_{n \rightarrow \infty} \hat{p}_n(k, c) > 0, \forall k \in \mathcal{K}, c \in C$ almost surely.

PROOF. We borrow the same proof of Lemmas 6–9 in Reference [9], where the estimator for $\mu(k, c)$ and $\sigma^2(k, c)$ is replaced by $\mu_n(k, c)$ and $N_n(k, c)\Sigma_n(k, c)$, respectively. \square

Although Theorem 6.9 shows that the sample allocation ratio of GP-C-OCBA converges to the optimal allocation ratio, this, by itself, does not say much about the actual convergence rate of the contextual PCS. In the following, we extend the convergence rate result presented by Reference [22] for the OCBA algorithm to the contextual setting. We show that the contextual PFS converges to 0 at the optimal exponential rate. To the best of our knowledge, this is the first convergence rate result in the literature for a contextual R&S algorithm.

THEOREM 6.12. Under the conditions of Theorem 6.9, and using an uninformative prior, i.e., $\Sigma_0(k, c) = \infty$, the contextual probability of false selection produced by GP-C-OCBA policy decreases at the optimal exponential rate, i.e.,

$$PFS_{\sim}^n \doteq e^{-\eta^* n/2}, \text{ where } \eta^* = \min_{c \in C} \min_{k \neq \pi^*(c)} \eta(k, c),$$

where PFS_{\sim}^n is either of PFS_E^n or PFS_M^n , and \doteq denotes logarithmic equivalence, i.e., for two sequences $a_n \doteq b_n \iff \lim_{n \rightarrow \infty} \frac{1}{n} \log \frac{a_n}{b_n} = 0$.

PROOF. Recall from the proof of Theorem 4.1 that PFS_{\sim}^n can be lower and upper bounded by a constant (1 and $|C|(|\mathcal{K}| - 1)$, respectively) multiple of

$$\max_{c \in C} \max_{k \neq \pi^*(c)} P(\mu_n(\pi^*(c), c) < \mu_n(k, c)).$$

With the prior as specified, the posterior mean is equal to the sample mean. In which case [3],

$$P(\mu_n(\pi^*(c), c) < \mu_n(k, c)) = \Phi\left(-\sqrt{\hat{\eta}_n(k, c)n}\right),$$

where $\Phi(\cdot)$ denotes the cumulative distribution function of the standard Gaussian distribution. Since for $x < 0$, $\Phi(x) \doteq \exp(-x^2/2)$ [17], we get

$$P(\mu_n(\pi^*(c), c) < \mu_n(k, c)) \doteq \exp(-\hat{\eta}_n(k, c)n/2).$$

By Theorem 6.9, $\hat{p}_n(k, c) \rightarrow p^*(k, c)$, which implies that $\hat{\eta}_n(k, c) \rightarrow \eta(k, c)$ as $n \rightarrow \infty$. Thus, putting it all together, we get

$$PFS_{\sim}^n \doteq \exp\left(-\left(\min_{c \in C} \min_{k \neq \pi^*(c)} \eta(k, c)\right)n/2\right). \quad \square$$

Note that Theorem 6.9 is an extension of the result originally proved for C-OCBA [9] and the convergence rate result in Theorem 6.12 also applies to C-OCBA under the assumption of known observation noise. This can be interpreted as saying that GP-C-OCBA and C-OCBA are asymptotically equivalent. A natural question in this case is whether GP-C-OCBA can be shown to have an advantage over C-OCBA under a finite budget setting. Though it is difficult to show this rigorously, we can argue for it in an informal way. By modeling the performance of an alternative-context with a Gaussian process, we implicitly assume that the contexts that are similar will have similar performances, with the particular measure of similarity being learned from the data. As a result, for a given alternative-context, by learning about the contexts that are similar to this context, we can refine our model of the contexts without having to evaluate the context itself. This is similar to the linearity assumptions that we see in the literature, e.g., in Reference [29]. In their case, there is an explicit assumption of linearity, which, if holds true, allows for evaluating only the extreme contexts to determine the policy for all contexts. Due to the assumed linearity, the performance of any given context can be estimated by solving the regression equations. There are many cases where linearity provides a good approximation, but there are many others where it does not hold true at all. Our modeling of similarity, as opposed to linearity, works in a similar way. It does not eliminate the need to evaluate all contexts. However, by allowing inference on one context's performance via other similar contexts' evaluations, it reduces the number of evaluations needed, leading to a better finite time performance. An example of this was shown in Figure 1, where we saw that the GP posterior mean produced predictions that resembled the true rewards much more closely, while the sample mean predictor required many more samples to refine the estimates.

6.3 Estimation for Contexts Outside C

In this section, we briefly discuss if a context $c' \notin C$ is given, how we estimate $F(k, c')$ and what statistical guarantee we can say about this estimate. As discussed in Section 3, we can use the same GP model by incorporating the point c' in the context set. Then, we can calculate the posterior estimator for any alternative-context pair (k, c') in the same way as

$$\mu_n(k, c') = \mu_0(k, c') + \Sigma_0(c', D_n(k); k)A_n^{-1}(k)(Y_n(k) - \mu_0(D_n(k)))^\top$$

and

$$\Sigma_n(c', c'; k) = \Sigma_0(c', c'; k) - \Sigma_0(c', D_n(k); k)A_n^{-1}(k)\Sigma_0(D_n(k), c'; k).$$

Since no simulation is run under context c' , $\mu_n(k, c')$ cannot converge to $F(k, c')$. This can also be reflected in the posterior variance $\Sigma_n(c', c'; k)$, which represents the confidence level for reward estimation. The following result quantifies such confidence level in the asymptotic sense. It is a byproduct from the proof of Lemma 6.10.

COROLLARY 6.13. *Suppose Assumption 6.1 holds. Given any context c' , not necessarily in C ,*

$$\lim_{n \rightarrow \infty} \Sigma_n(c', c'; k) = \Sigma_0(c', c'; k) - \Sigma_0(c', C; k)(\Sigma_0(C, C; k))^{-1}\Sigma_0(C, c'; k) \quad \text{almost surely.}$$

The expression on the right hand side provides us with some intuition. Let $Z(C')$ be a multivariate normal random vector following $\mathcal{N}(\mu_0(k, C'), \Sigma_0(C', C'; k))$, where $C' = C \cup \{c'\}$. Then, the posterior distribution for (k, c') converges to a normal distribution with variance being the same as that of the conditional distribution $Z(c')|Z(C)$. This is because as $n \rightarrow \infty$, we gather sufficient information for each (k, c) with $c \in C$. The posterior distribution for (k, C) will concentrate on the true value $F(k, C)$, and hence the posterior distribution for (k, c') converges to the marginal distribution of $Z(c')|Z(C)$ as $Z(C) = F(k, C)$. In particular, if $c' \in C$, then we have $\Sigma_n(c', c'; k)$ converges to 0, which coincides with Lemma 6.2 and Lemma 6.10. If $c' \notin C$ and $\Sigma_0(C', C'; k)$ is positive definite, then $\Sigma_n(c', c'; k)$ cannot converge to 0, meaning the uncertainty level remains positive, and the value of this uncertainty level is determined by the prior correlation, which usually depends on the distance from c' to C .

7 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 Reference [21] (DSCO), Reference [9] (C-OCBA), and with the IKG algorithm as described in Section 5.2 as well as with its cheaper approximation LEVI as presented in Reference [25]. In addition, we also compare against a modified version of the two-stage algorithms (TS and TS+) proposed by Reference [29]. Both TS and TS+ are designed for the fixed confidence setting under a linearity assumption on the reward function. We adopt them by keeping the allocation ratios suggested by the algorithms but scaling down the number of samples to the given budget. As recommended by the authors, we use the extreme design for both TS and TS+, with the extreme design being restricted to the points in the context set. We chose these benchmarks, since DSCO and C-OCBA were both proposed for the contextual R&S with the finite alternative-context setting that is studied in this article 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 [1] with the default priors. We used the Matern 5/2 kernel in all experiments, and trained the GP hyper-parameters every 10 iterations using the `fit_gpytorch_mll` routine, which uses the L-BFGS-B algorithm to maximize the *a posteriori* likelihood. The code is available at https://github.com/saitcakmak/contextual_rs.

7.1 Synthetic Test Functions

For the experiments, we generate the true rewards, $F(k, c)$, 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 alternatives, i.e., each alternative corresponds to a fixed value of x_1 , and spread the alternatives evenly across the corresponding domain. The remaining input dimensions are used for the contexts, and thus contexts are $d - 1$ dimensional vectors for a d dimensional test function. Put together, this corresponds to $F(k, c) = f(x_k, x_c)$ 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 1,000 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 alternatives 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 alternative-context pair for the initialization phase. For TS and TS+, the extreme design consists of the smallest and largest context values, with each receiving 10 samples from each alternative 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 Griewank function, using 10 alternatives and 20 contexts. We use the expected *PCS* with uniform weights and the worst-case *PCS*, and initialize with 2 samples from each alternative-context pair. For TS and TS+, the extreme design consists of the smallest and largest context values, with each receiving 20 samples from each alternative for the initialization phase.

- The 3D Hartmann function, evaluated on $[0, 1]^3$:

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

where the constants α , A , and P are given in Reference [30]. We run a single experiment with 20 alternatives and 20 contexts, using the expected *PCS* with uniform weights. Since the number of alternative-context pairs is quite large in this experiment, we select only 6 contexts for each alternative, uniformly at random, and draw a single sample from these contexts for the initial stage. Due to insufficient initial sampling budget, we do not run DSCO and C-OCBA for this problem. Since it is difficult to define the extreme design over a discrete set of points in two dimensions, we skip TS and TS+ as well and only run the GP-based algorithms for this problem.

- 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 alternatives and 40 contexts. For the initial stage, we randomly select 16 contexts for each alternative, and draw a single sample from these contexts, which is again due to the large number of alternative-context pairs in this experiment. Similar to the previous experiment, due to insufficient initial sampling budget, we do not run DSCO and C-OCBA for this problem. Since it is difficult to define the extreme design over a discrete set of points in 7 dimensions, we skip TS and TS+ as well and only run the GP-based algorithms for this problem.

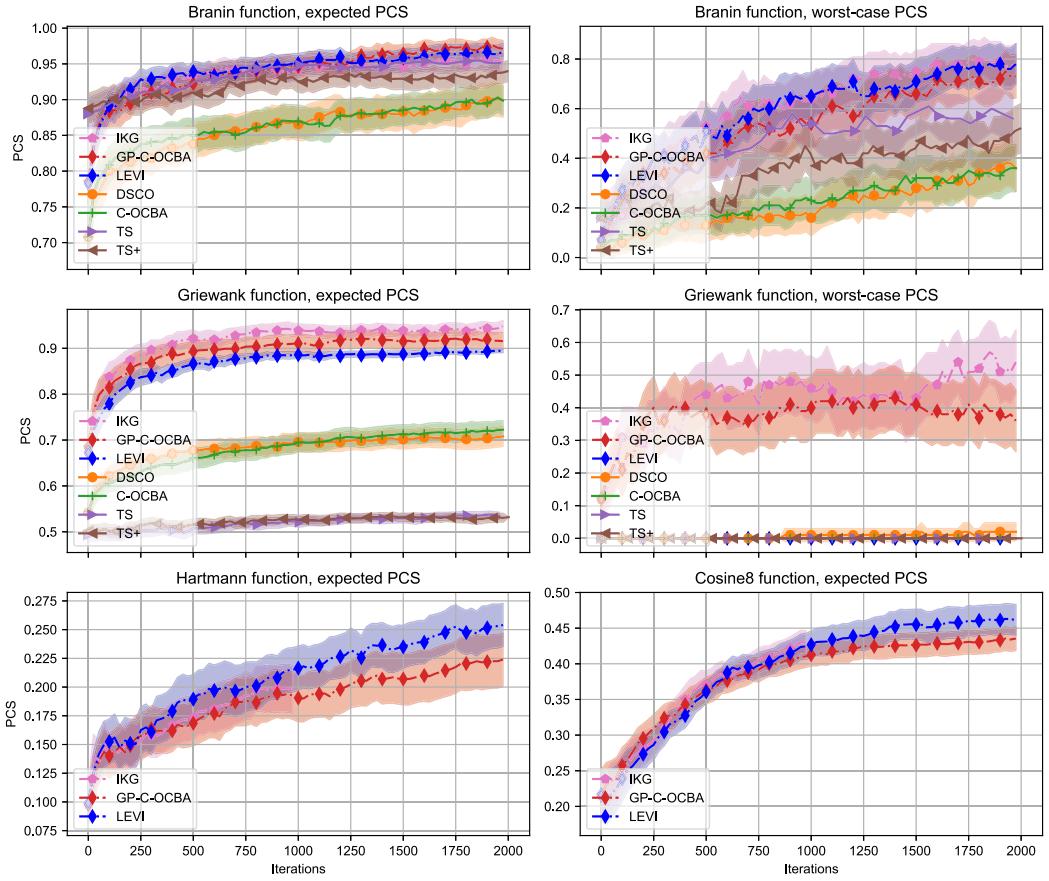


Fig. 2. Experiments using Branin function with PCS_E and PCS_M , Griewank function with PCS_E and PCS_M , Hartmann function with PCS_E , and Cosine8 with PCS_E . The plots show the empirical contextual PCS on the y-axis, and the number of iterations/samples (post-initialization) on the x-axis. The shaded area denotes the 95% confidence interval.

7.2 Results of Synthetic Test Functions

The experiment results are plotted in Figure 2. We ran each experiment for 2,000 iterations, except for IKG in Hartmann and Cosine8 functions, which were run for 1,000 iterations due to their excessive cost. The plots show the empirical contextual PCS , estimated using 100 replications. The first four plots compare all algorithms. However, the last two only compare GP-C-OCBA, LEVI and IKG, due to small initial budget preventing drawing of multiple samples from each alternative-context pair, which is necessary to form an initial estimate of sample mean and variance that is used by DSCO and C-OCBA, as well as the difficulty of setting up the extreme design for the two-stage algorithms. In all four of the experiments comparing all algorithms, we see that the algorithms using the GP models achieve the highest contextual PCS , with the exception of LEVI, which performs poorly on worst-case PCS . This demonstrates the benefit of using a statistical model that leverages the hidden correlation structure in the reward function, and also highlights the importance of using a sampling strategy that provides a good coverage of the search space. In particular, for the worst-case PCS , we see that the DSCO and C-OCBA perform significantly worse in both problems, with the performance of TS and TS+ also diminishing significantly. For TS and TS+, this

Table 1. Comparison of Computational Cost of IKG, LEVI, and GP-C-OCBA

Algorithm	Branin PCS_E/PCS_M	Griewank PCS_E/PCS_M	Hartmann	Cosine8
IKG	1,218	3,807	11,096	43,224
LEVI	295	433	576	683
GP-C-OCBA	249	293	441	544

We report the average wall-clock time, in seconds, for running 1,000 iterations of the given experiment. The experiments were run on a shared cluster using four cores of the allocated CPU. To save on space, we report the average run-time (in seconds) of the two objectives for Branin and Griewank.

clearly demonstrates the weakness of using a linear model when the underlying function is non-linear, which leads to 0 worst-case PCS in the Griewank function. Similarly, the poor performance of DSCO and C-OCBA in the worst-case PCS is explained by a total of 2,400 samples being far from sufficient to form reliable estimates for 200 alternative-context pairs using an independent statistical model. On a related note, although a comparison with opportunity cost as the performance measure is beyond the scope of this article, we include the result for opportunity cost in Appendix B. Both IKG and LEVI were originally designed to minimize opportunity cost, and it is not surprising that they outperform GP-C-OCBA when transitioning to the performance measure of opportunity cost. It is worthwhile to use different methods for different performance measures.

Although in some experiments GP-C-OCBA is slightly trailing behind IKG, which attributes to its asymptotic property, we see that GP-C-OCBA is highly competitive against IKG, while having significantly smaller computational complexity. As a cheaper alternative to IKG, LEVI provides good performance on two of the problems, while falling behind in others, particularly when considering the worst-case PCS . The wall-clock times for the experiments are reported in Table 1. We see that even in the experiments with smallest number of alternatives and contexts, 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 run time of LEVI is slightly longer than that of GP-C-OCBA, which makes it a good alternative in settings where it shows good performance. The reported run times are for 1,000 iterations of a full experiment, and include the cost of fitting the GP model, which is identical for all algorithms. It is worth noting that the cost of evaluating the test functions in these experiments is insignificant compared to the running time. As the cost of function evaluation increases, the results would look nicer for IKG, though it would still remain the more expensive alternative.

Overall, the experiments show that GP-C-OCBA is highly competitive in terms of sampling efficiency and computational cost. We believe that this makes GP-C-OCBA an attractive option for any practitioner that is faced with a contextual R&S problem.

7.3 Extension to Continuous Context Spaces

In this article, we have focused on the setting of discrete context spaces and developed the GP-C-OCBA algorithm under this assumption. However, there are applications where a user might be interested in using our approach with a continuous context set. In this section, we describe a simple extension of GP-C-OCBA to support continuous contexts, and present results comparing it with random sampling and LEVI on several synthetic test problems. We use the same problems from Section 7.2 with the Expected PCS objective. For each problem, we use the first input dimension for the alternatives with the same number of alternatives as before, and use the remaining input dimensions as the continuous context set.

As defined in Algorithm 1, GP-C-OCBA cannot be used with continuous contexts. The presence of an infinite number of contexts will lead to $\hat{p}(k, c)$ evaluating to zero with probability one, preventing us from making a meaningful comparison in step 5. To overcome this issue, we use

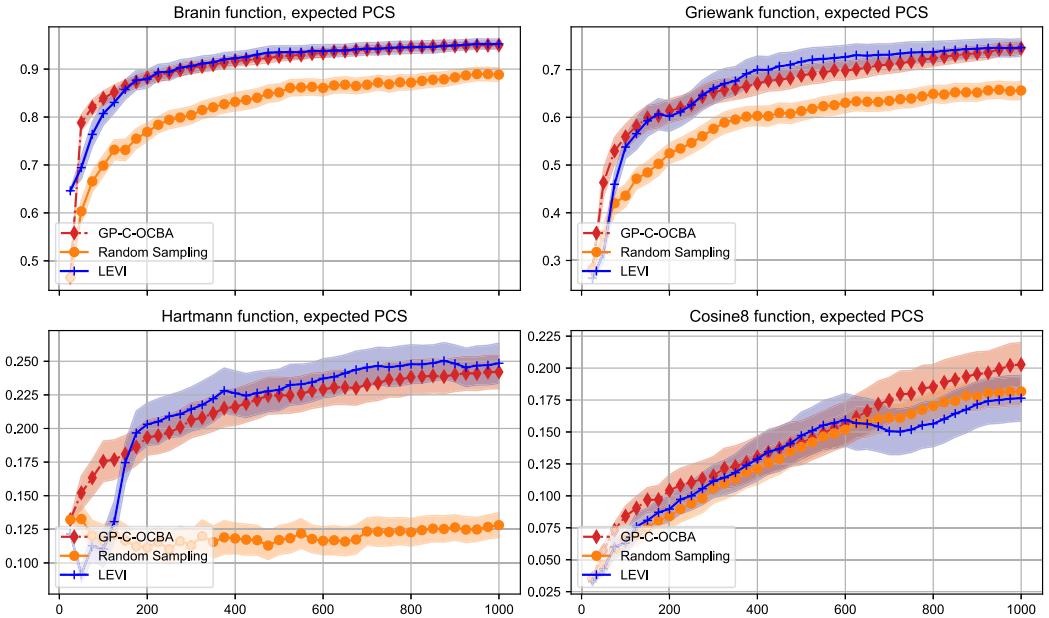


Fig. 3. Continuous context experiments using Branin, Griewank, Hartmann, and Cosine8 functions. The plots show the empirical expected *PCS* on the y-axis, and the number of iterations/samples (post-initialization) on the x-axis. The shaded area denotes the 95% confidence interval.

Table 2. Comparison of Computational Cost of Random Sampling, LEVI, and GP-C-OCBA

Algorithm	Branin	Griewank	Hartmann	Cosine8
Random Sampling	267	281	618	523
LEVI	3050	2651	11510	19300
GP-C-OCBA	3084	1445	6802	4260

We report the average wall-clock time, in seconds, for running 1,000 iterations of the given experiment. The experiments were run on a shared cluster using four cores of the allocated CPU.

kernel density estimation, defined as $\hat{p}(k, c) = \sum_{i=1}^n e^{-\|c - c_i\|_2} \mathbb{1}\{k = k_i\}$, which values each observation from a context proportional to its proximity to the context under consideration. With this modification, the rest of the algorithm is followed as is. For optimizing both GP-C-OCBA (minimizing $\zeta(k, c)$) and LEVI in this finite alternative - continuous context space, we follow a two step approach. First, we solve the continuous optimization problem $\min_c [\min_{k \neq \pi^n(c)} \zeta(k, c)]$ (and the corresponding maximization problem for LEVI) to find the context, then we find the corresponding alternative by fixing the context.

The results are shown in Figure 3 with the run times reported in Table 2. Of the four problems, we see LEVI and GP-C-OCBA performing quite similarly on the first two, and each one performing better than the other in one of the remaining problems. Overall, we see the *PCS* curves flattening out towards the end of the experiments, which is explained by both algorithms getting greedy and allocating a large portion of the budget on the predicted best alternatives. In addition, for the higher dimensional experiments, the expected *PCS* values are rather small, which highlights the difficulty of learning the best alternative across a high dimensional context space. Except for the Branin experiment, LEVI is considerably more expensive to run than GP-C-OCBA, which again highlights

GP-C-OCBA as a low cost, sample effective algorithm. It is also worth noting that the extension presented in this section is a very simple extension of GP-C-OCBA to continuous contexts. It is quite possible that there are other extensions that work much better, however a theoretically grounded study of such extensions is beyond the scope of this work.

8 CONCLUSION

We studied the contextual R&S problem under finite alternative-context setting, using a separate GP to model the reward for each alternative. 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. We proved consistency of GP-C-OCBA. We showed that its allocations converge to the optimal allocation ratio and the resulting contextual PFS converges to 0 at the optimal exponential rate. In numerical experiments with both finite and continuous contexts space, GP-C-OCBA was shown to achieve significant sampling efficiency, while having a significantly smaller computational overhead compared to other competitive alternatives. A simple extension of GP-C-OCBA to the continuous context space is also presented, which shows promising results and motivates future work in this direction.

In this work, we have followed a common practice in the R&S literature of developing sampling strategies with a focus on maximizing the large deviations rate function. An equally valuable approach, popular in the Bayesian optimization literature, is to develop algorithms based on a myopic optimality criterion. Although most acquisition functions focus on maximizing a reward (or minimizing regret), an acquisition function based on 0–1 loss could be developed to maximize the PCS. Development of acquisition functions such as opportunity cost (see References [6, 12, 14]) presents an interesting direction for future work.

APPENDICES

A PROOF OF THEOREM 4.1

With n denoting the total number of observations/samples, let $p(k, c)$ and $N_n(k, c) = np(k, c)$ 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 the sake of simplicity, we leave implicit the dependency of the probabilities and other quantities on the sampling policy. We start with the following lemma, which is crucial in proving the theorem.

LEMMA A.1. *Under assumptions of Theorem 4.1, for $k \neq \pi^*(c)$,*

$$\lim_{n \rightarrow \infty} \frac{1}{n} \log P(\mu_n(\pi^*(c), c) < \mu_n(k, c)) = -G_{(k, c)}(p(\pi^*(c), c), p(k, c)),$$

where

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

PROOF. We will follow the analysis of Reference [15] and use the Gartner-Ellis Theorem [7] 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 need to study the limiting behavior of the log **moment generating function (MGF)**:

$$\Lambda_n(\lambda; k, c) = \log \mathbb{E}[\exp(\lambda \mu_n(k, c))].$$

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

$$\mu_n(k, c) = \mu_0(k, c) + \sum_{i=1}^{|C|} [\Sigma_n^{i-1}(c, c_i; k)]_{1 \times N_n(k, c_i)} (A^i)^{-1} (Y_n(k, c_i) - [\mu_n^{i-1}(k, c_i)]_{N_n(k, c_i) \times 1}),$$

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

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

with $\Sigma_n^0(\cdot, \cdot; k) = \Sigma_0(\cdot, \cdot; k)$. The inverse of A^i can be calculated in closed form using the Sherman-Morrison formula [23]. After some algebra, we can rewrite $\mu_n(k, c)$ as follows:

$$\mu_n(k, c) = \mu_0(k, c) + \sum_{i=1}^{|C|} \frac{N_n(k, c_i) (\bar{Y}_n(k, c_i) - \mu_n^{i-1}(k, c_i)) \Sigma_n^{i-1}(c, c_i; k)}{\sigma^2(k, c_i) + N_n(k, c_i) \Sigma_n^{i-1}(c_i, c_i; k)}, \quad (3)$$

where $\bar{Y}_n(k, c_i)$ denotes the average of the observations. Similarly, we can rewrite $\Sigma_n^i(c, c'; k)$ as

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

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 $y(k, c) \sim \mathcal{N}(F(k, c), \sigma^2(k, c))$ 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

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

where

$$C_n(k, c, i) = \frac{N_n(k, c_i) \Sigma_n^{i-1}(c, c_i; k)}{\sigma^2(k, c_i) + N_n(k, c_i) \Sigma_n^{i-1}(c_i, c_i; k)}.$$

Let $\Lambda_n(\lambda_{\pi^*(c)}, \lambda_k; c)$ denote the log-MGF of $Z_n = (\mu_n(\pi^*(c), c), \mu_n(k, c))$. To use the Gartner-Ellis Theorem, we need to establish the limiting behavior of $\frac{1}{n}\Lambda_n(n\lambda_{\pi^*(c)}, n\lambda_k; c)$.

$$\lim_{n \rightarrow \infty} \frac{1}{n} \Lambda_n(n\lambda_{\pi^*(c)}, n\lambda_k; c) = \sum_{\kappa \in (\pi^*(c), k)} \lim_{n \rightarrow \infty} \mathbb{E}[\mu_n(\kappa, c)] \lambda_\kappa + \lim_{n \rightarrow \infty} \frac{n \text{Var}(\mu_n(\kappa, c)) \lambda_\kappa^2}{2}. \quad (4)$$

Let us start with the variance term. In the following, we use \rightarrow to denote the limit as $n \rightarrow \infty$ and $\stackrel{\sim}{\rightarrow}$ to denote equivalence in the limit. Note that

$$\text{Var}(\mu_n(k, c)) = \sum_{i=1}^{|C|} \frac{\sigma^2(k, c_i) C_n(k, c, i)^2}{N_n(k, c_i)}.$$

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 $\Sigma_n^i(\cdot, \cdot; k)$. Let us analyze $\text{Var}(\mu_n(\pi^*(c), c))$, 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 C . Note that

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

which implies that $\Sigma_n^i(\cdot, c_1; k) \rightarrow 0$, $i \geq 1$, and $C_n(k, c', i) \rightarrow \frac{\Sigma_0(c', c_i; k)}{\Sigma_0(c_i, c_i; k)}$ if $i = 1$ and $C_n(k, c', i) \rightarrow 0$ otherwise. Thus, we can ignore the rest of the terms in the summation and write

$$Var(\mu_n(k, c)) \xrightarrow{\sim} \frac{\sigma^2(k, c)}{N_n(k, c)} = \frac{\sigma^2(k, c)}{np(k, c)}.$$

Similarly for the expectation term, using $c_1 = c$, since $\Sigma_n^i(\cdot, c_1; k) \rightarrow 0$, $i \geq 1$, in the limit Equation (3) becomes

$$\mu_n(k, c) \xrightarrow{\sim} \mu_0(k, c) + (\bar{Y}_n(k, c_1) - \mu_n^0(k, c_1)) = \bar{Y}_n(k, c),$$

which implies that $\lim_{n \rightarrow \infty} \mathbb{E}[\mu_n(k, c)] = F(k, c)$. We are now ready to continue from Equation (4). Let $\Lambda^t(\lambda_k; k, c)$ denote the log-MGF of the observation $y(k, c) \sim \mathcal{N}(F(k, c), \sigma^2(k, c))$:

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

which is the exact term in Lemma 1 of Reference [15]. 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 $I^t(x_k; k, c) = \frac{(x_k - F(k, c))^2}{2\sigma^2(k, c)}$ is the Fenchel-Legendre transform of $\Lambda^t(\lambda_k/p(k, c); k, c)$. With the rate function of Z_n established, Reference [15] shows that

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

where the infimum can be calculated via differentiation [9], giving us

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

With the lemma established, the theorem can be proved as follows.

PROOF OF THEOREM 4.1. For a given context c , the probability of false selection after n observations $PFS^n(c) = 1 - PCS^n(c)$ is given by

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

We can lower and upper bound this, respectively, by

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

If for $k \neq \pi^*(c)$,

$$\lim_{n \rightarrow \infty} \frac{1}{n} \log P(\mu_n(\pi^*(c), c) < \mu_n(k, c)) = -G_{(k, c)}(p(\pi^*(c), c), p(k, c))$$

for some rate function $G_{(k, c)}$, then

$$\lim_{n \rightarrow \infty} \frac{1}{n} \log PFS^n(c) = - \min_{k \neq \pi^*(c)} G_{(k, c)}(p(\pi^*(c), c), p(k, c)).$$

Similarly, both the PFS_E^n and PFS_M^n can be lower and upper bounded by $\max_c PFS^n(c)$ and $(|\mathcal{K}| - 1) \max_c PFS^n(c)$ (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

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

where PFS_{\sim}^n is either of PFS_E^n or PFS_M^n . Combining this with Lemma A.1, we get the result

$$\lim_{n \rightarrow \infty} \frac{1}{n} \log PFS_{\sim}^n = - \min_{c \in C} \min_{k \neq \pi^*(c)} \frac{(F(\pi^*(c), c) - F(k, c))^2}{2(\sigma^2(\pi^*(c), c)/p(\pi^*(c), c) + \sigma^2(k, c)/p(k, c))}. \quad \square$$

B ADDITIONAL NUMERICAL RESULT FOR OPPORTUNITY COST

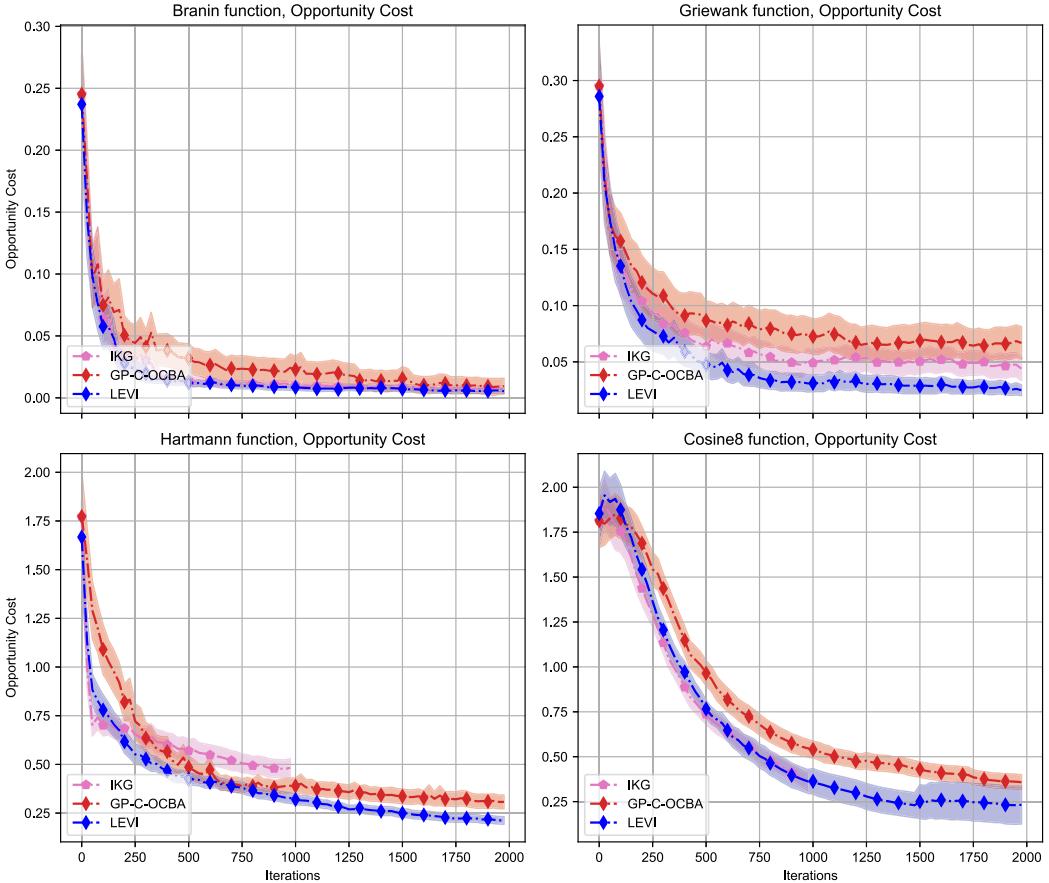


Fig. 4. Experiments on Benchmark functions with opportunity cost.

When using Opportunity Cost as the performance measure, Figure 4 indicates IKG and LEVI outperform GP-C-OCBA in most scenarios. It is worthwhile to use different methods for different performance measures.

REFERENCES

[1] Maximilian Balandat, Brian Karrer, Daniel R. Jiang, Samuel Daulton, Benjamin Letham, Andrew Gordon Wilson, and Eytan Bakshy. 2020. BoTorch: A framework for efficient Monte Carlo Bayesian optimization. In *Advances in Neural Information Processing Systems 33*. Curran Associates, Red Hook, NY, 21524–21538.

- [2] Chun-Hung Chen, Stephen E. Chick, Loo Hay Lee, and Nugroho A. Pujowidianto. 2015. Ranking and selection: Efficient simulation budget allocation. In *Handbook of Simulation Optimization*, Michael C. Fu (Ed.). Springer, New York, NY, 45–80.
- [3] Chun-Hung Chen and Loo Hay Lee. 2010. *Stochastic Simulation Optimization*. World Scientific, Singapore. <https://doi.org/10.1142/7437>
- [4] Chun-Hung Chen, Jianwu Lin, Enver Yücesan, and Stephen E. Chick. 2000. Simulation budget allocation for further enhancing the efficiency of ordinal optimization. *Discrete Event Dynam. Syst.* 10, 3 (2000), 251–270.
- [5] Ye Chen and Ilya O. Ryzhov. 2019. Complete expected improvement converges to an optimal budget allocation. *Adv. Appl. Probabil.* 51, 1 (2019), 209–235. <https://doi.org/10.1017/apr.2019.9>
- [6] Stephen E. Chick and Yaozhong Wu. 2005. Selection procedures with frequentist expected opportunity cost bounds. *Operat. Res.* 53, 5 (2005), 867–878.
- [7] Amir Dembo and Ofer Zeitouni. 1998. *Large Deviations Techniques and Applications* (2nd ed.). Springer-Verlag, Berlin.
- [8] Liang Ding, L. Jeff Hong, Haihui Shen, and Xiaowei Zhang. 2021. Technical note—Knowledge gradient for selection with covariates: Consistency and computation. *Naval Res. Log.* 69, 3 (2021), 496–507.
- [9] Jianzhong Du, Siyang Gao, and Chun-Hung Chen. 2022. Rate-Optimal Contextual Ranking and Selection. Retrieved from <https://arxiv.org/abs/2206.12640>. <https://doi.org/10.48550/ARXIV.2206.12640>
- [10] Qing Feng, Ben Letham, Hongzi Mao, and Eytan Bakshy. 2020. High-dimensional contextual policy search with unknown context rewards using Bayesian optimization. In *Advances in Neural Information Processing Systems 33*. Curran Associates, Red Hook, NY, 22032–22044.
- [11] Peter Frazier, Warren Powell, and Savas Dayanik. 2009. The knowledge-gradient policy for correlated normal beliefs. *INFORMS J. Comput.* 21, 4 (2009), 599–613.
- [12] Siyang Gao and Weiwei Chen. 2015. Efficient subset selection for the expected opportunity cost. *Automatica* 59 (2015), 19–26.
- [13] Siyang Gao, Jianzhong Du, and Chun-Hung Chen. 2019. Selecting the optimal system design under covariates. In *Proceedings of the 15th International Conference on Automation Science and Engineering*.
- [14] Siyang Gao and Leyuan Shi. 2014. An optimal opportunity cost selection procedure for a fixed number of designs. In *Proceedings of the Winter Simulation Conference*. IEEE, 3952–3958.
- [15] Peter Glynn and Sandeep Juneja. 2004. A large deviations perspective on ordinal optimization. In *Proceedings of the Winter Simulation Conference*. IEEE, Piscataway, NJ, 577–585.
- [16] Travis Goodwin, Jie Xu, Nurcin Celik, and Chun-Hung Chen. 2022. Real-time digital twin-based optimization with predictive simulation learning. *J. Simul.* (2022), 1–18.
- [17] Robert D. Gordon. 1941. Values of mills' ratio of area to bounding ordinate and of the normal probability integral for large values of the argument. *Ann. Math. Stat.* 12, 3 (1941), 364–366.
- [18] Cheng Guo and Felix Berkhahn. 2016. Entity embeddings of categorical variables. Retrieved from <https://arxiv:1604.06737>
- [19] Xiao Jin, Haobin Li, and Loo Hay Lee. 2019. Optimal budget allocation in simulation analytics*. In *Proceedings of the 15th International Conference on Automation Science and Engineering*.
- [20] Seong-Hee Kim and Barry L. Nelson. 2007. Recent advances in ranking and selection. In *Proceedings of the Winter Simulation Conference*. IEEE, Piscataway, NJ, 162–172.
- [21] Haidong Li, Henry Lam, Zhe Liang, and Yijie Peng. 2020. Context-dependent ranking and selection under a Bayesian framework. In *Proceedings of the Winter Simulation Conference*. IEEE, Piscataway, NJ, 2060–2070.
- [22] Yanwen Li and Siyang Gao. 2021. On the convergence of optimal computing budget allocation algorithms. In *Proceedings of the Winter Simulation Conference*. IEEE, Piscataway, NJ.
- [23] Carl D. Meyer. 2000. *Matrix Analysis and Applied Linear Algebra*. Society for Industrial and Applied Mathematics, Philadelphia, PA.
- [24] Maria Augusta S. N. Nunes and Rong Hu. 2012. Personality-based recommender systems: An overview. In *Proceedings of the 6th ACM Conference on Recommender Systems*.
- [25] Michael Pearce and Juergen Branke. 2017. Efficient expected improvement estimation for continuous multiple ranking and selection. In *Proceedings of the Winter Simulation Conference*. IEEE, Piscataway, NJ, 2161–2172.
- [26] Michael Pearce and Juergen Branke. 2018. Continuous multi-task Bayesian optimisation with correlation. *Eur. J. Operat. Res.* 270, 3 (2018), 1074–1085.
- [27] Michael Pearce, Janis Klaise, and Matthew Groves. 2020. Practical Bayesian optimization of objectives with conditioning variables. Retrieved from <https://arxiv:2002.09996>
- [28] Giulia Pedrielli, K Selcuk Candan, Xilun Chen, Logan Mathesen, Alireza Inanalouganji, Jie Xu, Chun-Hung Chen, and Loo Hay Lee. 2019. Generalized ordinal learning framework (golf) for decision making with future simulated data. *Asia-Pacific J. Operat. Res.* 36, 06 (2019), 1940011.

- [29] Haihui Shen, L. Jeff Hong, and Xiaowei Zhang. 2021. Ranking and selection with covariates for personalized decision making. *INFORMS J. Comput.* 33, 4 (2021), 1500–1519.
- [30] S. Surjanovic and D. Bingham. 2013. Hartmann 3-dimensional function. In *Virtual Library of Simulation Experiments*. Retrieved from <https://www.sfu.ca/~ssurjano/hart3.html>

Received 12 January 2022; revised 10 October 2023; accepted 8 November 2023