ABSTRACT

Title of dissertation: TOPICS IN STOCHASTIC OPTIMIZATION

Guowei Sun
Doctor of Philosophy, 2019

Dissertation directed by: Professor Michael Fu
Department of Decision, Operations,
and Information Technologies

In this thesis, we work with three topics in stochastic optimization: ranking and selection (R&S), multi-armed bandits (MAB) and stochastic kriging (SK). For R&S, we first consider the problem of making inferences about all candidates based on samples drawn from one. Then we study the problem of designing efficient allocation algorithms for problems where the selection objective is more complex than the simple expectation of a random output. In MAB, we use the autoregressive process to capture possible temporal correlations in the unknown reward processes and study the effect of such correlations on the regret bounds of various bandit algorithms. Lastly, for SK, we design a procedure for dynamic experimental design for establishing a good global fit by efficiently allocating simulation budgets in the design space.

The first two Chapters of the thesis work with variations of the R&S problem. In Chapter 1, we consider the problem of choosing the best design alternative under a small simulation budget, where making inferences about all alternatives from a single observation could enhance the probability of correct selection. We propose a new selection rule

exploiting the relative similarity between pairs of alternatives and show its improvement on selection performance, evaluated by the Probability of Correct Selection, compared to selection based on collected sample averages. We illustrate the effectiveness by applying our selection index on simulated R&S problems using two well-known budget allocation policies. In Chapter 2, we present two sequential allocation frameworks for selecting from a set of competing alternatives when the decision maker cares about more than just the simple expected rewards. The frameworks are built on general parametric reward distributions and assume the objective of selection, which we refer to as utility, can be expressed as a function of the governing reward distributional parameters. The first algorithm, which we call utility-based OCBA (UOCBA), uses the Δ-technique to find the asymptotic distribution of a utility estimator to establish the asymptotically optimal allocation by solving the corresponding constrained optimization problem. The second, which we refer to as utility-based value of information (UVoI) approach, is a variation of the Bayesian value of information (VoI) techniques for efficient learning of the utility. We establish the asymptotic optimality of both allocation policies and illustrate the performance of the two algorithms through numerical experiments.

Chapter 3 considers the restless bandit problem where the rewards on the arms are stochastic processes with strong temporal correlations that can be characterized by the well-known stationary autoregressive-moving-average time series models. We argue that despite the statistical stationarity of the reward processes, a linear improvement in cumulative reward can be obtained by exploiting the temporal correlation, compared to policies that work under the independent reward assumption. We introduce the notion of temporal exploration-exploitation trade-off, where a policy has to balance between learning more recent information to track the evolution of all reward processes and utilizing currently available predictions to gain better immediate reward. We prove a regret lower bound characterized by the bandit problem complexity and correlation strength along the time index and propose policies that achieve a matching upper bound.

Lastly, Chapter 4 proposes a fully sequential experimental design procedure for the stochastic kriging (SK) methodology of fitting unknown response surfaces from simulation experiments. The procedure first estimates the current SK model performance by jackknifing the existing data points. Then, an additional SK model is fitted on the jackknife error estimates to capture the landscape of the current SK model performance. Methodologies for balancing exploration and exploitation trade-off in Bayesian optimization are employed to select the next simulation point. Compared to existing experimental design procedures relying on the posterior uncertainty estimates from the fitted SK model for evaluating model performance, our method is robust to the SK model specifications. We design a dynamic allocation algorithm, which we call kriging-based dynamic stochastic kriging (KDSK), and illustrate its performance through two numerical experiments.

Topics in Stochastic Optimization

by

Guowei Sun

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2019

Advisory Committee:
Professor Michael Fu, Chair/Advisor
Professor Steve Marcus
Professor Eric Slud
Professor Maria Cameron
Professor Ilya Ryzhov

Dedication

To my family.

Acknowledgments

I owe my gratitude to all the people who have made this thesis possible and because of whom my graduate experience has been one that I will cherish forever.

First and foremost I'd like to thank my advisor, Professor Michael Fu, for his guidance and patience throughout the years. He is not only an amazing advisor, but also an excellent person. The thesis is only a tiny reflection of his lessons for me.

I would also like to thank Professor Steve Marcus, Professor Ilya Ryzhov, Professor Maria Cameron and Professor Eric Slud for agreeing to serve on my thesis committee. They are the true living embodiment of professorship. I truly enjoyed taking classes from them and observing how they would approach research problems.

I would pay special thanks to Professor Eric Slud and Professor Sean Downey, for their help in my transfer from the Chemical Physics Ph.D. program to the Applied Mathematics Ph.D. program. It was not an easy transition and would not have happened without their help.

I would also like to thank my fellow students, Qian Xu, Qian Wang,Yunchuan Li, Xiaoxu Meng, Mingze Gao, Chunxiao Liu, Qile Zhang, for their friendship and support. Ph.D. is a long and sometimes struggling process, and their support along the way has been critical.

Table of Contents

List of Figures

Chapter 1: Introduction

1.1 Ranking and Selection

Statistical ranking and selection (R&S) refers to the procedure of selecting a best (variously defined) system from a usually finite set of competing alternatives, where performance measures are expensive to collect and subject to noise. It is developed particularly for the simulation optimization setting where a computer program can be used to simulate a random output through techniques such as discrete event simulation (DES). A notable application is the design of semiconductor manufacturing chips. The process of a piece of silicon going through oxidization, etching and ion injection, each controlled by a parameter governing the underlying chemical and physical reactions, to be manufactured into a semiconductor device such as a transistor, can be simulated to estimate the throughput of a given design. One replication of the simulation of a modern production line could take hours or even days to complete [1]. Given a set of candidate designs represented by their respective control parameters, a decision maker has to efficiently allocate the available computing resources for selecting the optimal design.

Let $\mathscr{A}$ represent the set of candidate alternatives and $Y_i, i \in \mathscr{A}$ be the random output associated with alternative i. In its most basic form, the quality of alternatives are measured according to their expected output. Letting $\mu_i = \mathbb{E}[Y_i]$, the optimal alternative, which we denote as i^*, is defined as

$$i^* = \arg\max_{i \in \mathscr{A}} \{\mu_i\}.$$

A R&S procedure needs to draw samples on each alternative for committing to a final choice. Let $n_i, i \in \mathscr{A}$ be the number of samples collected on alternative i, and use $\bar{y}_i$ to denote the sample averages obtained through simulation, the probability of correctly selecting the true optimal alternative (PCS) can be expressed as

$$PCS = P\{\arg\max_{i \in \mathscr{A}} \{\bar{y}_i\} = i^*\},$$

if the sample means $\{\bar{y}_i\}_{i=1}^{k}$ are used for making the final selection. In the fixed budget setting, the goal is often to maximize PCS under the budget constraint $\sum_{i \in \mathscr{A}} n_i = N$, where N is the total simulation budget available. In the fixed confidence setting, the goal is to provide a confidence guarantee of the form $P\{PCS \geq 1-\delta\} \geq 1-\varepsilon$, where δ and ε are parameters that specify the target confidence guarantee.

The main challenge in designing efficient R&S procedures is how to address the exploration-exploitation trade-off. Consider the toy example of selecting among three alternatives associated with random rewards with normal densities with means $1, 0.9, 0$ and respective standard deviations $0.1, 0.5, 0.5$, as visualized in Figure 1.1. More replications should be allocated to alternatives with higher uncertainty and closer to optimal performance to achieve a higher PCS. For a fixed computing budget, policies include the optimal computing budget allocation (OCBA) [2] and Bayesian Value of Information

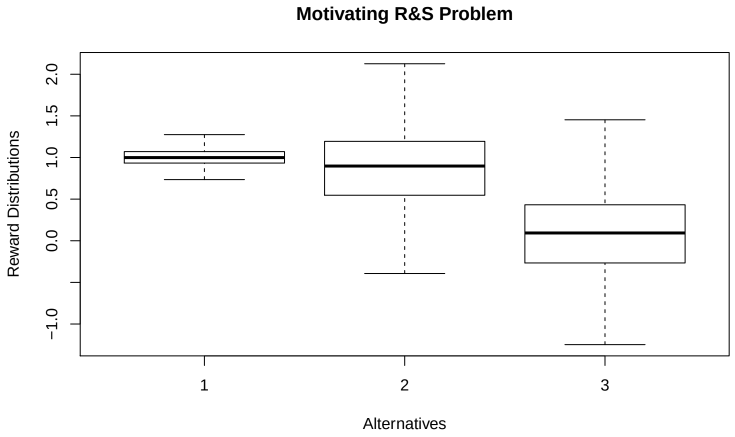

Figure 1.1: Alternative 3 is clearly inferior compared to the other two, and only needs a small number of samples. Alternative 2 is close to the true optimal alternative in terms of expected output, but has a much larger variance. Therefore the simulation budget should be focused on alternative 2 to obtain a high confidence results when comparing alternatives 1 and 2.

(VoI) are known to yield high PCS [3].

In this thesis, we will work on two variations of R&S problems: (1) when the alternatives are no longer independent, and (2) when the qualities of candidates are no longer being characterized by the expectations $\{\mu_i, i \in \mathscr{A}\}$.

1.2 Multi-armed Bandits

The multi-armed bandit (MAB) was first introduced by Robbins in 1952 in the context of medical trials [4] and its applications include areas such as dynamic pricing, personalized marketing and recommendation systems [5, 6]. MAB has a similar setup with R&S, but differs in its optimization objective. Let $\mathscr{A} = \{1, 2, \dots, k\}$ be the set of k arms, $X^i, i \in \mathscr{A}$, be the random reward associated with arm i with expectation μ_i and T be the time horizon. According to a policy Π, let a_t denote the arm selected at time step

t, MAB is interested in minimizing the expected regret, defined as

$$R_t = \mu^* T - \sum_{t=1}^{T} \mu_{a_t},$$

where $\mu^* = \max_{i \in \mathscr{A}} \{\mu_i\}$. R_t measures the loss the player suffered by not always playing the optimal arm (defined as the arm with maximum expected reward μ^*). To minimize R_t, the player has to carefully balance between exploration and exploration: observing more samples on each arm to learn about their respective reward distribution and choosing the currently observed best arm to obtain higher rewards.

In the most basic setting, $\{X_t^i\}_{i \in \mathscr{A}}$ are assumed to be stationary (See Definition 5 in Chapter 4). Further assuming that $X(a_i)$ is defined on the space $[0,1]$ for all $a_i \in \mathscr{A}$, a lower bound on the regret R_t is established as $R_t \geq O(\sqrt{T})$. Two popular polices, the Upper Confidence Index (UCB) [7] and Thompson Sampling (TS) [8] have been proved to achieve regret upper bound of $O(\sqrt{T})$. Let $\bar{x}_i$ be the sample average of reward from arm i and n_i be the number of samples, at time t, the UCB policy picks the arm a_t as

$$a_t = \arg\max_{a \in \mathscr{A}} \left\{ \bar{x}_a + \sqrt{\frac{\log t}{n_a}} \right\}.$$

The regret for the UCB policy is shown to have the upper bound

$$O(\sqrt{KT \log T}).$$

The UCB policy is the basis for developing policies for many variants of MAB such as Markovian bandits [9], linear bandits [10] and spectral bandits [11]. It is also the basis for our work in Chapter 4. Thompson sampling is a Bayesian sampling algorithm where a posterior belief is maintained on all arms to facilitate decision making. We refer the readers to for introduction on Thompson sampling [8].

1.3 Kriging Metamodel

Kriging was originally developed for analyzing geo-statistical data [1]. It could be viewed as a Bayesian technique for the estimation of an unknown function. It assumes a prior Gaussian distribution on the space of all smooth functions, so that the function values at a set of selected design points has a multivariate normal distribution. The posterior distribution of the function value at any new point, given observed function values, is also normal. Kriging model provides the inference capabilities as well as prediction, as the posterior variance could be viewed as an uncertainty estimate for the prediction performance. We delay the mathematical expressions for Kriging models to Chapter 4 and only illustrate through a toy example its prediction and inference capabilities.

Consider the problem of estimating the function

$$y(x) = \frac{\sin(20\pi x + 5)}{4x + (2x - 0.5)^4}$$

in the interval $[0,1]$, given the function values $y(0.30), y(0.71), y(0.51), y(0.97), y(0.17)$ with the design points selected with Latin Hyper-Cube design [1]. A kriging model returns an estimation of the underlying function as well as a posterior uncertainty estimation, as illustrated in Figure 1.2. We omit the implementation details such as the choice of correlation, as this example is simply for illustration purpose.

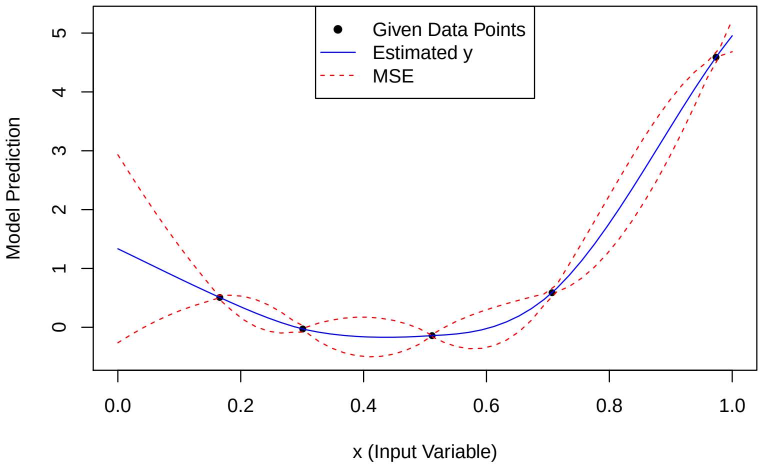

Figure 1.2: A Fitted Kriging Model on y. The estimated function interpolates the existing data points. The posterior uncertainty is given by the MSE.

In Chapter 4, we look at the problem of designing efficient experimental design algorithms for kriging: how to choose the points at which to evaluate the unknown function such that the kriging model will have optimal global fit.

Chapter 2: A Spectral Index for Selecting the Best System

2.1 Introduction

In the simulation optimization setting, stochastic ranking and selection (R&S) refers to the problem of selecting the best alternative through simulating samples on the performance measures on a finite set of candidate alternatives [12]. Given a fixed simulation budget, the goal is to design a selection algorithm to achieve some objective, such as maximizing probability of correct selection (PCS) or minimizing opportunity cost (OC). It is also known as the best arm identification problem in the bandit community. Most of the current research focuses on designing a budget allocation policy to efficiently collect performance samples. A few notable allocation policies include static allocation rules, such as OCBA (optimal computing budget allocation) [13], dynamic allocation rules balancing exploration-exploitation trade-off using expected improvement and knowledge gradient [14, 15], and procedures that focus on designing stopping rules for saving resources [16]. In some scenarios, it makes sense to optimize for different objectives such as opportunity cost or probability of selecting a good subset [17]. In this Chapter, we focus on the most popular probability of correct selection (PCS). For an overview of R&S problems, we refer the readers to [14] and [18].

Much of the R&S literature treats the alternatives to be independent of each other, and the samples collected are only used to make inferences about properties of the corresponding alternative. Common random numbers used to induce correlation in the performance samples can lead to better pairwise comparison accuracy (see, e.g., [19]), but mean performances are still estimated using only samples collected on each alternative. However, in many applications it makes sense to use samples collected on one alternative to infer about other alternatives. As a motivating example, consider the problem of selecting processing designs for manufacturing semiconductor chips. The values of some governing variables for processing steps such as oxidation, etching and ion injection must be specified [1]. Two designs with very similar input values on those steps can naturally be expected to have similar performance, and if one of the two is believed to be much inferior compared to some other designs, so should the other.

There are mainly two lines of research trying to address this phenomenon. One is to build a parametric model incorporating properties into the objective functions and learn the model parameters. One popular choice is to assume the expected performance is a linear function of some input feature and efficiently allocate the simulation budget to learn the feature parameters [20]. This approach was applied to a drug discovery problem and showed promising results [21]. Similar approaches were also proposed in the closely related multi-armed bandit setting and achieved good empirical performance even when the linear assumption is not met [10]. In this approach, a new performance sample on one alternative could provide information about all alternatives by updating the estimation of parameter values. Another line of research follows the Bayesian approach and treats the similarities between alternatives as correlations in prior beliefs. [15] model the similarities as assumed known correlations in a correlated normal prior, allocating the simulation budget using the knowledge gradient policy. [22] further extended the approach by assuming the correlations are unknown and can be learned using a conjugate Bayesian learning model where the correlations have a Wishart distribution. A related work is stochastic kriging, which constructs a metamodel to predict the expected performance on all alternatives using similarity as correlation [23].

Despite the richness of literature on R&S, an important part of the problem remains relatively neglected: the final selection rule. In [24], the authors proposed to select the alternative with maximum integrated posterior PCS rather than the one with the maximum sample average. Selection based on quantile estimates has also been proposed largely to avoid sensitivity to outliers and to address a different design objective than PCS [25]. In our work, we assume the sample data are obtained through some sampling allocation policy and focus on how to make a final selection based on the collected data.

Our approach is largely motivated by research outside the operations research and statistics community. The term *Spectral Methods* refers to a large family of methods constructing a similarity graph on the entity of interest to bring the vague similarity information into rigorous mathematical formulation to improve decision making. In image processing, a normalized graph cut approach for segmenting objects can be proven to be equivalent to a spectral clustering on the pixel values [26]. In semi-supervised learning, a similarity graph is constructed to transfer information on nodes of the graph to make classifications [27]. Spectral clustering is a novel clustering approach capable of discovering structures within data that cannot be detected by traditional K-means or K-mods clustering methods [28].

We propose a spectral selection index that is a transform of the collected data using a similarity graph constructed from known information about all alternatives. This approach addresses the problem of making inferences about all alternatives from new observations from the perspective of *selection* rather than that of *allocation*. It provides an alternative to the parametric and Bayesian approach. More interestingly, our approach does not depend on the data collection process, and therefore can be implemented with any existing simulation budget allocation policy. Our approach also has a provable improvement guarantee that is missing in many existing R&S procedures. Numerical experiments show that our approach will give good improvement with very mild assumptions on the similarity graph.

This Chapter is organized as follows. Section 2.2 motivates our approach with a toy R&S problem. Section 2.3 introduces relevant graph theoretic results and formulates our approach. Section 2.4 establishes some performance guarantees of the proposed selection index. We illustrate the performance of our approach on two synthetic problems in Section 2.5. Finally, we conclude in Section 2.6.

2.2 A Toy Motivating Problem

Consider a toy R&S problem with a fixed allocation policy: the three candidate alternatives have normal random rewards with means $1, 0, 0$ and the same standard deviation of 10. Each alternative is allocated one simulation replication and the selection rule is to choose the alternative with the maximum observed value. The probability of correctly selecting the first alternative can be computed to be 0.362, only slightly better than a uniform random selection. However, if some prior information indicates that alternative

2 and 3 should have similar performance, we construct a similarity matrix S as

$$S = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

for representing the belief that alternative 1 is quite *different* from alternatives 2 and 3 while 2 and 3 are believed to have similar performance. Let y_1, y_2, y_3 be the obtained samples in this allocation policy. We propose a new index z which minimizes the following expression

$$(z_1 - y_1)^2 + (z_2 - y_2)^2 + (z_3 - y_3)^2 + s_{23} \times (z_2 - z_3)^2,$$

where the first three terms force $z_i, i \leq 3$ to be close to the observed information $y_i, i \leq 3$, and the remaining terms force z_2 and z_3 to be closer, as they have a non-zero similarity measure. Minimizing the above objective we obtain $z_1 = y_1, z_2 = \frac{2}{3}y_2 + \frac{1}{3}y_3, z_3 = \frac{2}{3}y_3 + \frac{1}{3}y_2$, which is a new set of indices that is a weighted average of its similar neighbors. Selecting based on the new index z, the probability of correct selection is found to be 0.472. We call such a smoothed index the *Spectral Index*, as a similarity matrix (or graph) is used for computing the new index to incorporate the known information to facilitate better selection. In later sections, we rigorously formulate the spectral approach for transforming the observed data using available similarity information, and prove the performance improvement using the proposed index.

2.3 Similarity Graphs in R&S

Let $\mathscr{A} = \{1, 2, ..., k\}$ be the set of alternatives each associated with an known random reward with the expected values $(\mu_1, ..., \mu_k)$ and standard deviations $(\sigma_1, ..., \sigma_k)$. Without loss of generality, we assume that $\mu_1 \geq \mu_2 \geq ... \geq \mu_k$. The optimal alternative is defined to be the one with maximum expected performance and denoted by i^*. Therefore in our setup, $i^* \equiv 1$. In this section, we discuss how to use a *spectral approach* to rigorously incorporate pairwise similarity information into an R&S procedure.

2.3.1 Graph Notation

Let $S \in R_+^{k \times k}$ be the *similarity matrix* where the element $s_{ij} \geq 0$ denotes the similarity measurement between alternatives i and j, $\forall i,\ j \in \mathscr{A}$. The similarity information can be represented by a similarity graph $G = (\mathscr{A}, S)$, where vertex i represents alternative i and the edges between vertices i and j are weighted by the similarity between the two connecting nodes s_{ij}. We will later show that the choice of the diagonal elements of S is arbitrary and does not affect the final outcome, therefore for simplicity, we assume that $s_{ii} = 0, \forall i \leq k$, meaning all diagonal elements of S (or self similarity) are 0. For the weighted graph G, the degree of vertex $i \in \mathscr{A}$ is defined by

$$d_i = \sum_{j=1}^{k} s_{ij},$$

which is a measurement of the total connectivity of this vertex and represents how similar it is to all other alternatives.

Definition 1 (Degree Matrix)**.** *The degree matrix D of the similarity matrix S is a diagonal matrix with degrees $d_1, ..., d_k$ as its diagonal elements.*

Definition 2 (Graph Laplacian)**.** *The unnormalized graph Laplacian matrix is defined as*

$$L = D - S. \tag{2.1}$$

The graph Laplacian is a key concept in spectral methods. Here, we presented the unnormalized graph Laplacian. Other alternatives involve normalizing L using degree matrix D in various ways, such as $L_{sym} = D^{-1/2}LD^{-1/2}$ or $L_{rw} = D^{-1}L$. The normalized Laplacians have been shown to lead to better performance in various tasks, such as spectral clustering [28] and semi-supervised learning [29] . We use the unnormalized Laplacian, since it will provide an intuitive explanation to how similarity would affect the selection procedure. Notice that the diagonal elements of S will cancel out in the computation of L. In our approach this means that self similarity will have no effect on the final outcome.

Lemma 1 (Proposition 1 of [28])**.** *The graph Laplacian L has two important properties:*

1. *L is symmetric and positive semi-definite.*

2. *The smallest eigenvalue of L is* 0.

We refer readers to [28] for the detailed proof.

2.3.2 Computing Similarity Graphs

Similar to *spectral methods*, another technique that uses a similarity matrix to represent the affinity on a finite set of entities is kriging or Gaussian process regression. Both are Bayesian approaches assuming a correlated normal prior with correlations determined by a similarity or kernel matrix. The posteriors are updated using observed samples. In this part, we introduce some basic methods for constructing similarity graphs.

The first type of matrix is based on expert knowledge. A similarity score s_{ij} can be manually assigned for all pairs if k is moderate in size. Such examples appear in linguistics [30] and gene enrichment analysis [31].

More generally, the similarity could be computed from some features about the alternatives. Let $x(i)$ be an m-dimensional feature vector representing the properties of alternative i. In our motivating example, $x(i)$ could be a vector representing oxidization rates, etching time length and ion injection density for a manufacturing design. Some popular choices are the ε, Gaussian and exponential similarity graphs, which can be computed respectively by:

1. ε-Graph:

$$s_{ij} = \delta \mathbb{1}_{\{\|x(i)-x(j)\|_2 \leq \varepsilon\}}, \tag{2.2}$$

   where $\mathbb{1}_{\{\cdot\}}$ is the indicator function.

2. Gaussian Graph:

$$s_{ij} = e^{-\sum_{n=1}^{m} \theta_n (x_n(i)-x_n(j))^2}, \tag{2.3}$$

3. Exponential Graph:

$$s_{ij} = e^{-\sum_{n=1}^{m} \beta_n \|x_n(i) - x_n(j)\|_2}. \tag{2.4}$$

The $\delta, \varepsilon, \theta, \beta$ are all input hyper-parameters determining the magnitude of the similarity. The ε-Graph will return a sparse similarity matrix S, meaning one alternative is only connected with its few closest neighbors on the graph. The sparsity will also lead to faster numerical computations. Though a feature vector is used to describe each alternative and compute their similarities, there is no assumption on the parametric form of expected performance relating to the features. In Section 2.5, the proposed graphs will be evaluated through simulation.

2.4 The Spectral Selection Index

We assume the allocation is given by a policy Π. Let $y_{ij}, \forall i \in \mathscr{A}, j \leq n_i$ denote the samples collected in the simulation process, where n_i is the number of simulations allocated to alternative i after exhausting the total budget N. Denote $\bar{\mathbf{y}}$ as the vector of sample averages $(\bar{y}_1, ..., \bar{y}_k)$. Instead of choosing the alternative with maximum sample average, we propose a spectral index $\mathbf{z} = (z_1, \ldots, z_k)$ as our selection criteria. Use $i_N^{\bar{\mathbf{y}}}$ and $i_N^{\mathbf{z}}$ to denote the final selected alternative given all observed data using the sample average and our proposed selection index, respectively.

2.4.1 Smooth Index on Similarity Graph

The usage of sample averages as selection criteria is intuitive, since they generally provide an unbiased estimator of the true expected performance. However, as our goal is to maximize PCS, comparing *relative* performance is more critical than finding accurate estimates. This is also the basis of ordinal optimization [32, 33].

Given a similarity graph G and motivated by the fact that two similar alternatives should have similar selection index, , similar to the approach in semi-supervised learning [27], we propose a spectral index $\mathbf{z}$ that is the solution to the following optimization problem:

$$\min_{\mathbf{x}\in R^k} \sum_{i=1}^{k} |x_i - \bar{y}_i|^p + \frac{\lambda}{2} \sum_{1\le i,j\le k} s_{ij}|x_i - x_j|^q. \tag{2.5}$$

The first term forces x_i to be close to the sample averages $\bar{y}_i$, and the second term forces two alternatives with larger similarity to have closer index values. λ is a positive regularization coefficient that controls the weight between the two terms. p, q are positive integers specifying the norms to use when enforcing the smoothness. [34] provides algorithms for solving such an optimization problem with various choices of p, q. In this work, we set $p = q = 2$, both for computational convenience and for developing intuitive explanations. We have the following theorem for compact representation of the optimization problem.

Theorem 1. *When $p = q = 2$ in (2.5), the optimization problem can be expressed with the graph Laplacian matrix L as*

$$\mathbf{z} = \arg\min_{x\in\mathbb{R}^k} (\mathbf{x} - \bar{\mathbf{y}})' (\mathbf{x} - \bar{\mathbf{y}}) + \lambda \mathbf{x}' L \mathbf{x}, \tag{2.6}$$

where x' denotes the transpose of x.

Proof. We can write the second term in (2.6) using summations as

$$\begin{aligned}
\mathbf{x}'L\mathbf{x} &= \sum_{i,j\leq k} x_i L_{ij} x_j \\
&= \frac{1}{2}\left\{\sum_{i\leq k} L_{ii}x_i^2 + \sum_{j\leq k} L_{jj}x_j^2 + \sum_{i\neq j, i,j\leq k} 2L_{ij}x_i x_j\right\} \\
&= \frac{1}{2}\left\{\sum_{i\leq k} d_i x_i^2 + \sum_{j\leq k} d_j x_j^2 - \sum_{i\neq j} 2s_{ij}x_i x_j\right\} \\
&= \sum_{i,j\leq k} s_{ij}\left(x_i - x_j\right)^2.
\end{aligned}$$

The $(\mathbf{x}-\bar{\mathbf{y}})'(\mathbf{x}-\bar{\mathbf{y}})$ terms match trivially with the first term in (2.5). □

Taking the derivative of the optimization objective and setting it to zero yields

$$\mathbf{z} = (I+\lambda L)^{-1}\bar{\mathbf{y}}, \tag{2.7}$$

where I is the identity matrix of rank k.

Proposition 1. *$I+\lambda L$ is symmetric and positive definite with minimum eigenvalue* 1.

Proof. This is a direct consequence of Lemma 1 . □

Theorem 1 implies that our approach has stable numerical properties, as the condition number of our linear system computation will not be too large.

The computation of $\mathbf{z}$ involves taking the inverse of the matrix $I+\lambda L$, which could be expensive for problems with a large number of alternatives. The fact that the minimum eigenvalue of the inverted matrix is 1 usually means the condition number of the inverse computation will not be too large, giving stable computing results. The solution in Equation (2.7) can also be obtained using an iterative approach. Denote the t-th iterate by $\mathbf{z}^{(t)}$ and its i-th element by $z_i^{(t)}$. We propose Algorithm 1 for computing $\mathbf{z}$ iteratively.

Algorithm 1: Iterative Gradient Descent.

```
Input: Stepsize $a$, convergence criterion $\delta$.
Output: spectral index $\mathbf{z}$.
Set $t = 0$ and $z^{(0)} = \bar{b}y$.
while $|z^{(t)} - z^{(t-1)}| > \delta$ do
    $\mathbf{z}^{(t)} = \mathbf{z}^{(t-1)} + a\left(\bar{\mathbf{y}} - (I + \lambda L)\mathbf{z}^{(t-1)}\right)$
    set $t \leftarrow t + 1$
return $\mathbf{z}^{(t)}$
```

Theorem 2 (Convergence of Algorithm 1)**.** *Let δ be the largest eigenvalue of L, if $0 < a < \frac{1}{|1+\lambda\delta|}$, we have $\mathbf{z}^{(t)} \rightarrow \mathbf{z}$ as $t \rightarrow \infty$.*

Proof. By Theorem 1, $\mathbf{z}$ defined in Equation (2.7) is unique. Let $\mathbf{r}^{(t)}$ be the residual at the t-th iteration in Algorithm 1, i.e.,

$$\mathbf{r}^{(t)} = (I + \lambda L)\mathbf{z}^{(t)} - \bar{\mathbf{y}}. \tag{2.8}$$

It suffices to prove that $\|\mathbf{r}^{(t)}\| \rightarrow \mathbf{0}$. We have

$$\begin{aligned}
\mathbf{r}^{(t+1)} &= (I + \lambda L)\mathbf{z}^{(t+1)} - \bar{\mathbf{y}} \\
&= (I + \lambda L)\mathbf{z}^{(t)} - \bar{\mathbf{y}} - a(I + \lambda L)((I + \lambda L)\mathbf{z}^{(t)} - \bar{\mathbf{y}}) \\
&= (I - a(I + \lambda L))\,\mathbf{r}^{(t)} \\
&= (I - a(I + \lambda L))^{t+1}\,\mathbf{r}^{(0)}.
\end{aligned}$$

By Theorem 1 and the assumption that $0 < a < \frac{1}{|1+\lambda\delta|}$, the eigenvalues of the matrix

$I - a(I + \lambda L)$ are less than 1. Therefore,

$$\|\mathbf{r}^{(t+1)}\| \leq \alpha^{t+1} \|\mathbf{r}\|^{(0)},$$

where α is the largest eigenvalue of $I - a(I + \lambda L)$, and we have $0 < \alpha < 1$. Thus, $\|\mathbf{r}^{(t)}\| \to 0$, i.e., $\mathbf{r}^{(t)} \to \mathbf{0}$. □

Remark: Algorithm 1 is in fact a gradient descent approach for minimizing the objective function in the optimization problem (2.6). This approach for solving z will not only provide an alternative to the matrix inversion approach, but also provide a mechanism for proving performance guarantees of our method. Notice that in each time step t, the update for an alternative $i \in \mathscr{A}$ is

$$z_i^{(t+1)} = a\bar{y}_i + (1-a) z_i^{(t)} + a\lambda \sum_{j \neq i} s_{ij} \left(z_j^{(t)} - z_i^{(t)} \right). \tag{2.9}$$

The update is discounting the observed information $\bar{y}_i$ and mixing in information from alternative j according to the similarity s_{ij}, the regularization coefficient λ and update step size a. Algorithm 1 and Theorem 2 lead to the following result.

Corollary 3 (Weighted Averages)**.** *The index* $\mathbf{z}$ *satisfies the following weighted averaging property:*

$$z_i = \frac{\bar{y}_i + \lambda \sum_{j \neq i} s_{ij} z_j}{1 + \lambda d_i}. \tag{2.10}$$

Proof. In Equation (2.9), as $t \to \infty$, we know that $z_i^{(t+1)} = z_i^{(t)} = z_i, \forall i \leq k$ from Theorem

2. Rearranging the terms yields

$$a(1+\lambda d_i)z_i = a\bar{y}_i + a\lambda \sum_{j\neq i} s_{ij}z_j,$$

which is equivalent to Equation (2.10). □

Corollary 3 will be the key to our performance proofs. The expression also provides two key intuitions on the z index: (1) a vertex with a larger degree d_i will be less affected by its actual observations $\bar{y}_i$, (2) the weighted averaging is the source of potential *PCS* gain: a sub-optimal alternative with unusually high observed performance could be "dragged down" by its neighbors on the graph. In the next part, we define a class of graphs that will lead to PCS improvement.

2.4.2 Performance Improvement

With z computed either from Equation (2.7) or Algorithm 1, we compare the following two rules for making the final selection:

1. Using sample averages $\bar{\mathbf{y}}$: $i_N^{\bar{\mathbf{y}}} = \arg\max_{i\in\mathscr{A}}\{\bar{y}_1,\ldots,\bar{y}_k\}$
2. Using spectral index $\mathbf{z}$: $i_N^{\mathbf{z}} = \arg\max_{i\in\mathscr{A}}\{z_1,\ldots,z_k\}$

Let $PCS(\bar{\mathbf{y}})$ and $PCS(\mathbf{z})$ denote the PCS for each respective selection rule.

Definition 3 (Aligned Graph)**.** *A graph G is aligned if the similarities $\{s_{ij}\}_{i,j\in\mathscr{A}}$ are monotonically decreasing with $|\mu_i - \mu_j|, \forall i \neq j; i,j \in \mathscr{A}$ and $d_i = d_1, \forall i$.*

Though the similarities are functions of alternative feature vectors $x(i), i \in \mathscr{A}$, rather than expected performances $\mu_i, i \in \mathscr{A}$, we can still compare the similarity score between alternatives and their true underlying gap of expected performance. An aligned graph would use the prior known feature information $x(i)$ to correctly capture the relative closeness of alternatives: if i and j have smaller gap in expected performance, their similarity would be greater. It also requires the degree for all vertices to be the same, which could be achieved by normalizing an existing graph Laplacian L [28]. It may be a strong assumption to expect that such a graph can be constructed without knowing the true performance, but such a family of graphs will provide nice theoretical results and shed light on the intuition of our approach. The family of graphs that are aligned will have provable nice performance improvement.

Theorem 4 (Order Preserving Updates)**.** *For an aligned graph G and performance sample averages with correct ordering, i.e.,* $\bar{y}_1 \geq \bar{y}_2 \geq, \ldots, \geq \bar{y}_k \geq 0$*, the spectral index defined in Equation (2.7) preserves correct ordering, i.e.,* $z_1 \geq z_2 \geq, ..., \geq z_k \geq 0$.

Proof. We establish the proof by induction.

At iteration 0, $\mathbf{z}^{(0)} = \bar{\mathbf{y}}$. By assumption, we have $z_1^{(0)} \geq z_2^{(0)} \geq, ..., \geq z_k^{(0)} \geq 0$.

At iteration $t > 0$, assume $z_1^{(t-1)} \geq z_2^{(t-1)} \geq, \ldots, \geq z_k^{(t-1)} \geq 0$ holds. To prove Theorem 4, using Algorithm 1, we only need to prove that $z_i^{(t)} \geq z_{i+1}^{(t)}$ holds $\forall 1 \leq i \leq k-1$. Rearranging Equation (2.9), we write out $z_i^{(t)}$ and $z_{i+1}^{(t)}$ as

$$z_i^{(t)} = a\bar{y}_i + (1 - a - a\lambda d_i) z_i^{(t-1)} + a\lambda \sum_{m \neq i, i+1} \left(s_{i,m} z_m^{(t-1)} \right) + a\lambda s_{i,i+1} z_{i+1}^{(t-1)},$$

$$z_{i+1}^{(t)} = a\bar{y}_{i+1} + (1 - a - a\lambda d_{i+1}) z_{i+1}^{(t-1)} + a\lambda \sum_{m \neq i, i+1} \left(s_{i+1,m} z_m^{(t-1)} \right) + a\lambda s_{i,i+1} z_i^{(t-1)}.$$

Under the condition that $d_i = d_{i+1} = d \; \forall i$, taking the difference yields

$$z_i^{(t)} - z_{i+1}^{(t)} = a(\bar{y}_i - \bar{y}_{i+1}) + (1 - a - a\lambda d - a\lambda s_{i,i+1})(z_i^{(t-1)} - z_{i+1}^{(t-1)}) \tag{2.11}$$

$$+ a\lambda \left(\sum_{m \neq i,i+1} s_{i,m} z_m^{(t-1)} - \sum_{m \neq i,i+1} s_{i+1,m} z_m^{(t-1)} \right). \tag{2.12}$$

To show that $z_t^{(t)} - z_{i+1}^{(t)} \geq 0$, under the condition that $\bar{y}_i \geq \bar{y}_{i+1}$, $z_i^{(t-1)} \geq z_{i+1}^{(t-1)}$ and a is sufficiently small, we only need to show that

$$\sum_{m \neq i,i+1} s_{i,m} z_m^{(t-1)} - \sum_{m \neq i,i+1} s_{i+1,m} z_m^{(t-1)} \geq 0. \tag{2.13}$$

Intuitively it makes sense, as the former is a weighted average with higher weights placed on larger terms of $z_m^{(t-1)}$ for an aligned graph. With $d_i = d_{i+1}$ and $s_{i,i+1} = s_{i+1,i}$, we know that $\sum_{m \neq i,i+1} s_{i,m} = \sum_{m \neq i,i+1} s_{i+1,m}$, which can be written as

$$\sum_{m<i} (s_{i,m} - s_{i+1,m}) = \sum_{m>i+1} (s_{i+1,m} - s_{i,m}) \geq 0.$$

With $z_{i-1}^{(t-1)} > z_{i+1}^{(t-1)}$, we have

$$\left\{ \sum_{m<i} (s_{i,m} - s_{i+1,m}) \right\} z_{i-1}^{(t-1)} \geq \left\{ \sum_{m>i+1} (s_{i+1,m} - s_{i,m}) \right\} z_{i+1}^{(t-1)}. \tag{2.14}$$

Comparing the terms in Equations (2.13) and (2.14), for an aligned graph, we have

$$s_{i,m} > s_{i+1,m}, \; \forall m < i,$$

$$s_{i,m} < s_{i+1,m}, \; \forall m > i.$$

Then, using the assumption that $z_1^{(t-1)} \geq z_2^{(t-1)} \geq \cdots \geq z_k^{(t-1)} \geq 0$, we have

$$(s_{i,m} - s_{i+1,m})\, z_m^{(t-1)} > (s_{i,m} - s_{i+1,m})\, z_{i-1}^{(t-1)},\ \forall m \leq i-1$$
$$(s_{i+1,m} - s_{i,m})\, z_m^{(t-1)} < \{s_{i+1,m} - s_{i,m})\, z_{i+1}^{(t-1)},\ \forall m \geq i+1.$$

From the above conclusion, we naturally have the inequalities

$$\sum_{m<i} (s_{i,m} - s_{i+1,m})\, z_m^{(t-1)} \geq \left\{ \sum_{m<i} (s_{i,m} - s_{i+1,m}) \right\} z_{i-1}^{(t-1)} > \left\{ \sum_{m>i+1} (s_{i+1,m} - s_{i,m}) \right\} z_{i+2}^{(t-1)} \geq \sum_{m>i+1} (s_{i+1,m} - s_{i,m})\, z_m^{(t-1)}.$$

This proves Equation (2.13), concluding the proof. □

In the proof of Theorem 4, we assumed the step size a in Algorithm 1 is sufficiently small, meaning during such implementations, the correct ordering will be preserved during all update steps. Theorem 4 establishes the correctness of spectral selection in cases with $n_i \to \infty$ and $\bar{y}_i \to \mu_i$, where selecting using z will return the true optimal alternative. In cases where sample averages are incorrectly ordered, we expect the extra information from the aligned graph will correct the ordering in the final spectral indices. Therefore, our approach should work at least as well as sample averages, and we have the following conjectures.

Conjecture 1 (Selection Fixing)**.** *With an aligned graph, $P\{i_N^z = 1 | i_N^{\bar{y}} \neq 1\} > 0$ if all reward distributions have unbounded support.*

If the sample average for a sub-optimal alternative is the largest among all averages due to random sampling, its neighbors on the aligned similarity graph G will be able to negate the random error, thus making the spectral index more robust. This is the statistical intuition behind our approach: the spectral index z is a smoothed version of sample averages $\bar{y}$ using a similarity graph with better robustness.

Conjecture 2 (PCS Improvement)**.** *With an aligned graph,* $PCS(z) \geq PCS(\bar{y})$.

In circumstances where an aligned graph is not available, such as due to lack of information, we still expect the spectral index to improve PCS, i.e., an aligned graph is not necessary to achieve a better selection performance.

Conjecture 3 (Unaligned Graph)**.** *For any given R&S problem, for any budget allocation policy* Π*, there exist unaligned graphs* G *such that* $PCS(z) \geq PCS(\bar{y})$.

We validate Conjectures 1, 2 and 3 in our numerical experiments.

2.4.3 Information Collection on Graph

The constructed similarity graph motivates us to compute a selection index that utilizes the relative similarity information and given performance samples. However, it is expected that a graph (possibly unaligned) should also provide insight on how to allocate the simulation resources. A vertex $i \in \mathscr{A}$ with a higher degree d_i is *better connected* on the graph; therefore, one sample on this alternative would provide more information compared to a sample on a vertex with a small degree on the graph. This issue is partially addressed as active learning on graphs [27] or experimental design for kriging approaches [1]. It is possible to develop variants of the Knowledge Gradient or Expected

Improvement dynamic allocation policies exploiting a similarity graph for guiding the sampling. We defer this to future work.

2.5 Numerical Experiments

We test our approach on two synthetic R&S problems implemented with both the equal allocation and OCBA allocation policy. The final selection is made using both the sample averages and our proposed index, and the PCS is estimated for each policy. When implementing the OCBA policy, the allocation is computed with the known parameters μ_i and σ_i, as our goal is to illustrate the benefit of employing the spectral index for selection given an allocation policy.

In the first numerical experiment, we simulate a problem with the so-called *least favorable configuration* and test our spectral index using a manually constructed aligned graph. The effect of different values of the parameter λ is also illustrated. The second simulation is performed using a commonly used test problem, and we show that non-aligned graphs could give a spectral index that outperforms selection based on sample averages.

2.5.1 Least Favorable Configuration

We refer to the setting where all suboptimal alternatives have identical expected performance as the *least favorable configuration*, in the sense that none of the suboptimal ones can be easily identified [12]. We simulate 5 alternatives each with i.i.d. normal reward samples with mean $\mu_i = \mathbb{1}_{\{i=1\}}$ and $\sigma_i = 4, \forall i \leq 5$. For a total simulation budget

N ranging from 50 to 1500, we use both the equal allocation and the OCBA policy to allocate N and compute the PCS over 10000 simulation replications. The similarity graph is constructed as

$$s_{ij} = \begin{cases} 1, & i \neq j, 2 \leq i, j \leq 5 \\ 0, & \text{otherwise} \end{cases}. \tag{2.15}$$

This similarity graph reflects the belief that the 4 sub-optimal alternatives are closer to each other, whereas the optimal one is somewhat isolated. We test our approach using λ values of $0.1, 0.5, 1, 2$. Notice that the graph is an aligned graph according to Definition 3; thus z is guaranteed to give a better PCS.

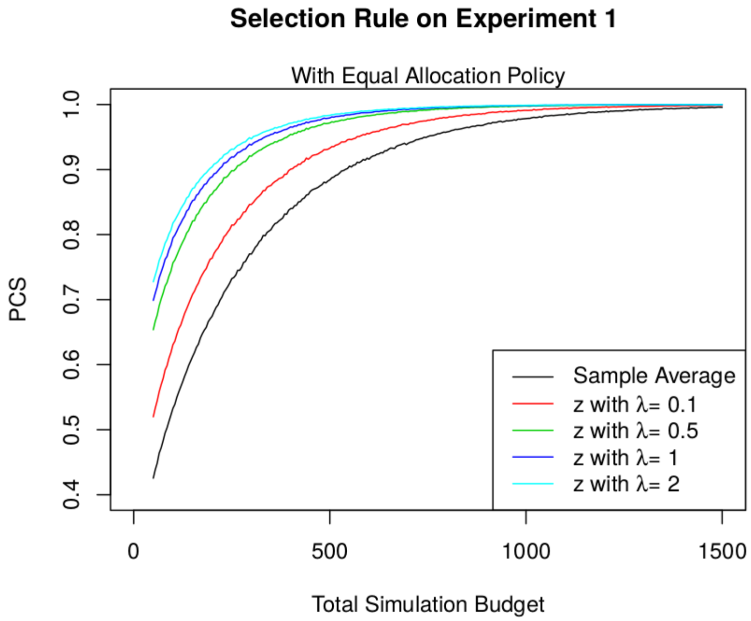

Figure 2.1: PCS Improvement under Equal Allocation with Different Similarity Graphs and Regularization Coefficients.

In Figures 2.1 and 2.2, we can see that the spectral index indeed dominates selection using sample averages for both policies over all simulation budgets tested. With an aligned graph, larger values of λ means stronger belief in the relative relationships in

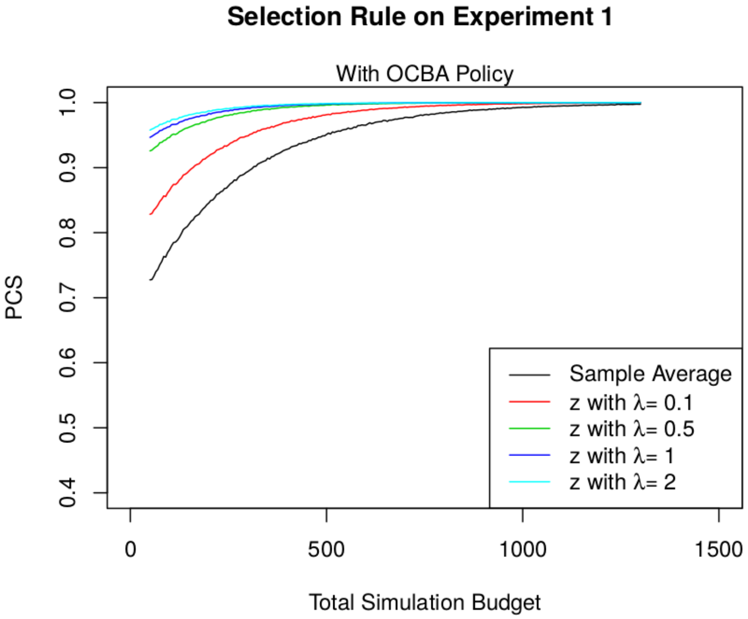

Figure 2.2: PCS Improvement under OCBA Allocation Policy with Different Similarity Graphs and Regularization Coefficients.

alternatives and indeed leads to better performance.

2.5.2 Non-Aligned Graph

For more complicated R&S problems, it might be difficult to construct an aligned graph, as in the first simulation experiment. However, it is often reasonable to believe the expected performances are smooth with respect to (w.r.t.) to some features.

Consider a problem with 10 alternatives with normal rewards, with means $\mu_i = \frac{1}{4}(i-5.75)^2, \forall i \in \{1,2,...,10\}$ and a common variance 10. We test the following similarity graphs:

1. ε-*Graph*: $s_{ij} = 0.4 \times \mathbb{1}_{|i-j|\leq 3}$. Only consider the effect of very close neighbors.

2. *Exponential Graph*: $s_{ij} = e^{-|i-j|}, \forall i,j \leq 10$.

3. *Gaussian Graph*: $s_{ij} = e^{-(i-j)^2}, \forall i, j \leq 10$.

The configuration of the test problem is shown in Figure 2.3 . On the left, we can see that the expected performance value μ is a smooth function of the alternative index, mimicking a smooth objective function w.r.t. some input features. On the right, we present the similarities between the optimal alternative with all other alternatives computed using the three graphs and present it with the gap in expected performance. The similarity is not monotonically decreasing; therefore, none of the three graphs is aligned.

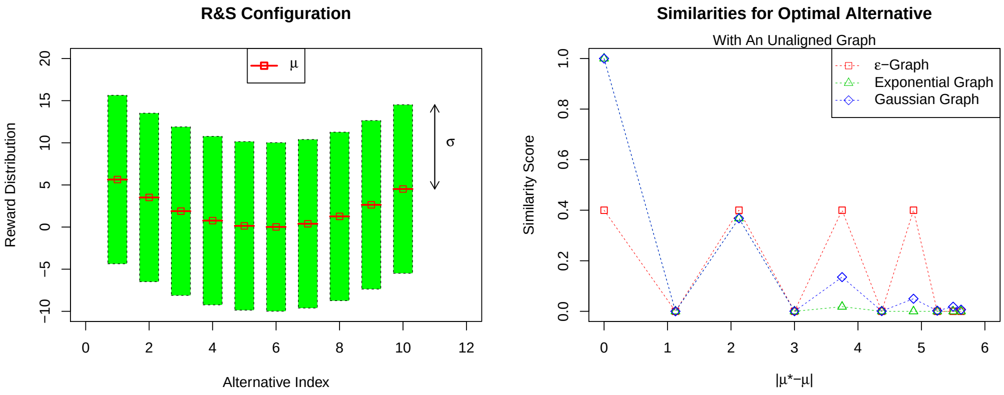

Figure 2.3: R&S Test with Non-aligned Graphs

The *PCS* is estimated based on 1000 simulation replications for budgets ranging from 50 to 3000 with both equal allocation and OCBA allocation policy. We test our approach using $\lambda = 0.2$ and 0.3. Though none of the three tested graphs are aligned, they still improve the selection performance for both allocation policies, as shown in Figures 2.4 and 2.5.

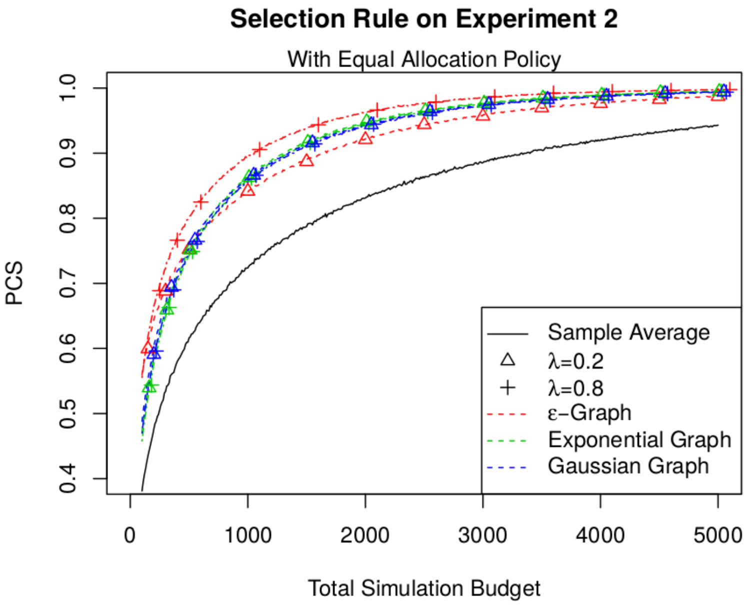

Figure 2.4: PCS with Non-Aligned Graphs with Equal Allocation

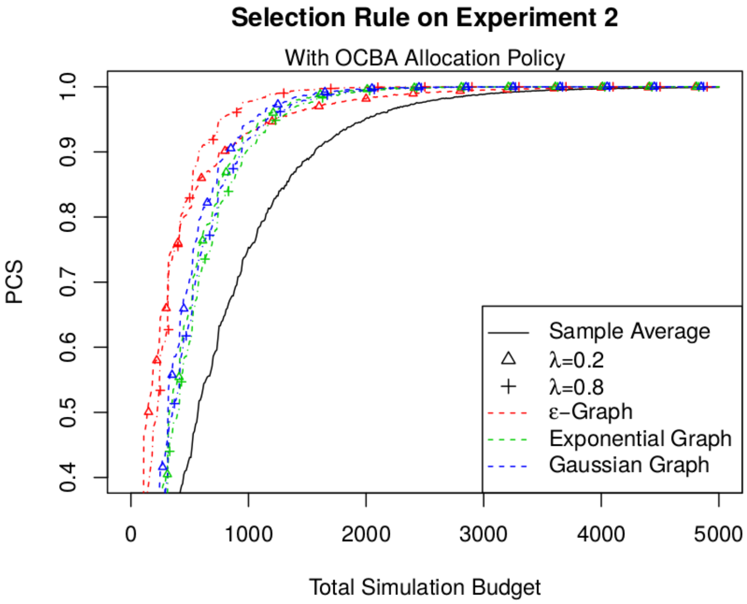

Figure 2.5: PCS with Non-Aligned Graphs with OCBA Allocation

2.6 Conclusion

We proposed a performance index that incorporates the similarity information between pairs of alternatives in R&S and proved that the proposed index will give a better selection performance if reasonable similarity information is available. Numerical experiments showed promising results on both theoretically good problem instances and more general problem instances. The main contribution is to provide an easy way for making inferences about all alternatives with samples from one alternative. The proposed approach is very intuitive and computationally trivial compared to many proposed Bayesian approaches. The proposed index does not affect the allocation policy procedures and therefore could be easily combined with any allocation policy, as illustrated with our numerical experiments.

Chapter 3: Utility-based Statistical Selection Procedures

3.1 Introduction

Ranking and Selection (R&S) refers to the procedure of selecting a best system from the usually finite set of competing alternatives, where performance measures are expensive to collect and subject to noise. The vast majority of R&S literature works with the expectation of the random output. For example, consider the problem of selecting an *optimal* route for a delivery service [35]. Modeling the traveling times on a designed route as a random variable to capture the unpredictable effects from factors such as weather and traffic conditions, the problem of testing a set of candidate routes to quickly identify the one with the smallest mean delivery time could then be treated as a R&S problem. Efficient testing strategies include optimal computing budget allocation (OCBA) and Bayesian value of information (VoI) based allocation algorithms. However, the mean delivery time may not be the appropriate measurement of route quality in this scenario. A route with a slightly higher mean but much less variance in delivery time, could be preferable compared to one with a smaller mean but much larger variance, as an unusually long delivery could cause packages to be delayed to the next day. Similar problems are also found in many other applications. In financial applications, value at risk (VaR) and conditional value at risk (CVaR) are two popular objectives when comparing different pricing strategies, as financial institutions are extremely sensitive to risks [36]. In behavioral economics, a cumulative prospect theoretic (CPT) utility is often used to properly capture people's perception of random rewards in games such as lotteries and gambling [37]. In such scenarios, a utility function could be used to capture the problem-specific preferences of decision makers. In a similar route selection scenario, a (CPT) utility was applied in the work of [38] in the multi-armed bandit setting for avoiding extraordinarily long traveling times. Prior research on ranking and selection algorithms designed for objectives other than expected values include [39] for minimizing variance and [40] for quantile. In this Chapter, we consider more general objective functions.

There is a rich literature on solving R&S problems with simple expectation being the utility. The OCBA framework maximizes probability of correct selection (PCS) under a budget constraint to find the asymptotically optimal policy, proposing sequential allocation algorithms using plug-in estimates for the unknown parameters in the optimal allocation [2]. The indifference-zone (IZ) approach provides a frequentist confidence guarantee for PCS under the assumption that there exists a gap of δ in expected performance between sub-optimal alternatives and the optimal one [41]. Another line of research, which is often referred to as the Bayesian VoI approach, works under a Bayesian framework for efficiently learning the expected value of the unknown random reward. At each step, the alternative that contains the most *information* (variously defined) is selected. Two notable examples are the expected improvement (EI) and knowledge gradient (KG) policies, which are shown to be more efficient in the finite-budget domain compared to asymptotically optimal policies such as OCBA [20,42,43]. Recent results in [3] and [44] connected the EI policy asymptotically to the OCBA policy, providing theoretical support to its empirical performance. Another advantage of the Bayesian framework is its flexibility in incorporating problem-specific information such as correlations into the allocation procedures, such as in [15] and [22], where similarity is modeled as correlation in the prior beliefs to facilitate better selection.

In this Chapter, we tackle the R&S problem where the quality of each alternative is measured by a utility function. Without assuming a specific form for the function, we first establish the asymptotically optimal allocation using techniques similar to OCBA and propose sequential selection algorithms based on the results. We then develop a Bayesian VoI approach for efficient learning of the utility rather than the simple expectation and establish the equivalence between the two approaches. We also discuss the issue of numerical computations and point out scenarios where the Bayesian approach could fail. Two numerical experiments using utility functions found in economics and operations research were performed validating the proposed algorithms.

The Chapter is organized as follows: Section 3.2 formulates the R&S problem. Section 3.3 finds the asymptotically optimal allocation configuration for maximizing PCS with a given utility function. Section 3.4 designs a Bayesian VoI dynamic allocation procedure by providing an information measure for selecting the next alternative. Section 3.5 formally states two algorithms based on the theoretical derivations, and Section 3.6 illustrates the performance of the algorithms on two simulation problems.

3.2 Problem Formulation

Let $\mathscr{A} = \{1, 2...k\}$ denote the set of candidate alternatives, each associated with an unknown random outcome $Y_i, i \leq k$, where Y_i follows a distribution with unknown parameter θ_i. Before formulating the problem, we list a set of notations used throughout the paper.

- $U(\cdot)$: the known utility function,
- U_i : the true utility of alternative i computed as $U_i \equiv U(\theta_i)$,
- y_{ij} : the jth sample obtained for alternative i,
- $\hat{\theta}_i$: an estimator of θ_i,
- $\hat{U}_i$: an estimator of U_i,
- N : the total budget,
- i_n : alternative selected at allocation step n,
- $y^{(n)}$: the sample obtained at allocation step n from the chosen alternative i_n,
- n_i: the budget allocated to alternative i.

We assume U is known and the goal is to maximize the utility. For instance, in the case of quantile selection in [40] with normal random rewards where $\theta_i = \{\mu_i, \sigma_i\}$, the quantile utilities are $U_i = \mu_i + \alpha\sigma_i$, where α is the quantile coefficient of a standard normal density. Without loss of generality, we assume that $U_1 \geq U_2 \geq ... \geq U_k$, so that alternative 1 is the best. Upon exhausting the budget N, we make the final selection as

$$i^N = \arg\max_{i \le k} \{\hat{U}_i\}, \tag{3.1}$$

where $\hat{U}_i$ is chosen as $U(\hat{\theta}_i)$ in the UOCBA approach and posterior expectation $\mathbb{E}[U(\hat{\theta}_i)]$ in the Bayesian VoI approach. Then, the problem of designing an allocation that maximizes PCS under a budget constraint N can be formulated as

$$\begin{aligned} &\max_{n_1, n_2 .. n_k} P\{\bigcap_{2 \le i \le k} \hat{U}_1 \ge \hat{U}_i\} \\ &\text{s.t. } \sum_{i=1}^{k} n_i = N. \end{aligned} \tag{3.2}$$

In Section 3.3, we solve (3.2) in the asymptotic domain with $n_i \to \infty$ when $\hat{\theta}$ is the maximum likelihood estimator (MLE). In Section 3.4, we work in the Bayesian framework where posterior densities are assumed on θ_i and updated upon receiving new samples, and design information criteria for dynamically allocating the simulation budget.

3.3 Utility-based Optimal Computing Budget Allocation

In this section, we consider the frequentist problem setting and explicitly find the asymptotically optimal allocation configuration by solving (3.2) when $n_i \to \infty, \forall i \le k$. Under the assumption that $\hat{\theta}_i$ is the MLE of θ, the asymptotic distribution of $\hat{U}$ can be shown with the Δ-method to be normal. Then we approximate PCS with its Bonferroni lower bound and derive the optimal allocation using standard techniques from the OCBA literature.

3.3.1 Asymptotic Distribution of Plug-in Utility Estimator

Most R&S literature works with normal random rewards where the mean μ_i and standard deviation σ_i (i.e. $\theta_i = \{\mu_i, \sigma_i\}$) can fully characterize the unknown random reward. In cases without normality, a simple batching procedure could be applied to obtain approximately normal samples [45]. We do not assume the normality of Y_i, but restrict them to a family of random distributions with the following property.

Assumption 1. $\{Y_i\}_{i=1}^k$ *belong to a family of random variables such that* $\hat{\theta}_i$ *has the asymptotic distribution*

$$\sqrt{n_i}(\hat{\theta}_i - \theta_i) \xrightarrow{\mathscr{D}} \mathcal{N}(0, I^{-1}(\theta_i)) \text{ as } n_i \to \infty, \tag{3.3}$$

where $I(\theta_i)$ *is the corresponding Fisher information matrix.*

For the definition of Fisher information and the exact condition on $\{Y_i\}$ for Assumption 1 to hold, we refer the readers to [46]. In the case of normal random outcomes parametrized by μ_i, σ_i, we have

$$\begin{aligned} \hat{\mu}_i &= \sum_{j=1}^{n_i} y_{ij}/n_i, \\ \hat{\sigma}_i^2 &= \sum_{j=1}^{n_i} (y_{ij} - \hat{\mu}_i)^2/n_i \\ I(\mu_i, \sigma_i) &= \begin{bmatrix} 1/\sigma_i^2 & 0 \\ 0 & 1/(2\sigma_i^4) \end{bmatrix}, \end{aligned}$$

and the corresponding asymptotic distribution of $\hat{\mu}_i, \hat{\sigma}_i$ with the sample size n_i is well-known to be

$$\sqrt{n_i}\,(\hat{\mu}_i - \mu) \xrightarrow{\mathscr{D}} \mathscr{N}(0, \sigma_i^2),$$

$$\sqrt{n_i}\,(\hat{\sigma}_i - \sigma_i) \xrightarrow{\mathscr{D}} \mathscr{N}(0, 2\sigma_i^4).$$

For densities other than normal, we refer the readers to [47] for conditions on which Assumption 1 will hold. Setting $\hat{U}_i$ to be the plug-in estimator $U(\hat{\theta}_i)$, we establish the asymptotic normality of $\hat{U}_i$ with the following lemma.

Lemma 2. *If $U(\cdot)$ is differentiable almost everywhere, then*

$$\sqrt{n}_i(\hat{U}_i - U_i) \xrightarrow{\mathscr{D}} \mathscr{N}(0, v_i^2) \text{ as } n_i \to \infty,$$

where

$$v_i^2 = \nabla^T U(\theta_i) I^{-1}(\theta_i) \nabla U(\theta_i). \tag{3.4}$$

Proof. A direct result of the Δ-technique [47] will prove the Lemma. □

Lemma 2 establishes the normality of $U(\hat{\theta}_i)$, which allows us to find an approximation of PCS when $n_i \to \infty$ and explicitly solve (3.2).

3.3.2 Asymptotically Optimal Allocation Policy

In the case with $n_i \to \infty$, we first construct an approximation of PCS in (3.2) using the asymptotic normality results in Lemma 2. Using the well-known Bonferroni lower bound, we have

$$\begin{aligned} PCS &= P\left\{\bigcap_{2\leq i\leq k} \hat{U}_1 \geq \hat{U}_i\right\} = 1 - P\left\{\bigcup_{2\leq i\leq k}\left(\bigcap_{j\neq i}\hat{U}_i > \hat{U}_j\right)\right\} \\ &= 1 - \sum_{2\leq i\leq k} P\left\{\bigcap_{j\neq i}\hat{U}_i \geq \hat{U}_j\right\} \\ &\geq 1 - \sum_{2\leq i\leq k} P\left\{\hat{U}_i \geq \hat{U}_1\right\} = APCS. \end{aligned}$$

For the ease of derivation, we assume $\{\hat{U}_i\}_{i=1}^k$ are independent. With the normality results in Lemma 2, the term $P\{\hat{U}_i \geq \hat{U}_1\}$ can be expressed in terms of the standard normal cumulative distribution function. Letting

$$\delta_i = U_i - U_1,\ \zeta_i = \frac{\delta_i}{\sqrt{v_i^2/n_i + v_1^2/n_1}}, \tag{3.5}$$

APCS can be written compactly as

$$APCS = 1 - \sum_{2\leq i\leq k} \Phi(\zeta_i), \tag{3.6}$$

where $\Phi(\cdot)$ is the cumulative distribution function for a standard normal distribution. Reaching (3.6) required two approximations: (1) approximating PCS with its Bonferroni lower bound, and (2) approximating $\{\hat{U}_i\}_{i=1}^k$ with their asymptotic normal densities. The quality of the approximations will depend on the exact value of n_i, the true θ values and the utility function U. Careful evaluation of the quality of approximations and their effect on the allocation performance is an active area in OCBA-related research, but beyond the scope of this Chapter. The optimization problem in (3.2) can now be approximated by

$$\max 1 - \sum_{2\leq i\leq k} \Phi(\zeta_i) \quad \text{subject to} \quad \sum_{i=1}^{k} n_i = N, \tag{3.7}$$

by replacing PCS with APCS. The approximate problem has the analytical solution presented in the following theorem.

Theorem 5 (Optimal Allocation for a Given Utility)**.** *Under the conditions of Lemma 2, the APCS is maximized as $N \to \infty$ when the allocations satisfy the conditions*

$$n_1 = \sqrt{v_1 \cdot \sum_{i=2}^{k} \frac{n_i^2}{v_i^2}}, \tag{3.8}$$

$$\frac{n_i}{n_j} = \left(\frac{\delta_i}{\delta_j}\right)^2 \cdot \frac{v_j^2}{v_i^2},\ 2 \leq i, j \leq k, \tag{3.9}$$

$$\sum_{i=1}^{k} n_i = N, \tag{3.10}$$

where v_i and δ_i are defined in (3.4) and (3.5), respectively.

Proof. To solve (3.7), let L be the associated Lagrangian associated with Lagrange multiplier λ. Then

$$L(n_1, n_2, .., n_k, \lambda) = 1 - \sum_{j\geq 2} \Phi(\zeta_j) - \lambda\left(\sum_{1\leq j\leq k} n_j - N\right).$$

The Karush-Kuhn-Tucker (KKT) conditions for solving the problem requires taking derivatives of L w.r.t the allocation configuration $\{n_i\}$ and the coefficient λ. $\frac{\partial L}{\partial \lambda}$ returns the budget constraint. For the allocation configuration variables $n_1, .., n_k$, we have

$$\frac{\partial L}{\partial n_1} = -\sum_{j \geq 2} \frac{\partial \Phi(\zeta_j)}{\partial \zeta_j} \frac{\partial \zeta_j}{\partial n_1} - \lambda = 0, \tag{3.11}$$

and

$$\frac{\partial L}{\partial n_j} = -\frac{\partial \Phi(\zeta_j)}{\partial \zeta_j} \frac{\partial \zeta_j}{\partial n_j} - \lambda = 0, \forall j \geq 2. \tag{3.12}$$

Using the fact that $\partial \Phi(\zeta)/\partial \zeta = \frac{1}{\sqrt{2\pi}} e^{-\zeta^2/2}$, we can write out Equation (3.11) as

$$\begin{aligned} \sum_{j \geq 2} \frac{\partial \Phi(\zeta_j)}{\partial \zeta_j} \frac{\partial \zeta_j}{\partial n_1} &= \sum_{j \geq 2} \frac{1}{2\sqrt{2\pi}} \cdot \frac{v_1^2}{n_1^2} \cdot \delta_j \cdot (\frac{v_j^2}{n_j} + \frac{v_1^2}{n_1})^{-\frac{3}{2}} \cdot e^{-\frac{1}{2}\zeta_j^2} \\ &= \frac{1}{2\sqrt{2\pi}} \frac{v_1^2}{n_1} \sum_{j \geq 2} \delta_j \cdot (\frac{v_j^2}{n_j} + \frac{v_1^2}{n_1^2})^{-\frac{3}{2}} \cdot e^{-\frac{1}{2}\zeta_j^2} = -\lambda. \end{aligned} \tag{3.13}$$

The terms within the summation can be replaced using results from Equation (3.12), from which we can obtain the relationship between n_1 and $n_j, \forall j \geq 2$. From Equation (3.12), we have

$$\frac{1}{2\sqrt{2\pi}} \frac{v_j^2}{n_j^2} \cdot \delta_j \cdot (\frac{v_j^2}{n_j^2} + \frac{v_1^2}{n_1^2})^{-\frac{3}{2}} \cdot e^{-\frac{\zeta_j^2}{2}} = -\lambda, \forall j \geq 2,$$

which can be re-arranged into

$$\delta_j \cdot (\frac{v_j^2}{n_j^2} + \frac{v_1^2}{n_1^2})^{-\frac{3}{2}} \cdot e^{-\frac{\zeta_j^2}{2}} = 2\sqrt{2\pi}\lambda \cdot \frac{n_j^2}{v_j^2}, \forall j \geq 2. \tag{3.14}$$

Substitute Equation (3.14) into the summation in Equation (3.13),

$$\frac{1}{2\sqrt{2\pi}}\frac{v_1^2}{n_1^2}\cdot\sum_{j\geq 2}2\sqrt{2\pi}\lambda\frac{n_j^2}{v_j^2}=-\lambda.$$

The relationship between the allocated budget on the optimal alternative, n_1, and budgets allocated to other alternatives is obtained by canceling out the constants on both sides as

$$n_1=v_1\sqrt{\sum_{j\geq 2}\frac{n_j^2}{v_j^2}}. \tag{3.15}$$

We then proceed to obtain the relationship between the optimal allocation budget on the sub-optimal alternatives. For $\forall i,j\neq 1$ and $i,j\in\mathscr{A}$, from Equation (3.12), we know that

$$\frac{\partial\Phi(\zeta_i)}{\partial\zeta_i}\frac{\partial\zeta_i}{\partial n_i}=\frac{\partial\Phi(\zeta_j)}{\partial\zeta_j}\frac{\partial\zeta_j}{\partial n_j}.$$

Plug in the definition of ζ_i from Equation (3.5), the formula can be written out as

$$\frac{1}{2\sqrt{2\pi}}e^{-\frac{1}{2}\cdot\frac{\delta_j^2}{v_j^2/n_j+v_1^2/n_1}}(\frac{\delta_j v_j^2}{n_j^2})(\frac{v_j^2}{n_j}+\frac{v_1^2}{n_1})^{-\frac{3}{2}}=\frac{1}{2\sqrt{2\pi}}e^{-\frac{1}{2}\cdot\frac{\delta_i^2}{v_i^2/n_i+v_1^2/n_1}}(\frac{\delta_i v_i^2}{n_i^2})(\frac{v_i^2}{n_i}+\frac{v_1^2}{n_1})^{-\frac{3}{2}}.$$

Combined with Equation (3.14) and the budget constraint $\sum_{i\in\mathscr{A}}n_i=N$, the set of equations can be used to solve for $n_i,\forall i\in\mathscr{A}$ explicitly to obtain the optimal allocation for APCS. However, as we work in the asymptotic domain, we further simplify the above equation to obtain tractable results. Canceling the constants and taking the natural log on both sides we would have

$$\begin{aligned}
&\frac{1}{2}\frac{\delta_j^2}{v_j^2/n_j+v_1^2/n_1}-\log(\delta_j v_j^2)-2\log n_j+\frac{3}{2}\log(\frac{v_j^2}{n_j}+\frac{v_1^2}{n_1})\\
=&\frac{1}{2}\frac{\delta_i^2}{v_i^2/n_i+v_1^2/n_1}-\log(\delta_i v_i^2)-2\log n_i+\frac{3}{2}\log(\frac{v_i^2}{n_i}+\frac{v_1^2}{n_1}).
\end{aligned}$$

In the case where $n_i \to \infty, \forall i \in \mathscr{A}$, we examine the asymptotic order of the terms on both sides and only keep the dominating ones for simplification. Using the $O(\cdot)$ notation, it is easy to obtain that

$$\begin{aligned}
&\frac{\delta_j^2}{v_j^2/n_j+v_1^2/n_1}=O(\max\{n_1,n_j\}),\\
&\log(\frac{v_j^2}{n_j}+\frac{v_1^2}{n_1})=O(\log(\max\{n_1,n_j\})).
\end{aligned}$$

Only keeping the $O(\max\{n_1,n_j\})$ term gives us the equation

$$\frac{\delta_j^2}{v_j^2/n_j+v_1^2/n_1}=\frac{\delta_i^2}{v_i^2/n_i+v_1^2/n_1}, \forall i,j\neq 1. \tag{3.16}$$

Before we simplify Equation (3.16),we rewrite Equation (3.15) as

$$\frac{n_1}{n_j}=v_1\sqrt{\sum_{k\geq 2}\frac{1}{v_k^2}(\frac{n_k}{n_j})^2}.$$

As our framework is most useful in R&S problem instances with a moderate to large k, we make the further assumption that $n_1/n_j \to \infty, \forall n_j \geq 2$. The denominator on both sides of Equation (3.16) can be further simplified by ignoring the v_1^2/n_1 term to obtain the result

$$\frac{n_j}{n_i} = \frac{(\delta_i/v_i)^2}{(\delta_j/v_j)^2}, \forall i, j \neq 1. \tag{3.17}$$

Combining Equations (3.15) and (3.17) yields the result stated in Theorem 5 . □

It is worth mentioning in the case of Y_i having normal densities with expected value being the utility, Theorem 5 reduces to the usual OCBA optimal allocation results.

3.4 Utility-based Bayesian VoI

We also attempt to tackle the problem by developing a variation of the Bayesian VoI technique for efficient learning of the unknown utility. In the frequentist setting, the utility is viewed as a function of fixed unknown distributional parameters, whereas in the Bayesian setting, we treat the utilities as random variables and design an information criteria for dynamically allocating the simulation budget. We first present two Bayesian models which we later use in our numerical experiments, and propose the expected utility improvement and establish its asymptotic equivalence with the UOCBA allocation results in Theorem 5.

3.4.1 Bayesian Models and VoI

Two common Bayesian models are the Normal-Normal and Beta-Bernoulli models, where tractable posterior updates are readily available.

Normal-Normal posterior updates: In the Bayesian model with known normal priors on the mean parameters $\mu_i \sim \mathcal{N}(t_i^0, (\tau_i^0)^2)$ and normally distributed samples $Y_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$ where σ_i is known, the posterior of μ_i at allocation step n is $\mathcal{N}(t_i^n, (\tau_i^n)^2)$ with the updates on the parameters

$$t_i^{n+1} = \begin{cases} \frac{(\tau_i^n)^{-2} t_i^n + \sigma_i^{-2} y^{n+1}}{(\tau_i^n)^{-2} + \sigma_i^{-2}}, & i_n = i \\ t_i^n, & i_n \neq i, \end{cases} \quad \left(\tau_i^{n+1}\right)^2 = \begin{cases} \left((\tau_i^n)^{-2} + \sigma_i^{-2}\right)^{-1}, & i_n = i \\ (\tau_i^n)^2, & i_n \neq i. \end{cases} \tag{3.18}$$

Beta-Bernoulli Posterior Updates: If probability of success p_i have $Beta(\alpha_i^0, \beta_i^0)$ priors, upon receiving y^n which is either 0 or 1, the posterior of p_i is $Beta(\alpha_i^n, \beta_i^n)$ with the parameter updates

$$\alpha_i^{n+1} = \begin{cases} \alpha_i^n + \mathbb{1}_{\{y^n=1\}}, & i_n = i, \\ \alpha_i^n, & i_n \neq i, \end{cases} \quad \beta_i^{n+1} = \begin{cases} \beta_i^n + \mathbb{1}_{\{y^n=0\}}, & i_n = i, \\ \beta_i^n, & i_n \neq i, \end{cases} \tag{3.19}$$

where $\mathbb{1}_{\{\cdot\}}$ is the indicator function.

VoI tends to favor alternatives with higher uncertainty or higher estimated mean. Two important examples are Expected improvement (EI) and Knowledge Gradient (KG). In this Chapter, we use a variant of EI for its simplicity in computation and its connection with the UOCBA policy.

3.4.2 Expected Utility Improvement

Denote the priors on $\theta_i,\ i \leq k$ by f_i^0 and the posteriors at step n by f_i^n. VoI seeks to measure the potential gain of learning θ_i by balancing the exploration-exploitation trade-off [20]. We propose the expected utility improvement (EUI)

$$g_i^{EUI,n} = \mathbb{E}_{f^n}\left[(U(\theta_i) - U^*)^+\right], \tag{3.20}$$

where the expectation is taken with respect to (w.r.t.) $\theta_i \sim f_i^n$ and $U^* = \max_{i \leq k}\{\mathbb{E}[U(\theta_i)]\}$ is the current expected optimal utility under the posteriors $\{f_i^n\}_{i=1}^k$. The EUI-based policies select the alternative with maximum EUI at allocation step n as

$$i_n = \underset{1\leq i\leq k}{\arg\max}\{g_i^{EUI,n}\}. \tag{3.21}$$

In the special case of Normal-Normal Bayesian models, similar to the results in [3], we have the following theorem relating the asymptotic allocations of EUI and UOCBA policies.

Theorem 6. *In the case of Normal-Normal Bayesian model with $\tau_i^0 \to \infty$ and $N \to \infty$, let n_i^{EUI} and n_i^{UOCBA} denote, respectively, the budget allocated to alternative i under the EUI and UOCBA policies in Equation (3.20) and Theorem 5, we have*

$$\frac{n_i^{UOCBA}}{n_j^{UOCBA}} \to \frac{n_i^{EUI}}{n_j^{EUI}},\ \forall i \neq j,\ i \neq 1,\ j \neq 1. \tag{3.22}$$

Proof. When $\tau_i^0 \to \infty$, we have $t_i^n = \bar{y}_i, \tau_i^n = \frac{\sigma_i}{\sqrt{n_i}}$. When $n_i \to \infty$, a direct application of the Δ-technique yields $U(\mu_i) \xrightarrow{\mathscr{D}} \mathscr{N}\left(U(\bar{y}_i), |\frac{\partial U}{\partial \mu_i}|^2 \cdot \frac{\sigma_i^2}{n_i}\right)$. The EUI computation is then effectively a usual EI computation on the random variable U with normal posterior densities. Applying the results in [3] on convergence rates of EI methods yields the theorem. □

The asymptotic result in Theorem 6 holds only for sub-optimal alternatives $(i, j \neq 1)$. [44] derived variants of EI policies that achieve asymptotically optimal allocation ratio for all alternatives, which can be easily applied to our scenario. We use the most basic EI policy for simplicity.

3.4.3 Practical Computation of Expected Utility Improvement

The original expected improvement policy in [48] was developed under a normal distribution assumption and has the closed-form expression

$$g^{EI,n} = (t^n - t^{*,n})\Phi(\frac{t^n - t^{*,n}}{\tau^n}) + \tau^n \phi(\frac{t^n - t^{*,n}}{\tau^n}) \tag{3.23}$$

where n is the sampling step, t^n and τ^n are the mean and standard deviation, respectively, of the posterior normal density following the notations in (3.18), and $t^{*,n}$ is the current threshold for improvement. It is easy to see that g^{EI} increases with τ for any $t \leq t^*$, therefore favors alternatives with higher uncertainty. However, in Equation (3.20), a higher uncertainty in θ does not necessarily lead to higher EUI. One such example is $U(\theta) = -e^{-4\theta} - \theta$, which we tested in Section 3.6.2. As illustrated in Figure 3.1, a higher uncertainty in θ in terms of higher variance leads to a smaller g^{EUI}, causing the EUI aproach to fail in the numerical experiments in Section 3.6.2. The EUI approach in (3.23) no longer encourages exploration, and we defer addressing the issue to future research.

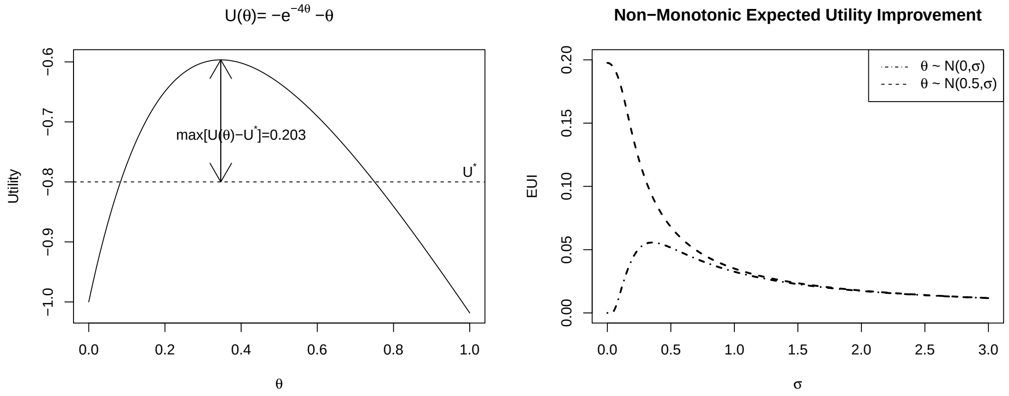

Figure 3.1: Left: A utility function. Right: Higher uncertainty leads to lower EUI.

3.5 Utility-based Allocation Algorithms

Using Theorem 5 and Equation (3.20), we design two fully sequential procedures for utility-based allocation problems, which we refer to as most-starving-UOCBA (MS-UOCBA) and EUI.

MS-UOCBA: An initial sampling budget n_0 is allocated to each alternative to obtain estimates of the unknown parameters in Theorem 5. At each time step, Equations (3.8), (3.9) and (3.10) are solved to find the estimated optimal allocations under the given total budget. The alternative that is furthest away from its currently estimated optimal allocation is selected. This "most-starving" implementation is fully sequential except for the initialization batch of samples.

EUI: The algorithm requires inputs for specifying the priors and posteriors updates, and therefore depends on the specific Bayesian model used for specific application problems.

```
Algorithm 2: $MS-UOCBA$
    Input: total budget $N$, initial budget $n_0$, utility function $U$
    Output: the final selection $i^N$
    Allocate $n_0$ to each alternative
    Compute $\hat{\theta}_i$, $U(\hat{\theta}_i)$ and $\nabla U(\hat{\theta}_i)$
    Set counter $n \leftarrow kn_0$ and $m_i \leftarrow n_0,\ i = 1, 2, \dots, k$
    while $n \leq N$ do
        compute $n_i$ by solving Equations (3.8), (3.9) and (3.10)
        select $i_n \leftarrow \arg\max\{n_i - m_i\}$
        update $m_{i_n} \leftarrow m_{i_n} + 1$ and $n \leftarrow n + 1$
        update $\hat{\theta}_{i_n}$, $U(\hat{\theta}_{i_n})$, $I(\hat{\theta}_{i_n})$, and $\nabla U(\hat{\theta}_{i_n})$
    return $i^N = \arg\max_i\{U(\hat{\theta}_i)\}$
```

```
Algorithm 3: EUI
    Input: the priors $f_i^0$, total budget $N$, utility function $U$
    Output: the final selection $i^N$
    Initialization: set $n = 0$ and select $i_0 = \arg\max_{i \leq k} v_i^{EUI,0}$
    while $n \leq N$ do
        collect one sample on alternative $i_n$ as $x_{i_n}$
        update $f_i^n(\theta_i)$ for $i = i_n$
        set $f_i^n = f_i^{n-1}$ for $i \neq i_n$
        compute $v_i^{EUI,n}$ with Equation (3.20)
        select $i_n = \arg\max_{i \leq k}\{v_i^{EUI,n}\}$
        update $n \leftarrow n + 1$
    return $i^N = \arg\max_{i \leq k} \mathbb{E}[U(\theta_i)]$
```

Computational considerations: in the MS-UOCBA algorithm, the most computationally intensive step would be solving for optimal allocation using Equations (3.8), (3.9) and (3.10). However, for the Bayesian approach, the posterior updates and computation of $v_i^{EUI,n}$ could be non-trivial, depending on the exact form of utility and prior-posterior pairs. We assume relevant computations are more efficient compared to obtaining an output from the simulation model, thus justifying the overhead for efficient sequential allocation.

3.6 Numerical Experiments

We test the performance of Algorithms 2 and 3 on two simulated experiments by comparing their performance with the simple equal allocation (EA) and usual OCBA allocation policies. The first experiment selects from alternatives with binary rewards and employs the beta-Bernoulli conjugate pair outlined in Equation (3.19) when implementing EUI. The second works with continuous random rewards and uses Normal-Normal Bayesian model in Equation (3.18).

3.6.1 Binary Rewards with Prospect Theoretic Utility

This experiment is motivated by the prospect theory which was awarded the 1992 Nobel Prize in economics for its effectiveness in modeling people's preference in fair games [37]. Let $\mathscr{A} = \{1, 2..k\}$ denote a set of lotteries and $Y_i, i \leq k$ be the Bernoulli random variable representing the outcome of a lottery ticket, i.e.,

$$Y_i = \begin{cases} 0, & \text{w.p. } 1 - p_i, \\ 1 & \text{w.p. } p_i, \end{cases}$$

where p_i are the unknown winning probabilities. Let a_i denote the winning prize of lottery i and b_i be cost of buying a lottery ticket. The prospect-theoretic utility [37] for a lottery has the form

$$U(p_i) = (a_i - b_i)p_i^{w_1} - b_i(1 - p_i)^{w_2},$$

with weights w_1 and w_2 reflecting people's perception of gains and losses. In the context of R&S, we assume a customer chooses from 19 different types of lotteries before committing to a favorite one with preference modeled by a prospect theoretic utility with weights $w_1 = 1.1$ and $w_2 = 100$. The winning probabilities are set to be $\{0.05,...,0.90,0.95\}$ with rewards $a_i = 1/p_i$ and cost $b_i = 1$, $\forall 1 \le i \le 19$, such that the expected net gain of all lotteries will be 0. Under the utility above, the optimal lottery will be the 2nd type with $p_2 = 0.1$ and $a_2 = 10$ in this setting, as it offers a large reward as well as a reasonable chance of winning. The simulated R&S problem is illustrated in Figure 3.2.

Implementation of MS-UOCBA: Given observed outcomes of Y_i, UOCBA first estimates the winning probabilities, and then updates the estimation of v_i and n_i for dynamic allocation. For lottery i, after n_i trials, we have

$$\hat{p}_i = \frac{\sum_{j=1}^{n_i} y_{ij}}{n_i}, \;\; \hat{v}_i = |w_1(a_i - b_1)\hat{p}_i^{w_1-1} + b_i(1-\hat{p}_i)^{w_2-1}|\sqrt{\hat{p}_i(1-\hat{p}_i)}).$$

We choose the initialization budget kn_0 to be 20% of the total budget N.

Implementation of EUI: We set the priors on p_i to be the flat beta distribution $Beta(1,1)$ for all $i \in \mathscr{A}$. Upon collecting new observations, the update is performed according to Equation (3.19). Given a beta posterior with shape parameters α and β, the posterior expected utility has the closed-form formula

$$\mathbb{E}[U(p_i)] = \frac{(a_i - b_i)B(\alpha + w_1, \beta) - b_i B(\alpha, \beta + w_2)}{B(\alpha, \beta)},$$

where $B(\alpha,\beta)$ is the Beta-function defined as $B(\alpha,\beta) = \int_0^1 x^{\alpha-1}(1-x)^{\beta-1}dx$. The expected improvement in Equation (3.20) is computed with numerical integration routine **integrate** in **R**. For posteriors with large shape parameters ($>$ 1000 in our experiments), a Monte Carlo integration is performed to compute the posterior expected utility and expected improvement with 1000 samples to avoid the numerical instability with extraordinarily small ($< 10^{-100}$) $B(\alpha,\beta)$.

We compare the performance of MS-OCBA and EUI using the above implementations with the Equal Allocation (EA) policy and the MS-OCBA policy for budgets ranging from 100 to 10000. For the MS-OCBA policy, we use the same implementation with MS-UCOBA with $U(\hat{p})$ set to be $\hat{p}$ and v_1 set as $\sqrt{p_i(1-p_i)}$. The PCS for each allocation algorithm is estimated using 1000 simulation replications.

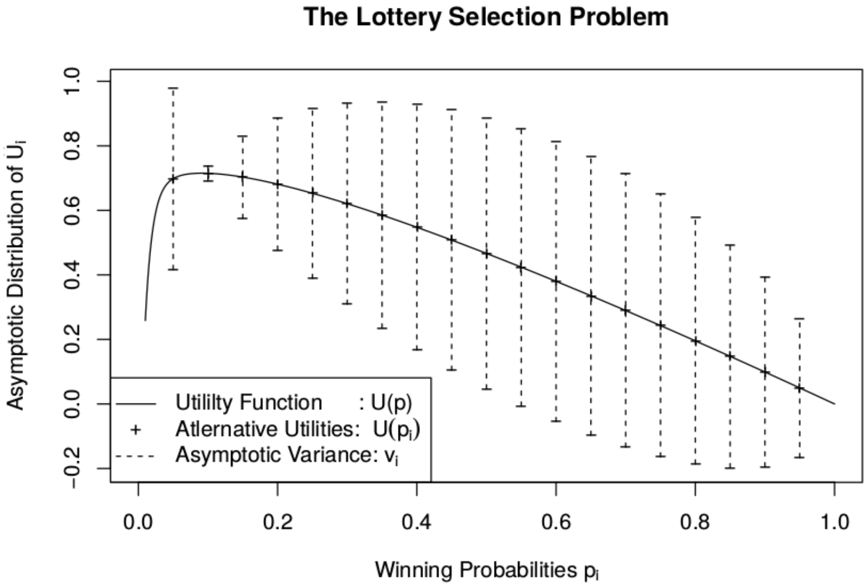

Figure 3.2: The prospect utility function $U(\cdot)$ and asymptotic distribution of the utility estimator $U(\hat{p}_i)$. The second lottery with winning probability 0.2 has the highest true utility ($U(p)$). The variances of the utility estimators $U(\hat{p}_i, i \leq 10)$ decrease in regions where U is flat.

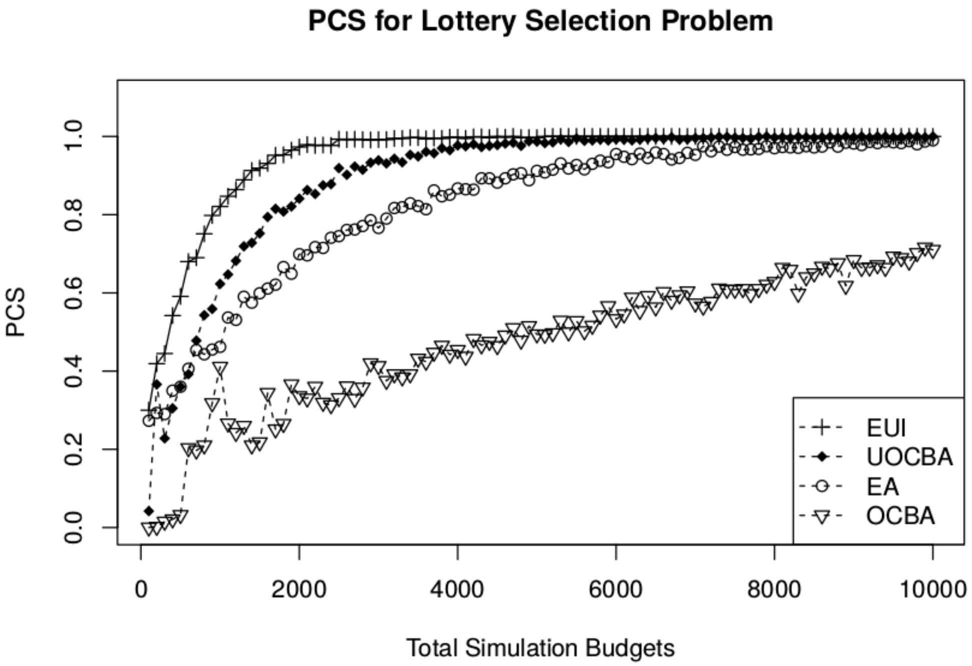

Figure 3.3: PCS under different allocation budgets.

The simulation results are presented in Figure 3.3. EUI has the best performance among all policies, with MS-UOCBA being the second best. The usual OCBA policy has the worst performance, as it devotes most of its allocation budget on alternatives with large p_i, which have relatively low values under the given utility measure.

3.6.2 Staffing with a Cost Utility

A company staffing a service center is often faced with the trade-off between quality of services and staffing costs (such as training and compensation). Let Y_i denote the service times, which are assumed to be normally distributed, i.e.,

$$Y_i \sim \mathcal{N}\left(\mu_i, \sigma_i^2\right)$$

where μ_i is unknown and σ_i is assumed to be 1. A smaller value of μ_i indicates higher service quality but requires higher training and wage costs. A utility function of the form $U(\mu_i) = -C(\mu_i) - Q(Y_i)$ in [49] is often used to capture the trade-off, where $C(\cdot)$ denotes the cost and $Q(\cdot)$ denotes the service quality. We use negative terms to make the problem a maximization rather than minimization, for consistency with our problem setting. We test two utilities

$$U_1(\mu) = e^{10\mu - 10}, \tag{3.24}$$

$$U_2(\mu) = -e^{-4\mu} - \mu, \tag{3.25}$$

where U_1 is monotonically increasing with μ, and U_2 balances between cost and quality.

Implementation of MS-OCBA: Given collected samples drawn from Y_i, MS-UOCBA estimates μ_i with sample averages, and v_i for two two utilities (denoted as $v_i^{U_1}$ and $v_1^{U_2}$ respectively) can be easily computed to be

$$\hat{\mu}_i = \frac{\sum_{j=1}^{n_i} y_{ij}}{n_i}, \tag{3.26}$$

$$v_i^{U_1} = |10e^{10\mu-10}|, \quad v_i^{U_2} = |4e^{-4\mu} + 1|. \tag{3.27}$$

The initialization budget kn_0 is chosen to be 20% of the total budget N.

Implementation of EUI: Given normal random observations, we use the Normal-Normal model outlined in Equation (3.18). The priors are set to be $t_i^{(0)} = 0$ and $\tau_i^{(0)} =$

1000 to create flat priors. Given a posterior $\mathcal{N}(t_i^n, (\tau_i^n)^2)$, the posterior expected utilities can be computed as

$$\mathbb{E}[U_1(\mu_i)] = e^{10t_i^n + 50\left(\tau_i^n\right)^2}, \quad \mathbb{E}[U_2(\mu_i)] = -t_i^n - e^{-4t_i^n + 2\left(\tau_i^n\right)^2\left(t_i^n\right)^2}.$$

EUI in Euqation (3.20) for U_1 and U_2 under the assumed Normal-Normal conjugate pair also has the closed-form expressions

$$\begin{aligned}
EUI(U_1^*) &= e^{-10}[1 - \Phi(\frac{t_i^n - x_1^c}{\tau_i^n})] + e^{10t_i^n + 50\left(\tau_i^n\right)^2 - 10}[1 - \Phi(\frac{t + 10\,(\tau_i^n)^2 - x_1^c}{\tau_i^n})], \\
EUI(U_2^*) &= (U_2^* + t_i^n)\left(\Phi\left(\frac{x_2^l - t_i^n}{\tau_i^n}\right) - \Phi\left(\frac{x_2^u - t_i^n}{\tau_i^n}\right)\right) + \tau_i^n\left(\phi\left(\frac{x_2^l - t_i^n}{\tau_i^n}\right) - \phi\left(\frac{x_2^u - t_i^n}{\tau_i^n}\right)\right) \\
&\quad + e^{4t_i^n + 8\left(\tau_i^n\right)^2}\left(\Phi\left(\frac{x_1^l - t_i^n - 4\,(\tau_i^n)^2}{\tau_i^n}\right) - \Phi\left(\frac{x_1^u - t_i^n - 4\,(\tau_i^n)^2}{\tau_i^n}\right)\right),
\end{aligned}$$

where $U_1(x_1^c) = U_1^*$ and $U_2(x_2^l) = U_2(x_2^u) = U_2^*$ with $x_2^l \leq x_2^u$. Under this notation, $[x_1^c, \infty)$ and $[x_2^l, x_2^u]$ will be the regions where the improvement function is positive for computing EUI for utility functions U_1 and U_2. The priors are chosen to be $\mathcal{N}(0, 4)$ for initializing the unknown μ_i inside the *interesting* region. At each time step, the alternative with maximum EUI is chosen. If there is a tie in EUI, one of the alternatives with the maximum EUI is randomly selected. In both tests, EUI allocation steps are terminated early due to clear convergence.

Implementation of MS-OCBA and EA: The MS-OCBA algorithms are implemented using the same setup for MS-UOCBA. For total simulation budgets ranging from 100 to 10000, a simulation replication of 1000 is used to estimate PCS. The numerical results are presented in Figure 3.4.

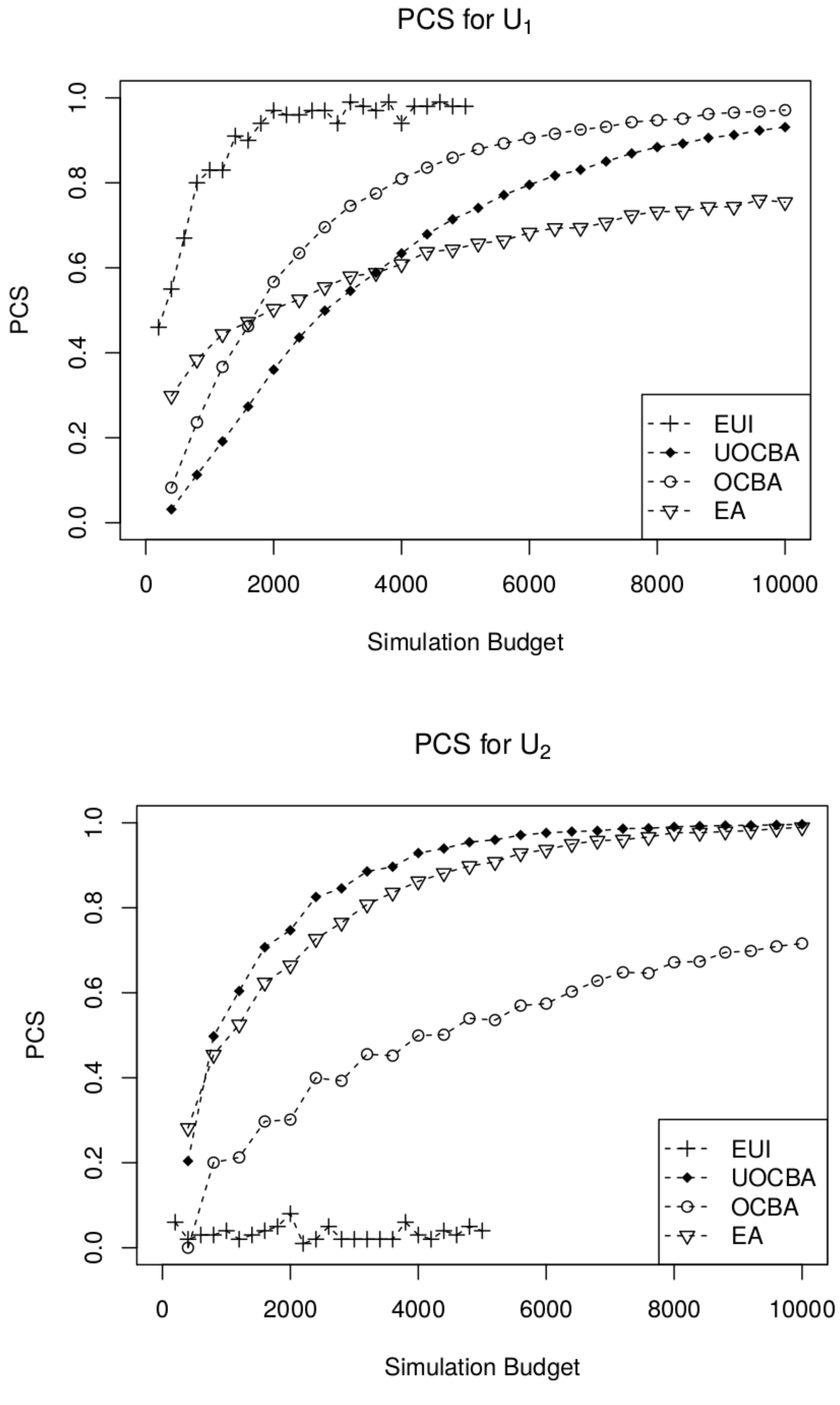

Figure 3.4: Performance of allocation algorithms for the two utilities U_1 and U_2 .

For U_1, the EUI algorithm outperforms all other algorithms. MS-OCBA outperforms MS-UOCBA, as U_1 being a monotoe function, both MS-OCBA and MS-UOCBA will treat alternative 20 as the true optimal choice and allocates budgets on alternatives closer to 20. Despite being optimal under their respective selection rule (OCBA with $i^N = \arg\max\{\hat{\mu}_i\}$ and UOCBA with $i^N = \arg\max\{U_1(\hat{\mu}_i)\}$), there is no easy theoretical analysis on their relative performance. The EA policy outperforms both MS-OCBA and MS-UOCBA when the total budget is small, as OCBA policies are known to be sensitive to initialization noise. For U_2, MS-UOCBA is the best among all tested algorithms. Despite the asymptotic optimality result in Equation (6), the shape of U_2 causes the EUI policy to fail in this problem, as higher uncertainty leads to a smaller expected utility improvement, causing the algorithm to perform similar to a uniform random choice, which we discussed in Section 3.4.3.

3.7 Concluding Remarks

In this Chapter, we considered the R&S problem where the selection objective can be expressed as a utility function of the observed random samples. We established a plug-in utility estimator to derive an asymptotically optimal allocation policy and provided insight on how a utility function would affect the optimal allocation policy. The main result in Theorem 5 can be easily extended to cases where the MLE is not easy to obtain, but some other parameter estimator with the same convergence rate, such as the method of moments estimator, could still be applied. We also developed a variation of the Bayesian VoI approach that showed better finite budget performance. We proposed two sequential allocation algorithms and discussed their practical implementations, including a scenario where the Bayesian VoI approach could fail. To the best of our knowledge, this is the first attempt at extending R&S techniques for general utility measures.

Chapter 4: Restless Temporal Bandits

4.1 Introduction

The multi-armed bandit (MAB) problem is a popular formulation to study the exploration-exploitation trade-off in various communities, where the goal is to maximize the cumulative reward through some time horizon T. It was introduced as early as the 1950s in the context of clinical trials [50] and has found numerous modern applications, such as recommendation systems and marketing [51]. Usually, a reward process $\{X_t^k\}$ is associated with arm k, and a realization of the corresponding process will be received by the player upon choosing the corresponding arm. The most popular and well studied assumption is that X_t^k are sequences of i.i.d. random variables. Under this assumption, at each time step the player has to balance between choosing arms with currently estimated high expected rewards (exploitation) and sampling on lesser known arms to obtain better estimates of expected rewards (exploration). A few important results under this line of research includes the $O(\log T)$ and $O(\sqrt{T})$ regret lower bounds [52] and computationally tractable policies that would achieve matching upper bound including the UCB policy [7, 53] and Thompson sampling policy [8, 54].

However, in many practical applications well suited for bandit algorithms, the i.i.d. assumption may not be appropriate. Consider the problem of recommending items on an e-commerce website. It is well known that customer preference varies in time with daily/monthly/seasonal periodic patterns, and the observed past could have strong predictive power into the future [55]. Thus, it is intuitive to adjust the recommendation policy to utilize the known patterns to achieve better revenue. The volatility in the environment is well acknowledged in the bandit community and has been studied in the context of bandit algorithms with Markovian [56], Brownian [57] or non-stationary [58] reward processes, to name a few. We follow the above literature and focus our efforts on the *restless* setting where the reward processes $\{X_t^k\}$ evolve regardless of whether the arm is being played or not. We emphasize that the reward processes $\{X_t^k\}$ evolve independently across the arm index k, while exhibits strong correlation on the time index t.

Despite the richness of related literature, the standard tool used in statistics and economics for modeling actual observed stochastic processes with temporal correlations, the autoregressive and moving average (ARMA) time series model [55], is relatively overlooked by the bandit community. We use the general form of ARMA processes to study the effect of temporal correlations on regret in bandit problems. The estimation and prediction of time series data is itself an active area of research, and many recent machine learning techniques such as Gaussian process models and convolutional LSTM have shown great promise in many applications [59, 60]. We assume the processes X_t^k are known, since (1) we are mainly interested in the effects of temporal correlations rather than studying how to estimate them; (2) time series model specification and estimation is beyond the scope of this Chapter; (3) many economic patterns are well studied with known trend or periodic pattern.

4.1.1 Non-stationary Regret and Temporal Oracle Process

Let $\mathscr{K} = \{1,2,...,K\}$ denote the set of arms and $\{X_t^k\}, k \in \mathscr{K}$ be the reward processes. Restricting to the family of stationary processes, the expected values are fixed throughout the time horizon, and we denote them as $\mu_k = \mathbb{E}[X_1^k]$ and use $\mu^* = \max_{i\in\mathscr{K}}\{\mu_i\}$ to denote the maximum of the stationary means. According to a policy Π, at times $1,2,..,t$, a sequence of arms $(a_1,...,a_t), a_i \in \mathscr{K}, i \leq t$ is chosen and the rewards $x_1^{a_1},...,x_t^{a_t}$ are revealed. Use $\mathscr{F}_t$ to denote the filtration generated by the sequence of observations representing the information available to the player. Use $\mathscr{H}_t$ to denote the entire history of the reward processes up to time t. It is easy to see that $\mathscr{F}_t \subseteq \mathscr{H}_t$. In the seminal paper of Lai and Robbins [52], the regret R_T over horizon T is defined as

$$R_T = \sum_{t=1}^{T} \max_{k\in\mathscr{K}} \mathbb{E}[X_t^k|\mathscr{F}_{t-1}] - \sum_{i=1}^{T} \mathbb{E}[X_t^{a_t}|\mathscr{F}_{t-1}]. \tag{4.1}$$

In the i.i.d setting, the conditional expectation would reduce to the simple static means, leading to the usual expression of regret

$$R_T = T\mu^* - \mathbb{E}[\sum_{t=1}^{T} \mu_{a_t}]. \tag{4.2}$$

The optimal policy minimizing regret defined in Equation (4.2) would be to always choose the arm with maximum stationary mean, which we refer to as the *static optimal policy* throughout this Chapter. With temporal correlations, we study the *dynamic regret* defined as

$$R_T = \mathbb{E}_{X_t^k, k\in\mathscr{K};\Pi}\{\sum_{t=1}^{T} \max_{k\in\mathscr{K}}\{X_t^k|\mathscr{F}_{t-1}\} - \sum_{t=1}^{T}\mathbb{E}[X_t^{a_t}|\mathscr{F}_{t-1}]\}. \tag{4.3}$$

The conditioning reflects the fact that past information $\mathscr{F}_{t-1}$ could affect the outcome in the future.

Definition 4 (Oracle Process and Instantaneous Regret)**.** *Let $S_t := \arg\max_{k\in\mathscr{K}}\{X_t^k\}$ represent the sequence of optimal arms in each realization of $X_t^k, \forall k \in \mathscr{K}$ and define it as the Temporal Oracle Process. Denote $\mu_t^S := X_t(S_t)$ as the corresponding reward sequence. Define the instantaneous regret at time t, given $\mathscr{F}_{t-1}$ and the chosen arm a_t, as*

$$r_t = \mathbb{E}[\mu_t^S|\mathscr{F}_{t-1}] - \mathbb{E}[X_t^{a_t}|\mathscr{F}_{t-1}]. \tag{4.4}$$

$\{S_t\}$ will be a stochastic process taking values in the set of arms $\mathscr{K}$ adapted to the filtration $\mathscr{H}_t$. The revealed information $\mathscr{F}_{t-1}$ is shared by the temporal oracle process and the player, so r_t is capturing the gap between the reward obtained by the policy and the optimal reward when part of $\mathscr{H}_t$ is revealed and fixed (as $\mathscr{F}_{t-1} \subseteq \mathscr{H}_{t-1}$). Though we will be working with stationary stochastic processes, S_t bridges our work with the non-stationary [58] and adversarial bandit [61] research in the sense that the optimal arm and its rewards are varying with time. In the existing research, the non-stationarity is usually assumed to be a fixed unknown sequence, while in our setting the non-stationarity is stochastic, as in S_t. In later discussions, we will mostly focus on r_t, as the volatility in S_t would lead to a constant lower bound on regret at each time step.

Our work can be motivated and visualized by Figure 4.1. Policies that utilize the

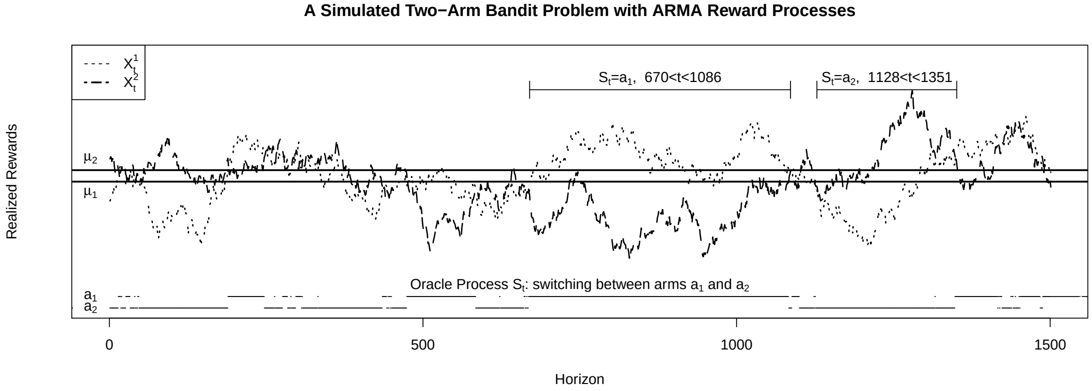

Figure 4.1: Illustration of a simulated two-arm bandit problem with ARMA reward processes: (a) The two arms have different stationary expected values μ_1 and μ_2. (b) The static optimal policy is to always choose a_2. (c) The oracle process $\{S_t\}$ is a stochastic process adapted to the filtration $\mathscr{H}_t$, taking values in the set $\{a_1, a_2\}$. (d) The arm with lower stationary rewards could have predictable higher future reward, given information on its past. (e) Policies that efficiently discover the dynamically optimal arm can accumulate much higher reward compared to the static optimal policy, such as in times between 670 and 1086 in this simulation.

predictive power coming from temporal correlations to efficiently discover the dynamically optimal arm could have better performance than a policy that works with independent rewards.

4.1.2 Related Research

Relaxing the i.i.d. assumption in bandit problems has drawn much attention recently. One line of research maintains the independence assumption across the time index and models the volatile environment as variations in the sequence of unknown expected rewards. Sublinear regret bounds have been proved conditioning on a known form of total variation in the mean values of rewards, and policies that gradually forget earlier biased samples have been proposed, achieving matching upper bounds [58, 62–64]. A similar setting is the adversarial setting, where the sequence of expected rewards are chosen by an adversary [61]. More closely related to our approach is previous work that assumes specific reward processes, including Markovian reward process [56], Lévy process [65], rotting processes [66], Brownian process [57] and mixing process [67]. [57] considers the discretized Brownian motion, which is a non-stationary ARMA(1,1) process, therefore a different yet simpler setup than ours. In [67], the authors assume the mean rewards are unknown and argue that temporal correlations would lead to inferior regret compared to the static optimal policy, therefore orthogonal to our work, as we assume the correlations are known and focus on maximizing a linear improvement over the static policy. With the close connection between ARMA processes and Kalman filters, it is also worth mentioning the research on studying bandits with rewards driven by state-space models [68, 69], which largely focus on theoretical analysis of indexability rather than characterizing regret bounds using characteristics of reward processes. Different from most work in the area, we do not assume that X_t^k are bounded within a certain interval, as such an assumption is too restrictive for general time series analysis.

4.1.3 Contribution

To the best of our knowledge, our work is the first attempt at incorporating time series analysis into bandit problems. We introduce the concept of a temporal oracle process to address the volatility of the environment to create a statistically stationary, yet dynamic, benchmark for analyzing regret. Let σ^k be the stationary standard deviation of $\{X_t^k\}$ and σ_w^t be the white noise standard deviation representing the part of $\{X_t^k\}$ not predictable using the past. We prove a lower bound on instantaneous regret for any policy as $\sqrt{\log K}\sigma_w^{min}$, analogous to a $\sqrt{\log K}\sigma^{min}$ lower bound for the static optimal policy. We propose the concept of temporal exploitation and exploration, referring to the need for exploring arms to learn the current evolution of reward processes and utilizing the predictive power from temporal correlations to gain better immediate reward. We establish a linear improvement in regret compared to the static optimal arm by studying a pure exploitation policy. Then we design a Temporal-Exp2 policy that performs a two-step lookahead computation to balance temporal exploration-exploitation, achieving an upper regret bound of $\sqrt{\log K}(\max_{k\in\mathscr{K}}\{\sigma^k - \gamma_1^k\})$. We develop a decomposition technique that rewrites a time series process into a projection and innovation, conditioning on a revealed past history, to facilitate our analysis. Despite a bit of complication in introducing the notation to fully describe our problem, the theoretical results are intuitive.

4.2 Time Series Models in Bandit Feedback

Using $\{w_t\}$ to denote i.i.d Gaussian white noise with mean zero and standard deviation σ_w, an ARMA(p,q) time series with shift μ is a stochastic process $\{X_t\}$ that can be expressed as

$$X_t + \sum_{i=1}^{p} \phi_i (X_{t-i} - \mu) = \mu + w_t + \sum_{j=1}^{q} \theta_j w_{t-j} \tag{4.5}$$

where $\mu; \phi_i, i \leq p; \theta_j, j \leq q$ are parameters characterizing the temporal correlation structure which we assume to be known throughout this paper. A *moving average* (MA) process refers to the processes that are expressed entirely using a sequence of white noises of the form $X_t = \mu + \sum_{i=-\infty}^{\infty} \psi_i w_{t-i}$, where $\psi_i \in R$ is the ith order moving average coefficients.

Definition 5 (Stationary Process)**.** *A discrete-time stochastic process $\{X_t\}$ is stationary if $(X_t, ..., X_{t+h})$ and $(X_s, ..., X_{s+h})$ are identically distributed $\forall t, s, h \in \mathbb{Z}^+$, where $\mathbb{Z}^+$ is the set of positive integers.*

Definition 6 (Causal ARMA(p,q) Process)**.** *An ARMA(p,q) process is causal if it can be expressed as a moving average process only depending on the past:*

$$X_t - \mu = w_t + \sum_{i=1}^{\infty} \psi_i w_{t-i}. \tag{4.6}$$

Remark: if X_t is not observed at time t, the uncertainty into future values of X_t would increase, as more white noise is included. We assume the reward processes are stationary and causal throughout our paper, as we are focusing on utilizing observed information for enhancing decision making.

Lemma 3 (Temporal Correlation and Stationary Distribution for MA Processes)**.** *For a given stationary moving average process $X_t = \mu + w_t + \sum_{i=1}^{\infty} \psi_i w_{t-1}$:*

1. *The stationary distribution of X_t is normal with mean μ and variance $\sigma_w^2(1 + \sum_{i=1}^{\infty} \psi_i^2)$.*

2. *Define $\gamma(i, j) = Cov(X_i, X_j)$, then $\gamma(i, j) = \sigma_w^2 \sum_{h=1}^{\infty} \psi_h \psi_{|i-j|+h}$.*

In the bandit feedback setting where only the reward on the chosen arm is revealed at each time step, we introduce a few additional definitions and notations to facilitate our expression and computation:

μ^k, σ^k: stationary mean, standard deviation of X_t^k.

σ_w^k: standard deviation of white noise.

$J_t^k := \{i < t : a_i = k\}, \forall k \in \mathscr{K}$: the times arm k is played before time t.

$z_t^k := (x_j^k - \mu^k)_{j \in J_t^k}$: vector of means shifted past realized rewards on arm k.

$\Sigma_t^k := [\gamma(i,j)]_{i,j \in J_t^k}$: covariance matrix of observed rewards on arm k.

$L_t^k := (\gamma(t,i))_{i \in J_t^k}$: covariance between X_t^k and past observed rewards for arm k.

$\mu_t^k := \mathbb{E}[X_t^k | \mathscr{F}_{t-1}], \sigma_t^k := sd(X_t^k | \mathscr{F}_{t-1})$: mean and standard deviation of X_t^k given $\mathscr{F}_{t-1}$.

$O(x)$: Use $y = O(x)$ for denoting $y/x \leq Cx$ for some $C \leq \infty$.

Under the above notation, μ_t^k and σ_t^k can be computed as

$$\mu_t^k = \mu^k + (L_t^k)'(\Sigma_t^k)^{-1} z_t^k \quad , \quad \sigma_t^k = \sigma^k - (L_t^k)'(\Sigma_t^k)^{-1} L_t^k. \tag{4.7}$$

We will be using Equation (4.7) for computing predictions into future time steps, and rely more on Equation (4.6) for proving lower and upper bounds on regret. It is worth mentioning that in Equation (4.7), for independent rewards, we have $L_k^t = (0,..,0)', \forall k,t$, and therefore $\mu_t^k = \mu^k$ and $\sigma_t^k = \sigma^k$.

Lemma 4 (Decomposition Lemma)**.** *Given a set of time indices J_t^k, the random variable X_t^k can be decomposed into two components:*

$$X_t^k = \hat{X}_t^k + \tilde{X}_t^k \tag{4.8}$$

where $\hat{X}_t^k = \mathbb{E}[X_t^k | X_j^k, i \in J_t^k]$ *is the projection of* X_t^k *onto the linear space spanned by* $X_j, j \in J_k^t$*, and* $\tilde{X}_t^k$ *is the residual of the projection.*

1. $\tilde{X}_t^k$ *is normally distributed with mean* 0 *and standard deviation* σ_t^k.

2. $Cov(\hat{X}_t^k, \tilde{X}_t^k) = 0$ *and* $\mathbb{E}[\hat{X}_t^k | \mathscr{F}_{t-1}] = \mu_t^k$.

Theorem 7 (Lower Bound on Instantaneous Regret Given Fixed Arm Selection)**.** *Given a fixed set of observation indices* $J_t^k, \forall k \in \mathscr{K}$*. Then,*

$$\mathbb{E}[r_t] \geq \mathbb{E}[\max_{k \in \mathscr{K}} \{\hat{X}_t^k + \tilde{X}_t^k\}] - \mathbb{E}[\max_{k \in \mathscr{K}} \{\hat{X}_t^k\}]. \tag{4.9}$$

Proof. The key is to properly decompose μ_t^S in Definition 4. Using the definition of r_t to write out expected instantaneous regret and plugging in Lemma 2,

$$\begin{aligned}
\mathbb{E}[r_t] &= \mathbb{E}[\mu_t^S | \mathscr{F}_{t-1}] - \mathbb{E}[X_t(a_t) | \mathscr{F}_{t-1}] \\
&\geq \mathbb{E}[\max_{k \in \mathscr{K}} \{\mathbb{E}[X_t^k | \mathscr{F}_{t-1}] + X_t^k - \mathbb{E}[X_t^k | \mathscr{F}_{t-1}]\}] - \mathbb{E}[\max_{k \in \mathscr{K}} [\mathbb{E}[X_t^k | \mathscr{F}_{t-1}]]] \\
&\geq \mathbb{E}[\max_{k \in \mathscr{K}} \{\hat{X}_t^k + \tilde{X}_t^k\}] - \mathbb{E}[\max_{k \in \mathscr{K}} \{\hat{X}_t^k\}].
\end{aligned}$$

□

Theorem 7 provides a key insight for our analysis: the oracle process $\{S_t\}$ could pick out the max among $\{\hat{X}_t^k + \tilde{X}_t^k\}_{k \in \mathscr{K}}$ with access to $\mathscr{H}_t$, while any policy only observing $\mathscr{F}_{t-1}$ can only pick out the max among $\{\hat{X}_k^t\}_{k \in \mathscr{K}}$. $\tilde{X}_t^k$ and its standard deviation σ_t^k represent the information gap between the oracle and the policy.

As we will be relying heavily on bounding the expected value of the maximum of independent normal random variables, we provide Theorem 8.

Theorem 8 (Lower Bound on Expectation of Maximum of Independent Normal Random Variables)**.** *Let $X_i, 1 \leq i \leq K$ be independent normal random variables with mean μ_i and variance σ_i^2. Use $a \vee b$ and $a \wedge b$ to denote, respectively, the maximum and minimum of a and b. w.l.o.g assume $\mu_1 \geq \mu_2 \geq .. \geq \mu_k$ and let $\Delta_i = \mu_1 - \mu_i$. Then*

$$\mathbb{E}[\max_{1 \leq i \leq K} \{X_i\}] - \mu_1 \geq$$
$$O(\{\sum_{i=1}^{k}((\frac{\sqrt{\sigma_i^2+\sigma_1^2}}{\Delta_i} - \frac{(\sqrt{\sigma_i^2+\sigma_1^2})^3}{\Delta^3}))^{k-1}(\Delta_i+\sigma_i+\sigma_1)\} \vee \{\sqrt{\log K}\sigma^{min}\})$$

Proof. Let $Z = \max_{i \leq k}\{X_i\}$, then by Jensen's Inequality

$$e^{t\mathbb{E}[z]} \leq \mathbb{E}[e^{tZ}] \leq \sum_{i=1}^{k} \mathbb{E}[e^{tX_i}] = ke^{t^2\sigma^2/2}.$$

Taking log on both sides, we can conclude

$$\mathbb{E}[Z] \leq \frac{\log k}{t} + \frac{t\sigma^2}{2} \leq \sigma\sqrt{2\log k}$$

□

It is worth mentioning that in many policies a *spatial* correlation would naturally be induced among $\hat{X}_t^k, \forall k \in \mathscr{K}$, as the information obtained on one arm would depend on the information on all other arms, invalidating Theorem 8. However, $\{\tilde{X}_t^k\}_{k \in \mathbb{K}}$ will still be independent. Theorem 8 is critical in understanding regret lower bounds, which we show later depend crucially on the increment of independent white noises $\{w_t^k\}$.

4.3 Regret Lower Bounds

We study the lower bounds for two types of policies: (1) those that exploit the known correlation in X_t^k, and (2) those that treat the observed rewards as independent samples.

Theorem 9 (Lower Bound for Any Policy)**.** *For any policy* Π, $\mathbb{E}[r_t] \geq O(\sqrt{\log K}\sigma_w^{min} - \Delta^{\mu})$*, where* $\sigma_w^{min} = \min_{k \in \mathscr{K}}\{\sigma_w^k\}, \Delta^{\mu} = \max_{k \in \mathscr{K}}\{\mu^k\} - \min_{k \in \mathscr{K}}\{\mu^k\}$.

Proof. The proof has two main components: first we prove that the instantaneous regret for any policy is greater than an imaginary policy that has access to $\mathscr{H}_{t-1}$ and exploits this information. This is true, as

$$\max\{\mathbb{E}[X_t^k|\mathscr{H}_t]\} > \max\{\mathbb{E}[X_t^k|\mathscr{H}_{t-1}]\}.$$

With $\mathscr{H}_{t-1}$ available, $X_t^k|\mathscr{H}_{t-1}$ is distributed as $\mathscr{N}(\mu_t^k, (\sigma_w^k)^2)$. Let $\mu_t^{min} = \min_{k \in \mathscr{K}}\{\mu_t^k\}$. A lower bound on the instantaneous regret can be obtained using a relaxed lower bound from Theorem 8:

$$\mathbb{E}[r_t] = \mathbb{E}[\mu_t^S - \mathbb{E}[\max_{k \in \mathscr{K}}\{X_t^k|\mathscr{F}_{t-1}\}] \geq \mathbb{E}[\max_{k \in \mathscr{K}}\{X_k^t|\mathscr{H}_{t-1}\}] - \max\{\mathbb{E}[\{X_k^t|\mathscr{H}_{t-1}\}]\}$$
$$\geq O(\mathbb{E}[\sigma_w^{min}\sqrt{\log K} + \mu_{t-1}^{min} - \mu_{t-1}^{max}]) = O(\sqrt{\log K}\sigma_w^{min} - \Delta^{\mu}).$$

For some ARMA processes such as $X_t^k = \phi_{t-k}X_{t-k}^k + w_t$, it is possible to have $X_t^k|\mathscr{F}_{t-1}$

being equivalent to $X_t^k|\mathscr{H}_{t-1}$ in distribution, meaning $\mathscr{F}_{t-1}$ captures all the predictive power of $\mathscr{H}_{t-1}$. In such scenarios, the above lower bound on the instantaneous regret will be tight. $\square$

Theorem 10 (Lower Bound for Policies Assuming Independent Rewards)**.** *For policies minimizing the regret defined in Equation 4.2 ,* $\mathbb{E}[r_t] \geq O(\sqrt{\log K}\sigma^{min})$*, where* $\sigma^{min} = \min_{k\in\mathscr{K}}\{\sigma^k\}$.

Proof. For independent rewards, the optimal policy is to always choose the arm with maximum stationary mean, therefore by Theorem 8,

$$\mathbb{E}[r_t] = \mathbb{E}[\mu_t^S] - \mu^* = \mathbb{E}[\max_{k\in\mathscr{K}}\{X_t^k\}] - \mu^* \geq O(\sqrt{\log K}\sigma^{min} - \Delta^{\mu}).$$

The key is to realize $\mathbb{E}[\mu_t^S] = \mathbb{E}[\max_{k\in\mathscr{K}}\{X_t^k\}]$ in this setting. $\square$

The two tight regret lower bounds in Theorems 9 and 10 only differ by a constant factor σ_w^{min} and σ^{min}, respectively. From Lemma 3, as $\sigma/\sigma_w = 1 + \sum_{i=1}^{\infty}\psi_i^2$, a stronger temporal correlation could mean a significantly improved lower bound. And without temporal correlations ($\phi_i = 0, \forall i \geq 0$), the two regret lower bounds naturally match.

4.4 Temporal Policies

The problem of designing a policy can be formally defined as an optimization problem on choosing the proper allocation configuration $J_T^k, \forall k \in \mathscr{K}$, conditioning on collected information $\mathscr{F}_{t-1}$ such that R_T can be minimized. More intuitively, the optimization requires a policy to balance between the **temporal exploration:** learning about

```
Algorithm 4: Temporal-Exploit
Initialization: For $t = 1, \dots, k$, choose $a_t = t$
while $k+1 \leq t \leq T$ do
    Update $\mu_t^k, \sigma_t^k$, $\forall k \in \mathscr{H}$ with Equation (4.7),
    Choose $a_t = \arg\max\{\mu_k^t\}$,
    Receive reward $X_t(a_t)$ and set $t \leftarrow t+1$
```

future states of each arm by minimizing σ_k^t and **temporal exploitation**: play the arm with maximum μ_t^k to obtain good immediate rewards. We first analyze a temporal exploitation policy to illustrate that a linear improvement on regret over policies assuming independent rewards can be easily obtained. Then we propose a Temporal-Exp2 policy that balances the temporal exploration and exploitation tradeoff by evaluating a two-step look-ahead expected reward computation and prove its regret matches the lower bound up to a K factor.

4.4.1 Temporal Exploitation

The most intuitive approach for exploiting the predictive power that comes with temporal correlations is to use a greedy approach and always choose the arm with maximum predicted reward on the next time step. This *pure exploitation* policy is similar to the greedy (or exploit) policy that always chooses the arm with maximum estimated mean rewards in the usual independent reward setting. The lack of exploration causes the policy to lose track of the current state of the sub-optimal arms, and if a reward process $\{X_t^k\}$ is evolving into a good state, it will not be discovered by the policy soon enough. Despite the simplicity and obvious drawbacks of the greedy policy, we prove that it still achieves a linear improvement in its cumulative reward compared to the optimal policy that assumes independent rewards, highlighting the necessity for exploiting temporal correlations.

Theorem 11 (Linear Improvement on Cumulative Reward)**.** *For the Temporal-Exploit policy, let a_t be the arm chosen by the policy at time t and $x_t(a_t)$ the corresponding received reward. Then,*

$$\mathbb{E}[x_t(a_t)] - \mu^* \geq \Delta^{min} + \sigma_w^{min} e^{-\frac{(\Delta^{max})^2}{2}},$$

where $\Delta_k = \mu^* - \mu^k$ and $\Delta^{min} = \min_{k \in \mathscr{K}}\{\Delta_k\}, \Delta^{max} = \max_{k \in \mathscr{K}}\{\Delta_k\}$.

Proof. From Lemma 4 and Theorem 7, it is easy to see that $\mathbb{E}[x_t(a_t)] = \mathbb{E}[\max_{k \in \mathscr{K}}\{\hat{X}_t^k\}]$. The gap between the cumulative rewards for the Temporal-Exploit Policy and the static policy of choosing $a_t = \arg\max_{k \in \mathscr{K}}\{\mu^k\}$ can be written as:

$$\mathbb{E}[r_t] = \mathbb{E}[\max_{k \in \mathscr{K}}\{\hat{X}_t^k\}] - \mu^*.$$

Though $\hat{X}_t^k, \forall k \in \mathscr{K}$ are correlated, preventing a simple proof using Theorem 8, notice that for this particular policy, given past information $\mathscr{F}_{t-1}$,

$$\mathbb{E}[\max_{k \in \mathscr{K}}\{X_t^k | \mathscr{F}_{t-1}\}] > \mathbb{E}[\max_{k \in \mathscr{K}}\{X_t^k | J_t^k\}],$$

since the policy is greedy. The lower bound on the right side

$$\min_{J_k^t}\{\mathbb{E}[\max_{k \in \mathscr{K}}\{X_t^k | J_t^k\}]\} - \mu^* \text{ with } X_t^k | J_t^k \sim \mathscr{N}(\mu^k, (L_t^k)'(\Sigma_t^k)^2 L_t^k).$$

can be found by replacing $\{\mu^k\}_{k \in \mathscr{K}}$ and $\{(L_t^k)'\Sigma_t^k L_t^k\}_{k \in \mathscr{K}}$ with their respective minimums

μ^{min} and σ_w^{min} and substitute into Theorem 8. □

It is worth mentioning that if the rewards are indeed independent, the temporal-Exploit policy reduces to the optimal policy in the independent setting, as $\hat{X}_t^k \equiv \mu^k$ in this scenario.

4.4.2 Temporal Exp2

The Temporal-Exploit Algorithm 4 established a linear improvement in cumulative reward by exploiting the predictability that comes with temporal correlations in the reward processes. However, we are still interested in designing policies that could achieve comparable performance with the oracle process $\{S_t\}$. We first discuss how an exploration step could potentially improve future collected rewards by bringing new information and propose a Temporal-Exp2 (temporal exploration and exploitation with two-step look ahead) policy based on our analysis.

At time t, conditioning on the available information $\mathscr{F}_{t-1}$, if arm a is chosen, an opportunity cost d of not fully exploiting the current available information would be

$$d^a = \max_{k\in\mathscr{K}}\{\mu_t^k\} - \mu_t^a. \tag{4.10}$$

Let (z_t^a, X_t^a) denote the vector including the most recent new observation X_t^a, which we still treat as a random variable. The extra information would make our prediction on X_{t+1}^a a normal random variable with mean μ_{t+1}^a and standard deviation σ_{t+1}^a, which can be computed using 4.7 to be

$$\mu_{t+1}^{a} = \mu^{k} + (L_{t+1}^{a})'(\Sigma_{t+1}^{a})^{-1}(z_{t}^{a}, X_{t}^{a}), \sigma_{t+1}^{a} = \sigma^{k} - (L_{t+1}^{a})'(\Sigma_{t+1}^{a})^{-1}L_{t+1}^{k}.$$

Using Lemma 4 , we decompose X_{t+1}^{a} conditioning on $\{\mathscr{F}_t, X_t^a\}$ as

$$X_{t+1}^{a} = \hat{X}_{t+1}^{a} + \tilde{X}_{t+1}^{a} = \mu_{t+1}^{a} + \hat{C}^{a} + \tilde{X}_{k+1}^{a}, \hat{C}^{a} \sim \mathscr{N}(0, (c^{a})^{2}) \tag{4.11}$$

where $\hat{C}^a$ is normally distributed with mean 0 and standard deviation $c^a = \sqrt{(\sigma_t^a)^2 - (\sigma_{t+1}^a)^2}$, and is independent of $X_j^a, j \in J_t^k$. Here $\hat{C}^a$ is the residual of $\hat{X}_{t+1}^a$'s projection onto $\mathscr{F}_{t-1}$. Then at time $t+1$, the extra information that is represented by $\hat{C}^a$ will give an enhancement b^a in the expected instantaneous reward that could be obtained compared to using only the information $\mathscr{F}_t$ for arm a:

$$b^{a} = \mathbb{E}[\max_{k \in \mathscr{K}} \{\mu_{t+1}^{k} + I_{\{k=a\}}\hat{C}^{a}\}] - \mathbb{E}[\max_{k \in \mathscr{K}} \{\mu_{t+1}^{k}\}] \tag{4.12}$$

where $I_{\{.\}}$ is the indicator function. The overall expected gain in choosing arm a in the next two steps would be $b^a - d^a$. The Temporal-Exp2 policy would explore an arm if the immediate opportunity cost could potentially be made up by future expected gain.

From Equations (4.10) and (4.12), an arm a would have a higher chance of being explored if (1) its current prediction is close to the current predicted optimal, as it indicates lower immediate opportunity cost and higher expected future gain, and (2) its reward process has strong temporal dependence such that an extra observation would bring in a large amount of information in terms of a larger c^a. Temporal-Exp2 would trivially reduce

Algorithm 5: Temporal-Exp2

```
Initialization: For $1 \leq t \leq k$, choose $a_t = t$.
while $k+1 \leq t \leq T$ do
    Update $\mu_t^k, \sigma_t^k$, $\forall k \in \mathscr{H}$ using Equation (4.7),
    Compute $d^k, b^k$ for all sub-optimal arms ( $k \neq \arg\max_{k \in \mathscr{H}} \{\mu_t^k\}$ ) using
      Equations (4.11 and 4.12)
    Exploration-Exploitation Balancing:
      i. If $\max_k \{b^k - d^k\} < 0$, exploit: choose $a_t = \arg\max_{k \in \mathscr{H}} \{\mu_t^k\}$,
      ii. Otherwise explore: choose $a_t = \arg\max_{k \in \mathscr{H}} \{b^k - d^k\}$,
```

to the static optimal policy if all reward processes are indeed independent, as $b^a \equiv 0$ and $d^k \leq 0, \forall k \in \mathscr{K}$ in this scenario.

Theorem 12 (Regret Upper Bound for Temporal-Exp2 Policy)**.** *For the Temporal-Exp2 policy,*

$$\mathbb{E}[r_t] \leq O(\sqrt{\log k} \max_{k \in \mathscr{K}} \{\sigma^k - \gamma_1^k\}),$$

where $\gamma_1^k = Cov(X_0^1, X_1^k)$.

Proof. Write $r_t = \max_{k \in \mathscr{K}} \{\hat{X}_t^k + \tilde{X}_t^k\} - \hat{X}_t^a = \max_{k \in \mathscr{K}} \{\hat{X}_t^k + \hat{C}_t^k + \tilde{X}_t^k - \hat{C}_t^k\} - \hat{X}_t^a$.

1. If a_t is being exploited, meaning $\hat{X}_t^a = \max\{\hat{X}_t^k\}$ and $\hat{X}_t^k + \hat{C}_t^k < 0$, then $r_t \leq \max_{k \in \mathscr{K}} \{\tilde{X}_t^k - \hat{C}_t^k\}$. As $\hat{C}_t^k$ captures the information of obtaining one new sample, therefore $\tilde{X}_t^k - \hat{C}_t^k$ has mean 0 and variance less than $\sigma^k - r_1^k$. Using results on normal distributions in Theorem 8,

$$\mathbb{E}[r_t] \leq \mathbb{E}[\max_{k \in \mathscr{K}} \{\hat{X}_t^k - \hat{C}_t^k\}] \leq \sqrt{\log K} \max\{\sigma^k - r_1^k\}$$

2. If a_t is being explored, its distance from the truly optimal reward can also be bounded by the fact that it could outperform the current estimated best with an extra observation. Let μ_t^* be the current estimated best predicted mean.

$$\mathbb{E}[r_t] \leq \mathbb{E}\{\max_{k\in\mathscr{K}}\{X_t^k\}\} - \mathbb{E}[\max_{k\in\mathscr{K}}\{b^k - d^k\}]$$
$$\leq \sqrt{\log k}\sigma^k - \sqrt{\log k} r_1^k \leq \sqrt{\log k}\max\{\sigma^k - \gamma_1^k\}.$$

The proof is concluded by relaxing the expression for instantaneous regret quite significantly. □

Comparing with the lower bound in Theorem 9, the temporal-Exp2 policy only considers the effect of having one extra observation, therefore introducing a γ_1^k reduction in the overall regret compared to the static optimal policy.

4.5 Future Research and Concluding Remarks

In our work, we assume that both the overall shift and the temporal correlation of the underlying reward processes are fully known, and we focused on exploiting the known temporal correlations to obtain better cumulative rewards. Not knowing the overall shift in the reward processes will bring us to the scenario in [67], where the authors proved that temporal correlations could lead to a worse performance in regret compared to the static optimal policy, as the estimation of mean rewards could have very high uncertainty. Not knowing the exact form of temporal correlations leads us to the notorious time series model specification and estimation with missing values problem [70,71]. Developing policies for the fully unknown setting to balance between estimating the overall shift, estimating the unknown temporal correlations and exploiting the estimated reward processes would be a challenging yet interesting problem. Another challenge in our work is that the decomposition steps in Equations (4.9), (4.10) and (4.12) involve computing and inverting a potentially large covariance matrix Σ_t^k. How to develop a properly scalable policy for large K and T remains to be solved.

Chapter 5: Bayesian Experimental Design for Stochastic Kriging Metamodels

5.1 Introduction

Kriging originated in the geostatistics community for analyzing data with spatial correlations [1]. Later it was extended for constructing metamodels in the design and analysis of deterministic computer experiments [72]. More recently, the stochastic kriging methodology has extended the kriging estimator to modeling outcomes of stochastic simulations by introducing intrinsic noise, which can be reduced by having more simulation replications at the corresponding design points [23]. Kriging, often referred to as Gaussian process regression in the machine learning community, is also the foundation for Bayesian optimization algorithms, which recently have enjoyed great success in machine learning applications [48, 73].

In the context of simulation metamodeling, stochastic kriging (SK) methods build a global estimate for the unknown function, and a carefully designed experiment is crucial in ensuring the model performance. The common practice is to use static designs such as the uniform design, Latin Hyper-cube Design (LHS), and maximum entropy designs [1]. Dynamic designs should be much more efficient, as more resources could be allocated to regions where the SK model is believed to have poor performance. The works of Ng and Yin [74], Chen and Zhou [75], and Wang and Hu [76] focus on utilizing the posterior uncertainty estimates from the fitted SK model for choosing design points. However, we illustrate in Section 5.2 that the posterior uncertainty estimates in SK models do not properly reflect the roughness of the unknown function and such policies often lead to a near uniform design. Van Beers and Kleijnen [77, 78] developed bootstrap procedures where new data points are sampled from the fitted kriging model for evaluating potential design choices. Our approach differs from theirs, as we perform re-sampling of existing data points rather than from the established SK models, which is more robust to SK model specifications.

In this Chapter, we propose a novel approach for sequential experimental design for SK models. We use a jackknife procedure to obtain an estimation of model prediction error at existing design points, and construct an additional SK model on the error estimates to obtain a landscape for the model performance. Value of information (VoI) measures developed for Bayesian optimization are then used for selecting the next design point. Our approach is robust to the parameter choice of a SK model, as it relies on jackknifing the existing data points for estimating model prediction error rather than on the estimated posterior distributions. To the best of our knowledge, this is the first such attempt to develop a dynamic experimental design procedure for building SK models.

The rest of this Chapter is organized as follows. Section 5.2 briefly reviews the SK model and techniques for its experimental design, and presents an example for motivating our approach. Section 5.3 introduces the jackknife error estimates and a kriging model that is used to search for new design points. Section 5.4 states a sequential design algorithm. Section 5.5 illustrates its performance with two numerical experiments. Finally, we conclude the Chapter in Section 5.6.

5.2 Stochastic Kriging and Its Experimental Design

We first provide a review on the SK models and formulate the dynamic experimental design problem.

5.2.1 Preliminaries on Stochastic Kriging

SK metamodels construct a response surface of an unknown function $y(\mathbf{x}) \in R$ for $\mathbf{x} \in \mathscr{H}$, where $\mathbf{x}$ is a d-dimensional vector and $\mathscr{H}$ is a compact subset of $\mathbb{R}^d$ denoting the domain of interest. We assume $y(\mathbf{x})$ is unknown, but independent samples of its noisy observations can be obtained with simulation. The standard SK model assumes the output of the jth simulation replication at the design point $\mathbf{x}$ can be modeled as

$$y_j(\mathbf{x}) = f(\mathbf{x})^T \beta + M(\mathbf{x}) + \varepsilon_j(\mathbf{x}), \tag{5.1}$$

where $f(\mathbf{x})$ is a feature vector at the point $\mathbf{x}$ and β is a fixed constant vector. The term $f(\mathbf{x})^T \beta$ describes a fixed trend in the unknown target function y, and empirical evidence shows that a constant term performs well in practice [23, 72]. The term $M(\cdot)$ is a second-order stationary zero mean Gaussian process, which models the deviation of the true function y from the fixed trend term $f(\mathbf{x})^T \beta$. The term $\varepsilon_j(\mathbf{x})$, often referred to as intrinsic noise, captures randomness from stochastic simulations [23]. We work with the most basic setting where $\varepsilon_j(\mathbf{x})$ can be considered independent and identically distributed (i.i.d.)

at each design point $\mathbf{x}$ with standard deviation σ^E.

For any two points $\mathbf{x}$ and $\mathbf{y}$ in $\mathscr{H}$, the SK model assumes the covariance between $M(\mathbf{x})$ and $M(\mathbf{y})$ is given by

$$\mathrm{Cov}(M(\mathbf{x}), M(\mathbf{y})) = \tau^2 R(d(\mathbf{x}, \mathbf{y}), \gamma), \tag{5.2}$$

where τ is the variance of the Gaussian process $M(\cdot)$, $d : \mathbb{R}^d \to \mathbb{R}^+$ describes the closeness of points $\mathbf{x}$ and $\mathbf{y}$ in the space $\mathscr{H}$ (e.g., a vector norm), the correlation kernel function R is chosen such that $R(0, \gamma) = 1$ and $\lim_{x\to\infty} R(x, \gamma) = 0, \forall\gamma$, and the parameter γ controls the smoothness of the random field (i.e., the fitted response surface). In practice, both τ and γ are often estimated through maximum likelihood estimation [1].

Given an experimental design configuration $\{\mathbf{x_i}, n_i\}_{i=1}^k$, where $\mathbf{x_i}$ are the design points to perform simulations and n_i is the number of simulation replications at $\mathbf{x}_i$, the samples $\mathscr{Y}_{ij}, i \leq k, j \leq n_i$ are used to compute the sample averages $\bar{\mathscr{Y}} = \{\bar{\mathscr{Y}}_i = \frac{1}{n_i}\sum_{j=1}^{n_i} \mathscr{Y}_{ij}\}_{i=1}^k$ as the input data for fitting an SK model. Denote $\mathbf{F} = [f(\mathbf{x}_1), f(\mathbf{x}_2), .., f(\mathbf{x}_k)]$ the matrix of feature vectors at the existing design points. Following the notation in [23], let Σ_M be the covariance matrix of the random variables $M(\mathbf{x}_i), i \leq k$, Σ_ε be the $k \times k$ diagonal matrix capturing the intrinsic noise under the iid noise assumption for the given experimental design, and $\Sigma_M(\mathbf{x}_0, \{\mathbf{x}_i\}_{i=1}^k)$ be the $k \times 1$ vector representing the correlation between the random field at a potentially new design point $M(\mathbf{x}_0)$ and the existing design points

$\{M(\mathbf{x}_i), i \leq k\}$, we have

$$\Sigma_M = [\text{Cov}(M(\mathbf{x}_i), M(\mathbf{x}_j))]_{i,j\leq k}$$

$$\Sigma_\varepsilon = \text{diag}[\text{Var}(\frac{1}{n_i}\sum_{j=1}^{n_i}\varepsilon_{ij}(\mathbf{x}_i))]_{i\leq k} = \text{diag}[\frac{Var(\varepsilon(\mathbf{x}_i))}{n_i}]_{i\leq k}$$

$$\Sigma_M(\mathbf{x}_0, \{\mathbf{x}_i\}_{i=1}^k) = [\text{Cov}(M(\mathbf{x}_0, M(\mathbf{x}_1))), \text{Cov}(M(\mathbf{x}_0, M(\mathbf{x}_2))), ..., \text{Cov}(M(\mathbf{x}_0, M(\mathbf{x}_k)))]$$

which can be computed by specifying the parameters in Equation (5.2) and σ^E. Assuming β, $\Sigma_M, \Sigma_\varepsilon$ and $\Sigma_M(\mathbf{x}_0, \{\mathbf{x}_i\}_{i=1}^k)$ are known fixed quantities, the best predictor of $y(\mathbf{x}_0)$ that minimizes the mean square error, which we denote as $\hat{y}(\mathbf{x}_0)$, is shown in [23] to be

$$\hat{y}(\mathbf{x}_0) = f(\mathbf{x}_0)^T\beta + \Sigma_M(\mathbf{x}_0, \cdot)^T(\Sigma_M + \Sigma_\varepsilon)^{-1}(\bar{\mathscr{Y}} - \mathbf{F}\beta) \tag{5.3}$$

with the optimal Mean Squared Error (MSE)

$$\text{MSE}(\hat{y}(\mathbf{x}_0)) = \Sigma_M(\mathbf{x}_0, \mathbf{x}_0) - \Sigma_M(\mathbf{x}_0, \{\mathbf{x}_i\}_{i=1}^k)^T(\Sigma_M + \Sigma_\varepsilon)^{-1}\Sigma_M(\mathbf{x}_0, \{\mathbf{x}_i\}_{i=1}^k). \tag{5.4}$$

If the coefficients β are estimated with generalized least squares regression, i.e.,

$$\hat{\beta} = (\mathbf{F}^T(\Sigma_M + \Sigma_\varepsilon)^{-1}\mathbf{F})^{-1}\mathbf{F}^T(\Sigma_M + \Sigma_\varepsilon)^{-1}\bar{\mathscr{Y}},$$

then, the optimal predictor becomes

$$\hat{y}(\mathbf{x}_0) = f(\mathbf{x}_0)^T\hat{\beta} + \Sigma_M(\mathbf{x}_0, \{\mathbf{x}_i\}_{i=1}^k)^T(\Sigma_M + \Sigma_\varepsilon)^{-1}(\bar{\mathscr{Y}} - \mathbf{F}\hat{\beta}) \tag{5.5}$$

with MSE

$$
\begin{aligned}
&\mathrm{MSE}(\hat{y}(\mathbf{x}_0)) = \Sigma_M(\mathbf{x}_0,\mathbf{x}_0) - \Sigma_M(\mathbf{x}_0,\{\mathbf{x}_i\}_{i=1}^k)(\Sigma_M+\Sigma_\varepsilon)^{-1}\Sigma_M(\mathbf{x}_0,\{\mathbf{x}_i\}_{i=1}^k)+ \\
&(f(\mathbf{x}_0)-\mathbf{F}^T(\Sigma_M+\Sigma_\varepsilon)^{-1})\Sigma_M(\mathbf{x}_0,\{\mathbf{x}_i\}_{i=1}^k))^T(\mathbf{F}^T(\Sigma_M+\Sigma_\varepsilon)^{-1}\mathbf{F})^{-1}\cdot \qquad (5.6)\\
&(f(\mathbf{x}_0)-\mathbf{F}^T(\Sigma_M+\Sigma_\varepsilon)^{-1})\Sigma_M(\mathbf{x}_0,\{\mathbf{x}_i\}_{i=1}^k)).
\end{aligned}
$$

In cases where Σ_M and Σ_ε also need to be estimated, the MSE expression becomes intractable. See [23, 72, 79] for reviews of the original kriging methods and its stochastic kriging variation for stochastic simulation experiments. For simplicity, we use θ to represent the hyperparameters for setting up an SK model in Equation (5.3).

5.2.2 Experimental Design For SK

Given a total simulation budget T, experimental design refers to the placement of $\{\mathbf{x}_i, i \leq T\}$ in the design space $\mathscr{H}$ and the corresponding number of replication n_i. In this work, we focus on the search of $\mathbf{x}_i$ and assume $n_i = 1$. Let $D_k = \{(\mathbf{x}_i, y_i),\ 1 \leq i \leq k\}$ denote a set of k observed data points and θ denote the parameters governing the model in Equation 5.1. The performance of the SK estimator $\hat{y}$ fitted on D_k can be evaluated using the integrated mean squared error (IMSE) defined as

$$IMSE = \int_{\mathscr{H}} (\hat{y}(\mathbf{x}) - y(\mathbf{x}))^2 d\mathbf{x}. \tag{5.7}$$

Wang and Hu [76] proved that IMSE will monotonically decrease if more data is inserted to D_k for SK models with known fixed θ. The experimental design problem can be formulated as an optimization problem for minimizing the IMSE with respect to (w.r.t.) the design choice. In both Wang and Hu [76] and Chen and Zhou [75], IMSE is estimated using MSE from fitted SK models. In Section 5.2, we illustrate the limit of such approaches through a motivating example: MSE from fitted Gaussian Processes often fail to capture the observed shape of existing data points, therefore provide little information on the landscape of model performance.

5.2.3 A Motivating Example: Uninformative MSE in SK

Consider the problem of fitting the unknown function $y = sin(3x)e^{-250(x-0.25)^2}$ on the interval $(0,1)$. For simplicity, we assume the observations are noiseless, i.e., $\varepsilon(x) = 0$ w.p. 1. We fit an SK model with a simple uniform design with design points $\{0, 0.1, \ldots, 0.9, 1\}$. The implementation details of the SK model are listed in Section 5.5.1. The same setup is used for generating Figures 5.1,5.2,5.35.4. Despite $y(0.1)$,$y(0.2)$ and $y(0.3)$ having larger jumps in function values compared to $y(0.8)$,$y(0.9)$ and $y(1)$ as illustrated in Figure 5.1, the MSE from the fitted SK model is roughly uniform across the design space $(0,1)$ (see Figure 5.3). The observed roughness in $\{y_i\}_{i=1}^{11}$ is not reflected in the posterior belief of prediction errors in the fitted SK model. The observation is consistent with the formula for MSE in Equation (5.4), where $\bar{\mathscr{Y}}$ does not appear in the expression. For addressing the issue, we use jackknife error estimates to capture the landscape of model prediction performance.

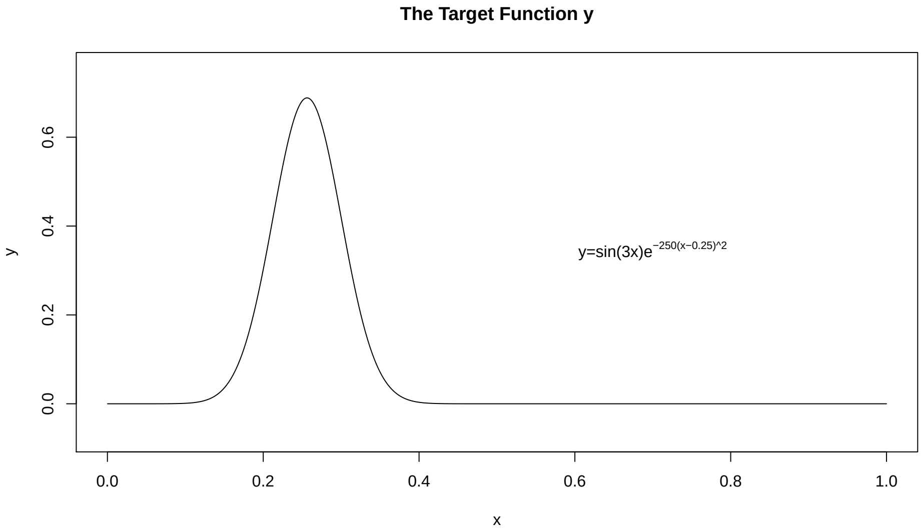

Figure 5.1: The target function $y(\mathbf{x})$ of SK model fitting: $y(\mathbf{x})$ is flat in the region $(0.5, 1.0)$ and has shape changes in $(0.1, 0.5)$, therefore more observations should be drawn in $(0.1, 0.5)$ for obtaining a good fit.

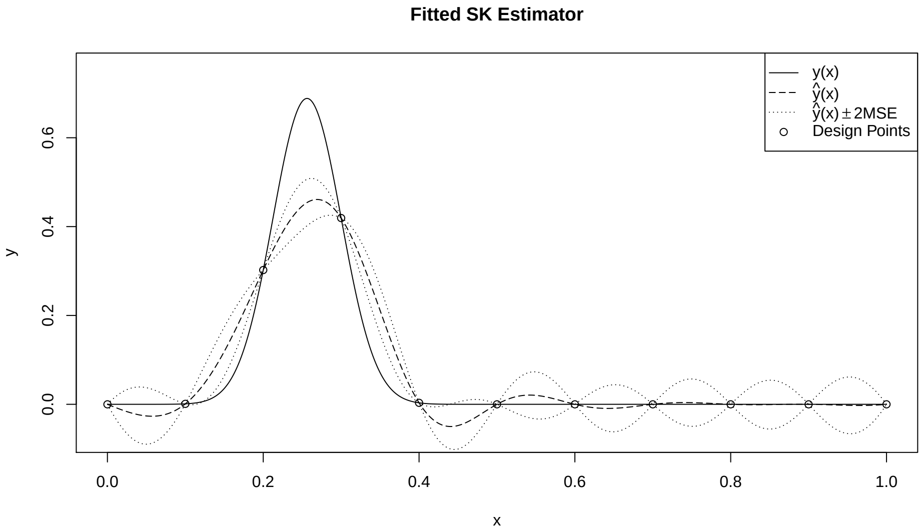

Figure 5.2: The fitted SK prediction under a uniform design. Among the 11 observed function values, $y(0.1), y(0.2), y(0.3), y(0.4)$ showed large jumps and $y(0.5)$ through $y(1.0)$ are roughly constant. The pattern should motivate more samples in the observed rough region.

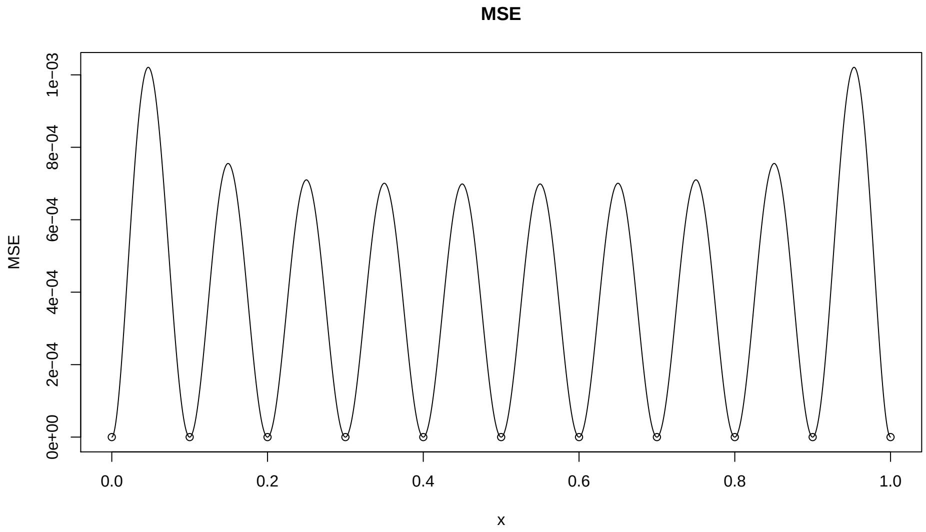

Figure 5.3: The MSE from fitted SK model. The posterior belief of uncertainty is uniform across the design space $(0,1)$. The large jumps in function values in the region $(0.1,0.4)$ are not captured by MSE.

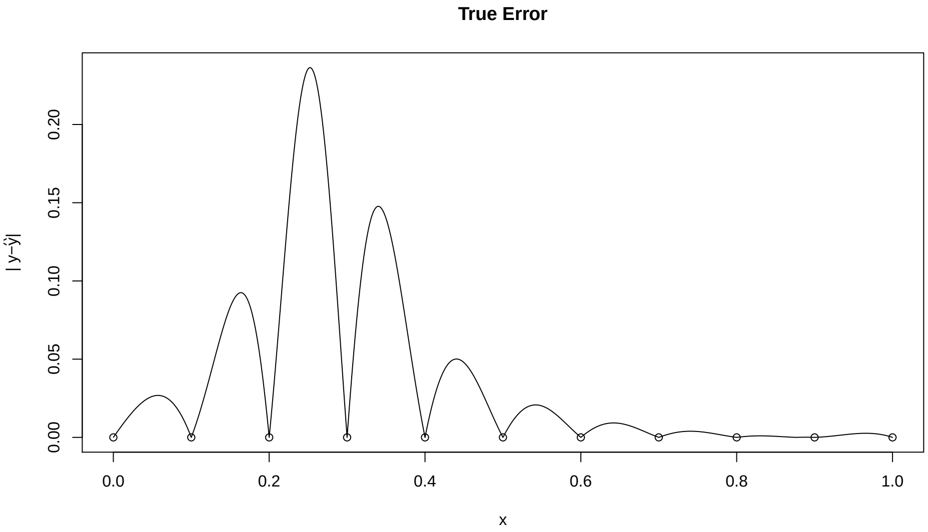

Figure 5.4: The true error for the motivating problem under its fitted SK model. The true error is 0 at the existing design points, as SK models without noise term is an interpolation curve fitting technique. However, the true error landscape still aligns well with the observations: for regions with large jumps in observed function values, the true error is higher in nearby regions.

5.3 Bayesian Experimental Design With Jackknife Error Estimates

Let $D_k = \{(\mathbf{x}_i, y_i)\}_{i=1}^{k}$ denote the current available observation of the unknown functions and T denote the total simulation budget, a dynamic allocation experimental design seeks to allocate the remaining $T-k$ by sequentially selecting the design points. We formulate our approach in this section. The intuition is to place budget on regions where the current model is believed to have larger prediction error either due to insufficient sampling or higher roughness in the underlying target function. We achieve this by constructing an SK model representing our belief of model prediction performance with jackknife error estimates and selecting the design points with Bayesian value of information criteria.

5.3.1 Jackknife Prediction Error Estimates

Jackknife is a resampling technique where one observation is left out from an existing dataset for computing an estimate of an unknown target [80]. The true prediction error of an SK predictor $\hat{y}$ fitted with a dataset D_k at the point $\mathbf{x}$, which we denote as $\delta(\mathbf{x})$, is

$$\delta(\mathbf{x}) = |\hat{y}(\mathbf{x}) - y(\mathbf{x})|. \tag{5.8}$$

Computation of $\delta(x)$ requires evaluating $y(\mathbf{x})$, which is assumed to be expensive in the simulation optimization setting. It is well known that kriging was originally an interpolation curve fitting technique, therefore $|y(\mathbf{x}_i) - y_i| = 0,\ \forall \mathbf{x}_i \in D_k$ and $\delta(\mathbf{x}_i), i \leq k$ do not reflect the prediction performance of $\hat{y}$ [1]. We use a jackknife procedure by leaving $(\mathbf{x}_i, y_i)$ out from D_k and use the remaining $\{y_j\}_{j \neq i}$ for computing an estimate of prediction error for $\hat{y}$.

Let $D_k[-i] = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_{i-1}, y_{i-1}), (\mathbf{x}_{i+1}, y_{i+1}), .., (\mathbf{x}_k, y_k)\}$ denote the data set with $(\mathbf{x}_i, y_i)$ left out from the D_k. An SK model with the same θ as $\hat{y}$ can be fitted on $D_k[-i]$ to obtain a prediction of $y(\mathbf{x}_i)$, which we denote as $\tilde{y}_i(\mathbf{x}_i)$. Then, we define the jackknife error estimate at point x_i as

$$\Delta_i = |\tilde{y}_i(\mathbf{x}_i) - y_i|. \tag{5.9}$$

By using the same θ as $\hat{y}$ for computing $\tilde{y}_i(\mathbf{x}_i)$, Δ_i captures the performance of such SK models under the dataset $D_k[-i]$. There are two issues with such an error estimator: (1) $\tilde{y}(\mathbf{x}_i)$ is estimated based on a smaller sample size, therefore Δ_i generally overestimates the prediction error of $\hat{y}$, and (2) the SK model could be extremely sensitive to the data point $(\mathbf{x_i}, y_i)$ and return drastically different $\tilde{y}_i(\mathbf{x})$ and $\hat{y}(\mathbf{x})$. We argue that despite the above issues, $\{\Delta_i\}_{i=1}^k$ still provide an indication of the SK model performance on $\mathscr{H}$ and could help in searching for the next design point $\mathbf{x}_{k+1}$. As illustrated in Figure 5.5, the $\{\Delta_i\}$ align well with the unknown true prediction error of $\hat{y}$ for our motivating problem.

5.3.2 Modeling the Prediction Error Landscape

With $\{(\mathbf{x}_i, \Delta_i)\}_{i=1}^k$ obtained from the jackknife estimation step, we construct a kriging model to represent our belief about the unknown prediction errors as

$$\Delta(\mathbf{x}) = \mu^\Delta + M^\Delta(\mathbf{x}) + \varepsilon^\Delta(\mathbf{x}), \tag{5.10}$$

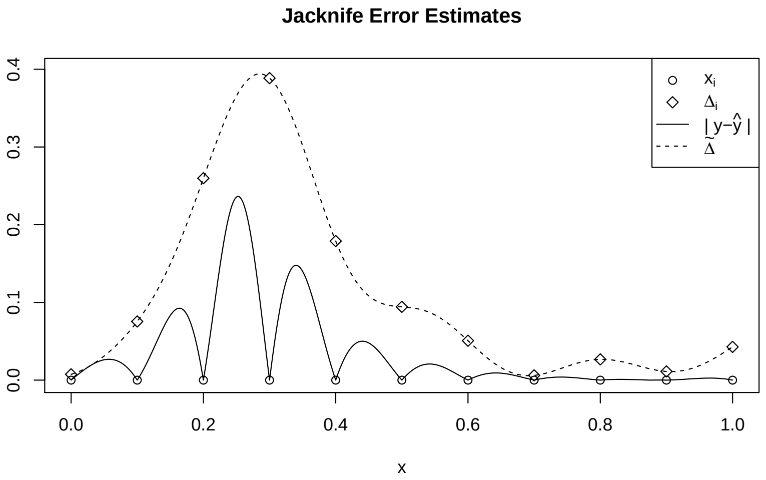

Figure 5.5: The jackknife error estimates $\{\Delta_i\}_{i=1}^{11}$ and the fitted SK model representing the belief on model predictions. The true prediction error is 0 at the existing design points, but the fitted error predictor $\tilde{\Delta}$ captures the region where true error is higher.

where $M^{\Delta}(\cdot)$ and $\tau^{\Delta}R^{\Delta}(\cdot,\theta^{\Delta})$ follow the standard SK model setup outlined in Section 5.2.1. Using similar notation to the SK model in Equations (5.3) and (5.4), given the existing design points and jackknife error estimates $\{(\mathbf{x}_i,\Delta_i)\}_{i=1}^{k}$, the MSE-optimal estimates of $\Delta(\mathbf{x}_0),\mathbf{x}_0\in\mathscr{H}$, for $\mathbf{x}_0\notin\{\mathbf{x}_i\}_{i=1}^{k}$ from the model in (5.10), denoted by $\tilde{\Delta}(\mathbf{x}_0)$, would be

$$\tilde{\Delta}(\mathbf{x}_0)=\mu^{\Delta}+\Sigma_{M^{\Delta}}(\mathbf{x}_0,\{\mathbf{x}_i\}_{i=1}^{k})^T(\Sigma_{M^{\Delta}}+\Sigma_{\varepsilon^{\Delta}})^{-1}(\Delta-\mu^{\Delta}), \tag{5.11}$$

with the MSE

$$\mathrm{MSE}(\tilde{\Delta}(\mathbf{x}_0))=\Sigma_{M^{\Delta}}(\mathbf{x}_0,\mathbf{x}_0)-\Sigma_{M^{\Delta}}(\mathbf{x}_0,\{\mathbf{x}_i\}_{i=1}^{k})^T(\Sigma_{M^{\Delta}}+\Sigma_{\varepsilon^{\Delta}})^{-1}\Sigma_{M^{\Delta}}(\mathbf{x}_0,\{\mathbf{x}_i\}_{i=1}^{k}). \tag{5.12}$$

A superscript of Δ is placed on all covariance matrices and vectors for clarification purpose. $\tilde{\Delta}$ can be viewed as a posterior belief of the upper bounds for model prediction error of $\hat{y}$. When selecting the next design point $\mathbf{x}_{k+1}$, we use a myopic policy of the form

$$\mathbf{x}_{k+1} = \underset{\mathbf{z}\in\mathscr{H},\mathbf{z}\notin\{\mathbf{x}_i\}_{i=1}^{k}}{\arg\max} \; g\left(\tilde{\Delta}(\mathbf{z}), \mathrm{MSE}(\tilde{\Delta}(\mathbf{z}))\right), \tag{5.13}$$

where g is a function for measuring the benefit of drawing an additional sample at $\mathbf{z}$. Following the terminology in the machine learning community, we call g the acquisition function. Instead of the greedy approach of setting the next design point to the maximizer of $\tilde{\Delta}$, we borrow ideas from Bayesian optimization for balancing exploitation (sampling at regions where $\tilde{\Delta}(x)$ is large) and exploration (sampling at regions where $\tilde{\Delta}(x)$ has higher uncertainty). We introduce two common choices of acquisition functions: the probability of improvement (PI) and expected improvement (EI).

5.3.3 Probability of Improvement and Expected Improvement

Let $\Delta^* = \max\{\Delta_i\}_{i=1}^{k}$ denote the current maximum among the jackknife error estimates. Under the model in (5.10), for any $\mathbf{x} \in \mathscr{H}$, $\Delta(\mathbf{x})$ conditioning on $\{\mathbf{x}_i, \Delta_i\}_{i=1}^{k}$ is a Gaussian random variable with mean $\tilde{\Delta}(\mathbf{x})$ and variance $\mathrm{MSE}(\tilde{\Delta}(\mathbf{x}))$ in (5.11) and (5.12), respectively. Let $\Phi(\cdot)$ and $\phi(\cdot)$ be the respective cumulative distribution function and density function for the standard normal distribution and use g^{PI} and g^{EI} for denoting the two acquisition functions.

PI at point $\mathbf{x}$ is defined as $P\{\Delta(\mathbf{x}) \geq \Delta^*\}$, and has the analytical expression

$$g^{PI}(\mathbf{x}) = \Phi\left(\frac{\tilde{\Delta}(\mathbf{x}) - \Delta^*}{\sqrt{\mathrm{MSE}(\tilde{\Delta}(\mathbf{x}))}}\right). \tag{5.14}$$

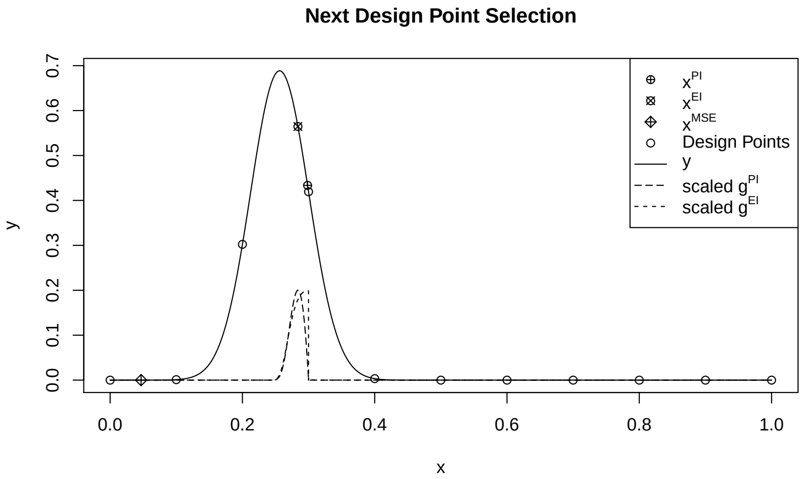

Figure 5.6: The MSE-based selection rule selects the next point to be close to the boundary of $\mathscr{H}$ where the true prediction error is small. *PI* and *EI* place the next point in regions where true error is higher.

EI computes the expected value of improvement $(\tilde{\Delta}(\mathbf{x}) - \Delta^*)^+$ and is given by

$$g^{EI}(\mathbf{x}) = (\Delta^* - \tilde{\Delta}(\mathbf{x}))\Phi\left(\frac{\Delta^* - \tilde{\Delta}(\mathbf{x})}{\sqrt{\mathrm{MSE}(\tilde{\Delta}(\mathbf{x}))}}\right) + \sqrt{\mathrm{MSE}(\tilde{\Delta}(\mathbf{x}))}\phi\left(\frac{\Delta^* - \tilde{\Delta}(\mathbf{x})}{\sqrt{\mathrm{MSE}(\tilde{\Delta}(\mathbf{x}))}}\right), \tag{5.15}$$

where $(x)^+ \equiv \max(x, 0)$ [48].

Both PI and EI are popular choices of acquisition functions for balancing exploration and exploitation trade-off, and are shown to be successful in many stochastic optimization problems such as global optimization (or Bayesian optimization in the machine learning community) [48, 73], multi-armed bandits [81], and ranking and selection [3]. In Figure 5.6, both g^{PI} and g^{EI} select the next design points in regions where the true error is higher for our motivating example, whereas the point with maximum MSE, x^{MSE}, lies in the region where the true error is small.

5.3.4 Practical Model Fitting for Jackknife Error Estimates

In our empirical tests we found $\{\Delta_i\}$ to be extremely non-smooth, even for a smooth underlying target function $y(x)$. For standard SK models, the underlying parameters governing the assumed Gaussian process are often estimated through maximum likelihood estimation [23]. Such an approach tends to lead to a $\tilde{\Delta}$ that overfits to the jackknife error estimates $\{\Delta_i\}$. As the jackknife procedure only provides a noisy indication of model prediction error, we recommend building a smooth model on $\{(\mathbf{x}_i, \Delta_i)\}$ by setting a stronger correlation matrix Σ_{M^Δ} and using noise covariance matrix $\Sigma_{\varepsilon^\Delta}$ with larger diagonal components. In Figures 5.7, 5.8, 5.9 and 5.10, we illustrate the jackknife error estimates for 4 different design choices, represented as xD in the figure, for the motivating problem outlined in Section 5.2.3. The model implementation details are listed in Section 5.5.1. For uniform designs illustrated in Figures 5.7, 5.8 and 5.9, more samples should be allocated to regions around 0.3, as the underlying functions exhibits shape changes in the nearby region, and a uniform design is not efficient by wasting samples on the regions where y is flat. However, overly emphasizing on the neighborhood of 0.3 could also be inefficient, as some samples should still be obtained in other regions. By manually setting a smooth model for the jackknife error estimates, the model $\tilde{\Delta}$ captures the regions that would benefit the most from additional data sample in all the 4 designs. For the three uniform designs in Figures 5.7, 5.8 and 5.9, $\tilde{\Delta}$ captures the need for more samples at regions around 0.3. For the design in Figure 5.10 where budget is allocated around 0.3 and 1, $\tilde{\Delta}$ captures the need for more samples around 0.8. A smooth $\tilde{\Delta}$ is often sufficient for capturing the overall landscape of model performance.

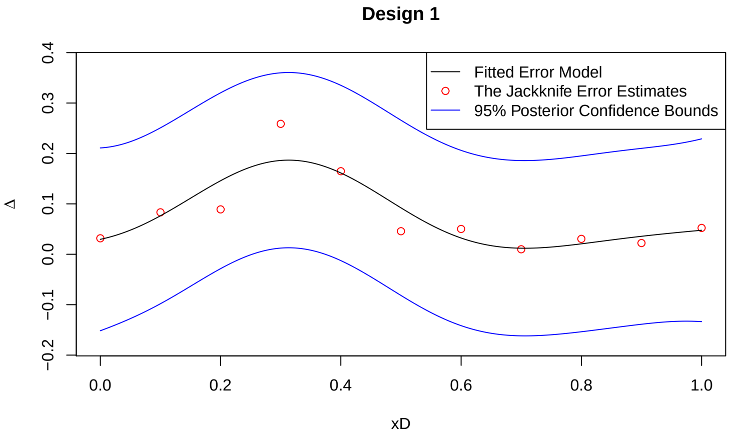

Figure 5.7: A uniform design with 11 design points and fitted $\tilde{\Delta}$.

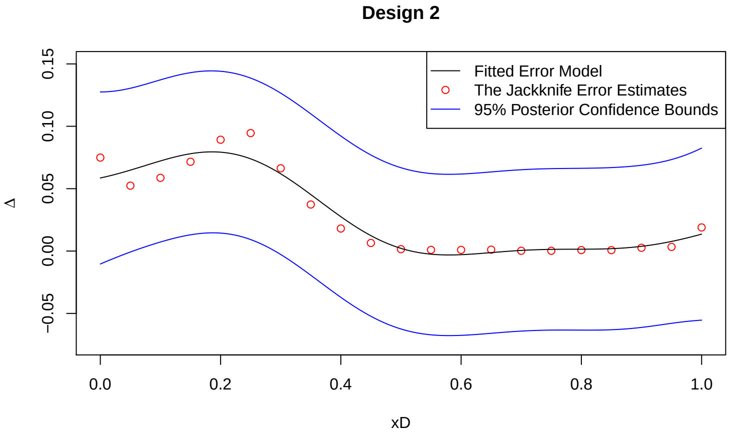

Figure 5.8: A uniform design with 21 design points and fitted $\tilde{\Delta}$.

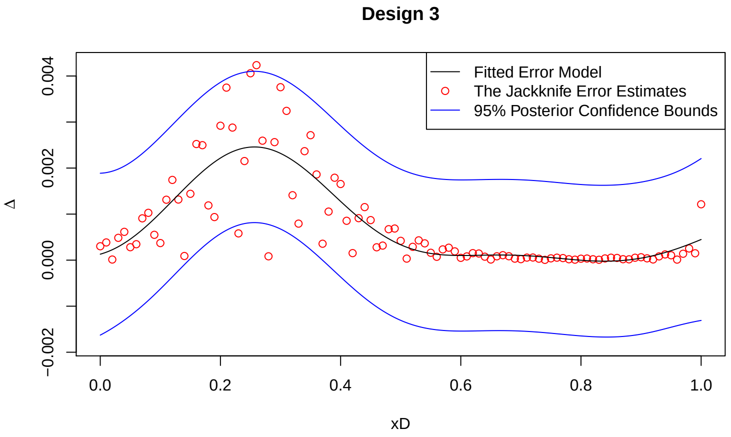

Figure 5.9: A uniform design with 100 design points and fitted $\tilde{\Delta}$.

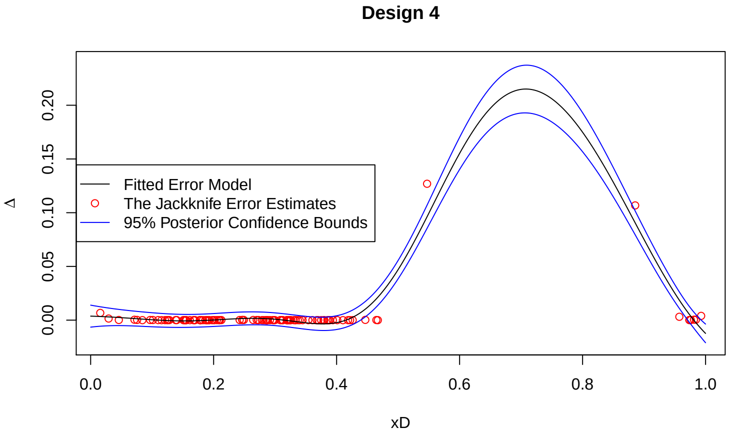

Figure 5.10: 100 design points sampled from $\mathcal{N}(0.3, 0.01)$ and $\mathcal{N}(1, 0.01)$ to focus on the regions near 0.3 and 1.

5.4 The Kriging-based Dynamic Stochastic Kriging (KDSK) Algorithm

We summarize our approach and propose the KDSK algorithm for sequential experimental design for SK models. In Algorithm 6, superscript (t) represents the allocation steps and $D_m^{(t)}$ represents a data set with m data points. At the tth iteration, t SK models will be constructed to obtain the jackknife error estimates, each with $(n_0 + t)^3$ computational complexity. The optimization of the acquisition function could be non-trivial, especially for SK models on a higher dimensional space [48]. However, under the standard assumption that obtaining an output from the underlying target function y is expensive, the computation overhead of KDSK is justified for obtaining a better $\hat{y}$.

Algorithm 6: KDSK

```
Input: Initialization budget $n_0$, initial data $D^{(0)}_{n_0} = \{x_i, \bar{\mathscr{Y}}_i\}_{i=1}^{n_0}$, total remaining
    budget $k$, model parameter choices $\theta$ for $\hat{y}$ and $\theta^{\Delta}$ for $\tilde{\Delta}$, acquisition
    function $g$.
Output: Final experimental design and observed values $\{x_i, \bar{\mathscr{Y}}_i\}_{i=1}^{k}$ and a fitted SK
    model $\hat{y}$
Set $t \leftarrow 1$,
while $t \leq k$ do
    Jackknife $D^{(t)}_{n_0+t}$ to obtain data sets $\left\{D^{(t)}_{n_0+t-1}[-i]\right\}_{i=1}^{n_0+t}$
    Compute the $\{\Delta_i\}_{i=1}^{n_0+t}$ with Equation (5.9) with $\hat{y}$ constructed according to $\theta$
    Fit a SK model on $\{(x_i, \Delta_i)\}_{i=1}^{n_0+t}$ with model specification $\theta^{\Delta}$
    Select $\mathbf{x}_{t+n_0}$ using Equation (5.13)
    Evaluate the unknown function and obtain $\bar{\mathscr{Y}}_{t+n_0}$
    Set $D^{(t+1)}_{n_0+t+1} \leftarrow \{(\mathbf{x}_{t+n_0}, \bar{\mathscr{Y}}_{t+n_0})\} \cup D^{(t)}_{t+n_0}$
    Set $t \leftarrow t+1$
Return $\hat{y}$.
```

5.5 Numerical Experiments

In this section, we illustrate the effectiveness of the proposed KDSK algorithm through two numerical experiments. Let the domain of interest $\mathscr{H}$ be $[0,1]$ and $\hat{y}$ denote the fitted SK model based on an experimental design D, the IMSE in Equation (5.7) is used to evaluate the quality of $\hat{y}$. For comparison purpose, we also implement two naive allocation policies, the UNIFORM policy and min-MSE policy. For UNIFORM allocation, the current total budget is uniformly allocated on $\mathscr{H}$; therefore it is not dynamic and has no initialization overhead. We use it as a benchmark and observe the benefit of having dynamic allocation algorithms. The *min-MSE* methods selects the next point to be the maximizer of posterior variance of the current SK model. In both experiments, the dynamic algorithms are initialized with a uniform design with 11 design points, and IMSE is computed at each allocation step for illustrating the effectiveness of allocation algorithms.

5.5.1 The Motivating Deterministic Function

We test with the target function $y(x) = sin(3x)e^{-250*(x-0.25)^2}$ for $x \in (0,1)$. $\hat{y}$ is constructed in two ways: (1) with fixed and known θ, and (2) with θ estimated through maximum likelihood estimation.

Model choices for $\hat{y}$: For the fixed parameter experiment, the correlation of the underlying Gaussian process is chosen to have the Gaussian kernel $R(d(x,y)) = \tau e^{-(x-y)^2/\sigma}$ with σ set to be 10 and τ equal to 1. The constant trend term μ is set to be 0. The noise covariance matrix is set to be $\Sigma_\varepsilon = diag(0.1)$. When constructing $\hat{y}$ with estimated parameter values, we obtain the fitted SK models with the **GPfit** software package, which estimates θ by maximizing the maximum likelihood with a multi-start gradient based search (L-BFGS-B) algorithm [82]. Note that in the second setting, the error covariance matrix is set to be 0; therefore the fitted SK model interpolates the existing data points.

Model Choices for $\tilde{\Delta}$: The correlation kernel is set to be Gaussian with $\tau^{\Delta} = 1, \sigma^{\Delta} = 1$. The trend term is set to be 0. We also include a noise term with $\Sigma_{\varepsilon^{\Delta}} = diag(0.005)$.

We use EI as the acquisition function. An additional 12 allocation steps are performed, with the numerical results is illustrated in Figures 5.11 and 5.12. For SK models with

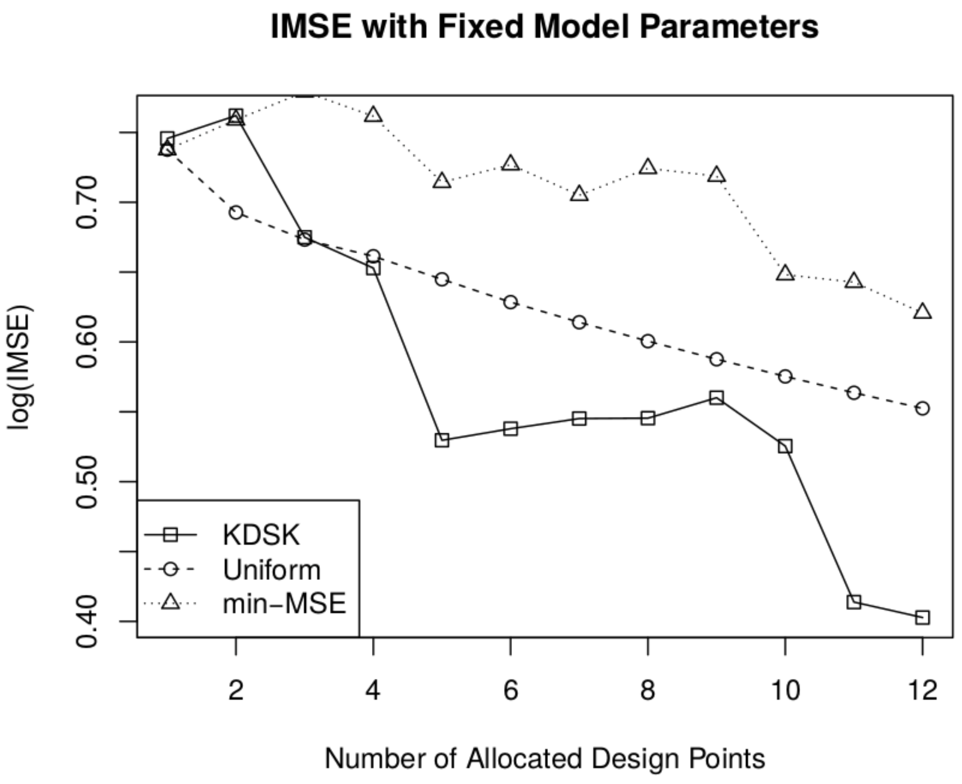

Figure 5.11: $\hat{y}$ constructed with fixed parameters with noise.

known and fixed θ, the IMSE for uniform design steadily decrease as the design sample size grows larger, which is consistent with the findings in Wang and Hu [76]. The

KDSK algorithm outperforms the UNIFORM design after the 3rd allocation step. The min-MSE algorithm has the worst performance among the three, as it generates a near uniform design, but is less efficient compared to the UNIFORM design with its initialization overhead. As $\hat{y}$ is constructed with an SK model with a noise term, the IMSE is on the order of 1 in this experiment.

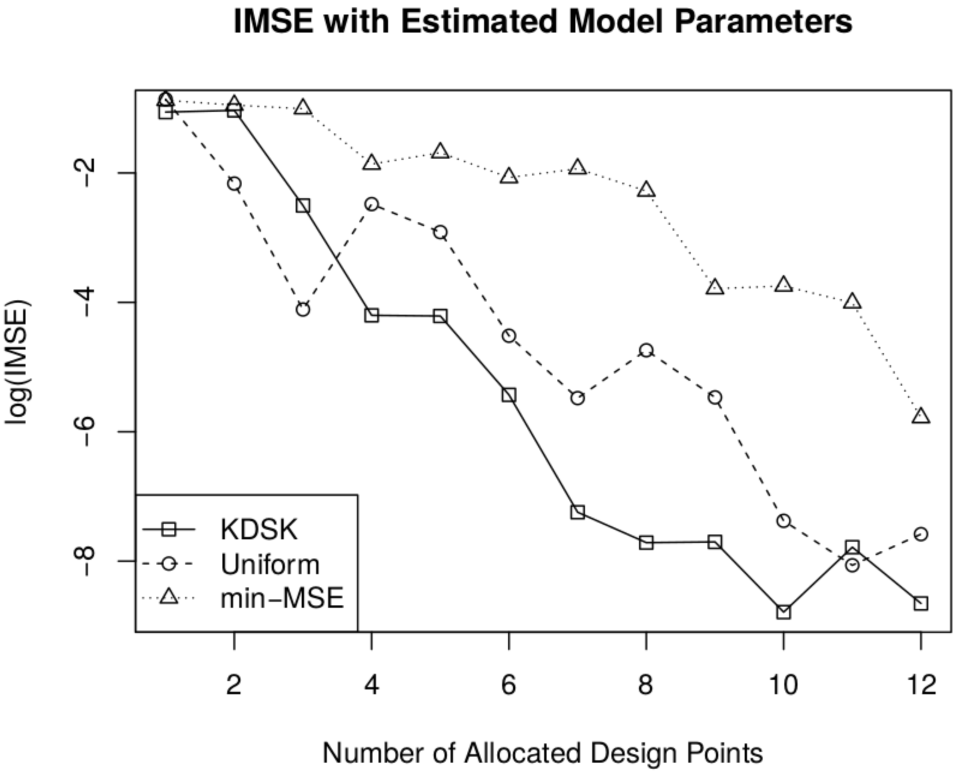

Figure 5.12: $\hat{y}$ constructed with estimated parameters without error term in the model

For the implementation with $\hat{y}$ having θ estimated through maximum likelihood estimation, the IMSE is on the order of 10^{-6} after 12 additional allocation steps. The KDSK algorithm outperforms the other two, with the min-MSE algorithm having the worst performance, The observation is consistent with the fixed parameter experiment.

5.5.2 The Steady-State M/M/1 Queue with Noise

This example is taken from [23], where the objective is to estimate the expected number of customers $y(x)$ in a steady-state M/M/1 queueing system with service rate 1.02 and arrival rate $x \in (0,1)$. For simplicity, we use the known steady-state result $y(x) = \frac{x}{1.02-x} + \varepsilon$, where ε is added zero mean normal noise with standard deviation 0.1. In this experiment, we test the performance of the KDSK algorithm with both the EI and PI activation functions.

Implementation of $\hat{y}$: The correlation function is chosen to be the Gaussian kernel $R(d(x,y)) = \tau e^{-(x-y)^2/\sigma}$ with $\tau = 1$ and $\sigma = 1$, and the constant term is set to be 0. The error covariance matrix is $\Sigma_\varepsilon = diag(0.01)$.

Implementation of $\tilde{\Delta}$: We use the same setup as for $\hat{y}$. And we test with two choices of $\Sigma_{\varepsilon^\Delta}$: $diag(0.05)$ and $diag(0.01)$.

20 dynamic allocation steps are performed, and the IMSE of the fitted $\hat{y}$ at each step is shown in Figures 5.13 and 5.14.

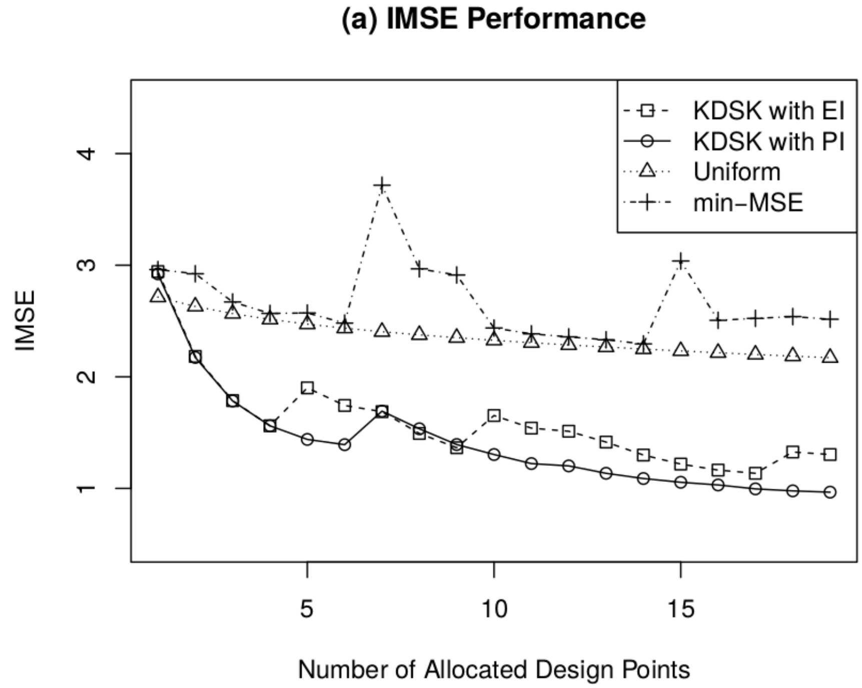

Figure 5.13: $\sigma^E = 0.05$ in the error model G^Δ.

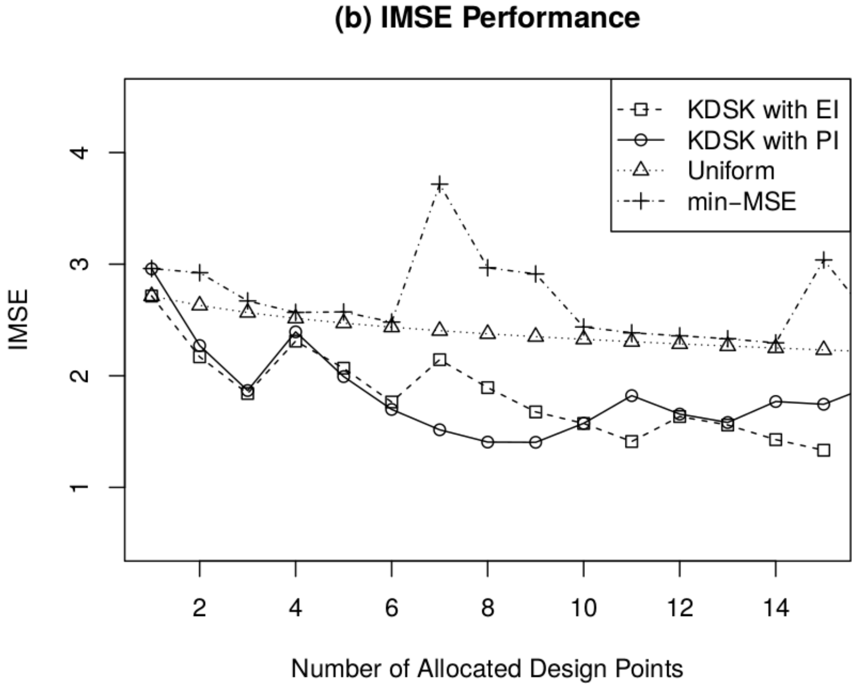

Figure 5.14: $\sigma^E = 0.01$ in the error model G^Δ.

The min-MSE approach has the worst performance. KDSK-PI and KDSK-EI have the best performance with KDSK-PI slightly outperforming KDSK-EI in both tests. We list the design choices of KDSK-PI and KDSK-EI in Table 5.1, with x_t denoting the choice of x at allocation step t. The allocation budget is placed heavily in the region where $y(x)$ has sharp changes.

	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}
KDSK-EI	0.999	0.858	0.866	0.866	0.998	0.758
KDSK-PI	0.994	0.889	0.878	0.999	0.725	0.890

Table 5.1: The Design Choices of KDSK Algorithms

5.6 Conclusion

In this Chapter, we propose a novel approach for the experimental design for the kriging methodology of fitting a global response surface for expensive black box functions. Instead of relying on the posterior error estimates, which is subject to parameter tuning and model choice, we propose the idea of using a jackknife sampling procedure for establishing a landscape of model performance and perform sequential design point selection with Bayesian information criterion. The performance of our approach is illustrated through two numerical experiments. We discussed challenges for implementing the proposed KDSK algorithm, including the smoothness of jackknife error estimates and scaling issue due to the computational complexity. Our approach successfully captures the observed shape of the target function and adjusts the design choices accordingly, therefore is more efficient compared with uniform and MSE-based design methods.

Bibliography

[1] JR Koehler and AB Owen. Chapter 9: Computer experiments. *Handbook of Statistics*, 13:261–308, 1996.

[2] Chun-Hung Chen, Jianwu Lin, Enver Yücesan, and Stephen E Chick. Simulation budget allocation for further enhancing the efficiency of ordinal optimization. *Discrete Event Dynamic Systems*, 10(3):251–270, 2000.

[3] Ilya O Ryzhov. On the convergence rates of expected improvement methods. *Operations Research*, 64(6):1515–1528, 2016.

[4] Herbert Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58(5):527–535, 1952.

[5] Joannes Vermorel and Mehryar Mohri. Multi-armed bandit algorithms and empirical evaluation. In *European Conference on Machine Learning*, pages 437–448. Springer, 2005.

[6] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World Wide Web*, pages 661–670. ACM, 2010.

[7] Peter Auer and Ronald Ortner. UCB revisited: improved regret bounds for the stochastic multi-armed bandit problem. *Periodica Mathematica Hungarica*, 61(1-2):55–65, 2010.

[8] Shipra Agrawal and Navin Goyal. Analysis of Thompson sampling for the multi-armed bandit problem. In *Conference on Learning Theory*, pages 39–1, 2012.

[9] Ronald Ortner, Daniil Ryabko, Peter Auer, and Rémi Munos. Regret bounds for restless Markov bandits. In *International Conference on Algorithmic Learning Theory*, pages 214–228. Springer, 2012.

[10] Sarah Filippi, Olivier Cappe, Aurélien Garivier, and Csaba Szepesvári. Parametric bandits: The generalized linear case. In *NIPS*, pages 586–594, 2010.

[11] Michal Valko, Rémi Munos, Branislav Kveton, and Tomáš Kocák. Spectral bandits for smooth graph functions. In *International Conference on Machine Learning*, pages 46–54, 2014.

[12] Michael C Fu, Chun-Hung Chen, and Leyuan Shi. Some topics for simulation optimization. In *Proceedings of the 2008 Winter Simulation Conference*, pages 27–38. IEEE, 2008.

[13] Chun-Hung Chen and Loo Hay Lee. *Stochastic Simulation Optimization: An Optimal Computing Budget Allocation*, volume 1. World Scientific, 2011.

[14] Seong-Hee Kim and Barry L Nelson. Recent advances in ranking and selection. In *Proceedings of the 2007 Winter Simulation Conference*, pages 162–172. IEEE, 2007.

[15] Peter Frazier, Warren Powell, and Savas Dayanik. The knowledge-gradient policy for correlated normal beliefs. *INFORMS Journal on Computing*, 21(4):599–613, 2009.

[16] Jun Luo, L Jeff Hong, Barry L Nelson, and Yang Wu. Fully sequential procedures for large-scale ranking-and-selection problems in parallel computing environments. *Operations Research*, 63(5):1177–1194, 2015.

[17] Chun-Hung Chen, Donghai He, Michael Fu, and Loo Hay Lee. Efficient simulation budget allocation for selecting an optimal subset. *INFORMS Journal on Computing*, 20(4):579–595, 2008.

[18] Jürgen Branke, Stephen E Chick, and Christian Schmidt. Selecting a selection procedure. *Management Science*, 53(12):1916–1932, 2007.

[19] Michael C Fu, Jian-Qiang Hu, Chun-Hung Chen, and Xiaoping Xiong. Simulation allocation for determining the best design in the presence of correlated sampling. *INFORMS Journal on Computing*, 19(1):101–111, 2007.

[20] Warren B Powell and Ilya O Ryzhov. *Optimal Learning*, volume 841. John Wiley & Sons, 2012.

[21] Diana M Negoescu, Peter I Frazier, and Warren B Powell. The knowledge-gradient algorithm for sequencing experiments in drug discovery. *INFORMS Journal on Computing*, 23(3):346–363, 2011.

[22] Huashuai Qu, Ilya O Ryzhov, Michael C Fu, and Zi Ding. Sequential selection with unknown correlation structures. *Operations Research*, 63(4):931–948, 2015.

[23] Bruce Ankenman, Barry L Nelson, and Jeremy Staum. Stochastic kriging for simulation metamodeling. *Operations Research*, 58(2):371–382, 2010.

[24] Yijie Peng, Chun-Hung Chen, Michael C Fu, and Jian-Qiang Hu. Dynamic sampling allocation and design selection. *INFORMS Journal on Computing*, 28(2):195–208, 2016.

[25] Jean Dickinson Gibbons, Ingram Olkin, and Milton Sobel. *Selecting and Ordering Populations: A New Statistical Methodology*. SIAM, 1999.

[26] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[27] Xiaojin Zhu. Semi-supervised learning literature survey. 2005.

[28] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[29] Rie Johnson and Tong Zhang. On the effectiveness of laplacian normalization for graph semi-supervised learning. *Journal of Machine Learning Research*, 8(Jul):1489–1517, 2007.

[30] Sean Downey, Guowei Sun, and Peter Norquest. Aliner: an R package for optimizing feature-weighted alignments and linguistic distances. *The R Journal*, 9(1):138–152, 2017.

[31] Bradley Efron. *Large-scale Inference: Empirical Bayes Methods for Estimation, Testing, and Prediction*, volume 1. Cambridge University Press, 2012.

[32] Yu-Chi Ho, Qian-Chuan Zhao, and Qing-Shan Jia. *Ordinal Optimization: Soft Optimization for Hard Problems*. Springer Science & Business Media, 2008.

[33] Chun-Hung Chen, Qing-Shan Jia, and Loo-Hay Lee. *Stochastic Simulation Optimization for Discrete Event Systems: Perturbation Analysis, Ordinal Optimization and Beyond*. World Scientific, 2013.

[34] Rasmus Kyng, Anup Rao, Sushant Sachdeva, and Daniel A Spielman. Algorithms for Lipschitz learning on graphs. In *Conference on Learning Theory*, pages 1190–1223, 2015.

[35] Takashi Kawabe, Yuuta Kobayashi, Setsuo Tsuruta, Yoshitaka Sakurai, and Rainer Knauf. Case based human oriented delivery route optimization. In *Evolutionary Computation (CEC), 2015 IEEE Congress on*, pages 2368–2375. IEEE, 2015.

[36] R Tyrrell Rockafellar, Stanislav Uryasev, et al. Optimization of conditional value-at-risk. *Journal of Risk*, 2:21–42, 2000.

[37] Amos Tversky and Daniel Kahneman. Advances in prospect theory: Cumulative representation of uncertainty. *Journal of Risk and Uncertainty*, 5(4):297–323, 1992.

[38] Cheng Jie, LA Prashanth, Michael Fu, Steve Marcus, and Csaba Szepesvári. Stochastic optimization in a cumulative prospect theory framework. *IEEE Transactions on Automatic Control*, 63(9):2867–2882, 2018.

[39] Lidija Trailovic and Lucy Y Pao. Computing budget allocation for efficient ranking and selection of variances with application to target tracking algorithms. *IEEE Transactions on Automatic Control*, 49(1):58–67, 2004.

[40] Yijie Peng, Chun-Hung Chen, Michael C Fu, Jian-Qiang Hu, and Ilya O Ryzhov. Efficient sampling allocation procedures for optimal quantile selection. *Submitted to INFORMS Journal of Computing*, 2019.

[41] Seong-Hee Kim and Barry L Nelson. A fully sequential procedure for indifference-zone selection in simulation. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 11(3):251–273, 2001.

[42] Stephen E Chick. Subjective probability and Bayesian methodology. *Handbooks in Operations Research and Management Science*, 13:225–257, 2006.

[43] Stephen E Chick, Jürgen Branke, and Christian Schmidt. Sequential sampling to myopically maximize the expected value of information. *INFORMS Journal on Computing*, 22(1):71–80, 2010.

[44] Y. Peng and M. C. Fu. Myopic allocation policy with asymptotically optimal sampling rate. *IEEE Transactions on Automatic Control*, 62(4):2041–2047, April 2017.

[45] Averill M Law and W David Kelton. *Simulation Modeling and Analysis*. McGraw-Hill New York, 4 edition, 2007.

[46] Alex Papanicolaou. Taylor approximation and the delta method. 2009.

[47] George Casella and Roger L Berger. *Statistical Inference*, volume 2. Duxbury Pacific Grove, CA, 2002.

[48] Donald R Jones, Matthias Schonlau, and William J Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, 1998.

[49] M. C. Fu, F. W. Glover, and J. April. Simulation optimization: A review, new developments, and applications. In Natalie Steiger et al., editors, *Proceedings of the 2005 Winter Simulation Conference*, Orlando, Florida, Dec 2005. IEEE.

[50] Herbert Robbins. Some aspects of the sequential design of experiments. In *Herbert Robbins Selected Papers*, pages 169–177. Springer, 1985.

[51] Sandeep Pandey, Deepak Agarwal, Deepayan Chakrabarti, and Vanja Josifovski. Bandits for taxonomies: A model-based approach. In *Proceedings of the 2007 SIAM International Conference on Data Mining*, pages 216–227. SIAM, 2007.

[52] Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1):4–22, 1985.

[53] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2-3):235–256, 2002.

[54] Olivier Chapelle and Lihong Li. An empirical evaluation of Thompson sampling. In *Advances in Neural Information Processing Systems*, pages 2249–2257, 2011.

[55] James Douglas Hamilton. *Time series analysis*, volume 2. Princeton University Press, 1994.

[56] Ronald Ortner, Daniil Ryabko, Peter Auer, and Rémi Munos. Regret bounds for restless Markov bandits. In *International Conference on Algorithmic Learning Theory*, pages 214–228. Springer, 2012.

[57] Aleksandrs Slivkins and Eli Upfal. Adapting to a changing environment: The Brownian restless bandits. In *COLT*, pages 343–354, 2008.

[58] Omar Besbes, Yonatan Gur, and Assaf Zeevi. Stochastic multi-armed-bandit problem with non-stationary rewards. In *NIPS*, pages 199–207, 2014.

[59] Agathe Girard, Carl Edward Rasmussen, Joaquin Quinonero Candela, and Roderick Murray-Smith. Gaussian process priors with uncertain inputs with application to multiple-step ahead time series forecasting. In *Advances in Neural Information Processing Systems*, pages 545–552, 2003.

[60] SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional LSTM network: a machine learning approach for precipitation forcasting. In *Advances in Neural Information Processing Systems*, pages 802–810, 2015.

[61] Yevgeny Seldin and Aleksandrs Slivkins. One practical algorithm for both stochastic and adversarial bandits. In *ICML*, pages 1287–1295, 2014.

[62] Chen-Yu Wei, Yi-Te Hong, and Chi-Jen Lu. Tracking the best expert in non-stationary stochastic environments. In *Advances in Neural Information Processing Systems*, pages 3972–3980, 2016.

[63] Ali Jadbabaie, Alexander Rakhlin, Shahin Shahrampour, and Karthik Sridharan. Online optimization: Competing with dynamic comparators. In *Artificial Intelligence and Statistics*, pages 398–406, 2015.

[64] Zohar S Karnin and Oren Anava. Multi-armed bandits: Competing with optimal sequences. In *Advances in Neural Information Processing Systems*, pages 199–207, 2016.

[65] Haya Kaspi and Avi Mandelbaum. Lévy bandits: Multi-armed bandits driven by Lévy processes. *The Annals of Applied Probability*, pages 541–565, 1995.

[66] Nir Levine, Koby Crammer, and Shie Mannor. Rotting bandits. In *Advances in Neural Information Processing Systems*, pages 3077–3086, 2017.

[67] Julien Audiffren and Liva Ralaivola. Cornering stationary and restless mixing bandits with remix-UCB. In *Advances in Neural Information Processing Systems*, pages 3339–3347, 2015.

[68] Christopher R Dance and Tomi Silander. When are Kalman-filter restless bandits indexable? In *Advances in Neural Information Processing Systems*, pages 1711–1719, 2015.

[69] Christopher R Dance and Tomi Silander. Optimal policies for observing time series and related restless bandit problems. *arXiv preprint arXiv:1703.10010*, 2017.

[70] Oren Anava, Elad Hazan, and Assaf Zeevi. Online time series prediction with missing data. In *International Conference on Machine Learning*, pages 2191–2199, 2015.

[71] Richard H Jones. Maximum likelihood fitting of arma models to time series with missing observations. *Technometrics*, 22(3):389–395, 1980.

[72] William J Welch, Tat-Kwan Yu, Sung Mo Kang, and Jerome Sacks. Computer experiments for quality control by parameter design. *Journal of Quality Technology*, 22(1):15–22, 1990.

[73] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical Bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems*, pages 2951–2959, 2012.

[74] Szu Hui Ng and Jun Yin. Bayesian kriging analysis and design for stochastic simulations. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 22(3):17, 2012.

[75] Xi Chen and Qiang Zhou. Sequential experimental designs for stochastic kriging. In *Proceedings of the 2014 Winter Simulation Conference*, pages 3821–3832. IEEE Press, 2014.

[76] Bing Wang and Jiaqiao Hu. Some monotonicity results for stochastic kriging metamodels in sequential settings. *INFORMS Journal on Computing*, 30(2):278–294, 2018.

[77] Wim CM Van Beers and Jack PC Kleijnen. Customized sequential designs for random simulation experiments: Kriging metamodeling and bootstrapping. *European Journal of Operational Research*, 186(3):1099–1113, 2008.

[78] Jack PC Kleijnen and WCM van Beers. Application-driven sequential designs for simulation experiments: Kriging metamodelling. *Journal of the Operational Research Society*, 55(8):876–883, 2004.

[79] Jack PC Kleijnen. Design and analysis of simulation experiments. In *International Workshop on Simulation*, pages 3–22. Springer, 2015.

[80] Bradley Efron and Charles Stein. The jackknife estimate of variance. *The Annals of Statistics*, pages 586–596, 1981.

[81] Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias W Seeger. Information-theoretic regret bounds for Gaussian process optimization in the bandit setting. *IEEE Transactions on Information Theory*, 58(5):3250–3265, 2012.

[82] Blake MacDonald, Pritam Ranjan, and Hugh Chipman. Gpfit: An R package for fitting a Gaussian process model to deterministic simulator outputs. *Journal of Statistical Software*, 64(i12), 2015.