Why Non-myopic Bayesian Optimization is Promising and How Far Should We Look-ahead? A Study via Rollout

Xubo Yue
University of Michigan

Raed Al Kontar
University of Michigan

Abstract

Lookahead, also known as non-myopic, Bayesian optimization (BO) aims to find optimal sampling policies through solving a dynamic programming (DP) formulation that maximizes a long-term reward over a rolling horizon. Though promising, lookahead BO faces the risk of error propagation through its increased dependence on a possibly mis-specified model. In this work we focus on the rollout approximation for solving the intractable DP. We first prove the improving nature of rollout in tackling lookahead BO and provide a sufficient condition for the used heuristic to be rollout improving. We then provide both a theoretical and practical guideline to decide on the rolling horizon stagewise. This guideline is built on quantifying the negative effect of a mis-specified model. To illustrate our idea, we provide case studies on both single and multi-information source BO. Empirical results show the advantageous properties of our method over several myopic and non-myopic BO algorithms.

1 Introduction

Bayesian optimization is a popular technique to optimize an unknown and expensive-to-evaluate objective function through sequential sampling strategies. Traditionally BO has focused on myopic (also referred to as greedy) algorithms, where sampling points are decided based on a one-step lookahead utility function, oblivious how this design will effect the future steps of the optimization and the remaining budget.

Recently, motivated by reinforcement learning, there has been attempts to extend greedy BO methods into multi-step lookahead algorithms that maximize a reward over a rolling horizon. Though it seems promising to look further into the future, this approach might sabotage performance due to accumulated errors and increased dependence on a possibly mis-specified model. This raises the question: is practical implementation of non-myopic approaches indeed useful? Although we cannot give a universal answer, we can shed light on a specific class of non-myopia: rollout dynamic programming.

Rollout is a sub-optimal approximation algorithm to sequentially solve intractable dynamic programming problems. It utilizes problem-dependent heuristics to approximate the future reward using simulations over several future steps (i.e., the rolling horizon). Indeed, rollout has been successfully applied to the non-myopic BO scenario (Lam et al., 2016; Lam and Willcox, 2017). Yet, rollout still faces two challenges: theoretical justification/guarantees and error propagation as errors from a mis-specified model will accumulate as we look further into the future. These challenges trigger researchers to question whether long term planning in BO is necessary.

In this work, we first provide a theoretical justification for the rollout in BO settings. Specifically we show that under the class of sequentially improving heuristics, the rollout is guaranteed to outperform its myopic counterpart. We then provide a guideline to carefully choose a rolling horizon at each stage of the discounted DP. Based on these facts, we argue that even a short horizon is beneficial and also computationally economical. Therefore, using non-myopia is promising and deserves further research attention.

We organize the remaining paper as follows. In Sec. 2, we briefly review BO, DP and rollout. We then prove the performance guarantee of rollout in Sec. 3 and give a practical guideline on choosing the rolling horizon in Sec. 4. In Sec. 5, we provide case studies to evidence our theoretical argument. Detailed literature review can be found in Sec. 6. Some algorithmic details are included in the supplementary material.

2 Background

In this section, we provide the problem description and a brief review on the technical background needed for this paper.

2.1 Bayesian Optimization

Let $f : \boldsymbol{\mathcal{X}} \to \mathbb{R}$ be an objective function which is expensive to evaluate. We consider the optimization problem:

$$\boldsymbol{x}^* = \arg\max_{\boldsymbol{x} \in \boldsymbol{\mathcal{X}}} f(\boldsymbol{x}), \qquad (1)$$

where $\boldsymbol{x}$ is a d-dimensional vector and $\boldsymbol{\mathcal{X}}$ is a compact (closed and bounded) set in $\mathbb{R}^d$. Given limited budget B, BO aims to search for the optimal $\boldsymbol{x}^*$ by iteratively updating a surrogate model of $f(\boldsymbol{x})$, where this surrogate is used to find the next design to evaluate. Typically, in BO, the surrogate model is a Gaussian process ($\mathcal{GP}$), due to its Bayesian interpretation and uncertainty quantification capability (see Rasmussen (2003) for more information).

More specifically, given the current data $\boldsymbol{D}_k$, BO aims to determine the next informative sampling point $\boldsymbol{x}_{k+1}$ by solving the auxiliary problem:

$$\boldsymbol{x}_{k+1} := \boldsymbol{x}'^* = \arg\max_{\boldsymbol{x} \in \boldsymbol{\mathcal{X}}} Q_k(\boldsymbol{x}; \boldsymbol{D}_k). \qquad (2)$$

where Q_k is a acquisition/utility function that only involves evaluating the surrogate and not the expensive objective function f. Typically, evaluation of acquisition function is relatively cheap. The rationale is to seek design points that produce maximum increment of the objective function. After Eq. (2) is solved, the iterative algorithm proceeds by augmenting the current training data $\boldsymbol{D}_k$ with a new observation to obtain $\boldsymbol{D}_{k+1} = \boldsymbol{D}_k \cup \{(\boldsymbol{x}_{k+1}, y_{k+1})\}$. Popular choices of acquisition functions are entropy search (ES) (Hennig and Schuler, 2012), predictive entropy search (PES) (Hernández-Lobato et al., 2014) and expectation improvement (EI) (Lam et al., 2016). All aforementioned functions exploit myopic strategies and ignore the future information.

2.2 Dynamic Programming

Lookahead BO can be directly viewed as an instance of DP. In such settings the non-myopic acquisition function quantifies rewards over future steps. Due to limited budge or sampling capacity, we consider a finite N-stage DP formulation. Denote by $k \in \{1, ..., N\}$. At each stage k, define the state space as $\mathcal{S}_k = (\boldsymbol{\mathcal{X}} \times \mathbb{R})$ and denote by dataset $\boldsymbol{D}_k := s_k \in \mathcal{S}_k$ the current state, where s_k is the state in the state space $\mathcal{S}_k$. A policy $\boldsymbol{\pi} = \{\pi_1, \ldots, \pi_N\}$ is a sequence of rules π_k mapping the state space $\mathcal{S}_k$ to the design space $\boldsymbol{\mathcal{X}}$. We use $\pi_k^{\boldsymbol{\pi}}$ to emphasize the kth rule under policy $\boldsymbol{\pi}$. Let $\pi_k(\boldsymbol{D}_k) = \boldsymbol{x}_{k+1}$. Now denote by $r_k : \mathcal{S}_k \times \boldsymbol{\mathcal{X}} \to \mathbb{R}$ the reward function at stage k. Define the end-stage reward as $r_{N+1} : \mathcal{S}_{N+1} \to \mathbb{R}$. The discounted expected cumulative reward of a finite N-step horizon under policy $\boldsymbol{\pi}$ given initial dataset $\boldsymbol{D}_1$ can be expressed as $R^{\boldsymbol{\pi}}(\boldsymbol{D}_1) =$

$$\mathbb{E}\left[\sum_{k=1}^{N} \alpha^{k-1} r_k(\boldsymbol{D}_k, \boldsymbol{x}_{k+1}) + \alpha^N r_{N+1}(\boldsymbol{D}_{N+1}) \right], \quad (3)$$

where $\alpha \in [0, 1]$ is the discount factor. The discount factor plays an important role in this setting, as it controls the effect of error propagation. *In the greedy algorithm, we have $\alpha = 0$.* In the policy space $\boldsymbol{\Pi}$, we are interested in the optimal policy $\boldsymbol{\pi}^* \in \boldsymbol{\Pi}$ which maximizes Eq. (3). Specifically,

$$R^{\boldsymbol{\pi}^*}(\boldsymbol{D}_1) := \max_{\boldsymbol{\pi} \in \boldsymbol{\Pi}} R^{\boldsymbol{\pi}}(\boldsymbol{D}_1). \qquad (4)$$

Based on the Bellman optimality equation, we can then formulate (3) and (4) as a recursive DP:

$$R_k(\boldsymbol{D}_k) = \max_{\boldsymbol{x}_{k+1} \in \boldsymbol{\mathcal{X}}} \mathbb{E}[r_k(\boldsymbol{D}_k, \boldsymbol{x}_{k+1}) + \alpha R_{k+1}(\boldsymbol{D}_{k+1})],$$
$$R_{N+1}(\boldsymbol{D}_{N+1}) = r_{N+1}(\boldsymbol{D}_{N+1}). \qquad (5)$$

2.3 Rollout

The DP formulation in Sec. 2.2 is subject to a huge computational burden and curse of dimensionality due to the uncountable state and action space. Furthermore, the formulation assumes that data in the last step is available and computes the acquisition function in a backward manner, which is impractical in the setting of BO. In order to solve the intractable DP, an approximate dynamic programming (ADP) approach - rollout (Bertsekas, 1995) has been proposed. The rollout has recently enjoyed success across a variety of domains as it builds on several heuristic policies and is efficient for large-scale and finite-horizon DP problems. More specifically, in the DP formulation, each reward-to-go function R_{k+1} is approximated by H_{k+1} using a heuristic (base) policy. Denote by $h(h \geq 1)$ the rolling horizon and $\tilde{N} = \min\{k + h, N\}$. The rolling horizon alleviates the high dimensionality by limiting the number of stages through which the approximate reward is calculated. Thus, h can be viewed as a non-negative random variable and a stopping time with respect to the family $(\sigma(s_k))_{k \in \mathbb{Z}^+}$ of σ-fields (Sethi and Sorger, 1991).

Given a specific heuristic, we can obtain the approximated reward-to-go functions:

$$H_k(\boldsymbol{D}_k) = \mathbb{E}[r_k(\boldsymbol{D}_k, \tilde{\pi}_k) + \alpha H_{k+1}(\boldsymbol{D}_{k+1})],$$
$$H_{\tilde{N}}(\boldsymbol{D}_{N+1}) = r_{\tilde{N}}(\boldsymbol{D}_{N+1}), \qquad (6)$$

where $\tilde{\pi}_k$ is the heuristic policy at every iteration $k \in [\tilde{N}] = \{1, \dots, \tilde{N}\}$. At the end stage, we define policy $\tilde{\pi}_{\tilde{N}+1}$ such that $\boldsymbol{x}^* = \arg\max_{\boldsymbol{x} \in \mathcal{X}} \mu_0^{\tilde{N}}(\boldsymbol{x})$, where $\mu_0^{\tilde{N}}$ is the updated mean function from data $\boldsymbol{D}_{\tilde{N}}$.

3 Rollout Performance Guarantees

Without loss of generality and for the sake of neatness, we omit the discount factor and assume $\alpha = 1$. Given a state s, an algorithm $\mathcal{H}(s)$ is a method to select a sequence of feasible rules $\{\pi_k\}_{k=1}^N$ and policy $\boldsymbol{\pi}_{\mathcal{H}(s)}$ which generates states $\{s_k\}_{k=1}^N$. Now, to establish theoretical guarantees, we first provide the following definitions (Bertsekas et al., 1997; Goodson et al., 2017).

Definition 1. *Consider a maximization problem. The algorithm $\mathcal{H}$ is said to be sequentially consistent if for every state $s_k \neq s_N$, whenever $\mathcal{H}$ generates the state path $(s_k, s_{k+1} \dots, s_N)$ starting at state s_k, $\mathcal{H}$ also generates the path $(s_{k+1}, \dots, s_N)$ starting at state s_{k+1}.*

In the context of DP, let $s \in \mathcal{S}$ and let s' be a state on a path generated by policy $\boldsymbol{\pi}$ using algorithm $\mathcal{H}(s)$. Denote this policy as $\boldsymbol{\pi}_{\mathcal{H}(s)}$. Consequently, sequential consistency can equivalently be defined as

Definition 2. *Consider a maximization problem. The algorithm $\mathcal{H}$ is said to be sequentially consistent if $\forall s$ and subsequent s', we have*

$$(\pi_k^{\boldsymbol{\pi}_{\mathcal{H}(s)}}, \pi_{k+1}^{\boldsymbol{\pi}_{\mathcal{H}(s)}}, \dots, \pi_N^{\boldsymbol{\pi}_{\mathcal{H}(s)}}) = (\pi_k^{\boldsymbol{\pi}_{\mathcal{H}(s')}}, \pi_{k+1}^{\boldsymbol{\pi}_{\mathcal{H}(s')}}, \dots, \pi_N^{\boldsymbol{\pi}_{\mathcal{H}(s')}}). \quad (7)$$

Definitions 1 implies that a heuristic algorithm is sequentially consistent if it produces the same subsequent states when started at any intermediate state of a path that it generates. Equivalently, by **Definition 2**, the algorithm will generate the same subsequent rules $(\pi_k, \dots, \pi_N)$.

Now, consider a probability space $(\Omega, \mathcal{F}, P)$. Define $\sigma(s)$ as the sub σ-algebra generated by the state s, then we have that

Definition 3. *Consider a maximization problem. The algorithm $\mathcal{H}$ is said to be sequentially improving if for every state $s_k \neq s_N$, whenever $\mathcal{H}$ generates the path $(s_k, s_{k+1}, \dots, s_N)$ starting at state s_k, the following property will hold*

$$\mathbb{E}\Big[\sum_{\ell=k}^{N} r_\ell(s_\ell, \pi_\ell^{\boldsymbol{\pi}_{\mathcal{H}(s)}}(s_\ell)) | \sigma(s')\Big]$$
$$\leq \mathbb{E}\Big[\sum_{\ell=k}^{N} r_\ell(s_\ell, \pi_\ell^{\boldsymbol{\pi}_{\mathcal{H}(s')}}(s_\ell)) | \sigma(s')\Big]. \quad (8)$$

It directly follows that if $\mathcal{H}$ is sequentially consistent, then the equality will hold in (8). Therefore, a sequentially consistent algorithm is also sequentially improving. However, the converse is not true. Next, We will present our theorem about *rollout improving*.

Theorem 1. *The algorithm $\mathcal{H}$ that uses a sequentially improving heuristic is rollout improving. Formally, given the rollout policy $\boldsymbol{\pi}$, we have the following property*

$$\mathbb{E}\Big[\sum_{\ell=k}^{N} r_\ell(s_\ell, \pi_\ell^{\boldsymbol{\pi}_{\mathcal{H}(s_k)}}(s_\ell))\Big] \leq \mathbb{E}\Big[\sum_{\ell=k}^{N} r_\ell(s_\ell, \pi_\ell^{\boldsymbol{\pi}}(s_\ell))\Big]. \quad (9)$$

Proof. We will prove this theorem by mathematical induction. When $\ell = N$, this statement is trivial. Now assume this statement holds for $\ell = k+1, \dots, N-1$. Then, when $\ell = k$, define $\sigma(s_k)$ as the sub σ-algebra generated by state s_k. Since each subsequent state s_{k+1} is an augmented s_k, we have $\sigma(s_k) \subseteq \sigma(s_{k+1}) \subseteq \mathcal{F}$. By the law of total expectation, we have

$$\mathbb{E}\Big[\sum_{\ell=k}^{N} r_\ell(s_\ell, \pi_\ell^{\boldsymbol{\pi}_{\mathcal{H}(s_k)}}(s_\ell)) | \sigma(s_k)\Big]$$
$$= \mathbb{E}\Big[\mathbb{E}\Big[\sum_{\ell=k}^{N} r_\ell(s_\ell, \pi_\ell^{\boldsymbol{\pi}_{\mathcal{H}(s_k)}}(s_\ell)) | \sigma(s_{k+1})\Big] \Big| \sigma(s_k)\Big]. \quad (10)$$

By assumption, since the heuristic is sequentially improving, we have

$$\mathbb{E}\Big[\mathbb{E}\Big[\sum_{\ell=k}^{N} r_\ell(s_\ell, \pi_\ell^{\boldsymbol{\pi}_{\mathcal{H}(s_k)}}(s_\ell)) | \sigma(s_{k+1})\Big] \Big| \sigma(s_k)\Big]$$
$$= \mathbb{E}\Big[\mathbb{E}\Big[\sum_{\ell=k}^{N} r_\ell(s_\ell, \pi_\ell^{\boldsymbol{\pi}_{\mathcal{H}(s_{k+1})}}(s_\ell)) | \sigma(s_{k+1})\Big] \Big| \sigma(s_k)\Big]$$
$$= \mathbb{E}\Big[r_k(s_k, \pi_k^{\boldsymbol{\pi}_{\mathcal{H}(s_k)}}(s_k)) +$$
$$\mathbb{E}\Big[\sum_{\ell=k+1}^{N} r_\ell(s_\ell, \pi_\ell^{\boldsymbol{\pi}_{\mathcal{H}(s_{k+1})}}(s_\ell)) | \sigma(s_{k+1})\Big] \Big| \sigma(s_k)\Big]$$
$$\leq \max_{\pi} \mathbb{E}\Big[r_k(s_k, \pi) +$$
$$\mathbb{E}\Big[\sum_{\ell=k+1}^{N} r_\ell(s_\ell, \pi_\ell^{\boldsymbol{\pi}_{\mathcal{H}(s_{k+1})}}(s_\ell)) | \sigma(s_{k+1})\Big] \Big| \sigma(s_k)\Big]$$
$$= \mathbb{E}\Big[r_k(s_k, \pi_k^{\boldsymbol{\pi}}(s_k)) +$$
$$\mathbb{E}\Big[\sum_{\ell=k+1}^{N} r_\ell(s_\ell, \pi_\ell^{\boldsymbol{\pi}_{\mathcal{H}(s_{k+1})}}(s_\ell)) | \sigma(s_{k+1})\Big] \Big| \sigma(s_k)\Big]. \quad (11)$$

The last equality follows the definition of rollout algorithm. The rest of the proof is completed by the

induction hypothesis. □

Theorem 1 shows that the rollout approach is guaranteed to perform better than its myopic counterpart under the same base heuristic. Intuitively, when rollout generates a path, it exploits the base heuristic to generate a collection of other paths and picks up the best one. In the next section, we will provide a guideline on choosing a sequence of feasible rolling horizons.

4 Deciding on the Rolling Horizon

One interesting question remains: how to decide the rolling horizon h? In most of the literature, h is chosen to be a fixed value within 2 and 5 (Lam and Willcox, 2017; Ulmer et al., 2018) in order to alleviate computational burden. Though those choices give very promising results, those decisions are very subjective. Fortunately, based on the rollout theory (Bertsekas et al., 1997; Bertsekas, 2005), we can provide a practical guideline to select a stagewise feasible h. The big picture is as follows: we quantitatively obtain the benefits of rollout given a modeling error and discount factor, we then compare this long-term discounted benefit with the reward from the greedy algorithm counterpart and decide a feasible rolling horizon accordingly. We provide a detailed argument below.

At each stage k, define a profit function $g_k : \mathbb{Z}^+ \to \mathbb{R}^+$ related to the rolling horizon h such that

$$g_k(h) = \sum_{i=k+1}^{k+h} \phi(i-k), \qquad (12)$$

where $\phi(\cdot)$ is a non-negative function. The rolling profit function can be viewed as the total benefits incurred when choosing a rolling horizon h at stage k. Although long horizons provide more future information, it is not guaranteed to be helpful. In practice we are running the risk of model mis-specification due to modeling the objective function using a $\mathcal{GP}$ and then using this surrogate to simulate scenarios over future steps. Therefore, a larger rolling horizon implies an increased dependence on a possibly erroneous model which might in turn cause adverse effects compared to myopic algorithms where errors accumulate only from a one-step lookahead. However, if we can arbitrarily quantify the error from mis-specified model, then we can utilize the rollout improving nature and accordingly decide on the feasible rolling horizon.

In order to quantify the aforementioned error, we define an error function $\mathcal{E}(\boldsymbol{x})$ bounded by a constant $\bar{e}_k$. The $\mathcal{E}(\cdot)$ is a metric to quantify the negative effect from model mis-specification. In the next section, we will provide an error bound on $\mathcal{GP}$ prediction and use this error bound as an error.

4.1 Error Bound on the GP

The recent work of Wang et al. (2019) sheds light on the model mis-specification issue.

Corollary 1. *(Wang et al., 2019) Assume a $\mathcal{GP}$ with zero mean and stationary convariance function. Then, under some regularity conditions, the interpolation error is (non-asymptotically)*

$$\sup_{\boldsymbol{x}} |y(\boldsymbol{x}) - \hat{f}(\boldsymbol{x})| \leq K\sigma^2 P_X \sqrt{\log(\frac{e}{P_X})} + u,$$

with probability $1 - \delta$, where δ is a function of u, P_X, σ^2, $y(\boldsymbol{x})$ is the true output at input $\boldsymbol{x}$ and $P_X = \sqrt{1 - K(\boldsymbol{x}^*, \boldsymbol{x})K(\boldsymbol{x}, \boldsymbol{x})^{-1}K(\boldsymbol{x}, \boldsymbol{x}^*)}$ is a power function with mis-specified covariance function at observation $\boldsymbol{x}$, K and u are some constants and σ^2 is the variance parameter.

Given this result, at each stage k, we can define $\mathcal{E}(\boldsymbol{x}) = |y(\boldsymbol{x}) - \hat{f}(\boldsymbol{x})| \leq \sup_{\boldsymbol{x}} |y(\boldsymbol{x}) - \hat{f}(\boldsymbol{x})| = \bar{e}_k$. One regularity condition in **Corollary 1** is that the mis-specified kernel is no smoother than the true kernel. The matérn kernel is one of the perfect candidates to this requirement. **Corollary 1** also requires P_X is small. Fortunately, this quantity can be controlled (Wendland, 2004) using fill distance design (Santner et al., 2003). To satisfy this condition, we can simply choose initial data to minimize fill distance. This procedure is known as the minimax distance design (Johnson et al., 1990). In the next section we use this error function to find feasible h stagewise. We note that the proposed framework can be substituted with a different error function, however we focus on $\mathcal{GP}$s as they are the most commonly used surrogate in BO.

4.2 Deciding Rolling Horizon

In this section, we provide our main theorem on deciding h stagewise. Define a function

$$A_k(h, s_k) = \max \left\{ \max_{\boldsymbol{x}_{k+1}} \mathbb{E}\big[r_{k+1}(s_{k+1}, \pi_{k+1}^{\boldsymbol{\pi}_{\mathcal{H}(s_k)}}(s_{k+1}))\big], \right.$$
$$\left. \max_{\boldsymbol{x}_{k+1}} \mathbb{E}\big[g_k(1) + \alpha \tilde{H}(s_{k+1})\big] \right\}, \qquad (13)$$

where $\tilde{H}(s_{k+1}) = g_{k+1}(2) - g_{k+1}(1) - \mathcal{E}(\boldsymbol{x}_{k+2}) + \alpha \tilde{H}(s_{k+2})$ is a modified rollout reward function to quantify both profit and error effects. Note that we do not consider $\mathcal{E}(\boldsymbol{x}_{k+1})$ since it is shared by both algorithms when the rolling horizon is 1. A_k denotes the optimal reward from stage k to N, given the current state s_k and an unknown rolling horizon h. Eq. (13) returns the maximum element between two values: the first one is the reward when we consider a greedy algorithm (i.e., $h = 1$) and the second one is the reward

when we consider the rollout algorithm given a certain error function. Based on Eq. (13), we can obtain the following theorem.

Theorem 2. *The set of feasible rolling horizons at stage k is defined as*

$$h^* = \left\{ 2 \leq h \leq N | A_k(h, s_k) \right.$$
$$\left. = \max_{\boldsymbol{x}_{k+1}} \mathbb{E}\big[g_k(1) + \alpha \tilde{H}(s_{k+1})\big] \right\}. \tag{14}$$

Theorem 2 implies that any h within this set the rollout is more beneficial than a greedy algorithm. In other words, the benefits gained from looking further ahead outweigh that of the error effects. However, calculating h^* is hard to implement in practice. In the next theorem, we will provide an equivalent but more practical equation.

Theorem 3. ***The Rolling Horizon Theorem*** *Given a constant $\bar{e}_k$ on the error function $\mathcal{E}(\boldsymbol{x})$ and the profit function defined in Eq. (12). The feasible rolling horizon at stage k is defined as*

$$h^* = \left\{ j \in \mathbb{Z} : \sum_{i=2}^{j} \alpha^{i-2}\phi(i) > \bar{e}_k \frac{1 - \alpha^{N-k}}{1 - \alpha} \right\}.$$

Proof. Based on **Theorem 2** and Eq. (13), it is equivalent to consider

$$\max_{\boldsymbol{x}_{k+1}} \mathbb{E}\big[g_k(1) + \alpha\tilde{H}(s_{k+1})\big] \geq$$
$$\max_{\boldsymbol{x}_{k+1}} \mathbb{E}\big[r_{k+1}(s_{k+1}, \pi_{k+1}^{\boldsymbol{\pi}_{\mathcal{H}(s_k)}}(s_{k+1}))\big]. \tag{15}$$

By definition, $\max_{\boldsymbol{x}_{k+1}} \mathbb{E}\big[r_{k+1}(s_{k+1}, \pi_{k+1}^{\boldsymbol{\pi}_{\mathcal{H}(s_k)}}(s_{k+1}))\big] \geq g_k(1)$. Since $\tilde{H}(s_{k+1}) \geq -\sum_{i=k+2}^{N+1} \alpha^{i-k-2}\mathcal{E}(\boldsymbol{x}_i) \geq -\bar{e}_k\frac{1-\alpha^{N-k-1}}{1-\alpha}$, we have $\tilde{H}(s_{k+1}) = g_{k+1}(2) - g_{k+1}(1) - \mathcal{E}(\boldsymbol{x}_{k+2}) + \alpha\tilde{H}(s_{k+2}) \geq \phi(2) - \bar{e}_k - \bar{e}_k\frac{1-\alpha^{N-k-1}}{1-\alpha}\alpha$. Therefore, Eq. (15) can be simplified as $\phi(2) \geq \frac{1-\alpha^{N-k}}{1-\alpha}\bar{e}_k$. The remaining part can be obtained by induction. □

In practice, we can pick up the minimal j from the set h^*. We can also set an upper bound on the rolling horizon. Denote by it $\bar{h}$. In the **Theorem 3**, if we could not find feasible h till $j = \bar{h}$, we stop searching and use $h = 1$ at the current stage.

5 Case Study

In this section we provide two case studies to evidence our theoretical arguments. We use the well-known knowledge gradient (KG, see Appendix) (Poloczek

et al., 2017) as the base heuristic in our rollout algorithm. We then show that KG is both sequentially consistent and improving, and thus it is rollout improving as shown in **Theorem 1**. We then illustrate that, under **Theorem 3** and through carefully choosing the rolling horizon, non-myopic BO has strong advantages over greedy BO. Our algorithm is tested for both single source and multi-information source BO (misoBO). In the misoBO setting, we sample from auxiliary information sources to make inference. Here we note that the details for misoBO are deferred to the appendix due to space limitation and similar conclusions to that of single source BO.

5.1 Setting

We use the same setting in 2.1. Specifically, when sampling from original function at input $\boldsymbol{x}$, we observe a noisy outcome $y(\boldsymbol{x})$. We assume the observation $y(\boldsymbol{x})$ is normally distributed with mean $f(\boldsymbol{x})$ and variance $\sigma^2(\boldsymbol{x})$. For the purpose of robustness, we assume that the covariance belongs to some non-smooth parametric family (Matern, etc..). Parameters are estimated using maximum likelihood estimation (MLE).

5.2 Algorithm

We utilize the non-greedy acquisition function defined in Sec. 2.2. This acquisition function considers far horizon planning and is given by the DP formulation. Specifically, we have $Q_k(\boldsymbol{x}; \boldsymbol{\mathcal{D}}_k) = \mathbb{E}\big[r_k(\boldsymbol{D}_k, \boldsymbol{x}_{k+1}) + \alpha R_{k+1}(\boldsymbol{D}_{k+1})\big]$. This is solved by the rollout with KG as the base heuristic. We denote our algorithm as DP-singleBO. The general procedure for DP-singleBO is listed in Algorithm 1. We also extend this algorithm to the multi-information sources scenario and denote it as DP-misoBO (see Appendix).

5.3 Guarantees

In order to apply **Theorem 1**, we need to show that the heuristic greedy KG is *sequentially consistent* and thus *sequentially improving*.

Corollary 2. *The KG algorithm is sequentially consistent and sequentially improving.*

Proof. Remember that state s_k is the dataset $\boldsymbol{D}_k$. Assume KG algorithm starts at a state s_k (i.e., current dataset $\boldsymbol{D}_k$). At each iteration of KG, given a path $(\boldsymbol{D}_k, \boldsymbol{D}_{k+1}, \ldots, \boldsymbol{D}_m)$ and $\boldsymbol{D}_m$ is not the state at the end, the next state $\boldsymbol{D}_{m+1}$ is obtained by solving the acquisition function of KG (see appendix) and augmenting $\boldsymbol{D}_m$ with $(\boldsymbol{x}^*, y)$. If $\boldsymbol{D}_{m+1}$ is not the terminating state, the algorithm will then start with the path $(\boldsymbol{D}_k, \boldsymbol{D}_{k+1}, \ldots, \boldsymbol{D}_m, \boldsymbol{D}_{m+1})$. Other-

---

Algorithm 1: The Non-myopic Single Information Source BO Algorithm

---

Data: Initial data $\boldsymbol{D}_1$, budget B and query cost c, number of remaining evaluations N, bound $\bar{h}$, number of node n.

Result: Data $\boldsymbol{D}_N$, optimal value $f_{max}^{\boldsymbol{D}_N}$, Gap G.

Fit $\mathcal{GP}$ to data $\boldsymbol{D}_1$ and obtain initial optimal value $f_{max}^{\boldsymbol{D}_1}$;

for $k = 1 : N$ **do**

  if $B < c$ **then**

    Directly return $\boldsymbol{D}_k$ as $\boldsymbol{D}_N$;

    STOP;

  else

    Calculate h in Sec. 4.2;

    Given h, select $\boldsymbol{x}_{k+1} = \arg\max_{\boldsymbol{x} \in \boldsymbol{\mathcal{X}}} Q_k(\boldsymbol{x}; \boldsymbol{D}_k)$
    s.t. $c(\boldsymbol{x}_{k+1}) \leq B$;

    $B \leftarrow B - c(\boldsymbol{x}_{k+1})$;

  end

  Evaluate $f(\cdot)$ at $\boldsymbol{x}_{k+1}$ and obtain y_{k+1};

  Augment the dataset
  $\boldsymbol{D}_{k+1} = \boldsymbol{D}_k \cup \{(\boldsymbol{x}_{k+1}, y_{k+1})\}$;

  Fit $\mathcal{GP}$ to data $\boldsymbol{D}_{k+1}$;

  $k \leftarrow k + 1$;

end

Fit $\mathcal{GP}$ to data $\boldsymbol{D}_N$;

Obtain optimal value $f_{max}^{\boldsymbol{D}_N}$;

Calculate the Gap G;

Return $\boldsymbol{D}_N$, $f_{max}^{\boldsymbol{D}_N}$ and G.

---

wise, the algorithm will terminate with state $\boldsymbol{D}_{m+1}$ and $N = m + 1$. We assume that, if there is a tie in the acquisition function, the algorithm will pick up the vector $\boldsymbol{x}$ with the smallest elements. Therefore, KG is sequentially consistent.

Let $(\boldsymbol{D}_1, \boldsymbol{D}_2, \ldots, \boldsymbol{D}_k, \ldots, \boldsymbol{D}_N)$ be the path generated by the rollout starting from $\boldsymbol{D}_1$. Define $\sigma(s)$ as the sub σ-algebra generated by state s. Since KG is sequentially consistent, we have

$$
\mathbb{E}\big[\sum_{\ell=k}^{N} r_\ell(s_\ell, \pi_\ell^{\boldsymbol{\pi}_{\mathcal{H}(s)}}(s_\ell))|\sigma(s')\big]
$$
$$
= \mathbb{E}\big[\sum_{\ell=k}^{N} r_\ell(s_\ell, \pi_\ell^{\boldsymbol{\pi}_{\mathcal{H}(s')}}(s_\ell))|\sigma(s')\big]. \tag{16}
$$

Therefore, KG is sequentially improving and we complete our proof. □

5.4 Results

5.4.1 Performance Comparison

In this section, we apply algorithms DP-singleBO and DP-misoBO to a variety of classical functions with

a range of dimensions, support sets and information sources. We provide three information sources in this experiment: original objective function (with noise) $y(\boldsymbol{x})$, biased source one $y(1, \boldsymbol{x})$ and biased source two $y(2, \boldsymbol{x})$. Following the setting from Poloczek et al. (2017), we define $y(1, \boldsymbol{x}) = y(\boldsymbol{x}) + 2\sin(10x_1 + 5x_2)$ in the two dimensional space and $y(1, \boldsymbol{x}) = y(\boldsymbol{x}) + 2\sin(10x_1 + 5x_2 + 3x_3)$ in the three dimensional space. We define $y(2, \boldsymbol{x}) = y(\boldsymbol{x}) + \delta(\boldsymbol{x})$, where $\delta(\boldsymbol{x})$ is simulated from $\mathcal{GP}$ with radial basis function (RBF) kernel with length-scale $l = 1$, signal variance $\sigma_f^2 = 1$ and noise variance $\sigma_n^2 = 0.5$. The RBF kernel is defined as $K_{RBF}(\boldsymbol{x}, \boldsymbol{x}') = \sigma_f^2 \exp\{-\frac{1}{2l^2}\|\boldsymbol{x} - \boldsymbol{x}'\|_2^2\} + \sigma_n^2 \mathbb{I}(\boldsymbol{x}, \boldsymbol{x}')$. See Table 1 for more information. For the Goldstein-price and Bohachevsky functions, we provide two biased sources and run DP-misoBO algorithm. For the Branin-Hoo, Six-Hump and Griewant, we run DP-singleBO algorithm. These objective functions have two notable challenges: (1) six-hump and Goldsterin-price have several local maxima; (2) Griewant function is in a high dimension with large design space. We benchmark our algorithms with several state-of-the-art techniques. A detailed code is available in the supplementary materials.

Table 1: Functions used in the experiment. More information about each function can be found at the open source library http://www.sfu.ca/~ssurjano/optimization.html.

Name	Function Domain
Branin-Hoo	$[-5, 10] \times [0, 15]$
Six-hump Camel	$[-3, 3] \times [-2, 2]$
Goldstein-price	$[-2, 2]^2$
Bohachevsky	$[-100, 100]^2$
Griewant-3	$[-600, 600]^3$

Experimental Details To mitigate the negative effect of model mis-specification, we fit $\mathcal{GP}$s with the matérn $p + \frac{1}{2}$ kernel and all hyperparameters are optimized by MLE. We set discount factor α to be 0.9. The optimal rolling horizon h is calculated at each stage. We also set $n = 10$ in the GaussHermite quadrature. The initial 9 sampling points are chosen by the fill distance design. For a fixed dimension d, we set an upper limit for sampling budget B and only allow around $10d$ evaluations of each algorithm. For example, we can set $B = 10d^2$, cost $c = 5d$ and $c_i = d, \forall i$. For each algorithm, we conduct 30 experiments with different initial points. In Table 2 we provide the testing results in terms of the mean and median of Gap, defined in Eq. (17).

Benchmark Models There is a limited literature on the non-greedy BO. We will benchmark our model

Table 2: Mean and median Gap G over 30 experiments with different initial points. The best result for each function is bolded. "NA" indicates not applicable. The discount factor is set to be 0.9.

| Function Name		GLASSES	M-EI	misoKG	MPI	LCB	**DP-singleBO/DP-misoBO**
Branin-Hoo	Mean	0.761	0.837	0.819	0.606	0.612	**0.864**
	Median	0.814	0.856	0.827	0.614	0.637	**0.889**
Six-Hump Camel	Mean	0.735	0.843	0.801	0.625	0.638	**0.870**
	Median	0.793	0.843	0.810	0.593	0.638	**0.866**
Goldstein-Price	Mean	NA	0.831	0.811	NA	NA	**0.867**
	Median	NA	0.837	0.846	NA	NA	**0.857**
Bohachevsky	Mean	NA	0.806	0.786	NA	NA	**0.872**
	Median	NA	0.821	0.820	NA	NA	**0.870**
Griewant-3	Mean	0.725	0.814	0.820	0.704	0.704	**0.861**
	Median	0.742	0.817	0.827	0.678	0.731	**0.856**
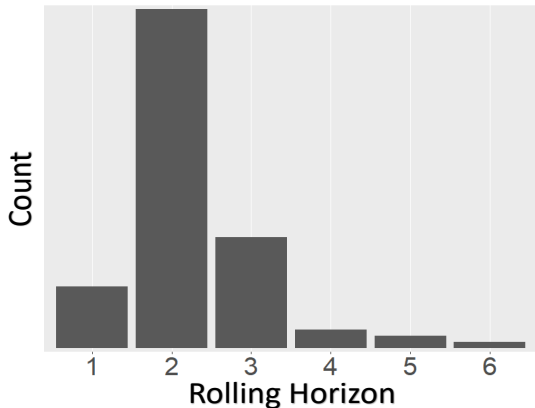
Figure 1: Distribution of Rolling Horizon

with the state-of-the-art GLASSES algorithm with fixed horizon, a DP-based algorithm using M-EI with fixed rolling horizon (Lam et al., 2015), Markov chain Monte Carlo (MCMC) based maximum probability of improvement (MPI) (Snoek et al., 2012), MCMC based lower confidence bound (LCB) (Snoek et al., 2012) and the misoKG (Poloczek et al., 2017). Note that GLASSES, MPI and LCB cannot be applied to the miso setting. We refer to section 6 for more details on the benchmarked models.

Performance The performance is measured in terms of Gap G, which is a common metric in many BO literature (Huang et al., 2006; González et al., 2016; Lam et al., 2016). Specifically,

$$G := \frac{f_{max}^{\boldsymbol{D}_1} - f_{max}^{\boldsymbol{D}_N}}{f_{max}^{\boldsymbol{D}_1} - f_{max}^*}, \qquad (17)$$

where $f_{max}^{\boldsymbol{D}_1}$ and $f_{max}^{\boldsymbol{D}_N}$ are optimal values given the initial and augmented data at stage N respectively and f_{max}^* is the global maximum of the testing function. Table 2 shows the comparative results across different functions and algorithms.

Based on Table 2, we can obtain some important insights. First, the results indicate that our model

clearly outperforms the state-of-the-art methods including non-myopic algorithms. The average and median Gaps of our algorithm are above 0.85, indicating that the estimations are improved 85% compared to the initial iteration. The key reason is that GLASSES and M-EI only consider fixed rolling horizon h, which is risky: the error propagation might eliminate the benefits of looking ahead. Indeed, choosing h stagewise allows us to carefully avoid the negative effect of model mis-specification. As shown in Table 3, when we choose fixed rolling horizon at each stage, the resulting Gap will be affected. When $h = 4, 5$, the non-greedy algorithm will even sabotage the performance. Here we note that we believe a dynamic rolling horizon can also improve the performance of GLASSES and M-EI. However, this requires further analysis and theoretical inquiries.

Second, the non-myopic algorithms are capable of beating the greedy algorithms. Interestingly, the feasible rolling horizon h^* is usually not large (Figure 1). This result is encouraging as it implies that the computational burden does not need to increase significantly since even a short horizon is beneficial. Therefore, it is over-pessimistic to discard non-myopia if one is afraid of error accumulation and computational complexity.

Lastly, the results indicate that the benefits of our method become increasingly significant for the high dimensional scenarios. This is intuitively understandable, due to ability of the non-greedy algorithm to efficiently explore the horizon.

5.4.2 Discount Factor

We study the effect of different discount factors. Specifically, we choose α from set $\{0.6, 0.7, 0.8, 0.9\}$. The discount factor plays a role in ceiling the value of rolling horizon as shown in **Theorem 3**. An extreme case is when $\alpha = 0$, the horizon is always 1 and reward is collected immediately (i.e., greedily). Based on Table 2 and 4, it seems that when $\alpha \in \{0.8, 0.9\}$, the performance is promising. This result is intuitive as a moderate discount factor encourages an algorithm

to consider collecting reward further and is capable of generating improving results.

Table 3: Mean Gap G with respect to different fixed rolling horizon over 30 experiments with different initial points.

Function Name	$h = 2$	$h = 3$	$h = 4$	$h = 5$
Branin-Hoo	0.830	0.805	0.777	0.700
Six-Hump Camel	0.855	0.860	0.671	0.665
Goldstein-Price	0.829	0.824	0.732	0.667
Bohachevsky	0.865	0.788	0.721	0.648
Griewant-3	0.802	0.755	0.621	0.683

Table 4: Mean Gap G with respect to different discount factor over 30 experiments with different initial points.

Function Name	$\alpha = 0.6$	$\alpha = 0.7$	$\alpha = 0.8$	$\alpha = 0.9$
Branin-Hoo	0.812	0.801	0.867	0.864
Six-Hump Camel	0.780	0.810	0.871	0.870
Goldstein-Price	0.826	0.804	0.844	0.867
Bohachevsky	0.803	0.818	0.853	0.872
Griewant-3	0.764	0.830	0.845	0.861

6 Literature Review

6.1 Nonmyopia

Few literature has focused on the non-myopic BO. Ginsbourger and Le Riche (2010) propose an expectation improvement (EI) criterion to derive sequential sampling strategies using Monte-Carlo simulation. Later, some approximation algorithms have been proposed that provide theoretical guarantees when sampling spaces are finite (Marchant et al., 2014; Ling et al., 2016). Unfortunately such algorithms scale poorly with the number of rolling horizon considered. Later, González et al. (2016) provided the GLASSES algorithm that relieves the myopia assumption of BO and can efficiently tackle an uncountable sampling space. The GLASSES utilize the long-sight loss function in Osborne (2010) and then propose an efficient optimization-marginalization scheme to solve that loss. Despite its strength, this approach assumes that the objective function is L-Lipschitz continuous. Besides the aforementioned methods, there exists some efficient multi-step look-ahead algorithms in the area of Bayesian feasibility determination and root-finding problems (Waeber et al., 2013; Cashore et al., 2016). Nevertheless, they are only applicable to a very specific physical setting and cannot be easily generalized to a general framework. More Recently, Lam (Lam et al., 2016; Lam and Willcox, 2017) proposed a lookahead DP formulation using EI as a heuristic reward function. A direct extension to this work includes using the modified-EI (M-EI) (Groot et al., 2010; Lam et al., 2015) instead of EI to handle multi-information

sources. However, a crucial drawback of the M-EI is that its selects sampling point and query sources separately. This might lead to reduced accuracy as joint optimality is not considered. Recently, Jian and Peter (2019) has proposed a practical two-step lookahead BO algorithm. This is one successful example that illustrates the benefits of looking sightly ahead.

6.2 Multi-information Source

We provide a short review on misoBO for completeness. Multi-information source optimization was thoroughly studied by Swersky et al. (2013). The authors argue that auxiliary tasks can aid in solving some expensive optimization problems. Swersky et al. (2013) utilize a multivariate Gaussian process $\mathcal{GP}$ (Seeger et al., 2005; Bonilla et al., 2008) to model uncertainties in the objective function and predictive entropy search to decide on the next sampling location. Very recently, Poloczek et al. (2017) improved the misoBO algorithm through utilizing a more flexible $\mathcal{GP}$ construction, using the linear model of coregionalization, and extending the KG algorithm to the setting with variable cost across multiple information sources. They showed that the improved method (denoted as misoKG) can find sampling locations with higher value at reduced cost. Despite this seminal work, the misoKG does not consider far horizon planning since it uses a one-step lookahead approach that only considers reducing regret at the next step. Besides misoBO, other closely related work belong to the problem of multi-fidelity optimization (McLeod et al., 2017; Kandasamy et al., 2016; Cutajar et al., 2019). These models have been mainly based on hierarchical model structures that restrict the information to be shared from low fidelity models. Also, they implement a myopic approach and fail to account for the future information such as remaining budgets.

Conclusion

We provide a theoretical proof of the "improving" nature of the rollout DP algorithm and a practical guideline on choosing a sequence of rolling horizons. We argue that rollout with a well chosen rolling horizen is beneficial in the sense that the error propagation is not catastrophic and the profits from the rollout improving nature remain. Therefore, the rollout DP has great promise in BO theory and applications. One possible future work is to generalize our analysis and apply it to other non-myopic methods. We hope our work will help inspire continued exploration into the non-myopic algorithms.

Reference

D. P. Bertsekas. *Dynamic programming and optimal control*, volume 1. Athena scientific Belmont, MA, 1995.

D. P. Bertsekas. Rollout algorithms for constrained dynamic programming. *Lab. for Information and Decision Systems Report*, 2646, 2005.

D. P. Bertsekas, J. N. Tsitsiklis, and C. Wu. Rollout algorithms for combinatorial optimization. *Journal of Heuristics*, 3(3):245–262, 1997.

E. V. Bonilla, K. M. Chai, and C. Williams. Multi-task gaussian process prediction. In *Advances in neural information processing systems*, pages 153–160, 2008.

J. M. Cashore, L. Kumarga, and P. I. Frazier. Multi-step bayesian optimization for one-dimensional feasibility determination. *arXiv preprint arXiv:1607.03195*, 2016.

K. Cutajar, M. Pullin, A. Damianou, N. Lawrence, and J. González. Deep gaussian processes for multi-fidelity modeling. *arXiv preprint arXiv:1903.07320*, 2019.

D. Ginsbourger and R. Le Riche. Towards gaussian process-based optimization with finite time horizon. In *mODa 9–Advances in Model-Oriented Design and Analysis*, pages 89–96. Springer, 2010.

J. González, M. Osborne, and N. D. Lawrence. Glasses: Relieving the myopia of bayesian optimisation. 2016.

J. C. Goodson, B. W. Thomas, and J. W. Ohlmann. A rollout algorithm framework for heuristic solutions to finite-horizon stochastic dynamic programs. *European Journal of Operational Research*, 258(1):216–229, 2017.

P. Groot, A. Birlutiu, and T. Heskes. Bayesian monte carlo for the global optimization of expensive functions. In *ECAI*, pages 249–254, 2010.

P. Hennig and C. J. Schuler. Entropy search for information-efficient global optimization. *Journal of Machine Learning Research*, 13(Jun):1809–1837, 2012.

J. M. Hernández-Lobato, M. W. Hoffman, and Z. Ghahramani. Predictive entropy search for efficient global optimization of black-box functions. In *Advances in neural information processing systems*, pages 918–926, 2014.

D. Huang, T. T. Allen, W. I. Notz, and N. Zeng. Global optimization of stochastic black-box systems via sequential kriging meta-models. *Journal of global optimization*, 34(3):441–466, 2006.

W. Jian and F. Peter. Practical two-step lookahead bayesian optimization. In *Advances in neural information processing systems*, 2019.

M. E. Johnson, L. M. Moore, and D. Ylvisaker. Minimax and maximin distance designs. *Journal of statistical planning and inference*, 26(2):131–148, 1990.

K. Kandasamy, G. Dasarathy, J. B. Oliva, J. Schneider, and B. Póczos. Gaussian process bandit optimisation with multi-fidelity evaluations. In *Advances in Neural Information Processing Systems*, pages 992–1000, 2016.

R. Lam and K. Willcox. Lookahead bayesian optimization with inequality constraints. In *Advances in Neural Information Processing Systems*, pages 1890–1900, 2017.

R. Lam, D. L. Allaire, and K. E. Willcox. Multifidelity optimization using statistical surrogate modeling for non-hierarchical information sources. In *56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, page 0143, 2015.

R. Lam, K. Willcox, and D. H. Wolpert. Bayesian optimization with a finite budget: An approximate dynamic programming approach. In *Advances in Neural Information Processing Systems*, pages 883–891, 2016.

C. K. Ling, K. H. Low, and P. Jaillet. Gaussian process planning with lipschitz continuous reward functions: Towards unifying bayesian optimization, active learning, and beyond. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

R. Marchant, F. Ramos, S. Sanner, et al. Sequential bayesian optimisation for spatial-temporal monitoring. In *Uncertainty in Artificial Intelligence*, pages 553–562, 2014.

M. McLeod, M. A. Osborne, and S. J. Roberts. Practical bayesian optimization for variable cost objectives. *arXiv preprint arXiv:1703.04335*, 2017.

M. A. Osborne. *Bayesian Gaussian processes for sequential prediction, optimisation and quadrature*. PhD thesis, Oxford University, UK, 2010.

M. Poloczek, J. Wang, and P. Frazier. Multi-information source optimization. In *Advances in Neural Information Processing Systems*, pages 4288–4298, 2017.

C. E. Rasmussen. Gaussian processes in machine learning. In *Summer School on Machine Learning*, pages 63–71. Springer, 2003.

T. J. Santner, B. J. Williams, W. Notz, and B. J. Williams. *The design and analysis of computer experiments*, volume 1. Springer, 2003.

M. Seeger, Y.-W. Teh, and M. Jordan. Semiparametric latent factor models. Technical report, 2005.

S. Sethi and G. Sorger. A theory of rolling horizon decision making. *Annals of Operations Research*, 29 (1):387–415, 1991.

J. Snoek, H. Larochelle, and R. P. Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959, 2012.

K. Swersky, J. Snoek, and R. P. Adams. Multi-task bayesian optimization. In *Advances in neural information processing systems*, pages 2004–2012, 2013.

M. W. Ulmer, J. C. Goodson, D. C. Mattfeld, and M. Hennig. Offline–online approximate dynamic programming for dynamic vehicle routing with stochastic requests. *Transportation Science*, 53 (1):185–202, 2018.

R. Waeber, P. I. Frazier, and S. G. Henderson. Bisection search with noisy responses. *SIAM Journal on Control and Optimization*, 51(3):2261–2279, 2013.

W. Wang, R. Tuo, and C. Jeff Wu. On prediction properties of kriging: Uniform error bounds and robustness. *Journal of the American Statistical Association*, pages 1–27, 2019.

H. Wendland. *Scattered data approximation*, volume 17. Cambridge university press, 2004.