*Proceedings of the 2022 Winter Simulation Conference*
*B. Feng, G. Pedrielli, Y. Peng, S. Shashaani, E. Song, C.G. Corlu, L.H. Lee, E.P. Chew, T. Roeder, and P. Lendermann, eds.*

# NON-MYOPIC KNOWLEDGE GRADIENT POLICY FOR RANKING AND SELECTION

Kexin Qin
L. Jeff Hong

School of Data Science
Fudan University
220 Handan Road
Shanghai, 200433, CHINA

Weiwei Fan

Advanced Institute of Business
Tongji University
1239 Siping Road
Shanghai, 200092, CHINA

## ABSTRACT

We consider the ranking and selection (R&S) problem with fixed simulation budget, in which the budget is assumed to be allocated sequentially. Deriving the optimal sampling procedure for this problem amounts to solving a stochastic dynamic program that is highly intractable. To overcome this difficulty, the existing R&S procedures are often designed from a myopic viewpoint. However, these myopic procedures are only single-step optimal and may have a poor performance for general sequential R&S problems. Therefore, in this paper, we combine two popular lookahead strategies and design a non-myopic knowledge gradient (KG) procedure. Meanwhile, to streamline the computation of procedure, we propose a modified Monte Carlo tree search method specifically designed under the R&S context. We show that the new procedure can exhibit a performance superior to the classic KG.

## 1 INTRODUCTION

Ranking and selection (R&S) is a classic problem of the simulation community, in which we aim to select the best alternative from a finite set of alternatives regarding their mean performances. Such problems have found a lot of applications in the risk measurement, healthcare management and so on. Typically, the mean performance of each alternative is unknown and needs to be estimated via running stochastic simulation. Due to the inherited randomness, the estimates become more accurate as more simulation samples are collected on the alternatives. However, running simulation is often computationally expensive. When the total simulation budget is limited, the main task of R&S is to determine a sampling decision rule that allocates this budget in an efficient manner. Such decision rule is also known as a fixed-budget R&S procedure in the literature.

Suppose that the simulation budget is allocated sequentially, according to a certain decision rule. Particularly, at each step, the decision rule chooses one alternative and collects a random sample from it. When the total budget is exhausted, the best alternative is selected based on all the samples collected. We call a decision rule as optimal if it optimizes the quality of final selection. In fact, it is known that, solving the optimal decision rule can be essentially formulated as a finite-horizon stochastic dynamic program (DP) (see, e.g., Frazier et al. (2008) and Hong et al. (2021)). Traditionally, this DP could be solved exactly by backward iterations through the Bellman equation. Unfortunately, this is subject to a computation burden due to the "curse of dimensionality", thereby making the exact solution of DP possibly intractable.

To overcome this difficulty, a series of R&S papers turn to explore a reasonable and simultaneously tractable approximation for the underlying DP. Based on these approximations, a set of sub-optimal decision rules are proposed as a compromise. For instance, Chen et al. (2000) approximate the original DP with a static optimization problem and derive a static-allocation R&S procedure instead using the large-deviation theory. This static procedure is known as the optimal computing budget allocation (OCBA) procedure and

is later extended to the sequential setting. In contrast, another stream of papers take a myopic perspective to approximate the Bellman equation in a tractable form, and accordingly design several myopic DP procedures including the expected value of information (EVI) procedure (Chick and Inoue 2001) and the knowledge gradient (KG) procedure (Frazier et al. 2008). These myopic procedures have now served as the cornerstone of the fixed-budget R&S.

Beyond the R&S, it has been noticed that the myopic procedures often perform poorly for general DP problems, e.g., Gonzalez et al. (2016), Lam et al. (2016), and Yue and Kontar (2020). The main reason is that, the myopic procedures, a.k.a., the single-step lookahead procedures, make each-step decision without regarding the long-term impact of future samples. Intuitively, if we could look more step ahead into the future, we may obtain a new procedure that improves the myopic procedures to a large extent. Actually, there has been a large volume of literature working on this topic in the DP area and proposing a variety of powerful tools for the design of non-myopic procedures. In spite of the above, it is rarely discussed in the R&S literature whether a non-myopic procedure can improve the existing myopic R&S procedure; and if so, how to design such non-myopic procedures. Addressing this issue is the goal of this paper.

In general, a sequential R&S procedure mainly contains two iterative stages, namely, the approximation of value function and optimization over the approximation. Therefore, to design an efficient non-myopic R&S procedure, we need to first tackle these two issues. For the approximation issue, we combine two popular lookahead strategies in the literature, i.e., multi-step lookahead and rollout strategies, to provide a new approximation for the value function. When the classic myopic KG is chosen as the base policy in the rollout, it is shown that the new approximation appears more accurate than the classic KG. For the optimization issue, notice that it is clearly a multi-step lookahead optimization problem in the non-myopic setting. To tackle this problem efficiently, we propose a modified Monte Carlo Tree Search (MCTS) technique by taking advantage of the special structure of R&S. As a result, a non-myopic KG procedure is obtained, and we can show that it has a superior performance over its myopic counterpart.

The rest of the paper is organized as follows. Section 2 introduces the problem formulation as well as the classic myopic KG procedure. Section 3 explains two possible strategies that enable to improve the myopic KG procedure and a combined new approach. Section 4 proposes the non-myopic KG policy and shows its theoretical properties. Section 5 demonstrates the superiority of the non-myopic KG procedure through numerical experiments and Section 6 concludes the paper.

## 2 PROBLEM FORMULATION

We consider a R&S problem with $K$ alternatives, denoted by $\mathscr{K} = \{1, \ldots, K\}$. For each alternative $x \in \mathscr{K}$, let $\mu_x$ denote its unknown mean performance, which needs to be evaluated via running stochastic simulations. Without loss of generality, we assume that the best alternative has the largest mean performance, i.e., $\max_{x \in \mathscr{K}} \mu_x$. When the total simulation budget $N$ is fixed, our goal is to design a decision rule (or R&S procedure) which tells how to allocate this budget efficiently.

We first introduce some notations. Assume that the samples from each alternative $x \in \mathscr{K}$ are independent and normally distributed with mean $\mu_x$ and variance $\sigma_x^2$. Following the Bayesian viewpoint, the unknown parameter $\mu_x$ is typically viewed as a random variable, and we assume it follows a prior normal distribution with mean $\mu_x^0$ and variance $(\sigma_x^0)^2$. We use $\beta_x^0 := (\sigma_x^0)^{-2}$ to denote the precision of the corresponding normal distribution.

### 2.1 Fixed-Budget R&S versus Dynamic Program

Suppose that the simulation budget is allocated one by one. At each step $0 \le n < N$, we choose one alternative $x^n \in \mathscr{K}$ to sample from and then collect an observed sample, termed by $y^{n+1}$. Define a filtration $\{\mathscr{F}^n : 0 \le n \le N\}$, with $\mathscr{F}^n$ being the sigma-algebra generated by $\{x^0, y^1, x^1, ..., x^{n-1}, y^n\}$. Generally, the sampling decision $x^n$ is obligated to be $\mathscr{F}^n$-measurable so that the decision is only determined by the sampling decisions made and the samples collected in the past.

Along with the sequential sampling process, we also use the Bayes rule to iteratively calculate the posterior distribution of $\mu_x$. In particular, let $\mu_x^n$ and $\beta_x^n$ denote the posterior mean and precision of $\mu_x$ after $n$ steps. When the total simulation budget is exhausted, the alternative with the largest posterior mean (i.e., $\max_x \mu_x^N$) is selected as the best. Apparently, the quality of final selection should hinge on the sampling policy $\pi = \left(x^0, \ldots, x^{N-1}\right)$ that is used. Let $\Pi$ denote the set of all the possible sampling policies, namely

$$\Pi = \left\{ \pi = \left(x^0, \ldots, x^{N-1}\right) : x^n \in \mathscr{K} \text{ is } \mathscr{F}^n\text{-measurable}, \forall 0 \le n < N \right\},$$

and therefore our goal is to find the optimal sampling policy $\pi$ which solves

$$\sup_{\pi \in \Pi} \mathbb{E}^{\pi} \left[ \max_x \mu_x^N \right], \tag{1}$$

where $\mathbb{E}^{\pi}[\cdot]$ denotes the expectation taken when the sampling policy $\pi$ is used.

Clearly, Problem (1) above can be viewed as a DP. To facilitate the presentation, we write the posterior means and precisions of $\mu_x's$ in a matrix form, $\mu^n = (\mu_1^n, \mu_2^n, \ldots, \mu_K^n)$ and $\beta^n = (\beta_1^n, \beta_2^n, \ldots, \beta_K^n)$. Let $S^n = (\mu^n, \beta^n)$ denote the state at step $n$, and define the state space $\mathbb{S} = \mathbb{R}^K \times (0, \infty]^K$. Then, the terminal value function is written as

$$V^N(s) = \max_x \mu_x, \quad \text{for every } s \in \mathbb{S}.$$

### 2.2 Myopic Knowledge-Gradient Policy

From above, the optimal sampling policy $\pi_* = (x_*^0, x_*^1, \ldots, x_*^{N-1})$ can be derived through solving the DP stated in (1). In particular, at each step $0 \le n < N$, the optimal sampling decision $x_*^n$ is determined by the associated Bellman equation

$$\begin{aligned} V^n(s) &= \max_{x \in \mathscr{K}} \mathbb{E}\left[ V^{n+1}(S^{n+1}) | S^n = s, x^n = x \right], \\ x_*^n(s) &= \arg\max_{x \in \mathscr{K}} \mathbb{E}\left[ V^{n+1}(S^{n+1}) | S^n = s, x^n = x \right]. \end{aligned} \tag{2}$$

If the value function $V^{n+1}(\cdot)$ has an explicit form or can be computed efficiently, then the optimal decision $x_*^n$ may be readily obtained. Unfortunately, $V^{n+1}(\cdot)$ is often computational intensive to compute due to "the curse of dimensionality". Therefore, to avoid solving this intractable problem directly, many approximation methods have been utilized in the literature.

Among these approximation methods, the Knowledge Gradient (KG) policy takes a myopic viewpoint which always regards the next stage as the terminal stage, and consequently approximates the intractable $V^{n+1}(\cdot)$ by the terminal value function $V^N(\cdot)$. It follows that, an approximated sampling decision is given by

$$\begin{aligned} x^{KG}(s) &= \arg\max_{x \in \mathscr{K}} \mathbb{E}\left[ V^N(S^{n+1}) | S^n = s, x^n = x \right] \\ &= \arg\max_{x \in \mathscr{K}} \left\{ \mathbb{E}\left[ V^N(S^{n+1}) - V^N(S^n) | S^n = s, x^n = x \right] \right\}. \end{aligned} \tag{3}$$

The second equality holds because the added term $V^N(S^n)$ is a constant unrelated to the decision $x$. The policy in (3) is named knowledge-gradient policy because it finds the sampling decision that maximizes the expected improvement of value over the next sampling. The major advantage of the KG policy is its computational complexity which grows linear with the number of alternatives, i.e., $|\mathscr{K}|$ (Frazier, Powell, and Dayanik 2008). This appears much more efficient than directly solving the original DP problem, in the computational sense.

However, the KG policy is clearly sub-optimal. It makes each-step sampling decision by only considering the next-step sample, thereby ignoring the long-term impact of all the future samples. Intuitively, when the remaining simulation budget $N-n$ is relatively large, there might exist a large gap between the desired value function $V^{n+1}$ and its approximation $V^N$. To some extent, this gap explains why the myopic procedures could perform poorly in certain situations as stated in Section 1.

## 3 IMPROVEMENT OF MYOPIC POLICY

As stated in Section 2, the limitation of myopic procedures arises mainly because they are built upon a less accurate approximation for value function. In particular, the approximated value function they use is single-step lookahead in the sense that it only takes the next-step sample into consideration. Thus, to improve the performance of myopic procedures, the key task is to explore the long-term value of more future samples and provide a better approximation for the value function.

### 3.1 From Single-step to Multi-step

Suppose that, at each step $0 \leq n < N$, we seek to compute the current-step value function $V^n$ by considering more than one future sample, say $t \geq 1$ samples. This essentially requires us to build a bridge connecting $V^n$ with the future value function $V^{n+t}$ after $t$ steps. Actually, their relationship can be given by the $t$-step lookahead version of Bellman equation, that is

$$V^n(s) \quad = \quad \max_{x^n,\ldots,x^{n+t-1}} \mathbb{E}\left[V^{n+t}(S^{n+t})|S^n = s\right], \text{ for } 0 \leq n < N. \tag{4}$$

Generally, $V^{n+t}$ is still computationally intractable because it has to be computed recursively by (4). Similar to the myopic policy, we may simply pretend step $n+t$ as the terminal step and then replace $V^{n+t}$ by the terminal value function $V^N$. Then, an approximation for $V^n(s)$ is given by

$$\max_{x^n,\ldots,x^{n+t-1}} \mathbb{E}\left[V^N(S^{n+t})|S^n = s\right] = V^{N-t}(s). \tag{5}$$

We use $E^t V^n$ to denote the approximation error arising here, namely,

$$E^t V^n(s) := V^n(s) - V^{N-t}(s), \ \forall\ s \in \mathbb{S} \text{ and } n \leq N-t. \tag{6}$$

In the special case as $t = 1$, $E^t V^n(s)$ refers to the approximation error in the single-step lookahead (i.e., myopic) scheme. Intuitively, one may conjecture that the $t$-step lookahead scheme could provide a better approximation for (4) than the myopic one, or equivalently, $E^t V^n(s) \leq E^1 V^n(s)$ for any $t \geq 1$. However, it is not always true for general DP problems. Luckily, by exploring the special structure of R&S, we show in Lemma 3.1 that this conjecture is true.

**Lemma 3.1** For any state $s \in \mathbb{S}$ and any $1 \leq t \leq N-n$, we have that

$$0 \leq E^t V^n(s) \leq E^{t-1} V^n(s) \leq \cdots \leq \ldots E^1 V^n(s).$$

Lemma 3.1 also implies that, whenever we look one more steps ahead into the future, the corresponding approximation error can be further reduced. However, such reduction tends to be marginally diminishing as shown in Lemma 3.2. Therefore, in practice, even a relatively small $t$ may be enough to help improve the value function approximation.

**Lemma 3.2** For any $1 \leq t \leq N-n-2$, we have that $\|E^{t+2}V^n - E^{t+1}V^n\|_\infty \leq \|E^{t+1}V^n - E^t V^n\|_\infty$, where $\|\cdot\|_\infty$ denotes the $L^\infty$ norm.

### 3.2 Rollout Strategy

When the long-term impact of future samples needs to be considered, we have to evaluate the value function according to the Bellman equation (2) iteratively. Each iteration involves a stochastic optimization problem to solve. Obviously, this is computationally infeasible. To overcome this barrier, one popular technique, called rollout strategy, is proposed and has been shown to perform excellently in many practical situations (Bertsekas 2012; Sutton and Barto 2018). More specifically, the rollout strategy frees itself from solving the optimization problem at each iteration, but instead implements a heuristic policy (often called base policy) over the future steps. As a result, a reasonable approximation of the desired value function is yielded.

Suppose that a base policy $B$ will be implemented after step $n$. The approximation for the intractable $V^{n+1}$ in (2) by rollout strategy is constructed as the reward-to-go at step $n+1$ under the base policy $B$, i.e., $V^{n+1,B}$, which represents the policy value under base policy $B$. Denote $x^{B,n}(s)$ as the alternative chosen by policy $B$ at time $n$, the policy value is defined as

$$V^{n,B}(s) := \mathbb{E}^B \left[ V^N \left( S^N \right) \mid S^n = s \right] = \mathbb{E} \left[ V^{n+1,B} \left( S^{n+1} | S^n = n, x^n = x^{B,n}(s) \right) \right].$$

Thus, the approximation for $V^n(s)$ of rollout strategy under base policy $B$ is given by

$$\max_{x \in \mathscr{X}} \mathbb{E} \left[ V^{n+1,B}(S^{n+1}) | S^n = s, x^n = x \right].$$

It follows that, the corresponding approximation error is given by

$$E^B V^n(s) := V^n(s) - \max_{x \in \mathscr{X}} \mathbb{E} \left[ V^{n+1,B}(S^{n+1}) | S^n = s, x^n = x \right], \ \forall \ s \in \mathbb{S}.$$

Generally, there is no clear rule on the choice of a proper base policy. In fact, it can be chosen according to the users' own interest. In this paper, we pick up the myopic KG in (3) for its convenience to implement. As shown in Section 2, the myopic KG policy is clearly sub-optimal for the original DP. But, with the help of rollout strategy, the KG does provide a possible way to take into account the long-term value of future samples. In light of this, we show in Lemma 3.3 that the rollout strategy can bring a better value function approximation for $V^n$ than the corresponding myopic KG policy. To understand this lemma, notice from (6) that $E^1 V^n(s)$ refers to the approximation error of the myopic KG policy, and $E^{KG} V^n(s)$ refers to the approximation error of the rollout strategy under base policy $KG$.

**Lemma 3.3** For any state $s \in \mathbb{S}$, we have that $E^{KG} V^n(s) \geq E^1 V^n(s)$.

### 3.3 Combination of Multi-step and Rollout

So far, we have introduced two strategies to improve the value approximation function: multi-step and rollout. The multi-step lookahead policy can be regarded as a relatively accurate "short-term" approximation, as the lookahead steps cannot be set too large. Otherwise the associated multi-step optimization problem as shown in (4) could become difficult to address. In contrast, the rollout policy enables to provide the "long-term" approximation because it does not involve any optimization. However, such approximation might not be so accurate, compared to multi-step lookahead policy. Therefore, it seems promising to combine these two strategies to further improve the value function approximation.

Suppose that the combined approach is used to construct an approximation for $V^n(s)$ at each step $0 \leq n < N$ and state $s$. It employs the multi-step lookahead policy at the first $t$ steps and then implements the rollout strategy until the terminal $N$. In doing so, an approximation for $V^n(s)$ of is obtained as follow,

$$\max_{x^n, x^{n+1}, \ldots, x^{n+t-1}} \mathbb{E} \left[ V^{n+t,B} \left( S^{n+t} \right) \mid S^n = s \right]. \tag{7}$$

The new approximation above differs from the previous multi-step lookahead approximation (5) mainly in the way of dealing with the intractable $V^{n+t}(\cdot)$ in the multi-step Bellman equation (4). More specifically, the combined approach replaces $V^{n+t}(\cdot)$ with $V^{n+t,KG}(\cdot)$ by the rollout strategy, whereas (5) simply replaces $V^{n+t}(\cdot)$ by the terminal value $V^N(\cdot)$. Denote the approximation error of combined strategy as

$$E^{t,KG} V^n(s) := V^n(s) - \max_{x^n, x^{n+1}, \ldots, x^{n+t-1}} \mathbb{E} \left[ V^{n+t,KG} \left( S^{n+t} \right) \mid S^n = s \right], \ \forall \ s \in \mathbb{S}.$$

Then we show in the following lemma that, the combined approach has a smaller approximation error of $V^n$ than using either lookahead strategy separately.

**Lemma 3.4** For any state $s \in \mathbb{S}$, we have that $E^{t,KG} V^n(s) \leq E^t V^n(s)$ and $E^{t,KG} V^n(s) \leq E^{KG} V^n(s)$.

## 4 NON-MYOPIC KG POLICY

Generally, deriving the optimal sampling decision under a DP formulation often involves two consecutive tasks: value function approximation and computation of the optimal decision from the approximation. Section 3 has introduced an effective approach to the value function approximation as in (7). From this approximation, the core of this section is on how to efficiently compute the corresponding optimal decision rule, namely, deciding the alternative to simulate from at each step in the R&S.

### 4.1 Computation of Non-myopic KG Policy

As the value function approximation in (7) is developed under a $t$-step lookahead scheme, solving it fairly suggests the sampling decisions in the following $t$ steps given current state. To be more specific, at each step $0 \leq n < N$ with state $S^n = s$, we solve (7) to obtain

$$\left(R_t^n, R_{t-1}^{n+1}, ..., R_1^{n+t-1}\right)(s) = \underset{x^n, x^{n+1}, ..., x^{n+t-1}}{\arg\max} \mathbb{E}\left[V^{n+t,KG}\left(S^{n+t}\right) \mid S^n = s\right], \tag{8}$$

in which $(R_t^n, R_{t-1}^{n+1}, ..., R_1^{n+t-1})$ denote the suggested sampling decisions for the next $t$ steps. As the desired sampling process is sequential, it is reasonable to implement the decision $R_t^n$ at current step $n$ and then move to the next step with an additional sample. We call this sampling rule $R_t^n$ the non-myopic KG policy.

Unfortunately, computing of the non-myopic KG policy by (8) is not an easy issue. Firstly, the objective function in (8) has no analytical form, and needs to be obtained via Monte Carlo simulation. Particularly, the simulation samples here are collected according to the posterior sampling distribution of each alternative. Such samples are often called "fantasy samples" in the literature as they mimic the true samples that should be collected from alternatives. Secondly, exactly solving (8) requires visiting every possible sample allocation path $(x^n, x^{n+1}, \ldots, x^{n+t-1}) \in \mathscr{K}^t$ over the following $t$ steps. Apparently, this would take extremely lots of simulation samples as the number of all possible sampling paths reaches $|\mathscr{K}|^t$. Notice that Problem (8) could be viewed as a scenario tree, as illustrated in Figure 1, where each branch in the tree refers to a particular alternative and each path from the root node (the topmost node of the tree, $S^n$) to the leaf node (nodes without children) refers to a certain sampling allocation path. In light of this, we resort to Monte Carlo tree search (MCTS) method (Browne et al. 2012) which intelligently selects these promising sampling paths rather than viewing all the paths equally important.

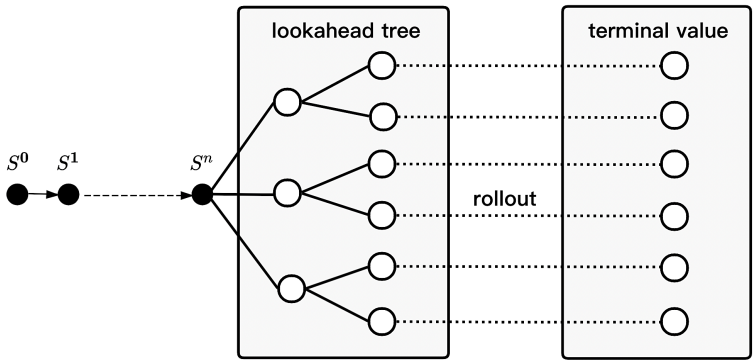


Figure 1: Tree Structure of Optimization Problem (8).

We follow the four steps of traditional MCTS structure (Browne et al. 2012), namely, Selection, Expansion, Simulation and Backpropagation, but with some adaptations to the R&S problem. Particularly, the implementation of MCTS in non-myopic KG context is shown in Figure 2. At the Selection step, the main goal is to select the "best" child node according to some heuristic selection algorithm. One commonly

used algorithm in MCTS for selection, called UCT (Upper Confidence Bound Applied to Trees, Kocsis and Szepesvári (2006)), is designed based on the UCB formula (Auer et al. 2002). The value of node $v^n$ which corresponds to an alternative $v$ at time $n$ given by the UCT algorithm is

$$UCT(v^n) = \frac{Q(v^n)}{N(v^n)} + c\sqrt{\frac{2\log N(v_p^n)}{N(v^n)}}, \tag{9}$$

where $Q(v^n)$ is the accumulated reward (sum of reward) of node $v^n$, $N(v^n)$ represents the visited times of $v^n$ in tree simulations, $v_p^n$ is the parent node of $v^n$ in tree. The reward is backpropagated to $v^n$ via the backpropagation step from the leaf node as shown in Figure 2.

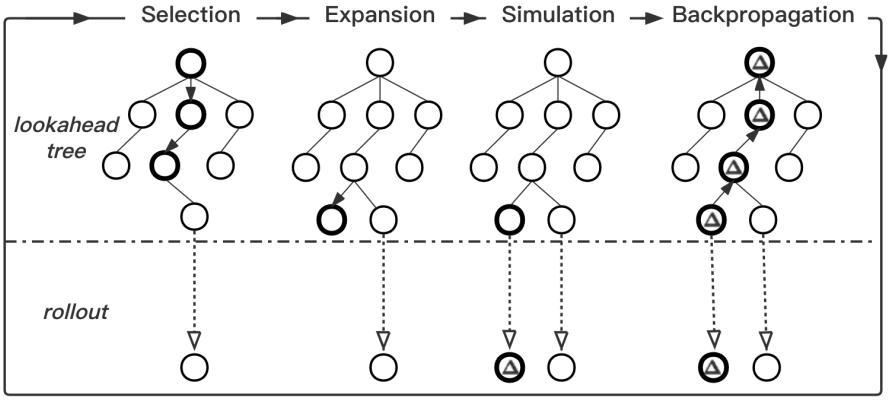


Figure 2: Four steps of implementing MCTS in non-myopic KG policy .

However, the search tree is of exponential growth, and typically has a huge search space especially when the number of alternatives is large. So it might be quite simulation-consuming to implement MCTS. A series of papers have studied the method of narrowing the beam of the search tree to high-probability moves to improve the efficiency, e.g. Coulom (2007), Rosin (2011) and Silver et al. (2016). In this paper, we choose to add a prior information provided by OCBA procedure (Chen et al. 2000) to speed up the MCTS process.

The motivation comes from the design of game algorithms, such as Alpha Go (Silver et al. 2016), which adds a "prior" to the exploration term to reduce the search space and improve the efficiency. The core idea is to increase the chance of simulations for nodes with high prior probability of "winning", while for nodes with poor predicted performance, we could just reduce the simulation cost and save computation. This selection control strategy initially prefers alternatives with high prior probability and low visit count, but asymptotically prefers actions with high value. When the prior offers an accurate estimation, the algorithm will have a significant speedup compared to the original UCT, and converges much quicker. In the Go games, the prior is the probability of "winning", which represents the probability of making a move at each place, while in the R&S problem, our prior should be the probability of "choosing", which represents the probability of taking a sample at each alternative. We find that a static-allocation procedure for R&S, the OCBA procedure (Chen et al. 2000), provides an optimal sampling allocation proportion under the assumption that the total sample budget tends to infinity. Despite the strong assumption, we can still use the proportion of sample allocation for each alternative given by OCBA, as our prior of the probability of "choosing" each alternative. Under the Bayesian framework, the sample allocation proportion of OCBA

(Chen et al. 2000) is given by

$$P_{v^n}^{OCBA} \propto \frac{\sigma_v^2}{\left(\mu_v^n - \mu_q^n\right)^2}(v \neq q), \quad P_{q^n}^{OCBA} \propto \sqrt{\sigma_q^2}\sqrt{\sum_{x \neq q} \frac{\sigma_x^2}{\left(\mu_x^n - \mu_q^n\right)^4}}, \quad q = \arg\max_x \mu_x^n. \tag{10}$$

Our prior $P_{v^n}^{OCBA}$ is then derived via the constraint $\sum_x P_{x^n}^{OCBA} = 1$ . With the defined prior probability in (10), the selection value of node $v^n$ of non-myopic KG policy (following the notations in UCT algorithm) is defined as:

$$\widetilde{UCT}(v^n) = \frac{Q(v^n)}{N(v^n)} + c \cdot P_{v^n}^{OCBA} \cdot \sqrt{\frac{2\log N(v_p^n)}{N(v^n)}}. \tag{11}$$

### 4.2 The Procedure

Section 4.1 shows that the non-myopic KG policy can be computed efficiently via a modified MCTS method. Given this, now we are ready to propose the sequential R&S procedure, which iteratively implements the non-myopic policy at each step and takes samples. Notice that the non-myopic policy, as a multi-step lookahead policy, is often constructed with a prescribed length of lookahead steps, $t$. To avoid unnecessary lookahead steps, we slightly change the setting of lookahead steps for the last $t$ steps. Particularly, for each step $N-t \leq n < N$, we choose the length of lookahead steps as the number of remaining steps, i.e., $N-n$. As a consequence, our sequential R&S procedure keeps using the non-myopic KG policy $R_t^n$ with $t$-step lookahead at each step $n < N-t$, and then turns to use the non-myopic KG policy $R_{N-n}^n$ with $(N-n)$-step lookahead for each remaining step $N-t \leq n < N$. Or equivalently, our sequential R&S procedure essentially refers to a sequence of decisions, i.e.,

$$\pi_t = \left(R_t^0, R_t^1, \ldots, R_t^{N-t-1}, R_t^{N-t}, R_{t-1}^{N-t+1}, \ldots, R_1^{N-1}\right).$$

In what follows, we propose our new sequential R&S procedure that is design based on the non-myopic KG policy, as shown in Algorithm 1. This procedure is initialized by an initial state $s_0 = (\mu_0, \beta_0)$ at step 0, the total simulation budget $N$, the length of lookahead steps $t$ and the simulation budget $T$ used in the MCTS. After the simulation budget $N$ is exhausted, the alternative with the largest sample mean is selected as the best. Moreover, to facilitate the understanding, we extract the tree search process specifically as Algorithm 2. It follows the four steps in Figure 2 iteratively and returns the alternative selected by non-myopic KG policy at each step.

**Algorithm 1:** Non-myopic KG Procedure

**Require:** state $s_0$, Simulation budget $N$ for R&S, length of lookahead steps $t$, simulation budget $T$ in MCTS
**Ensure:** $\arg\max_x \mu_x^N$
  Set $n = 0$, $s = s_0$
  **while** $n < N$ **do**
    $x^n$ = Tree_Search$(s, \min(t, N-n), T)$.
    $s = s^{n+1}|s, x^n$
    $n = n+1$
  **end while**
  $x^N = \arg\max_x \mu_x^N$

Intuitively, a better approximation for the value function is supposed to yield a superior sampling policy. However, this is generally not guaranteed for general DP problems. Luckily, by the special structure of R&S problems, we are able to show in the following theorem that, starting from any step $n$ with any state $S^n = s$, Algorithm 1 achieves a better performance at the terminal than the myopic KG policy.

**Algorithm 2:** Tree_Search ($s_n$, $t$, $T$)

**Require:** state $s_n$, length of lookahead steps $t$, simulation budget $T$ in MCTS
**Ensure:** $\arg\max_{v^n \in \{\text{children of root}\}} \widetilde{UCT}(v^n)$
  Set $j = 0$, create root node $v^n$ from state $s_n$
  **while** $j \leq T_N$ **do**
    **while** the depth of $v^n$ less than t **do**
      **if** $v^n$ is not fully expanded **then**
        $v^n$ = expand($v^n$)
      **else**
        $v^n = \arg\max_{\hat{v}^n \in \{\text{children of } v^n\}} \widetilde{UCT}(\hat{v}^n)$
      **end if**
    **end while**
    Rollout with KG as the base policy until $N$
    $\Delta = \max_x \mu_x^N - \max_x \mu_x^n$
    Backup($v^n, \Delta$)
    $j = j + 1$
  **end while**
  return $\arg\max_{v^n \in \{\text{children of root}\}} \widetilde{UCT}(v^n)$

**Theorem 4.1** For every state $s \in \mathbb{S}$, any $t \geq 1$, and any step $0 \leq n < N$, we have that $V^{n,\pi_t}(s) \geq V^{n,KG}(s)$.

## 5 EXPERIMENTS

We follow the experimental setting of Frazier et al. (2008) and test our non-myopic KG policy against other competing policies on 100 randomly generated problems. The numerical results show that our non-myopic KG policy has competitive performance. When measured by the average performance of all problems, non-myopic KG policy significantly outperforms the other policies. In particular, it works exceptionally well when the simulation budget is small with respect to the number of alternatives.

The random problem is initialized by the total simulation budget $N$, the number of alternatives $K$ and a common initial state $S^0 = (\mu_0, \beta_0)$ across all alternatives. These parameters are chosen as follows: $K$ is an integer randomly chosen between 2 and 100; $N$ is determined with $N/K$ drawn uniformly from the set $\{1, 3, 10\}$. Besides, each $\mu_x$ is uniform distributed at interval $[-1, 1]$, $\beta_x$ is set as 1 with probability 0.9 and 1000 with probability 0.1. We compare our non-myopic KG policy against three policies: myopic KG policy (Frazier et al. 2008), multi-step policy (as defined in Section 3.1) and myopic KG policy with rollout (as defined in Section 3.2) . On each of the 100 randomly generated problems, we simulate each policy 100 times and record the true mean of selected alternative at terminal. For our non-myopic KG policy, we set the simulation budget $T$ in MCTS as 500 and set the depth $t$ of MCTS as 2.

The average mean values of all problems for each policy are shown in Table 1. The table illustrates that all the three lookahead policies significantly outperform the myopic KG policy in terms of the average mean values of final selection. Among them, the non-myopic KG policy performs the best, almost achieving 250% of the average value of KG. The rollout policy and the multistep policy have close performances, with rollout policy being slightly better. This might be due to that the rollout policy considers the impact of all future samples while the multi-step policy is only two-step lookahead. In addition, we show the sample

Table 1: Average values for all problems.

| Policy | Non-myopic KG | Multistep | Myopic with rollout | KG |
|---|---|---|---|---|
| Value | 0.206 | 0.189 | 0.191 | 0.079 |

estimates of the selected value difference $V(\pi) - V(KG)$ aggregated across the 100 randomly generated problems in Figure 3, where $\pi$ are the three lookahead policies and the difference $V(\pi) - V(KG)$ on any particular problem is estimated as the difference in selected mean value. Bars to the right of 0 indicate that $\pi$ outperforms KG policy on those problems, and bars to the left of 0 indicate the converse. The

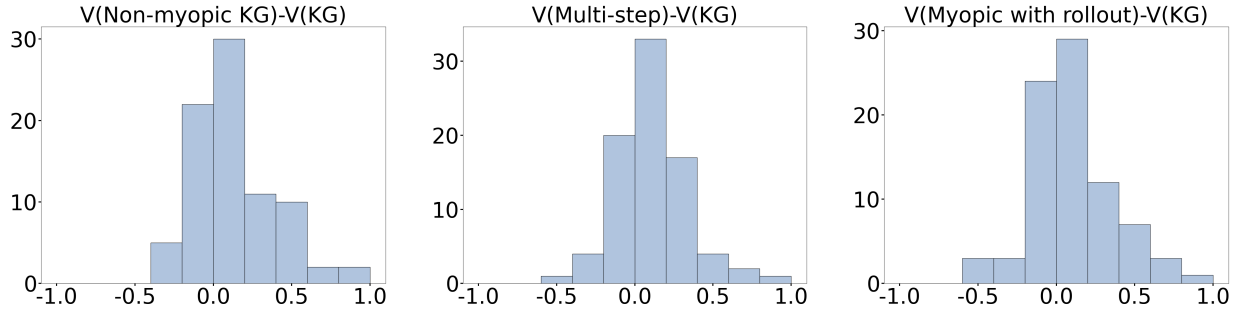


Figure 3: Histogram of the sampled difference in value for KG policy versus lookahead policies across the 100 randomly generated problems.

histograms show that non-myopic KG policy, multi-step policy and myopic policy with rollout all have better performance than KG as the number of cases to the right of 0 for lookahead policies is significantly larger than to the left. The better performance of non-myopic KG might benefit from the MCTS structure, which introduces randomness to the process instead of calculating a closed form as KG does. In the early stage, introducing greater randomness to the decision process is similar to preferring *exploration* to *exploitation*, which is a rather good choice since we do not know much about the alternatives. Besides,

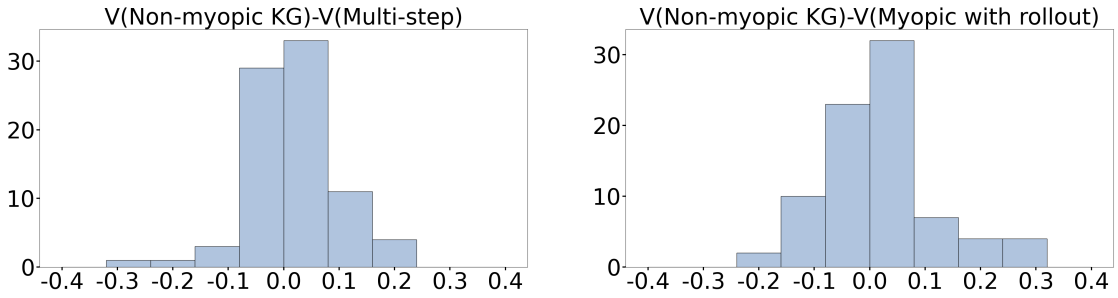


Figure 4: Histogram of the sampled difference in value for non-myopic KG policy versus other lookahead policies across the 100 randomly generated problems.

we compare the performance of non-myopic KG policy against the multi-step policy and myopic policy with rollout, as shown in Figure 4. It can be seen from the results that the amount by which non-myopic KG policy outperforms the other two policies is larger than the amount by which it is outperformed. This justifies the improvement by combining two lookahead stategies.

Moreover, we have tested the parameter settings for lookahead steps, rollout steps and MCTS simulation times. First, for lookahead steps, the numerical results reveal that the increment of lookahead steps has a diminishing marginal benefit, which suggests the largest gain is obtained from $t = 1$ to $t = 2$ ($t$ is the depth of MCTS). After two-step lookahead, the increment in depth will lead to exponential increase in search space, but brings minimal performance improvement, as shown on the left of Figure 5, which plots the histogram of sampled difference for $t = 3$ versus $t = 2$. Thus, in the above experiments, we only discuss the case of $t = 2$. Second, for the rollout steps, we have tried full rollout (until $N$) and truncated rollout (until $\log N$ ). The performance of non-myopic KG policy with full rollout is quite close to that with truncated

rollout, as shown in on the right of Figure 5. This phenomenon implies that the increment of rollout steps might also have a diminishing marginal benefit, so the truncated rollout could already reach a good performance. Moreover, we find that the modifications of UCT algorithm could significantly accelerate the MCTS process. The simulation times required to achieve convergence for a random problem in the search tree would be reduced by 70% by using the modified UCT algorithm. The above discussion about parameter settings suggests that, although it is quite time-consuming to develop a multi-step MCTS with full rollout and UCT algorithm, we could just reduce the complexity by setting the lookahead steps as 2 with modified UCT and implementing truncated rollout in practice, which also achieves a good performance.

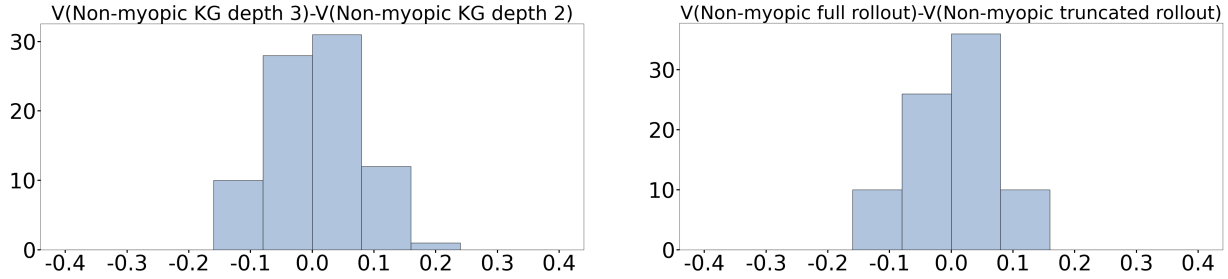


Figure 5: Histogram of the sampled difference in value for non-myopic KG policy with different parameters.

## 6 CONCLUSION

In this paper, we address the fixed-budget R&S problem through a DP perspective and propose a non-myopic KG procedure. We show that the non-myopic KG policy is superior than the classic myopic KG policy in terms of both value function approximation and policy performance, especially when the total sampling budget is small. Beyond the classic KG policy, our non-myopic structure is quite universal, and may be extended to other myopic policies like Greedy, Expected Value of Information and so on.

## REFERENCES

Auer, P., N. Cesa-Bianchi, and P. Fischer. 2002. "Finite-time analysis of the multiarmed bandit problem". *Machine learning* 47(2):235–256.

Bertsekas, D. 2012. *Dynamic programming and optimal control: Volume I*, Volume 1. Athena scientific.

Browne, C. B., E. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton. 2012. "A survey of monte carlo tree search methods". *IEEE Transactions on Computational Intelligence and AI in games* 4(1):1–43.

Chen, C.-H., J. Lin, E. Yücesan, and S. E. Chick. 2000. "Simulation budget allocation for further enhancing the efficiency of ordinal optimization". *Discrete Event Dynamic Systems* 10(3):251–270.

Chick, S. E., and K. Inoue. 2001. "New two-stage and sequential procedures for selecting the best simulated system". *Operations Research* 49(5):732–743.

Coulom, R. 2007. "Computing "elo ratings" of move patterns in the game of go". *ICGA journal* 30(4):198–208.

Frazier, P. I., W. B. Powell, and S. Dayanik. 2008. "A knowledge-gradient policy for sequential information collection". *SIAM Journal on Control and Optimization* 47(5):2410–2439.

Gonzalez, J., M. Osborne, and N. Lawrence. 2016. "GLASSES: Relieving The Myopia Of Bayesian Optimisation". In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*. May 9$^{th}$-11$^{th}$, Cadiz, Spain, 790–799.

Hong, L. J., W. Fan, and J. Luo. 2021. "Review on ranking and selection: A new perspective". *Frontiers of Engineering Management* 8(3):321–343.

Kocsis, L., and C. Szepesvári. 2006. "Bandit Based Monte-Carlo Planning". In *European conference on machine learning*, edited by J. Fürnkranz, T. Scheffer, and M. Spiliopoulou, 282–293. Berlin, Heidelberg: Springer.

Lam, R., K. Willcox, and D. H. Wolpert. 2016. "Bayesian Optimization with a Finite Budget: An Approximate Dynamic Programming Approach". In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, edited by D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, 883891. Barcelona, Spain: Curran Associates Inc.

Rosin, C. D. 2011. "Multi-armed bandits with episode context". *Annals of Mathematics and Artificial Intelligence* 61(3):203–230.

Silver, D., A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot et al. 2016. "Mastering the game of Go with deep neural networks and tree search". *nature* 529(7587):484–489.

Sutton, R. S., and A. G. Barto. 2018. *Reinforcement learning: An introduction*. MIT press.

Yue, X., and R. A. Kontar. 2020. "Why Non-myopic Bayesian Optimization is Promising and How Far Should We Look-ahead? A Study via Rollout". In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, edited by S. Chiappa and R. Calandra, 2808–2818. Online: Proceedings of Machine Learning Research.

## AUTHOR BIOGRAPHIES

**KEXIN QIN** is a master student in the School of Data Science at Fudan University in Shanghai, China. Her research interests include simulation optimization, stochastic modeling, and machine learning methods with applications to risk management. Her email address is 21210980117@m.fudan.edu.cn.

**WEIWEI FAN** is the associated professor at Advanced Institute of Business from Tongji University in Shanghai, China. Her research interests include simulation optimization, robust optimization and their applications in healthcare management and supply chain management. Her email address is wfan@tongji.edu.cn

**L. JEFF HONG** is the Fudan Distinguished Professor and Hongyi Chair Professor with joint appointment at School of Management and School of Data Science at Fudan University in Shanghai, China. His research interests include stochastic simulation, stochastic optimization, risk management and supply chain management. He is currently the simulation area editor of Operations Research, an associate editor of Management Science and the President of INFORMS Simulation Society. His email address is hong_liu@fudan.edu.cn.