Information collection on a graph

Ilya O. Ryzhov   Warren Powell

February 10, 2010

Abstract

We derive a knowledge gradient policy for an optimal learning problem on a graph, in which we use sequential measurements to refine Bayesian estimates of individual edge values in order to learn about the best path. This problem differs from traditional ranking and selection, in that the implementation decision (the path we choose) is distinct from the measurement decision (the edge we measure). Our decision rule is easy to compute, and performs competitively against other learning policies, including a Monte Carlo adaptation of the knowledge gradient policy for ranking and selection.

1   Introduction

Consider a path-finding problem on a graph in which the lengths or values of the edges are random, and their distributions are unknown. We begin with independent normal Bayesian priors for the edge values, and we can obtain noisy measurements of the values which we can use to refine our estimates through Bayesian updating. We are allowed to make N measurements of individual edges, and we can measure any edge at any time, regardless of its location in the graph. After the measurements are complete, we must make a guess as to the best path. Our problem is to sequentially determine the best edges to evaluate, where we can make each choice given what we have learned from prior measurements.

This problem contains an important distinction between measurement and implementation decisions. While we measure, we collect information about individual *edges*. However, our overarching goal is to find a *path*. We must choose the edges we measure in such a way as to collect the most information about the graph as a whole. We are not constrained by the graph structure when choosing what to measure, in the sense that we can always measure any edge at any time. Nonetheless, we must still keep the graph structure in mind when choosing edges, because it is relevant to the final implementation decision.

1

The distinction between measurement and implementation has not been considered in earlier work on optimal learning. A major goal of this work is to open up new avenues for optimal learning in the context of operations research problems (e.g. on graphs). Three possible examples of graph problems where a learning component may come into play are the following:

1. *PERT/CPM project management.* A complex project can be represented as a graph in which edges correspond to tasks. Suppose that there are multiple sequences of tasks that will enable us to complete the project objectives; every such sequence is a particular path in the graph. We wish to find the sequence that can be completed in the shortest possible time. We can change our estimate of the time required to complete a task by analyzing historical data from previous projects involving that task. We do not have time to analyze all available records (they may be expensive to access), and can only perform a small number of historical studies.

2. *Biosurveillance.* We are planning a route for a single medical specialist through a region in a developing country. The route should maximize the specialist's total effectiveness in the region. Before committing to a route, we can make contact with hospitals in the region and ask for recent medical data that could change our beliefs about the specialist's potential effectiveness there. A hospital can be modeled as a pair of nodes connected by a single edge, where the value of the edge is a measure of the specialist's effectiveness. Each contact requires money and time to analyze the data, so the total number of hospitals we can visit is limited. Thus, the goal is to find the path with the highest value given a fixed number of edges.

3. *Defense of intellectual property.* Certain stores may be unwittingly selling counterfeit products such as printer ink. The ink manufacturer has an estimate of how much counterfeit ink is sold in each store, and wishes to plan a route for a detective to investigate a number of the stores. The estimates can be improved by ordering samples of ink from individual stores. This incurs inventory, transportation and storage costs, so the number of orders is limited. Again, we want to maximize the value of a path with a fixed number of edges.

Optimal information collection has a long history in the context of simple problems such as multi-armed bandits (see e.g. Gittins (1989)) and ranking and selection. A general overview of ranking and selection can be found in Bechhofer et al. (1995) and Kim & Nelson (2006), while Law & Kelton (1991) and Goldsman (1983) provide a simulation-oriented perspective. In these problems, there is a finite set of alternatives with unknown values, and the goal is to find the highest value.

We can improve our estimates of the values by sequentially measuring different alternatives. In the problem of learning on a graph, we also have a finite set of edges that can be measured, but we are not simply looking for the best edge. We learn by measuring individual edges, but we use the information we collect to improve our ability to find a path.

Stochastic shortest path problems have also been widely studied. An overview is available in Snyder & Steele (1995). However, many of these studies assume that the edge values have known distributions, for example the exponential distribution (Kulkarni (1986), Peer & Sharma (2007)). The work by Frieze & Grimmett (1985) describes a probabilistic shortest-path algorithm for more general classes of non-negative distributions, and analyzes the length of the shortest path in the special case of uniformly distributed edge values. Correlations among the edge values have also been studied by Fan et al. (2005), again with the assumption of known distributions. For online graph problems, in which we learn in the process of traversing the graph, methods such as Q-learning by Watkins & Dayan (1992) use stochastic approximation to estimate unknown values, while Bayesian approaches have been proposed by Dearden et al. (1998) and Duff & Barto (1996).

We build on a class of approximate policies originally developed for ranking and selection, where each measurement maximizes the value of information that can be collected in a single time step. This technique was first proposed by Gupta & Miescke (1996) for ranking and selection with independent Gaussian priors, and subsequently expanded in the work on Value of Information Procedures (VIP) by Chick & Inoue (2001a) and Chick & Inoue (2001b). Additional theoretical properties were established by Frazier et al. (2008) for the knowledge gradient (KG) policy. A KG-like methodology was also applied to other learning problems: by Chick et al. (2009), for ranking and selection with unknown measurement noise; by Frazier et al. (2009), for ranking and selection with correlated Gaussian priors; and by Ryzhov et al. (2009) and Ryzhov & Powell (2009), for the online multi-armed bandit problem.

In addition to their theoretical properties, KG-type policies have been shown to perform well experimentally. In the offline setting, thorough empirical studies were performed by Inoue et al. (1999) and Branke et al. (2007). In the online case, the variant of KG studied in Ryzhov et al. (2009) performs competitively even against the known, optimal Gittins policy for multi-armed bandits, while being much easier to compute than Gittins indices. These features make KG policies attractive for information collection problems.

This paper makes the following contributions: 1) We present a new class of optimal learning

problems beyond the scope of the literature on ranking and selection and multi-armed bandits. In this problem class, our goal is to solve an optimization problem on a graph with unknown edge values. We can improve our estimate of the optimal solution by making sequential measurements of individual edges. 2) We show that the knowledge gradient concept can be applied to this problem class, while retaining its theoretical and computational advantages. 3) We propose an alternate learning policy that treats the problem as a ranking and selection problem, using Monte Carlo sampling to avoid having to enumerate all paths. 4) We conduct an experimental study comparing these and other learning policies on a diverse set of graph topologies. The study indicates that the KG policy is effective for graphs where there are many paths that could potentially be the best, and the Monte Carlo policy is effective when we are allowed to make a large number of measurements.

Section 2 lays out a mathematical model for information collection on a graph. In Section 3, we derive the exact KG decision rule for an acyclic graph problem, and approximate it for general graphs. We also show that the KG policy is asymptotically optimal as the number of measurements becomes large. In Section 4, we give a decision rule for the Monte Carlo KG policy. Finally, we present numerical results comparing the performance of KG to existing learning policies.

2 Mathematical model

Consider a graph described by a finite set S of nodes and a set $E \subseteq S \times S$ of directed edges. Every edge $(i, j) \in E$ has a value μ_{ij}. For notational simplicity, and without loss of generality, we assume that every path must start at some fixed origin node $a \in S$, and that every path must contain exactly T edges. We wish to find the path with the largest total value

$$\max_p \sum_{(i,j) \in E} \delta_{ij}^p \mu_{ij} \tag{1}$$

where p denotes a path that starts at a and contains T edges, and δ_{ij}^p is an indicator function that equals 1 if the edge (i, j) appears in the path p, and zero otherwise. Throughout our analysis, we assume that the graph is acyclic, so any edge can appear at most once in a given path.

The best path can be found using Bellman's equation for dynamic programming:

$$V_t(i) = \max_j \mu_{ij} + V_{t+1}(j), \tag{2}$$

$$V_T(i) = 0. \tag{3}$$

These quantities are defined for each $i \in S$ and each $t = 0, ..., T$. Thus, $V_t(i)$ is the length of the best path that starts at node i and contains $T - t$ edges. It follows that $V_0(a)$ is the optimal value of the problem (1). The actual edges that make up the best path can be found by keeping track of the nodes j that achieve the maximum in (2) for each i.

If the values μ_{ij} are known, (2) gives us the exact optimal solution to (1). If the values are random with known distribution, (2) still solves the problem in the sense that it gives us the path with the highest expected total value. However, in our work, the distributions of the values are unknown, and our beliefs about them change as we learn more about them.

Because of this uncertainty, the problem consists of two phases. In the first phase, we will make sequential measurements of individual edges, in order to improve our estimate of the solution to (2). This is called the *learning phase* of the problem. After the measurements have been completed, we will enter the *implementation phase*, where we will choose the path that we think is the best, based on all the measurements we made in the learning phase. There is a clear distinction between these two stages. When we learn, we make measurements of individual edges; when we implement, we choose a path. In this section, we will describe the dynamics of the learning phase and the way in which our beliefs change when we measure an edge. The issue of how to choose which edge to measure will be discussed later in Section 3.

2.1 Learning about individual edges

Suppose that the mean values μ_{ij} are unknown, but we can estimate them by measuring individual edges. When we choose to measure edge $(i, j) \in E$, we observe a random value $\hat{\mu}_{ij}$, which follows a Gaussian distribution with mean μ_{ij} and variance σ_ε^2. We assume that the measurement error σ_ε^2 is known, and we sometimes use the notation $\beta_\varepsilon = \sigma_\varepsilon^{-2}$ to refer to the measurement precision. Because μ_{ij} is itself unknown, we assume that $\mu_{ij} \sim \mathcal{N}\left(\mu_{ij}^0, \left(\sigma_{ij}^0\right)^2\right)$, where μ_{ij}^0 and σ_{ij}^0 represent our prior beliefs about μ_{ij}. We also assume that the values of the edges are mutually independent, conditioned on μ_{ij}, $(i, j) \in E$.

We learn about the graph by making N sequential measurements, where N is given. One measurement corresponds to exactly one edge. Any edge can be measured at any time, regardless of graph structure. Let $\mathcal{F}^n$ be the sigma-algebra generated by our choices of the first n edges, as well as the observations we made on those edges. We say that something happens "at time n" if it

happens immediately after we have made exactly n measurements. Then we can define

$$\mu_{ij}^n = \mathbb{E}^n \left(\mu_{ij} \right)$$

where $\mathbb{E}^n \left(\cdot \right) = \mathbb{E} \left(\cdot | \mathcal{F}^n \right)$. Similarly, we let $\left(\sigma_{ij}^n \right)^2$ be the conditional variance of μ_{ij} given $\mathcal{F}^n$, with $\beta_{ij}^n = \left(\sigma_{ij}^n \right)^{-2}$ being the conditional precision. Thus, at time n, we believe that $\mu_{ij} \sim \mathcal{N} \left(\mu_{ij}^n, \left(\sigma_{ij}^n \right)^2 \right)$. Our beliefs evolve according to the Bayesian updating equation

$$\mu_{ij}^{n+1} = \begin{cases} \frac{\beta_{ij}^n \mu_{ij}^n + \beta_\varepsilon \hat{\mu}_{ij}^{n+1}}{\beta_{ij}^n + \beta_\varepsilon} & \text{if } (i,j) \text{ is the } (n+1)\text{st edge measured} \\ \mu_{ij}^n & \text{otherwise.} \end{cases} \tag{4}$$

The values of the edges are independent, so we update only our beliefs about the edge that we have just measured. The quantity $\hat{\mu}_{ij}^{n+1}$ is the random value observed by making that measurement. The precision of our beliefs is updated using the equation

$$\beta_{ij}^{n+1} = \begin{cases} \beta_{ij}^n + \beta_\varepsilon & \text{if } (i,j) \text{ is the } (n+1)\text{st edge measured} \\ \beta_{ij}^n & \text{otherwise.} \end{cases} \tag{5}$$

We use the notation $\mu^n = \left\{ \mu_{ij}^n | (i,j) \in E \right\}$ and $\beta^n = \left\{ \beta_{ij}^n | (i,j) \in E \right\}$. We also let

$$\left(\tilde{\sigma}_{ij}^n \right)^2 = Var \left(\mu_{ij}^{n+1} | \mathcal{F}^n \right) = Var \left(\mu_{ij}^{n+1} | \mathcal{F}^n \right) - Var \left(\mu_{ij}^n | \mathcal{F}^n \right) \tag{6}$$

be the reduction in the variance of our beliefs about (i,j) that we achieve by measuring (i,j) at time n. It can be shown that

$$\tilde{\sigma}_{ij}^n = \sqrt{ \left(\sigma_{ij}^n \right)^2 - \left(\sigma_{ij}^{n+1} \right)^2 } = \sqrt{ \frac{1}{\beta_{ij}^n} - \frac{1}{\beta_{ij}^n + \beta_\varepsilon} }.$$

It is known (for instance, from DeGroot (1970)), that the conditional distribution of μ_{ij}^{n+1} given $\mathcal{F}^n$ is $\mathcal{N} \left(\mu_{ij}^n, \left(\tilde{\sigma}_{ij}^n \right)^2 \right)$. In other words, given $\mathcal{F}^n$, we can write

$$\mu_{ij}^{n+1} = \mu_{ij}^n + \tilde{\sigma}_{ij}^n \cdot Z \tag{7}$$

where Z is a standard Gaussian random variable. It follows that $\mathbb{E}^n \left(\mu_{ij}^{n+1} \right) = \mu_{ij}^n$.

Our beliefs about the values after n measurements are completely characterized by μ^n and β^n. We can define a *knowledge state*

$$s^n = (\mu^n, \beta^n)$$

to completely capture all the information we have at time n. If we choose to measure edge $(i, j) \in E$ at time n, we write

$$s^{n+1} = K^M \left(s^n, (i, j), \hat{\mu}_{ij}^{n+1} \right)$$

where the transition function K^M is described by (4) and (5).

To streamline our presentation, the measurement error σ_ε^2 is taken to be constant for all edges, similar to Frazier et al. (2008). However, we can allow the measurement error to be edge-dependent without significant changes in our analysis. If we suppose that $\hat{\mu}_{ij}^{n+1} \sim \mathcal{N} \left(\mu_{ij}, \lambda_{ij}^2 \right)$, we obtain the same model, but with σ_ε^2 and β_ε replaced by λ_{ij}^2 and λ_{ij}^{-2} in (4), (5) and (6). Except for the modifications in these equations, all theoretical and computational results presented in this paper remain unchanged in the case where the measurement error varies across edges.

The validity of our assumption of Gaussian priors and measurements is problem-dependent. If the measurement is done through statistical sampling with a large enough sample size, the Gaussian distribution is a good approximation. The method of batch means (see e.g. Schmeiser (1982), Kim & Nelson (2007)) can be used to design the observations to mitigate the non-normality of the underlying data. Additionally, Hoff (2009) states, based on a result by Lukacs (1942), that a Gaussian sampling model can be used if we believe the sample mean to be independent from the sample variance (in particular, if the sample variance is known).

A Gaussian prior may work well even when the measurements are non-Gaussian. Gelman et al. (2004) suggests that a unimodal and "roughly symmetric" posterior can be approximated by a Gaussian distribution. Under certain conditions, the posterior is asymptotically normal as the number of measurements becomes large (see Bernardo & Smith (1994)). In short, a Gaussian sampling model is appropriate for many learning problems.

2.2 Estimating the length of the best path with dynamic programming

At time n, our beliefs about the path that solves (2) are expressed using Bellman's equation, with the unknown values μ replaced by the most recent beliefs μ^n:

$$V_t^n (i; s^n) = \max_j \mu_{ij}^n + V_{t+1}^n (j; s^n), \tag{8}$$

$$V_T^n (i; s^n) = 0. \tag{9}$$

As with (2), we compute V_t^n for all i and t, from which we can construct the path that we believe to be the best at time n. It is important to understand the distinction between (2) and (8). The quantity $V_0(a)$ represents the true length of the true best path. The quantity $V_0^n(a; s^n)$ represents our time-n beliefs about which path is the best, and thus depends on s^n. The path that solves (8) is our best time-n guess of the path that solves (2).

Intuitively, the solution to (8) should be worse than the expected solution to (2). In other words, there is a penalty for not having perfect information. The following proposition formalizes this idea. The proof uses an induction argument, and can be found in the Appendix.

Proposition 2.1. *For all $i \in S$, for all $t = 0, ..., T$, and for all knowledge states s^n,*

$$V_t^n(i; s^n) \leq \mathbb{E}^n V_t(i) \qquad \text{almost surely.} \tag{10}$$

We can also make a time-n estimate of the length of a fixed path p:

$$V_t^{p,n}(i; s^n) = \mu_{ij}^n + V_{t+1}^{p,n}(j; s^n), \qquad \text{where } j = x^p(i),$$
$$V_T^{p,n}(i; s^n) = 0.$$

Here, $x^p(i)$ denotes the node that follows node i in path p. The true length of path p is given by

$$V_t^p(i) = \mu_{ij} + V_{t+1}^p(j), \qquad \text{where } j = x^p(i),$$
$$V_T^p(i) = 0.$$

From these equations, it is clear that $\mathbb{E}^n V_t^p(i) = V_t^{p,n}(i; s^n)$ for fixed p.

Our use of the index t is a technical convention of dynamic programming. Bellman's equation constructs the best path one edge at a time, and the index t merely serves to indicate how many edges in the path have already been built. It does not have any bearing on how many edges we have measured in the learning problem. For convenience, we will use the notation

$$V^n(s^n) = V_0^n(a; s^n),$$
$$V^{p,n}(s^n) = V_0^{p,n}(a; s^n),$$

to refer to our time-n estimates, dropping the index t. Similarly, we use V and V^p to denote $V_0(a)$ and $V_0^p(a)$.

We can now describe our objective function. In the learning phase of our problem, we will choose a policy π that selects an edge for us to measure in every time step. In the second phase, we will simply solve (8) using our final estimates μ^N to find the path that seems to be the best, based on all the measurements we made in the first phase. This is the intuitive choice of implementation decision. If we cannot make any more measurements, the best we can do is to solve Bellman's equation using the information we have accumulated.

The measurement policy π can be viewed as a collection of decision rules $X^{\pi,0}, ..., X^{\pi,N-1}$, where each $X^{\pi,n}$ is a function mapping the knowledge state s^n to an edge in E. The time-n decision rule uses the most recent knowledge state s^n to make a decision. The main challenge in our problem is to choose a measurement policy π for selecting individual edges in the first phase, and our objective function can be written as

$$\sup_{\pi} \mathbb{E}^{\pi} V^N \left(s^N \right). \tag{11}$$

Our implementation decision is thus fixed in the objective function. We choose measurements in the learning phase to maximize the expected value of the path that we will choose in the implementation phase.

Remark 2.1. *By taking an expectation of both sides of (10), we find that, for any policy π, $\mathbb{E}^{\pi} V^N \left(s^N \right) \leq \mathbb{E}^{\pi} V$, where $V = V_0 \left(a \right)$ is the true length of the path that solves (2). Since the true edge values μ do not depend on the policy π, it follows that $\mathbb{E}^{\pi} V = \mathbb{E} V$ for any π, hence*

$$\mathbb{E}^{\pi} V^N \left(s^N \right) \leq \mathbb{E} V$$

for all π. Thus, Proposition 2.1 gives us a global upper bound on the objective value achieved by any measurement policy.

Note that we use a time-staged graph model, where we are always looking for a path with T edges. This is convenient for modeling, because it enables us to easily write the solution to the path problem using Bellman's equation. However, the KG policy that we derive in Section 3 does not require a time-staged graph, and can be used for many different path problems. For example, if our graph has both a source and a destination node, we would simply let $V^n \left(s^n \right)$ be the time-n estimate of the best path from the source to destination. We are also not bound to the maximization problem in (1). For a shortest-path problem, the derivation in Section 3 will be identical, except $V^n \left(s^n \right)$ will be obtained using a shortest-path algorithm. In fact, our computational study in Section 5 solves shortest-path problems on graphs with sources and sinks.

3 The knowledge gradient policy

The remainder of this paper will discuss the problem of how to choose the measurement policy π in (11). Finding the optimal measurement policy is an intractable problem, but we propose a heuristic policy called the *knowledge gradient* policy, which yields a computable algorithm.

Suppose that we are at time n, in knowledge state s^n. Let p^n be the path that achieves $V^n(s^n)$. Thus, p^n is the path that we believe is the best, given our most recent information, and $V^n(s^n)$ is our estimate of its length. The knowledge gradient policy is based on the idea first developed by Gupta & Miescke (1996) and later studied by Chick & Inoue (2001a), Chick & Inoue (2001b) and Frazier et al. (2008) for the ranking and selection problem. This idea can be stated as "choosing the measurement that would be optimal if it were the last measurement we were allowed to make." If we are at time $N-1$, with only one more chance to measure, the best choice is given by

$$\arg \max_{(i,j)\in E} \mathbb{E}_{ij}^{N-1} V^N\left(s^N\right) = \arg \max_{(i,j)\in E} \mathbb{E}_{ij}^{N-1}\left(V^N\left(s^N\right) - V^{N-1}\left(s^{N-1}\right)\right)$$

where $\mathbb{E}_{ij}^{N-1}$ observes all the information known at time $N-1$, as well as the choice to measure (i,j) at time $N-1$. We bring $V^{N-1}\left(s^{N-1}\right)$ into the maximum because this quantity is known at time n, and does not depend on the choice of measurement.

If we always assume that we have only one more chance to measure, at every time step, then the decision rule that follows from this assumption is

$$X^{KG,n}\left(s^n\right) = \arg \max_{(i,j)\in E} \mathbb{E}_{ij}^n\left(V^{n+1}\left(s^{n+1}\right) - V^n\left(s^n\right)\right). \tag{12}$$

In words, we measure the edge that maximizes the expected improvement in our estimate of the length of the best path that can be obtained from a single measurement. The term "knowledge gradient" is due to (12) being written as a difference.

Remark 3.1. *By definition, the KG policy is optimal for $N = 1$. In this case, a measurement policy consists of only one measurement, and (11) becomes*

$$\max_{(i,j)\in E} \mathbb{E}_{ij}^0 V^1\left(s^1\right).$$

Below, we find the value of a single measurement, and present the knowledge gradient policy.

3.1 The effect of one measurement

In order to compute the right-hand side of (12), we consider the effects of measuring one edge on our beliefs. Fix an edge $(i,j) \in E$ and let $A_{ij} = \left\{ p : \delta_{ij}^p = 1 \right\}$ be the set of all paths containing (i,j). Then A_{ij}^c is the set of all paths not containing that edge. Now define a path p_{ij}^n as follows. If $p^n \in A_{ij}$, let

$$p_{ij}^n = \arg \max_{p \in A_{ij}^c} V^{p,n} (s^n) .$$

On the other hand, if $p^n \in A_{ij}^c$, let

$$p_{ij}^n = \arg \max_{p \in A_{ij}} V^{p,n} (s^n) .$$

Thus, if (i,j) is already in the best time-n path, then p_{ij}^n is the best path that does not contain this edge. If (i,j) is not part of the path we believe to be the best, then p_{ij}^n is the best path that does contain that edge. Thus, by definition, $p_{ij}^n \neq p^n$.

Proposition 3.1. *If we measure edge (i,j) at time n, the path that achieves $V^{n+1} \left(s^{n+1} \right)$ will be either p^n or p_{ij}^n.*

Proof: Suppose that $p^n \in A_{ij}$. By definition, $p^n = \arg \max_p V^{p,n} (s^n)$, so in particular

$$p^n = \arg \max_{p \in A_{ij}} V^{p,n} (s^n) .$$

Depending on the outcome of our measurement of (i,j), our beliefs about all paths in A_{ij} will change, but they will all change by the same amount $\mu_{ij}^{n+1} - \mu_{ij}^n$. This is because we assume that the graph contains no cycles, so all paths in A_{ij} contain only one copy of (i,j). Therefore,

$$p^n = \arg \max_{p \in A_{ij}} V^{p,n+1} \left(s^{n+1} \right)$$

for every outcome. Thus, p^n is the only path in A_{ij} that can be the best time-$(n+1)$ path. Our beliefs about the paths in A_{ij}^c will remain the same, because none of those paths contain (i,j), and our beliefs about the other edges do not change as a result of measuring (i,j). Therefore,

$$\arg \max_{p \in A_{ij}^c} V^{p,n+1} \left(s^{n+1} \right) = \arg \max_{p \in A_{ij}^c} V^{p,n} (s^n) = p_{ij}^n$$

for every outcome. Thus, p_{ij}^n is the only path in A_{ij}^c that can be the best time-$(n+1)$ path. It follows that p^n and p_{ij}^n are the only two paths that can be the best at time $n+1$.

If $p^n \in A_{ij}^c$, the argument is the same. By definition, p^n is the best path, so

$$p^n = \arg \max_{p \in A_{ij}^c} V^{p,n}(s^n).$$

Our beliefs about the paths in A_{ij}^c do not change after measuring (i,j), so p^n will still be the best path in A_{ij}^c at time $n+1$. Our beliefs about all paths in A_{ij} will change by the same amount after the measurement, so p_{ij}^n will still be the best path in A_{ij} at time $n+1$. Therefore, p^n and p_{ij}^n are again the only two paths that can be the best at time $n+1$. $\qquad\square$

Because p^n and p_{ij}^n figure prominently in the KG policy, we must remark on their computation. We can obtain p^n via (8). If $p^n \in A_{ij}$, then p_{ij}^n can be found by solving a modified version of (8) with μ_{ij}^n set to $-\infty$. This ensures that we obtain a path in A_{ij}^c. If $p^n \notin A_{ij}^c$, we can again solve a modified version of (8) with μ_{ij}^n chosen to be some large number, for instance the sum of the other μ^n values. This will construct a path that includes (i,j), with the other edges chosen optimally.

3.2 Computation of the KG policy

Define a function $f(z) = z\Phi(z) + \phi(z)$, where ϕ and Φ are the standard Gaussian pdf and cdf, respectively. Also, for notational convenience, we define $V_{ij}^n(s^n) = V^{p_{ij}^n,n}(s^n)$. This quantity is our time-n estimate of the length of the path p_{ij}^n defined in Section 3.1. With these definitions, we can present the main result of this section, namely the exact solution of the expectation in (12).

Theorem 3.1. *The KG decision rule in (12) can be written as*

$$X^{KG,n}(s^n) = \arg \max_{(i,j) \in E} \nu_{ij}^{KG,n} \tag{13}$$

where

$$\nu_{ij}^{KG,n} = \tilde{\sigma}_{ij}^n \cdot f\left(-\frac{V^n(s^n) - V_{ij}^n(s^n)}{\tilde{\sigma}_{ij}^n}\right). \tag{14}$$

Proof: As in the proof of Proposition 3.1, we consider two cases, one where $\delta_{ij}^{p^n} = 1$ and one where $\delta_{ij}^{p^n} = 0$. The two cases differ slightly, but in the end we derive one unified formula for $\nu_{ij}^{KG,n}$.

Case 1: $p^n \in A_{ij}^c$. Suppose that the edge (i,j) is not currently part of the best path. Nonetheless, we can potentially gain by measuring it. From Proposition 3.1 we know that only p^n or p_{ij}^n can be the best path at time $n+1$. Observe that p_{ij}^n will become the best path (beating p^n) if

$$\mu_{ij}^{n+1} > \mu_{ij}^n + \left(V^n(s^n) - V_{ij}^n(s^n)\right),$$

that is, our beliefs about (i, j) increase by an amount that is large enough to make up the time-n difference between p^n and p_{ij}^n. Note that $V^n(s^n) - V_{ij}^n(s^n) \geq 0$ by assumption, because $V^n(s^n)$ is the time-n length of the best time-n path. For all other outcomes of the measurement (that is, if our beliefs about (i, j) do not increase enough), p^n will continue to be the best path at time $n+1$.

The one-period increase in our beliefs about the length of the best path, denoted by $V^{n+1}(s^{n+1}) - V^n(s^n)$, depends on the outcome of the measurement in the following fashion:

$$
V^{n+1}(s^{n+1}) - V^n(s^n)
$$
$$
= \begin{cases} \mu_{ij}^{n+1} - \mu_{ij}^n - \left(V^n(s^n) - V_{ij}^n(s^n)\right) & \text{if } \mu_{ij}^{n+1} - \mu_{ij}^n \geq V^n(s^n) - V_{ij}^n(s^n) \\ 0 & \text{otherwise.} \end{cases} \tag{15}
$$

The shape of this function can be seen in Figure 1(a). Then, the knowledge gradient obtained by measuring (i, j) is

$$
\begin{aligned}
\nu_{ij}^{KG,n} &= \mathbb{E}_{ij}^n \left(V^{n+1}(s^{n+1}) - V^n(s^n) \right) \\
&= \mathbb{E}_{ij}^n \left[\left(\mu_{ij}^{n+1} - \mu_{ij}^n - \left(V^n(s^n) - V_{ij}^n(s^n)\right) \right) \cdot 1_{\left\{\mu_{ij}^{n+1} - \mu_{ij}^n \geq V^n(s^n) - V_{ij}^n(s^n)\right\}} \right].
\end{aligned}
$$

Equation (7) tells us that, given $\mathcal{F}^n$, $\mu_{ij}^{n+1} \sim \mathcal{N}\left(\mu_{ij}^n, \left(\tilde{\sigma}_{ij}^n\right)^2\right)$. Thus,

$$
\begin{aligned}
\nu_{ij}^{KG,n} &= \tilde{\sigma}_{ij}^n \cdot \mathbb{E}\left(Z \cdot 1_{\left\{ Z \geq \frac{V^n(s^n) - V_{ij}^n(s^n)}{\tilde{\sigma}_{ij}^n} \right\}} \right) \\
&\quad - \left(V^n(s^n) - V_{ij}^n(s^n)\right) \cdot P\left(Z \geq \frac{V^n(s^n) - V_{ij}^n(s^n)}{\tilde{\sigma}_{ij}^n} \right)
\end{aligned}
$$

where $Z \sim \mathcal{N}(0, 1)$. It follows that

$$
\begin{aligned}
\nu_{ij}^{KG,n} &= \tilde{\sigma}_{ij}^n \cdot \phi\left(-\frac{V^n(s^n) - V_{ij}^n(s^n)}{\tilde{\sigma}_{ij}^n} \right) \\
&\quad - \left(V^n(s^n) - V_{ij}^n(s^n)\right) \cdot \Phi\left(-\frac{V^n(s^n) - V_{ij}^n(s^n)}{\tilde{\sigma}_{ij}^n} \right) \\
&= \tilde{\sigma}_{ij}^n \cdot f\left(-\frac{V^n(s^n) - V_{ij}^n(s^n)}{\tilde{\sigma}_{ij}^n} \right). \tag{16}
\end{aligned}
$$

Case 2: $p^n \in A_{ij}$. If we measure an edge that is part of the best path, our estimate of the best path can become better or worse, depending on the outcome of the measurement. Then p_{ij}^n, the

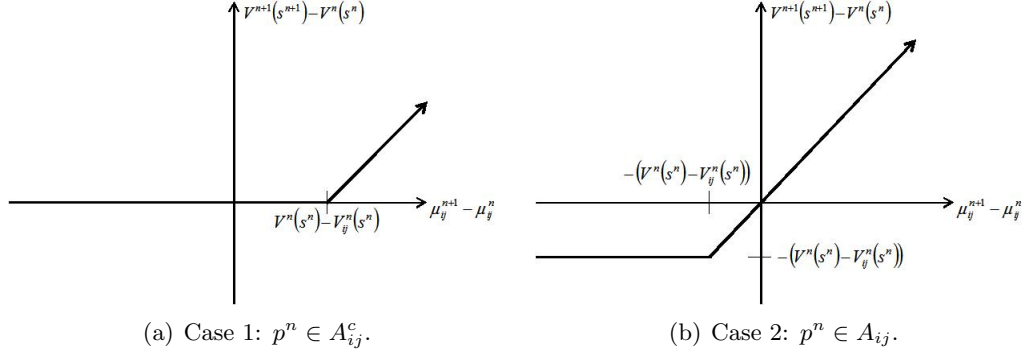(a) Case 1: $p^n \in A_{ij}^c$.   (b) Case 2: $p^n \in A_{ij}$.

Figure 1: Structure of the one-period increase in our beliefs about the best path.

best path not containing that edge, will become the best path at time $n+1$ if

$$\mu_{ij}^{n+1} < \mu_{ij}^n - \left(V^n \left(s^n \right) - V_{ij}^n \left(s^n \right) \right),$$

that is, if our beliefs about (i,j) drop far enough. In this case, the one-period improvement is given by

$$
\begin{aligned}
& V^{n+1} \left(s^{n+1} \right) - V^n \left(s^n \right) \\
&= \begin{cases} - \left(V^n \left(s^n \right) - V_{ij}^n \left(s^n \right) \right) & \text{if } \mu_{ij}^{n+1} - \mu_{ij}^n < - \left(V^n \left(s^n \right) - V_{ij}^n \left(s^n \right) \right) \\ \mu_{ij}^{n+1} - \mu_{ij}^n & \text{otherwise.} \end{cases}
\end{aligned}
\tag{17}
$$

The shape of this function is shown in Figure 1(b). The knowledge gradient is

$$
\begin{aligned}
\nu_{ij}^{KG,n} &= \mathbb{E}_{ij}^n \left(V^{n+1} \left(s^{n+1} \right) - V^n \left(s^n \right) \right) \\
&= - \left(V^n \left(s^n \right) - V_{ij}^n \left(s^n \right) \right) \cdot P \left(\mu_{ij}^{n+1} - \mu_{ij}^n < - \left(V^n \left(s^n \right) - V_{ij}^n \left(s^n \right) \right) \right) \\
&\quad + \mathbb{E}_{ij}^n \left[\left(\mu_{ij}^{n+1} - \mu_{ij}^n \right) \cdot 1_{\left\{ \mu_{ij}^{n+1} - \mu_{ij}^n \geq - \left(V^n(s^n) - V_{ij}^n(s^n) \right) \right\}} \right].
\end{aligned}
$$

As before, $\mu_{ij}^{n+1} \sim \mathcal{N} \left(\mu_{ij}^n, \left(\tilde{\sigma}_{ij}^n \right)^2 \right)$ given $\mathcal{F}^n$. Therefore,

$$
\begin{aligned}
\nu_{ij}^{KG,n} &= - \left(V^n \left(s^n \right) - V_{ij}^n \left(s^n \right) \right) \cdot P \left(Z < - \frac{\left(V^n \left(s^n \right) - V_{ij}^n \left(s^n \right) \right)}{\tilde{\sigma}_{ij}^n} \right) \\
&\quad + \tilde{\sigma}_{ij}^n \cdot \mathbb{E}_{ij}^n \left(Z \cdot 1_{\left\{ Z \geq - \frac{\left(V^n(s^n) - V_{ij}^n(s^n) \right)}{\tilde{\sigma}_{ij}^n} \right\}} \right)
\end{aligned}
$$

14

which becomes

$$
\begin{aligned}
\nu_{ij}^{KG,n} &= -\left(V^n\left(s^n\right) - V_{ij}^n\left(s^n\right)\right) \cdot \Phi\left(-\frac{V^n\left(s^n\right) - V_{ij}^n\left(s^n\right)}{\tilde{\sigma}_{ij}^n}\right) \\
&\quad + \tilde{\sigma}_{ij}^n \cdot \phi\left(-\frac{V^n\left(s^n\right) - V_{ij}^n\left(s^n\right)}{\tilde{\sigma}_{ij}^n}\right).
\end{aligned}
$$

This is the same expression as in (16). $\qquad\square$

The right-hand side of (14) provides us with a simple, easily computable formula for the knowledge gradient. The formula resembles an analogous formula for ranking and selection, examined by Frazier et al. (2008). However, (14) is designed specifically for the graph problem; to run the KG policy at time n, we are required to solve one shortest-path problem for each edge, to find $V_{ij}^n\left(s^n\right)$.

Equations (13) and (14) give an exact computation of (12) when the graph contains no cycles. If we allow cycles in the graph, then any path that is the best time-n path containing k copies of (i, j), for any $k = 0, 1, ...,$ can become the best time-$(n+1)$ path after measuring (i, j). It is difficult to enumerate all such paths; if the graph has cycles, we suggest (14) as an approximation to this difficult computation. For shortest-path problems, however, no path with a positive-cost cycle can ever be the shortest, so (13) and (14) closely approximate (12) as long as negative-cost cycles occur with negligible probability.

3.3   Asymptotic optimality of the KG policy

Define the risk function $R\left(p\right) = V - V^p$ to represent the loss incurred by choosing p instead of the true best path at time N. In this section, we show that the KG policy is asymptotically optimal in the sense of Frazier & Powell (2009), that is,

$$
\lim_{N \to \infty} \mathbb{E}\left(\min_p \mathbb{E}^N R\left(p\right)\right) = \mathbb{E}\left(\min_p \mathbb{E}\left(R\left(p\right) \mid \mu\right)\right). \tag{18}
$$

In words, the minimum-risk decision after N measurements will attain the minimum risk possible if all values are perfectly known, in the limit as $N \to \infty$. The crucial point is that the KG policy is the only learning policy that is optimal for both $N = 1$ (in the sense of Remark 3.1) and for $N \to \infty$ (in the sense of (18). This combination of myopic and asymptotic optimality suggests that KG could also perform well for finite measurement budgets.

All expectations in this discussion are under the KG policy; we drop the policy name from $\mathbb{E}^{KG}$

for notational convenience. Observe that

$$
\begin{aligned}
\lim_{N\to\infty} \mathbb{E}\left(\min_{p} \mathbb{E}^N R\left(p\right)\right) &= \lim_{N\to\infty} \mathbb{E}\left(\min_{p} \mathbb{E}^N\left(V - V^p\right)\right) \\
&= \lim_{N\to\infty} \mathbb{E}\left(\mathbb{E}^N V + \min_{p}\left(-\mathbb{E}^N V^p\right)\right) \\
&= \lim_{N\to\infty} \mathbb{E}V - \max_{p} V^{p,N}\left(s^N\right) \\
&= \mathbb{E}V - \lim_{N\to\infty} \mathbb{E}V^N\left(s^N\right).
\end{aligned}
$$

Using similar calculations, it can be shown that $\mathbb{E}\left(\min_p \mathbb{E}\left(R\left(p\right)\mid\mu\right)\right) = 0$, which means that we can rewrite (18) as

$$
\lim_{N\to\infty} \mathbb{E}V^N\left(s^N\right) = \mathbb{E}V. \tag{19}
$$

From Remark 2.1 we know that $\mathbb{E}V^N\left(s^N\right) \leq \mathbb{E}V$ for any N, so (19) means that an asymptotically optimal policy, with our usual implementation decision, achieves the highest possible objective value. The definition given in (18) is in line with the intuitive meaning of asymptotic optimality.

Theorem 3.2. *The KG policy of Theorem 3.1 is asymptotically optimal in the sense of (19).*

The proof of Theorem 3.2 is technical in nature, and can be found in the Appendix. The work by Frazier & Powell (2009) provides sufficient conditions for the asymptotic optimality of a KG-like learning policy in a general optimal learning setting. Our contribution is to verify that these conditions are satisfied by the KG policy for the graph setting. In the Appendix, we list the conditions in the context of the graph problem, then show that they are satisfied.

4 A Monte Carlo learning policy

In this section, we offer a different strategy for choosing edges. This approach views the paths of the graph as alternatives in a ranking and selection problem. We explain how to model this problem and solve it using the correlated KG algorithm by Frazier et al. (2009), assuming that we can enumerate all the paths. We then discuss how to use Monte Carlo simulation to avoid having to enumerate all the paths, instead generating a small subset of the set of all paths.

4.1 Ranking and selection on paths

Recall from Section 2.2 that V^p denotes the true value of a path p. Suppose for now that we can enumerate all the paths of the graph as $p_1, ..., p_P$. Let $V^{paths} = (V^{p_1}, ..., V^{p_P})$ denote the true lengths of these paths. Let $V^{paths,n}(s^n) = (V^{p_1,n}(s^n), ..., V^{p_P,n}(s^n))$ represent the paths' time-n lengths. Also, let E_p be the set of edges contained in path $p \in \{p_1, ..., p_P\}$. Because a path is characterized by its index, we will use p to refer to a path, as well as the path's index in the set $\{1, ..., P\}$.

From before, we know that $\mathbb{E}^n V^p = V^{p,n}(s^n)$ for any path p. Because $V^p = \sum_{(i,j) \in E_p} \mu_{ij}$, the conditional covariance of V^p and $V^{p'}$, given $\mathcal{F}^n$, is expressed by

$$\Sigma_{p,p'}^{paths,n}(s^n) = \sum_{(i,j) \in E_p \cap E_{p'}} \left(\sigma_{ij}^n\right)^2. \tag{20}$$

As before, the individual edges of the graph are independent. However, two paths are not independent if they have at least one edge in common, and the covariance of two path lengths is the sum of the variances of the edges that the two paths have in common. Then, given $\mathcal{F}^n$, we have

$$V^{paths} \sim \mathcal{N}\left(V^{paths,n}(s^n), \Sigma^{paths,n}(s^n)\right) \tag{21}$$

where $\Sigma^{paths,n}(s^n)$ is defined by (20). Thus, we have a multivariate Gaussian prior distribution on the vector V^{paths} of true path lengths.

Now suppose that, instead of measuring one edge in each time step, we can measure a path containing T edges, and use (4) and (5) to update our beliefs about every edge in that path. Because our measurements are independent, the variance of such a measurement is $\sigma_\varepsilon^2 T$. Our goal is to find $\arg\max_p V^p$, the path with the largest true value. This can be viewed a traditional ranking and selection problem with correlated Gaussian priors. The alternatives of the problem are paths, our beliefs are given by (21), and we choose a path to measure in every time step.

To solve this problem, we can apply the correlated knowledge gradient algorithm from Frazier et al. (2009). The knowledge gradient for path p in this problem is

$$\nu_p^{KGC,n} = \mathbb{E}_p^n\left(V^{n+1}(s^{n+1}) - V^n(s^n)\right). \tag{22}$$

For path p, we define a vector

$$\tilde{\sigma}^{KGC,n}(p) = \frac{\Sigma^{paths,n} e_p}{\sqrt{\sigma_\varepsilon^2 T + \Sigma_{pp}^{paths,n}}} \tag{23}$$

to represent the reduction in the variance of our beliefs about all paths achieved by measuring path p. Here e_p is a vector with 1 at index p and zeros everywhere else. Then, (22) can be rewritten as

$$\nu_p^{KGC,n} = \sum_{y=1}^{P-1} \left(\tilde{\sigma}_{y+1}^{KGC,n}(p) - \tilde{\sigma}_y^{KGC,n}(p) \right) f(-|c_y|) \qquad (24)$$

where the paths have been sorted in order of increasing $\tilde{\sigma}_y^{KGC,n}(p)$, f is as in Section 3, and the numbers c_y are such that $y = \arg\max_{p'} \left(V_{p'}^{paths,n}(s^n) + \tilde{\sigma}_{p'}^{KGC,n}(p) \cdot z \right)$ for $z \in [c_{y-1}, c_y)$, with ties broken by the largest-index rule. Then, the correlated KG policy for choosing a path is given by

$$X^{KGC,n}(s^n) = \arg\max_p \nu_p^{KGC,n}. \qquad (25)$$

4.2 Using Monte Carlo sampling to generate a choice set

There are two major problems with using the correlated KG policy to find a path. First, we want to measure individual edges, not paths. If we use (25) to find a path, we also need a rule for choosing an edge from that path. Second, and more importantly, it is difficult to enumerate paths, and thus we cannot use traditional ranking and selection methods on them. As an alternative to the KG policy described in Section 3, we propose a Monte Carlo-based policy that generates a small set of paths, and runs (25) on that set.

We run our Monte Carlo-based version of KG over the paths by first generating K sample realizations of the random variable $\bar{\mu}_{ij}^n \sim \mathcal{N}\left(\mu_{ij}^n, \left(\sigma_{ij}^n\right)^2\right)$ for every edge (i,j). Let $\bar{\mu}^n(\omega_k) = \left\{\bar{\mu}_{ij}^n(\omega_k) \,|\, (i,j) \in E\right\}$ be the kth sample realization. We can find a path corresponding to the kth realization by solving Bellman's equation using $\bar{\mu}^n(\omega_k)$ as the edge values. Because some sample realizations might yield the same best path, let K_0 be the number of distinct paths obtained from this procedure, and let $\ell_1, ..., \ell_{K_0}$ represent those distinct paths. As before, we will use ℓ to refer to a path as well as the path's index in $\{1, ..., K_0\}$.

Define the vector $V^{MC} = \left(V^{\ell_1}, ..., V^{\ell_{K_0}}\right)$ to represent the true lengths of the paths (using μ_{ij}), and similarly let $V^{MC,n}(s^n) = \left(V^{\ell_1,n}(s^n), ..., V^{\ell_{K_0},n}(s^n)\right)$ represent the paths' time-n lengths. Then, given $\mathcal{F}^n$, we have $V^{MC} \sim \mathcal{N}\left(V^{MC,n}(s^n), \Sigma^{MC,n}(s^n)\right)$ where

$$\Sigma_{\ell,\ell'}^{MC,n}(s^n) = \sum_{(i,j) \in E_\ell \cap E_{\ell'}} \left(\sigma_{ij}^n\right)^2.$$

To put it in words, we first find a set of K_0 different paths by solving K Monte Carlo shortest-path problems. Given the information we know at time n, the mean length of a path is the sum of

18

the time-n lengths of the links in that path, and the covariance of two path lengths is the sum of the variances of the edges that the two paths have in common. Then, given $\mathcal{F}^n$, the vector of path lengths has the multivariate Gaussian prior distribution given above. We can now apply the correlated KG algorithm for ranking and selection to the K_0 paths generated, and repeat the computations (23), (24) and (25) using $V^{MC,n}$ and $\Sigma^{MC,n}$ instead of $V^{P,n}$ and $\Sigma^{P,n}$. This procedure returns a path $\ell^{MC,n}$.

It remains to select an edge from this path. We propose the highest-variance rule

$$X^{MCKG,n}\left(s^n\right) = \arg\max_{(i,j)\in E_{\ell^{MC,n}}} \sigma_{ij}^n. \tag{26}$$

In the special case where $K_0 = 1$, we can simply follow (26) for the sole path generated, without additional computation.

In Section 5, we use the MCKG policy as a competitive strategy to evaluate the performance of the KG policy. However, we note that MCKG is also a new algorithm for this problem class. It can be used in a situation where (14) is too expensive to compute, but we can still solve K path problems for some $K < |E|$. The MCKG policy is equally suitable for cyclic and acyclic graphs.

5   Computational experiments

We examined the ways in which the performance of KG on a graph, relative to several other learning policies, was affected by the physical structure of the graph, the size of the graph, the measurement budget N, and the amount of information given by the prior. Our methods of graph generation are discussed in Section 5.1. As stated at the end of Section 3, it does not matter whether we are looking for the shortest or longest path, because the KG formula in (14) will be the same in both cases. In our experiments, we minimized path length on graphs with a clearly defined source node and destination node; for all of our learning policies, we used a freeware implementation of Dijkstra's algorithm to solve the shortest-path problems. In this setting, if π is a measurement policy, and p^π is the path that seems to be the best at time N after having followed π, then the opportunity cost of π is defined to be

$$C^\pi = V^{p^\pi} - V, \tag{27}$$

the difference in the true length of the path p^π and the true length of the true best path. This is the error we make by choosing the path that seems to be the best after running policy π. The

quantity V is found using (2), with the maximum replaced by a minimum. For policies π_1 and π_2,

$$C^{\pi_2} - C^{\pi_1} = V^{p^{\pi_2}} - V^{p^{\pi_1}} \tag{28}$$

is the amount by which policy π_1 outperforms policy π_2. Positive values of (28) indicate that π_1 found a shorter (better) path, whereas negative values of (28) mean the opposite. For every experiment in our study, we ran each measurement policy 10^4 times, starting from the same initial data, thus obtaining 10^4 samples of (27) for each policy. The 10^4 sample paths were divided into groups of 500 in order to obtain approximately normal samples of opportunity cost and the standard errors of those averages. The standard error of the difference in (28) is the square root of the sum of the squared standard errors of C^{π_1}, C^{π_2}.

Crucially, this performance metric requires us to know the true values μ for every graph we consider. In order to test a learning policy, we first assume a truth, then evaluate the ability of the policy to find that truth. For this reason, the starting data for our experiments were randomly generated, including the physical graph structure itself. Because we minimized path length, we generated μ and μ^0 large enough to avoid negative edge values in our measurements.

For each graph, we generated two sets of numbers. In the *heterogeneous-prior* set, the prior means μ^0 were generated from a uniform distribution on $[450, 550]$. The prior variances were generated from a uniform distribution on $[95, 105]$; the purpose of using such a narrow interval was to ensure that all of them would be approximately equal, but any one would be equally likely to be the largest. Then, for each edge (i, j), the true value μ_{ij} was generated from a Gaussian distribution with mean μ_{ij}^0 and variance $\left(\sigma_{ij}^0\right)^2$. This represents a situation in which our prior beliefs are accurate on average, and give us a reasonably good idea about the true values. The measurement noise σ_ε^2 was chosen to be 100^2.

In the second set of initial parameters, referred to as the *equal-prior* set, we generated the prior means μ^0 from a uniform distribution on $[495, 505]$, the purpose of the narrow interval again being to break ties among the priors. The true means μ were generated from a uniform distribution on $[300, 700]$. The prior variances and the measurement noise were obtained the same way as in the heterogeneous-prior experiments. The true edge lengths fall into roughly the same range as in the heterogeneous-prior experiments, but the priors now give us much less information about them.

Five policies were tested overall; we briefly describe their implementation.

Knowledge gradient on a graph (KG). This policy is defined by the decision rule (13), the exact

KG policy for acyclic graphs. The quantity $V^n(s^n)$ is found by solving a shortest-path problem using μ^n as the edge values. The quantity $V_{ij}^n(s^n)$ is found in a similar fashion, with the value of (i,j) modified as described in Section 3.

Pure exploitation (Exp). The pure exploitation policy consists of finding the path p^n that solves (8) with max replaced by min, then measuring the edge given by $X^{Exp,n}(s^n) = \arg\min_{(i,j)\in p^n} \mu_{ij}^n$.

Variance-exploitation (VExp). This policy is a slight modification of the pure exploitation policy. It measures the edge given by $X^{VExp,n}(s^n) = \arg\max_{(i,j)\in p^n} \sigma_{ij}^n$. Instead of simply choosing the edge that looks the best on the path that looks the best, it chooses the edge that we are least certain about on that same path.

Monte Carlo correlated KG (MCKG). The Monte Carlo policy is described in Section 4. The decision rule for this policy is given by (26). The policy has one parameter K, the number of Monte Carlo samples generated. In our experiments, we used $K = 30$. We found that smaller values of K resulted in very few paths. On the other hand, larger values did not appreciably increase the number K_0 of distinct paths generated (which was typically in the single digits), while requiring substantially more computational time.

Pure exploration (Explore). In every iteration, the pure exploration policy chooses an edge uniformly at random and measures it.

5.1 Effect of graph structure on KG performance

We considered three general types of graph structure:

Layered graphs (Layer (L, B, c)). The layered graph is closest in form to the time-staged model we developed in Section 2. It consists of a source node, a destination node, and L layers in between. Each layer contains B nodes, and every node in every layer except for the last one is connected to c randomly chosen nodes in the next layer. The source is connected to every node in the first layer, and every node in the last layer is connected to the destination. The total number of nodes in the graph is $L \cdot B + 2$, and the total number of edges is $(L-1) \cdot B \cdot c + 2 \cdot B$. The edges are directed, so every layered graph is acyclic.

Erdős-Renyi graphs $(ER(D, p))$. The Erdős-Renyi random graph model was introduced by Gilbert (1959) and Erdős & Renyi (1959). A graph has D nodes, and any two nodes have a fixed
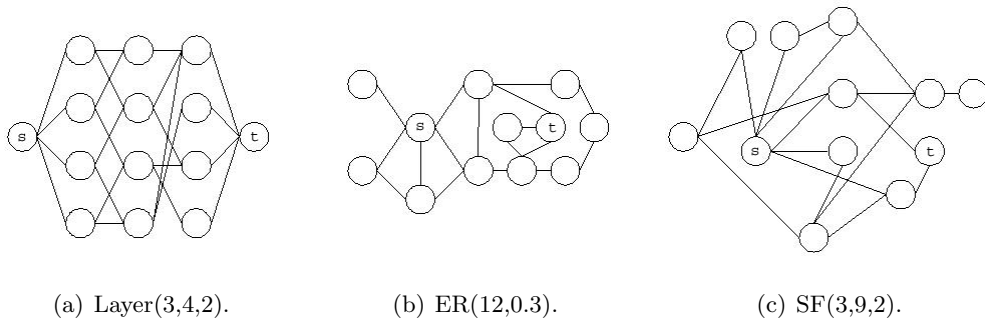
(a) Layer(3,4,2). (b) ER(12,0.3). (c) SF(3,9,2).

Figure 2: Examples of layered, Erdős-Renyi and scale-free graphs. The source and destination nodes are marked by s and t.

probability p of being connected by an edge. Thus, the total number of edges in the graph varies, but on average is equal to $\binom{D}{2} \cdot p$. In our experiments, the source is the node with index 1 and the sink is the node with index D.

Scale-free graphs $(SF(S, I, c))$. We use the scale-free graph model created by Barabási & Albert (1999). We start with S nodes, and run I iterations. In every iteration, we add one new node and connect it to c randomly chosen, previously existing nodes. The total number of nodes is equal to $S + I$, and the total number of edges is equal to $I \cdot c$. In our experiments, the source is the first node added and the sink is the last node added.

Figure 2 gives examples of all three types. In the layered graph, any path from source to destination contains the same number of edges. In the other graphs, several nodes have very high degrees, so there tends to be at least one very short path from one node to another. Layered graphs are acyclic, so (13) and (14) give the exact computation of (12). The other two types of graphs can have cycles, so we use (13) as a close approximation of (12). Our edge values are high enough to make the probability of negative-cost cycles negligible.

We generated 10 graphs of each type, each with approximately 30 nodes and 50 edges. The exact types were $Layer(4, 5, 3)$, $ER(30, 0.1)$ and $SF(5, 25, 2)$. The minimum, average, and maximum values of the difference (28) across ten graphs of each type are given in Tables 1, 2 and 3 for both the heterogeneous-prior and equal-prior experiments. The measurement budget was taken to be $N = 30$, or approximately 60% of the number of edges.

The KG policy gives the best performance on the layered graphs, where it outperforms all other policies on average. In the worst case, it can be outperformed by pure exploitation and variance-

	Heterogeneous-prior			Equal-prior		
	Min	Average	Max	Min	Average	Max
KG-Exp	-2.4410	151.5178	337.4421	162.2031	367.6961	673.0983
KG-VExp	-3.7512	62.6977	104.9868	-15.9721	72.6030	130.8938
KG-MCKG	29.6875	60.8563	89.8636	-30.9532	54.7674	195.6884
KG-Explore	13.1494	93.1405	145.6467	33.5755	95.8332	167.7865

Table 1: Mean differences in opportunity cost across ten $Layer\,(4, 5, 3)$ graphs.

	Heterogeneous-prior			Equal-prior		
	Min	Average	Max	Min	Average	Max
KG-Exp	-29.5931	14.7976	161.8455	-3.4380	29.8378	249.9321
KG-VExp	-82.0065	-8.5772	2.9997	-43.8978	8.4519	79.0177
KG-MCKG	-161.1705	-17.5574	8.9841	-51.1068	-3.7332	10.5969
KG-Explore	-49.7891	4.9316	53.6483	0.0	24.0246	94.4013

Table 2: Mean differences in opportunity cost across ten $ER\,(30, 0.1)$ graphs.

	Heterogeneous-prior			Equal-prior		
	Min	Average	Max	Min	Average	Max
KG-Exp	-4.9423	9.6666	53.1207	-36.3735	3.4179	80.8175
KG-VExp	-5.0684	6.6864	66.7394	-21.9450	6.2264	85.9265
KG-MCKG	-3.2274	0.2408	3.2134	-30.5455	-2.0340	12.9562
KG-Explore	-5.2182	9.7683	82.3290	0.0	22.2583	88.9437

Table 3: Mean differences in opportunity cost across ten $SF\,(5, 25, 2)$ graphs.

exploitation. However, even then, the difference is negligible, as the value of a typical path in one of these layered graphs is around 2500. Furthermore, in the best case, the KG policy outperforms the competition by a much larger margin.

For the other two types of graphs, KG performs competitively on average, but is outperformed by every policy in the worst case, though the margin is very small for scale-free graphs. The Monte Carlo policy performs especially well on both Erdős-Renyi and scale-free graphs, with a slight edge over KG on average. In general, the competition is much tighter than for the layered graphs.

In Erdős-Renyi and scale-free graphs, there tends to be at least one path from source to destination that contains very few edges. When the values on the edges are similar in magnitude, this means that a path with fewer edges is more likely to be the best. In such graphs, our consideration is narrowed down to a small number of very short paths; even in the equal-prior case, the graph topology provides a great deal of information. In fact, all five of our policies were able to find the true best path in five out of ten of the Erdős-Renyi graphs. For this reason, Table 2 contains one 0.0 value, meaning that both policies under consideration achieved a zero opportunity cost.

	Heterogeneous-prior			Equal-prior		
	Layer	ER	SF	Layer	ER	SF
KG-Exp	1.7887	0.3589	0.2364	2.0525	0.5892	0.3060
KG-VExp	1.9584	0.3447	0.2349	2.4267	0.5329	0.3710
KG-MCKG	1.7512	0.2479	0.1414	2.1261	0.2482	0.2857
KG-Explore	2.0358	0.3728	0.2652	2.3589	0.6671	0.4579

Table 4: Average standard errors of the differences in opportunity cost.

In a layered graph, however, every path contains the same number of edges. In this case, small differences in our prior beliefs matter much more, and there are many more paths that could potentially be the best. In this setting, exploitation-based methods quickly get stuck on an incorrect path, while exploration is unable to discover enough useful information. The KG policy, on the other hand, is more effective at finding a good path. Thus, the KG policy is a particularly good choice for a time-staged graph model.

We also see that KG tends to perform better in the equal-prior setting for layered and Erdős-Renyi graphs. On scale-free graphs, the performance of KG suffers in the worst case, but benefits in the best case, with a slight drop in average-case performance. For the most part, we see that KG can learn effectively when the prior gives relatively little information, especially in the layered graph setting. The most effective policy after KG is MCKG, which has the most sophisticated learning mechanism among the competition. This is also the only policy among the competition which adapts well to the equal-prior setting, maintaining its performance relative to KG on average.

Table 4 shows the average standard errors of our estimates of (28) across each set of graphs. The numbers are much smaller than most of the mean differences reported in Tables 1-3. As expected, the standard error is larger for layered graphs, when we have more paths to choose from.

Finally, Table 5 reports the average number of distinct edges measured by each policy for each set of graphs. Once again, we find that all policies except pure exploration examine fewer distinct

	Heterogeneous-prior			Equal-prior		
	Layer	ER	SF	Layer	ER	SF
KG	20.5801	10.7087	7.6630	22.9280	8.9953	9.1416
Exp	3.4162	1.5208	2.2032	3.1745	2.1769	1.9843
VExp	10.7140	3.9947	3.9494	12.5684	3.6904	3.9536
MCKG	29.0706	5.8198	5.6184	29.5703	5.4339	5.4444
Explore	23.2830	21.5285	22.7159	23.2794	21.5349	22.7244

Table 5: Average number of distinct edges measured by each policy.

	Heterogeneous-prior			Equal-prior		
	Min	Average	Max	Min	Average	Max
KG-Exp	212.1008	364.1344	513.2963	375.6234	554.0195	901.4031
KG-VExp	-46.8154	101.1566	228.8304	31.5401	112.3906	167.6988
KG-MCKG	48.1381	113.5038	169.0582	42.5510	123.5008	298.5638
KG-Explore	119.0861	175.2994	296.3735	106.6856	185.7381	304.1708

Table 6: Mean differences in opportunity cost across ten $Layer\,(6,6,3)$ graphs.

edges on Erdős-Renyi and scale-free graphs than on layered graphs. For instance, when the MCKG policy takes Monte Carlo samples of the best path on a layered graph, the choice of the best path can be decided by minor variations in the edge samples, and we will sample more distinct paths. However, on a graph where there are one or two paths with very few edges, those few paths will almost always come out on top in the Monte Carlo sampling, and MCKG will do much less exploration than before.

5.2   Effect of graph size on KG performance

We examined the effect of graph size on the performance of the KG policy on the layered graphs discussed in Section 5.1, the graph type that most resembles the time-staged model introduced in Section 2. We generated a set of ten $Layer\,(6,6,3)$ graphs. Thus, each graph in the set had 38 nodes and 102 edges, approximately twice as many as the graphs in Section 5.1. We also increased the measurement budget to $N = 60$, again approximately 60% of the number of edges.

The performance of the KG policy on this set is summarized in Table 6. Every path in these graphs contains two more edges than for the $Layer\,(4,5,3)$ graphs, so the typical path length is greater by about 1000, a 40% increase. However, the average values in Table 6 are about twice as large as the analogous values in Table 1, indicating that the competing policies fall behind the KG policy as the graph size increases.

These results are consistent with our observations in Section 5.1: KG is more effective in situations where there are more paths to choose from. Essentially, KG is better-equipped to manage large numbers of alternatives than the other learning policies.
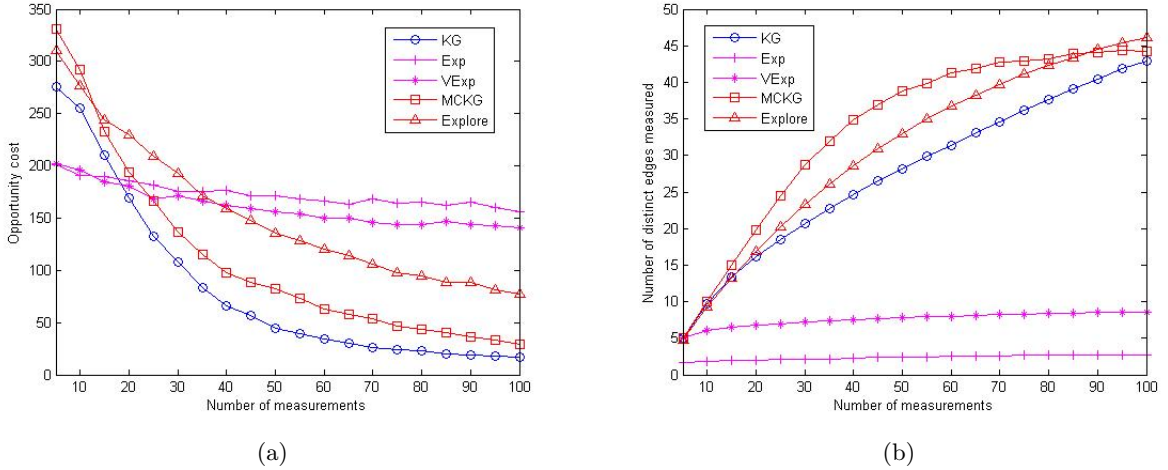
Figure 3: Opportunity cost and number of edges measured as functions of N.

5.3 Effect of measurement budget on KG performance

We tested our learning policies on one randomly chosen $Layer\,(4, 5, 3)$ problem from Section 5.1 using measurement budgets of $N = 5, 10, ..., 100$. Figure 3(a) shows the way in which the opportunity cost C^π changes with the measurement budget for each of our learning policies. We see that the KG policy consistently achieves the lowest opportunity cost. The only close competition comes from MCKG, which lags behind the KG policy for N around 50, but catches up for large N.

Figure 3(b) tracks the number of distinct edges (out of 50 total) measured by each policy for each measurement budget. As in Table 5, KG is exactly in the middle of the five policies. The variants of pure exploitation do not explore enough. Pure exploration and MCKG explore more edges than the KG policy, but do so less effectively. For $N > 50$, we see that MCKG explores more slowly, until it almost exactly matches the KG policy for $N = 100$.

We can conclude that KG is effective for small measurement budgets. As long as we are allowed fewer measurements than there are edges (the given graph has about 50 edges), KG uses those measurements more efficiently than the competition. When the measurement budget is relatively large, the Monte Carlo policy becomes a viable alternative. Together with our study of graph size and structure, these results indicate that KG performs better relative to other policies when there are many interesting paths to consider, but relatively few chances to measure them.

6 Conclusion

We have proposed a strategy for a new type of learning problem: the problem of finding the best path in a graph with unknown edge weights, given finitely many chances to measure individual edges. When the edge weights have normally distributed priors and normal sampling error with known variance, the KG policy results in a simple decision rule which requires us to compute a closed-form expression for every edge in the graph. Like analogous policies for offline ranking and selection, the KG policy is myopically and asymptotically optimal. Our decision rule computes the knowledge gradient exactly for acyclic graph problems, and approximates it for graphs with cycles.

In our experiments, we considered how the performance of the KG policy is affected by several factors: the general type of graph structure, the size of the graph, the measurement budget, and the amount of information conveyed by the prior. We found that the KG policy on average outperforms several learning heuristics, including a Monte Carlo adaptation of a KG-type policy for ranking and selection. The KG policy is particularly effective on problems where there are many possible paths that could potentially be the best, but where the measurement budget is relatively small. This is precisely the sort of problem where it is important to learn efficiently. We conclude that the KG policy has strong potential for application in graph problems, where the physical structure makes it difficult to use traditional ranking and selection methods.

The KG logic of choosing a measurement to maximize the expected improvement in our beliefs about the optimal value of some objective function is very general. It is possible to envision problems that are more general than the graph problem, just as the graph problem itself is more general than the ranking and selection problem. The generality of the KG concept suggests that KG-like policies can also be derived in learning problems with still more complex objective functions. For example, it is possible to envision an objective where the sampling cost has an economic representation other than just a finite measurement budget. The ranking and selection literature includes work on economic analysis (see e.g. Chick & Gans (2009)), and the work on KG methods has also begun to incorporate some of these ideas (see Chick & Frazier (2009)). The primary challenge in this case would be the computation of the KG factor, which is problem-specific. Still, we believe that the KG methodology can potentially open a new direction in the modeling and analysis of complex optimal learning problems.

Acknowledgements

References

Barabási, A. & Albert, R. (1999), 'Emergence of Scaling in Random Networks', *Science* **286**(5439), 509–512.

Bechhofer, R., Santner, T. & Goldsman, D. (1995), *Design and Analysis of Experiments for Statistical Selection, Screening and Multiple Comparisons*, J.Wiley & Sons, New York.

Bernardo, J. M. & Smith, A. F. M. (1994), *Bayesian Theory*, John Wiley and Sons, New York.

Branke, J., Chick, S. & Schmidt, C. (2007), 'Selecting a selection procedure', *Management Science* **53**(12), 1916–1932.

Chick, S. & Gans, N. (2009), 'Economic analysis of simulation selection options', *Management Science* **55**(3), 421–437.

Chick, S. & Inoue, K. (2001*a*), 'New procedures to select the best simulated system using common random numbers', *Management Science* **47**(8), 1133–1149.

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

Chick, S., Branke, J. & Schmidt, C. (2009), 'Sequential Sampling to Myopically Maximize the Expected Value of Information', *INFORMS J. on Computing (to appear)*.

Chick, S. E. & Frazier, P. I. (2009), The Conjunction Of The Knowledge Gradient And The Economic Approach To Simulation Selection, *in* M. Rosetti, R. Hill, B. Johansson, A. Dunkin & R. Ingalls, eds, 'Proceedings of the 2009 Winter Simulation Conference', pp. 528–539.

Dearden, R., Friedman, N. & Russell, S. (1998), Bayesian Q-learning, *in* 'Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98)'.

DeGroot, M. H. (1970), *Optimal Statistical Decisions*, John Wiley and Sons.

Duff, M. & Barto, A. (1996), Local bandit approximation for optimal learning problems, *in* 'Advances in Neural Information Processing Systems', Vol. 9, Cambridge, MA: MIT Press, pp. 1019–1025.

Erdős, P. & Renyi, A. (1959), 'On random graphs', *Publicationes Mathematicae* **6**, 290–297.

Fan, Y., Kalaba, R. & Moore, J. (2005), 'Shortest Paths in Stochastic Networks with Correlated Link Costs', *Computers and Mathematics with Applications* **49**, 1549–1564.

Frazier, P. I. & Powell, W. B. (2009), 'Convergence to Global Optimality with Sequential Bayesian Sampling Policies', *Submitted for publication.*

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

Frazier, P. I., Powell, W. B. & Dayanik, S. (2009), 'The knowledge-gradient policy for correlated normal rewards', *INFORMS J. on Computing* **21**(4), 599–613.

Frieze, A. & Grimmett, G. (1985), 'The shortest-path problem for graphs with random arc-lengths', *Discrete Applied Mathematics* **10**(1), 57–77.

Gelman, A., Carlin, J., Stern, H. & Rubin, D. (2004), *Bayesian data analysis (2nd ed.)*, CRC Press.

Gilbert, E. (1959), 'Random graphs', *Annals of Mathematical Statistics* **30**(4), 1141–1144.

Gittins, J. (1989), *Multi-Armed Bandit Allocation Indices*, John Wiley and Sons, New York.

Goldsman, D. (1983), Ranking and selection in simulation, *in* 'Proceedings of the 15th Conference on Winter Simulation, Volume 2', IEEE Press Piscataway, NJ, USA, pp. 387–394.

Gupta, S. & Miescke, K. (1996), 'Bayesian look ahead one-stage sampling allocations for selection of the best population', *Journal of statistical planning and inference* **54**(2), 229–244.

Hoff, P. (2009), *A First Course in Bayesian Statistical Methods*, Springer Verlag.

Inoue, K., Chick, S. & Chen, C. (1999), 'An empirical evaluation of several methods to select the best system', *ACM Transactions on Modeling and Computer Simulation (TOMACS)* **9**(4), 381–407.

Kim, S. & Nelson, B. (2006), Selecting the best system, *in* S. Henderson & B. Nelson, eds, 'Handbooks of Operations Research and Management Science, vol. 13: Simulation', North-Holland Publishing, Amsterdam, pp. 501–534.

Kim, S. & Nelson, B. (2007), Recent advances in ranking and selection, *in* 'Proceedings of the 39th conference on Winter simulation', IEEE Press Piscataway, NJ, USA, pp. 162–172.

Kulkarni, V. (1986), 'Shortest paths in networks with exponentially distributed arc lengths', *Networks* **16**, 255–274.

Law, A. & Kelton, W. (1991), *Simulation Modeling and Analysis (2nd ed.)*, McGraw Hill, Inc., New York.

Lukacs, E. (1942), 'A characterization of the normal distribution', *The Annals of Mathematical Statistics* **13**(1), 91–93.

Peer, S. & Sharma, D. (2007), 'Finding the shortest path in stochastic networks', *Computers and Mathematics with Applications* **53**, 729–740.

Ryzhov, I. O. & Powell, W. B. (2009), The knowledge gradient algorithm for online subset selection, *in* 'Proceedings of the 2009 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning, Nashville, TN', pp. 137–144.

Ryzhov, I. O., Powell, W. B. & Frazier, P. I. (2009), 'The knowledge gradient algorithm for a general class of online learning problems', *Submitted for publication*.

Schmeiser, B. (1982), 'Batch size effects in the analysis of simulation output', *Operations Research* **30**(3), 556–568.

Snyder, T. & Steele, J. (1995), Probabilistic networks and network algorithms, *in* M. Ball, T. Magnanti & C. Monma, eds, 'Handbooks of Operations Research and Management Science, vol. 7: Networks', North-Holland Publishing, Amsterdam, pp. 401–424.

Watkins, C. & Dayan, P. (1992), 'Q-learning', *Machine Learning* **8**(3), 279–292.

7 Appendix

7.1 Proof of Proposition 2.1

We use an inductive argument. Observe that $V_T^n(i; s^n) = 0 = V_T(i)$ for all i and all s^n. Now suppose that $V_{t'}^n(i; s^n) \leq \mathbb{E}^n V_{t'}(i)$ for all i, for all s^n, and for all $t' > t$. Then,

$$
\begin{aligned}
V_t^n(i; s^n) &= \max_j \mu_{ij}^n + V_{t+1}^n(j; s^n) \\
&\leq \max_j \mu_{ij}^n + \mathbb{E}^n V_{t+1}(j) \\
&= \max_j \mathbb{E}^n(\mu_{ij} + V_{t+1}(j)) \\
&\leq \mathbb{E}^n\left(\max_j \mu_{ij} + V_{t+1}(j)\right) \\
&= \mathbb{E}^n V_t(i)
\end{aligned}
$$

where the second line is due to the inductive hypothesis for $t' = t + 1$, the third line is true because μ_{ij}^n is measurable with respect to $\mathcal{F}^n$, and the fourth line follows by Jensen's inequality. $\quad\square$

7.2 Proof of asymptotic optimality

As in Section 3.3, all expectations in this discussion are under the KG policy. For each n, we define the quantities

$$
h_{ij}^n = \min_p \mathbb{E}^n R(p) - \mathbb{E}_{ij}^n\left(\min_p \mathbb{E}^{n+1} R(p)\right) \tag{29}
$$

$$
g_{ij}^n = \min_p \mathbb{E}^n R(p) - \mathbb{E}^n\left(\min_p \mathbb{E}^n(R(p) \mid \mu_{ij})\right) \tag{30}
$$

$$
g^n = \min_p \mathbb{E}^n R(p) - \mathbb{E}^n\left(\min_p \mathbb{E}^n(R(p) \mid \mu)\right). \tag{31}
$$

The quantity h_{ij}^n represents the one-period decrease in risk obtained by measuring edge (i, j) at time n. The quantity g_{ij}^n represents the expected decrease in risk obtained by knowing μ_{ij} exactly. Finally, g^n is the expected decrease in risk obtained by knowing all the true edge values μ exactly.

Theorem 7.1. *(Theorem 4 from Frazier & Powell (2009)) Suppose that the following conditions hold for all n:*

1. *If $g_{ij}^n = 0$ for every $(i, j) \in E$, then $g^n = 0$ also.*

2. For any $(i,j) \in E$, if $h_{ij}^n = 0$, then $g_{ij}^n = 0$ also.

3. For any $(i,j) \in E$, both h_{ij}^n and g_{ij}^n are continuous in s^n, that is, they are continuous at all $\mu^n \in \mathbb{R}^{|E|}$ and $\sigma^n \geq 0$, including the boundary points where at least one component of σ^n is zero.

Then, the KG policy of Section 3 is asymptotically optimal in the sense of (19).

We now verify that the KG policy for learning on a graph satisfies these conditions. By substituting the definition of $R(p)$ into (29), we see that

$$
\begin{aligned}
h_{ij}^n &= \min_p \mathbb{E}^n (V - V^p) - \mathbb{E}_{ij}^n \left(\min_p \mathbb{E}^{n+1} (V - V^p) \right) \\
&= \mathbb{E}^n V - \max_p V^{p,n}(s^n) - \mathbb{E}_{ij}^n \left(\mathbb{E}^{n+1} V - \max_p V^{p,n+1}(s^{n+1}) \right) \\
&= \mathbb{E}^n V - \max_p V^{p,n}(s^n) - \mathbb{E}^n V + \mathbb{E}_{ij}^n \max_p V^{p,n+1}(s^{n+1}) \\
&= \mathbb{E}_{ij}^n \left(V^{n+1}(s^{n+1}) - V^n(s^n) \right) \\
&= \nu_{ij}^{KG,n},
\end{aligned}
\tag{32}
$$

so the one-period decrease in risk is the same as the one-period increase in value. Thus, the KG policy we derived in section 3 maximizes the one-period decrease in risk, and therefore satisfies the definition of a knowledge gradient policy given in Frazier & Powell (2009).

Repeating the same computations as in (32) for g_{ij}^n, we find that

$$
g_{ij}^n = \mathbb{E}^n \left(\max_p \mathbb{E}^n (V^p \,|\, \mu_{ij}) - V^n(s^n) \right).
\tag{33}
$$

This difference is very similar to the difference in (12). When we compute $\mathbb{E}^n V^{n+1}(s^{n+1})$ in (12), we are computing the expected length of the best path, given that our beliefs about one edge (i,j) will change from μ_{ij}^n to μ_{ij}^{n+1}, over the time-n distribution of μ_{ij}^{n+1}. In (33), we do the same thing, given that our beliefs about (i,j) will change from μ_{ij}^n to μ_{ij}. At time n, $\mu_{ij} \sim \mathcal{N}\left(\mu_{ij}^n, \left(\sigma_{ij}^n\right)^2 \right)$. Therefore, to compute (33), we simply repeat the calculations of Section 3, using σ_{ij}^n instead of $\tilde{\sigma}_{ij}^n$. It follows that

$$
g_{ij}^n = \sigma_{ij}^n \cdot f \left(-\frac{V^n(s^n) - V_{ij}^n(s^n)}{\sigma_{ij}^n} \right)
\tag{34}
$$

where $V_{ij}^n(s^n)$ and f are defined the same way as in Section 3.

Finally, we can repeat the computations from (32) once more for g^n and find that

$$g^n = \mathbb{E}^n V - V^n (s^n).$$

Thus, g^n is precisely the difference between the right and left sides of (10).

Proposition 7.1. $g_{ij}^n = 0$ if and only if $\sigma_{ij}^n = 0$. Also, $h_{ij}^n = 0$ if and only if $\sigma_{ij}^n = 0$.

Proof: Suppose that $\sigma_{ij}^n = 0$. Because $\mu_{ij} \sim \mathcal{N}\left(\mu_{ij}^n, \left(\sigma_{ij}^n\right)^2\right)$ given $\mathcal{F}^n$, it follows that $\mu_{ij} = \mu_{ij}^n$ almost surely at time n. Therefore, the right-hand side of (33) is zero.

However, if $\sigma_{ij}^n > 0$, we repeat the calculations from Section 3 and arrive at (34). The right-hand side of (34) will be strictly positive, because f has no zeros on the real line. Observe that $f'(z) = \Phi(z) > 0$, so if f had any zeros, there would have to be some region of the real line on which f was strictly negative. However, f is known to be positive, e.g. from Frazier et al. (2008), so f can have no zeros. It follows that $g_{ij}^n > 0$ as well. Thus, $g_{ij}^n = 0$ if and only if $\sigma_{ij}^n = 0$.

The same reasoning applies to h_{ij}^n. If $\sigma_{ij}^n = 0$, then $\tilde{\sigma}_{ij}^n = 0$, which means that $\mu_{ij} = \mu_{ij}^n$ almost surely at time n, and the right-hand side of (12) is zero. If $\sigma_{ij}^n > 0$, then $\tilde{\sigma}_{ij}^n > 0$ and we apply the reasoning from Section 3 to arrive at (14). However, because f has no zeros, it follows that $\nu_{ij}^{KG,n} > 0$ and $h_{ij}^n > 0$. Consequently, $h_{ij}^n = 0$ if and only if $\sigma_{ij}^n = 0$. $\qquad\square$

Proposition 7.2. Let s^n be any knowledge state, that is, take any $\mu^n \in \mathbb{R}^{|E|}$ and any $\sigma^n \geq 0$. Suppose that $g_{ij}^n = 0$ for all $(i, j) \in E$. Then, $g^n = 0$ also.

Proof: If $g_{ij}^n = 0$ for all (i, j), it follows that $\sigma_{ij}^n = 0$ for all (i, j) by Proposition 7.1. This means that, at time n, $\mu_{ij} = \mu_{ij}^n$ almost surely for every (i, j), which means that $\mathbb{E}^n V = V^n (s^n)$. $\qquad\square$

Proposition 7.3. For any knowledge state s^n with $\mu^n \in \mathbb{R}^{|E|}$ and $\sigma^n \geq 0$, if $h_{ij}^n = 0$, then $g_{ij}^n = 0$.

Proof: Using Proposition 7.1 twice, if $h_{ij}^n = 0$, then $\sigma_{ij}^n = 0$, but this also means that $g_{ij}^n = 0$. $\qquad\square$

Proposition 7.4. For every $(i, j) \in E$, g_{ij}^n and h_{ij}^n are continuous in s^n, that is, they are continuous at all $\mu^n \in \mathbb{R}^{|E|}$ and $\sigma^n \geq 0$, including the boundary points where at least one component of σ^n is zero.

Proof: If $\sigma_{ij}^n > 0$, then h_{ij}^n and g_{ij}^n are given by (14) and (34). The function $f(z)$ is continuous in z. Furthermore, $V^n (s^n)$ and $V_{ij}^n (s^n)$ are continuous in μ^n, because the time-n length of a fixed

path is continuous in μ^n (being the sum of some components of μ^n), therefore a maximum over a finite set of fixed paths is also continuous in μ^n. Thus, h_{ij}^n and g_{ij}^n are continuous at all μ^n and all σ^n with $\sigma_{ij}^n > 0$.

If we take $\sigma_{ij}^n \to 0$, then we also have $\tilde{\sigma}_{ij}^n \to 0$. Because f is increasing, it follows that

$$\tilde{\sigma}_{ij}^n \cdot f\left(-\frac{V^n(s^n) - V_{ij}^n(s^n)}{\tilde{\sigma}_{ij}^n}\right) \leq \tilde{\sigma}_{ij}^n \cdot f(0). \tag{35}$$

As $\tilde{\sigma}_{ij}^n \to 0$, the right-hand side of (35) vanishes, which means that $h_{ij}^n \to 0$ also. The same holds for g_{ij}^n, if we replace $\tilde{\sigma}_{ij}^n$ with σ_{ij}^n in (35).

Thus, the limit of (14) and (34) as $\sigma_{ij}^n \to 0$ is zero. We know from Proposition 7.1 that $h_{ij}^n = g_{ij}^n = 0$ if $\sigma_{ij}^n = 0$, so h_{ij}^n and g_{ij}^n are also continuous at all μ^n and all σ^n with $\sigma_{ij}^n = 0$. □

These results constitute the proof of Theorem 3.2. The three conditions of Theorem 7.1 are verified by Proposition 7.2, Proposition 7.3 and Proposition 7.4, respectively.