

INFORMATION COLLECTION FOR LINEAR PROGRAMS WITH UNCERTAIN OBJECTIVE COEFFICIENTS*

ILYA O. RYZHOV[†] AND WARREN B. POWELL[‡]

Abstract. Consider a linear program with uncertain objective coefficients, for which we have a Bayesian prior. We can collect information to improve our understanding of these coefficients, but this may be expensive, giving us a separate problem of optimizing the collection of information to improve the quality of the solution relative to the true cost coefficients. We formulate this information collection problem for linear programs for the first time, and derive a knowledge gradient policy which finds the marginal value of each measurement by solving a sequence of LPs. We prove that this policy is asymptotically optimal, and demonstrate its performance on a network flow problem.

Key words. optimal learning, stochastic linear programming, ranking and selection, knowledge gradient

AMS subject classifications. 90C05, 90C15, 90C31

1. Introduction. Consider the standard form of a linear program, given by

$$\begin{aligned} V(c) = \max_x \quad & c^T x \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0, \end{aligned} \tag{1.1}$$

and suppose that the vector c of objective coefficients is unknown. However, we have certain initial information about the problem that allows us to construct a Bayesian prior around c . Thus, we view c as a random variable whose distribution represents the uncertainty in our beliefs. We also have the ability to make noisy measurements of individual coefficients of c . Each measurement provides new information that can be used to update and improve our beliefs about the objective coefficients. Measurements are assumed to be time-consuming and expensive; we can choose any coefficient to measure at any time, but the total budget allocated for measurements is finite. However, our beliefs about different coefficients may be correlated, meaning that a single measurement could potentially provide useful information about more than one component of c . Our problem is to determine which coefficients to measure in order to come as close as possible to the optimal solution of the true LP in (1.1). Our decision at each time step is informed by what we have learned from prior measurements.

In general, *optimal learning* problems [44] arise when we are faced with an optimization problem with unknown parameters, but we are able to learn the values of these parameters through experiments. The present paper assumes that the original optimization problem is a linear program, and studies the efficient collection of information when this underlying LP has unknown objective coefficients. Examples of linear programming problems potentially involving optimal learning include the following:

1. *Logistics.* Network flow models are used to design shipment routes and schedules in supply chain management [24]. In global operations, such models are

*This work was supported in part by AFOSR contract FA9550-08-1-0195 through the Center for Dynamic Data Analysis.

[†]Robert H. Smith School of Business, University of Maryland, College Park MD 20742 (iryzhov@rhsmith.umd.edu).

[‡]Operations Research and Financial Engineering, Princeton University, Princeton NJ 08544 (powell@princeton.edu).

subject to much greater uncertainty about the reliability of suppliers in developing economies [52, 53]. Signing a contract with a vendor allows us to learn about the vendor’s service reliability through the costs we incur. Trying one vendor may teach us about other vendors in the same region (delays may be due e.g. to transportation and infrastructure problems).

2. *Marketing and production.* A production problem can be written as a linear program maximizing profit subject to resource constraints. The profit margins for new products are uncertain, and the exact optimal production strategy is unknown. However, we can run test markets (see [14] and the references therein on testing methods and strategies) to obtain improved estimates for some products. Each test market costs both time and money, and so only a small number of tests can be conducted.
3. *Agriculture.* In agricultural planning, linear programming can be used to obtain optimal crop rotations [17, 30]. The exact yield from planting certain fields is unknown, though prior estimates can be constructed via regression techniques on historical data. We can obtain new information through soil analysis on different fields. In particular, if analysis of one field suggests that crop yields there may be higher than expected, we might believe that crop yields from nearby fields may also be higher.

The basic idea of a linear program with stochastic parameters has been studied in operations research since [15] and the early theoretical work by [31, 40]. This problem has also been considered by the closely related field of stochastic programming; see [39], [34], [9] or [46] for an introduction. However, stochastic programming typically adopts the philosophy of classical stochastic optimization, with a fixed distribution for the unknown parameters. Once a distribution is specified, the solution is often approximated; see e.g. [8] for a formulation where the problem is solved repeatedly over many time periods. By contrast, we allow the the distribution of the unknown parameters to evolve over time due to the additional dimension of sequential learning.

Sequential optimal learning has been widely studied, but mostly for very simple underlying optimization models. In ranking and selection [3, 36, 37] and multi-armed bandit problems [2, 5, 26], there is a finite set of alternatives with unknown values, and the optimization consists of simply choosing the alternative with the highest value. More recent work such as [35] uses an empirical approach to weigh statistical and optimization considerations. A recent study by [48] considered a learning problem with a more complex optimization component, namely a shortest-path problem on a graph. Essentially, this is a special case of the problem posed in this paper: we now measure a single objective coefficient with the goal of learning about the optimal solution to the entire LP. We also use a more general learning model incorporating correlations into the Bayesian belief structure. Our goal is to bridge classical mathematical programming with optimal learning through a rigorous formula relating the value of learning about c to our eventual decision in the LP.

We approach the problem using the method of knowledge gradients (KG), originally developed by [28] and [29], and later studied by [20], in the context of the simple ranking and selection problem. In this approach, each measurement is chosen to maximize the expected value of information that can be collected at a given time step. The initial work on KG considered ranking and selection with independent Gaussian priors. It was found, however, that the KG concept can be used as a general methodology for learning problems, potentially involving more complicated statistical models such as unknown measurement noise [12], or correlated beliefs [21],

or linear belief models [42]. Other learning problems where KG algorithms have been studied include multi-armed bandits [47, 49], information collection on a graph [48], and Markov decision processes with unknown transition probabilities [50].

In the problem of learning in an LP, the KG algorithm consists of measuring the objective coefficient that yields the greatest expected improvement in our beliefs about the optimal value of the LP. Thus, the primary computational challenge is to find the expected value of the future LP that we will obtain after the measurement is completed. Conditionally, this future problem can be written as a parametric LP with a single stochastic parameter. Parametric linear programming typically considers deterministic problems [1, 22, 41], but the problem of computing the expectation can be viewed from the perspective of LP sensitivity analysis. Much of the work in sensitivity analysis (e.g. [32, 54]) concerns itself with finding a range in which a single optimal solution will remain optimal. We adapt the technique of [25] to compute the exact expected value of the parametric LP. This computation requires us to solve a sequence of LPs; the computational cost is potentially high, although numerical results suggest that exact computation adds significant value over approximations.

This paper makes the following contributions: 1) We present a new class of optimal learning problems where information is sequentially collected to improve a Bayesian belief about the optimal value of an LP. We use a multivariate Gaussian learning model which allows a single measurement to provide information about multiple unknown parameters of the LP. Section 2 describes this model. 2) We derive, in Section 3, a knowledge gradient algorithm for this problem class. The algorithm computes the expected value of information exactly, through an elegant synthesis of concepts from optimal learning, stochastic linear programming, and sensitivity analysis. 3) We prove, in Section 4, the asymptotic optimality of the KG algorithm as the number of measurements becomes large. 4) We present, in Section 5, numerical examples demonstrating the performance of the algorithm on a minimum-cost flow problem.

2. Mathematical model. We assume that the feasible region \mathcal{A} defined by $Ax = b$ and $x \geq 0$ is bounded. The optimal value of the LP in (1.1), for a fixed M -vector c of objective coefficients, is denoted by $V(c)$. Also let $x^*(c)$ represent the optimal solution of the LP, that is, the value of x that achieves the value $V(c)$. If there are multiple optimal solutions, we can let $x^*(c)$ be an arbitrary point in this set without loss of generality. The dual of the LP is given by

$$\begin{aligned} V(c) = \quad & \min_y \quad b^T y \\ & \text{s.t.} \quad A^T y - s = c \\ & \quad \quad s \geq 0. \end{aligned} \tag{2.1}$$

By strong duality, the optimal value of the dual for fixed c remains $V(c)$. We can let $y^*(c)$ and $s^*(c)$ represent the optimal dual solution as a function of the objective coefficients.

Denote by c^{true} the *true* objective function, which is unknown to us. We adopt the Bayesian philosophy, and take c^{true} to be a random vector whose distribution reflects the decision-maker's beliefs and uncertainty about the objective coefficients. We begin with a multivariate Gaussian prior on c^{true} , given by $c^{true} \sim \mathcal{N}(c^0, \Sigma^0)$. We can improve these estimates by making N sequential measurements of individual objective coefficients. When we choose to measure the j th coefficient, we make a noisy observation $\hat{c}_j \sim \mathcal{N}(c_j^{true}, \lambda_j)$ of the true value. We assume that the measurements are conditionally independent given the sequence of measurement decisions

j^0, \dots, j^{N-1} . We also assume that the noise variance λ_j is known, though we allow it to depend on j .

Each observation is used to update our beliefs about c^{true} . Denote by \mathcal{F}^n the sigma-algebra generated by our first n choices of individual components of c^{true} , as well as the observations we made as a result. Because the measurements and our prior beliefs all follow Gaussian distributions, the conditional distribution of c^{true} given \mathcal{F}^n is also multivariate Gaussian [16]. Let $\mathbf{E}^n = \mathbf{E}(\cdot | \mathcal{F}^n)$ represent the conditional expectation operator given \mathcal{F}^n . Then, we can define a vector $c^n = \mathbf{E}^n(c^{true})$ representing our beliefs about c^{true} “at time n ,” that is, immediately after exactly n measurements have been made. Similarly, let Σ^n denote the conditional covariance matrix of c^{true} at time n . It can be shown [21, 23] that

$$c^{n+1} = c^n + \frac{\hat{c}_j^{n+1} - c_j^n}{\lambda_j^n + \Sigma_{j^n j^n}^n} \Sigma^n e_{j^n}, \quad (2.2)$$

$$\Sigma^{n+1} = \Sigma^n - \frac{\Sigma^n e_{j^n} e_{j^n}^T \Sigma^n}{\lambda_j^n + \Sigma_{j^n j^n}^n}, \quad (2.3)$$

where j^n is the $(n+1)$ st coefficient chosen for measurement, and e_{j^n} is a vector of zeros with only the j^n th entry equal to 1. In (2.2) and (2.3), and further on throughout the paper, c_j^n denotes the j th component of the vector c^n , whereas $\Sigma_{i,j}^n$ refers to the (i,j) th entry of the matrix Σ^n .

In practice, λ_j is not known, and would have to be estimated from data (perhaps in a first stage of sampling). However, within our learning algorithm, assuming λ_j to be known greatly simplifies computation, while retaining the key feature of correlated beliefs and greatly increasing the power of a single measurement. Current work on learning with unknown sampling noise [11, 12] is mostly limited to independent beliefs.

Because we start with a multivariate prior, with correlations in our beliefs, a single measurement can potentially change our beliefs about the entire objective vector. Together, the vector c^n and the matrix Σ^n completely characterize our distribution of belief at time n . We can refer to these parameters as the *knowledge state*

$$k^n = (c^n, \Sigma^n).$$

If we choose to measure the j th coefficient at time n , we can write

$$k^{n+1} = K^{model}(k^n, j, \hat{c}_j^{n+1})$$

where the transition function (or “system model”) K^{model} is described by (2.2) and (2.3). The knowledge state encodes all the information that is available to us when we make our measurement decision at time n .

As in [48], our problem can be divided into two phases. First, in the *learning phase*, we sequentially make measurement decisions j^0, \dots, j^{N-1} . A *learning policy* π is a sequence of decision rules $J^{\pi,0}, \dots, J^{\pi,N-1}$, where each $J^{\pi,n}$ is a function mapping the knowledge state k^n to an index in the set $\{1, \dots, M\}$. Thus, a policy tells us what to measure in any possible circumstance. After all N measurements have been made, we must commit to a final decision (e.g. production strategy or agricultural plan) based on our final beliefs k^N . This is known as the *implementation phase*. This final implementation decision is represented by a function χ mapping the final knowledge state k^N to a point in the feasible region \mathcal{A} . Since χ is also a rule for making decisions rather than a fixed decision, we refer to it as an *implementation policy*.

We assume that the decision-maker is risk-neutral and seeks to maximize the value of the implementation decision in expectation. This objective (not to be confused with the objective function of the underlying LP) can be written as

$$\sup_{\pi} \sup_{\chi} \mathbf{E}^{\pi} (c^{true})^T \chi (k^N). \quad (2.4)$$

The notation \mathbf{E}^{π} denotes a conditional expectation given that $j^n = J^{\pi,n}(k^n)$ for all n , that is, the policy π is used for each measurement. The implementation policy is chosen to maximize the expectation of the true objective value $(c^{true})^T \chi$ over all possible outcomes of k^N and ensuing implementation decisions $\chi(k^N)$.

Risk-neutrality is a standard assumption in optimal learning models [13, 16, 26], which often use a dynamic programming formulation to characterize the optimal policy. We retain the same assumption, and focus on integrating linear optimization into this framework. Under risk-neutrality, (2.4) can be simplified by replacing the implementation policy $\chi(k^N)$ with $x^*(c^N)$, the solution obtained by simply plugging our final beliefs c^N into (1.1). The value of the LP with objective vector c^N is generally not the same as the expected value of the stochastic LP with the true objective vector c^{true} ; see [40] for a study of this issue and some conditions under which equality does hold. In our setting, these quantities are equal in expectation over the outcomes of the measurements under the same learning policy. At the final time N , the solution $x^*(c^N)$ is the best possible implementation decision: by maximizing our final guess of the optimal value, we also maximize (in expectation) the true objective value.

THEOREM 2.1. $\sup_{\pi} \sup_{\chi} \mathbf{E}^{\pi} (c^{true})^T \chi (k^N) = \sup_{\pi} \mathbf{E}^{\pi} V (c^N)$.

The proof is technical in nature, and can be found in the Appendix. Due to Theorem 2.1, the problem formulated in (2.4) reduces to choosing a policy π for making measurements during the learning phase, and (2.4) can be rewritten as

$$\sup_{\pi} \mathbf{E}^{\pi} V (c^N). \quad (2.5)$$

Our next result allows us to place a global upper bound on the objective value in (2.5). This bound will be used later in Section 4 to show the asymptotic optimality of the KG policy proposed in Section 3.

PROPOSITION 2.2. *For all n and all knowledge states $k^n = (c^n, \Sigma^n)$,*

$$V (c^n) \leq \mathbf{E}^n V (c^{true}) \quad \textit{almost surely.} \quad (2.6)$$

Proof. The optimal value function V is convex in c (concave for minimization problems; see e.g. [33], Lemma 5). Applying Jensen's inequality together with the definition of c^n , we obtain

$$V (c^n) = V (\mathbf{E}^n c^{true}) \leq \mathbf{E}^n V (c^{true}),$$

as required. \square

COROLLARY 2.3. *For all π ,*

$$\mathbf{E}^{\pi} V (c^N) \leq \mathbf{E} V (c^{true}),$$

that is, there is a global upper bound on the objective value achieved by any policy.

Proof. Taking the expectation of both sides of (2.6) yields $\mathbf{E}^{\pi} V (c^N) \leq \mathbf{E}^{\pi} V (c^{true})$ for any policy π . However, c^{true} does not depend on the policy π , whence $\mathbf{E}^{\pi} V (c^{true}) = \mathbf{E} V (c^{true})$ for all π , and the desired result follows. \square

The optimal learning policy π^* that maximizes (2.5) can be characterized using dynamic programming notation. The objective value achieved by following the optimal learning policy from time n onward can be described using a form of Bellman's equation [4] adapted for learning problems, given by

$$\mathcal{V}^{*,n}(k^n) = \max_j \mathbf{E}_j^n \mathcal{V}^{*,n+1}(K^{model}(k^n, j, \hat{c}_j^{n+1})), \quad (2.7)$$

$$\mathcal{V}^{*,N}(k^N) = V(c^N). \quad (2.8)$$

The notation \mathbf{E}_j^n denotes an expectation given \mathcal{F}^n and the decision to measure the j th coefficient at time n . If there are no measurements remaining, we make the optimal implementation decision $x^*(c^N)$. Otherwise, we make the decision that puts us in the best position for following the optimal policy starting at time $n+1$.

Although it is conceptually useful to characterize the optimal learning policy in this way, the dynamic program given by (2.7) and (2.8) is computationally intractable, because the knowledge state k^n is multi-dimensional and continuous. In the next section, we propose a heuristic learning policy that yields a computable algorithm.

3. The knowledge gradient algorithm. The knowledge gradient (KG) concept was originally advanced by [29] and later developed by [10], [11] and [20] for the ranking and selection problem. The KG approach consists of a one-period look-ahead, viewing each time period as if it were the last, and making the measurement that would be optimal under those circumstances. In our problem, the optimal measurement decision at time $N-1$ is given by

$$J^{*,N-1}(k^{N-1}) = \arg \max_j \mathbf{E}_j^{N-1} V(c^N). \quad (3.1)$$

Equivalently, (3.1) can be written in terms of the difference

$$J^{*,N-1}(k^{N-1}) = \arg \max_j \mathbf{E}_j^{N-1} (V(c^N) - V(c^{N-1})),$$

because $V(c^{N-1})$ is \mathcal{F}^{N-1} -measurable and has no effect on the argmax.

The KG decision rule is given by

$$J^{KG,n}(k^n) = \arg \max_j \nu_j^{KG,n} \quad (3.2)$$

where

$$\nu_j^{KG,n} = \mathbf{E}_j^n (V(c^{n+1}) - V(c^n)). \quad (3.3)$$

Under the KG policy, we always measure the coefficient that yields the greatest expected improvement between our current estimate $V(c^n)$ of the optimal LP value, and the future estimate $V(c^{n+1})$ that will result from the measurement. The term "knowledge gradient" arises because the expected improvement is written as a difference. Observe that, by definition, KG is the optimal learning policy if $N=1$. Also note that $J^{KG,n}$ only depends on n through k^n , and thus we can drop the time index n from the decision rule.

At time n , the quantity $V(c^n)$ can be found by solving a deterministic LP with objective vector c^n . However, the quantity $\mathbf{E}_j^n (V(c^{n+1}))$ is the expected value of a linear program with stochastic parameters, since c^{n+1} is random at time n . The remainder of this section explains how this quantity can be computed exactly.

3.1. Derivation. From [21], we know that the conditional distribution of c^{n+1} , given \mathcal{F}^n as well as the decision to measure j at time n , can be expressed by the equation

$$c^{n+1} = c^n + \frac{\Sigma^n e_j}{\sqrt{\lambda_j + \Sigma_{jj}^n}} \cdot Z,$$

where $Z \sim \mathcal{N}(0, 1)$. Thus, we can write

$$\mathbf{E}_j^n V(c^{n+1}) = \mathbf{E}V\left(c^n + \frac{\Sigma^n e_j}{\sqrt{\lambda_j + \Sigma_{jj}^n}} \cdot Z\right).$$

For a fixed value z of Z , the expression inside the expectation can be rewritten as

$$\begin{aligned} V(c^n + z\Delta c_j^n) = \max_x & (c^n + z\Delta c_j^n)^T x \\ \text{s.t.} & Ax = b \\ & x \geq 0, \end{aligned} \tag{3.4}$$

where

$$\Delta c_j^n = \frac{\Sigma^n e_j}{\sqrt{\lambda_j + \Sigma_{jj}^n}}.$$

The LP in (3.4) is a perturbed version of (1.1). The perturbation is random (and follows a standard normal distribution), and the expected value of the perturbed LP is given by

$$\mathbf{E}V(c^n + Z\Delta c_j^n) = \int_{-\infty}^{\infty} V(c^n + z\Delta c_j^n) \phi(z) dz$$

with ϕ being the standard normal pdf. Now observe that

$$\int_{-\infty}^{\infty} V(c^n + z\Delta c_j^n) \phi(z) dz = \int_{-\infty}^{\infty} (c^n + z\Delta c_j^n)^T (x^*(c^n + z\Delta c_j^n)) \phi(z) dz.$$

The quantity $(c^n + z\Delta c_j^n)^T (x^*(c^n + z\Delta c_j^n))$ is a piecewise linear function of z . This is discussed by [25], but can also be seen from the following. If z is inside a certain range, the optimal solution $x^*(c^n + z\Delta c_j^n)$ will remain the same, and the value $(c^n + z\Delta c_j^n)^T (x^*(c^n + z\Delta c_j^n))$ will be linear in z within that range. Figure 3.1 gives an example in two dimensions. The red lines represent the level curves of the objective function. Varying z has the effect of rotating these curves. For small values of z , the tangent point does not change; it will only change once the perturbed level curves become parallel to one of the faces of the feasible region.

The value of z for which $x^*(c^n + z\Delta c_j^n)$ does not change is known as the invariant support set. Thus, the expectation $\mathbf{E}V(c^n + Z\Delta c_j^n)$ can be rewritten as a sum of integrals over all possible invariant support sets. We can let $-\infty = z_1 < z_2 < \dots < z_I = \infty$ be a finite set of points such that $x^*(c^n + z\Delta c_j^n)$ is constant for $z \in (z_i, z_{i+1})$. Let x_i be the optimal solution of the LP for $z \in (z_i, z_{i+1})$. Then,

$$\begin{aligned} \mathbf{E}V(c^n + Z\Delta c_j^n) &= \sum_i \int_{z_i}^{z_{i+1}} (c^n + z\Delta c_j^n)^T x_i \phi(z) dz \\ &= \sum_i (c^n)^T x_i \int_{z_i}^{z_{i+1}} \phi(z) dz + \sum_i (\Delta c_j^n)^T x_i \int_{z_i}^{z_{i+1}} z\phi(z) dz \tag{3.5} \\ &= \sum_i (c^n)^T x_i (\Phi(z_{i+1}) - \Phi(z_i)) + \sum_i (\Delta c_j^n)^T x_i (\phi(z_i) - \phi(z_{i+1})) \end{aligned}$$

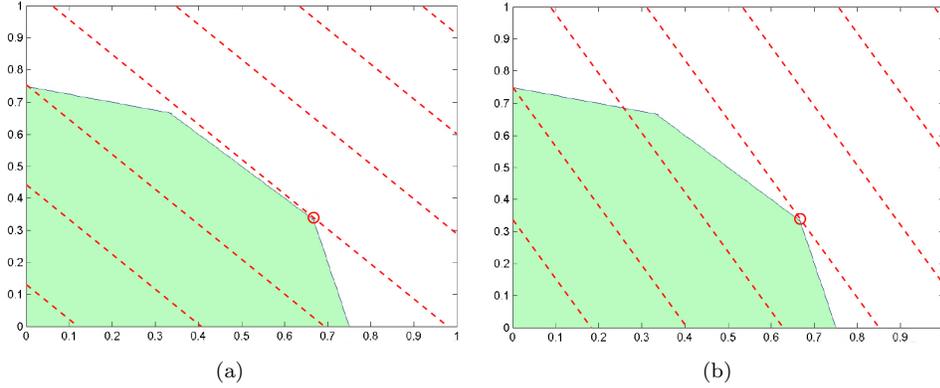


FIG. 3.1. Example of an LP with two decision variables x_1 (hor. axis) and x_2 (vert. axis), with a polyhedral feasible region (shaded). The dashed lines represent level curves of the perturbed objective function for two different values of z . The optimal solution (circled) remains constant for a range of values of z .

with Φ being the standard normal cdf. The first sum in (3.5) requires integrating the density of z over intervals, yielding the differences $\Phi(z_{i+1}) - \Phi(z_i)$ in (3.6). The second sum in (3.5) can be computed by observing that $\int_{z_i}^{\infty} z\phi(z) dz = \phi(z_i)$.

The sum of integrals in the first line of (3.6) recalls a similar expression in [31], though that study did not consider the issue of computation. We can write this more concisely as

$$\mathbf{EV}(c^n + Z\Delta c_j^n) = \sum_i a_i (\Phi(z_{i+1}) - \Phi(z_i)) + b_i (\phi(z_i) - \phi(z_{i+1}))$$

where $a_i = (c^n)^T x_i$ and $b_i = (\Delta c_j^n)^T x_i$. Using the analysis of [21], we can rewrite this as

$$\mathbf{EV}(c^n + Z\Delta c_j^n) = \left(\max_i a_i\right) + \sum_i (b_{i+1} - b_i) f(-|z_i|)$$

where $f(z) = z\Phi(z) + \phi(z)$. Now observe that $\max_i a_i = \max_i (c^n)^T x_i$. For any value of z , including $z = 0$, we can find i such that the point x_i maximizes $c^n + z\Delta c_j^n$ over the feasible region. Taking $z = 0$, this means that there is a point x_i satisfying $x_i = \arg \max_{x \in \mathcal{A}} (c^n)^T x$, whence

$$\max_i (c^n)^T x_i = \max_x (c^n)^T x = V(c^n),$$

whence (3.3) becomes

$$\nu_j^{KG,n} = \sum_i (b_{i+1} - b_i) f(-|z_i|). \quad (3.7)$$

The problem of computing the KG factor of the j th coefficient thus reduces to the problem of finding the breakpoints z_i of the piecewise linear function, as well as the corresponding optimal solutions x_i . We now present an algorithm for finding these quantities.

3.2. Computation. We will compute the breakpoints z_i iteratively by starting with $x^*(c^n)$, the optimal solution to (3.4) when $z = 0$, and solving two sequences of LPs, one where z increases from zero, and one where z decreases. We construct these sequences by adapting the procedure developed for LP sensitivity analysis by [25].

The easiest way to visualize the procedure is to interpret it as moving along a sequence of adjacent extreme points of the feasible region \mathcal{A} , with each extreme point x_i being optimal for $z \in (z_i, z_{i+1})$. For greater clarity, let us suppose that all LPs in this section are solved using the simplex method, so that the optimal solution will always be a corner point. At the end of this section, we will briefly discuss how this supposition can be relaxed. For now, however, we simply note that it is not necessary for the algorithm, and is included purely to make our presentation clearer.

We also note that all computations described in this section are made at time n . While $c^n = \mathbf{E}^n c^{true}$ is a random vector, it is \mathcal{F}^n -measurable, and so its exact value is known at time n . Thus, our algorithm views c^n as a deterministic quantity.

Recall that $x^*(c^n)$ solves (3.4) for $z = 0$. First, we determine whether zero is itself a breakpoint of the piecewise linear function. This can be done by solving two LPs

$$\begin{aligned} z_- = \min_{y,s,z} \quad & z \\ \text{s.t.} \quad & A^T y - s - z \Delta c_j^n = c^n \\ & x^*(c^n)^T s = 0 \\ & s \geq 0, \end{aligned} \tag{3.8}$$

and

$$\begin{aligned} z_+ = \max_{y,s,z} \quad & z \\ \text{s.t.} \quad & A^T y - s - z \Delta c_j^n = c^n \\ & x^*(c^n)^T s = 0 \\ & s \geq 0. \end{aligned} \tag{3.9}$$

It is shown in [45] that these two problems find the smallest and largest values of z , respectively, for which $x^*(c^n)$ is still an optimal solution of (3.4). The feasible region in each problem merely ensures that there is a feasible solution of the dual of (3.4) that maintains complementary slackness with $x^*(c^n)$, meaning that $x^*(c^n)$ is optimal. If $z_- < 0 < z_+$, then zero is not a breakpoint, but the numbers z_-, z_+ are. In this situation, c^n corresponds to the tangent line in Figure 3.1.

We will now build one sequence of breakpoints monotonically decreasing from z_- , and another sequence monotonically increasing from z_+ . We handle the two sequences separately, since one will involve minimizing z over invariant support sets, as in (3.8), whereas the other will maximize, as in (3.9). However, this is the only difference between the two types of computations. We first discuss the decreasing sequence, starting with z_- . As before, Figure 3.2 provides a visual aid in two dimensions.

Suppose that $z_- = 0$. Then, zero is a breakpoint, and the level curve of c^n is tangent to a face of the feasible region, as in Figure 3.2(a). In this case, there are multiple optimal extreme-point solutions of (3.4) at $z = 0$, of which one is $x^*(c^n)$ itself, and another is the optimal solution to the LP

$$\begin{aligned} V_l(c^n) = \min_x \quad & (\Delta c_j^n)^T x \\ \text{s.t.} \quad & Ax = b \\ & (s^*(c^n))^T x = 0 \\ & x \geq 0. \end{aligned} \tag{3.10}$$

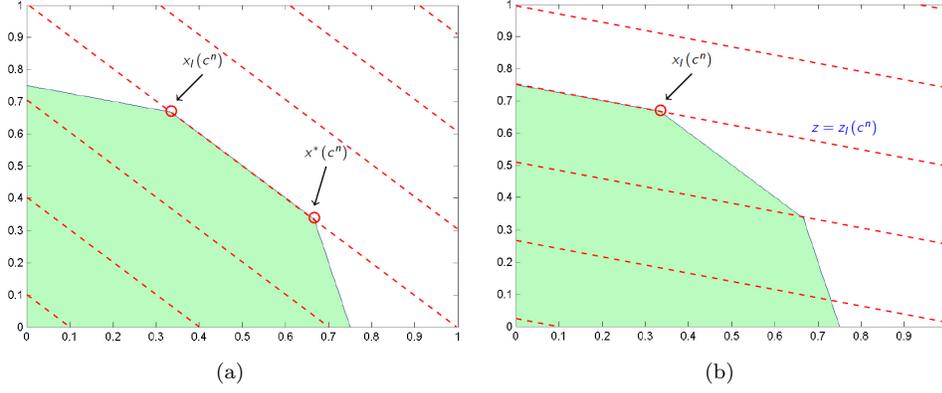


FIG. 3.2. Illustration of the relationship between $x^*(c^n)$, $x_l(c^n)$ and $z_l(c^n)$. The extreme point $x_l(c^n)$ is adjacent to $x^*(c^n)$ and is optimal for $z \in (z_l(c^n), 0)$.

Let $x_l(c^n)$ denote this point. The quantity $(\Delta c_j^n)^T(x_l(c^n))$ is precisely the left derivative of the piecewise linear function at the breakpoint $z = 0$. (The right derivative is $x^*(c^n)$ itself.) Among all the corner points that maximize (3.4) at $z = 0$, $x_l(c^n)$ is located furthest in the direction of rotation of the level curves. It will continue to be optimal for $z \in (z_l, 0)$ for some other breakpoint $z_l < 0$.

This next breakpoint z_l is the optimal value of the LP

$$\begin{aligned} z_l(c^n) = \min_{y,s,z} \quad & z \\ \text{s.t.} \quad & A^T y - s - z \Delta c_j^n = c^n \\ & (x_l(c^n))^T s = 0 \\ & s \geq 0. \end{aligned} \quad (3.11)$$

The LP in (3.11) is almost identical to that in (3.8), with the only difference being that $x^*(c^n)$ is replaced by $x_l(c^n)$. Figure 3.2 gives a visual illustration of the relationship between the solutions $x^*(c^n)$ and $x_l(c^n)$, and the breakpoint $z_l(c^n)$.

It is now clear that we can iterate (3.10) and (3.11) to find a descending sequence of breakpoints. We replace c^n by $c^n + z_l \Delta c_j^n$ and $x^*(c^n)$ by $x_l(c^n)$, then solve (3.10) to find a new corner point, then solve (3.11) to find the next breakpoint. The exact details are described in Step 3, Figure 3.3.

To find breakpoints greater than zero, we follow the same procedure, but maximize instead of minimizing. If $z_+ = 0$, then zero is still a breakpoint, and there are still multiple optimal extreme-point solutions of (3.4) at $z = 0$. This time, the next extreme point is the optimal solution to

$$\begin{aligned} V_u(c^n) = \max_x \quad & (\Delta c_j^n)^T x \\ \text{s.t.} \quad & Ax = b \\ & (s^*(c^n))^T x = 0 \\ & x \geq 0. \end{aligned} \quad (3.12)$$

We let $x_u(c^n)$ be the value of x that solves (3.12), and find the next breakpoint $z_u > 0$

-
- 1:** Solve (3.4) with $z = 0$ to obtain $x^*(c^n)$, $y^*(c^n)$ and $s^*(c^n)$.
 - 2:** Solve the LPs given by (3.8) and (3.9), and let z_- , z_+ be their optimal values.
 - 2a:** If $z_- < 0 < z_+$, let $\bar{z} = (z_-, z_+)$ and $\bar{x} = (x^*(c^n))$.
 - 2b:** Otherwise, let $\bar{z} = (0)$ and let \bar{x} be the empty set.
 - 3:** While $\min(\bar{z}) > -\infty$, do the following:
 - 3a:** Let $c' = c^n + \min(\bar{z}) \cdot \Delta c_j^n$.
 - 3b:** Find $V_l(c')$ and $z_l(c')$ using (3.10) and (3.11).
 - 3c:** Update $\bar{z} \leftarrow (\min(\bar{z}) + z_l(c'), \bar{z})$ and $\bar{x} \leftarrow (x_l(c'), \bar{x})$.
 - 4:** While $\max(\bar{z}) < \infty$, do the following:
 - 4a:** Let $c' = c^n + \max(\bar{z}) \cdot \Delta c_j^n$.
 - 4b:** Find $V_u(c')$ and $z_u(c')$ using (3.12) and (3.13).
 - 4c:** Update $\bar{z} \leftarrow (\bar{z}, \max(\bar{z}) + z_u(c'))$ and $\bar{x} \leftarrow (\bar{x}, x_u(c'))$.
-

FIG. 3.3. Algorithm for finding the vector \bar{z} of breakpoints and the set \bar{x} of corresponding invariant solutions.

by computing

$$\begin{aligned}
 z_u(c^n) = \max_{y,s,z} \quad & z \\
 \text{s.t.} \quad & A^T y - s - z \Delta c_j^n = c^n \\
 & (x_u(c^n))^T s = 0 \\
 & s \geq 0.
 \end{aligned} \tag{3.13}$$

Step 4 of Figure 3.3 shows how this procedure can be iterated. By running the algorithm in Figure 3.3, we will obtain a vector \bar{z} of the breakpoints, together with a set \bar{x} containing the corresponding invariant solutions. These quantities can then be used to compute the KG formula given in (3.7).

The algorithm is guaranteed to terminate in finite time, because there is only a finite number of extreme points (see [6], Corollary 2.1). Furthermore, the breakpoints are already returned sorted. However, the algorithm can be computationally expensive, requiring us to solve a sequence of linear programs for every coefficient of the objective function at every time step n . The advantage is that the KG factor $\nu_j^{KG,n}$ is computed exactly.

We now briefly remark on the effect of allowing optimal solutions that are not corner points. In higher dimensions, it may be possible for (3.10) and (3.12) to produce points that are on the boundary of a face of \mathcal{A} , but not necessarily on a corner. For example, in three dimensions, an interior-point method may cause (3.10) to yield a point x_l on the line segment connecting two corners of \mathcal{A} . However, for any such point, we can find a corner point \bar{x}_l that also optimizes (3.10), and also the unperturbed problem, since (3.10) requires complementarity with $s^*(c^n)$. The constraint $x_l^T s = 0$ in (3.10) can be replaced by $(c^n + z \Delta c_j^n)^T x_l = b^T y$, which in turn can be replaced by $(c^n + z \Delta c_j^n)^T \bar{x}_l = b^T y$, since $(c^n)^T x_l = (c^n)^T \bar{x}_l$ and $(\Delta c_j^n)^T x_l = (\Delta c_j^n)^T \bar{x}_l$ by the preceding discussion. Thus, (3.11) will yield the same solution regardless of whether or not x_l is a corner point.

Observe that, in the above analysis, the procedure for calculating breakpoints is independent of our assumptions on the prior and sampling distributions. These assumptions come into play only when we plug those breakpoints into (3.7). This suggests that the algorithm in Figure 3.3 could potentially be applied to other types of prior distributions on c (e.g. independent gamma priors combined with exponential

observations, or other conjugate learning models in [16]). However, we focus on the multivariate Gaussian distribution because it provides an elegant and powerful framework for modeling correlations in our beliefs.

4. Asymptotic optimality. In this section, we prove that the KG algorithm laid out in Section 3 possesses the property of asymptotic optimality. The precise meaning of this property is given in Section 4.1, along with other definitions and preliminaries required for the proof. The proof itself is given in Section 4.2. It is based on the convergence analysis of [19]: essentially, we show that the KG policy avoids getting “stuck” at a knowledge state where some, but not all components of c are known perfectly. Given infinitely many measurements, KG eventually reaches a state where all of c is known.

For the purpose of our convergence analysis, we suppose that the feasible region \mathcal{A} can be written as a polyhedron in the form $Ax \leq b, x \geq 0$. We further assume that this polyhedron has dimension M .

4.1. Preliminaries. We define the space of knowledge states as

$$\mathcal{K} = \{(c, \Sigma) \mid c \in \mathbb{R}^M, \Sigma \in S_+^M, \Sigma_{jj} > 0 \forall j\},$$

where S_+^M is the space of $M \times M$ positive-semidefinite matrices. The space \mathcal{K} includes all possible posterior distributions at all times n . Without loss of generality, we assume that Σ^0 is a full-rank covariance matrix with $\Sigma_{jj}^0 > 0$ for all j , that is, we do not have perfect information about any objective coefficient at the beginning. The space \mathcal{K} thus includes all possible mean vectors and covariance matrices that we will observe over the course of a finite number of measurements. The closure of this space is

$$\bar{\mathcal{K}} = \{(c, \Sigma) \mid c \in \mathbb{R}^M, \Sigma \in S_+^M\},$$

which includes all possible legitimate covariance matrices Σ , including those where $\Sigma_{jj} = 0$ for some j . Such a knowledge state can be reachable by measuring j infinitely often.

We define convergence in the space of knowledge states as follows. Suppose that (k^n) is a sequence of knowledge states $(c^n, \Sigma^n) \in \bar{\mathcal{K}}$, and $k = (c^{(k)}, \Sigma^{(k)})$ is a knowledge state in $\bar{\mathcal{K}}$. We say that $k^n \rightarrow k$ if $c^n \rightarrow c^{(k)}$ and $\Sigma_{j,j'}^n \rightarrow \Sigma_{j,j'}^{(k)}$ for all j, j' . Our parametrization is different from the mean-value parametrization used in [19]. However, our definition of convergence in \mathcal{K} is equivalent to convergence in the mean-value parameters. This is because both our sampling density and our posterior density are multivariate Gaussian, and thus come from an exponential family. It is known (e.g. from [7]) that, for an exponential family, there is a continuous bijection between the natural parameters of the distribution, and the mean-value parameters. Thus, every pair $(c^{(k)}, \Sigma^{(k)})$ for $k \in \bar{\mathcal{K}}$ corresponds to exactly one set of mean-value parameters. Furthermore, a sequence in $\bar{\mathcal{K}}$ converges to a point in $\bar{\mathcal{K}}$ if and only if the corresponding sequence of mean-value parameters converges to the mean-value parametrization of the limit. For our purposes, however, it is much more convenient to work with c^n and Σ^n , because we use these parameters to define the KG policy.

We now give the meaning of asymptotic optimality. The risk function

$$R(c, x) = V(c) - (c)^T x$$

represents the loss incurred (for a particular choice of the objective coefficients) by choosing $x \in \mathcal{A}$ as the implementation decision instead of solving for $V(c)$. The work

by [19] defines asymptotic optimality as

$$\lim_{N \rightarrow \infty} \mathbf{E} \left(\min_x \mathbf{E}^N R(c^{true}, x) \right) = \mathbf{E} \left(\min_x \mathbf{E} (R(c^{true}, x) | c^{true}) \right), \quad (4.1)$$

that is, the minimum-risk decision at time N will achieve the lowest risk possible if c^{true} is perfectly known, in the limit as $N \rightarrow \infty$. Every expectation in this section can be assumed to be under the KG policy.

The left-hand side of (4.1) can be rewritten as

$$\begin{aligned} \lim_{N \rightarrow \infty} \mathbf{E} \left(\min_x \mathbf{E}^N R(c^{true}, x) \right) &= \lim_{N \rightarrow \infty} \mathbf{E} \left(\min_x \mathbf{E}^N V(c^{true}) - (c^{true})^T x \right) \\ &= \lim_{N \rightarrow \infty} \mathbf{E} \left(\mathbf{E}^N V(c^{true}) + \min_x \left(-\mathbf{E}^N (c^{true})^T x \right) \right) \\ &= \lim_{N \rightarrow \infty} \mathbf{E} V(c^{true}) - \mathbf{E} \max_x (c^N)^T x \\ &= \mathbf{E} V(c^{true}) - \lim_{N \rightarrow \infty} \mathbf{E} V(c^N). \end{aligned}$$

Analogously, it can be shown that $\mathbf{E}(\min_x \mathbf{E}(R(c^{true}, x) | c^{true})) = 0$, whence (4.1) becomes

$$\lim_{N \rightarrow \infty} \mathbf{E} V(c^N) = \mathbf{E} V(c^{true}). \quad (4.2)$$

Thus, the meaning of asymptotic optimality is in line with our objective function in (2.5). The objective value achieved by an asymptotically optimal learning policy converges to the global upper bound in Corollary 2.3.

The last preliminaries bring concepts from [19] to our LP setting. We define three functions

$$\begin{aligned} h(k^n, j) &= \min_x \mathbf{E}^n R(c^{true}, x) - \mathbf{E}_j^n \left(\min_x \mathbf{E}^{n+1} R(c^{true}, x) \right), \\ g(k^n, j) &= \min_x \mathbf{E}^n R(c^{true}, x) - \mathbf{E}^n \left(\min_x \mathbf{E}^n (R(c^{true}, x) | c_j^{true}) \right), \\ \bar{g}(k^n) &= \min_x \mathbf{E}^n R(c^{true}, x) - \mathbf{E}^n \left(\min_x \mathbf{E}^n (R(c^{true}, x) | c^{true}) \right). \end{aligned}$$

Each function represents a decrease in risk under different conditions. The function h is the one-period decrease in risk obtained by measuring the j th objective coefficient at time n . The function g is the decrease in risk that would be achieved if we were given the exact value of the j th coefficient after reaching a given knowledge state. Finally, \bar{g} represents the decrease in risk achieved if we were given the exact value of c^{true} .

Note that the minimum operators in the definitions of h , g and \bar{g} are defining random variables. For example, in the definition of h , let $R_x^{n+1} = \mathbf{E}^{n+1} R(c^{true}, x)$. For each fixed x , R_x^{n+1} is a random variable. Then, $\bar{R}^{n+1}(\omega) = \min_x R_x^{n+1}(\omega)$ defines another random variable \bar{R}^{n+1} whose value on the sample path ω is a minimum of the realized values $R_x^{n+1}(\omega)$ over the fixed feasible region.

Our last definitions relate these concepts to the notion of convergence. For each coefficient j , let

$$\mathcal{M}_j = \{k \in \bar{\mathcal{K}} | \exists (k^n) \subseteq \mathcal{K} : k^n \rightarrow k, g(k^n, j) \rightarrow 0\}$$

be the set of all knowledge states for which we gain nothing (no decrease in risk) by knowing the exact value of c_j . Similarly, let

$$\mathcal{B}_k = \{j | k \in \mathcal{M}_j\}$$

be the set of all such j for the fixed knowledge state k . Finally, let

$$\mathcal{M}_* = \{k \in \bar{\mathcal{K}} \mid \forall (k^n) \subseteq \mathcal{K} \text{ with } k^n \rightarrow k, g(k^n, j) \rightarrow 0 \forall j\}$$

be the set of all knowledge states for which nothing is gained by knowing all of c exactly.

4.2. Main result. The work by [19] provides sufficient conditions under which the asymptotic optimality property from (4.2) holds. We first restate these conditions in terms of our LP problem.

THEOREM 4.1. *Suppose that the following conditions hold:*

1. (Assumption 1 from [19]) *If k^n is a sequence of knowledge states with $k^n \rightarrow k$ and $g(k^n, j) \rightarrow 0$ for all j , then $\bar{g}(k^n) \rightarrow 0$ also.*
2. (Theorem 1 from [19]) *If $k \notin \mathcal{M}_*$, then k has an open neighbourhood $U \subseteq \bar{\mathcal{K}}$ such that*

$$\sup_{k' \in U} P(J^{KG}(k') \in \mathcal{B}_k) < 1. \quad (4.3)$$

Then, the KG policy of Section 3 is asymptotically optimal in the sense of (4.2).

The first condition of Theorem 4.1 does not depend on any policy, but rather is a regularity condition ensuring that our problem is well-behaved. If, in the limit, there is no benefit from learning the exact value c_j^{true} of any individual coefficient j , then there should be no benefit from learning the exact value of the entire vector c^{true} at once. The second condition, expressed in (4.3), ensures that the KG policy avoids getting stuck measuring coefficients whose values are already known.

First condition. We now calculate

$$\begin{aligned} h(k^n, j) &= \mathbf{E}^n V(c^{true}) - \max_x (c^n)^T x - \mathbf{E}^n V(c^{true}) + \mathbf{E}_j^n \max_x (c^{n+1})^T x \\ &= \mathbf{E}_j^n (V(c^{n+1}) - V(c^n)) \\ &= \nu_j^{KG, n}. \end{aligned} \quad (4.4)$$

Thus, the KG policy as defined in (3.2) and (3.3) maximizes the one-period decrease in risk. Repeating the same computations as in (4.4), we find that

$$g(k^n, j) = \mathbf{E}^n \left(\max_x \mathbf{E}^n \left((c^{true})^T x \mid c_j^{true} \right) - V(c^n) \right). \quad (4.5)$$

Because c_j^{true} is correlated with the values of other $c_{j'}^{true}$, we find that

$$\begin{aligned} \mathbf{E}^n \left((c^{true})^T x \mid c_j^{true} \right) &= \sum_{j'=1}^M x_{j'} \mathbf{E}^n (c_{j'}^{true} \mid c_j^{true}) \\ &= c_j^{true} x_j + \sum_{j' \neq j} x_{j'} \left(c_{j'}^n + \frac{\Sigma_{j,j'}^n}{\Sigma_{j,j}^n} (c_j^{true} - c_j^n) \right). \end{aligned}$$

The time- n marginal distribution of c_j^{true} is $\mathcal{N}(c_j^n, \Sigma_{j,j}^n)$, whence

$$\begin{aligned} \mathbf{E}^n \left(\max_x \mathbf{E}^n \left((c^{true})^T x \mid c_j^{true} \right) \right) &= \mathbf{E} \left(\max_x x_j \left(c_j^n + \sqrt{\Sigma_{j,j}^n} Z \right) + \sum_{j' \neq j} x_{j'} \left(c_{j'}^n + \frac{\Sigma_{j,j'}^n}{\Sigma_{j,j}^n} \sqrt{\Sigma_{j,j}^n} Z \right) \right) \\ &= \mathbf{E} \left(\max_x \sum_{j'=1}^M c_{j'}^n x_{j'} + x_j \frac{\Sigma_{j,j'}^n}{\sqrt{\Sigma_{j,j}^n}} Z \right). \end{aligned}$$

It follows that

$$g(k^n, j) = \mathbb{E}V(c^n + Z\Delta\bar{c}_j^n) - V(c^n)$$

with $\Delta\bar{c}_j^n = \frac{\Sigma^n e_j}{\sqrt{\Sigma_{j,j}^n}}$. Repeating the arguments of Section 3.1, we find that

$$g(k^n, j) = \sum_i (\bar{b}_{i+1} - \bar{b}_i) f(-|\bar{z}_i|). \quad (4.6)$$

As before, $-\infty = \bar{z}_1 < \bar{z}_2 < \dots < \bar{z}_{I'} < \infty$ is a finite set of points for which $x^*(c^n + z\Delta\bar{c}_j^n)$ is constant for $z \in (\bar{z}_i, \bar{z}_{i+1})$, \bar{x}_i is the optimal solution of the LP for $z \in (\bar{z}_i, \bar{z}_{i+1})$, and $\bar{b}_i = (\Delta\bar{c}_j^n)^T \bar{x}_i$.

PROPOSITION 4.2. *Suppose that k^n is the current knowledge state. Then, $g(k^n, j) = 0$ if and only if $\Sigma_{j,j}^n = 0$. Also, $h(k^n, j) = 0$ if and only if $\Sigma_{j,j}^n = 0$.*

Proof. Suppose that $\Sigma_{j,j}^n = 0$. Because the marginal distribution of c_j^{true} given k^n is $\mathcal{N}(c_j^n, \Sigma_{j,j}^n)$, it follows that $c_j^{true} = c_j^n$ almost surely given \mathcal{F}^n . Consequently, the right-hand side of (4.5) is zero.

Suppose now that $\Sigma_{j,j}^n > 0$. Then, $g(k^n, j)$ is given by (4.6). We claim that the right-hand side of (4.6) must be strictly positive. To see this, first observe that the function f has no zeros on the real line [48]. Furthermore, we have $\bar{b}_{i+1} \geq \bar{b}_i$ for all i , so the only way (4.6) can equal zero is if

$$\sum_{j'=1}^M \bar{x}_{i,j'} \Sigma_{j,j'}^n = \sum_{j'=1}^M \bar{x}_{i',j'} \Sigma_{j,j'}^n$$

for all $i, i' \in \{1, \dots, I'\}$. It follows that

$$\begin{pmatrix} \bar{x}_{1,1} & \dots & \bar{x}_{1,M} & 1 \\ \dots & \dots & \dots & \dots \\ \bar{x}_{I',1} & \dots & \bar{x}_{I',M} & 1 \end{pmatrix} \begin{pmatrix} \Sigma_{j,1}^n \\ \dots \\ \Sigma_{j,M}^n \\ -\beta \end{pmatrix} = 0 \quad (4.7)$$

for some constant β . However, the polyhedron \mathcal{A} is assumed to have dimension M , which means that it has exactly $M+1$ affinely independent extreme points [43]. Thus, the matrix in (4.7) must contain an invertible $M+1 \times M+1$ submatrix. Because $\Sigma_{j,j}^n > 0$ by assumption, this submatrix cannot yield zero when multiplied by the vector in (4.7). We conclude that $g(k^n, j) = 0$ if and only if $\Sigma_{j,j}^n = 0$. The same argument can be used to show that $h(k^n, j) = 0$ if and only if $\Sigma_{j,j}^n = 0$. \square

We now state a technical lemma that will be used to verify the first condition of Theorem 4.1. The lemma will be used to establish certain continuity conditions on h , g and \bar{g} , which is crucial to the proof of asymptotic optimality. The proof of the lemma is given in the Appendix.

LEMMA 4.3. *Let a^n and B^n be sequences in \mathbb{R}^I and $\mathbb{R}^{I \times I'}$, respectively, such that $a^n \rightarrow a$ and $B_{i,i'}^n \rightarrow B_{i,i'}$ for all i, i' . Define*

$$y^n = \mathbf{E} \max_i a_i^n + \sum_{i'=1}^{I'} B_{i,i'}^n Z_{i'},$$

$$y = \mathbf{E} \max_i a_i + \sum_{i'=1}^{I'} B_{i,i'} Z_{i'},$$

where $Z_1, \dots, Z_{I'}$ are i.i.d. $\mathcal{N}(0, 1)$ random variables. Then, $y^n \rightarrow y$.

COROLLARY 4.4. *The function $h(\cdot, j)$ is continuous for all j . That is, if $k^n \rightarrow k$, then $h(k^n, j) \rightarrow h(k, j)$.*

Proof. Recall that $h(k^n, j) = \mathbf{E}V(c^n + Z\Delta c_j^n) - V(c^n)$. Because any LP always has a basic optimal solution, V can be rewritten as a finite maximum over extreme points, that is,

$$V(c^n + Z\Delta c_j^n) = \max_i (c^n)^T x_i + Z(\Delta c_j^n)^T x_i,$$

which means that the expected value $\mathbf{E}V(c^n + Z\Delta c_j^n)$ can be put in the framework of Lemma 4.3 with $I' = 1$ and

$$a_i^n = (c^n)^T x_i, \quad B_{i,1}^n = (\Delta c_j^n)^T x_i.$$

If $k^n \rightarrow k$, it follows that $c^n \rightarrow c^{(k)}$ and $\Delta c_j^n \rightarrow \Delta c_j^{(k)}$ for $\Delta c_j^{(k)} = \frac{\Sigma^{(k)} e_j}{\sqrt{\lambda_j + \Sigma_{j,j}^{(k)}}}$, and $h(k^n, j) \rightarrow h(k, j)$ by Lemma 4.3. \square

Lemma 4.3 can also be applied to the functions $g(\cdot, j)$ and \bar{g} . However, in the case where $k^n \rightarrow k$ and $\Sigma_{j,j}^{(k)} = 0$, the limit $\Delta \bar{c}_j^{(k)}$ of $\Delta \bar{c}_j^n = \frac{\Sigma^n e_j}{\sqrt{\Sigma_{j,j}^n}}$ may be undefined.

Observe, however, that $\frac{\Sigma_{j,j'}^n}{\sqrt{\Sigma_{j,j}^n}} \leq \sqrt{\Sigma_{j',j'}^n}$ by the Cauchy-Schwarz inequality. Thus, in the special case where $\Sigma_{j,j}^{(k)} = 0$ for all j , $\Delta \bar{c}_j^n \rightarrow 0$ and Lemma 4.3 can be applied to show the continuity of $g(\cdot, j)$ at k .

We can also show that \bar{g} is continuous at k with $\Sigma^{(k)} = 0$. Repeating the computation from (4.4) once more, we find that

$$\bar{g}(k^n) = \mathbf{E}^n V(c^{true}) - V(c^n).$$

Observe that the distribution of c^{true} given k^n can be written as $c^n + \tilde{B}^n Z$ where $Z = (Z_1, \dots, Z_{I'})$ for some finite integer I' , $Z_1, \dots, Z_{I'}$ are i.i.d. $\mathcal{N}(0, 1)$ random variables, and \tilde{B}^n is an $M \times I'$ matrix satisfying $\Sigma^n = (\tilde{B}^n) (\tilde{B}^n)^T$. Again writing V as a finite maximum over extreme points, we obtain $\mathbf{E}^n V(c^{true}) = \mathbf{E} \max_i (c^n)^T x_i + Z^T (\tilde{B}^n)^T x_i$. We can put this in the framework of Lemma 4.3 by letting

$$a_i^n = (c^n)^T x_i, \quad B_{i,i'}^n = \sum_{j=1}^M x_{i,j} \tilde{B}_{j,i'}^n.$$

If $\Sigma^n \rightarrow 0$ componentwise, it follows that $\tilde{B}^n \rightarrow 0$ as well. Our proof of asymptotic optimality relies on the continuity of $g(\cdot, j)$ and \bar{g} only at those knowledge states k for which $\Sigma^{(k)} = 0$.

PROPOSITION 4.5. *Suppose that k^n is a sequence of knowledge states with $k^n \rightarrow k$, and $g(k^n, j) \rightarrow 0$ for all j . Then, $\bar{g}(k^n) \rightarrow 0$ also.*

Proof. By Lemma 9 of [19], we know that $0 \leq h(k^n, j) \leq g(k^n, j)$ for all n . It follows that $h(k^n, j) \rightarrow 0$ for all j . By Corollary 4.4, it follows that $h(k, j) = 0$ for all j . Then, by Proposition 4.2, it follows that $\Sigma_{j,j}^{(k)} = 0$ for all j , whence $\Sigma^{(k)} = 0$.

Recall that $\bar{g}(k) = \mathbf{E}^{(k)} V(c^{true}) - V(c^{(k)})$. Because $\Sigma^{(k)} = 0$, it follows that $c^{true} = c^{(k)}$ almost surely, whence $\bar{g}(k) = 0$. By using the continuity of \bar{g} at the particular state k , we have that $\bar{g}(k^n) \rightarrow \bar{g}(k)$, as required. \square

Second condition. Proposition 4.5 completes our proof of the first condition for asymptotic optimality. It remains to show that (4.3) holds for all $k \notin \mathcal{M}_*$.

THEOREM 4.6. *Let $k \notin \mathcal{M}_*$. Then, k has an open neighbourhood $U \subseteq \bar{\mathcal{K}}$ such that*

$$\sup_{k' \in U} P(J^{KG}(k') \in \mathcal{B}_k) < 1.$$

Proof. Let $k \in \bar{\mathcal{K}} - \mathcal{M}_*$. There are two possibilities: either $\mathcal{B}_k = \emptyset$, or $\mathcal{B}_k \neq \emptyset$. If $\mathcal{B}_k = \emptyset$, then $P(J^{KG}(k') \in \emptyset) = 0$ for any k' , so we can take $U = \bar{\mathcal{K}}$, which is open in itself, and (4.3) holds.

Suppose now that k is such that $\mathcal{B}_k \neq \emptyset$. We argue that $h(k, j) = 0$ for all $j \in \mathcal{B}_k$. To see this, take $j \in \mathcal{B}_k$. It follows that $k \in \mathcal{M}_j$. Then there exists a sequence k^n of knowledge states with $k^n \rightarrow k$ and $g(k^n, j) \rightarrow 0$. It follows that $h(k^n, j) \rightarrow 0$. By the continuity of h , it follows that $h(k, j) = 0$.

Now define

$$U = \left\{ k' \in \bar{\mathcal{K}} \mid \max_{j \in \mathcal{B}_k} h(k', j) < \min_{j \notin \mathcal{B}_k} h(k', j) \right\}.$$

This set is open by the continuity of h . We argue that $k \in U$. To see this, observe that $\max_{j \in \mathcal{B}_k} h(k, j) = 0$. Thus, the only way we can have $k \notin U$ is if $h(k, j) = 0$ for all j . But if this were true, by Proposition 4.2 we would have $\Sigma_{j,j}^{(k)} = 0$ and $g(k, j) = 0$ for all j . Then, for every $k^n \rightarrow k$, we would have $g(k^n, j) \rightarrow 0$ for all j , which would imply that $k \in \mathcal{M}_*$ by Proposition 4.5. Thus, if $k \notin \mathcal{M}_*$, then $k \in U$.

Finally, for any $k' \in U$, the definition of U implies that $\arg \max_j h(k', j) \notin \mathcal{B}_k$, whence we have $P(J^{KG}(k') \in \mathcal{B}_k) = 0$, and (4.3) holds. We conclude that (4.3) holds for any $k \notin \mathcal{M}_*$. \square

5. Numerical examples. We now present a numerical illustration of the performance of the KG method. In many respects, the problem posed in this paper, of allocating sequential measurements to adaptively learn about cost coefficients in a linear program, is itself fundamentally new. To our knowledge, the KG policy is the first approach to be proposed for systematically making measurement decisions in this problem. Thus, our objective in this section is to illustrate how systematic measurements via KG lead us to better LP solutions than several baseline policies.

We chose to test the KG policy on a minimum-cost flow problem. This problem class covers a wide variety of LP applications in logistics and supply chain management, and also has the advantage that non-trivial test problems can be easily generated. We used the NETGEN algorithm of [38] to generate a network with 50 nodes and 100 arcs. The network contains 10 supply nodes and 10 demand nodes. The total flow in the system is 500. The arc costs generated by NETGEN are integers between 1 and 10. Thus, when converted to the standard form of (1.1), our problem has 100 constraints and 200 variables.

The arc costs generated by NETGEN were used as our starting prior c^0 . The initial covariance matrix Σ^0 was created as follows. First, the prior variance of the cost on arc j was set to $\Sigma_{jj}^0 = 2$. Second, the correlation coefficient of arcs i and j was set to 0.25 if those arcs were adjacent. The correlation structure was thus based on the physical proximity of arcs in the network. In larger networks, it may be desirable to consider correlation structures with multiple degrees of correlation, gradually decreasing the correlation coefficient as the distance between two arcs increases. For

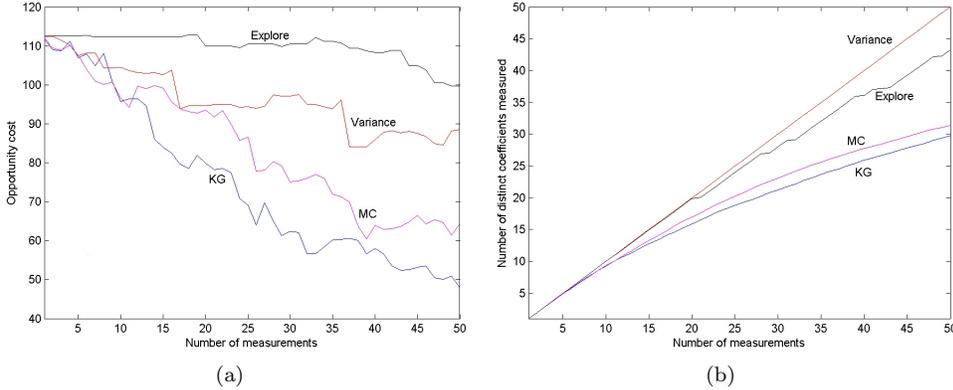


FIG. 5.1. Results for heterogeneous-prior experiments: (a) opportunity costs for both policies, and (b) number of distinct arcs measured by each policy.

the purposes of our demonstration, we consider a simple model where correlations are determined by adjacency only.

We used the CVX package of [27] to implement the algorithm for finding breakpoints given in Figure 3.3. In our implementation, we encountered some numerical issues. For example, the constraint $s^*(c^n)^T x = 0$ in (3.10) and (3.12) occasionally caused the solver to erroneously report the LP as being infeasible. In this situation, the equivalent constraint $(c^n)^T x = (c^n)^T x^*(c^n)$ can be used for much more stable performance [51]. Similarly, the analogous constraints $x_l(c^n)^T s = 0$ and $x_u(c^n)^T s = 0$ in (3.11) and (3.13) can be replaced by $(c^n + z\Delta c_j^n)^T x_l(c^n) = b^T y$ and $(c^n + z\Delta c_j^n)^T x_u(c^n) = b^T y$.

To evaluate the performance of a learning policy π , we first fix a vector c^{true} representing the true values of the objective coefficients. These true values are used to generate the measurements $\hat{c}_{J^{\pi,n}(s^n)}^{n+1}$ when we follow the policy π at each time step. However, the policy does not see the exact values of c_j^{true} . After N measurements, the *opportunity cost* incurred by following π is defined as

$$C^\pi = V(c^{true}) - (c^{true})^T x^*(c^{\pi,N})$$

where $c^{\pi,N}$ is our posterior mean at time N , based on the measurements we made according to π . This is the difference between the true objective value of the solution $x^*(c^{\pi,N})$ that is optimal according to our final beliefs under policy π , and the true optimal value of the LP.

As in [48], we consider two different prior structures. The *heterogeneous-prior* approach generates c from the prior distribution $\mathcal{N}(c^0, \Sigma^0)$. This represents a situation where we have access to accurate historical data, so that our starting estimate is reasonably good. In the *equal-prior* approach, the true objective function c^{true} is generated by NETGEN, but all of the components of c^0 are equal. We thus have less starting information; the prior gives us a rough idea of the magnitudes of the costs, but no way to distinguish between them. We conducted simulations for 100 randomly generated cost functions of each type. In both sets of experiments, the measurement noise λ_j was set to 2 uniformly for all arcs in the network. We compared the KG policy of Section 3 to three heuristics, briefly described below.

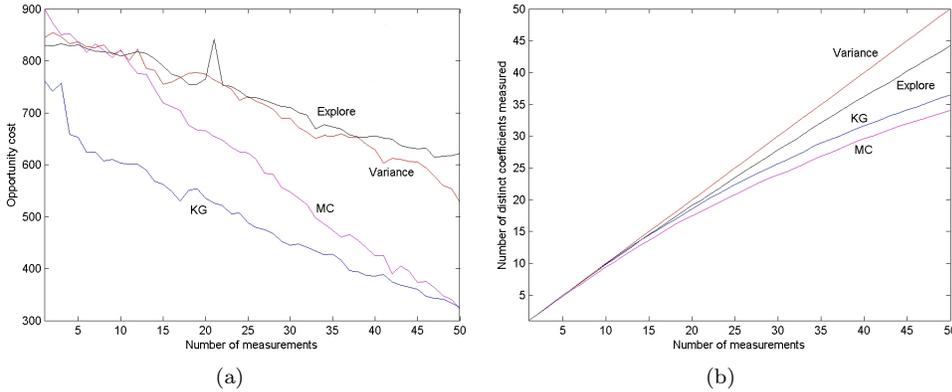


FIG. 5.2. Results for equal-prior experiments: (a) opportunity costs for both policies, and (b) number of distinct arcs measured by each policy.

Variance optimization (Variance). The Variance policy always measures the arc for which our beliefs have the highest variance. This is analogous to choosing the arc that minimizes the expected squared error of our prior estimate from the true value of the objective coefficient.

Approximate look-ahead using Monte Carlo simulation (MC). Due to the significant computational cost of evaluating the expectation in (3.3), it is reasonable to ask whether exact computation contributes value over approximate integration. The MC policy computes (3.3) approximately by generating sample realizations $Z(\omega_1), \dots, Z(\omega_K)$ from the standard normal distribution and averaging the deterministic calculations $V(c^n + \Delta c_j^n \cdot Z(\omega_k))$ over $k = 1, \dots, K$. The sample size K was chosen to make MC policy run for roughly the same length of time as the KG policy.

Pure exploration (Explore). The pure exploration policy measures a randomly chosen arc in each time step.

Figure 5.1(a) shows how the opportunity cost for both policies (averaged over 100 heterogeneous-prior problems) changes with the number of measurements. The KG policy pulls ahead after about 10 measurements; there is a clear downward trend in the opportunity cost, indicating that KG finds better solutions as the measurement budget increases. This empirical result is in line with our asymptotic analysis in Section 4. Figure 5.1(b) shows the average number of distinct arcs measured by each policy in both types of experiments. KG measures fewer arcs than other policies, meaning that there are more arcs that get measured multiple times under KG.

The MC policy exhibits generally the same behaviour as KG, but with noticeable degradation in performance due to inaccuracies from Monte Carlo sampling. Increasing the sample size can improve performance, but we found that this also greatly increased computational cost. Although the LP sensitivity analysis procedure used by KG can potentially visit a large number of extreme points, in practice many of these solutions are dominated, and will not achieve the maximum in (3.4) for any value of z . Furthermore, every new solution found by the procedure is adjacent to the previous one (as in Figure 3.2), and this structure can be exploited to speed up computation further. Overall, we found that a small sample size $K = 10$ was enough to make the MC policy run for as long as KG in this problem.

Figure 5.2 shows analogous results for the equal-prior case. Because the prior conveys much less information about the true objective function, our initial solution

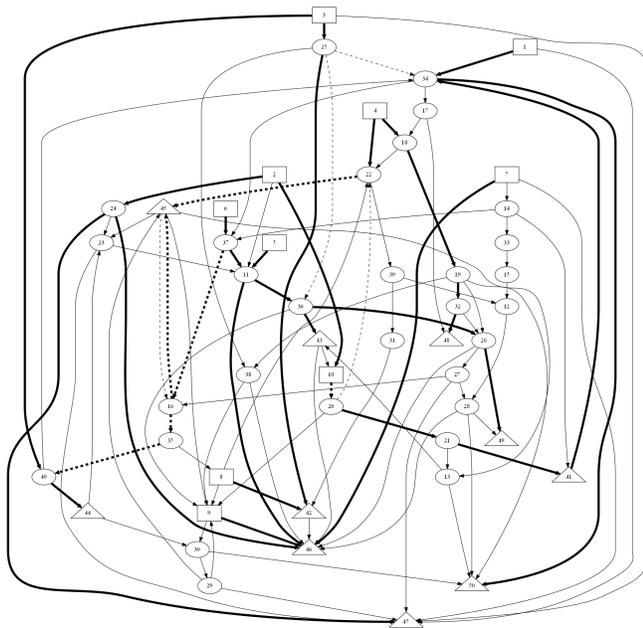


FIG. 5.3. Diagram showing decision-making at time 0 under the heterogeneous-prior setup. Squares represent supply nodes; triangles represent demand nodes. Lines in bold indicate edges with non-zero flow according to $x^*(c^0)$. Dashed lines indicate edges with high KG factors.

$x^*(c^0)$ is much worse than in the heterogeneous-prior case. This explains the offset between KG and the other policies in the first data point (for $n = 1$): the prior is so uninformative that the very first measurement contributes a big improvement. The policies have the same general behaviour as in the heterogeneous-prior case. The MC policy lags noticeably more in the early stages, but catches up to KG more quickly in the later stages, reflecting the intuitive idea that it is more important to calculate the value of information exactly in the early stages, when information is most valuable.

Lastly, Figure 5.3 gives some intuition of how the KG policy selects arcs to measure. The figure shows a schematic of the network generated for our experiments, drawn using the Graphviz package [18]. Lines drawn in bold indicate edges with non-zero flow according to the solution $x^*(c^0)$, which is believed to be optimal under the prior. Dashed lines indicate edges with the 10 largest KG factors, calculated using the initial beliefs c^0 and Σ^0 . Although only one of these edges is actually chosen for measurement, the figure sheds light on how KG values edges in general. We see that edges with high KG factors are either part of the current shipment schedule (assigned non-zero flow under the prior), or they are adjacent to such edges. In the latter case, a small change in our beliefs about an edge is more likely to result in a change to the overall shipment schedule. For example, the prior ships flow through nodes 35 and 16 to satisfy demand at 45, but there is a lot of uncertainty about whether this is really a good route, or whether 45 should be used as a transshipment node. Both edges between 16 and 45 are especially valuable.

These numerical examples indicate that there is a clear benefit by incorporating the value of information, as defined by the KG logic, into our decision-making. The empirical evidence shown here reflects the particular problem setting chosen for our

illustration. However, we feel that these results demonstrate that the KG algorithm proposed in this paper runs as expected, and produces successively better solutions to the underlying LP.

6. Conclusion. We have posed a new type of optimal learning problem: solving an LP with unknown cost coefficients, given finitely many chances to learn about them through noisy measurements. Our uncertainty about the objective function is represented by a Bayesian belief structure. We have developed a knowledge gradient strategy for making measurement decisions efficiently. In the setting of a multivariate Gaussian prior and independent Gaussian measurement errors, the KG approach provides us with a closed-form expression for the value of information contributed by a single measurement to our understanding of the underlying LP. This quantity can be computed exactly by applying techniques from LP sensitivity analysis. We have shown that this policy is asymptotically optimal, meaning that it learns the true optimal value of the LP as the number of measurements becomes large. Our numerical examples show how, on average, KG gradually leads us to improve our LP solution.

We view this work as a bridge between the fields of stochastic optimization and optimal learning. Many practical optimization problems are characterized by a high degree of uncertainty about key model parameters. Whenever approximations are used to make decisions, we run the risk of making a poor decision by relying on an inaccurate approximation. It becomes crucial to develop intelligent, systematic strategies for balancing existing approximations with the value of new information that can improve the approximation. This dilemma is studied extensively in the literature on optimal learning, but typically for a very simple underlying optimization structure. We believe that the idea explored in our work, namely the idea of quantifying and formalizing the value of information in a more general and complex optimization framework, has a great deal of potential for resolving information collection issues in many broad classes of stochastic optimization problems.

Appendix. Proofs.

A.1. Proof of Theorem 2.1. By the tower property of conditional expectation,

$$\sup_{\pi} \sup_{\chi} \mathbf{E}^{\pi} (c^{true})^T \chi(k^N) = \sup_{\pi} \sup_{\chi} \mathbf{E}^{\pi} \mathbf{E}^{\pi, N} (c^{true})^T \chi(k^N)$$

because \mathbf{E}^{π} only conditions on the choice of learning policy, whereas $\mathbf{E}^{\pi, N}$ conditions on that as well as \mathcal{F}^n . Observe that, because k^N is \mathcal{F}^N -measurable, we have

$$\mathbf{E}^{\pi, N} (c^{true})^T \chi(k^N) = \chi(k^N)^T \mathbf{E}^{\pi, N} (c^{true}) = (c^N)^T \chi(k^N)$$

by the linearity of expected values. Thus, we can write

$$\begin{aligned} \sup_{\pi} \sup_{\chi} \mathbf{E}^{\pi} (c^{true})^T \chi(k^N) &= \sup_{\pi} \sup_{\chi} \mathbf{E}^{\pi} (c^N)^T \chi(k^N) \\ &\leq \sup_{\pi} \sup_{\chi} \mathbf{E}^{\pi} \max_{x \in \mathcal{A}} (c^N)^T x \\ &= \sup_{\pi} \mathbf{E}^{\pi} \max_{x \in \mathcal{A}} (c^N)^T x \\ &= \sup_{\pi} \mathbf{E}^{\pi} V(c^N). \end{aligned} \tag{A.1}$$

The second line is due to the monotonicity of conditional expectations because, for any outcome ω of the measurements,

$$(c^N(\omega))^T \chi(k^N(\omega)) \leq \max_{x \in \mathcal{A}} (c^N(\omega))^T x.$$

Thus, we can remove the supremum over χ because the expression inside that supremum no longer depends on the implementation policy.

Now, fix π and let χ' be defined by $\chi'(k^N) = x^*(c^N)$. Then,

$$\mathbf{E}^\pi V(c^N) = \mathbf{E}^\pi (c^N)^T \chi'(k^N) \leq \sup_{\chi} \mathbf{E}^\pi (c^N)^T \chi(k^N).$$

It follows that

$$\sup_{\pi} \mathbf{E}^\pi V(c^N) \leq \sup_{\pi} \sup_{\chi} \mathbf{E}^\pi (c^N)^T \chi(k^N). \quad (\text{A.2})$$

Combining (A.1) with (A.2) yields the desired result.

A.2. Proof of Lemma 4.3. We define random variables

$$Y^n = \max_i a_i^n + \sum_{i'=1}^{I'} B_{i,i'}^n Z_{i'}, \quad Y = \max_i a_i + \sum_{i'=1}^{I'} B_{i,i'} Z_{i'},$$

and show that $Y^n \rightarrow Y$ in L^1 . To see this, let $\varepsilon > 0$ and take N such that, for all $n > N$, we have $|a_i^n - a_i|, |B_{i,i'}^n - B_{i,i'}| < \varepsilon$ for all i, i' . Then, for $n > N$,

$$\begin{aligned} \mathbf{E} |Y^n - Y| &= \mathbf{E} \left| \max_i \left(a_i^n + \sum_{i'=1}^{I'} B_{i,i'}^n Z_{i'} \right) - \max_i \left(a_i + \sum_{i'=1}^{I'} B_{i,i'} Z_{i'} \right) \right| \\ &\leq \mathbf{E} \max_i \left| \left(a_i^n + \sum_{i'=1}^{I'} B_{i,i'}^n Z_{i'} \right) - \left(a_i + \sum_{i'=1}^{I'} B_{i,i'} Z_{i'} \right) \right| \\ &\leq \mathbf{E} \max_i |a_i^n - a_i| + \left| \sum_{i'=1}^{I'} (B_{i,i'}^n - B_{i,i'}) Z_{i'} \right| \\ &\leq \varepsilon + \mathbf{E} \max_i \sum_{i'=1}^{I'} |B_{i,i'}^n - B_{i,i'}| \cdot |Z_{i'}| \\ &\leq \varepsilon + \max_{i,i''} |B_{i,i''}^n - B_{i,i''}| \sum_{i'=1}^{I'} \mathbf{E} |Z_{i'}| \\ &\leq \varepsilon \left(1 + \sum_{i'=1}^{I'} \mathbf{E} |Z_{i'}| \right). \end{aligned}$$

Since the random variables $Z_{i'}$ are i.i.d. and integrable, we have $\mathbf{E} |Y^n - Y| \rightarrow 0$ as $n \rightarrow \infty$.

References.

- [1] I. ADLER AND R.D.C. MONTEIRO, *A Geometric View of Parametric Linear Programming*, *Algorithmica*, 8 (1992), pp. 161–176.
- [2] P. AUER, N. CESA-BIANCHI, AND P. FISCHER, *Finite-time analysis of the multiarmed bandit problem*, *Machine Learning*, 47 (2002), pp. 235–256.
- [3] R. E. BECHHOFFER, T. J. SANTNER, AND D. M. GOLDSMAN, *Design and Analysis of Experiments for Statistical Selection, Screening and Multiple Comparisons*, J.Wiley & Sons, New York, 1995.
- [4] R. BELLMAN, *Dynamic Programming*, Princeton University Press, Princeton, 1957.
- [5] D. A. BERRY AND B. FRISTEDT, *Bandit Problems*, Chapman and Hall, London, 1985.
- [6] D. BERTSIMAS AND J.N. TSITSIKLIS, *Introduction to Linear Optimization*, Athena Scientific, 1997.
- [7] P.J. BICKEL AND K.A. DOKSUM, *Mathematical statistics: basic ideas and selected topics*, Prentice Hall, 2nd ed. updated printing ed., 2007.
- [8] J.R. BIRGE, *The value of the stochastic solution in stochastic linear programs with fixed recourse*, *Mathematical Programming*, 24 (1982), pp. 314–325.
- [9] J.R. BIRGE AND F. LOUVEAUX, *Introduction to Stochastic Programming*, Springer-Verlag, New York, 1997.
- [10] S.E. CHICK AND K. INOUE, *New procedures to select the best simulated system using common random numbers*, *Management Science*, 47 (2001), pp. 1133–1149.
- [11] ———, *New two-stage and sequential procedures for selecting the best simulated system*, *Operations Research*, 49 (2001), pp. 732–743.
- [12] S. E. CHICK, J. BRANKE, AND C. SCHMIDT, *Sequential Sampling to Myopically Maximize the Expected Value of Information*, *INFORMS Journal on Computing*, 22 (2010), pp. 71–80.
- [13] S. E. CHICK AND N. GANS, *Economic analysis of simulation selection problems*, *Management Science*, 55 (2009), pp. 421–437.
- [14] E. DAHAN AND J. R. HAUSER, *Product development: managing a dispersed process*, in *Handbook of Marketing*, B. A. Weitz and R. Wensley, eds., Sage Publication, Thousand Oaks, CA, 2002, pp. 179–222.
- [15] G.B. DANTZIG, *Linear programming under uncertainty*, *Management Science*, 1 (1955), pp. 197–206.
- [16] M. H. DEGROOT, *Optimal Statistical Decisions*, John Wiley and Sons, 1970.
- [17] T. EL-NAZER AND B.A. MCCARL, *The choice of crop rotation: a modeling approach and case study*, *American Journal of Agricultural Economics*, 68 (1986), pp. 127–136.
- [18] J. ELLSON, E. GANSNER, L. KOUTSOFIOS, S. NORTH, AND G. WOODHULL, *Graphviz—open source graph drawing tools*, in *Proceedings of the 9th International Symposium on Graph Drawing*, P. Mutzel, M. Junger, and S. Leipert, eds., 2002, pp. 594–597.
- [19] P. I. FRAZIER AND W. B. POWELL, *Consistency of Sequential Bayesian Sampling Policies*, *SIAM Journal on Control and Optimization*, 49 (2011), pp. 712–731.
- [20] P. I. FRAZIER, W. B. POWELL, AND S. DAYANIK, *A knowledge gradient policy for sequential information collection*, *SIAM Journal on Control and Optimization*, 47 (2008), pp. 2410–2439.
- [21] ———, *The knowledge-gradient policy for correlated normal rewards*, *INFORMS*

- Journal on Computing, 21 (2009), pp. 599–613.
- [22] S. GASS AND T. SAATY, *The computational algorithm for the parametric objective function*, Naval Research Logistics Quarterly, 2 (1955), pp. 39–45.
- [23] A.B. GELMAN, J.B. CARLIN, H.S. STERN, AND D.B. RUBIN, *Bayesian data analysis*, CRC Press, second ed., 2004.
- [24] J. GEUNES AND P.M. PARDALOS, *Network optimization in supply chain management and financial engineering: An annotated bibliography*, Networks, 42 (2003), pp. 66–84.
- [25] A. GHAFFARI-HADIGHEH AND T. TERLAKY, *Sensitivity analysis in linear optimization: Invariant support set intervals*, European Journal of Operational Research, 169 (2006), pp. 1158–1175.
- [26] J. C. GITTINS, K. D. GLAZEBROOK, AND R. WEBER, *Multi-armed bandit allocation indices (2nd ed.)*, John Wiley and Sons, 2011.
- [27] M. GRANT AND S. BOYD, *CVX: Matlab software for disciplined convex programming, version 1.21*. <http://cvxr.com/cvx>, October 2010.
- [28] S.S. GUPTA AND K.J. MIESCKE, *Bayesian look ahead one stage sampling allocations for selecting the largest normal mean*, Statistical Papers, 35 (1994), pp. 169–177.
- [29] ———, *Bayesian look ahead one-stage sampling allocations for selection of the best population*, Journal of Statistical Planning and Inference, 54 (1996), pp. 229–244.
- [30] W.K. HANEVELD AND A.W. STEGEMAN, *Crop succession requirements in agricultural production planning*, European Journal of Operational Research, 166 (2005), pp. 406–429.
- [31] H. ITAMI, *Expected Objective Value of a Stochastic Linear Program and the Degree of Uncertainty of Parameters*, Management Science, 21 (1974), pp. 291–301.
- [32] B. JANSEN, J.J. DE JONG, C. ROOS, AND T. TERLAKY, *Sensitivity analysis in linear programming: just be careful!*, European Journal of Operational Research, 101 (1997), pp. 15–28.
- [33] B. JANSEN, C. ROOS, T. TERLAKY, AND J. VIAL, *Interior-point methodology for linear programming: duality, sensitivity analysis and computational aspects*, Tech. Report 93-28, Faculty of Technical Mathematics and Informatics, Delft University of Technology, 1993.
- [34] P. KALL AND S.W. WALLACE, *Stochastic Programming*, John Wiley & Sons, New York, 1994.
- [35] Y. KAO, B. VAN ROY, AND X. YAN, *Directed regression*, in Advances in Neural Information Processing Systems, Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, eds., vol. 22, 2009, pp. 889–897.
- [36] S.-H. KIM AND B. L. NELSON, *Selecting the best system*, in Handbooks of Operations Research and Management Science, vol. 13: Simulation, S. G. Henderson and B. L. Nelson, eds., North-Holland Publishing, Amsterdam, 2006, pp. 501–534.
- [37] ———, *Recent advances in ranking and selection*, in Proceedings of the 2007 Winter Simulation Conference, S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, eds., 2007, pp. 162–172.
- [38] D. KLINGMAN, A. NAPIER, AND J. STUTZ, *NETGEN: A program for generating large scale capacitated assignment, transportation, and minimum cost flow network problems*, Management Science, 20 (1974), pp. 814–821.
- [39] V.V. KOLBIN, *Stochastic Programming*, D. Reidel Publishing Company, Dordrecht, Holland, 1977. Translated from Russian by I.P. Grigoryev.

- [40] A. MADANSKY, *Inequalities for stochastic linear programming problems*, Management Science, 6 (1960), pp. 197–204.
- [41] K.G. MURTY, *Computational complexity of parametric linear programming*, Mathematical Programming, 19 (1980), pp. 213–219.
- [42] D. M. NEGOESCU, P. I. FRAZIER, AND W. B. POWELL, *The Knowledge-Gradient Algorithm for Sequencing Experiments in Drug Discovery*, INFORMS Journal on Computing, 23 (2010), pp. 346–363.
- [43] M.W. PADBERG, *Linear optimization and extensions*, Springer, 1999.
- [44] W. B. POWELL AND I. O. RYZHOV, *Optimal Learning*, John Wiley and Sons, 2012.
- [45] C. ROOS, T. TERLAKY, AND J. P. VIAL, *Theory and Algorithms for Linear Optimization: An Interior Point Approach*, John Wiley and Sons, Chichester, UK, 1997.
- [46] A. RUSZCZYŃSKI AND A. SHAPIRO, *Handbooks in Operations Research and Management Science: Stochastic Programming*, vol. 10, Elsevier, Amsterdam, 2003.
- [47] I. O. RYZHOV AND W. B. POWELL, *The knowledge gradient algorithm for on-line subset selection*, in Proceedings of the 2009 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning, 2009, pp. 137–144.
- [48] ———, *Information collection on a graph*, Operations Research, 59 (2011), pp. 188–201.
- [49] I. O. RYZHOV, W. B. POWELL, AND P. I. FRAZIER, *The knowledge gradient algorithm for a general class of online learning problems*, Operations Research, 60 (2012), pp. 180–195.
- [50] I. O. RYZHOV, M. R. VALDEZ-VIVAS, AND W. B. POWELL, *Optimal Learning of Transition Probabilities in the Two-Agent Newsvendor Problem*, in Proceedings of the 2010 Winter Simulation Conference, B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, and E. Yücesan, eds., 2010, pp. 1088–1098.
- [51] T. TERLAKY, *Private communication*. 2010.
- [52] Y. WANG, *Essays on supply risk in global operations*, PhD thesis, University of North Carolina at Chapel Hill, 2007.
- [53] Y. WANG, W. GILLAND, AND B. TOMLIN, *Mitigating supply risk: Dual sourcing or process improvement?*, Manufacturing & Service Operations Management, 12 (2010), pp. 489–510.
- [54] R.E. WENDELL, *The Tolerance Approach to Sensitivity Analysis in Linear Programming*, Management Science, 31 (1985), pp. 564–578.