Operations Research

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Bayesian Optimization via Simulation with Pairwise Sampling and Correlated Prior Beliefs

Jing Xie, Peter I. Frazier, Stephen E. Chick

To cite this article:
Jing Xie, Peter I. Frazier, Stephen E. Chick (2016) Bayesian Optimization via Simulation with Pairwise Sampling and Correlated Prior Beliefs. Operations Research 64(2):542-559. https://doi.org/10.1287/opre.2016.1480

Operations Research
Vol. 64, No. 2, March–April 2016, pp. 542–559
ISSN 0030-364X (print) | ISSN 1526-5463 (online)

http://dx.doi.org/10.1287/opre.2016.1480

Bayesian Optimization via Simulation with Pairwise Sampling and Correlated Prior Beliefs

Jing Xie
Credibly, New York, New York 10010, joyce.jingx@gmail.com

Peter I. Frazier
School of Operations Research and Information Engineering, Cornell University, Ithaca, New York 14853, pf98@cornell.edu

Stephen E. Chick
Technology and Operations Management Area, INSEAD, 77300 Fontainebleau, France,
stephen.chick@insead.edu

This paper addresses discrete optimization via simulation. We show that allowing for both a correlated prior distribution on the means (e.g., with discrete Kriging models) and sampling correlation (e.g., with common random numbers, or CRN) can significantly improve the ability to quickly identify the best alternative. These two correlations are brought together for the first time in a highly sequential knowledge-gradient sampling algorithm, which chooses points to sample using a Bayesian value of information (VOI) criterion. We provide almost sure convergence guarantees as the number of samples grows without bound when parameters are known and provide approximations that allow practical implementation including a novel use of the VOI's gradient rather than the response surface's gradient. We demonstrate that CRN leads to improved optimization performance for VOI-based algorithms in sequential sampling environments with a combinatorial number of alternatives and costly samples.

Keywords: discrete optimization via simulation; value of information; Kriging; correlated samples.
Subject classifications: simulation: design of experiments; statistics: Bayesian.
Area of review: Stochastic Models.
History: Received August 2013; revisions received December 2014, September 2015; accepted December 2015. Published online in *Articles in Advance* March 25, 2016.

1. Introduction

We seek to solve the following stochastic optimization problem via simulation,

$$\max_{x\in\Upsilon} \mathbb{E}[F(x, D)], \tag{1}$$

where $\Upsilon \equiv \{1, 2, \ldots, k\}$ is a finite set of k alternatives, the random quantity D represents the uncertainty in the problem, and the function F is taken to be a simulation that outputs a real-valued reward for a given x and D. We are interested in contexts where k is large enough to render it difficult to estimate each $\theta(x) \equiv \mathbb{E}[F(x, D)]$ precisely. We propose an algorithm that aims to reach the optimal solution in such contexts by doing a limited number of simulation runs for a subset of alternatives at every iteration.

Because of its importance, previous authors have proposed algorithms of several types to address this problem, including randomized search (Andradóttir 1998, 2006; Zhou et al. 2008), metaheuristics (Shi and Ólafsson 2000), metamodel-based algorithms (Barton 2009, van Beers and Kleijnen 2008), Bayesian value-of-information algorithms (Chick 2006, Frazier 2010), local search algorithms (Wang et al. 2013, Hong and Nelson 2006, Xu et al. 2010), model-based search (Hu et al. 2012, Wang et al. 2010), and ranking and selection algorithms (Kim and Nelson 2006, Chen and Lee 2010, Branke et al. 2007). Andradóttir (1998) and Fu (2002) provide surveys of the field.

We study this problem in a Bayesian context, where we place a prior probability distribution on the unknown means of the outputs of the alternatives, and use value of information (VOI) calculations to decide which alternative, or collection of alternatives, would be most useful to sample next. The advantage of doing so is that making decisions based on the VOI per unit of computing effort (a so-called knowledge gradient, or KG factor), automatically addresses the exploration versus exploitation trade-off, and tends to reduce the number of simulations required on average to reach a given solution quality, potentially (but not necessarily) at the cost of requiring more computation to decide where to sample.

The prior probability distribution that we consider is a multivariate normal distribution and allows for correlation in our prior belief between two alternatives. This models a belief that two alternatives with similar characteristics often have similar expected performance, and allows the algorithm that we construct to do well even in problems where the number of alternatives is much larger than the number of samples that we can take.

We allow common random numbers (CRN), in which multiple alternatives are simulated using the same stream of random numbers. This induces correlation in the noise, which can be advantageous for optimization when the correlation is positive, because it allows more accurate estimation of the differences between alternatives' values.

Several previous authors have considered Bayesian formulations of optimization via simulation. The setting most frequently studied is that of ranking and selection, with relatively few alternatives, an independent prior distribution, and independent sampling (Gupta and Miescke 1996, Chick and Inoue 2001b, Frazier et al. 2008, Chick and Frazier 2012). Bayesian optimization via simulation with correlated prior distributions (but not with CRN) for problems with many alternatives was considered in a discrete setting (Frazier et al. 2009) and in a continuous setting (Villemonteix et al. 2009, Huang et al. 2006, Scott et al. 2011). This work in a continuous setting parallels work on noise-free Bayesian global optimization (Jones et al. 1998, Forrester et al. 2008, Brochu et al. 2009).

Our analysis differs from this previous literature by allowing the use of CRN. This has been perceived to be difficult, because sampling with CRN makes it difficult to compute the VOI, and to maintain a closed-form posterior distribution. We overcome these difficulties by calculating the VOI for observing the *difference* in value between two alternatives, which can be done analytically, and by calculating the posterior with adaptively updated point estimates of the noise covariance. We show that, in the context of VOI-based algorithms, using CRN can greatly improve performance.

Sampling with correlated means and CRN in the Bayesian setting using VOI methods has been considered by Chick and Inoue (2001a), who focused on two-stage sampling algorithms, and restricted attention to conjugate prior distributions for the unknown means. Others have considered sampling with CRN in the optimal computing budget allocation framework (Fu et al. 2004), in the indifference-zone setting (Clark and Yang 1986, Nelson and Matejcik 1995), and in the multiple comparisons problem (Yang and Nelson 1991, Nakayama 2000, Kim 2005). The current work differs from this previous work in its focus on problems with many alternatives, enabled by a multivariate normal prior distribution with arbitrary covariance.

The current work, in its use of multivariate normal prior distributions, makes a link to Gaussian process (GP) priors (Rasmussen and Williams 2006) and stochastic Kriging (Ankenman et al. 2010; Chen et al. 2012, 2013). When alternatives correspond to points on a grid, as they do in many resource allocation problems (e.g., each alternative specifies the number of each of several employee types to have present), our use of a multivariate normal prior distribution can be implemented by placing a GP prior over the continuum, and then only considering points on the grid.

We present three techniques that reduce the computation required to find an alternative, or pair of alternatives, with a large VOI for sampling. The first is to use the gradient of the VOI in performing this search, calculating it over an embedding of our discrete alternatives into a continuous space. This use of the gradient of the VOI differs from the more common use of gradients of the response surface in optimization. The second is to approximate the VOI with a useful lower bound. The third is to use data structures that avoid enumerating alternatives, instead tracking only those alternatives that have been sampled, and reconstructing required portions of the posterior distribution as needed. This is standard in GP regression, but contrasts with previous work on optimization via simulation with CRN (Clark and Yang 1986, Nelson and Matejcik 1995, Chick and Inoue 2001a, Fu et al. 2004). These three techniques were applied in Scott et al. (2011) to a continuous setting without CRN.

We also provide an almost sure guarantee of convergence to the global optimum, as the number of samples taken grows without bound, when parameters are known. In addition to allowing correlated sampling, this theoretical result contrasts with Scott et al. (2011) in having conditions that are easier to verify. It also contrasts with other work that focuses on convergence to local optima (Hong and Nelson 2006, Xu et al. 2010, Wang et al. 2013).

The current paper extends a report of our preliminary work (Frazier et al. 2011) in a number of ways. It provides an enhanced version of the algorithm that scales to much larger problems, a theoretical analysis showing convergence to a global optimum, a derivation of a maximum likelihood estimation method for estimating covariance parameters from samples observed with CRN, and additional numerical comparisons with other algorithms on larger problems.

We begin in §2 by formally defining the problem, the relevant statistical model, and the VOI for our context. Section 3 shows how the VOI and the corresponding KG factor can be computed in the context of optimization via simulation with correlated sampling. Section 4 describes a generic sampling algorithm that forms the basis for specific sampling algorithms. Section 5 uses the VOI and KG factor computations to create new KG sampling algorithms. Section 6 states theoretical results that show that these KG algorithms can produce consistent estimates of the global optimum in the limit as the sampling budget grows large, when parameters are known. Section 7 discusses practical implementation issues when parameters are unknown. Numerical results in §8 show a distinct advantage to the ability to sequentially sample with CRN in discrete optimization via simulation problems. Appendices in the paper and the electronic companion (available as supplemental material at http://dx.doi.org/10.1287/opre.2016.1480) prove theoretical results, derive gradient and statistical estimation results used in the algorithm, and discuss additional implementation issues.

2. Sampling Model and Mechanism for Posterior Inference

We estimate the expectation in (1) with simulation, and let $\theta = [\theta(1), \ldots, \theta(k)]^T$ be the vector of means, where ^{T} denotes matrix transposition. The use of CRN implies that the covariance matrix of the vector $\mathbb{E}[F(x, D)]_{x\in\Upsilon}$, which we call Λ, may have nonzero off-diagonal elements when all alternatives are simulated.

For the theory developed here, we assume that the vector $\mathbb{E}[F(x, D)]_{x\in\Upsilon}$ has a multivariate normal distribution (with

mean θ and covariance matrix Λ). This normality assumption is widely used in statistical selection and simulation optimization (Kim and Nelson 2006, Chick 2006) and in Gaussian process regression (Rasmussen and Williams 2006). The assumption may also be useful when output is not normally distributed but might be made approximately so through a technique known as batching (averaging several observations per sample to obtain an approximately normal sample, Law 2014). Large batches increase the computation per observation, making our approach most useful when means of small batches are approximately normal.

We use a Bayesian formulation, in which we begin with a multivariate normal prior distribution on θ to describe uncertainty about its unknown value,

$$\theta \sim \text{Normal}(\mu_0, \Sigma_0). \tag{2}$$

The choice of Σ_0 allows for conjugate prior distributions for θ (Chick and Inoue 2001a) or for GP priors (Rasmussen and Williams 2006), which are related to Kriging models (Cressie 1993). A parametric family can be used to specify μ_0 and Σ_0 in terms of a function taking the alternatives and a few additional parameters as arguments. In practice, the parameters specifying μ_0 and Σ_0, as well as the sampling covariance Λ, are unknown, but we will initially assume they are fully known for simplicity. Then, we will relax this assumption in §7.

In this section, we describe how sampling occurs, the nature and computation of the posterior distribution, and how that posterior distribution is used to measure the value of sampling information that drives our algorithm below. This will require some vector and matrix notation.

The ith entry of a length-k vector v (e.g., θ and μ_0) is written $v(i)$, and the (i,j)th entry of a k-by-k matrix M (e.g., Σ_0 and Λ) is written $M(i,j)$. Moreover, for an ordered collection of m alternatives $\vec{x} = (x^{(1)}, x^{(2)}, \ldots, x^{(m)})$ with elements $x^{(i)} \in \{1,2,\ldots,k\}$ for each i, we use $v(\vec{x})$ to denote the length-m subvector of v with the ith entry equal to $v(x^{(i)})$. Let $\vec{x}'$ with elements in $\{1,2,\ldots,k\}$ be another vector of alternatives with m' entries. We denote by $M(\vec{x},\vec{x}')$ the m-by-m' submatrix of M with the (i,j)th entry equal to $M(x^{(i)}, x'^{(j)})$.

2.1. Sampling Model and Distribution of Outputs

At each iteration $n = 1, 2, \ldots$ we choose a set of the alternatives to sample, specified as a row vector $\vec{x}_n$ with elements in $\{1,2,\ldots,k\}$, and sample each of the chosen alternatives once using CRN. Each alternative may appear at most once in $\vec{x}_n$. We then observe a column vector $\vec{y}_n$, with one entry for each alternative sampled. The conditional distribution of $\vec{y}_n$ given $\vec{x}_n$, θ is assumed to be Gaussian and independent of previous observations,

$$\vec{y}_n \,|\, \theta, \vec{x}_n, (\vec{x}_m, \vec{y}_m \colon m < n) \sim \text{Normal}(\theta(\vec{x}_n), \Lambda(\vec{x}_n, \vec{x}_n)). \tag{3}$$

Although (3) is general, in our algorithm below, the sampling decision $\vec{x}_n$ is either a singleton x_n, with corresponding observation y_n, or a pair of alternatives $(x_n^{(1)}, x_n^{(2)})$, with corresponding observations $(y_n^{(1)}, y_n^{(2)})$. The notation $\vec{x}_n$ and $\vec{y}_n$ indicates the general case, in which one or more alternatives is sampled, whereas x_n and y_n always indicates a single alternative. The sampling distribution of (3) for these two cases (singletons and pairs) are

$$y_n \,|\, \theta, x_n \sim \text{Normal}(\theta(x_n), \Lambda(x_n, x_n)), \quad \text{and}$$

$$(y_n^{(1)}, y_n^{(2)}) \,|\, \theta, (x_n^{(1)}, x_n^{(2)}) \sim \text{Normal}\left(\begin{bmatrix} \theta(x_n^{(1)}) \\ \theta(x_n^{(2)}) \end{bmatrix}, \begin{bmatrix} \Lambda(x_n^{(1)}, x_n^{(1)}) & \Lambda(x_n^{(1)}, x_n^{(2)}) \\ \Lambda(x_n^{(2)}, x_n^{(1)}) & \Lambda(x_n^{(2)}, x_n^{(2)}) \end{bmatrix}\right).$$

These sampling distributions are sufficient for calculating posterior distributions from observations in the sampling algorithms that we propose, but when computing the VOI in §2.3, we will also consider three additional sampling distributions. First, we will consider the sampling distribution of observing only the *difference* between a pair $(x_n^{(1)}, x_n^{(2)})$ of alternatives,

$$y_n^{(1)} - y_n^{(2)} \,|\, \theta, (x_n^{(1)}, x_n^{(2)}) \sim \text{Normal}\big(\theta(x_n^{(1)}) - \theta(x_n^{(2)}), \Lambda(x_n^{(1)}, x_n^{(1)}) + \Lambda(x_n^{(2)}, x_n^{(2)}) - 2\Lambda(x_n^{(1)}, x_n^{(2)})\big).$$

Second, we will consider the sampling distribution of observing not necessarily one but $\beta_n \geqslant 1$ vectors of samples from the distribution given by (3), each generated using an independent CRN stream. We do this to compute an average VOI per sample because related sampling procedures can work better by allowing β_n to exceed 1 or to adaptively depend on $\vec{x}$ (Chick and Frazier 2012). We generalize $\vec{y}_n$ to refer to the average of these β_n observations, so

$$\vec{y}_n \,|\, \theta, \vec{x}_n, \beta_n \sim \text{Normal}\big(\theta(\vec{x}_n), \Lambda(\vec{x}_n, \vec{x}_n)/\beta_n\big). \tag{4}$$

Third, we will consider the sampling distribution of observing $\beta_n \geqslant 1$ independent differences between a pair $(x_n^{(1)}, x_n^{(2)})$, continuing to let $\vec{y}_n = (y_n^{(1)}, y_n^{(2)})$ denote the average of these observations,

$$y_n^{(1)} - y_n^{(2)} \,|\, \theta, (x_n^{(1)}, x_n^{(2)}), \beta_n \sim \text{Normal}\big(\theta(x_n^{(1)}) - \theta(x_n^{(2)}), \big[\Lambda(x_n^{(1)}, x_n^{(1)}) + \Lambda(x_n^{(2)}, x_n^{(2)}) - 2\Lambda(x_n^{(1)}, x_n^{(2)})\big]/\beta_n\big). \tag{5}$$

These last three sampling distributions are used only to compute the VOI. In the sampling algorithms that we propose below, we observe from both alternatives when sampling from a pair. We also take only one sample at a time from a singleton or pair even when we calculate a VOI with $\beta_n > 1$ in the spirit of fully sequential sampling and to simplify notation below (taking β_n samples at a time may reduce the computational time per sampling decision but would involve adaptations of certain covariance terms below).

2.2. Posterior Distribution for Unknown Means and Its Computation

With the sampling scheme in (3), and the assumption that the sampling covariance matrix Λ is known, we can com-

pute a closed-form expression for the posterior distribution on θ. We let $\mathbb{E}_n$ and Var_n indicate the conditional expectation and variance, respectively, with respect to the data $\vec{x}_1, \vec{y}_1, \vec{x}_2, \vec{y}_2, \ldots, \vec{x}_n, \vec{y}_n$, where each $\vec{y}_n$ is sampled according to (3). Define $\mu_n = \mathbb{E}_n \theta$ and $\Sigma_n = \mathrm{Var}_n \theta$. The posterior distribution on θ is normal (see, e.g., Gelman et al. 2004, section 14.6),

$$\theta \mid \vec{x}_1, \vec{y}_1, \vec{x}_2, \vec{y}_2, \ldots, \vec{x}_n, \vec{y}_n \sim \mathrm{Normal}(\mu_n, \Sigma_n),$$

where the posterior mean μ_n and variance Σ_n can be computed analytically, either directly from the prior and the full data, or recursively, updating as each new data point $\vec{x}_n, \vec{y}_n$ is added.

When the number of alternatives k is large, it is computationally infeasible to store μ_n and Σ_n, because Σ_n is a k-by-k matrix. Therefore, we use a method commonly used in GP regression, which calculates the posterior distribution on the sampled alternatives and any desired additional alternatives, without requiring a k-by-k matrix. We briefly describe this method here, giving some notation to be used later, and focusing on singletons and pairs.

Let $\mathscr{X}_n$ denote the cumulative row vector of alternatives sampled from iterations 1 to n, i.e., the concatenation of $\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n$ into a row. Alternatives appear more than once if they are sampled more than once. For example, if $\vec{x}_1 = x_1$ and $\vec{x}_2 = (x_2^{(1)}, x_2^{(2)})$, then $\mathscr{X}_1 = (x_1)$ and $\mathscr{X}_2 = (x_1, x_2^{(1)}, x_2^{(2)})$. In addition, if $x_1 = x_2^{(1)} = x$ then $\mathscr{X}_2 = (x, x, x_2^{(2)})$.

In §2.3 we will compute the VOI for an arbitrary (singleton or pair) sampling decision $\vec{x}$ at iteration $n+1$. Let the vector $\mathscr{X}_{n,\vec{x}}$ denote the row concatenation of $\mathscr{X}_n$ and $\vec{x}$. To compute the VOI, we require the posterior distribution on $\theta(\mathscr{X}_{n,\vec{x}})$, which is multivariate normal with mean $\mu_n(\mathscr{X}_{n,\vec{x}})$ and covariance $\Sigma_n(\mathscr{X}_{n,\vec{x}}, \mathscr{X}_{n,\vec{x}})$. We introduce the following expressions for computing these quantities. Let $\mathscr{Y}_n$ be the cumulative column vector of sampling observations up to iteration n, i.e., the columnar concatenation of $\vec{y}_1, \vec{y}_2, \ldots, \vec{y}_n$, so each entry of $\mathscr{Y}_n$ is the observation from the corresponding entry in $\mathscr{X}_n$. Let Γ_n be the block diagonal matrix with n blocks: $\Lambda(\vec{x}_1, \vec{x}_1), \Lambda(\vec{x}_2, \vec{x}_2), \ldots, \Lambda(\vec{x}_n, \vec{x}_n)$. We then define three quantities, the measurement residual $\tilde{\mathscr{Y}}_n$, the residual covariance S_n, and the optimal Kalman gain $K_n(\vec{x})$, by

$$\tilde{\mathscr{Y}}_n = \mathscr{Y}_n - \mu_0(\mathscr{X}_n), \quad S_n = \Sigma_0(\mathscr{X}_n, \mathscr{X}_n) + \Gamma_n,$$
$$K_n(\vec{x}) = \Sigma_0(\mathscr{X}_{n,\vec{x}}, \mathscr{X}_{n,\vec{x}}) L [S_n]^{-1}. \tag{6}$$

Here, the matrix L is defined by concatenating an $|\mathscr{X}_n|$-by-$|\mathscr{X}_n|$ identity matrix with an $|\mathscr{X}_n|$-by-$|\vec{x}_n|$ matrix of zeros, so $L = [I_{|\mathscr{X}_n|}, \vec{0}]^T$ if $\vec{x} = x$, and $L = [I_{|\mathscr{X}_n|}, \vec{0}, \vec{0}]^T$ if $\vec{x} = (x^{(1)}, x^{(2)})$. Here and elsewhere, $|\cdot|$ denotes the length of a vector. We will assume in §6 that Σ_0 and Λ are positive definite. This assumption implies that $\Sigma_0(\mathscr{X}_n, \mathscr{X}_n)$ is positive semidefinite and that Γ_n is positive definite, so that S_n is positive definite and that its inverse $[S_n]^{-1}$ exists.

The posterior mean and covariance matrix of $\theta(\mathscr{X}_{n,\vec{x}})$ at iteration n are then given by

$$\mu_n(\mathscr{X}_{n,\vec{x}}) = \mu_0(\mathscr{X}_{n,\vec{x}}) + K_n(\vec{x})\tilde{\mathscr{Y}}_n, \tag{7}$$
$$\Sigma_n(\mathscr{X}_{n,\vec{x}}, \mathscr{X}_{n,\vec{x}}) = (I_{|\mathscr{X}_{n,\vec{x}}|} - K_n(\vec{x})L^T)\Sigma_0(\mathscr{X}_{n,\vec{x}}, \mathscr{X}_{n,\vec{x}}). \tag{8}$$

In implementing (6), one should not invert S_n directly, as doing so when n is large is both slow and numerically unstable. Instead, one can perform a Cholesky decomposition, and then solve a numerical system, as is described in Rasmussen and Williams (2006, §2.2). This is more stable, and faster. If one implements precisely the above procedure, then the number of statistics grows without bound as n grows without bound. When n gets large, one can "collapse" the above vectors and matrices by tracking only the sample means and sample variance of each alternative, and the sample average of the product of the output of each pair. Thus, the storage required to track this information is $O(\min(n, k^2))$. For further discussion of implementation issues in GP regression, see Rasmussen and Williams (2006).

2.3. Value of Information (VOI)

In the VOI approach, information is valued according to the expected improvement it produces in some decision to be made later (Raiffa and Schlaifer 1961). In this paper, the decision to be made later is which alternative to select as the best and to implement in reality. We call this decision the "implementation decision." The implementation decision and its value are a function of the posterior mean after sampling. In this section, we derive analytic expressions for computing the VOI, resulting from sampling singletons, or sampling the difference between pairs of alternatives.

The value of an implementation decision x is $\theta(x)$ and has expectation $\mu_{n+1}(x)$ under the posterior at iteration $n+1$. Thus, the expected value of the best implementation decision that can be made at iteration $n+1$ is $\max_{x \in \{1,2,\ldots,k\}} \mu_{n+1}(x) = \max \mu_{n+1}$. The increment in this value in going from iteration n to iteration $n+1$ is $\max \mu_{n+1} - \max \mu_n$ and depends on y_{n+1}. Here, the VOI is the expected value of this increment, under the posterior at iteration n, under the hypothetical that an alternative is to be selected after a single stage of sampling.

In this framework, the VOI for a set of β samples collected by observing $\vec{y}_{n+1}$ with a general sampling decision $\vec{x}$ at iteration $n+1$ according to (3) can be written

$$V_n(\vec{x}, \beta) = \mathbb{E}_n[\max \mu_{n+1} \mid \vec{x}_{n+1} = \vec{x}, \beta_{n+1} = \beta]. \tag{9}$$

If the implementation decision is restricted to a set $A_n(\vec{x})$ that may depend upon on $\mathscr{X}_n$, $\mathscr{Y}_n$, and $\vec{x}$, then the VOI is

$$V_n(\vec{x}, A_n(\vec{x}), \beta) = \mathbb{E}_n[\max[\mu_{n+1}(A_n(\vec{x}))] \mid \vec{x}_{n+1} = \vec{x}, \beta_{n+1} = \beta] - \max[\mu_n(A_n(\vec{x}))]. \tag{10}$$

When $A_n(\vec{x}) = \{1, 2, \ldots, k\}$, then $V_n(\vec{x}, A_n(\vec{x}), \beta) = V_n(\vec{x}, \beta)$. This VOI also satisfies a monotonicity property: if

$A \subseteq B$ then $V_n(\vec{x}, A, \beta) \leqslant V_n(\vec{x}, B, \beta)$. This monotonicity property implies that $V_n(\vec{x}, A_n(\vec{x}), \beta)$ is actually a lower bound on $V_n(\vec{x}, \beta)$.

There is no restriction on the implementation decision in practice, but we use $V_n(\vec{x}, A_n(\vec{x}), \beta)$ as an approximation to $V_n(\vec{x}, \beta)$ because it can be computed more quickly, especially when $|A_n(\vec{x})|$ is small. Methods for choosing $A_n(\vec{x})$ are discussed below in §5.2.

2.4. Predictive Distribution for Posterior Means to be Observed

The VOI in (9) or (10) depends on the predictive distribution for $\mu_{n+1}(A)$ that results from a particular decision to sample $\vec{x}_{n+1}$ for β_{n+1} times, for any given set A. We consider two specific types of sampling decisions $\vec{x}_{n+1}$: observing singletons $\vec{x}_{n+1} = (x_{n+1})$ as in (4); and observing the difference between a pair of alternatives $\vec{x}_{n+1} = (x^{(1)}_{n+1}, x^{(2)}_{n+1})$ as in (5). Observing either the singleton y_n or the difference $y^{(1)}_n - y^{(2)}_n$ admits an analytic expression for $V_n(\vec{x}, A, \beta)$ below. Observing both $y^{(1)}_n$ and $y^{(2)}_n$ together does not: we use the VOI of sampling their difference as a lower bound on the VOI of observing both values. This lower bound proves to be useful in numerical experiments.

For both singletons and differences between pairs, the predictive distribution is

$$\mu_{n+1}(A) \mid \mathscr{X}_n, \mathscr{Y}_n, \vec{x}_{n+1}, \beta_{n+1} \sim \text{Normal}\big(\mu_n(A), \tilde{\sigma}_n(\vec{x}_{n+1}, A, \beta_{n+1}) \cdot \tilde{\sigma}_n(\vec{x}_{n+1}, A, \beta_{n+1})^T\big), \quad (11)$$

where $\tilde{\sigma}_n(\vec{x}_{n+1}, A, \beta_{n+1})$ is a $|A| \times 1$ vector defined, respectively, in the two cases as

$$\tilde{\sigma}_n(x, A, \beta) = \frac{\Sigma_n(A, x)}{\sqrt{\beta^{-1}\Lambda(x, x) + \Sigma_n(x, x)}},$$
$$\tilde{\sigma}_n((x^{(1)}, x^{(2)}), A, \beta) = \frac{\Sigma_n(A, x^{(1)}) - \Sigma_n(A, x^{(2)})}{\sqrt{\beta^{-1}P + Q_n}}, \quad (12)$$

which follows directly from Frazier et al. (2011, §2.2). Here, $\Sigma_n(A, x)$ is a column vector containing the entries from Σ_n in column x with rows in A, and P and Q_n are defined by

$$P = \Lambda(x^{(1)}, x^{(1)}) + \Lambda(x^{(2)}, x^{(2)}) - 2\Lambda(x^{(1)}, x^{(2)}),$$
$$Q_n = \Sigma_n(x^{(1)}, x^{(1)}) + \Sigma_n(x^{(2)}, x^{(2)}) - 2\Sigma_n(x^{(1)}, x^{(2)}). \quad (13)$$

This expression will be used in §3 to compute the VOI in (10) explicitly.

3. Evaluation of the Value of Information and Knowledge Gradient

This section describes computational issues for the general VOI principles described in §2.3 when applied to sampling singletons or pairs with CRN. These results are used for the sampling allocation rules in §5 based on the so-called knowledge gradient (KG). Qualitatively, the KG is a rate of information per unit of computing effort.

3.1. Evaluation of the Value of Information

From (11), we know that when $\mathscr{X}_n$, $\mathscr{Y}_n$, $\vec{x}_{n+1}$, and β_{n+1} are given, $\mu_{n+1}(A)$ is equal in distribution to $\mu_n(A) + \tilde{\sigma}_n(\vec{x}_{n+1}, A, \beta_{n+1})Z$, where Z is a standard normal random variable. Using this observation in (10) shows that

$$V_n(\vec{x}, A_n(\vec{x}), \beta) = \mathbb{E}_n[\max[\mu_n(A_n(\vec{x})) + \tilde{\sigma}_n(\vec{x}, A_n(\vec{x}), \beta)Z]] - \max[\mu_n(A_n(\vec{x}))]. \quad (14)$$

To compute (14), we consider three cases: when $A_n(\vec{x})$ has one, two, or more than two elements. This third case is the most common in the allocation rules developed in §5.

When $A_n(\vec{x})$ has exactly one element, one can show that $V_n(\vec{x}, A_n(\vec{x}), \beta) = 0$. In other words, if only one alternative can ever be selected, information has no value.

When $A_n(\vec{x})$ has exactly two elements, computation of $V_n(\vec{x}, A_n(\vec{x}), \beta)$ is similar to related computations for the VOI in a pairwise comparison (Frazier et al. 2008, Jones et al. 1998, Chick and Inoue 2001a). Namely, let Δ be the absolute value of the difference of $\mu_n(x)$ between the two different $x \in A_n(\vec{x})$, and let s be the absolute value of the difference of the two components of $\tilde{\sigma}_n(\vec{x}, A_n(\vec{x}), \beta)$. Then

$$V_n(\vec{x}, A_n(\vec{x}), \beta) = sf(-\Delta/s),$$

where $f(-z) = \varphi(z) - z\Phi(-z)$, and φ and Φ are the density and cumulative distribution functions, respectively, of a standard normal random variable.

When $A_n(\vec{x})$ contains more than two elements, computation of $V_n(\vec{x}, A_n(\vec{x}), \beta)$ is more involved, but still can be performed analytically. Recalling (14), we see that we can write

$$V_n(\vec{x}, A_n(\vec{x}), \beta) = h\big(\mu_n(A_n(\vec{x})), \tilde{\sigma}_n(\vec{x}, A_n(\vec{x}), \beta)\big), \quad (15)$$

where $h(a, b) = \mathbb{E}[\max_i a(i) + b(i)Z] - \max_i a(i)$ for two vectors a and b of equal length. Frazier et al. (2009) gives an exact algorithm for computing h, and Frazier (2009–2010) provides a Matlab implementation. More details are given in Appendix B.1.

Lemma 1 says that the VOI from sampling a pair is increasing in the correlation of the simulation output. We take advantage of this fact in implementations as described in §7.1.

Lemma 1. *Suppose* $\vec{x} = (x^{(1)}, x^{(2)})$. *Let* μ_n, Σ_n, $\Lambda(x^{(1)}, x^{(1)})$, $\Lambda(x^{(2)}, x^{(2)})$ *be fixed. Then for any* A *and* β, $V_n(\vec{x}, A, \beta)$ *is an increasing function of the sampling correlation between* $x^{(1)}$ *and* $x^{(2)}$,

$$\rho(x^{(1)}, x^{(2)}) = \frac{\Lambda(x^{(1)}, x^{(2)})}{\sqrt{\Lambda(x^{(1)}, x^{(1)})\Lambda(x^{(2)}, x^{(2)})}}.$$

3.2. Knowledge Gradient Factors

The KG factor is a metric that measures the VOI per unit of sampling effort. The KG factor uses the predictive

distribution in (11) and the computational cost $c(\vec{x})$ of sampling at $\vec{x}$, measured by the computation time required per sample.

Thus, the KG_β factor at iteration n for observing the value at a given singleton $x \in \{1, 2, \ldots, k\}$ is

$$\nu_n^{\mathrm{KG}_\beta}(x) = V_n(x, A_n(x), \beta_n)/[\beta_n c(x)], \tag{16}$$

where β_n and $A_n(\cdot)$ may be chosen in an implementation-specific way (see §5.2). Similarly, the KG_β factor at iteration n for observing the difference in value between a pair of alternatives $(x^{(1)}, x^{(2)})$ is

$$\nu_n^{\mathrm{KG}_\beta}(x^{(1)}, x^{(2)}) = \frac{V_n((x^{(1)}, x^{(2)}), A_n(x^{(1)}, x^{(2)}), \beta_n)}{\beta_n c((x^{(1)}, x^{(2)}))}. \tag{17}$$

If the computation time for a sample does not depend on $\vec{x}$, then $c(\vec{x}) = c|\vec{x}|$, where c is a positive constant cost per sample, and $|\vec{x}|$ is the length of $\vec{x}$. We adopt this model in numerical tests below.

4. Generic Sampling Algorithm

We now formalize our proposed DOvS algorithm. The notation in §2 allows us to formalize it in a way that is amenable to handling a very large number of alternatives: statistics are tracked only for alternatives that have been sampled or are being considered for sampling in the next iteration.

The algorithm samples in a sequential manner. This requires the specification of an allocation rule, which maps $\mathscr{X}_n$, $\mathscr{Y}_n$ to a sampling decision (i.e., a set of alternatives to sample next), and a stopping rule, which decides whether or not to stop sampling. The allocation rules we use are based on VOI principles described in §2.3 and are presented in §5. The default stopping rule we use in this paper is to stop after a prespecified number of samples is observed.

The generic algorithm is written to be able to handle either a known or an unknown sampling covariance matrix Λ. When it is unknown, as is typical in applications, the relevant statistical parameters are estimated as described below.

1. *Initialize*: Select an allocation rule and a stopping rule. If the sampling covariance Λ and the mean vector μ_0 and the covariance matrix Σ_0 for the unknown sampling means θ are known, then specify these parameters, initialize $n = 0$ to be the number of iterations done so far, and initialize $\mathscr{X}_0$ and $\mathscr{Y}_0$ to be empty vectors. If Λ, μ_0, and Σ_0 are not all known, then describe the functional forms of Λ, μ_0, and Σ_0 in terms of a collection of parameters (see §7.1), and make initial iterations of sampling to estimate those parameters, setting n, $\mathscr{X}_n$ and $\mathscr{Y}_n$ accordingly (see §7.2).

2. *Update parameters* (*Empirical Bayes*): If the parameters determining Λ are unknown and their estimates are to be updated, then use the maximum likelihood estimator described in §7.2 to estimate them using all data (collected in $\mathscr{X}_n$ and $\mathscr{Y}_n$).

3. *Check allocation and stopping rule*: If the stopping rule says to stop sampling, go to step 5. Otherwise, use the allocation rule to choose a set of alternatives, $\vec{x}_{n+1}$, to simulate next.

4. *Sample*: Simulate alternatives $\vec{x}_{n+1}$ (using CRN if called for) to get simulation output $\vec{y}_{n+1}$. Concatenate $\vec{y}_{n+1}$ with $\mathscr{Y}_n$ to get $\mathscr{Y}_{n+1}$, and $\vec{x}_{n+1}$ with $\mathscr{X}_n$ to get $\mathscr{X}_{n+1}$. Increment n, go to step 2.

5. *Selection rule*: Select as the best the alternative in $\mathscr{X}_n$ with the largest posterior mean. This can be found by computing $\mu_n(\mathscr{X}_n)$ according to (7) with $\mathscr{X}_{n,\vec{x}} = \mathscr{X}_n$, and then taking the largest component of this vector.

Step 4 samples from the simulation black box, not from a metamodel of the simulation output.

5. Allocation Rules

This section discusses allocation rules, which use previous sampling information to decide how to take the next sample or samples, and which appear in step 3 of the generic sampling algorithm in §4. The allocation rules discussed all search over a set of possible sampling decisions to find the one with the largest KG_β factor, but differ in the way in which this search is performed.

Let $\Xi = \{1, 2, \ldots, k\} \cup \{(x^{(1)}, x^{(2)}) \in \{1, 2, \ldots, k\}^2 \colon x^{(1)} \neq x^{(2)}\}$ denote the set of all singletons and pairs. For each allocation rule below, we let $\Xi_n \subseteq \Xi$ denote a possibly smaller set of sampling decisions available at iteration n. For each n, an allocation rule selects the sampling decision that maximizes the KG_β factor from §3.2 over this set,

$$\vec{x}_n = \arg\max_{\vec{x} \in \Xi_n} \nu_n^{\mathrm{KG}_\beta}(\vec{x}). \tag{18}$$

The expression (18) depends upon the choice for Ξ_n, and implicitly on the choice of β_n and $A_n(\vec{x})$ used to calculate $\nu_n^{\mathrm{KG}_\beta}(\vec{x})$. Thus, different allocation rules are specified by different methods for choosing Ξ_n, β_n, and $A_n(\vec{x})$. Certain ways of choosing the Ξ_n will be shown to improve the computation time of the algorithm while retaining theoretical convergence guarantees (in §6) and good empirical performance (in §8).

5.1. Classes of KG Allocation Rules

We define a class of allocation rules, called *KG_β allocation rules*, to be any that includes at least one singleton and one pair of alternatives in Ξ_n, and includes both $x^{(1)}$ and $x^{(2)}$ in $A_n(\vec{x})$, if $\vec{x} = (x^{(1)}, x^{(2)})$, for each n. Within this larger class, we now define two more specific types of KG_β^2 allocation rules, which place additional conditions on Ξ_n.

An *idealized* KG_β^2 allocation rule (proposed in Frazier et al. 2011) is one in which $\Xi_n = \Xi$ for each n. Thus, an idealized KG_β^2 allocation rule looks over all the singleton and pairwise-difference KG_β factors and finds the largest one.

When k is large, the exhaustive maximization in (18) by an idealized KG_β^2 rule is too computationally intensive. Frazier et al. (2011) proposed an alternative to this exhaustive maximization, which checks only singletons and a subset of pairs of alternatives, but even that approach is too computationally intensive when $k \gg 10^3$ and is not easily amenable to theoretical analysis.

To allow for better performance in large problems in a way that also supports theoretical analysis, we propose here two new classes of KG^2_β allocation rules. The first is the class of *randomized* KG^2_β allocation rules, which chooses the set Ξ_n of eligible sampling decisions iteration n to contain $n_s \geqslant 1$ singletons from $\{1, 2, \ldots, k\}$ and n_p pairs from $\{(x^{(1)}, x^{(2)}) \in \{1, 2, \ldots, k\}^2 \colon x^{(1)} \neq x^{(2)}\}$. Although the selection of these singletons and pairs can happen according to a fixed schedule, in our algorithm we select them randomly with uniform distribution. If $n_s + n_p \ll k$, this results in a considerable reduction in the number of KG factor evaluations in (18).

The second new class contains *accelerated* KG^2_β allocation rules. Accelerated allocation rules are motivated by the heuristic that an increased KG factor can increase the speed of finding better solutions, and use a local search to locally optimize the KG factor in the neighborhoods of randomly sampled solutions. We let $\tilde{f}(\vec{x})$ be such a "local search" function that returns a sampling decision whose KG factor is at least as great as that of $\vec{x}$. An accelerated KG^2_β allocation rule selects $n_s \geqslant 1$ singletons and n_p pairs, and lets Ξ_n contain the sampling decisions determined by applying the local search function to them. That is, for each randomized KG^2_β allocation rule with sets Ξ'_n, then, there is an accelerated KG^2_β allocation rule with sets $\Xi_n = \{\tilde{f}(x') \colon x' \in \Xi'_n\}$.

The class of KG_β *allocation rules* is defined analogously to the class of KG^2_β allocation rules, except that only singletons (not pairs) may be sampled. That is, $\Xi_n \subseteq \{1, 2, \ldots, k\}$ for KG_β allocation rules. The notions of idealized, randomized, and accelerated KG_β allocation rules are defined as for the KG^2_β allocation rules, except that pairs are excluded. The allocation rule proposed in Frazier et al. (2009) is the idealized KG_β with $\Xi_n = \{1, 2, \ldots, k\}$, $A_n(\vec{x}) = \{1, 2, \ldots, k\}$, and $\beta_n = 1$.

5.2. Specification of Parameters of KG Allocation Rules

Specific instances of the KG_β and KG^2_β allocation rules from §5.1 require the specification of the parameters β_n, $A_n(\vec{x})$, and Ξ_n. The accelerated allocation rules also require the specification of a local search function $\tilde{f}$. We discuss these in turn.

Except where otherwise noted in our numerical experiments, we set $\beta_n = 1$ and we chose $A_n(\vec{x})$ to be the alternatives in $\vec{x}$ and the best other sampled alternative given the observations available. So, for singletons $\vec{x} = (x)$, we set $A_n(x) = \{x, x_*\}$, where $x_* = \arg\max_{x' \in \mathscr{X}_n \setminus \{x\}} \mu_n(x')$. For pairs, $\vec{x} = (x^{(1)}, x^{(2)})$, we set $A_n(\vec{x}) = \{x^{(1)}, x^{(2)}, x_*\}$, where $x_* = \arg\max_{x' \in \mathscr{X}_n \setminus \{x^{(1)}, x^{(2)}\}} \mu_n(x')$.

The choice of Ξ_n used in our numerical experiments below requires additional notation. Denote the best and second best alternatives (in terms of posterior mean) after n samples as

$$x_{n,b} = \arg\max_{x \in \mathscr{X}_n} \mu_n(x), \quad x_{n,s} = \arg\max_{x \in \mathscr{X}_n \setminus \{x_{n,b}\}} \mu_n(x). \tag{19}$$

In the first iterations, $\mathscr{X}_n$ may have too few elements for x_*, $x_{n,b}$, or $x_{n,s}$ to be defined. In such a case, a random sampling decision is used instead.

In randomized KG_β allocation rules, we tested the choice of using n_s randomly chosen singletons, plus the option of including $x_{n,b}$, or both $x_{n,b}$ and $x_{n,s}$. In randomized KG^2_β allocation rules, we tested the choice of including n_p randomly chosen pairs in Ξ_n, plus the various combinations of singletons as in the KG_β allocation rules. We also tested the inclusion of the pair $(x_{n,b}, x_{n,s})$ in Ξ_n. We tested several values for n_s and set $n_p = n_s/2$ unless otherwise specified below.

For accelerated allocation rules, we applied a local search function $\tilde{f}$ to each instance of a randomized KG_β or randomized KG^2_β allocation rule mentioned above. The $\tilde{f}$ we tested is based on a gradient search to do hill climbing to find alternatives with improved KG factors. In particular, we embed the lattice of sampling decisions in a continuous space, extend the KG factor from the lattice to the reals in a natural way by using a continuous Gaussian process model as described in §7, use a gradient search (Matlab's `fmincon` function) to find a local maximum in continuous space near each given $\vec{x} \in \Xi_n$, then project it to the closest feasible sampling decision $\vec{x}'$. If $\nu_n^{\mathrm{KG}_\beta}(\vec{x}') > \nu_n^{\mathrm{KG}_\beta}(\vec{x})$ then we set $\tilde{f}(\vec{x}) = \vec{x}'$, else we set $\tilde{f}(\vec{x}) = \vec{x}$. The relevant gradients are derived in Appendix B.

In summary, there are a variety of ways to choose Ξ_n, β_n, $A_n(\vec{x})$, and $\tilde{f}$. §8 focuses on assessing ways to select Ξ_n in combination with the new gradient techniques embedded in $\tilde{f}$. Appendix EC.2, in the online appendix, discusses our choices and additional implementation issues for potential further exploration.

6. Convergence Properties

This section shows that the generic sampling algorithm from §4, when used with known Λ, μ_0, and Σ_0, and with a KG^2_β allocation rule from §5 satisfying mild conditions, samples every alternative infinitely often, so that we learn the value of every alternative, and are able to find a global maximum $x^* \in \arg\max_x \theta(x)$ almost surely in the limit as the number of samples grows without bound. Frazier et al. (2009) proved these consistency results for the idealized KG_β algorithm with $A_n(\vec{x}) = \{1, 2, \ldots, k\}$ and $\beta_n = 1$, and so the results here can be viewed as a generalization to KG^2_β and to algorithms that do not require exhaustive optimization over all alternatives. The presence of sampling correlations, however, requires substantially different proof techniques from those used in Frazier et al. (2009).

These results depend on two assumptions and a condition, which are stated precisely below. The first assumption states that the parameters Λ, μ_0, and Σ_0 are known and fixed. The second assumption states that there is genuine uncertainty about each alternative's performance. The condition restricts the choice of KG^2_β allocation rule, and is satisfied by the idealized KG^2_β and accelerated KG^2_β allocation rules from §5 as long as every $\vec{x} \in \Xi$ is chosen as a starting point for the

local search infinitely often, with probability one. This can be assured by randomly selecting alternatives to include in $\{\Xi_n\}$ infinitely often in those allocation rules.

ASSUMPTION 1. *μ_0, Σ_0, and Λ are known.*

ASSUMPTION 2. *Σ_0 and Λ are positive definite.*

CONDITION 1. *Each $\vec{x} \in \Xi$ is included in Ξ_n infinitely often, with probability* 1.

We now state our main result: we become certain of the vector of true means θ eventually, as the conditional variance $\Sigma_n(x, x)$ of $\theta(x)$ converges to 0, and the conditional mean $\mu_n(x)$ converges to $\theta(x)$, for each x; and that the implementation decision that would be chosen if sampling stopped at iteration n, $\arg\max_x \mu_n(x)$, is eventually globally optimal. The proof may be found in Appendix A.

THEOREM 1. *If Assumptions 1 and 2 hold, and if sampling occurs according to a KG^2_β allocation rule satisfying Condition 1, then $\lim_{n\to\infty} \Sigma_n(x, x) = 0$ almost surely for each x; $\lim_{n\to\infty} \mu_n(x) = \theta(x)$ almost surely and in L^2 for each x; and $\lim_{n\to\infty} \arg\max_x \mu_n(x) = \arg\max_x \theta(x)$ almost surely.*

7. Implementation Features and Practicalities

This section discusses practical implementation choices arising in steps 1 to 3 of the generic algorithm in §4 and allocation rules in §5. This includes the specification of the functional form of the prior distribution in step 1, the empirical Bayes estimator (or Bayes motivated) used to assess μ_0, Σ_0, and Λ in step 2, the choice of $A_n(\vec{x})$ and β_n used in KG^2_β allocation rules, and derivations of the gradients of the VOI and KG factors used by accelerated KG^2_β allocation rules. The convergence results in §6 do not depend on how these implementation issues are addressed, as long as Assumptions 1 and 2, and Condition 1 are valid.

Several of the implementation choices discussed assume that the k alternatives may be represented as elements in a lattice in $\mathbb{Z}^d$. For example, in a manufacturing problem, there may be d decision variables, each of which represents the number of resources (machines, employees with given skill sets, etc.) that are combined to define a specific alternative manufacturing system design. That is, for any alternative x, we can specify its grid coordinates $\{\zeta_i(x)\}_{i=1}^d$, where $\zeta_i(x)$ is the ith coordinate for the embedding of x in $\mathbb{Z}^d$ (for example, the number of resources of the ith type).

7.1. Functional Form of the Prior Distribution and Sampling Covariance (Step 1)

Step 1 of the generic sampling algorithm requires specification of the functional form of the sampling covariance and prior distribution for the unknown means, either fully, or more frequently in terms of parameters to be estimated later in step 2. We discuss this choice here.

The functional form of the sampling covariance Λ is considered first. Although several different forms are possible, we assume compound sphericity for simplicity. The compound sphericity assumption means that Λ can be specified with exactly two parameters: a common sampling variance σ^2_ϵ on the diagonals and a common sampling correlation across any pair of alternatives, ρ. All off-diagonal elements of Λ are the same. Although the compound sphericity assumption is strong, it has been used by others to model the effect of CRN (Schruben and Margolin 1978, Tew and Wilson 1992), including in the context of CRN with kriging (Chen et al. 2012). By Lemma 1, one may simulate with independent samples rather than with CRN if the sampling correlation is estimated to be negative for a given pair of alternatives and for data observed to the time of the decision.

We now discuss the functional form of the prior distribution for the unknown means. When the alternatives may be embedded in a lattice, there may be a belief that the performance of two alternatives that are "near" each other in this lattice are more likely to be similar than the performance of two alternatives that are "distant" from each other. This motivates the notion that the prior distribution may be a multivariate normal distribution under which the covariance between the values of any two alternatives is a decreasing function of their distance from each other on the lattice. This is analogous to covariance functions used in GP priors over continuous functions. Inspired by this link to GP priors, we adopt the commonly used Gaussian kernel:

$$\Sigma_0(x, x') = \sigma_0^2 \exp\left\{-\sum_{i=1}^{d} \alpha_i [\zeta_i(x) - \zeta_i(x')]^2\right\}. \tag{20}$$

Here σ_0^2 is the homogeneous prior variance of the unknown means and $\vec{\alpha} = \{\alpha_i\}_1^d$ is a vector of scaling parameters. We also let η be a parameter for the mean in this model and let $\vec{1}$ be a vector of k ones, so that (20) and $\mu_0 = \eta\vec{1}$ define the prior distribution in (2).

Specification of the prior distribution parameters μ_0, Σ_0 can therefore be accomplished by specifying σ_0^2, $\vec{\alpha}$, and η. Kernels other than that in (20) would be handled similarly.

7.2. Initial Iteration of Sampling (Step 1) and Empirical Bayes Parameter Update (Step 2)

Here we discuss the initial iteration of sampling performed in step 1, and the periodic empirical Bayes updates performed in step 2 of the generic sampling algorithm. These steps are used when Λ, μ_0, Σ_0 or some parameters of their functional forms are unknown, and require some estimation.

If an initial iteration of sampling is required to estimate unknown parameters, we randomly select a set $\vec{x}_{01}$ of N_1 alternatives, sample once from each of them using CRN, sort them in descending order. We then take another sample from each of the first N_2 alternatives, denoted by the vector $\vec{x}_{02}$, using CRN ($N_2 \leqslant N_1$), to estimate correlations among the alternatives with better sampling means. We initialize the number of iterations so far to be $n = 2$ (one for each use of CRN), $\mathscr{X}_2$ to be the row concatenation of $\vec{x}_{01}$ and $\vec{x}_{02}$, and $\mathscr{Y}_2$ to be the outputs at those alternatives.

Once this initialization is complete, and also periodically thereafter according to a fixed schedule, we estimate the parameters determining μ_0, Σ_0, and Λ in step 2 of the generic sampling algorithm using maximum likelihood estimators (MLEs). Appendix EC.1, in the online appendix, derives MLEs assuming that μ_0, Σ_0, and Λ take the functional form specified in §7.1, which has parameters σ_0^2, $\vec{\alpha}$, η, σ_ϵ^2, and ρ. This use of maximum likelihood estimation to estimate parameters within a Bayesian model is known as an empirical Bayes approach and is common in GP regression.

Relaxing the compound sphericity assumption or using a different GP prior in our proposed algorithm simply involves providing alternative MLEs for Λ, μ_0, and Σ_0.

We let $\mathcal{N}$ denote the set of iterations at which the maximum likelihood estimation will be performed, so $\mathcal{N}$ contains $N_1 + N_2$. If computation time for the allocation rule is unimportant (e.g., because the simulations themselves are very time consuming), one may perform maximum likelihood estimation before each new iteration, in which case $\mathcal{N} = \{N_1 + N_2, N_1 + N_2 + 1, N_1 + N_2 + 2, \ldots\}$. In other situations, because computation of the MLE may be time consuming, it may be beneficial to avoid recomputing the MLE at every iteration. In our implementation, we update the MLE more frequently at first when additional samples tend to have more impact on parameter estimates, and then less frequently as more samples are acquired. If the parameters are known, we may skip these updates by setting $\mathcal{N} = \varnothing$.

8. Numerical Results

Beyond asymptotic convergence to the optimal solution, we are interested in the rate in which solutions improve for even small numbers of samples. We measure this performance by the expected opportunity cost (the difference between the true best and the estimated best $x_{n,b}$, as defined in §7, at each iteration n), $\mathbb{E}[\max_x \theta_x - \theta_{x_{n,b}} \mid \theta = \theta_0]$, where θ_0 is the "true" mean vector. This expectation averages over uncertainty about θ (in problems when θ_0 is not known, as in §8.1) and uncertainty about $x_{n,b}$ (in each of the examples below), which depends on the realized samples in applications of an allocation rule.

In this section we present numerical results to explore the behavior of the proposed algorithm, allocations rules, and implementation choices from §7 in order to answer the following questions. Does pairwise sampling with CRN provide an efficiency benefit, even if approximations are made to simplify computations? How much benefit can the KG and KG^2 allocation rules give, on problems with combinatorially large numbers of solutions, as compared to other benchmark algorithms such as a random search that is enhanced with a Gaussian process metamodel (which we call RSGP and describe below) and Industrial Strength COMPASS (Xu et al. 2010).

Except as noted below, the KG and KG^2 allocation rules used the Gaussian process prior for unknown means, compound sphericity assumption for samples, maximum likelihood estimation, and other parameters as described in §7. When μ_0, Σ_0, and Λ were not known, the parameters for the initial iteration were $N_1 = 10d$, $N_2 = 2d$, where d is the dimension of the problem, and we let $\mathcal{N}$ contain $(N_1 + N_2)\{2^0, 2^{0.5}, 2^1, 2^{1.5}, \ldots\}$, rounding up to an integer if necessary, so that the number of samples between MLE updates increased by a factor of $\sqrt{2}$ each time. In cases where sampling was done without CRN, we performed maximum likelihood estimation with ρ fixed to 0.

8.1. How Do the Approximations Interact?

This section assesses the relative importance of several features and approximations described above: the allocation rule, approximations due to accelerated allocations and parameter estimation, and deviations from the assumed sampling correlation structure under CRN. Specifically, we assess the $12 = 2 \times 3 \times 2$ combinations that result from combining each level of the following three factors:

Allocation. KG$_\beta$ allocation rule (no CRN); or KG$_\beta^2$ allocation rule (CRN allowed).

Approximation. Idealized allocation rule with known parameters; accelerated allocation rule with known parameters; or accelerated allocation with unknown parameters $(\sigma_0^2, \vec{\alpha}, \eta, \sigma_\epsilon^2, \rho)$ fit as in §7.2.

Sampling with CRN. Samples satisfy compound sphericity (with $\rho(i, j) = 0.25$ for $i \neq j$); or decreasing correlations (with $\rho(i, j) = \exp[-(i - j)^2/50]$ for $i \neq j$) even though compound sphericity may be (incorrectly) assumed by the parameter fitting.

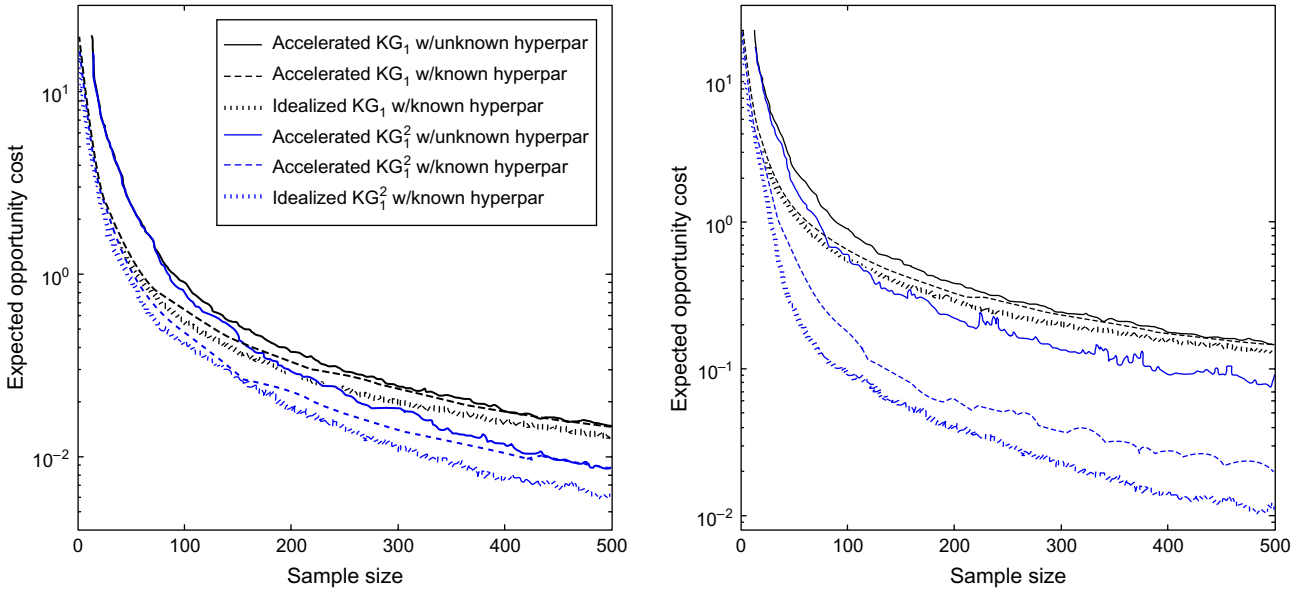
We do so for randomly generated problem instances with a small (100) number of alternatives. We generate 500 problem instances ($\theta_{0,i}$ for $i = 1, 2, \ldots, 500$) and average the opportunity cost across those problem instances. In each problem, the 100 alternatives had means distributed as a Normal(μ_0, Σ_0) with $\mu_0 = \vec{0}$ and $\Sigma_0(i, j) = 100 \exp[-(i - j)^2/50]$ for $i, j = 1, 2, \ldots, 100$. We assumed a homogeneous sampling variance $\sigma_\epsilon^2 = 50$. We set $\beta = 1$ and so refer to KG$_1$ and KG$_1^2$.

Figure 1 shows the expected opportunity cost of a potentially incorrect selection, on a logarithmic scale, as a function of the total number of samples. The maximum size of the 95% confidence intervals is 0.28 at sample size 100, 0.09 at sample size 300, and 0.05 at sample size 500.

Not surprisingly, idealized KG allocation rules performed better than their accelerated counterparts (idealized rules do an exhaustive, not local, search to maximize the KG factor). The degree of suboptimality was not particularly great in any setting where parameters were known. A greater degree of suboptimality was seen when parameter estimation was used. The deterioration due to parameter estimation was not significant for the KG$_1$ allocation, even when sphericity did not apply and parameters were (incorrectly) estimated with the sphericity assumption (right panel, top three lines). The degradation in performance due to parameter estimation with the KG$_1^2$ allocation was not too significant when sphericity was correctly assumed (left panel, bottom three curves).

All else fixed, a KG^2 allocation with CRN improved upon the performance of its corresponding KG allocation with

Figure 1. Performance of selected algorithms in the grid test problem with compound sphericity (left plot) and decreasing correlations (right plot).



independent sampling. Thus, the ability of sampling pairs with CRN offered an important benefit beyond sampling only one alternative independently at a time (both panels).

Moreover, we observed that the accelerated KG_1^2 allocation rule, even when parameter estimation was used, performed better than the idealized KG_1 allocation, which had the advantage of "knowing" the true sampling correlation and of doing an exhaustive search over KG factors. Thus, the benefit of CRN outweighed the penalties associated with suboptimality in the accelerated KG_1^2 allocation rule with unknown parameters, once 200 samples were observed to get stable parameter estimates (even when sphericity was incorrectly assumed by the MLE, right panel).

In experiments not shown here for reasons of space, we found other interesting observations. One, when we set $\rho(i,j) = 0.5$ rather than $\rho(i,j) = 0.25$ for all $i \neq j$, the expected opportunity costs decreased. This is consistent with the benefit offered by sampling pairs being increasing in a (common) sampling correlation ρ. Two, we experimented with the number of randomly selected singletons and pairs that were included in Ξ_n for the accelerated allocations. Increasing that number from 1 to 3 provided a practical improvement in performance in the approximate KG and KG^2 allocations, but the benefit of adding random points beyond 4 or 5 had little marginal increase (see Appendix EC.2 in the online appendix). Figure 1 shows the case of $n_s = 4$. Three, experiments with several values of β_n for KG_β^2 allocation rules did not reveal a large difference in performance because of the choice of β_n.

8.2. Comparison with RSGP on a Rosenbrock Problem with 10^6 Alternatives

This section explores the performance of the procedures when there considered is a discretized version of a six-dimensional Rosenbrock function with 10^6 alternatives. Each alternative x corresponds to a point in the grid with coordinates $\zeta(x) = \{\zeta_i(x)\}_{i=1}^6 \in [-0.8, -0.5, \ldots, 1.9]^6$ and has value

$$\theta_0(x) = -\sum_{i=1}^{5}\left[100[\zeta_i(x)^2 - \zeta_{i+1}(x)]^2 + [\zeta_i(x) - 1]^2\right].$$

The computation required for idealized KG allocation rules is not practical when there are such a large number of alternatives. This section assesses differences in performance between the accelerated KG_1 and accelerated KG_1^2 allocation rules, as well as a benchmark algorithm that we introduce, called RSGP. The RSGP samples uniformly at random and uses the Gaussian process model and parameter estimation tools in §7 to estimate the performance for each alternative by its posterior mean when selecting the best alternative.

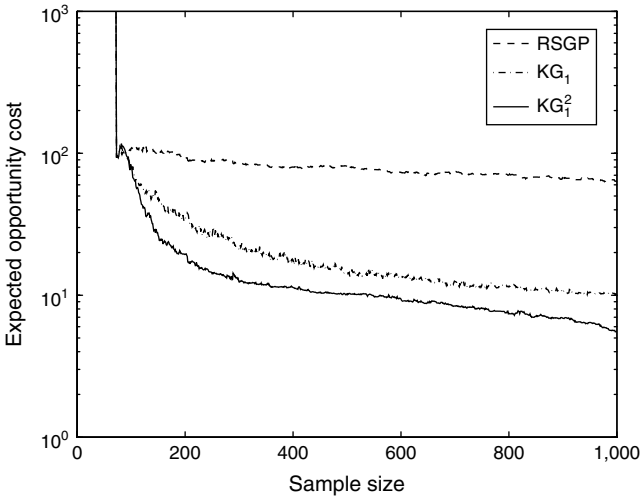
The sampling noise satisfies the compound sphericity assumption, with $\sigma_\epsilon^2 = 125$ and $\rho(i,j) = 0.4$ for all $i \neq j$. These values were assumed unknown in this test, and the empirical Bayes approach described in §7.2 was used to estimate the GP prior and sampling covariance.

Figure 2 shows the opportunity cost, averaged over 100 sample paths, of the accelerated KG_1 and accelerated KG_1^2 allocation rules, and the benchmark RSGP. Both KG allocation rules dramatically outperformed RSGP. This is because the KG factors steered sampling to areas that more efficiently identified local extrema. The KG_1^2 allocation rule outperformed KG_1 because the use of CRN helped KG_1^2 find better solutions more quickly, as compared to KG_1.

8.3. Comparison with ISC on the Assemble to Order Problem

We now compare the accelerated KG_1 and KG_1^2 allocation rules with a well-known algorithm, Industrial Strength

Figure 2. Expected opportunity cost for a benchmark algorithm, (Random Search with Gaussian Processes, RSGP), accelerated KG_1 (KG_1) and accelerated KG_1^2 (KG_1^2) allocation rules as a function of the total number of samples (on a discrete Rosenbrock function).



COMPASS (ISC, developed by Xu et al. 2010). We do so for the assemble to order (ATO) problem described in Hong et al. (2012), which is a variation on the problem studied by Hong and Nelson (2006), and has a combinatorially large number (21^8) of alternatives.

In the ATO problem, orders for 5 different products arrive according to independent Poisson processes with constant arrival rates. Products are made up of a collection of items of 8 different types. Items are either key items or nonkey items. If any of the key items are out of stock then the product order is lost. If all key items are in stock, then the order is assembled from all key items and the available nonkey items. Each item sold brings a profit, and each item in inventory incurs a holding cost per unit time. There is an inventory capacity of 20 for each item. Items are produced one at a time on dedicated machines. The production time for each item is normally distributed, truncated at 0. The system operates under a continuous-review base stock policy under which each item l has a target base stock b_l, and each demand for an item triggers a replenishment order for that item. Each simulation replication starts from a fully stocked system with no orders in production, has a warm-up period of 20 time units, then captures statistics for the next 50 time units of operation. See Appendix EC.2, in the online appendix, for an assessment of the normality of this time-averaged response. The goal is to maximize the expected total profit per unit time by selecting the target inventory level vector $b = (b_1, b_2, \ldots, b_8)$. See Hong et al. (2012) for more details and code.

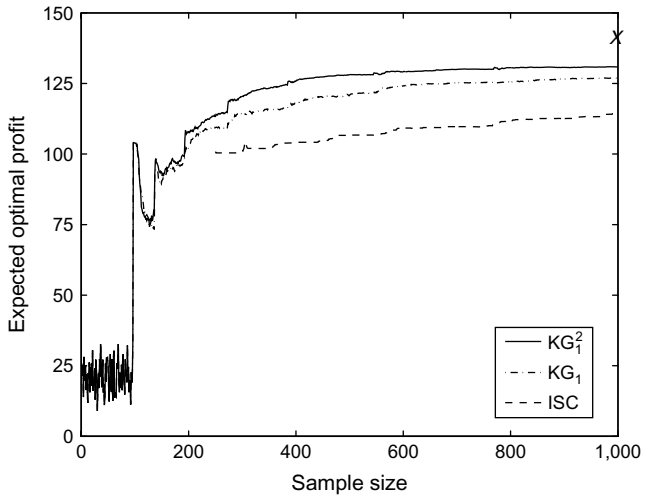
We calculate each algorithm's performance by collecting the true expected total profit θ_0 (estimated separately through exhaustive simulation) of the algorithm's current solution as a function of the sample size. We then average this value over 100 independent sample paths for each algorithm. Each solution reported by the KG algorithms in the initial stage is selected uniformly at random from the feasible set, $\{b\colon 0 \leqslant b_l \leqslant 20, b_l \in \mathbb{Z}\}$. ISC does not report a solution during its initial stage of 250 measurements.

Figure 3 shows the average performance of the three algorithms. This average performance jumped when algorithms finished their initialization phases (step 1 of the KG algorithms), which occurred at 250 samples for ISC and 95 samples for the two KG algorithms. The height of the x at the right edge of the plot (at $x = 1{,}000$) gives the value (141, estimated through exhaustive simulation) of the best solution found by all sample paths across all three algorithms. This best solution was discovered by a KG algorithm. The true optimal solution is unknown.

The accelerated KG_1^2 allocation rule outperformed the accelerated KG_1 allocation rule, which in turn outperformed ISC for this problem, in terms of achieving a higher quality solution with fewer samples. ISC required an average of 1,084 samples to complete and its average profit upon completion was 115.53. To reach this same level of solution quality achieved by ISC, KG_1 took 341 samples on average, whereas KG_1^2 took 273 samples. The two KG algorithms required significantly fewer samples than did ISC, with KG_1^2 delivering additional efficiency above and beyond that delivered by KG_1. Measured by computation time, including time simulating and time doing algorithmic computation, KG_1^2 required 14 minutes to achieve this average performance of 115.53, whereas ISC required 27 minutes.

Although the KG algorithms outperformed ISC in terms of the number of samples required to reach a given level of solution quality, algorithms like KG_β and KG_β^2 that rely on Kriging or Gaussian-process regression may consume substantial computational resources in deciding where to sample, which may make them less suitable for problems

Figure 3. Performance of the accelerated KG_1^2 and accelerated KG_1 allocation rules, and Industrial Strength COMPASS for the assemble-to-order problem.



in which simulation can be performed very quickly. When simulation samples come from a complex, long-running simulator, this is relatively unimportant, and algorithms like KG^2_β that find good solutions in few samples also work well in terms of overall computation time. Appendix EC.2, in the online appendix, discusses further these computational issues.

In summary, our algorithms demonstrate superior efficiency compared to others in problems with large solution spaces and when samples are moderately to very computationally expensive.

9. Conclusions

We contributed to the area of discrete optimization via simulation, where the value of the best alternative is to be estimated by simulation, by developing a fully sequential algorithm based on new VOI tools. Those tools are able to take advantage of both correlated prior beliefs and correlated sampling distributions. We gave easy-to-verify conditions under which almost sure convergence to the optimal solution can be guaranteed. The implementation presented here takes advantage of machine learning tools that enable exploring combinatorially large solution spaces, with run times that are a low order polynomial in the number of samples observed (which is better than a low order polynomial in the size of the solution space when only a fraction of alternatives are visited).

We also derived "accelerated" versions of the algorithms that use local search when alternatives can be embedded in a continuous space. That acceleration takes advantage of gradient information about the Bayesian value of information, rather than the more common technique of using gradient information about the response surface, to improve practical performance. Numerical results show that there is a distinct benefit for being able to use both correlated prior beliefs and correlated sampling in simulation optimization using the Bayesian VOI framework.

Future work may include the derivation of the MLEs with a richer set of kernels for the GP prior and/or sampling covariance matrix to embody a broader set of correlations, a more detailed assessment of how to handle simulation runs that do not have the same length or sampling variance, the role of other stopping rules, how the local search function $\tilde{f}$ might be improved in various contexts (e.g., discrete neighbor search as compared to gradient search, or with feasible solutions in narrow corridors or isolated parts of the solution space), and the handling of output distributions that do not satisfy the normality assumption.

Supplemental Material

Supplemental material to this paper is available at http://dx.doi.org/10.1287/opre.2016.1480.

Acknowledgments

The authors thank Jeff Hong and Barry Nelson for discussions about the assemble to order model and ISC. The second author was supported by AFOSR [FA9550-11-1-0083, FA9550-12-1-0200, and FA9550-15-1-0038], National Science Foundation [IIS-1247696, IIS-142251, and CMMI-1254298], and by the Atkinson Center for a Sustainable Future Academic Venture Fund. The first author was supported by AFOSR [FA9550-11-1-0083].

Appendix A. Mathematical Proofs

Proof of Lemma 1. From the definition of the function $h(\cdot,\cdot)$ (just after (15)), for any vectors a, b, and b' with $b(i) \leqslant b'(i)$ for all i, we have $h(a,b) \leqslant h(a,b')$. From (15), we have $V_n(\vec{x},A,\beta) = h(\mu_n(A), \tilde{\sigma}_n(\vec{x},A,\beta))$, where for all $x' \in A$, the element of $\tilde{\sigma}_n(\vec{x},A,\beta)$ corresponding to x' is $[\Sigma_n(x',x^{(1)}) - \Sigma_n(x',x^{(2)})]/B$, where B is the denominator (in the lower equation for pairs) in (12). Hence, we only need to show that B is a decreasing function of $\rho(x^{(1)},x^{(2)})$. The result follows immediately by observing $\Lambda(x^{(1)},x^{(2)}) = \rho(x^{(1)},x^{(2)})[\Lambda(x^{(1)},x^{(1)})\Lambda(x^{(2)},x^{(2)})]^{1/2}$. □

We now state and prove several lemmas needed to prove the convergence results stated in §6. These lemmas all assume Assumptions 1 and 2. Condition 1 is assumed only in the proof of Theorem 1.

Lemma 2. *There exist random variables $\mu_\infty \in \mathbb{R}^k$ and $\Sigma_\infty \in \Sigma^k_+$ (the space of $k \times k$ positive semidefinite matrices), such that μ_n converges to μ_∞, and Σ_n converges to Σ_∞ almost surely.*

Proof of Lemma 2. Let (μ_n,Σ_n) and $M_n = (\mu_n, \Sigma_n + \mu_n\mu_n^T)$. We can write the components of M_n as the conditional expectation of an integrable random variable with respect to $\mathscr{X}_n$, $\mathscr{Y}_n$ by $\mu_n = \mathbb{E}_n\,\theta$, $\Sigma_n + \mu_n\mu_n^T = \mathbb{E}_n\,\theta\theta^T$. This implies that M_n is a uniformly integrable martingale and hence converges almost surely (Doob's second martingale convergence theorem, e.g., see Øksendal 2003, Appendix C). Because (μ_n,Σ_n) is a continuous transformation of M_n, it also converges almost surely to some random variable $(\mu_\infty,\Sigma_\infty)$. □

Lemma 3. $\Sigma_n(x',x) = \Sigma_0(x',x) - \Sigma_0(x,\mathscr{X}_n)[S_n]^{-1}\Sigma_0(\mathscr{X}_n,x')$.

Proof of Lemma 3. Let $i_{x'}$ be the index of x' in $\mathscr{X}_{n,x}$. (If x' appears more than once, let it be the index of one occurrence.) Let $e_{x'}$ be a column vector with length $|\mathscr{X}_n|+1$ that has value 1 at entry $i_{x'}$ and 0 elsewhere. Let e_x be defined similarly. Using (6), (8), and the symmetry of $[S_n]^{-1}$, we then have

$$\begin{aligned}
\Sigma_n(x',x) &= e_{x'}^T\Sigma_n(\mathscr{X}_{n,x},\mathscr{X}_{n,x})e_x \\
&= e_{x'}^T(I_{|\mathscr{X}_n|+1} - K_n(x)[I_{|\mathscr{X}_n|},\vec{0}])\Sigma_0(\mathscr{X}_{n,x},\mathscr{X}_{n,x})e_x \\
&= e_{x'}^T\Sigma_0(\mathscr{X}_{n,x},\mathscr{X}_{n,x})e_x \\
&\quad - e_{x'}^TK_n(x)[I_{|\mathscr{X}_n|},\vec{0}]\Sigma_0(\mathscr{X}_{n,x},\mathscr{X}_{n,x})e_x \\
&= \Sigma_0(x',x) - e_{x'}^T\Sigma_0(\mathscr{X}_{n,x},\mathscr{X}_{n,x})[I_{|\mathscr{X}_n|},\vec{0}]^T[S_n]^{-1} \\
&\quad \cdot [I_{|\mathscr{X}_n|},\vec{0}]\Sigma_0(\mathscr{X}_{n,x},\mathscr{X}_{n,x})e_x \\
&= \Sigma_0(x',x) - \Sigma_0(x',\mathscr{X}_n)[S_n]^{-1}\Sigma_0(\mathscr{X}_n,x) \\
&= \Sigma_0(x',x) - \Sigma_0(x,\mathscr{X}_n)[S_n]^{-1}\Sigma_0(\mathscr{X}_n,x'). \quad \square
\end{aligned}$$

Lemma 4. $\Sigma_{n+1}(x,x) \leqslant \Sigma_n(x,x)$ *for all* x.

Proof of Lemma 4. Using standard results from Bayesian linear regression (e.g., Gelman et al. 2004, §14.6) and the Sherman-Woodbury formula (e.g., Rasmussen and Williams 2006,

Appendix A.3), the posterior variance Σ_{n+1} of θ can be computed recursively by

$$\Sigma_{n+1} = \Sigma_n - \Sigma_n X_{n+1}[X_{n+1}^T(\Lambda+\Sigma_n)X_{n+1}]^{-1}X_{n+1}^T\Sigma_n,$$

where

$$X_{n+1} = \begin{cases} e_x, & \text{if } \vec{x}_{n+1} = x, \\ [e_{x^{(1)}}, e_{x^{(2)}}], & \text{if } \vec{x}_{n+1} = (x^{(1)}, x^{(2)}), \end{cases}$$

and e_x is a $k \times 1$ vector with a value of 1 at the entry for x and 0 elsewhere.

It is clear that $\Lambda + \Sigma_n$ and $X_{n+1}^T(\Lambda+\Sigma_n)X_{n+1}$ are positive definite. Hence, for any x,

$$\Sigma_{n+1}(x,x) = \Sigma_n(x,x) - e_x^T\Sigma_n X_{n+1}[X_{n+1}^T(\Lambda+\Sigma_n)X_{n+1}]^{-1} \cdot X_{n+1}^T\Sigma_n e_x \leqslant \Sigma_n(x,x). \quad \square$$

Lemma 5. *For all x and $x^{(1)} \neq x^{(2)}$, $P(x^{(1)}, x^{(2)}) = \Lambda(x^{(1)}, x^{(1)}) + \Lambda(x^{(2)}, x^{(2)}) - 2\Lambda(x^{(1)}, x^{(2)}) > 0$ and*

$$\nu_n^{\mathrm{KG}_\beta}(x) \leqslant \frac{1}{c(x)}\sqrt{\frac{2\max_{x'}\Sigma_0(x',x')\Sigma_n(x,x)}{\beta\pi\Lambda(x,x)}},$$

$$\nu_n^{\mathrm{KG}_\beta}(x^{(1)}, x^{(2)}) \leqslant \frac{1}{c(x^{(1)}, x^{(2)})}\sqrt{\frac{2\max_{x'}\Sigma_0(x',x')}{\beta\pi P(x^{(1)}, x^{(2)})}} \cdot \left[\sqrt{\Sigma_n(x^{(1)}, x^{(1)})} + \sqrt{\Sigma_n(x^{(2)}, x^{(2)})}\right].$$

Proof of Lemma 5. First, we have

$$\begin{aligned}
&V_n(\vec{x}, A_n(\vec{x}), \beta) \\
&= \mathbb{E}_n[\max\mu_{n+1}(A_n(\vec{x})) \mid \vec{x}_{n+1} = \vec{x}, \beta_{n+1} = \beta] - \max\mu_n(A_n(\vec{x})) \\
&= \mathbb{E}[\max\{\mu_n(A_n(\vec{x})) + \tilde{\sigma}_n(\vec{x}, A_n(\vec{x}), \beta)Z\}] - \max\mu_n(A_n(\vec{x})) \\
&\leqslant \max\mu_n(A_n(\vec{x})) + \mathbb{E}[\max\{\tilde{\sigma}_n(\vec{x}, A_n(\vec{x}), \beta)Z\}] - \max\mu_n(A_n(\vec{x})) \\
&= \mathbb{E}[\max\{\tilde{\sigma}_n(\vec{x}, A_n(\vec{x}), \beta)Z\}] \\
&\leqslant \mathbb{E}[\max\{|\tilde{\sigma}_n(\vec{x}, A_n(\vec{x}), \beta)| \cdot |Z|\}] \\
&= \mathbb{E}|Z| \cdot \max\{|\tilde{\sigma}_n(\vec{x}, A_n(\vec{x}), \beta)|\} \\
&= \sqrt{2/\pi} \cdot \max\{|\tilde{\sigma}_n(\vec{x}, A_n(\vec{x}), \beta)|\} \\
&= \sqrt{2/\pi} \cdot \max_{j=1,2,\ldots,|A_n(\vec{x})|} |e_j^T\tilde{\sigma}_n(\vec{x}, A_n(\vec{x}), \beta)|,
\end{aligned}$$

where e_j is a $|A_n(\vec{x})| \times 1$ vector with 1 at entry j and 0 elsewhere.

We now derive an upper bound on $e_j^T\tilde{\sigma}_n(\vec{x}, A_n(\vec{x}), \beta)$. First, $P(x^{(1)}, x^{(2)}) = [e_{x^{(1)}} - e_{x^{(2)}}]^T\Lambda[e_{x^{(1)}} - e_{x^{(2)}}] > 0$ because Λ is positive definite by Assumption 2, where $e_{x^{(j)}}$ is a vector with 1 at entry $x^{(j)}$ and 0 elsewhere. Similarly, we have $\Sigma_n(x^{(1)}, x^{(1)}) + \Sigma_n(x^{(2)}, x^{(2)}) - 2\Sigma_n(x^{(1)}, x^{(2)}) \geqslant 0$ because Σ_n is positive semidefinite. Now applying (12) and Lemma 4, for $\vec{x} = x$ we have

$$\begin{aligned}
|e_j^T\tilde{\sigma}_n(x, A_n(x), \beta)| &= \frac{|e_j^T\Sigma_n(A_n(x), x)|}{\sqrt{\beta^{-1}\Lambda(x,x) + \Sigma_n(x,x)}} \\
&= \frac{|\Sigma_n(A_n^{(j)}(x), x)|}{\sqrt{\beta^{-1}\Lambda(x,x) + \Sigma_n(x,x)}} \\
&\leqslant \sqrt{\frac{\Sigma_n(A_n^{(j)}(x), A_n^{(j)}(x))\Sigma_n(x,x)}{\beta^{-1}\Lambda(x,x)}} \\
&\leqslant \sqrt{\frac{\Sigma_0(A_n^{(j)}(x), A_n^{(j)}(x))\Sigma_n(x,x)}{\beta^{-1}\Lambda(x,x)}},
\end{aligned}$$

where $A_n^{(j)}(\vec{x})$ is the jth component of $A_n(\vec{x})$. Similarly, for $\vec{x} = (x^{(1)}, x^{(2)})$ we have

$$\begin{aligned}
&|e_j^T\tilde{\sigma}_n(\vec{x}, A_n(\vec{x}), \beta)| \\
&= \left(|e_j^T\Sigma_n(A_n(\vec{x}), x^{(1)}) - e_j^T\Sigma_n(A_n(\vec{x}), x^{(2)})|\right) \\
&\quad \cdot \left((\beta^{-1}P(x^{(1)}, x^{(2)}) + \Sigma_n(x^{(1)}, x^{(1)}) + \Sigma_n(x^{(2)}, x^{(2)}) - 2\Sigma_n(x^{(1)}, x^{(2)}))^{1/2}\right)^{-1} \\
&\leqslant \frac{|\Sigma_n(A_n^{(j)}(\vec{x}), x^{(1)})| + |\Sigma_n(A_n^{(j)}(\vec{x}), x^{(2)})|}{\sqrt{\beta^{-1}P(x^{(1)}, x^{(2)})}} \\
&\leqslant \sqrt{\frac{\Sigma_n(A_n^{(j)}(\vec{x}), A_n^{(j)}(\vec{x}))}{\beta^{-1}P(x^{(1)}, x^{(2)})}}\left[\sqrt{\Sigma_n(x^{(1)}, x^{(1)})} + \sqrt{\Sigma_n(x^{(2)}, x^{(2)})}\right] \\
&\leqslant \sqrt{\frac{\Sigma_0(A_n^{(j)}(\vec{x}), A_n^{(j)}(\vec{x}))}{\beta^{-1}P(x^{(1)}, x^{(2)})}}\left[\sqrt{\Sigma_n(x^{(1)}, x^{(1)})} + \sqrt{\Sigma_n(x^{(2)}, x^{(2)})}\right].
\end{aligned}$$

The claimed bounds in the lemma for $\nu_n^{\mathrm{KG}_\beta}(x)$ and for $\nu_n^{\mathrm{KG}_\beta}(x^{(1)}, x^{(2)})$ follow directly. $\square$

Lemma 6. *Under the allocation rule $x_1 = x_2 = \cdots = x_n = x$, $\Sigma_n(x,x)$ decreases to 0 as $n \to +\infty$. Under the allocation rule $\vec{x}_1 = \vec{x}_2 = \cdots = \vec{x}_n = (x^{(1)}, x^{(2)})$, $\Sigma_n(x^{(1)}, x^{(1)})$ and $\Sigma_n(x^{(2)}, x^{(2)})$ decrease to 0 as $n \to +\infty$.*

Proof of Lemma 6. Lemma 4 shows that $\Sigma_n(x,x)$ is a decreasing sequence bounded below by zero, for all x. It suffices to show that the limit is 0 under these two cases.

First, consider the case when $x_1 = x_2 = \cdots = x_n = x_0$. Note $\Sigma_0(\mathscr{X}_n, \mathscr{X}_n) = \Sigma_0(x_0, x_0)ee^T$, where e is an $n \times 1$ vector with n entries of 1. By Lemma 3 and the Sherman-Woodbury formula, for any x and x',

$$\begin{aligned}
\Sigma_n(x,x') &= \Sigma_0(x,x') - \Sigma_0(x,\mathscr{X}_n)[S_n]^{-1}\Sigma_0(\mathscr{X}_n, x') \\
&= \Sigma_0(x,x') - \Sigma_0(x,x_0)\Sigma_0(x',x_0)e^T \cdot [\Sigma_0(x_0,x_0)ee^T + \Lambda(x_0,x_0)I_n]^{-1}e \\
&= \Sigma_0(x,x') - \frac{\Sigma_0(x,x_0)\Sigma_0(x',x_0)}{\Lambda(x_0,x_0)}e^T \cdot \left[I_n - \frac{\Sigma_0(x_0,x_0)}{n\Sigma_0(x_0,x_0) + \Lambda(x_0,x_0)}ee^T\right]e \\
&= \Sigma_0(x,x') - \frac{n\Sigma_0(x,x_0)\Sigma_0(x',x_0)}{n\Sigma_0(x_0,x_0) + \Lambda(x_0,x_0)}.
\end{aligned}$$

Specifically,

$$\Sigma_n(x_0,x_0) = \Sigma_0(x_0,x_0)\left[1 - \frac{n\Sigma_0(x_0,x_0)}{n\Sigma_0(x_0,x_0) + \Lambda(x_0,x_0)}\right] \to 0, \quad \text{as } n \to +\infty.$$

Next, consider the case when $\vec{x}_1 = \vec{x}_2 = \cdots = \vec{x}_n = (x_0^{(1)}, x_0^{(2)})$. Let

$$D_1 = \begin{pmatrix} \Sigma_0(x_0^{(1)}, x_0^{(1)}) & \Sigma_0(x_0^{(1)}, x_0^{(2)}) \\ \Sigma_0(x_0^{(1)}, x_0^{(2)}) & \Sigma_0(x_0^{(2)}, x_0^{(2)}) \end{pmatrix},$$

$$D_2 = \begin{pmatrix} \Lambda(x_0^{(1)}, x_0^{(1)}) & \Lambda(x_0^{(1)}, x_0^{(2)}) \\ \Lambda(x_0^{(1)}, x_0^{(2)}) & \Lambda(x_0^{(2)}, x_0^{(2)}) \end{pmatrix}.$$

Let $U=[I_2, I_2, \ldots, I_2]^T$ be a $2n\times 2$ matrix with n I_2-blocks. Let $u=[\Sigma_0(x,x_0^{(1)}), \Sigma_0(x,x_0^{(2)})]^T$ and $v=[\Sigma_0(x',x_0^{(1)}), \Sigma_0(x',x_0^{(2)})]^T$ be two 2×1 vectors. Then, $\Sigma_0(\mathscr{X}_n,\mathscr{X}_n)=UD_1U^T$, and Γ_n is a block diagonal matrix with n blocks, with each block equal to D_2. Similar to the above argument, we have

$$\begin{aligned}
&\Sigma_n(x,x')\\
&=\Sigma_0(x,x')-\Sigma_0(x,\mathscr{X}_n)[S_n]^{-1}\Sigma_0(\mathscr{X}_n,x')\\
&=\Sigma_0(x,x')-\Sigma_0(x,\mathscr{X}_n)[UD_1U^T+\Gamma_n]^{-1}\Sigma_0(\mathscr{X}_n,x')\\
&=\Sigma_0(x,x')-\Sigma_0(x,\mathscr{X}_n)\\
&\quad\cdot\left[\Gamma_n^{-1}-\Gamma_n^{-1}U(D_1^{-1}+U^T\Gamma_n^{-1}U)^{-1}U^T\Gamma_n^{-1}\right]\Sigma_0(\mathscr{X}_n,x')\\
&=\Sigma_0(x,x')-n[u^TD_2^{-1}v-nu^TD_2^{-1}\\
&\quad\cdot[D_1^{-1}+nD_2^{-1}]^{-1}D_2^{-1}v]\\
&=\Sigma_0(x,x')-nu^T(nD_1+D_2)^{-1}v,
\end{aligned}$$

where the last line follows from the previous line by the following computation, which uses the matrix identity $A^{-1}B^{-1}=(BA)^{-1}$ and the Sherman-Woodbury formula:

$$\begin{aligned}
&[I-nD_2^{-1}(D_1^{-1}+nD_2^{-1})^{-1}]D_2^{-1}\\
&\quad=[I-n(D_1^{-1}D_2+nI)^{-1}]D_2^{-1}\\
&\quad=\left[I-\left(I+\frac{D_1^{-1}D_2}{n}\right)^{-1}\right]D_2^{-1}\\
&\quad=[I-(I-D_1^{-1}(nI+D_2D_1^{-1})^{-1}D_2]D_2^{-1}\\
&\quad=D_1^{-1}(nI+D_2D_1^{-1})^{-1}=(nD_1+D_2)^{-1}.
\end{aligned}$$

Simple algebra then yields $\lim_{n\to+\infty} nu^T(nD_1+D_2)^{-1}v=(d_2+d_3-d_4)/d_1$, where

$$\begin{aligned}
d_1&=\Sigma_0(x_0^{(1)},x_0^{(1)})\Sigma_0(x_0^{(2)},x_0^{(2)})-[\Sigma_0(x_0^{(1)},x_0^{(2)})]^2,\\
d_2&=\Sigma_0(x_0^{(1)},x_0^{(1)})\Sigma_0(x,x_0^{(2)})\Sigma_0(x',x_0^{(2)}),\\
d_3&=\Sigma_0(x_0^{(2)},x_0^{(2)})\Sigma_0(x,x_0^{(1)})\Sigma_0(x',x_0^{(1)}),\\
d_4&=\Sigma_0(x_0^{(1)},x_0^{(2)})\left[\Sigma_0(x,x_0^{(1)})\Sigma_0(x',x_0^{(2)})+\Sigma_0(x,x_0^{(2)})\Sigma_0(x',x_0^{(1)})\right].
\end{aligned}$$

Under Assumption 2, we always have $d_1>0$ because Σ_0 is positive definite. Specifically, when $x=x'=x_0^{(i)}$, $(i=1,2)$, $[d_2+d_3-d_4]/d_1=\Sigma_0(x_0^{(i)},x_0^{(i)})$. Hence, $\Sigma_n(x_0^{(i)},x_0^{(i)})\to 0$ as $n\to+\infty$ for $i=1,2$. □

Lemma 7. *If alternative x is sampled infinitely often, then $\Sigma_n(x,x)\to 0$ and $\nu_n^{\mathrm{KG}_\beta}(x)\to 0$ as $n\to\infty$. If alternative $x'\neq x$ is also sampled infinitely often, then $\Sigma_n(x',x')\to 0$ and $\nu_n^{\mathrm{KG}_\beta}(x,x')\to 0$ as $n\to\infty$.*

Proof of Lemma 7. There are k possible decisions in Ξ that involve sampling alternative x, namely, x and (x,x') for $x'\neq x$. Because x is sampled infinitely many times, at least one of these k decisions is chosen infinitely often. Let $\vec{x}$ be one such decision and $\{q_n\}_{n=1}^{\infty}$ be a strictly increasing subsequence of $\mathbb{Z}^+$ such that $\vec{x}_{q_n}=\vec{x}$ for $n=1,2,\ldots$. Because the ordering of the decision-observation pairs can be changed without altering $\Sigma_n(x,x)$, and because taking additional observations can only decrease $\Sigma_n(x,x)$ by Lemma 4, we know that an upper bound on $\Sigma_{q_n}(x,x)$ is given by the posterior variance of θ_x at iteration n under an allocation rule, call it π, that chooses $x_1=x_2=\cdots=x_n=\vec{x}$. Call this posterior variance $\Sigma_n^{\pi}(x,x)$, so we have $\Sigma_{q_n}(x,x)\leqslant\Sigma_n^{\pi}(x,x)$. Lemma 6 shows $\lim_{n\to\infty}\Sigma_n^{\pi}(x,x)=0$. Hence, $\lim_{n\to\infty}\Sigma_{q_n}(x,x)=0$. Because $\{\Sigma_n(x,x)\}_n$ is a nonnegative decreasing sequence, $\lim_{n\to\infty}\Sigma_n(x,x)$ exists and equals 0, due to the uniqueness of the limit. Combining this with Lemma 5 and the nonnegativity of the KG_β factors, we have $\lim_{n\to\infty}\nu_n^{\mathrm{KG}_\beta}(x)=0$.

If $x'\neq x$ is also sampled infinitely often, similarly we have

$$\lim_{n\to\infty}\Sigma_n(x',x')=0.$$

Thus $\lim_{n\to\infty}\nu_n^{\mathrm{KG}_\beta}(x,x')=0$ by Lemma 5. □

Lemma 8. *If $\liminf_{n\to\infty}\nu_n^{\mathrm{KG}_\beta}(\vec{x})=0$ for all $\vec{x}\in\Xi$, then $\lim_{n\to\infty}\Sigma_n(x,x)=0$ for all x.*

Proof of Lemma 8. Consider an arbitrary sample path on which μ_n converges to μ_∞. Lemma 2 shows that the set of such sample paths is almost sure. We will show that the claim holds on this sample path.

Lemma 4 shows that $\{\Sigma_n(x,x)\}_n$ is a nonnegative decreasing sequence and hence $\lim_{n\to\infty}\Sigma_n(x,x)$ exists and is nonnegative for any x. We prove the contrapositive of the statement of the lemma. That is, we suppose that $\max_x[\lim_{n\to\infty}\Sigma_n(x,x)]>0$ and show that $\liminf_{n\to\infty}\nu_n^{\mathrm{KG}_\beta}(\vec{x})>0$ for some $\vec{x}\in\Xi$.

We choose two alternatives on which to focus our analysis. First, because at least one decision $\vec{x}'\in\Xi$ is chosen by the algorithm infinitely often, Lemma 7 shows that there exists an alternative x' with $\lim_{n\to\infty}\Sigma_n(x',x')=0$. Second, by our choice of sample path, $\lim_{n\to\infty}\mu_n=\mu_\infty$. Let $x^*=\arg\max\mu_\infty$, breaking ties arbitrarily. Then, $\mu_\infty(x^*)\geqslant\mu_\infty(x)$ for all x. It follows that there exists N large enough and a sequence $\{\epsilon_n\}$ decreasing to 0 such that $\mu_n(x^*)\geqslant\mu_n(x)-\epsilon_n$ for all x for $n\geqslant N$. If $\lim_{n\to\infty}\Sigma_n(x^*,x^*)>0$, let $x^{(1)}=x^*$ and $x^{(2)}=x'$; otherwise pick $x^{(1)}$ with $\lim_{n\to\infty}\Sigma_n(x^{(1)},x^{(1)})>0$ and let $x^{(2)}=x^*$. Let $\vec{x}=(x^{(1)},x^{(2)})$.

For each $n\geqslant N$, let

$$\begin{aligned}
a_n^1&=\mu_n(x^{(1)}),\quad a_n^2=\mu_n(x^{(2)}),\\
b_n^1&=\frac{\Sigma_n(x^{(1)},x^{(1)})-\Sigma_n(x^{(1)},x^{(2)})}{\sqrt{\beta^{-1}P+Q_n}},\\
b_n^2&=\frac{\Sigma_n(x^{(2)},x^{(1)})-\Sigma_n(x^{(2)},x^{(2)})}{\sqrt{\beta^{-1}P+Q_n}},
\end{aligned}$$

where P and Q_n are given in (13). Then, we have the following:

$$\begin{aligned}
&V_n(\vec{x},A_n(\vec{x}),\beta)\\
&=\mathbb{E}_n\left[\max\mu_{n+1}(A_n(\vec{x}))\mid\vec{x}_{n+1}=\vec{x},\beta_{n+1}=\beta\right]-\max\mu_n(A_n(\vec{x}))\\
&\geqslant\mathbb{E}_n\left[\max\{\mu_{n+1}(x^{(1)}),\mu_{n+1}(x^{(2)})\}\mid\vec{x}_{n+1}=\vec{x},\beta_{n+1}=\beta\right]\\
&\quad-\max\{\mu_n(x^{(1)}),\mu_n(x^{(2)})\}-\epsilon_n\\
&=\mathbb{E}\left[\max\{a_n^1+b_n^1Z,a_n^2+b_n^2Z\right]-\max\{a_n^1,a_n^2\}-\epsilon_n\\
&=|b_n^1-b_n^2|f\left(-\frac{|a_n^1-a_n^2|}{|b_n^1-b_n^2|}\right)-\epsilon_n,
\end{aligned}\tag{A1}$$

where $f(-s)=\varphi(s)-s\Phi(-s)$ is as defined in §3.1, and (A1) is understood to be 0 when $|b_n^1-b_n^2|=0$. In this sequence of expressions, the first line applies (14); the second line uses

$\max \mu_n(A_n(\vec{x})) \leqslant \mu_n(x^*) + \epsilon_n = \max\{\mu_n(x^{(1)}, \mu_n(x^{(2)})\} + \epsilon_n$ together with the fact that $A_n(\vec{x})$ contains $x^{(1)}$ and $x^{(2)}$; the third line uses (11) and (12); and the last line follows from computations involving the normal distribution, which may be found in Frazier et al. (2009, Equation (14)). We will take the limit of (A1) as n goes to ∞.

By our choice of sample path, μ_n converges to μ_∞, so

$$\lim_{n\to\infty} |a_n^1 - a_n^2| = |\mu_\infty(x^{(1)}) - \mu_\infty(x^{(2)})| := \gamma_1 < \infty.$$

We now show that $\lim_{n\to\infty} |b_n^1 - b_n^2|$ is strictly positive. First,

$$|b_n^1 - b_n^2| = \frac{|\Sigma_n(x^{(1)}, x^{(1)}) - 2\Sigma_n(x^{(1)}, x^{(2)}) + \Sigma_n(x^{(2)}, x^{(2)})|}{\sqrt{\beta^{-1}P + Q_n}} = \frac{|Q_n|}{\sqrt{\beta^{-1}P + Q_n}}.$$

Then, $\{\Sigma_n(x^{(1)}, x^{(1)})\}_n$ is bounded above by $\Sigma_0(x^{(1)}, x^{(1)})$ by Lemma 4, and $\lim_{n\to\infty} \Sigma_n(x^{(2)}, x^{(2)}) = 0$, so

$$\lim_{n\to\infty} |\Sigma_n(x^{(1)}, x^{(2)})| \leqslant \lim_{n\to\infty} \sqrt{\Sigma_n(x^{(1)}, x^{(1)})\Sigma_n(x^{(2)}, x^{(2)})} = 0.$$

Hence, $\lim_{n\to\infty} \Sigma_n(x^{(1)}, x^{(2)}) = 0$. It follows that

$$\lim_{n\to\infty} Q_n = \lim_{n\to\infty} \Sigma_n(x^{(1)}, x^{(1)}) - 2\Sigma_n(x^{(1)}, x^{(2)}) + \Sigma_n(x^{(2)}, x^{(2)}) = \lim_{n\to\infty} \Sigma_n(x^{(1)}, x^{(1)}),$$

which is strictly positive by the construction of $x^{(1)}$. Thus,

$$\lim_{n\to\infty} |b_n^1 - b_n^2| = \liminf_{n\to\infty} \frac{|Q_n|}{\sqrt{\beta^{-1}P + Q_n}} = \frac{|\lim_{n\to\infty} \Sigma_n(x^{(1)}, x^{(1)})|}{\sqrt{\beta^{-1}P + \lim_{n\to\infty} \Sigma_n(x^{(1)}, x^{(1)})}} := \gamma_2 > 0.$$

Recall (A1). The function $s \mapsto f(-s)$ is continuous, so (A1) is continuous in $(|a_n^1 - a_n^2|, |b_n^1 - b_n^2|)$ on $[0,\infty)\times(0,\infty)$, and the limit of (A1) as $n\to\infty$ is $\gamma_2 f(-\gamma_1/\gamma_2)$. Because $V_n(\vec{x}, A_n(\vec{x}), \beta)$ is bounded below by (A1),

$$\liminf_{n\to\infty} \nu_n^{\mathrm{KG}_\beta}(\vec{x}) = \frac{\liminf_{n\to\infty} V_n(\vec{x}, A_n(\vec{x}), \beta)}{\beta c(\vec{x})} \geqslant \frac{1}{\beta c(\vec{x})}\gamma_2 f\left(-\frac{\gamma_1}{\gamma_2}\right) > 0,$$

where we have used that $\gamma_1 < \infty$ and $\gamma_2 > 0$, and $f(-s)$ is strictly positive for $s < \infty$. □

Proof of Theorem 1. We first show, by contradiction, that $\liminf_{n\to\infty} \nu_n^{\mathrm{KG}_\beta}(\vec{x}) = 0$ almost surely for all $\vec{x} \in \Xi$. Consider an arbitrary sample path of the KG_β^2 algorithm from the almost sure set on which the claim of Lemma 8 holds. Let

$$\chi_0 := \{\vec{x} \in \Xi\colon \lim_{n\to\infty} \nu_n^{\mathrm{KG}_\beta}(\vec{x}) \text{ exists and is } 0\} \quad \text{and}$$

$$\chi_1 := \{\vec{x} \in \Xi\colon \liminf_{n\to\infty} \nu_n^{\mathrm{KG}_\beta}(\vec{x}) = 0\}.$$

Suppose for contradiction that $\chi_1 \neq \Xi$, i.e., that $\Xi\backslash\chi_1 = \{\vec{x} \in \Xi\colon \liminf_{n\to\infty} \nu_n^{\mathrm{KG}_\beta}(\vec{x}) > 0\}$ is not empty.

Pick $\vec{x} \in \Xi\backslash\chi_1$. By Condition 1, there exists a subsequence of $\mathbb{Z}^+$, denoted by $\{n_i\}_{i=1}^\infty$, such that $\vec{x}_{n_i} \in \Xi_{n_i}$ for all i. Also, $\liminf_{i\to\infty} \nu_{n_i}^{\mathrm{KG}_\beta}(\vec{x}) \geqslant \liminf_{n\to\infty} \nu_n^{\mathrm{KG}_\beta}(\vec{x}) > 0$. Thus, there exists some $\epsilon > 0$ and a subsequence of $\{n_i\}_{i=1}^\infty$, denoted $\{n_j\}_{j=1}^\infty$, such that $\nu_{n_j}^{\mathrm{KG}_\beta}(\vec{x}) \geqslant \epsilon$ for all j. Then, $\nu_{n_j}^{\mathrm{KG}_\beta}(\vec{x}_{n_j}) \geqslant \nu_{n_j}^{\mathrm{KG}_\beta}(\vec{x}) \geqslant \epsilon$ for all j.

For each $\vec{x}' \in \Xi\backslash\chi_0$, the contrapositive of Lemma 7 implies there exists a finite number $N(\vec{x}')$ such that the KG_β^2 algorithm does not choose $\vec{x}'$ for $n > N(\vec{x}')$. Let $N := \max_{\vec{x}'\in\Xi\backslash\chi_0} N(\vec{x}')$. Then, $\vec{x}_n \in \chi_0$ for all $n > N$.

For each $\vec{x}' \in \chi_0$, $\lim_{n\to\infty} \nu_n^{\mathrm{KG}_\beta}(\vec{x}') = 0$. Hence, there exists a finite number $N_0(\vec{x}')$ such that $\nu_n^{\mathrm{KG}_\beta}(\vec{x}') < \epsilon$ for all $n > N_0(\vec{x}')$. Let $N_0 := \max_{\vec{x}'\in\chi_0} N_0(\vec{x}')$. Then, for all $n > N_0$, $\nu_n^{\mathrm{KG}_\beta}(\vec{x}') < \epsilon$ for any $\vec{x}' \in \chi_0$.

It follows that $\nu_n^{\mathrm{KG}_\beta}(\vec{x}_n) < \epsilon$ for all $n > \max\{N_0, N\}$, which contradicts $\nu_{n_j}^{\mathrm{KG}_\beta}(\vec{x}_{n_j}) \geqslant \epsilon$ for all j. We thus conclude that $\chi_1 = \Xi$ on this sample path, i.e., $\liminf_{n\to\infty} \nu_n^{\mathrm{KG}_\beta}(\vec{x}) = 0$ for all $\vec{x} \in \Xi$. Since the chosen sample path was arbitrary, this holds almost surely.

Since we chose a sample path on which Lemma 8 holds, $\lim_{n\to\infty} \Sigma_n(x,x) = 0$ on this sample path. Moreover, as the set of sample paths on which Lemma 8 holds is almost sure, $\lim_{n\to\infty} \Sigma_n(x,x) = 0$ almost surely.

To show that $\lim_n \mu_n(x) = \theta(x)$ almost surely for each x, we first show this limit holds in L^2. For each x, $E[(\mu_n(x) - \theta(x))^2] = E[E_n[(\mu_n(x) - \theta(x))^2]] = E[\Sigma_n(x,x)]$. Taking the limit as $n\to\infty$ and using $0 \leqslant \Sigma_n(x,x) \leqslant \Sigma_0(x,x)$ with the dominated convergence theorem implies $\lim_{n\to\infty} E[(\mu_n(x) - \theta(x))^2] = E[\lim_{n\to\infty} \Sigma_n(x,x)] = 0$. Then, since $\mu_n(x)$ converges to $\theta(x)$ in L^2, and Lemma 2 implies $\lim_{n\to\infty} \mu_n(x)$ exists almost surely, this almost sure limit equals $\theta(x)$.

We now show that $\lim_{n\to\infty} \arg\max_x \mu_n(x) = \arg\max_x \theta(x)$ almost surely. First, $x^* \in \arg\max_x \theta(x)$ is almost surely unique as a realization of a multivariate normal random variable, and so $\epsilon = \theta(x^*) - \max_{x\neq x^*} \theta(x)$ is almost surely strictly positive. Fix a sample path on which $\lim_{n\to\infty} \mu_n(x) = \theta(x)$ for each x (which occurs almost surely). There exists $N < \infty$ such that $|\mu_n - \theta(x)| < \epsilon/2$ for all $n > N$. Then, for all $n > N$ and all $x \neq x^*$, $\mu_n(x^*) > \theta(x^*) - \epsilon/2 > \theta(x) + \epsilon/2 > \mu_n(x)$, implying x^* is the unique element in $\arg\max_x \mu_n(x)$. This shows that $\lim_{n\to\infty} \arg\max_x \mu_n(x) = \arg\max_x \theta(x)$ almost surely. □

Appendix B. Gradients Results

Appendix B.1 derives the gradient of the VOI and KG factors with respect to sampling decisions, under the assumption that singletons are embedded in $\mathbb{R}^d$ and pairs are embedded in $\mathbb{R}^{2d}$. Those derivations are supported by gradient results for the posterior means and predictive covariances in B.2 and B.3, respectively.

The expressions provided in Appendices B.2–B.3 are for general priors and sampling models, and have within them terms such as $\nabla_x[\mu_0(x)]$, $\nabla_x[\Sigma_0(x,x)]$, and $\nabla_x[\Lambda(x,x)]$, whose values depend on the specific prior and form of sampling correlation assumed. Appendix B.4 demonstrates simplifications of those expressions for the special case of a GP prior distribution with Gaussian kernel and constant mean, and a sampling covariance that satisfies a compound sphericity assumption (as in §7.1 and §7.2).

We abuse notation slightly by writing gradients in terms of derivatives with respect to the coordinates x_i rather than with respect to their embedding, $\zeta_i(x)$ in $\mathbb{R}$, in order to simplify notation.

B.1. Gradient of the VOI and the KG Factor

The gradient of the VOI is required to determine the gradient of the KG factor. Thus, we start by assessing the gradient

of $V_n(x, A_n(x), \beta)$ in $\mathbb{R}^d$ and of $V_n((x^{(1)}, x^{(2)}), A_n(x^{(1)}, x^{(2)}), \beta)$ in $\mathbb{R}^{2d}$, where $A_n(\vec{x})$ is as described in §5.

First, consider the singleton $\vec{x} = x$. Recall that $V_n(x, A_n(x), \beta) = sf(-\Delta/s)$, where $\Delta = |\mu_n(x) - \mu_n(x_*)|$, $s = |\tilde{\sigma}_n(x, x, \beta) - \tilde{\sigma}_n(x, x_*, \beta)|$, and $f(z) = \varphi(z) + z\Phi(z)$. Direct calculation reveals that

$$\begin{aligned}
&\nabla_x[V_n(x, A_n(x), \beta)] \\
&= \varphi(\Delta/s) \cdot \operatorname{sign}[\tilde{\sigma}_n(x, x, \beta) - \tilde{\sigma}_n(x, x_*, \beta)] \\
&\quad \cdot \nabla_x[\tilde{\sigma}_n(x, x, \beta) - \tilde{\sigma}_n(x, x_*, \beta)] \\
&\quad - \Phi(-\Delta/s) \cdot \operatorname{sign}[\mu_n(x) - \mu_n(x_*)] \nabla_x[\mu_n(x) - \mu_n(x_*)]. \qquad \text{(B1)}
\end{aligned}$$

Detailed derivations of $\nabla_x[\mu_n(x')]$ and $\nabla_x[\tilde{\sigma}_n(x, x', \beta)]$ for arbitrary x' are given in Appendix B.2. Second, consider the pair $\vec{x} = (x^{(1)}, x^{(2)})$. Let $a = \mu_n(A_n(\vec{x}))$ and $b = \tilde{\sigma}(\vec{x}, A_n(\vec{x}), \beta)$. We have from §3.1 that

$$V_n(\vec{x}, A_n(\vec{x}), \beta) = h(a, b).$$

To support taking the derivative of this quantity, we now recall Algorithms 1 and 2 from Frazier et al. (2009) for computing $h(a, b) = E[\max_i a(i) + b(i)Z] - \max_i a(i)$. We first reorder the components of a and b so that the $b(i)$ are in nondecreasing order and ties in b are broken so that $a(i) \leqslant a(i+1)$ if $b(i) = b(i+1)$. Then, we remove all those entries i for which $a(i) + b(i)z < \max_{j \neq i}\{a(j) + b(j)z\}$ for all values of z (this is accomplished by Frazier et al. 2009, Algorithm 1). This gives new vectors a' and b' with $|a'| = |b'| \leqslant |a| = |b|$. Set $\gamma(i) = (a'(i+1) - a'(i))/(b'(i+1) - b'(i))$ for $i = 1, 2, \ldots, |a'| - 1$. Then,

$$V_n(\vec{x}, A_n(\vec{x}), \beta) = h(a, b) = \sum_{i=1}^{|a'|-1} [b'(i+1) - b'(i)] f(-|\gamma(i)|)$$

if $|a'| > 1$, and the sum is taken to be 0 if $|a'| = 1$. Computation then reveals that

$$\begin{aligned}
&\nabla_{\vec{x}}[V_n(\vec{x}, A_n(\vec{x}), \beta)] \\
&= \sum_{i=1}^{|a'|-1} \varphi(\gamma(i)) \nabla_{\vec{x}}[b'(i+1) - b'(i)] \\
&\quad - \Phi(-|\gamma(i)|) \operatorname{sign}[a'(i+1) - a'(i)] \nabla_{\vec{x}}[a'(i+1) - a'(i)]. \qquad \text{(B2)}
\end{aligned}$$

For each i, $a'(i)$ and $b'(i)$ are equal to $a(j)$ and $b(j)$ for j given by the reordering procedure above, and $a(j)$ and $b(j)$ are the jth components of $a = \mu_n(A_n(\vec{x}))$ and $b = \tilde{\sigma}(\vec{x}, A_n(\vec{x}), \beta)$, respectively. Thus, $\nabla_{\vec{x}}[a'(i)]$ and $\nabla_{\vec{x}}[b'(i)]$ are equal to $\nabla_{\vec{x}}[\mu_n(x')]$ and $\nabla_{\vec{x}}[\tilde{\sigma}_n(\vec{x}, x', \beta)]$, where x' is the jth element in $A_n(\vec{x})$. Derivations of these quantities are given in Appendix B.3.

We now consider the gradient of the KG_β factors in $\mathbb{R}^d$. Recalling (16) and (17), we have

$$\begin{aligned}
&\nabla_{\vec{x}}[\nu_n^{\mathrm{KG}_\beta}(\vec{x})] \\
&= \frac{\nabla_{\vec{x}}[V_n(\vec{x}, A_n(\vec{x}), \beta)] \cdot c(\vec{x}) - V_n(\vec{x}, A_n(\vec{x}), \beta) \nabla_{\vec{x}}[c(\vec{x})]}{\beta[c(\vec{x})]^2} \qquad \text{(B3)}
\end{aligned}$$

for $\vec{x} = x$ or $(x^{(1)}, x^{(2)})$. In the case of homogeneous sampling costs for each alternative ($c(\vec{x}) = c|\vec{x}|$), we have $\nabla_{\vec{x}}[c(\vec{x})] = 0$. Hence, (B3) is determined by the preceding results as

$$\nabla_{\vec{x}}[\nu_n^{\mathrm{KG}_\beta}(\vec{x})] = \nabla_{\vec{x}}[V_n(\vec{x}, A_n(\vec{x}), \beta)]/(\beta c(\vec{x})).$$

B.2. Gradients of $\mu_n(x')$ and $\tilde{\sigma}_n(x, x', \beta)$ When Sampling a Singleton

In this section, we provide expressions for $\nabla_x[\mu_n(x')]$ and $\nabla_x[\tilde{\sigma}_n(x, x', \beta)]$ for an arbitrary alternative x'. These expressions can be substituted in (B1) to obtain an expression for the gradient of the VOI $V_n(x, A_n(x), \beta)$ when sampling a singleton $\vec{x} = x$, that holds when $A_n(x)$ is as described in §5.2.

To support this computation, let $J_n(x') := \nabla_{x'}[\Sigma_0(x', \mathscr{X}_n)]$ be a $d \times |\mathscr{X}_n|$ matrix, the ith column of which is $\nabla_{x'}[\Sigma_0(x', \mathscr{X}_n(i))]$, where $\mathscr{X}_n(i)$ is the ith entry of $\mathscr{X}_n$. Recall $\tilde{\mathscr{Y}}_n$, S_n and $K_n(\vec{x})$ from (6).

We first provide an expression for $\nabla_x[\mu_n(x')]$.

Lemma 9. *$\nabla_x[\mu_n(x')] = \nabla_x[\mu_0(x)] + J_n(x)[S_n]^{-1}\tilde{\mathscr{Y}}_n$ if $x = x'$, and is $\vec{0}$ if $x \neq x'$.*

Proof of Lemma 9. If $x \neq x'$, then $\mu_n(x')$ does not depend on x, so $\nabla_x[\mu_n(x')] = 0$. Now consider $x = x'$. Note that x is the last element of $\mathscr{X}_{n,x}$. Let e_x be the column vector $[0, 0, \ldots, 0, 1]^T$ with length $|\mathscr{X}_n| + 1$. Then

$$\begin{aligned}
\mu_n(x) &= e_x^T \mu_n(\mathscr{X}_{n,x}) \\
&= e_x^T[\mu_0(\mathscr{X}_{n,x}) + K_n(x)\tilde{\mathscr{Y}}_n] \\
&= e_x^T \mu_0(\mathscr{X}_{n,x}) + e_x^T \Sigma_0(\mathscr{X}_{n,x}, \mathscr{X}_{n,x})[I_{|\mathscr{X}_n|}, \vec{0}]^T [S_n]^{-1}\tilde{\mathscr{Y}}_n \\
&= \mu_0(x) + \Sigma_0(x, \mathscr{X}_n)[S_n]^{-1}\tilde{\mathscr{Y}}_n,
\end{aligned}$$

where we use (6) and (7) in the last line. Because $[S_n]^{-1}\tilde{\mathscr{Y}}_n$ does not depend on x, the gradient is $\nabla_x[\mu_n(x)] = \nabla_x(\mu_{0x}) + \nabla_x[\Sigma_0(x, \mathscr{X}_n)][S_n]^{-1}\tilde{\mathscr{Y}}_n = \nabla_x[\mu_0(x)] + J_n(x)[S_n]^{-1}\tilde{\mathscr{Y}}_n$. □

We now provide an expression for $\nabla_x[\tilde{\sigma}_n(x, x', \beta)]$.

Lemma 10.

$$\nabla_x[\tilde{\sigma}_n(x, x', \beta)] = \frac{B\nabla_x[\Sigma_n(x', x)] - \Sigma_n(x', x)\nabla_x(B)}{B^2}$$

where $B := \sqrt{\Lambda(x, x)/\beta + \Sigma_n(x, x)}$, and

$$\nabla_x[\Sigma_n(x', x)] = \begin{cases} \nabla_x[\Sigma_0(x', x)] - J_n(x)[S_n]^{-1}\Sigma_0(\mathscr{X}_n, x'), \\ \quad \text{if } x' \neq x, \\ \nabla_x[\Sigma_0(x, x)] - 2J_n(x)[S_n]^{-1}\Sigma_0(\mathscr{X}_n, x), \\ \quad \text{if } x' = x, \end{cases}$$

$$\nabla_x(B) = \frac{1}{2B}\left\{\nabla_x[\Lambda(x, x)/\beta + \Sigma_0(x, x)] - 2J_n(x)[S_n]^{-1}\Sigma_0(\mathscr{X}_n, x)\right\}.$$

Proof of Lemma 10. Recall that $\tilde{\sigma}_{nx'}(X, \beta) = \Sigma_n(x', x)/B$, so $\nabla_x[\tilde{\sigma}_n(x, x', \beta)]$ is as claimed. Next, recall from Lemma 3 that $\Sigma_n(x', x) = \Sigma_0(x', x) - \Sigma_0(x, \mathscr{X}_n)[S_n]^{-1}\Sigma_0(\mathscr{X}_n, x')$. Thus, if $x' \neq x$, then

$$\begin{aligned}
&\nabla_x[\Sigma_n(x', x)] \\
&\quad = \nabla_x[\Sigma_0(x', x)] - \nabla_x[\Sigma_0(x, \mathscr{X}_n)][S_n]^{-1}\Sigma_0(\mathscr{X}_n, x') \\
&\quad = \nabla_x[\Sigma_0(x', x)] - J_n(x)[S_n]^{-1}\Sigma_0(\mathscr{X}_n, x').
\end{aligned}$$

If $x' = x$, then using standard matrix differentiation, we can compute the gradient as

$$\nabla_x[\Sigma_n(x, x)] = \nabla_x[\Sigma_0(x, x)] - 2J_n(x)[S_n]^{-1}\Sigma_0(\mathscr{X}_n, x).$$

The claimed formula for $\nabla_x(B)$ follows from simple algebra. □

B.3. Gradients of $\mu_n(x')$ and $\tilde{\sigma}_n(\vec{x}, x', \beta)$ When Sampling a Pair

In this section, we describe computation of $\nabla_{\vec{x}}[\mu_n(x')]$ and $\nabla_{\vec{x}}[\tilde{\sigma}_n(\vec{x}, x', \beta)]$ for an arbitrary alternative x'. These expressions can be substituted in (B2) to obtain an expression for the gradient of the VOI $V_n(x, A_n(x), \beta)$ when sampling a pair $\vec{x}$, that holds when $A_n(x)$ is as described in §5.2.

The gradient $\nabla_{x^{(i)}}[\mu_n(x')]$ for $i = 1, 2$ is given in Lemma 9 where we replace x by $x^{(i)}$. The derivation is similar and is hence omitted. The derivation of $\nabla_{x^{(i)}}[\tilde{\sigma}_n(\vec{x}, x', \beta)]$ when sampling pairs differs from that of the gradient when sampling a singleton, so details follow.

Lemma 11. *For $i = 1, 2$,*

$$\begin{aligned}
&\nabla_{x^{(i)}}[\tilde{\sigma}_n(\vec{x}, x', \beta)] \\
&\quad = \frac{1}{B^2}\big\{B\nabla_{x^{(i)}}[\Sigma_n(x', x^{(1)}) - \Sigma_n(x', x^{(2)})]. \\
&\qquad\qquad - [\Sigma_n(x', x^{(1)}) - \Sigma_n(x', x^{(2)})]\nabla_{x^{(i)}}(B)\big\}, \qquad \text{(B4)}
\end{aligned}$$

where

$$\begin{aligned}
B &:= \big\{\beta^{-1}[\Lambda(x^{(1)}, x^{(1)}) + \Lambda(x^{(2)}, x^{(2)}) - 2\Lambda(x^{(1)}, x^{(2)})] \\
&\qquad + \Sigma_n(x^{(1)}, x^{(1)}) + \Sigma_n(x^{(2)}, x^{(2)}) - 2\Sigma_n(x^{(1)}, x^{(2)})\big\}^{1/2}, \qquad \text{(B5)}
\end{aligned}$$

$$\begin{aligned}
&\nabla_{x^{(i)}}[\Sigma_n(x', x^{(1)}) - \Sigma_n(x', x^{(2)})] \\
&= \begin{cases}
\nabla_{x^{(1)}}[\Sigma_0(x', x^{(1)})] - J_n(x^{(1)})[S_n]^{-1}\Sigma_0(\mathscr{X}_n, x'), \\
\quad \text{if } i = 1, x' \neq x^{(1)} \\
\nabla_{x^{(1)}}[\Sigma_0(x^{(1)}, x^{(1)}) - \Sigma_0(x^{(1)}, x^{(2)})] \\
\quad - J_n(x^{(1)})[S_n]^{-1}[2\Sigma_0(\mathscr{X}_n, x^{(1)}) - \Sigma_0(\mathscr{X}_n, x^{(2)})], \\
\quad \text{if } i = 1, x' = x^{(1)} \\
-\nabla_{x^{(2)}}[\Sigma_0(x', x^{(2)})] + J_n(x^{(2)})[S_n]^{-1}\Sigma_0(\mathscr{X}_n, x'), \\
\quad \text{if } i = 2, x' \neq x^{(2)} \\
\nabla_{x^{(2)}}[\Sigma_0(x^{(1)}, x^{(2)}) - \Sigma_0(x^{(2)}, x^{(2)})] \\
\quad + J_n(x^{(2)})[S_n]^{-1}[2\Sigma_0(\mathscr{X}_n, x^{(2)}) - \Sigma_0(\mathscr{X}_n, x^{(1)})], \\
\quad \text{if } i = 2, x' = x^{(2)}
\end{cases} \qquad \text{(B6)}
\end{aligned}$$

and

$$\begin{aligned}
\nabla_{x^{(i)}}(B) = \frac{1}{B}\big\{&\nabla_{x^{(i)}}\big[\tfrac{1}{2}[\beta^{-1}\Lambda(x^{(i)}, x^{(i)}) + \Sigma_0(x^{(i)}, x^{(i)})] \\
&- [\beta^{-1}\Lambda(x^{(1)}, x^{(2)}) + \Sigma_0(x^{(1)}, x^{(2)})]\big] \\
&+ J_n(x^{(i)})[S_n]^{-1}[\Sigma_0(\mathscr{X}_n, x^{3-i}) - \Sigma_0(\mathscr{X}_n, x^i)]\big\}.
\end{aligned}$$

Proof of Lemma 11. First, recall that $\tilde{\sigma}_n(\vec{x}, x', \beta) = (1/B) \cdot [\Sigma_n(x', x^{(1)}) - \Sigma_n(x', x^{(2)})]$, hence (B4) follows. Using (6) and (7), similar to the proof of Lemma 10, we have for $i = 1, 2$ that $\Sigma_n(x', x^{(i)}) = \Sigma_0(x', x^{(i)}) - \Sigma_0(x^{(i)}, \mathscr{X}_n)[S_n]^{-1}\Sigma_0(\mathscr{X}_n, x')$.

We show the first two cases ($i = 1$) of (B6). The other two cases ($i = 2$) follow similarly, and are omitted. In the first case, $x' \neq x^{(1)}$, we have $\nabla_{x^{(1)}}[\Sigma_n(x', x^{(1)}) - \Sigma_n(x', x^{(2)})] = \nabla_{x^{(1)}}[\Sigma_n(x', x^{(1)})] = \nabla_{x^{(1)}}[\Sigma_0(x', x^{(1)})] - J_n(x^{(1)})[S_n]^{-1}\Sigma_0(\mathscr{X}_n, x')$. In the second case, $x' = x^{(1)}$, then from the observation that $\Sigma_n(x^{(1)}, x^{(1)}) - \Sigma_n(x^{(1)}, x^{(2)}) = \Sigma_0(x^{(1)}, x^{(1)}) - \Sigma_0(x^{(1)}, x^{(2)}) - \Sigma_0(x^{(1)}, \mathscr{X}_n)[S_n]^{-1}\Sigma_0(\mathscr{X}_n, x^{(1)}) + \Sigma_0(x^{(1)}, \mathscr{X}_n)[S_n]^{-1}\Sigma_0(\mathscr{X}_n, x^{(2)})$, it follows from standard matrix differentiation and the definition of $J_n(x^{(1)})$ that $\nabla_{x^{(1)}}[\Sigma_n(x^{(1)}, x^{(1)}) - \Sigma_n(x^{(1)}, x^{(2)})] = \nabla_{x^{(1)}}[\Sigma_0(x^{(1)}, x^{(1)}) - \Sigma_0(x^{(1)}, x^{(2)})] - 2J_n(x^{(1)})[S_n]^{-1}\Sigma_0(\mathscr{X}_n, x^{(1)}) + J_n(x^{(1)})[S_n]^{-1}\Sigma_0(\mathscr{X}_n, x^{(2)})$.

It remains to compute $\nabla_{x^{(i)}}(B)$. Notice that for $i = 1$,

$$\begin{aligned}
&\nabla_{x^{(1)}}[\Sigma_n(x^{(1)}, x^{(1)}) + \Sigma_n(x^{(2)}, x^{(2)}) - 2\Sigma_n(x^{(1)}, x^{(2)})] \\
&\quad = \nabla_{x^{(1)}}[\Sigma_n(x^{(1)}, x^{(1)}) - \Sigma_n(x^{(1)}, x^{(2)})] \\
&\qquad - \nabla_{x^{(1)}}[\Sigma_n(x^{(2)}, x^{(1)}) - \Sigma_n(x^{(2)}, x^{(2)})] \\
&\quad = \nabla_{x^{(1)}}[\Sigma_0(x^{(1)}, x^{(1)}) - 2\Sigma_0(x^{(1)}, x^{(2)})] \\
&\qquad + 2J_n(x^{(1)})[S_n]^{-1}[\Sigma_0(\mathscr{X}_n, x^{(2)}) - \Sigma_0(\mathscr{X}_n, x^{(1)})],
\end{aligned}$$

where the last equation follows from (B6). The formular for $\nabla_{x^{(1)}}(B)$ then follows from the definition of B.

The formula for $\nabla_{x^{(2)}}(B)$ is similar. □

B.4. Simplification Under Compound Sphericity, Constant Prior Mean, and Gaussian Kernel

The gradients of the VOI and KG factors in Appendices B.2–B.3 involve the gradients of the sampling covariance matrix, and of the mean and covariance for the unknown mean θ. These terms, $\nabla_x[\Lambda(x, x')]$, $\nabla_x[\mu_0(x)]$ and $\nabla_x[\Sigma_0(x, x')]$ for arbitrary x, x', are significantly simplified if one adopts the modeling choices from §§7.1–7.2: a GP prior with a Gaussian kernel and constant mean, and compound sphericity.

First, under compound sphericity, $\nabla_x[\Lambda(x, x')] = \vec{0}$ for arbitrary x and x'. Second, under constant prior mean, $\nabla_x[\mu_0(x)] = \nabla_x[\eta] = \vec{0}$. Third, we compute $\nabla_x[\Sigma_0(x, x')]$ for arbitrary x and x'. Denote by $\circ$ the Hadamard (componentwise) product of two vectors u and v of the same length, so that $(u \circ v)(i) = u(i)v(i)$. Then $\nabla_x[\Sigma_0(x, x')] = 2\Sigma_0(x, x')\alpha \circ [\zeta(x) - \zeta(x')]$. In particular, $\nabla_x[\Sigma_0(x, x)] = \nabla_x[\sigma_0^2] = \vec{0}$.

References

Andradóttir S (1998) Simulation optimization. *Handbook of Simulation: Principles, Methodology, Advances, Applications, and Practice* (John Wiley & Sons, New York), 307–333.

Andradóttir S (2006) An overview of simulation optimization via random search. Henderson SG, Nelson BL, eds. *Handbooks in Operations Research and Management Science: Simulation* (North Holland, Amsterdam), 617–631.

Ankenman B, Nelson BL, Staum J (2010) Stochastic Kriging for simulation metamodeling. *Oper. Res.* 58(2):371–382.

Barton RR (2009) Simulation optimization using metamodels. Rossetti MD, Hill RR, Johansson B, Dunkin A, Ingalls RG, eds. *Proc. Winter Simulation Conf.* (IEEE, Piscataway, NJ), 230–238.

Branke J, Chick SE, Schmidt C (2007) Selecting a selection procedure. *Management Sci.* 53(12):1916–1932.

Brochu E, Cora VM, de Freitas N (2009) A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. Technical Report TR-2009-23, Department of Computer Science, University of British Columbia, Vancouver.

Chen CH, Lee LH (2010) *Stochastic Simulation Optimization: An Optimal Computing Budget Allocation* (World Scientific, Singapore).

Chen X, Ankenman BE, Nelson BL (2012) The effects of common random numbers on stochastic Kriging metamodels. *ACM TOMACS* 22(7):1–20.

Chen X, Ankenman BE, Nelson BL (2013) Enhancing stochastic Kriging metamodels with gradient estimators. *Oper. Reesarch* 61(2):512–528.

Chick SE (2006) Bayesian ideas and discrete event simulation: Why, what and how. Perrone LF, Wieland FP, Liu J, Lawson BG, Nicol DM, Fujimoto RM, eds. *Proc. Winter Simulation Conf.* (IEEE, Piscataway, NJ), 96–105.

Chick SE, Frazier PI (2012) Sequential sampling for selection with economics of selection procedures. *Management Sci.* 58(3):550–569.
Chick SE, Inoue K (2001a) New procedures to select the best simulated system using common random numbers. *Management Sci.* 47(8): 1133–1149.
Chick SE, Inoue K (2001b) New two-stage and sequential procedures for selecting the best simulated system. *Oper. Res.* 49(5):732–743.
Clark GM, Yang W (1986) A Bonferroni selection procedure when using common random numbers with unknown variances. Wilson J, Henriksen J, Roberts S, eds. *Proc. Winter Simulation Conf.* (IEEE, Piscataway, NJ), 313–315.
Cressie NAC (1993) *Statistics for Spatial Data*, Revised ed. Wiley Series in Probability and Mathematical Statistics: Applied Probability and Statistics (Wiley Interscience, New York).
Forrester A, Sobester A, Keane A (2008) *Engineering Design via Surrogate Modelling: A Practical Guide* (Wiley, West Sussex, UK).
Frazier P (2009–2010) http://people.orie.cornell.edu/pfrazier/src.html.
Frazier PI (2010) Decision-theoretic foundations of simulation optimization. *Wiley Encyclopedia of Operations Research and Management Science* (John Wiley & Sons, Hoboken, NJ).
Frazier PI, Powell WB, Dayanik S (2008) A knowledge gradient policy for sequential information collection. *SIAM J. Control Optim.* 47(5): 2410–2439.
Frazier P, Powell WB, Dayanik S (2009) The knowledge gradient policy for correlated normal beliefs. *INFORMS J. Comput.* 21(4):599–613.
Frazier PI, Xie J, Chick SE (2011) Value of information methods for pairwise sampling with correlations. Jain S, Creasey RR, Himmelspach J, White KP, Fu M, eds. *Proc. Winter Simulation Conf.* (IEEE, Piscataway, NJ), 3979–3991.
Fu MC (2002) Optimization for simulation: Theory vs. practice. *INFORMS J. Comput.* 14(3):192–215.
Fu MC, Hu JQ, Chen CH, Xiong X (2004) Optimal computing budget allocation under correlated sampling. Ingalls RG, Rossetti MD, Smith JS, Peters BA, eds. *Proc. Winter Simulation Conf.* (IEEE, Piscataway, NJ), 595–603.
Gelman AB, Carlin JB, Stern HS, Rubin DB (2004) *Bayesian Data Analysis*, Second ed. (CRC Press, Boca Raton, FL).
Gupta SS, Miescke KJ (1996) Bayesian look ahead one-stage sampling allocations for selection of the best population. *J. Statist. Planning and Inference* 54(2):229–244.
Hong LJ, Nelson BL (2006) Discrete optimization via simulation using COMPASS. *Oper. Res.* 54(1):115–129.
Hong LJ, Nelson BL, Henderson SG, Xie J (2012) Accessed July 1, 2013, http://simopt.org/wiki/index.php?title=Assemble_to_order .
Hu J, Wang Y, Zhou E, Fu MC, Marcus SI (2012) A survey of some model-based methods for global optimization. *Optimization, Control, and Applications of Stochatic Systems: In honor of Onésimo Hernández-Lerma* (Birkhäuser), 157–179.
Huang D, Allen TT, Notz WI, Zeng N (2006) Global optimization of stochastic black-box systems via sequential Kriging meta-models. *J. Global Optim.* 34(3):441–466.
Jones DR, Schonlau M, Welch WJ (1998) Efficient global optimization of expensive black-box functions. *J. Global Optim.* 13(4):455–492.
Kim SH (2005) Comparison with a standard via fully sequential procedures. *ACM TOMACS* 15(2):155–174.
Kim SH, Nelson BL (2006) Selecting the best system. Henderson SG, Nelson BL, eds. *Handbook in Operations Research and Management Science: Simulation* (North Holland, Amsterdam), 501–534.
Law AM (2014) *Simulation Modeling and Analysis*, Fifth ed. (McGraw-Hill, New York).
Nakayama MK (2000) Multiple comparisons with the best using common random numbers for steady-state simulations. *J. Statist. Planning and Inference* 85(1–2):37–48.
Nelson BL, Matejcik FJ (1995) Using common random numbers for indifference-zone selection and multiple comparisons in simulation. *Management Sci.* 41(12):1935–1945.
Øksendal B (2003) *Stochastic Differential Equations: An Introduction with Applications* (Springer, Berlin).
Raiffa H, Schlaifer R (1961) *Applied Statistical Decision Theory* (Harvard University, Cambridge, MA).
Rasmussen CE, Williams CKI (2006) *Gaussian Processes for Machine Learning* (MIT Press, Cambridge, MA).
Schruben LW, Margolin BH (1978) Pseudorandom number assignment in statistically designed simulation and distribution sampling experiments. *JASA* 73(363):504–525.
Scott W, Frazier PI, Powell WB (2011) The correlated knowledge gradient for simulation optimization of continuous parameters using Gaussian process regression. *SIAM J. Optim.* 21:996–1026.
Shi L, Ólafsson S (2000) Nested partitions method for stochastic optimization. *Methodology Comput. Appl. Probab.* 2(3):271–291.
Tew J, Wilson JR (1992) Validation of simulation analysis methods for the Schruben-Margolin correlation-induction strategy. *Oper. Res.* 40(1):87–103.
van Beers WCM, Kleijnen JPC (2008) Customized sequential designs for random simulation experiments: Kriging metamodeling and bootstrapping. *EJOR* 186(3):1099–1113.
Villemonteix J, Vazquez E, Walter E (2009) An informational approach to the global optimization of expensive-to-evaluate functions. *J. Global Optim.* 44(4):509–534.
Wang H, Pasupathy R, Schmeiser BW (2013) Integer-ordered simulation optimization using R-SPLINE: Retrospective search with piecewise-linear interpolation and neighborhood enumeration. *ACM Trans. Model. Comput. Simulation* 23(3):17:1–17:24.
Wang Y, Fu MC, Marcus SI (2010) Model-based evolutionary optimization. Johansson B, Jain S, Montoya-Torres J, Hugan J, Y ucesan E, eds. *Proc. Winter Simulation Conf.* (IEEE, Piscataway, NJ), 1199–1210.
Xu J, Nelson BL, Hong LJ (2010) Industrial Strength COMPASS: A comprehensive algorithm and software for optimization via simulation. *ACM Trans. Model. Comput. Simulation* 20(1):3:1–3:29.
Yang WN, Nelson BL (1991) Using common random numbers and control variates in multiple-comparison procedures. *Oper. Res.* 39(4):583–591.
Zhou E, Fu MC, Marcus SI (2008) A particle filtering framework for randomized optimization algorithms. Mason SJ, Hill RR, Mönch L, Rose O, Jefferson T, Fowler JW, eds. *Proc. Winter Simulation Conf.* (IEEE, Piscataway, NJ), 647–654.

Jing Xie holds a PhD in operations research and information engineering from Cornell University. Her thesis research was on Bayesian designs for sequential learning problems. She is now a director of data science at Credibly, a FinTech startup in New York City.

Peter I. Frazier is an associate professor in the School of Operations Research and Information Engineering at Cornell University, and a Senior Data Scientist at Uber. His research interests include optimal learning, sequential decision making under uncertainty, and machine learning, focusing on applications in simulation optimization, materials science, e-commerce, and medicine.

Stephen E. Chick is a professor of Technology and Operations Management at INSEAD, and the Novartis Chair of Healthcare Management. He teaches operations with applications in manufacturing and services, particularly the healthcare sector. He enjoys Bayesian statistics, stochastic models, and simulation.