Gaussian Markov Random Fields for Discrete Optimization via Simulation: Framework and Algorithms

Peter Salemi
MITRE Corporation

Eunhye Song
Barry L. Nelson
Jeremy Staum
Northwestern University

July 16, 2015

Abstract

We consider optimizing the expected value of some performance measure of a dynamic stochastic simulation when the decision variables are discrete, in particular integer-ordered, and the number of feasible solutions is very large. Our goal is to have inference like the correct-selection guarantee of ranking and selection without having to simulate all feasible solutions. To accomplish this we learn and exploit spatial relationships among the decision variables and objective function values within a discrete optimization via simulation (DOvS) framework based on Gaussian Markov random fields (GMRFs). Gaussian random fields on continuous domains are already used in deterministic and stochastic optimization because they facilitate the computation of measures like expected improvement (EI) that balance exploration and exploitation. We show that GMRFs are particularly well suited to the DOvS problem, both from a modeling and a computational perspective. Specifically, GMRFs permit the definition of a sensible neighborhood structure, and they are defined by their precision matrices which can be constructed to be sparse. We also introduce a new EI criterion that more fully accounts for the spatial relationships as they are learned. Using this framework we create both single and multiresolution algorithms, prove their asymptotic convergence, and evaluate their finite-time performance empirically.

1 Introduction

Optimization plays a central role in operations research. For many important stochastic optimization problems that occur in practice, the objective function cannot be evaluated exactly and instead must be estimated using simulation. For example, if we are interested in minimizing the expected waiting time for an arbitrary customer contacting a call center that has constraints on its service budget, and the structure of the call center makes the expected waiting time mathematically intractable, then customer waiting times for any particular staffing assignment must be simulated. Problems such as this are commonly referred to as optimization via simulation (OvS). OvS problems in which the decision variables can only assume discrete values are called discrete optimization via simulation (DOvS) problems, and they present particular challenges. In this paper, we focus on DOvS with integer-ordered decision variables.

Due to its practical importance, DOvS has been an active area of research for many years, and the literature on DOvS is plentiful. Remarkable theoretical and practical success has been achieved using exhaustive search algorithms known collectively as ranking and selection (R&S). R&S algorithms simulate all feasible solutions and terminate with either a guaranteed probability of correct selection (frequentist) or posterior assessment of the relative quality of the selected solution (Bayesian); see Kim [2013] and Frazier [2012], respectively, for surveys. These guarantees or assessments are comforting because even when all feasible solutions are simulated it is not possible to promise optimality in finite time except in very special cases.

When the number of feasible solutions is too large to exhaust, then many DOvS algorithms fall into the category of adaptive random search (ARS). ARS algorithms contain a mixture of both local and global searching in an effort to explore favorable areas of the feasible region in depth (local searching), while continuing to explore other areas of the feasible region to avoid getting trapped at a poor local optimum (global searching). These algorithms attempt to leverage, at least in an informal way, anticipated spatial structure such as "the decision-variable values of good feasible solutions tend to be clustered." ARS algorithms are usually proven to be either globally or locally convergent as the simulation effort increases without bound, and they may be augmented with a R&S algorithm to provide either a statistical guarantee of selecting the best solution actually simulated, or of local optimality. See, for example, the survey in Nelson [2010].

In this paper we provide a DOvS framework to learn spatial relationships, to balance exploration and exploitation, and to facilitate a stopping criterion that considers the uncertainty at feasible solutions that have and have not been simulated for very large-scale problems. Loosely speaking, we want efficient search with R&S-like conclusions when we stop. We will represent the objective-function response surface as a Gaussian random field (GRF), because GRFs support assessments of the benefit of expending simulation effort in various ways, and statistical inference on the potential of unseen solutions. GRF-based optimization methods were introduced for deterministic computer experiments in Jones et al. [1998], followed by Huang et al. [2006] which considered computer experiments with noisy output. Recently, Frazier et al. [2011] and Quan et al. [2013] created GRF-based optimization methods for DOvS. Other GRF-based methods include Xu [2012], which combines ARS with stochastic kriging [Ankenman et al., 2010] to help increase the efficiency of the ARS method, and Gaussian process-based search [Sun et al., 2011], which uses the GRF to create a sampling distribution that favors feasible solutions likely to be optimal. All of these methods use a GRF with a continuous domain and then project the problem into the discrete setting. Unfortunately, the standard covariance functions for continuous domains lead to a dense covariance matrix, and the inverse of this matrix is used for fitting the GRF and generating inference that guides the search. Therefore, these methods can be poorly suited for large-scale problems with many feasible solutions due to slow and ill-conditioned numerical computations.

A central contribution of this paper is introducing a GRF that is appropriate for the DOvS problem. Specifically, we model the objective function values at the feasible solutions as a realization of a discrete Gaussian Markov random field (GMRF) [Rue and Held, 2005]. The Markov structure of GMRFs is intuitive for problems in operations research [Salemi et al., 2013]. For example, if we are interested in predicting the value of the objective function at a feasible solution, then the values of the objective function at the feasible solutions in a neighborhood of it would often be sufficient; other feasible solutions would provide little additional information. This type of structure can be represented in a GMRF, and the neighborhood is user selected. Further, GMRFs can be defined on a lattice so the use of GMRFs in DOvS problems is more natural than using a GRF with a continuous domain. Most importantly, the Markov structure lends itself to efficient and numerically stable calculations for large-scale problems. The dependence in a GMRF is defined by its precision matrix, which is the inverse of the covariance matrix. Using the Markov structure of GMRFs, the

precision matrix can be constructed to be sparse. Thus, we can use sparse matrix techniques to calculate expressions that involve the precision matrix.

A second contribution is to establish an appropriate expected improvement (EI) guidance and stopping criterion. Jones et al. [1998] first introduced EI for optimization problems with deterministic computer experiments. In their setting the objective function can be observed without noise, so their EI criterion does not incorporate the uncertainty in the output from a stochastic simulation. Huang et al. [2006] and Quan et al. [2013] proposed other EI criteria for the case of noisy output; however, both criteria treat the value at the current sample-best solution as fixed instead of as a random variable. We propose an EI criterion similar to that of Williams et al. [2000], which we call complete expected improvement (CEI). CEI incorporates the uncertainty in stochastic simulation by treating the value at the current sample-best solution as a random variable, just like the values of the feasible solutions being considered to improve it. The main difference between the EI of Williams et al. [2000] and our CEI is the origin of the uncertainty. Williams et al. [2000] consider deterministic computer experiments with uncertainty introduced by inputs that contain noise, whereas our uncertainty is introduced by the stochastic noise in the simulation output. Similar to EI, the Knowledge Gradient (KG) policy [Frazier et al., 2009] can also be used to guide the allocation of simulation effort and automatically regulate the exploration vs. exploitation trade-off. However, the KG policy evaluates the benefit of running additional replications at a feasible solution, whereas EI estimates how much better the unknown true value of another feasible solution is with respect to the current sample-best solution. This is important because we use the CEI not only for search guidance, but also to decide when we have sufficiently searched the feasible region: we stop when the CEI is too small to warrant more searching.

Although our GMRF framework facilitates the use of sparse matrix techniques that support algorithms for large numbers of feasible solutions, there will nevertheless be a computational limit on the size of problem we can handle directly. *A third contribution of this paper is to extend to the GMRF approach to a multiresolution framework that can be used to solve DOvS problems with vast numbers of solutions.* The multiresolution framework exploits a region-level GMRF to learn about the quality of disjoint regions of the solution space, and solution-level GMRFs to learn about the quality of individual feasible solutions within each region.

The paper presents both a GMRF framework within which algorithms can be designed, and spe-

cific solution-level and multiresolution DOvS algorithms. The algorithms are shown to converge to a globally optimal solution as simulation effort increases without bound under very mild conditions (essentially finite variance). More importantly, however, these algorithms can terminate well short of infinite effort with statistical assurance of having discovered an optimal or near-optimal solution; this inference depends on the GMRF assumptions.

In the next section we present the GMRF framework, which includes defining GMRFs and explaining how they can represent a DOvS problem. Section 3 details a specific solution-level DOvS algorithm, including parameter estimation and sparse matrix calculation techniques. For problems with vast numbers of feasible solutions, we extend the framework to a multiresolution algorithm in Section 4. Results of numerical experiments are found in Section 5, and a brief summary is provided in Section 6.

2 A Framework for DOvS using GMRFs

A GMRF is a non-degenerate multivariate Gaussian random vector $\mathbb{Y} = (\mathbb{Y}_1, \mathbb{Y}_2, \ldots, \mathbb{Y}_n)^\top$ that is associated with an undirected and labeled graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes the set of nodes and $\mathcal{E}$ denotes the set of edges; see Rue and Held [2005]. Each node in $\mathcal{V}$ is associated with a unique element of $\mathbb{Y}$. Two nodes in the graph are called neighbors if they are connected by an edge. If we denote the mean vector and precision matrix of $\mathbb{Y}$ by $\boldsymbol{\mu}$ and $\mathbf{Q}$, respectively, then we can write the probability density function of the GMRF as

$$(2\pi)^{-n/2}|\mathbf{Q}|^{1/2}\exp\left(-\frac{1}{2}(\mathbf{y}-\boldsymbol{\mu})^\top\mathbf{Q}\,(\mathbf{y}-\boldsymbol{\mu})\right),$$

where the positive-definite precision matrix $\mathbf{Q}$ is the inverse of the covariance matrix. The diagonal entries of the precision matrix are the conditional precisions

$$\mathrm{Prec}(\mathbb{Y}_i|\mathbb{Y}_{\mathcal{V}\setminus\{i\}}) = \mathrm{Q}_{ii}, \tag{1}$$

where $\mathbb{Y}_{\mathcal{V}\setminus\{i\}}$ is the vector of values of the GMRF observed only at the nodes $\mathcal{V}\setminus\{i\}$. The scalar precision Q_{ii} is the reciprocal of the conditional variance. The off-diagonal elements are proportional

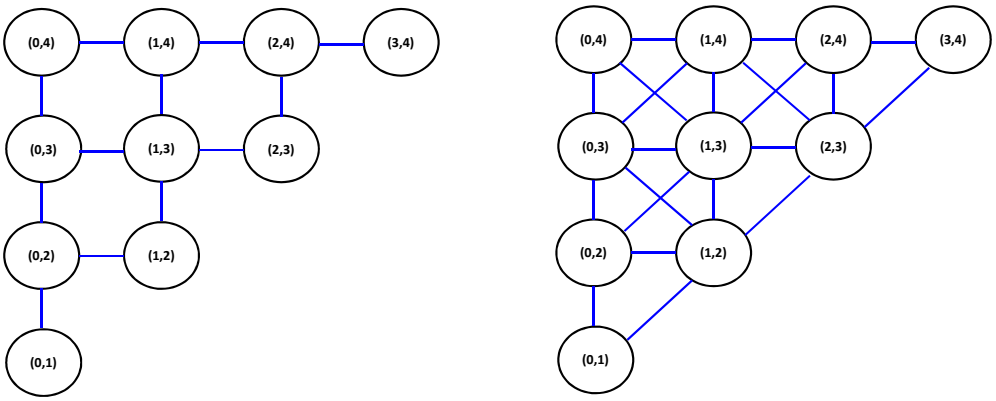


Figure 1: A portion of two graphs $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ corresponding to a subset of the integer lattice $\mathbb{Z}^2$ with the constraint $x_1 < x_2$ and illustrating the neighborhoods $\mathcal{N}(\mathbf{x})$ (left) and $\mathcal{N}^+(\mathbf{x})$ (right).

to conditional correlations, specifically

$$\text{Corr}(\mathbb{Y}_i, \mathbb{Y}_j | \mathbb{Y}_{\mathcal{V}\setminus\{i,j\}}) = -\frac{\text{Q}_{ij}}{\sqrt{\text{Q}_{ii}\text{Q}_{jj}}},$$

where $\mathbb{Y}_{\mathcal{V}\setminus\{i,j\}}$ is the vector of values of the GMRF observed only at the nodes $\mathcal{V}\setminus\{i,j\}$. The graph $\mathcal{G}$ determines the non-zero pattern of the precision matrix $\mathbf{Q}$, and vice versa, since for a GMRF $\text{Q}_{ij} \neq 0$ if and only if $\{i,j\} \in \mathcal{E}$. Thus, the precision matrix will be sparse if the set of edges is small in size, and vice versa. GMRFs are "Markov" because they possess the local Markov property:

$$\mathbb{Y}_i \perp \mathbb{Y}_{\mathcal{V}\setminus\{i,\mathcal{N}(i)\}} | \mathbb{Y}_{\mathcal{N}(i)} \quad \text{for every} \quad i \in \mathcal{V},$$

where $\mathcal{N}(i)$ is the set of neighbors of node i in $\mathcal{G}$; that is, $\mathcal{N}(i) = \{j\colon\ \{i,j\} \in \mathcal{E}\}$.

To better understand the local Markov property, consider the left-hand graph $\mathcal{G}$ in Figure 1, which represents a portion of a feasible region where the nodes correspond to solutions $\mathbf{x}^\top = (x_1, x_2)$ with constraint $x_1 < x_2$. If we observe the values of the GMRF at $\mathbf{x} = (0,3)$, $(1,4)$, $(2,3)$ and $(1,2)$, then our prediction of the value of the GMRF at $(1,3)$ is not affected by information about the GMRF at any other lattice point in the graph. The Markov property does *not* imply that nodes far away from one another are independent, however, but rather that if we know the value of the GMRF at nodes close by, then we can ignore nodes farther away conditional on those values. To completely specify a GMRF we only need to specify the mean $\boldsymbol{\mu}$ and the precision matrix $\mathbf{Q}$, whose non-zero pattern is associated with the structure of the graph $\mathcal{G}$.

2.1 A Precision Matrix for DOvS

In a DOvS problem with integer-ordered decision variables, the feasible region $\mathcal{X}$ is a finite subset of the d-dimensional integer lattice $\mathbb{Z}^d$. Thus, the straightforward construction of the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ starts with defining the nodes of the graph $\mathcal{V}$ to be $\mathcal{X}$. To finish the construction of $\mathcal{G}$, we must specify the neighborhood structure for any given node/solution.

Two sensible neighborhoods for a solution $\mathbf{x} \in \mathcal{X}$ are $\mathcal{N}(\mathbf{x}) = \{\mathbf{x}' \in \mathcal{X} \colon ||\mathbf{x} - \mathbf{x}'||_2 = 1\}$, which has an upper bound of $2d$ neighbors in d dimensions, and $\mathcal{N}^+(\mathbf{x}) = \{\mathbf{x}' \in \mathcal{X} \colon ||\mathbf{x} - \mathbf{x}'||_\infty = 1\}$, which has an upper bound of $3^d - 1$ neighbors in d dimensions. Both are illustrated in Figure 1 for $d = 2$.

Since we are particularly interested in DOvS problems with large feasible regions, and because (roughly speaking) the more sparse $\mathbf{Q}$ is the larger the DOvS problems we can solve, our algorithm will use the neighborhood structure $\mathcal{N}(\mathbf{x})$. For this neighborhood $|\mathcal{E}| \leq 2d|\mathcal{X}|$, and thus the fraction of non-zero entries in the precision matrix $\mathbf{Q}$ is bounded from above by $2d/|\mathcal{X}|$, which is very small for large problems. Although we assume this specific neighborhood for efficient computations, our GMRF framework allows any neighborhood structure and our convergence proof does not depend on it.

Defining the graph $\mathcal{G}$, and thus the non-zero pattern of the precision matrix, does not specify the values of the non-zero entries of $\mathbf{Q}$. A standard approach is to have the entries of $\mathbf{Q}$ given by a function $p(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is a vector of parameters; i.e., $\mathrm{Q}_{ij} \triangleq p(\mathbf{x}_i, \mathbf{x}_j; \boldsymbol{\theta})$. As one possible parameterization for the neighborhood $\mathcal{N}(\mathbf{x})$, we propose $\boldsymbol{\theta} = (\theta_0, \theta_1, \theta_2, \ldots, \theta_d)^\top$ and

$$p(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta}) = \begin{cases} \theta_0, & \text{if } \mathbf{x} = \mathbf{x}' \\ -\theta_0\theta_j, & \text{if } |\mathbf{x} - \mathbf{x}'| = \mathbf{e}_j \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

for $\mathbf{x}, \mathbf{x}' \in \mathbb{Z}^d$, where $\mathbf{e}_j$ is the jth standard basis vector. Recall that Q_{ii} is the conditional precision of solution i, and $-\mathrm{Q}_{ij}/\sqrt{\mathrm{Q}_{ii}\mathrm{Q}_{jj}}$ is the conditional correlation between solutions i and j. Thus, θ_0 is the conditional precision of each solution, and θ_j is the conditional correlation between solutions that differ by 1 in the jth coordinate direction. That the conditional correlations can depend on the coordinate direction is important to allow for response surfaces that change more rapidly in

one direction as compared to another.

Since the conditional precisions must be positive, it follows that $\theta_0 > 0$. We also want neighbors to have non-negative conditional correlations, so we restrict the values of $\theta_1, \theta_2, \ldots, \theta_d$ to be non-negative as well. Furthermore, we need $\theta_j \leq 1$ for $j = 1, 2, \ldots, d$, since the conditional correlations must be less than one. Lastly, $\mathbf{Q}$ should be positive-definite. We will use these restrictions later in Section 3.2 when we calculate the maximum likelihood estimates of the parameters. With this parameterization and these restrictions, $\mathbf{Q}$ is a non-singular M-matrix so its inverse is nonnegative, i.e., $[\mathbf{Q}^{-1}]_{ij} \geq 0$ for all i and j. In other words, there are no negative unconditional correlations among nodes in the GMRF, which is a property that makes sense in many DOvS problems.

Our function p leads to a nonstationary GMRF because the unconditional variances of feasible solutions with more neighbors are larger than the variances of feasible solutions with fewer neighbors. We are not concerned with the nonstationarity of the GMRF since we only use the conditional variance, given the observed simulation output, in our framework. We write $\mathbf{Q}(\boldsymbol{\theta})$ to explicitly show the dependence of the precision matrix on the parameters.

2.2 A GMRF Representation of DOvS

We consider the DOvS problem minimize $y(\mathbf{x}) \triangleq \mathrm{E}[Y(\mathbf{x})]$ subject to $\mathbf{x} \in \mathcal{X}$, where $\mathcal{X}$ is a finite subset of the d-dimensional integer lattice $\mathbb{Z}^d$ and $n \triangleq |\mathcal{X}|$. The distribution of the random variable $Y(\mathbf{x})$, as a function of the feasible solution $\mathbf{x}$, is unknown. However, the expectation $y(\mathbf{x}) \triangleq \mathrm{E}[Y(\mathbf{x})]$ can be estimated using stochastic simulation. More formally, for any feasible solution $\mathbf{x}$ we are able to observe

$$Y_j(\mathbf{x}) = y(\mathbf{x}) + \epsilon_j(\mathbf{x}) \tag{3}$$

on replication $j = 1, 2, \ldots$, where $\{\epsilon_j(\mathbf{x})\}$ are i.i.d. with mean 0 and finite variance that may depend on $\mathbf{x}$. Below we will also assume that the $\epsilon_j(\mathbf{x})$ are normally distributed. While we assume the ability to make replications in this paper, a single-run steady-state simulation setting could be accommodated with, say, batch means playing the role of $Y_j(\mathbf{x})$.

Let $\mathbf{y}$ denote the vector of objective function values $(y(\mathbf{x}_1), y(\mathbf{x}_2), \ldots, y(\mathbf{x}_n))^\top$. Of course $\mathbf{y}$ is

unknown, so we model it as a realization of the GRF

$$\mathbb{Y} \triangleq (\mathbb{Y}(\mathbf{x}_1), \mathbb{Y}(\mathbf{x}_2), \ldots, \mathbb{Y}(\mathbf{x}_n))^\top \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{Q}(\boldsymbol{\theta})^{-1}) \tag{4}$$

with mean vector $\boldsymbol{\mu}$ and precision matrix $\mathbf{Q}(\boldsymbol{\theta})$, as defined above.

Similar to virtually all methods for optimization based on GRFs, the essence of our approach is to use the conditional distribution of the GMRF after having simulated some feasible solutions as the guidance and inference engine for our search. Throughout the paper we will use the term "design point" to refer to a feasible solution $\mathbf{x}$ that has been simulated for any non-zero number of replications, and we will use the terms "point" and "feasible solution" interchangeably. Therefore, we are interested in the conditional distribution of the GMRF given simulation output at the design points.

Let $\Xi_2 \subseteq \mathcal{X}$ denote the current set of design points and partition $\mathcal{X}$ into the two disjoint sets Ξ_2 and $\Xi_1 = \mathcal{X} \setminus \Xi_2$. Thus, Ξ_2 is the set of feasible solutions that have been simulated, and Ξ_1 is the set of feasible solutions that have not. For simplicity, we use "1" as a subscript to denote quantities associated with the set Ξ_1 and "2" as a subscript to denote quantities associated with the set Ξ_2. For instance, $n_1 = |\Xi_1|$ and $n_2 = |\Xi_2|$ are the numbers of solutions in each set.

Using these disjoint sets, we can partition the vectors $\mathbf{y}, \mathbb{Y}, \boldsymbol{\mu}$, and the precision matrix $\mathbf{Q}(\boldsymbol{\theta})$, and rewrite expression (4) as

$$\begin{pmatrix} \mathbb{Y}_1 \\ \mathbb{Y}_2 \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{pmatrix}, \begin{pmatrix} \mathbf{Q}_{11}(\boldsymbol{\theta}) & \mathbf{Q}_{12}(\boldsymbol{\theta}) \\ \mathbf{Q}_{12}(\boldsymbol{\theta})^\top & \mathbf{Q}_{22}(\boldsymbol{\theta}) \end{pmatrix}^{-1} \right).$$

Let $\bar{\mathbb{y}}_2$ be the vector of sample means of the simulation output at the design points. Consistent with the output model (3), we represent $\bar{\mathbb{y}}_2$ as a realization of the GMRF $\mathbb{Y}_2^\epsilon = \mathbb{Y}_2 + \boldsymbol{\epsilon}$, with $\mathbb{Y}_2$ and $\boldsymbol{\epsilon}$ independent, and

$$\boldsymbol{\epsilon} \sim \mathcal{N}(\vec{\mathbf{0}}_{n_2 \times 1}, \mathbf{Q}_\epsilon^{-1}), \tag{5}$$

where the matrix $\mathbf{Q}_\epsilon$, called the intrinsic precision matrix, is the precision matrix of the noise inherent to the stochastic simulation output $\bar{\mathbb{y}}_2$. When we simulate design points independently, $\mathbf{Q}_\epsilon$ is a diagonal matrix, whereas when we simulate with common random numbers (CRN), $\mathbf{Q}_\epsilon$ is a

dense matrix. The values in $\mathbf{Q}_\epsilon$ also depend on how many replications have been averaged, which need not be the same at all design points.

In Appendix A we show that the conditional distribution of $\mathbb{Y}$ given $\mathbb{Y}_2^\epsilon = \bar{\mathcal{Y}}_2$ is

$$\mathcal{N}\left(\begin{pmatrix}\boldsymbol{\mu}_1\\ \boldsymbol{\mu}_2\end{pmatrix} + \bar{\mathbf{Q}}(\boldsymbol{\theta})^{-1}\begin{pmatrix}\vec{\mathbf{0}}_{n_1\times 1}\\ \mathbf{Q}_\epsilon(\bar{\mathcal{Y}}_2 - \boldsymbol{\mu}_2)\end{pmatrix}, \bar{\mathbf{Q}}(\boldsymbol{\theta})^{-1}\right), \tag{6}$$

where

$$\bar{\mathbf{Q}}(\boldsymbol{\theta}) \triangleq \begin{pmatrix}\mathbf{Q}_{11}(\boldsymbol{\theta}) & \mathbf{Q}_{12}(\boldsymbol{\theta})\\ \mathbf{Q}_{12}(\boldsymbol{\theta})^\top & \mathbf{Q}_{22}(\boldsymbol{\theta})\end{pmatrix} + \begin{pmatrix}\mathbf{0}_{n_1\times n_1} & \mathbf{0}_{n_1\times n_2}\\ \mathbf{0}_{n_1\times n_2}^\top & \mathbf{Q}_\epsilon\end{pmatrix}$$

is the conditional precision matrix, and $\mathbf{0}_{n_i\times n_j}$ is the $n_i \times n_j$ matrix of zeros. The sparsity of $\bar{\mathbf{Q}}(\boldsymbol{\theta})$ is inherited directly from the sparsity of $\mathbf{Q}(\boldsymbol{\theta})$ and $\mathbf{Q}_\epsilon$. Furthermore, as will be discussed in Section 3.3, we can avoid direct inversion of $\bar{\mathbf{Q}}(\boldsymbol{\theta})$ by solving a sparse linear system whose coefficient matrix is the sparse matrix $\bar{\mathbf{Q}}(\boldsymbol{\theta})$.

2.3 Complete Expected Improvement

The EI criterion we introduce in this paper takes into account both the uncertainty at a candidate feasible solution $\mathbf{x}$ and the uncertainty at the current sample-best solution $\tilde{\mathbf{x}}$, the design point with the smallest sample mean value, by treating both as random variables. Let $\mathbb{Y}(\mathbf{x})$ be the Gaussian random variable from the GMRF $\mathbb{Y}$ corresponding to the feasible solution $\mathbf{x}$.

Given the simulation output $\bar{\mathcal{Y}}_2$, the conditional joint distribution of $\mathbb{Y}(\tilde{\mathbf{x}})$ and $\mathbb{Y}(\mathbf{x})$ is bivariate normal; it can be found from the rows and columns of Equation (6) corresponding to $\tilde{\mathbf{x}}$ and $\mathbf{x}$. Denote the conditional means by $M(\tilde{\mathbf{x}})$ and $M(\mathbf{x})$, the conditional variances by $V(\tilde{\mathbf{x}})$ and $V(\mathbf{x})$, and the conditional correlation by $\rho(\tilde{\mathbf{x}}, \mathbf{x})$. Let

$$V(\tilde{\mathbf{x}}, \mathbf{x}) \triangleq V(\tilde{\mathbf{x}}) + V(\mathbf{x}) - 2\rho(\tilde{\mathbf{x}}, \mathbf{x})\sqrt{V(\tilde{\mathbf{x}})V(\mathbf{x})}$$

be the variance of the difference $\mathbb{Y}(\tilde{\mathbf{x}}) - \mathbb{Y}(\mathbf{x})$. Then the CEI of the feasible solution $\mathbf{x}$, denoted by

CEI($\mathbf{x}$), is defined as

$$\begin{aligned}\text{CEI}(\mathbf{x}) &\triangleq \mathrm{E}\left[\max\left\{\mathbb{Y}(\tilde{\mathbf{x}}) - \mathbb{Y}(\mathbf{x}), 0\right\} | \bar{\mathcal{Y}}_2\right] \\ &= (M(\tilde{\mathbf{x}}) - M(\mathbf{x}))\,\Phi\left(\frac{M(\tilde{\mathbf{x}}) - M(\mathbf{x})}{\sqrt{V(\tilde{\mathbf{x}}, \mathbf{x})}}\right) + \sqrt{V(\tilde{\mathbf{x}}, \mathbf{x})}\phi\left(\frac{M(\tilde{\mathbf{x}}) - M(\mathbf{x})}{\sqrt{V(\tilde{\mathbf{x}}, \mathbf{x})}}\right), \qquad (7)\end{aligned}$$

where ϕ and Φ are the density and cumulative distribution function, respectively, of a standard normal random variable. We can use the expression for EI in Jones et al. [1998] to derive (7), by noting that conditional on $\bar{\mathcal{Y}}_2$ the difference $\mathbb{Y}(\tilde{\mathbf{x}}) - \mathbb{Y}(\mathbf{x})$ is a Gaussian random variable with mean $M(\tilde{\mathbf{x}}) - M(\mathbf{x})$ and variance $V(\tilde{\mathbf{x}}, \mathbf{x})$.

From the analysis in Jones et al. [1998], we know that CEI($\mathbf{x}$) is increasing in both $M(\tilde{\mathbf{x}}) - M(\mathbf{x})$ and $V(\tilde{\mathbf{x}}, \mathbf{x})$. Since CEI($\mathbf{x}$) is increasing in $M(\tilde{\mathbf{x}}) - M(\mathbf{x})$, CEI($\mathbf{x}$) is decreasing in $M(\mathbf{x})$. Therefore, solutions with smaller conditional means will tend to have larger CEIs. As the conditional correlation coefficient $\rho(\tilde{\mathbf{x}}, \mathbf{x})$ decreases, with $V(\tilde{\mathbf{x}})$ and $V(\mathbf{x})$ held constant, $V(\tilde{\mathbf{x}}, \mathbf{x})$ increases. Similarly, as $V(\mathbf{x})$ increases with $\rho(\tilde{\mathbf{x}}, \mathbf{x})$ and $V(\tilde{\mathbf{x}})$ held constant, $V(\tilde{\mathbf{x}}, \mathbf{x})$ increases. Feasible solutions that are far away from $\tilde{\mathbf{x}}$ will tend to have smaller values of $\rho(\tilde{\mathbf{x}}, \mathbf{x})$, and feasible solutions that are far away from the design points will tend to have larger values of $V(\mathbf{x})$. Thus, feasible solutions that are far away from $\tilde{\mathbf{x}}$ or far away from the design points will tend to have large CEIs. The solutions near $\tilde{\mathbf{x}}$, on the other hand, tend to have smaller means but larger values of $\rho(\tilde{\mathbf{x}}, \mathbf{x})$. The CEI criterion balances these factors.

By treating both the values at the current sample-best solution $\tilde{\mathbf{x}}$ and the candidate feasible solution $\mathbf{x}$ as random variables, we take into account their conditional correlation $\rho(\tilde{\mathbf{x}}, \mathbf{x})$. Thus, CEI is a more comprehensive assessment than other EI measures proposed for stochastic simulation.

3 A Solution-Level Algorithm

In this section we present an instance of our framework, the Gaussian Markov Improvement Algorithm (GMIA). We refer to it as a "solution-level algorithm" because the nodes of the GMRF correspond to individual feasible solutions $\mathbf{x}$, and we employ the neighborhood structure $\mathcal{N}(\mathbf{x})$ described in Section 2.1. An outline is given in Section 3.1, followed by estimation of the parameters of the GMRF in Section 3.2. We then present a method for efficiently calculating the conditional

distribution (6) using sparse matrix techniques in Section 3.3. GMIA represents specific choices that we have made and found effective, but additional enhancements, other neighborhood structures, and fine-tuning using known properties of a specific problem, and are certainly possible. Even using sparse matrix techniques, $|\mathcal{X}|$ can be so large that direct application of GMIA is computationally impossible. To address such cases we describe a multiresolution algorithm in Section 4 that exploits a region-level GMRF, as well as solution-level GMRFs.

3.1 GMIA Outline

GMIA utilizes the CEI to search for solutions to simulate, and terminates when the CEI drops below a specified threshold δ. This stopping criterion has been suggested in other EI-based methods, such as Huang et al. [2006].

GMIA

0. Generate a set of k_s design points. Simulate r replications for each design point and use the simulation output to calculate the maximum likelihood estimates (MLEs) for the GMRF parameters using the method in Section 3.2 and Appendix B.

1. Let $\tilde{\mathbf{x}}$, the current sample-best solution, be the design point with the smallest sample mean.

2. Calculate the CEI with respect to $\tilde{\mathbf{x}}$, defined in Section 2.3, for each candidate feasible solution. If the maximum CEI is less than δ, then go to Step 5. Otherwise, go to Step 3.

3. Simulate r replications at both $\tilde{\mathbf{x}}$, and the candidate feasible solution $\mathbf{x}^*_{\text{CEI}}$ that maximizes the CEI over the set of all candidate feasible solutions.

4. Update the simulation output at $\tilde{\mathbf{x}}$ with the new replications. If $\mathbf{x}^*_{\text{CEI}}$ is a design point then update the simulation output at $\mathbf{x}^*_{\text{CEI}}$ and go to Step 1. If $\mathbf{x}^*_{\text{CEI}}$ is not a design point, then add $\mathbf{x}^*_{\text{CEI}}$ to the set of design points, add the simulation output obtained at $\mathbf{x}^*_{\text{CEI}}$ to the collection of simulation output, and go to Step 1.

5. Return $\tilde{\mathbf{x}}$ as the estimated optimal solution.

The inputs to the algorithm are the initial number of design points k_s, the number of replications to be allocated each time a solution is simulated r (which need not be constant and can be adaptive

provided $r > 0$), and the CEI improvement threshold δ. We treat δ like the indifference-zone parameter in R&S by interpreting it as the smallest practically significant difference in solution quality that concerns us.

Although the parameters $\boldsymbol{\mu}, \boldsymbol{\theta}$ of the GMRF must be estimated initially (as described below), they do not need to be re-estimated each time we obtain additional simulation output. In fact, results in Diggle et al. [2010] show that a detrimental bias can be introduced in MLE-based parameter estimates when design points are selected using "preferential sampling," which means selecting design points that are expected to be associated with good regions of the response surface. This suggests that the initial number of design points k_s should not be too small, and should be selected to fill the feasible region to obtain good parameter estimates that need not be updated. Our experience supports this approach.

3.2 Parameter Estimation

A GMRF is defined by its mean vector and precision matrix, and we have made the precision matrix a function of a vector of parameters $\boldsymbol{\theta}$. For the mean vector we assume that $\boldsymbol{\mu} = (\beta_0, \beta_0, \ldots, \beta_0)^\top$; i.e., we assume a constant mean, which is appropriate when we have no prior information about the objective function values at the feasible solutions. If, for instance, a parametric trend model is known to be valid, such as $\mu(\mathbf{x}) = g(\mathbf{x})^\top \boldsymbol{\beta}$, then its incorporation is straightforward.

We use MLEs for parameters β_0 and $\boldsymbol{\theta}$ based on the simulation output. Following Ankenman et al. [2010], we use a plug-in estimator $\widehat{\mathbf{Q}}_\epsilon$ for the intrinsic precision matrix $\mathbf{Q}_\epsilon$. We use independent simulation across solutions, giving a diagonal intrinsic precision matrix, because CRN leads to a dense intrinsic precision matrix, negating the sparsity of $\bar{\mathbf{Q}}(\boldsymbol{\theta})$ and therefore inhibiting our ability to solve large-scale DOvS problems (but not compromising the validity of the algorithm). Of course, CRN is known to yield sharper comparisons, so we revisit this choice later in our empirical evaluation. In Appendix B we provide details about how the MLEs may be obtained.

3.3 Calculating the Conditional Distribution and CEI

Substituting the plug-in estimate for the intrinsic precision matrix and the MLEs for the parameters of the GMRF into the conditional distribution (6), we obtain the estimated conditional distribution

$$\mathcal{N}\left(\widehat{\beta}_0(\widehat{\boldsymbol{\theta}})\mathbf{1}_{n\times 1} + \bar{\mathbf{Q}}(\widehat{\boldsymbol{\theta}})^{-1}\begin{pmatrix}\vec{\mathbf{0}}_{n_1\times 1}\\ \widehat{\mathbf{Q}}_\epsilon(\bar{\mathcal{Y}}_2 - \boldsymbol{\mu}_2)\end{pmatrix}, \bar{\mathbf{Q}}(\widehat{\boldsymbol{\theta}})^{-1}\right) \tag{8}$$

where

$$\bar{\mathbf{Q}}(\widehat{\boldsymbol{\theta}}) \triangleq \begin{pmatrix}\mathbf{Q}_{11}(\widehat{\boldsymbol{\theta}}) & \mathbf{Q}_{12}(\widehat{\boldsymbol{\theta}})\\ \mathbf{Q}_{12}(\widehat{\boldsymbol{\theta}})^\top & \mathbf{Q}_{22}(\widehat{\boldsymbol{\theta}})\end{pmatrix} + \begin{pmatrix}\mathbf{0}_{n_1\times n_1} & \mathbf{0}_{n_1\times n_2}\\ \mathbf{0}_{n_1\times n_2}^\top & \widehat{\mathbf{Q}}_\epsilon\end{pmatrix}.$$

In this conditional distribution, we ignore the variability that is added from estimating β_0, so the conditional covariance matrix is still $\bar{\mathbf{Q}}(\widehat{\boldsymbol{\theta}})^{-1}$.

The computationally expensive calculation required to compute the conditional distribution is the inversion of the matrix $\bar{\mathbf{Q}}(\widehat{\boldsymbol{\theta}})$. The computational complexity of inverting a dense $n \times n$ matrix is, in general, $O(n^3)$. One goal of sparse matrix techniques is to get this down to something close to $O(n^2)$. Here is one way that this can be achieved in our setting.

A red-black coloring of a graph $\mathcal{G}$ is possible when the nodes $\mathcal{V}$ can be assigned either the color red or black in such a way that no two neighbors, as defined by the edges $\mathcal{E}$, share the same color. A red-black ordering renumbers the nodes with the reds first, then the blacks, to obtain the permuted matrix

$$\begin{pmatrix}\bar{\mathbf{Q}}_{rr}(\widehat{\boldsymbol{\theta}}) & \bar{\mathbf{Q}}_{rb}(\widehat{\boldsymbol{\theta}})\\ \bar{\mathbf{Q}}_{rb}(\widehat{\boldsymbol{\theta}})^\top & \bar{\mathbf{Q}}_{bb}(\widehat{\boldsymbol{\theta}})\end{pmatrix}. \tag{9}$$

Figure 2 illustrates the red-black coloring and ordering for the example in Figure 1. The reordered precision matrices $\bar{\mathbf{Q}}_{rr}(\widehat{\boldsymbol{\theta}})$ and $\bar{\mathbf{Q}}_{bb}(\widehat{\boldsymbol{\theta}})$ will be diagonal, while $\bar{\mathbf{Q}}_{rb}(\widehat{\boldsymbol{\theta}})$ will be sparse. This permuted matrix is very amenable to techniques like sparse Cholesky factorization; for instance, the Matlab® function `chol` adjusts for matrix sparsity [Chen et al., 2008]. Our key insight is that a red-black coloring is always possible for the neighborhood $\mathcal{N}(\mathbf{x})$ in any dimension, which is one reason that we exploit it. Although this particular method depends on the neighborhood $\mathcal{N}(\mathbf{x})$, there are other sparse matrix methods that can be employed for more general neighborhoods; see, for instance, Saad [2003] and Chen [2005].

Conditional means and covariances from the distribution (8) are needed in each iteration of

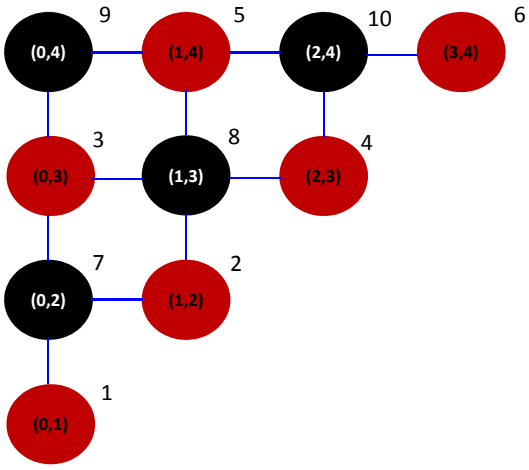


Figure 2: Red-black coloring and relabeling of the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ from Figure 1.

GMIA to calculate the CEIs, and of course $\bar{\mathbf{Q}}(\widehat{\boldsymbol{\theta}})$ changes as more solutions are simulated. Once $\bar{\mathbf{Q}}(\widehat{\boldsymbol{\theta}})^{-1}$ is computed in approximately $O(n^2)$, there are an additional $O(nn_2)$ operations to compute the conditional means, where n_2 is the current number of design points and typically $n_2 \ll n$.

As a basis for comparison, consider using a Gaussian process model that is defined on a continuous domain, such as the stochastic kriging method of Ankenman et al. [2010], but only evaluating it at the integer solutions $\mathbf{x}$. In this case only the covariance matrix of the design points needs to be inverted, which will be dense and therefore require $O(n_2^3)$ operations. However, the computation of the conditional means is $O(nn_2^2)$, and the computation of the conditional covariances between all solutions is $O(n^2 n_2)$. Therefore, we can expect GMIA to be practical for significantly larger problems, and the multiresolution algorithm described in Section 4 extends its reach even further.

3.4 Asymptotic Convergence of GMIA

GMIA exploits the GMRF framework to terminate when the expected improvement from additional searching is small. In this section we show that under very mild conditions GMIA, without any stopping condition, converges with probability 1 to a globally optimal solution. This result does not depend in any way on the validity of the GMRF framework, but only on the following assumptions:

1. $y(\mathbf{x}) > -\infty$ for all $\mathbf{x} \in \mathcal{X}$.

2. $0 < \mathrm{Var}[Y(\mathbf{x})] < +\infty$ for all $\mathbf{x} \in \mathcal{X}$.

3. The initial estimated precision matrix $\mathbf{Q}(\widehat{\boldsymbol{\theta}})$ is positive definite and is not updated thereafter.

In Appendix C we prove the following theorem:

Theorem 1. *The GMIA algorithm without a stopping condition simulates each solution* $\mathbf{x} \in \mathcal{X}$ *infinitely often with probability* 1 *as the number of iterations goes to infinity.*

As a consequence of Theorem 1, the fact that the number of feasible solutions n is finite, and that the estimated optimal solution is the simulated solution with the smallest sample mean, the asymptotic convergence of GMIA follows by the strong law of large numbers.

Theorem 1 does not depend on our chosen neighborhood structure being "correct," nor does it depend on having "good" parameter estimates for the GMRF. In fact, the result will hold with any neighborhood and GMRF parameterization that satisfies Assumption 3. The efficiency of the CEI computations, and the validity of the CEI inference, will be improved by good choices and estimates, however. Further, we can relax Assumption 3 and allow $\widehat{\boldsymbol{\mu}}$ and $\widehat{\boldsymbol{\theta}}$, and therefore $\mathbf{Q}(\widehat{\boldsymbol{\theta}})$, to be updated a finite number of times.

4 The Multiresolution Framework and Algorithm

The core problem in DOvS is balancing exploration and exploitation. Gaussian-process-based optimization obtains this balance by incorporating measures like EI to guide the search. Unfortunately, to fully benefit from this approach we a need to compute an EI measure for every feasible solution on every iteration. There will always be feasible regions $\mathcal{X}$ that are too large for this to be possible. What constitutes "too large" depends on a number of implementation-specific and problem-specific factors.

In this section we show that our GMRF framework lends itself to a powerful multiresolution extension: We first partition $\mathcal{X}$ into m disjoint regions $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_m$, where each region is subset of $\mathcal{X}$ that is connected in the graph $\mathcal{G}$. Then we let the nodes of a region-level GMRF represent a measure of the overall solution quality within each region, with a neighborhood structure again defined by proximity (i.e., adjacent regions). The region-level GMRF provides global guidance by facilitating a CEI comparison among regions. As in the previous section, the quality of individual solutions within a region $\mathcal{P}_j$ is represented by a solution-level GMRF, except that the GMRFs for $\mathcal{P}_j$ and $\mathcal{P}_\ell$ for $j \neq \ell$ are assumed to be independent.

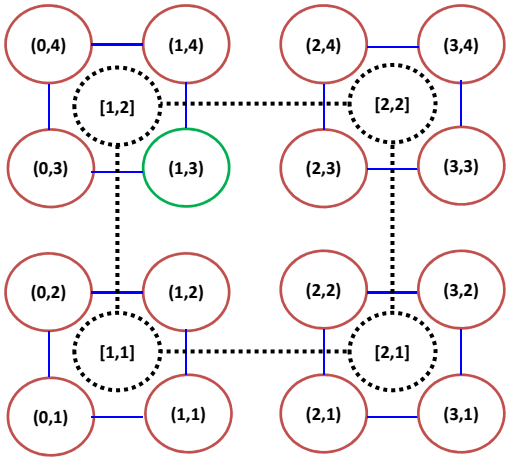


Figure 3: Illustration of a region-level GMRF and four solution-level GMRFs in two dimensions.

In this framework the solutions across regions are connected by the region-level GMRF, rather than by their neighboring individual solutions, and the solutions within a region are connected by their solution-level GMRF. We want the regions to have a large number of feasible solutions, to make the independence approximation reasonable, but not so many that we cannot efficiently calculate the CEI for each individual solution within a region. Further, it is beneficial (but not required) that the regions be of similar shape and contain a similar number of feasible solutions. Figure 3 shows a (very small) region-level and four solution-level GMRFs of a rectangular feasible region for (x_1, x_2). Conceptually, we are not limited to just two resolutions: a GMRF for *regions of regions* could also be defined. Fortunately, region-level and solution-level GMRFs will be sufficient for many problems of practical interest.

In the multiresolution framework the region-level GMRF describing the quality and spatial structure of regions has its own mean and precision matrix, as described below. The solution-level and the region-level GMRFs work together to balance exploration and exploitation, and terminate when the joint CEIs fall below a threshold. The specific multiresolution algorithm described in this section is a first step toward realizing this general framework. For an alternative approach that manages the computational burden by heuristically choosing only a small subset of candidate feasible solutions from $\mathcal{X}$ for which to calculate solution-level CEIs, see Salemi et al. [2014].

The response associated with the region $\mathcal{P}_\ell$ is

$$z(\mathcal{P}_\ell) = |\mathcal{P}_\ell|^{-1} \sum_{\mathbf{x} \in \mathcal{P}_\ell} y(\mathbf{x}).$$

That is, the measure of region-level quality $z(\mathcal{P}_\ell)$ is the average of the objective-function values of all solutions in $\mathcal{P}_\ell$. Similar to the solution-level GMRF, we model $\mathbf{z} = (z(\mathcal{P}_1), z(\mathcal{P}_2), \ldots, z(\mathcal{P}_m))^\top$ as a realization of a region-level GMRF

$$\mathbb{Z} \triangleq (\mathbb{Z}(\mathcal{P}_1), \mathbb{Z}(\mathcal{P}_2), \ldots, \mathbb{Z}(\mathcal{P}_m))^\top \sim \mathcal{N}(\boldsymbol{\eta}, \mathbf{T}(\boldsymbol{\tau})^{-1})$$

with mean vector $\boldsymbol{\eta}$ and precision matrix $\mathbf{T}(\boldsymbol{\tau})$ as a function of parameters $\boldsymbol{\tau}$, which are analogous to $\boldsymbol{\mu}, \mathbf{Q}(\boldsymbol{\theta})$, and $\boldsymbol{\theta}$ at the solution level. Similar to the solution-level GMRF, our main interest is to find the conditional distribution of $\mathbb{Z}$ given observations from design regions.

Let $\boldsymbol{\mathcal{P}} = \{\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_m\}$ denote the set of region-level "feasible solutions." A "design region" $\mathcal{P}_\ell$ is any region for which at least one of its feasible solutions $\mathbf{x}$ has been simulated; that is, there exists $\mathbf{x} \in \mathcal{P}_\ell$ such that $\mathbf{x} \in \Xi_2$. The set of design regions is denoted by Π_2, and $\Pi_1 = \boldsymbol{\mathcal{P}} \backslash \Pi_2$. We can partition $\mathbf{z}, \mathbb{Z}, \boldsymbol{\eta}$, and $\mathbf{T}(\boldsymbol{\tau})$ using Π_1 and Π_2 as we did at the solution level. Within design region $\mathcal{P}_\ell$, let $\Xi_2(\mathcal{P}_\ell) \subseteq \mathcal{P}_\ell$ be the set of feasible solutions in $\mathcal{P}_\ell$ that have been simulated, and let $\Xi_1(\mathcal{P}_\ell) = \mathcal{P}_\ell \backslash \Xi_2(\mathcal{P}_\ell)$. See Appendix D for a complete specification of the region-level GMRF.

For the solution-level GMRFs, we may choose to fit a different set of parameters for each region—$\boldsymbol{\mu}_\ell$ and $\boldsymbol{\theta}_\ell$ for $\mathcal{P}_\ell$—or assume all regions share the same solution-level parameters $\boldsymbol{\mu}$ and $\boldsymbol{\theta}$, which substantially reduces the computation for parameter estimation. We employ the latter approach.

On each iteration of the algorithm below, we select three regions in which to run simulations: (i) design region $\mathcal{P}_{\min}$ that contains the solution with the smallest sample mean among all simulated solutions in $\mathcal{X}$, (ii) design region $\tilde{\mathcal{P}}$ that has the smallest region-level estimator $\bar{\mathbb{Z}}(\mathcal{P}_\ell)$ among $\mathcal{P}_\ell \in \Pi_2$ (see Appendix D), and (iii) region $\mathcal{P}^\star_{\text{CEI}}$ that has the largest region-level CEI among all regions in $\boldsymbol{\mathcal{P}}$. Since it is possible that $\mathcal{P}_{\min}$ is the same as $\tilde{\mathcal{P}}$ or $\mathcal{P}^\star_{\text{CEI}}$, we may only select two regions on some iterations. Notice that $\mathcal{P}^\star_{\text{CEI}}$ may be a region in which no solution has yet been simulated, in which case we simulate k_s initial design points from $\mathcal{P}^\star_{\text{CEI}}$. For $\mathcal{P}_{\min}$ and $\tilde{\mathcal{P}}$, and for $\mathcal{P}^\star_{\text{CEI}}$ if it

already is a design region, we apply the same method as in the GMIA algorithm: at each iteration, we simulate two solutions, the solution with the smallest sample mean and the solution with the largest solution-level CEI in that region.

MR-GMIA

0. Choose k_r regions as design regions. For each design region, choose k_s design points. Simulate r replications for each design point and use the simulation output to calculate the MLEs for the region-level and the solution-level GMRF parameters using the method in Section 3.2 and Appendix B.

1. Let $\mathcal{P}_{\min}$ be the design region that contains the solution with the smallest sample mean among all simulated solutions. Let $\tilde{\mathcal{P}}$ be the design region with the smallest $\bar{\mathcal{Z}}(\mathcal{P}_\ell)$ among $\mathcal{P}_\ell \in \Pi_2$.

2. Estimate the region-level CEI with respect to $\tilde{\mathcal{P}}$ for each region in $\mathcal{P}$. Let $\mathcal{P}^\star_{\text{CEI}}$ be the region with the largest region-level CEI.

3. If $\mathcal{P}^\star_{\text{CEI}}$ is not a design region, then perform Step 3a for $\mathcal{P}^\star_{\text{CEI}}$ and Step 3b for each of $\mathcal{P}_{\min}$ and $\tilde{\mathcal{P}}$. Otherwise, perform Step 3b for each of $\mathcal{P}^\star_{\text{CEI}}, \mathcal{P}_{\min}$ and $\tilde{\mathcal{P}}$.

   (a) Sample k_s design points within the region and simulate r replications at each solution.

   (b) Select two solutions in each region, the solution with the smallest sample mean and the solution with the largest solution-level CEI, and simulate r replications from each solution.

4. Update the solution-level GMRFs of $\mathcal{P}_{\min}, \tilde{\mathcal{P}}$, and $\mathcal{P}^\star_{\text{CEI}}$ using the replications obtained from the selected solutions in Step 3. Update the region-level GMRF.

5. If $\mathcal{P}^\star_{\text{CEI}}$ was not a design region, then go to Step 1. Otherwise, go to Step 6.

6. If the region-level CEIs of all regions calculated in Step 2 as well as the solution-level CEIs of all solutions calculated in Step 3b are less than the threshold δ, then return the solution with the smallest sample mean as the estimated optimal solution. Otherwise, go to Step 1.

5 Numerical Experiments

In this section we apply GMIA and MR-GMIA to two problems to evaluate various aspects of their performance. The first is an inventory optimization problem that exhibits behavior like a practical DOvS problem. This problem is small enough that we can compare GMIA, MR-GMIA and a R&S procedure that exhausts all feasible solutions. The second problem is obtained by adding noise to a difficult response function; we use this problem to study the performance of MR-GMIA as we move from 10,000 to 1,000,000 feasible solutions.

5.1 (s, S) Inventory Problem

The objective function for the (s, S) inventory problem suggested in Koenig and Law [1985] is the expected average cost per period of the inventory system over 30 periods. We assume the demand for inventory in each period is a sequence of i.i.d. Poisson random variables with a common mean of 25. The order and unit shipping costs are \$32 and \$3, respectively. Furthermore, the holding cost is \$1 per unit, per period, and the back-order cost is \$5 per unit, per period. To obtain a simple rectangular feasible region as in Figure 3, we chose our decision variables to be s and $S - s$, where the feasible region is defined by constraints, $1 \leq s \leq 100$ and $1 \leq S - s \leq 100$, which leads to 10,000 feasible solutions. The estimated optimal solution to this DOvS problem is $s = 18$ and $S - s = 35$ with an expected average cost per period of \$106.17 based on 1 million replications of Monte Carlo simulation at each feasible solution. We assume the minimum difference in the objective function that is worth detecting is $\delta = \$1$, which is used for the stopping criterion in our algorithms.

We first ran GMIA on the problem with $k_s = 20$ initial design points in the feasible region selected by Latin hypercube sampling. For each design point, we simulated $r = 10$ replications and computed MLEs from the simulation output. MLEs were calculated only once at the beginning of each run. On each iteration, GMIA selects the current sample-best solution $\tilde{\mathbf{x}}$ based on the sample means at the design points, then calculates the CEI for all other solutions in the feasible region and selects $\mathbf{x}^*_{\text{CEI}}$ as the solution with the largest CEI. If the CEI at $\mathbf{x}^*_{\text{CEI}}$ is smaller than $\delta = \$1$, then the algorithm stops and returns $\tilde{\mathbf{x}}$ as the optimal. Otherwise, it simulates 10 replications each at $\tilde{\mathbf{x}}$ and $\mathbf{x}^*_{\text{CEI}}$, then proceeds to the next iteration. The conditional means and the conditional covariance matrix, both of which are needed to calculate the CEI for every feasible solution, were obtained

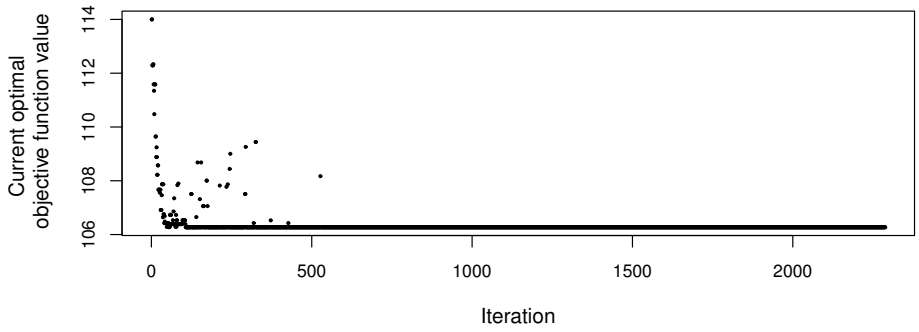


Figure 4: The actual objective function value of the current sample-best solution at each iteration from one run of GMIA for the (s, S) inventory problem.

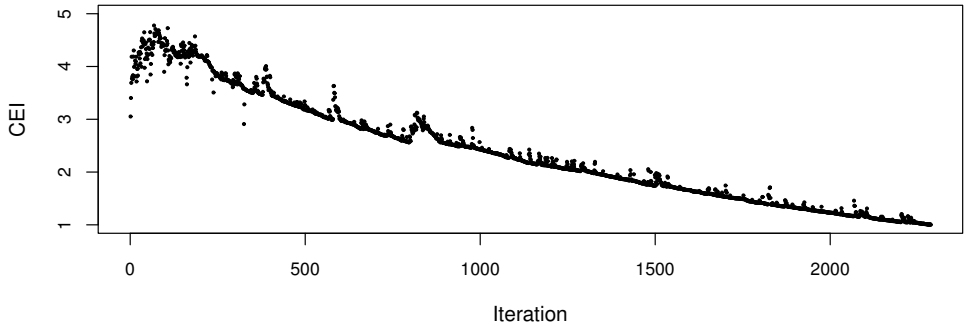


Figure 5: The largest CEI at each iteration from one run of GMIA for the (s, S) inventory problem.

using the Cholesky factorization. Specifically, we calculated $\mathbf{Q}(\boldsymbol{\theta})^{-1}$ by calculating the Cholesky factorization $\mathbf{L}(\boldsymbol{\theta})$ of $\mathbf{Q}(\boldsymbol{\theta})$, a 10,000$\times$10,000 sparse matrix, and then $\mathbf{Q}(\boldsymbol{\theta})^{-1} = (\mathbf{L}(\boldsymbol{\theta})^{\top})^{-1}(\mathbf{L}(\boldsymbol{\theta})^{-1})$.

Figure 4 displays the actual objective function value of the current sample-best solution at each iteration from one run of GMIA until the algorithm stopped at the 2288th iteration. Notice that the current sample-best solution became close to the global minimum within the first 100 iterations, although it fluctuates until the 528th iteration due to the stochastic noise. For instance, at the 527th iteration GMIA chose $(10, 41)$ to be the current sample-best solution based on the sample mean at this solution, although its actual objective function value is \$108.17. However, in the next iteration, the algorithm quickly realized that $(10, 41)$ is unlikely to be the optimal after simulating an additional 10 replications of this solution. The current sample-best solution at the 528th iteration was $(18, 36)$ whose objective function value is \$106.27 and this remained the sample best until the algorithm stopped.

Despite the quick convergence of the current sample-best solution as seen in Figure 4, GMIA

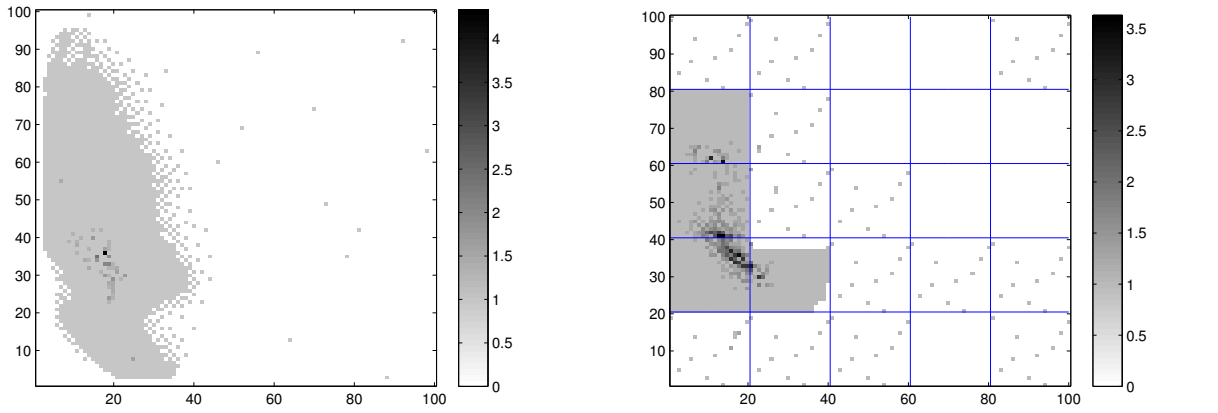
continues its search for many more iterations before meeting the stopping criterion. Figure 5 shows the trajectory of the maximum CEI of the run. The largest CEI slowly decreases over iterations until the 2288th iteration when it becomes less than 1, then the algorithm stops. This shows that even after being very close to the global optimal, GMIA explores and exploits further until it is statistically confident that there is not much gain from additional search. In fact, in all 2288 iterations the solution with the largest CEI at each iteration was a new design point; that is, 2307 solutions were simulated in total, which is 23% of the feasible solutions. This indicates that according to the CEI criterion, it was more beneficial to explore new feasible solutions than to simulate additional replications of the current design points at each iteration.

Figure 6a is a map of simulation replications allocated across the feasible solutions by GMIA for this run. The large light grey area indicates the solutions that were simulated 10 times, i.e., they were selected only once. Points with darker shades were selected as the current sample best for one or more iterations and simulated repeatedly. The darkest point $(18, 36)$ is the solution that GMIA chose on this run. Notice that the design points are clustered with each other except for a few initial design points obtained from Latin hypercube sampling. The algorithm did not simulate any solutions near the upper right and the lower left corners of the feasible region where the objective function increases steeply. This indicates that the estimated GMRF structure guides the algorithm to rule out inferior solutions.

We also ran MR-GMIA on the same problem with region-level neighborhood structure as depicted in Figure 6b. We divided the same feasible solutions into 25 square regions each of which contains 400 feasible solutions numbered bottom to top starting from the lower-left corner. Notice that the true optimal solution $(18, 35)$ is contained in region $\mathcal{P}_2$, while $\mathcal{P}_3$ has the smallest region-level response.

We have two precision matrices: region-level precision matrix $\mathbf{T}(\boldsymbol{\tau})$ and solution-level precision matrix $\mathbf{Q}(\boldsymbol{\theta})$. The former, in this example, is a 25 by 25 matrix and the latter is 400×400, both much smaller than the precision matrix used in the GMIA run. As Cholesky decomposition of a sparse $n \times n$ matrix is $O(n^2)$, the smaller dimensions of the precision matrices significantly reduces the computation time involved in both MLE and CEI calculations.

We chose $k_r = 5$ initial design regions by Latin hypercube sampling. To select the design points within each region, we used the same Latin hypercube sample of size $k_s = 10$ for all design regions;



(a) GMIA

(b) MR-GMIA: Regions are numbered 1–25 bottom to top from the lower left corner.

Figure 6: Allocations of simulation replications at feasible solutions for the (s, S) inventory problem in base-10 log scale.

for this reason some of the regions in Figure 6b show the same light grey pattern. From each selected design point we simulated 10 replications and used the result to compute the MLEs. As in the GMIA runs, MLEs were computed only once at the beginning of the algorithm and not updated thereafter. At each iteration, MR-GMIA chooses the region $\tilde{\mathcal{P}}$ that has the smallest region-level sample mean and computes the region-level CEI of all 25 regions. Also at each iteration, the algorithm simulates solutions from three regions: $\tilde{\mathcal{P}}$, region $\mathcal{P}_{\min}$ that contains the current sample-best solution, and region $\mathcal{P}^{\star}_{\text{CEI}}$ that has the largest region-level CEI. The solution-level CEIs are calculated for all solutions within a design region, if the region was simulated previously. The algorithm stops if the largest region-level CEI and the largest solution-level CEIs of all three regions drop below $\delta = \$1$.

Figure 6b shows that MR-GMIA allocated most of the simulation replications within $\mathcal{P}_2, \mathcal{P}_3, \mathcal{P}_4$ and $\mathcal{P}_7$. The surrounding regions of these four regions were selected once or a few more times before the algorithm stopped. After the initial sampling, MR-GMIA quickly finds the region that is likely to contain the global optimal solution using the region-level CEI. Notice that $\mathcal{P}_{25}$ at the upper right corner whose response $z(\mathcal{P}_{25})$ is the largest among all regions was simulated only once and never selected again. Also, the neighboring regions of $\mathcal{P}_{25}$ were not simulated. This shows that the region-level GMRF informed the algorithm correctly that the regions close to $\mathcal{P}_{25}$ are not likely to contain the optimal solution. Compared to the simulation replication allocation of GMIA in

Figure 6a, the biggest difference is that MR-GMIA first chooses regions to simulate then selects the solutions within those regions; therefore, sampling is restricted within the boundaries of selected regions. This, in fact, helps MR-GMIA concentrate on the regions that are likely to contain the global optimal solution. Since each region is much smaller than the entire feasible region, it is easier to visit all solutions within a selected region. If the algorithm does not stop after visiting all solutions within the region at least once, the solution-level CEI criterion makes the algorithm revisit the design points that are likely to be the global optimal. From Figure 6b we can confirm that the ellipsoidal area that includes the global optimal is simulated more frequently than other parts of the feasible region.

Figure 7 displays the actual objective function value at the current sample-best solution at each iteration from this run of MR-GMIA. While the trajectory is very similar to that of GMIA in Figure 4, this MR-GMIA run stopped at an earlier iteration (767th) than the GMIA run above (2288th iteration). If at each iteration the algorithm had simulated a new design point in each of $\tilde{\mathcal{P}}, \mathcal{P}_{\min}$, and $\mathcal{P}^{\star}_{\text{CEI}}$ by the solution-level CEI criterion, then the number of simulated solutions would have been roughly $767 \times 3 = 2301$. However, the number of simulated solutions in this run was actually 1631, 16% of the entire feasible solutions, which confirms that the solution-level CEI criterion did not always select a new design point. Figure 8 displays the largest region-level CEI and the largest solution-level CEIs in $\tilde{\mathcal{P}}, \mathcal{P}_{\min}$, and $\mathcal{P}^{\star}_{\text{CEI}}$ at each iteration. The region-level CEI drops below 1 after the 8th iteration, which indicates that the algorithm is quite certain that there is not much gain from exploring a new region. The largest solution-level CEIs in $\tilde{\mathcal{P}}$ and $\mathcal{P}_{\min}$ drop below 1 by the 385th iteration. MR-GMIA simulated almost all solutions within $\tilde{\mathcal{P}}$ and $\mathcal{P}_{\min}$ so that there is not much improvement gained by simulating additional replications. What keeps the algorithm from stopping is the largest solution-level CEI in $\mathcal{P}^{\star}_{\text{CEI}}$, which shows a decreasing trend while displaying frequent jumps before finally dropping below 1. Some of the big jumps are caused when the algorithm selects a new region as $\mathcal{P}^{\star}_{\text{CEI}}$. When the algorithm concluded that there was not much improvement to expect from simulating additional replications from $\mathcal{P}^{\star}_{\text{CEI}}$, it stopped.

As a benchmark to evaluate the relative efficiency and the quality of inference provided by our algorithms, we applied the KN R&S procedure [Kim and Nelson, 2001] to the same (s, S) problem. KN is an efficient, fully sequential, eliminating procedure that simulates all feasible solutions. When we apply KN we use CRN, set the number of first-stage replications of each feasible solution to

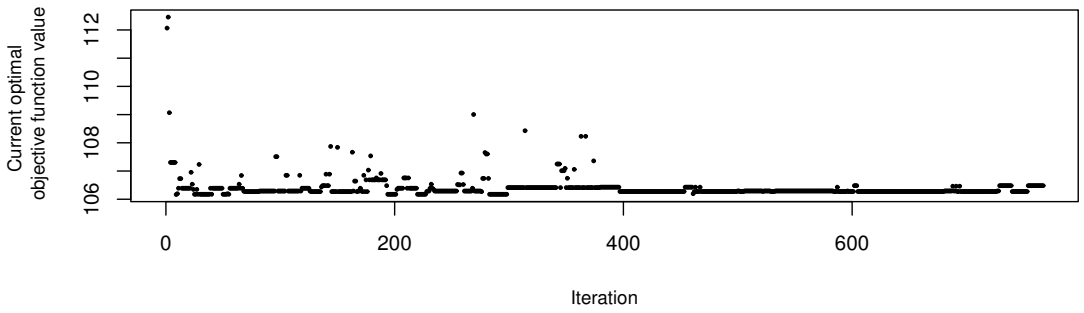


Figure 7: The actual objective function value of the current sample-best solution at each iteration from one run of MR-GMIA for the (s, S) inventory problem.

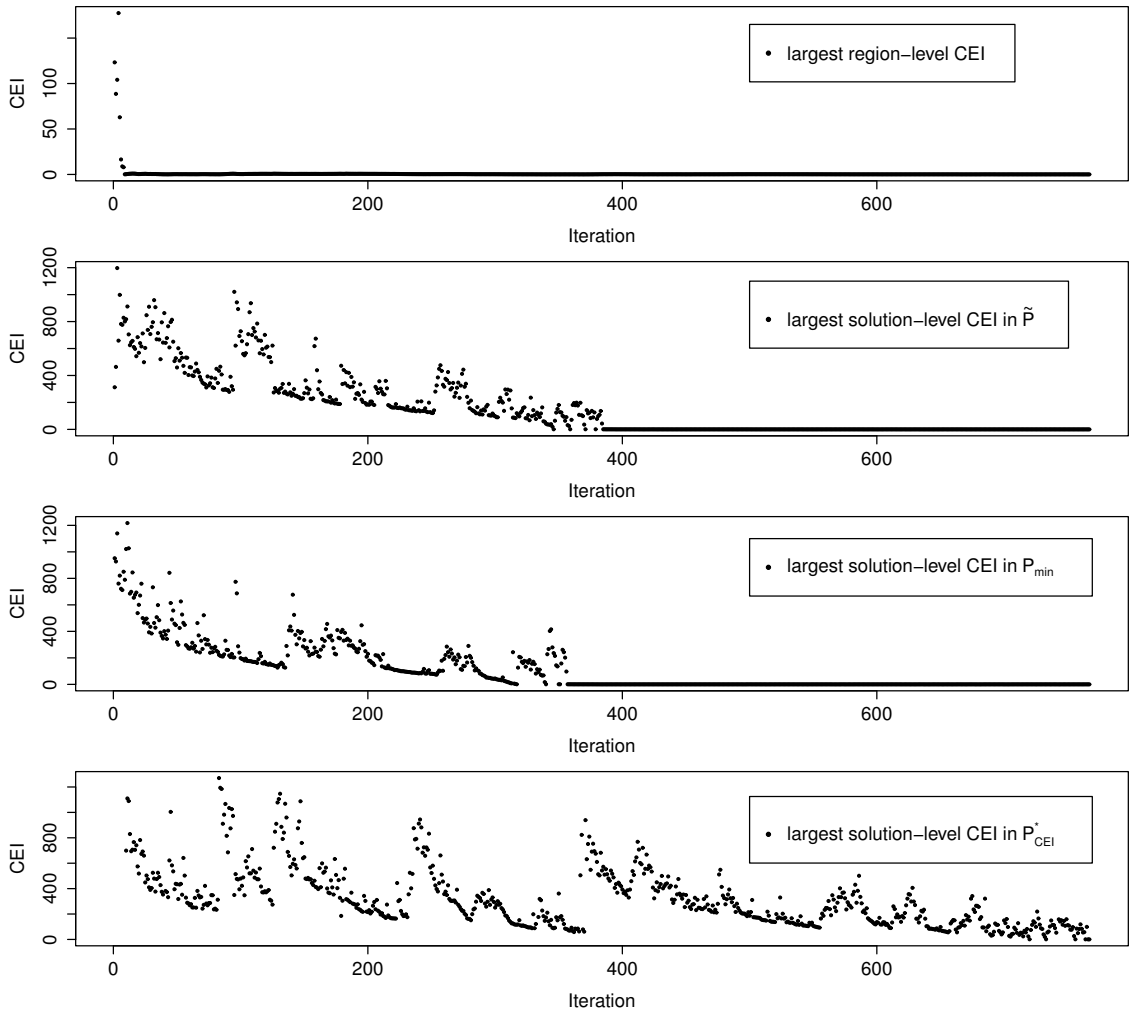
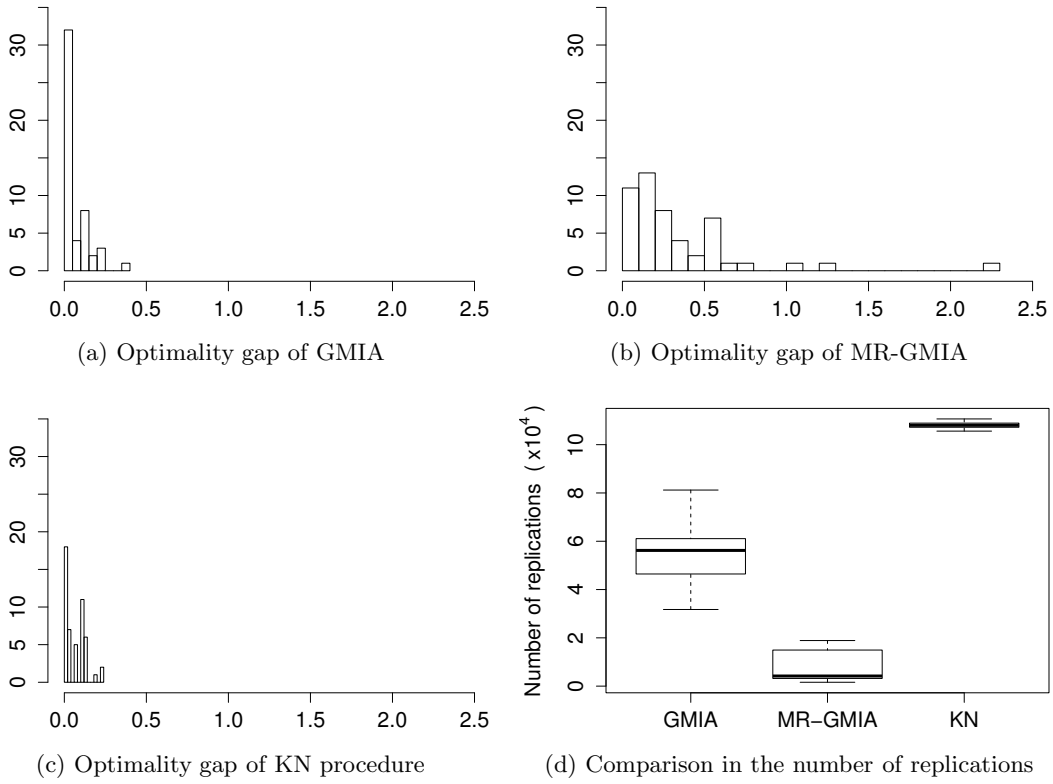


Figure 8: The largest region-level CEI and the largest solution-level CEIs in $\widetilde{\mathcal{P}}, \mathcal{P}_{\min}$, and $\mathcal{P}^{\star}_{\mathrm{CEI}}$ at each iteration from one run of MR-GMIA for the (s, S) inventory problem.



(a) Optimality gap of GMIA

(b) Optimality gap of MR-GMIA

(c) Optimality gap of KN procedure

(d) Comparison in the number of replications

Figure 9: (a), (b), (c) Histograms of the difference between the global optimal objective value and objective values returned by 50 runs of each method. (d) Boxplots of number of replications required for 50 runs of each method.

$n_0 = 10$, and use $\alpha = 0.05$ and $\delta = \$1$. The role of δ in the GMIA algorithms can be compared to the role of δ in the KN procedure: KN guarantees that the probability of selecting a feasible solution whose objective function value is within δ of the optimal is at least $1 - \alpha$. Thus, δ specifies the set of feasible solutions that are acceptable for the algorithm to select as optimal; we are indifferent to which feasible solution in this set is chosen. For the GMIA algorithms, we stop searching for better feasible solutions when the CEI of every feasible solution is less than δ, which happens well before every feasible solution is simulated.

Figure 9 compares the quality of the optimal solutions and the number of replications spent in 50 runs of GMIA, MR-GMIA and KN. Notice that the objective function gap between the solutions selected by both GMIA and KN and the optimal solution are much less than $\delta = 1$ in all 50 runs, whereas 3 runs of MR-GMIA have gaps larger than 1. The important insight is that GMIA and MR-GMIA have performance comparable to KN while expending significantly less effort, simulating

a fraction of the feasible solutions and *not* using CRN. Notice that the number of replications spent by KN is more than 10 times larger than that of MR-GMIA on average.

Although the solutions obtained from GMIA are closer to the optimal solution, there is an advantage of using MR-GMIA when we compare the number of replications. Also, the dimension of the precision matrices of both algorithms affect the run-time of the algorithms. For the GMIA runs, the Cholesky decomposition of the 10,000 × 10,000 precision matrix, the additional multiplications to invert it, and computing the conditional mean of all feasible solutions, took 13.61 (standard error = 0.23), 78.00 (0.92) and 0.000069 (0.000002) seconds, respectively. For the MR-GMIA runs, the same calculation times for the region-level GMRF took 0.000026 (0.000002), 0.000053 (0.000003), and 0.00012 (0.00008) seconds, respectively, and those for the solution-level GMRF took 0.0024 (0.0001), 0.0090 (0.0002), and 0.00063 (0.00003) seconds, respectively. Even after considering that MR-GMIA performs the solution-level matrix calculations for up to three regions at each iteration, the computation time of MR-GMIA is far less than that of GMIA.

5.2 Griewank Function

The Griewank function (`http://www.sfu.ca/~ssurjano/griewank.html`) has many local minima, which makes it a challenging test function for optimization algorithms. Figure 10 shows the two-dimensional Griewank function on the domain $[-5, 5] \times [-5, 5]$. Within this domain the range of the function is $[0, 2.0044]$, and the global minimum is at $(0, 0)$. Notice that there are four local minima near the four corners of the domain and their response values are 0.0086.

To create a DOvS problem, we take as feasible solutions the 100×100 lattice on $[-5, 5] \times [-5, 5]$ where the response at each solution is given by the Griewank function in Figure 10. Normally distributed simulation noise with mean 0 and variance σ^2 is added to the response function to make it stochastic. As depicted in Figure 11a, we divided the feasible solutions into 25 square regions, each of which contains 400 solutions. Notice that this is the same number of solutions/regions that we had when we applied MR-GMIA to the (s, S) inventory problem. To analyze the impact of δ on the performance of the algorithm, we tested three values of δ: $0.005, 0.01$, and 0.02. We chose these values relative to 0.0086, the difference between the responses at the local minima and the global minimum, so that we can test the effect of δ on the solution quality of the algorithm. We adjusted the standard error of the sample mean at each design point when it is first visited to be

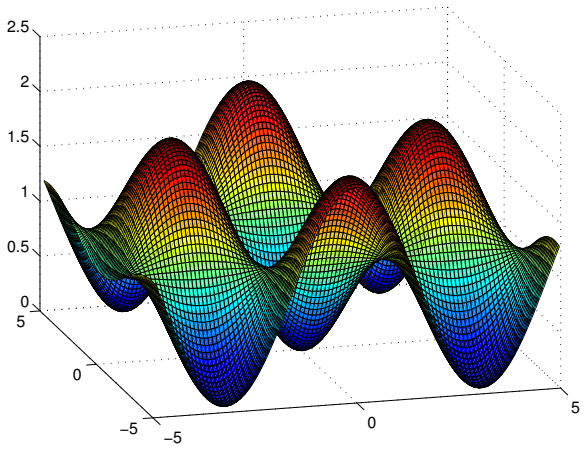


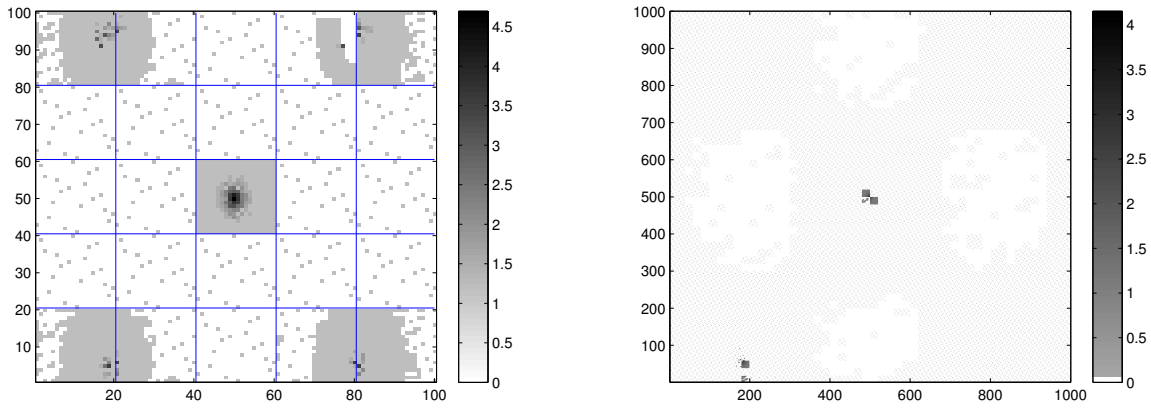
Figure 10: Two-dimensional Griewank function on $[-5, 5] \times [-5, 5]$.

relative to δ. Given r, the number of replications at each solution, and δ, we tested three values of $\sigma/\sqrt{r}$: $\delta/2, \delta$, and 2δ.

For each run, we used a Latin hypercube sample of size 5 to select the initial design regions. Any time a solution is selected we simulated $r = 16$ replications.

Figure 11a shows how simulation replications were allocated across the feasible region in one run of MR-GMIA. Notice that all regions were visited at least once, unlike the (s, S) inventory problem in Figure 6b. Compared to the (s, S) inventory problem, the response surface of the Griewank function is harder to predict as the region-level response $z(\mathcal{P}_\ell)$ for all 25 regions do not vary much. Therefore, the region-level CEI did not rule out any region before it was visited at least once. However, once the regions were visited, the algorithm correctly found the regions that either contain the global and local minima or are close to them and concentrated the simulation effort there. Notice that the algorithm simulated solutions in the ellipsoidal area that contains the global optimal solution most frequently.

Table 1 presents the results of 50 runs of MR-GMIA with the chosen δ and σ values. Common random numbers were adopted for each case of (δ, σ) in Table 1; that is, each (δ, σ) combination experienced the same source of randomness. Notice that our algorithm returned the global optimal solution in all 50 runs of all settings except for $(\delta, \sigma) = (0.02, 0.08)$, and $(0.02, 0.16)$. Even in these two cases, the average response at the solution is within δ from 0. As expected, both the number of simulated solutions and the number of replications spent tend to increase as δ decreases and as



(a) 100×100 lattice.

(b) $1{,}000 \times 1{,}000$ lattice.

Figure 11: Allocations of simulation replications at feasible solutions of the Griewank function in Figure 10 in base-10 log scale.

σ increases. However, MR-GMIA is not very sensitive to the choice of δ for smaller σ.

To test the impact of the number of feasible solutions on the performance of the algorithm, we ran MR-GMIA with $(\delta, \sigma) = (0.005, 0.01)$ but a finer lattice, as depicted in Figure 11b: a 1000×1000 grid instead of 100×100, which results in 1,000,000 feasible solutions. These solutions are divided into 2500 regions, each of which contains 400 feasible solutions. Figure 11b is a map of replications spent in the feasible region in one run of MR-GMIA. Notice that the areas in white, which indicate that they were never simulated, correspond to the peaks of the Griewank function in Figure 10. Unlike in the 100×100 case, the region-level responses actually differ in this case as we have a finer grid. The region-level CEI effectively rules out the regions that are not likely to contain the optimal solution using the region-level GMRF structure. Most of the simulation effort is concentrated in the regions near the global minimum and a local minimum in the lower-left corner.

The average response at the optimal solutions returned by 50 runs of MR-GMIA is 0.0002 (standard error = 0.00004) and the maximum response was 0.0016, which is smaller than $\delta = 0.005$. The average number of simulated solutions is 24,696 (1307), which is only 2.5% of the 1 million feasible solutions. Compared to the 100×100 case, which simulated 19% of the feasible solutions on average, we can confirm that the algorithm works very efficiently even if we have a large number of feasible solutions. The average number of replications spent is 89,079 (2775), which is less than

Table 1: Results from 50 runs of MR-GMIA applied to the Griewank functions with different δ and σ. Standard errors are provided in the parentheses.

δ	0.005			0.01			0.02		
σ	0.01	0.02	0.04	0.02	0.04	0.08	0.04	0.08	0.16
Average response at optimal solution	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0002 (0.0001)	0.0015 (0.0006)
Max response at optimal solution	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0025	0.0219
Average number of simulated solutions	1,927 (26)	1,945 (25)	2065 (27)	1,945 (25)	2065 (27)	2444 (29)	2065 (27)	2509 (28)	2837 (81)
Average number of replications	108,955 (2417)	111,229 (2377)	122,036 (2877)	111,229 (2377)	121,974 (2872)	155,254 (2742)	121,974 (2872)	157,207 (3007)	184,147 (6,634)

8 times larger than that of the 100×100 case in Table 1. These results show the multiresolution algorithm scales well as the number of feasible solutions increases.

In this section we used the Griewank function to create a difficult DOvS problem on a lattice. However, the quality of our results suggest that MR-GMIA could be applied to discretized versions of continuous-variable OvS problems to exploit the benefits of sparse matrix computations.

6 Summary

In this paper we constructed a DOvS algorithm for problems with integer-ordered decision variables based on GMRFs. Unlike GRFs with continuous domains, which are defined by dense covariance matrices that lead to slow and ill-conditioned numerical computations for large-scale problems, GMRFs are defined by their precision matrices and can be constructed to be sparse. For problems with very large numbers of solutions, we created a multiresolution GMRF-based algorithm. To allocate simulation effort in either case, we introduced an expected improvement criterion that incorporates the uncertainty in stochastic simulation by treating the current sample-best solution as a random variable.

Acknowledgments

This article is based upon work supported by the National Science Foundation under Grant No. CMMI-0900354. We would also like to thank Håvard Rue for helping with the conditional distri-

bution. Portions of this work were previously published in Salemi et al. [2014].

References

T. Amemiya. *Advanced Econometrics*. Harvard University Press, Cambridge, MA, 1985.

B. Ankenman, B. L. Nelson, and J. Staum. Stochastic kriging for simulation metamodeling. *Operations Research*, 58(2):371–382, 2010.

K. Chen. *Matrix Preconditioning Techniques and Applications*. Cambridge University Press, Cambridge, UK, 2005.

Y. Chen, T. A. Davis, and W. W. Hager. Algorithm 887: CHOLMOD, supernodal sparse cholesky factorization and update/downdate. *ACM Transactions on Mathematical Software*, 35(3):1–14, 2008.

W.G. Cochran. *Sampling Techniques*. Wiley, New York, 1977.

P. J. Diggle, R. Menezes, and T. Su. Geostatistical inference under preferential sampling. *Journal of the Royal Statistical Society: Series C*, 59(2):191–232, 2010.

P. Frazier. Tutorial: Optimization via simulation with Bayesian statistics and dynamic programming. In C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A. M. Uhrmacher, editors, *Proceedings of the 2012 Winter Simulation Conference*, pages 1–16, Piscataway, NJ, 2012. IEEE.

P. Frazier, W. Powell, and S. Dayanik. The knowledge-gradient policy for correlated normal beliefs. *INFORMS Journal on Computing*, 21(4):599–613, 2009.

P. I. Frazier, J. Xie, and S. E. Chick. Value of information methods for pairwise sampling with correlations. In S. Jain, R. R. Creasey, J. Himmelspach, K. P. White, and M. Fu, editors, *Proceedings of the 2011 Winter Simulation Conference*, pages 3979–3991, Piscataway, NJ, 2011. IEEE.

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

D. R. Jones, M. Schonlau, and W. J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, 1998.

S. Kim. Statistical ranking and selection. In *Encyclopedia of Operations Research and Management Science*, pages 1459–1469. Springer, New York, 2013.

S. Kim and B. L. Nelson. A fully sequential procedure for indifference-zone selection in simulation. *ACM Transactions on Modeling and Computer Simulation*, 11(3):251–273, 2001.

L. W. Koenig and A. M. Law. A procedure for selecting a subset of size m containing the l best of k independent normal populations, with applications to simulation. *Communications in Statistics*, B14(3):719–734, 1985.

K. S. Miller. On the inverse of the sum of matrices. *Mathematics Magazine*, 54(2):67–72, 1981.

B. L. Nelson. Optimization via simulation over discrete decision variables. *TutORials in Operations Research*, 7:193–207, 2010.

N. Quan, J. Yin, S. H. Ng, and L. H. Lee. Simulation optimization via kriging: A sequential search using expected improvement with computing budget constraints. *IIE Transactions*, 45(7): 763–780, 2013.

H. Rue and L. Held. *Gaussian Markov Random Fields: Theory and Applications*. Chapman and Hall/CRC, New York, 2005.

Y. Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, second edition, 2003.

P. Salemi, J. Staum, and B. L. Nelson. Generalized integrated Brownian fields for simulation metamodeling. In R. Pasupathy, S.-H. Kim, A. Tolk, R. Hill, and M. E. Kuhl, editors, *Proceedings of the 2013 Winter Simulation Conference*, pages 543–554, Piscataway, NJ, 2013. IEEE.

P. Salemi, B. L. Nelson, and J. Staum. Discrete optimization via simulation using Gaussian Markov random fields. In A. Tolk, S. Y. Diallo, L. Yilmaz I. O. Ryzhov, S. Buckley, and J. A. Miller, editors, *Proceedings of the 2014 Winter Simulation Conference*, pages 3809–3820, Piscataway, NJ, 2014. IEEE.

T. J. Santner, B. J. Williams, and W. Notz. *The Design and Analysis of Computer Experiments*. Springer, New York, 2003.

L. Sun, L. J. Hong, and Z. Hu. Optimization via simulation using Gaussian process-based search. In S. Jain, R. R. Creasey, J. Himmelspach, K. P. White, and M. Fu, editors, *Proceedings of the 2011 Winter Simulation Conference*, pages 4139–4150, Piscataway, NJ, 2011. IEEE.

B. J. Williams, T. J. Santner, and W. I. Notz. Sequential design of computer experiments to minimize integrated response functions. *Statistica Sinica*, 10:1133–1152, 2000.

J. Xu. Efficient discrete optimization via simulation using stochastic kriging. In C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A. M. Uhrmacher, editors, *Proceedings of the 2012 Winter Simulation Conference*, pages 466–477, Piscataway, NJ, 2012. IEEE.

Appendix

Results in this appendix are intended for an Online Supplement.

A Derivation of the Conditional Distribution

In this section we derive of the conditional distribution of $(\mathbb{Y}_1, \mathbb{Y}_2)$ given observed $\mathbb{Y}_2^\epsilon$ in (6). We do this by first deriving the joint distribution of $(\mathbb{Y}_1, \mathbb{Y}_2, \mathbb{Y}_2^\epsilon)$ and then applying Lemma 2.1 of Rue and Held [2005] to obtain the conditional distribution. To simplify the notation we suppress the dependence of $\mathbf{Q}$ on $\boldsymbol{\theta}$.

Notice that $\mathbb{Y}_1$ and $\mathbb{Y}_2^\epsilon$ are conditionally independent, given $\mathbb{Y}_2$, because $\mathbb{Y}$ is a GMRF and is also independent of the intrinsic noise. From Theorem 2.5 in Rue and Held [2005], the conditional distribution of $\mathbb{Y}_1$ given $\mathbb{Y}_2 = \mathbf{y}_2$ is

$$\mathcal{N}\left(\boldsymbol{\mu}_1 - \mathbf{Q}_{11}^{-1}\mathbf{Q}_{12}(\mathbf{y}_2 - \boldsymbol{\mu}_2), \mathbf{Q}_{11}^{-1}\right).$$

From our assumption about the simulation output process, the conditional distribution of $\mathbb{Y}_2^\epsilon$ given $\mathbb{Y}_2 = \mathbf{y}_2$ is $\mathcal{N}\left(\mathbf{y}_2, \mathbf{Q}_\epsilon^{-1}\right)$. Further,

$$\mathbb{Y}_2 \sim \mathcal{N}\left(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_{22}\right) = \mathcal{N}\left(\boldsymbol{\mu}_2, [\mathbf{Q}_{22} - \mathbf{Q}_{21}\mathbf{Q}_{11}^{-1}\mathbf{Q}_{12}]^{-1}\right)$$

using standard results for the inverse of a partitioned matrix. Therefore, the joint distribution $f(\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_2^\epsilon)$ satisfies

$$\begin{aligned} f(\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_2^\epsilon) \quad \propto \quad & \exp\left\{-\frac{1}{2}\left[\mathbf{y}_1 - (\boldsymbol{\mu}_1 - \mathbf{Q}_{11}^{-1}\mathbf{Q}_{12}(\mathbf{y}_2 - \boldsymbol{\mu}_2))\right]^\top \mathbf{Q}_{11}\left[\mathbf{y}_1 - (\boldsymbol{\mu}_1 - \mathbf{Q}_{11}^{-1}\mathbf{Q}_{12}(\mathbf{y}_2 - \boldsymbol{\mu}_2))\right]\right\} \\ & \times \exp\left\{-\frac{1}{2}\left(\mathbf{y}_2 - \boldsymbol{\mu}_2\right)^\top \left(\mathbf{Q}_{22} - \mathbf{Q}_{21}\mathbf{Q}_{11}^{-1}\mathbf{Q}_{12}\right)\left(\mathbf{y}_2 - \boldsymbol{\mu}_2\right)\right\} \\ & \times \exp\left\{-\frac{1}{2}\left(\mathbf{y}_2^\epsilon - \mathbf{y}_2\right)^\top \mathbf{Q}_\epsilon \left(\mathbf{y}_2^\epsilon - \mathbf{y}_2\right)\right\}. \end{aligned} \tag{10}$$

A significant quantity of matrix algebra leads to

$$f(\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_2^\epsilon) \propto \exp\left\{ -\frac{1}{2} \begin{pmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \mathbf{y}_2^\epsilon \end{pmatrix}^\top \begin{pmatrix} \mathbf{Q}_{11} & \mathbf{Q}_{12} & \mathbf{0} \\ \mathbf{Q}_{21} & \mathbf{Q}_{22} + \mathbf{Q}_\epsilon & -\mathbf{Q}_\epsilon \\ \mathbf{0} & -\mathbf{Q}_\epsilon & \mathbf{Q}_\epsilon \end{pmatrix} \begin{pmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \mathbf{y}_2^\epsilon \end{pmatrix} + \begin{pmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \mathbf{y}_2^\epsilon \end{pmatrix}^\top \begin{pmatrix} \mathbf{Q}_{11}\boldsymbol{\mu}_1 + \mathbf{Q}_{12}\boldsymbol{\mu}_2 \\ \mathbf{Q}_{21}\boldsymbol{\mu}_1 + \mathbf{Q}_{22}\boldsymbol{\mu}_2 \\ \mathbf{0} \end{pmatrix} \right\}. \tag{11}$$

Rue and Held [2005] refer to (11) as being in canonical form, for which their Lemma 2.1 can be used to show that the conditional distribution of $(\mathbb{Y}_1, \mathbb{Y}_2)$ given $\mathbb{Y}_2^\epsilon$ is as in (6).

B MLE for the GMRF Parameters

Denote the design points by $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{n_2}$, and let

$$S^2(\mathbf{x}) = \frac{1}{r(\mathbf{x})} \sum_{j=1}^{r(\mathbf{x})} \left(Y_j(\mathbf{x}) - \bar{Y}(\mathbf{x}) \right)^2,$$

where $r(\mathbf{x})$ is the cumulative number of replications that have been allocated to design point $\mathbf{x}$, $Y_j(\mathbf{x})$ is the simulation output at $\mathbf{x}$ on the jth replication, and $\bar{Y}(\mathbf{x})$ is the sample mean of the simulation output across the $r(\mathbf{x})$ replications at design point $\mathbf{x}$. The plug-in estimator for the intrinsic precision matrix is

$$\widehat{\mathbf{Q}}_\epsilon = \operatorname{diag}\left(\frac{r(\mathbf{x}_1)}{S^2(\mathbf{x}_1)}, \frac{r(\mathbf{x}_2)}{S^2(\mathbf{x}_2)}, \ldots, \frac{r(\mathbf{x}_{n_2})}{S^2(\mathbf{x}_{n_2})} \right).$$

Remark: Under our assumptions, $S^2(\mathbf{x}) > 0$ with probability 1 once the number of simulation replications $r(\mathbf{x})$ is large enough. However, if the output $Y_j(\mathbf{x})$ has a discrete distribution (e.g., Bernoulli), then it is possible that $S^2(\mathbf{x}) = 0$ for finite $r(\mathbf{x})$, which makes the estimated intrinsic precision matrix undefined. To resolve this technical issue, we can modify the definition of $S^2(\mathbf{x})$

to be

$$S^2(\mathbf{x}) = \frac{1}{r(\mathbf{x})} \sum_{j=1}^{r(\mathbf{x})} \left(Y_j(\mathbf{x}) - \bar{Y}(\mathbf{x})\right)^2 + \frac{\gamma}{r(\mathbf{x})}$$

for small constant $\gamma > 0$. Therefore, $S^2(\mathbf{x})$ is still a consistent estimator of the variance as $r(\mathbf{x}) \to \infty$ and $S^2(\mathbf{x}) > 0$ for finite $r(\mathbf{x})$.

After substituting $\widehat{\mathbf{Q}}_\epsilon$ for $\mathbf{Q}_\epsilon$, we integrate the conditional distribution over $\mathbf{y}_2$ to obtain the unconditional distribution

$$\mathbb{Y}_2^\epsilon \sim \mathcal{N}(\beta_0 \mathbf{1}_{n_2 \times 1}, \boldsymbol{\Sigma}_{22}(\boldsymbol{\theta}) + \widehat{\mathbf{Q}}_\epsilon^{-1}),$$

where $\boldsymbol{\Sigma}_{22}(\boldsymbol{\theta}) \triangleq (\mathbf{Q}_{22}(\boldsymbol{\theta}) - \mathbf{Q}_{12}(\boldsymbol{\theta})^\top \mathbf{Q}_{11}(\boldsymbol{\theta})^{-1} \mathbf{Q}_{12}(\boldsymbol{\theta}))^{-1}$, and $\mathbf{1}_{n_2 \times 1}$ is the $n_2 \times 1$ vector of ones. Disregarding terms that do not depend on β_0 or $\boldsymbol{\theta}$, the log-likelihood function given the simulation output is

$$\begin{aligned}\mathcal{L}(\beta_0, \boldsymbol{\theta} | \bar{\mathbf{y}}_2) \triangleq & \frac{1}{2} \log |(\boldsymbol{\Sigma}_{22}(\boldsymbol{\theta}) + \widehat{\mathbf{Q}}_\epsilon^{-1})^{-1}| \\ & - \frac{1}{2} (\bar{\mathbf{y}}_2 - \beta_0 \mathbf{1}_{n_2 \times 1})^\top (\boldsymbol{\Sigma}_{22}(\boldsymbol{\theta}) + \widehat{\mathbf{Q}}_\epsilon^{-1})^{-1} (\bar{\mathbf{y}}_2 - \beta_0 \mathbf{1}_{n_2 \times 1}).\end{aligned}$$

Miller [1981] provides a method for efficiently calculating $(\boldsymbol{\Sigma}_{22}(\boldsymbol{\theta}) + \widehat{\mathbf{Q}}_\epsilon^{-1})^{-1}$ using only $\boldsymbol{\Sigma}_{22}(\boldsymbol{\theta})^{-1}$ and matrix-vector products with the columns of $\widehat{\mathbf{Q}}_\epsilon^{-1}$. Furthermore, we can calculate $\boldsymbol{\Sigma}_{22}(\boldsymbol{\theta})^{-1} = \mathbf{Q}_{22}(\boldsymbol{\theta}) - \mathbf{Q}_{12}(\boldsymbol{\theta})^\top \mathbf{Q}_{11}(\boldsymbol{\theta})^{-1} \mathbf{Q}_{12}(\boldsymbol{\theta})$ by solving the sparse linear system $\mathbf{Q}_{11}(\boldsymbol{\theta})\mathbf{X} = \mathbf{Q}_{12}(\boldsymbol{\theta})$ and substituting $\mathbf{X}$ for $\mathbf{Q}_{11}(\boldsymbol{\theta})^{-1}\mathbf{Q}_{12}(\boldsymbol{\theta})$. This sparse linear system can be solved using techniques like those cited in Section 3.3. Fixing the parameters in $\boldsymbol{\theta}$, the MLE for β_0 is

$$\widehat{\beta}_0(\boldsymbol{\theta}) = (\mathbf{1}_{n_2 \times 1}^\top (\boldsymbol{\Sigma}_{22}(\boldsymbol{\theta}) + \widehat{\mathbf{Q}}_\epsilon^{-1})^{-1} \mathbf{1}_{n_2 \times 1})^{-1} \mathbf{1}_{n_2 \times 1}^\top (\boldsymbol{\Sigma}_{22}(\boldsymbol{\theta}) + \widehat{\mathbf{Q}}_\epsilon^{-1})^{-1} \bar{\mathbf{y}}_2,$$

where we have written $\widehat{\beta}_0$ as a function of $\boldsymbol{\theta}$ to explicitly show dependence. Substituting the MLE $\widehat{\beta}_0(\boldsymbol{\theta})$ into $\mathcal{L}(\beta_0, \boldsymbol{\theta} | \bar{\mathbf{y}}_2)$, we get the profile log-likelihood function

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta} | \bar{\mathbf{y}}_2) \triangleq & \frac{1}{2} \log |(\boldsymbol{\Sigma}_{22}(\boldsymbol{\theta}) + \widehat{\mathbf{Q}}_\epsilon^{-1})^{-1}| \\ & - \frac{1}{2} (\bar{\mathbf{y}}_2 - \widehat{\beta}_0(\boldsymbol{\theta}) \mathbf{1}_{n_2 \times 1})^\top (\boldsymbol{\Sigma}_{22}(\boldsymbol{\theta}) + \widehat{\mathbf{Q}}_\epsilon^{-1})^{-1} (\bar{\mathbf{y}}_2 - \widehat{\beta}_0(\boldsymbol{\theta}) \mathbf{1}_{n_2 \times 1}),\end{aligned}$$

which only depends on $\boldsymbol{\theta}$. The MLE $\widehat{\boldsymbol{\theta}}$ for $\boldsymbol{\theta}$ is the maximizer of the profile log-likelihood function

$\mathcal{L}(\boldsymbol{\theta}|\bar{\mathcal{Y}}_2)$ over the possible values of $\boldsymbol{\theta}$, i.e.,

$$\widehat{\boldsymbol{\theta}} \triangleq \arg\max_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \mathcal{L}(\boldsymbol{\theta}|\bar{\mathcal{Y}}_2)$$

where $\boldsymbol{\Theta}$ is the set of values of $\boldsymbol{\theta}$ with $\theta_0 > 0$ and $0 \leq \theta_i \leq 1, i = 1, 2, \ldots, d$, and for which $\mathbf{Q}(\boldsymbol{\theta})$ is positive-definite. The restricted space for the correlation parameters $0 \leq \theta_i \leq 1, i = 1, 2, \ldots, d$, makes the optimization easier than working with the corresponding parameters of typical parametric correlation functions that can take on any nonnegative value (see, for instance Santner et al. [2003]). The following proposition can be used to help determine which values of $\boldsymbol{\theta}$ result in a positive-definite precision matrix.

Proposition B.1. *Let $\mathbf{Q}(\boldsymbol{\theta})$ be the precision matrix corresponding to the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Furthermore, let $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$ be a subgraph of $\mathcal{G}$ with the corresponding precision matrix $\mathbf{Q}'(\boldsymbol{\theta})$. If the precision matrix $\mathbf{Q}'(\boldsymbol{\theta})$ is not positive-definite, then the precision matrix $\mathbf{Q}(\boldsymbol{\theta})$ is not positive-definite.*

Proof. To prove the proposition, we use proof by contrapositive. Indeed, we prove that if the precision matrix $\mathbf{Q}(\boldsymbol{\theta})$ is positive-definite, then the precision matrix $\mathbf{Q}'(\boldsymbol{\theta})$ is positive-definite.

Assume that the precision matrix $\mathbf{Q}(\boldsymbol{\theta})$ is positive-definite. Since $\mathbf{Q}(\boldsymbol{\theta})$ is positive-definite, the corresponding graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ represents a GMRF. By Theorem 2.5 in Rue and Held [2005] the subgraph $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$ also represents a GMRF whose positive-definite precision matrix is $\mathbf{Q}'(\boldsymbol{\theta})$. Thus, we have the result. □

This proposition is the GMRF interpretation of the property that every principal submatrix of a positive-definite matrix is positive-definite, and can be used to eliminate precision matrices by checking whether a submatrix of the precision matrix is not positive-definite. For example, consider a feasible region that is an $M \times M$ grid. If one wanted to check whether $\boldsymbol{\theta}$ yields a positive-definite precision matrix for the $M \times M$ grid, then one could check whether $\boldsymbol{\theta}$ yields a positive-definite precision matrix for an $N \times N$ grid, with $N \ll M$. In our experiments, we have found that the set of $\boldsymbol{\theta}$ for which the precision matrix of an $M \times M$ grid is positive-definite is nearly the same as the set of $\boldsymbol{\theta}$ for which the precision matrix of an $N \times N$ grid is positive-definite.

C Global Convergence Proof

In this appendix we provide the proofs of several lemmas, Theorem 1, and Corollary 1 which extends the global convergence result to the multiresolution algorithm MR-GMIA. In either case we consider GMIA or MR-GMIA with the stopping condition suppressed.

Recall that we use $\mathbf{y}$ to denote the true mean of all solutions $y(\mathbf{x})$ with $\mathbf{x} \in \mathcal{X}$. Our goal is to find $\mathbf{x} \in \mathcal{X}_{\min} = \{\mathbf{x} \in \mathcal{X} \colon y(\mathbf{x}) = y_{\min}\}$, where $y_{\min} = \min_{\mathbf{x}\in\mathcal{X}} y(\mathbf{x})$. We allow $|\mathcal{X}_{\min}| \geq 1$.

In the following we use s to denote the stage/iteration of the algorithm and $\bar{\mathbf{Q}}_s$ and $\boldsymbol{\mu}_s$ to denote the estimated conditional precision matrix and mean in (8), respectively, at stage s. Both $\bar{\mathbf{Q}}_s$ and $\boldsymbol{\mu}_s$ are functions of the cumulative simulation output up to stage s. We drop the dependence of these terms on $\widehat{\boldsymbol{\theta}}$ for notational convenience.

Also, let $\mathbf{Q}_{\epsilon,s}$ denote the $n_2 \times n_2$ estimated intrinsic precision matrix at stage s, whose diagonal element corresponding to design point $\mathbf{x}$ is $r_s(\mathbf{x})/S_s^2(\mathbf{x})$, where $r_s(\mathbf{x})$ is the cumulative number of replications obtained at solution $\mathbf{x}$ up to stage s, and $S_s^2(\mathbf{x})$ is the corresponding sample variance (as defined in Appendix B if we are concerned about discrete outputs).

The vector of sample means obtained from design points in Ξ_2 up to stage s is denoted by $\bar{\mathcal{Y}}_{2,s}$. The design point corresponding to the smallest sample mean at stage s is denoted by $\tilde{\mathbf{x}}_s$. We define $\mathrm{CEI}_s(\mathbf{x})$ to be the CEI of $\mathbf{x} \in \mathcal{X}$ at stage s with respect to $\tilde{\mathbf{x}}_s$.

Finally, when we need to represent the dependence of a random quantity on a particular sample path ω of the simulation, we include ω as an argument; e.g., $\mathrm{CEI}_s(\mathbf{x}, \omega)$.

As stated in Section 3.4, we assume:

1. $y(\mathbf{x}) > -\infty$ for all $\mathbf{x} \in \mathcal{X}$.
2. $0 < \mathrm{Var}[Y(\mathbf{x})] < +\infty$ for all $\mathbf{x} \in \mathcal{X}$.
3. The initial estimated precision matrix $\mathbf{Q}$ is positive definite and is not updated thereafter.

Since we enforce Assumption 3 in GMIA, it is a mild condition. Notice that while $\mathbf{Q}$ is fixed by fixing $\widehat{\boldsymbol{\theta}}$, the contents of the sets Ξ_1 and Ξ_2 change, as does $\widehat{\mathbf{Q}}_{\epsilon,s}$ and therefore $\bar{\mathbf{Q}}_s$, as the algorithm progresses. However, if $\mathbf{Q}$ and $\widehat{\mathbf{Q}}_{\epsilon,s}$ are positive definite, then so is $\bar{\mathbf{Q}}_s$.

We first prove the following lemmas. Lemma 1 shows what would happen to the conditional distribution if a set of solutions Ξ_1 was never simulated.

Lemma 1. *Suppose that* Ξ_1 *and* Ξ_2 *are fixed sets of solutions and* $\Xi_1 \cup \Xi_2 = \mathcal{X}$*. Then if* $r_s(\mathbf{x}) \to \infty$ *as* $s \to \infty$ *for all* $\mathbf{x}$ *in* Ξ_2*, and* $r_s(\mathbf{x}) = 0$ *for all* s *and for all* $\mathbf{x}$ *in* Ξ_1*, then*

$$\bar{\mathbf{Q}}_s^{-1} \to \begin{pmatrix} \mathbf{Q}_{11}^{-1} & \mathbf{0}_{n_1 \times n_2} \\ \mathbf{0}_{n_2 \times n_1} & \mathbf{0}_{n_2 \times n_2} \end{pmatrix} \text{ and } \boldsymbol{\mu}_s \to \begin{pmatrix} \boldsymbol{\mu}_1 - \mathbf{Q}_{11}^{-1}\mathbf{Q}_{12}(\mathbf{y}_2 - \boldsymbol{\mu}_2) \\ \mathbf{y_2} \end{pmatrix}$$

with probability 1*.*

Proof. The conditional covariance matrix at stage s is

$$\bar{\mathbf{Q}}_s^{-1} = \begin{pmatrix} (\mathbf{Q}_{11} - \mathbf{Q}_{12}(\mathbf{Q}_{22} + \mathbf{Q}_{\epsilon,s})^{-1}\mathbf{Q}_{12}^\top)^{-1} & -\mathbf{Q}_{11}^{-1}\mathbf{Q}_{12}(\mathbf{Q}_{22} + \mathbf{Q}_{\epsilon,s} - \mathbf{Q}_{12}^\top\mathbf{Q}_{11}^{-1}\mathbf{Q}_{12})^{-1} \\ -((\mathbf{Q}_{11}^{-1}\mathbf{Q}_{12}(\mathbf{Q}_{22} + \mathbf{Q}_{\epsilon,s} - \mathbf{Q}_{12}^\top\mathbf{Q}_{11}^{-1}\mathbf{Q}_{12})^{-1})^\top & (\mathbf{Q}_{22} + \mathbf{Q}_{\epsilon,s} - \mathbf{Q}_{12}^\top\mathbf{Q}_{11}^{-1}\mathbf{Q}_{12})^{-1} \end{pmatrix}.$$

Under Assumption 3, $\mathbf{Q}_{11}, \mathbf{Q}_{12}$ and $\mathbf{Q}_{22}$ are fixed once initialized, and $\mathbf{Q}_{11}$ and $\mathbf{Q}_{22}$ are positive definite. Since the diagonal elements of $\mathbf{Q}_{\epsilon,s}$ diverge to ∞ with probability 1 as $s \to \infty$, $(\mathbf{Q}_{22} + \mathbf{Q}_{\epsilon,s})^{-1} \to \mathbf{0}_{n_2 \times n_2}$ and $(\mathbf{Q}_{22} + \mathbf{Q}_{\epsilon,s} - \mathbf{Q}_{12}^\top\mathbf{Q}_{11}^{-1}\mathbf{Q}_{12})^{-1} \to \mathbf{0}_{n_2 \times n_2}$ with probability 1 by Slutsky's theorem [Amemiya, 1985]. Thus,

$$\bar{\mathbf{Q}}_s^{-1} \to \begin{pmatrix} \mathbf{Q}_{11}^{-1} & \mathbf{0}_{n_1 \times n_2} \\ \mathbf{0}_{n_2 \times n_1} & \mathbf{0}_{n_2 \times n_2} \end{pmatrix} \text{ with probability 1.}$$

The conditional sample mean at stage s is

$$\boldsymbol{\mu}_s = \boldsymbol{\mu} + \begin{pmatrix} -\mathbf{Q}_{11}^{-1}\mathbf{Q}_{12}(\mathbf{Q}_{22} + \mathbf{Q}_{\epsilon,s} - \mathbf{Q}_{12}^\top\mathbf{Q}_{11}^{-1}\mathbf{Q}_{12})^{-1}\mathbf{Q}_{\epsilon,s}(\bar{\mathcal{Y}}_{2,s} - \boldsymbol{\mu}_2) \\ (\mathbf{Q}_{22} + \mathbf{Q}_{\epsilon,s} - \mathbf{Q}_{12}^\top\mathbf{Q}_{11}^{-1}\mathbf{Q}_{12})^{-1}\mathbf{Q}_{\epsilon,s}(\bar{\mathcal{Y}}_{2,s} - \boldsymbol{\mu}_2) \end{pmatrix}.$$

Note that as $s \to \infty$, $(\mathbf{Q}_{22} + \mathbf{Q}_{\epsilon,s} - \mathbf{Q}_{12}^\top\mathbf{Q}_{11}^{-1}\mathbf{Q}_{12})^{-1}\mathbf{Q}_{\epsilon,s} \to \mathbf{I}_{n_2}$ with probability 1, where $\mathbf{I}_{n_2}$ is the $n_2 \times n_2$ identity matrix. Then since $\bar{\mathcal{Y}}_{2,s} \to \mathbf{y}_2$ with probability 1,

$$\boldsymbol{\mu}_s \to \begin{pmatrix} \boldsymbol{\mu}_1 - \mathbf{Q}_{11}^{-1}\mathbf{Q}_{12}(\mathbf{y}_2 - \boldsymbol{\mu}_2) \\ \mathbf{y}_2 \end{pmatrix} \text{ with probability 1.}$$

□

Lemma 2. *If for all* $\mathbf{x} \in \mathcal{X}$, $r_s(\mathbf{x}) \to \infty$ *as* $s \to \infty$, *then*

(i) $\boldsymbol{\mu}_s \to \mathbf{y}$ *with probability 1.*

(ii) $\mu_s(\tilde{\mathbf{x}}_s) \to y_{\min}$ *with probability 1.*

(iii) Results (i) and (ii) hold if $\mathbf{x}$ *is restricted to a subset* $\mathcal{B} \subset \mathcal{X}$, *and* $\mathbf{y}$ *and* $y_{\min}$ *are replaced by* $\mathbf{y}_{\mathcal{B}}$ *and* $y_{\min,\mathcal{B}}$ *defined just over the subset* $\mathcal{B}$.

Proof. From Lemma 1, it follows immediately that $\boldsymbol{\mu}_s \to \mathbf{y}$ with probability 1 by taking $\Xi_1 = \emptyset$.

For (ii), we first show that $y(\tilde{\mathbf{x}}_s) \to y_{\min}$ with probability 1. If $y(\mathbf{x}) = y_{\min}$ for all $\mathbf{x} \in \mathcal{X}$, then it is trivially true that $\mu_s(\tilde{\mathbf{x}}_s) \to y_{\min}$ with probability 1. Otherwise, let

$$\varepsilon = \min\{y(\mathbf{x})\colon \mathbf{x} \in \mathcal{X}\backslash\mathbf{X}_{\min}\} - y_{\min}.$$

Notice that we can find such $\varepsilon > 0$, as we have a finite number of solutions in $\mathcal{X}$. Since we simulate every solution infinitely often, for almost all sample paths ω there exists $s^\star(\mathbf{x}, \omega)$ such that $|\bar{\mathcal{Y}}_s(\mathbf{x}, \omega) - y(\mathbf{x})| < \varepsilon/2$ for all $s > s^\star(\mathbf{x}, \omega)$ by the strong law of large numbers. Let $s^\star(\omega) = \max_{\mathbf{x}\in\mathcal{X}} s^\star(\mathbf{x}, \omega)$. Then, for all $s > s^\star(\omega)$, $y(\tilde{\mathbf{x}}_s) = y_{\min}$. Thus, $y(\tilde{\mathbf{x}}_s) \to y_{\min}$ with probability 1 as $s \to \infty$. Combined with $\mu_s(\tilde{\mathbf{x}}_s) \to y(\tilde{\mathbf{x}}_s)$ with probability 1 from (i), we conclude (ii). Part (iii) follows immediately.

□

To prove Theorem 1, we suppose that there is a solution, say $\mathbf{x}' \in \mathcal{X}$, that is only simulated finitely often by GMIA. For this to be the case, $\mathbf{x}'$ must neither be the solution with the smallest sample mean nor have the largest CEI beyond some finite stage. We then show that in the limit this outcome has probability 0 when we apply the GMIA algorithm in Section 3.1.

Proof. For each non-empty subset of solutions $\mathcal{A} \in 2^{\mathcal{X}}\backslash\emptyset$, let $\mathcal{E}_{\mathcal{A}}$ be the collection of sample paths such that all solutions in $\mathcal{A}$ are simulated finitely often, while all solutions in $\mathcal{X}\backslash\mathcal{A}$ are simulated infinitely often, as $s \to \infty$ in GMIA. Then,

$$\Pr\{\exists \mathbf{x} \in \mathcal{X}\colon \mathbf{x} \text{ is simulated finitely often}\} \leq \sum_{\mathcal{A}\in 2^{\mathcal{X}}\backslash\emptyset} \Pr\{\mathcal{E}_{\mathcal{A}}\}.$$

Hence, it suffices to show that for all $\mathcal{A} \in 2^{\mathcal{X}} \backslash \emptyset$, the $\Pr\{\mathcal{E}_{\mathcal{A}}\} = 0$.

To set up a contradiction, suppose there exists $\mathcal{A} \in 2^{\mathcal{X}} \backslash \emptyset$ such that $\Pr\{\mathcal{E}_{\mathcal{A}}\} > 0$. Permute precision matrix $\mathbf{Q}$ so that

$$\mathbf{Q} = \begin{pmatrix} \mathbf{Q}_{\mathcal{A},\mathcal{A}} & \mathbf{Q}_{\mathcal{A},\mathcal{X}\backslash\mathcal{A}} \\ \mathbf{Q}_{\mathcal{X}\backslash\mathcal{A},\mathcal{A}} & \mathbf{Q}_{\mathcal{X}\backslash\mathcal{A},\mathcal{X}\backslash\mathcal{A}} \end{pmatrix},$$

where $\mathbf{Q}_{\mathcal{A},\mathcal{A}}$ is the block matrix of $\mathbf{Q}$ that corresponds to solutions in $\mathcal{A}$, etc. Then for each sample path $\omega \in \mathcal{E}_{\mathcal{A}}$, we can find finite $s^\star(\mathbf{x}, \omega)$ for each $\mathbf{x} \in \mathcal{A}$ such that $\mathbf{x}$ is never simulated again after stage $s^\star(\mathbf{x}, \omega)$. Let $s^\star(\omega) = \max_{\mathbf{x}\in\mathcal{A}} s^\star(\mathbf{x}, \omega)$. Therefore, given ω, each solution in $\mathcal{A}$ is not simulated for all $s > s^\star(\omega)$.

Recall that at each stage s, the GMIA algorithm simulates (i) the solution with the smallest sample mean, $\tilde{\mathbf{x}}_s$, and (ii) the solution with the largest CEI with respect to $\tilde{\mathbf{x}}_s$. Therefore, for each $\omega \in \mathcal{E}_{\mathcal{A}}$, (i) $\tilde{\mathbf{x}}_s(\omega)$ is not in $\mathcal{A}$ for all $s > s^\star(\omega)$, and (ii) the largest CEI solution with respect to $\tilde{\mathbf{x}}_s(\omega)$ is not in $\mathcal{A}$, or equivalently, there exists $\mathbf{x}'_s \in \mathcal{X}\backslash\mathcal{A}$ such that for each $\mathbf{x} \in \mathcal{A}, \mathrm{CEI}_s(\mathbf{x}, \omega) \leq \mathrm{CEI}_s(\mathbf{x}'_s, \omega)$ for all $s > s^\star(\omega)$.

Therefore, if the GMRF had been initialized by precision matrix $\bar{\mathbf{Q}}_{s^\star(\omega)}$ and prior mean $\widehat{\boldsymbol{\mu}}_{s^\star(\omega)}$ at stage $s = s^\star(\omega)$ along the same sample path, then no solutions in $\mathcal{A}$ would ever be simulated for all $s > s^\star(\omega)$, i.e., $r_s(\mathbf{x}) = r_{s^\star(\omega)}(\mathbf{x})$ for all $s > s^\star(\omega)$. Let $\bar{\mathbf{Q}}^\star_{11}(\omega)$ denote the block matrix of $\bar{\mathbf{Q}}_{s^\star(\omega)}$ corresponding to the solutions in $\mathcal{A}$ and $\mathbf{Q}^{11}_{\epsilon,s^\star(\omega)}$ denote the submatrix of $\mathbf{Q}_{\epsilon,s^\star(\omega)}$ corresponding to solutions in $\mathcal{A}$, then $\bar{\mathbf{Q}}^\star_{11}(\omega) = \mathbf{Q}_{11} + \mathbf{Q}^{11}_{\epsilon,s^\star(\omega)}$. Conditional on $\mathcal{E}_{\mathcal{A}}$, we drop ω from $s^\star(\omega), \bar{\mathbf{Q}}_{s^\star(\omega)}, \bar{\mathbf{Q}}^\star_{11}(\omega)$, and $\mathbf{Q}^{11}_{\epsilon,s^\star}(\omega)$ to denote random quantities.

Now, define the following events:

$$\mathcal{E}_1 = \left\{ \lim_{s\to\infty} \bar{\mathbf{Q}}_s^{-1} = \begin{pmatrix} \mathbf{A}^{-1} & \mathbf{0}_{|\mathcal{A}|\times|\mathcal{X}\backslash\mathcal{A}|} \\ \mathbf{0}_{|\mathcal{X}\backslash\mathcal{A}|\times|\mathcal{A}|} & \mathbf{0}_{|\mathcal{X}\backslash\mathcal{A}|\times|\mathcal{X}\backslash\mathcal{A}|} \end{pmatrix} \text{ for some matrix } \mathbf{A} \text{ such that } \right.$$
$$\left. \forall \mathbf{x} \in \mathcal{A}, \mathbf{A}^{-1}(\mathbf{x}, \mathbf{x}) > 0, \text{ and } \forall \mathbf{x} \in \mathcal{A}, \lim_{s\to\infty} (\mu_s(\tilde{\mathbf{x}}_s) - \mu_s(\mathbf{x})) > -\infty \right\},$$
$$\mathcal{E}_2 = \left\{ \forall \mathbf{x} \in \mathcal{X}\backslash\mathcal{A}, \lim_{s\to\infty} (\mu_s(\tilde{\mathbf{x}}_s) - \mu_s(\mathbf{x})) \begin{cases} = 0, & \text{if } y(\mathbf{x}) = \min_{\mathbf{x}'\in\mathcal{X}\backslash\mathcal{A}} y(\mathbf{x}') \\ < 0, & \text{otherwise} \end{cases} \right\}.$$

Note that $\mathbf{A}$ in $\mathcal{E}_1$ is a random matrix that depends on the sample path $\omega \in \mathcal{E}_1$. Let us first show $\Pr\{\mathcal{E}_1 | \mathcal{E}_{\mathcal{A}}\} = 1$. Conditional on $\mathcal{E}_{\mathcal{A}}$, if we pretend the GMRF is initialized at $s = s^\star$ with $\bar{\mathbf{Q}}_{s^\star}$, then

no solution in $\mathcal{A}$ is simulated thereafter, whereas $r_s(x) \to \infty$ for all $\mathbf{x} \in \mathcal{X}\backslash\mathcal{A}$. Hence, by Lemma 1

$$\Pr\left\{ \lim_{s\to\infty} \bar{\mathbf{Q}}_s^{-1} = \begin{pmatrix} \bar{\mathbf{Q}}_{11}^{\star\,-1} & \mathbf{0}_{|\mathcal{A}|\times|\mathcal{X}\backslash\mathcal{A}|} \\ \mathbf{0}_{|\mathcal{X}\backslash\mathcal{A}|\times|\mathcal{A}|} & \mathbf{0}_{|\mathcal{X}\backslash\mathcal{A}|\times|\mathcal{X}\backslash\mathcal{A}|} \end{pmatrix} \text{ and } \forall \mathbf{x} \in \mathcal{A},\, \lim_{s\to\infty} (\mu_s(\tilde{\mathbf{x}}_s) - \mu_s(\mathbf{x})) > -\infty \,\middle|\, \mathcal{E}_{\mathcal{A}} \right\} = 1.$$

Conditional on $\mathcal{E}_{\mathcal{A}}$, let $S^2_{s^\star}(\mathbf{x})$ be a sample variance from $r_{s^\star}(\mathbf{x})$ replications at $\mathbf{x}$. Then,

$$\Pr\left\{ \forall \mathbf{x} \in \mathcal{A}, r_{s^\star}(\mathbf{x})/S^2_{s^\star}(\mathbf{x}) < \infty | \mathcal{E}_{\mathcal{A}} \right\} = 1. \tag{12}$$

Note that $r_{s^\star}(\mathbf{x})/S^2_{s^\star}(\mathbf{x})$ is the diagonal element of $\mathbf{Q}^{11}_{\epsilon,s^\star}$ corresponding to $\mathbf{x} \in \mathcal{A}$. Hence, $\bar{\mathbf{Q}}^\star_{11} = \mathbf{Q}_{11} + \mathbf{Q}^{11}_{\epsilon,s^\star}$ is positive definite with probability 1, since we guarantee $\mathbf{Q}_{11}$ to be positive definite and $\mathbf{Q}^{11}_{\epsilon,s^\star}$ is positive definite with probability 1 from (12). Therefore, $\Pr\{\mathcal{E}_1|\mathcal{E}_{\mathcal{A}}\} = 1$. Also, due to the sampling rule of GMIA, for each $\omega \in \mathcal{E}_{\mathcal{A}}$, $\tilde{\mathbf{x}}_s \in \mathcal{X}\backslash\mathcal{A}$ for all $s > s^\star(\omega)$. Thus, Lemma 2 applies to $\mathcal{X}\backslash\mathcal{A}$. Therefore,

$$\Pr\left\{ \lim_{s\to\infty} \mu_s(\mathbf{x}) = y(\mathbf{x}), \forall \mathbf{x} \in \mathcal{X}\backslash\mathcal{A}, \text{ and } \lim_{s\to\infty} \mu_s(\tilde{\mathbf{x}}_s) = \min_{\mathbf{x}' \in \mathcal{X}\backslash\mathcal{A}} y(\mathbf{x}') | \mathcal{E}_{\mathcal{A}} \right\} = 1$$

which implies $\Pr\{\mathcal{E}_2|\mathcal{E}_{\mathcal{A}}\} = 1$.

Recall that $\mathrm{CEI}_s(\mathbf{x})$ is

$$\mathrm{CEI}_s(\mathbf{x}) = (\mu_s(\tilde{\mathbf{x}}_s) - \mu_s(\mathbf{x}))\Phi\left(\frac{\mu_s(\tilde{\mathbf{x}}_s) - \mu_s(\mathbf{x})}{\sigma_s(\tilde{\mathbf{x}}_s, \mathbf{x})}\right) + \sigma_s(\tilde{\mathbf{x}}_s, \mathbf{x})\phi\left(\frac{\mu_s(\tilde{\mathbf{x}}_s) - \mu_s(\mathbf{x})}{\sigma_s(\tilde{\mathbf{x}}_s, \mathbf{x})}\right), \tag{13}$$

where $\sigma^2_s(\tilde{\mathbf{x}}_s, \mathbf{x})$ is the posterior variance of $\mathbb{Y}(\tilde{\mathbf{x}}_s) - \mathbb{Y}(\mathbf{x})$ at stage s as defined in Section 2.3. Recall that for each $\omega \in \mathcal{E}_{\mathcal{A}}$, $\tilde{\mathbf{x}}_s \in \mathcal{X}\backslash\mathcal{A}$ for all $s > s^\star(\omega)$. Therefore, $\mathcal{E}_1 \cap \mathcal{E}_{\mathcal{A}}$ implies $\lim_{s\to\infty} \sigma^2(\tilde{\mathbf{x}}_s) = 0$, which again implies that for each $\mathbf{x} \in \mathcal{A}$, $\lim_{s\to\infty} \sigma^2_s(\tilde{\mathbf{x}}_s, \mathbf{x}) = \lim_{s\to\infty} \sigma^2_s(\mathbf{x})$. From the definition of $\mathcal{E}_1$, $\lim_{s\to\infty} \sigma^2_s(\mathbf{x}) = A^{-1}(\mathbf{x}, \mathbf{x})$, thus $\lim_{s\to\infty} \sigma^2_s(\mathbf{x}) > 0$. Also, $\mathcal{E}_1 \cap \mathcal{E}_{\mathcal{A}}$ implies $\lim_{s\to\infty}(\mu_s(\tilde{\mathbf{x}}_s) - \mu_s(\mathbf{x})) > -\infty$. Because (13) is a non-negative increasing function in $\mu_s(\tilde{\mathbf{x}}_s) - \mu_s(\mathbf{x})$, and $\mathrm{CEI}_s(\mathbf{x}) \to 0$ as $\mu_s(\tilde{\mathbf{x}}_s) - \mu_s(\mathbf{x}) \to -\infty$ given $\lim_{s\to\infty} \sigma_s(\tilde{\mathbf{x}}_s, \mathbf{x}) > 0$, $\mathcal{E}_1 \cap \mathcal{E}_{\mathcal{A}}$ guarantees $\lim_{s\to\infty} \mathrm{CEI}_s(\mathbf{x}) > 0$ for $\mathbf{x} \in \mathcal{A}$. For $\mathbf{x} \in \mathcal{X}\backslash\mathcal{A}$, $\mathcal{E}_1 \cap \mathcal{E}_{\mathcal{A}}$ implies $\sigma_s(\tilde{\mathbf{x}}_s, \mathbf{x}) \to 0$ and $\mathcal{E}_2 \cap \mathcal{E}_{\mathcal{A}}$ implies $\lim_{s\to\infty}(\mu_s(\tilde{\mathbf{x}}_s) - \mu_s(\mathbf{x})) \leq 0$, therefore, $\mathrm{CEI}_s(\mathbf{x}) \to 0$.

To summarize, conditional on $\mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{E}_{\mathcal{A}}$, the following holds with probability 1:

$\lim_{s\to\infty} \mathrm{CEI}_s(\mathbf{x}) > 0$ for all $\mathbf{x} \in \mathcal{A}$ and $\lim_{s\to\infty} \mathrm{CEI}_s(\mathbf{x}) = 0$ for all $\mathbf{x} \in \mathcal{X}\backslash\mathcal{A}$. Therefore, we must have $\Pr\{\mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{E}_{\mathcal{A}}\} = 0$ for $\mathcal{A}$ to exist according to the sampling rule in the GMIA algorithm. However,

$$\begin{aligned}\Pr\{\mathcal{E}_{\mathcal{A}}\backslash(\mathcal{E}_1 \cap \mathcal{E}_2)\} &= \Pr\{(\mathcal{E}_{\mathcal{A}}\backslash\mathcal{E}_1) \cup (\mathcal{E}_{\mathcal{A}}\backslash\mathcal{E}_2)\} \\ &\leq \Pr\{\mathcal{E}_{\mathcal{A}}\backslash\mathcal{E}_1\} + \Pr\{\mathcal{E}_{\mathcal{A}}\backslash\mathcal{E}_2\} \\ &= 0.\end{aligned}$$

Notice that the last equality follows from $\Pr\{\mathcal{E}_1|\mathcal{E}_{\mathcal{A}}\} = \Pr\{\mathcal{E}_2|\mathcal{E}_{\mathcal{A}}\} = 1$. Thus,

$$\Pr\{\mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{E}_{\mathcal{A}}\} + \Pr\{\mathcal{E}_{\mathcal{A}}\backslash(\mathcal{E}_1 \cap \mathcal{E}_2)\} = \Pr\{\mathcal{E}_{\mathcal{A}}\} = 0,$$

which contradicts the assumption $\Pr\{\mathcal{E}_{\mathcal{A}}\} > 0$. Therefore, we conclude that $\Pr\{\mathcal{E}_{\mathcal{A}}\} = 0$ for any $\mathcal{A} \in 2^{\mathcal{X}}\backslash\emptyset$. □

Together with Lemma 1, Theorem 1 guarantees the global convergence of the GMIA algorithm to the true optimal value, $y_{\min}$, regardless of the problem structure.

Remark: We can relax Assumption 3 and allow $\widehat{\boldsymbol{\mu}}$ and $\widehat{\boldsymbol{\theta}}$, and therefore $\mathbf{Q}$, to be updated c times for some finite constant c. Then it is easy to show that Lemmas 1–2, and Theorem 1, still hold.

The following corollary establishes the global convergence of our multiresolution GMRF algorithm.

Corollary 1. *The MR-GMIA algorithm without a stopping criterion simulates each solution* $\mathbf{x} \in \mathcal{X}$ *infinitely often with probability* 1 *as the number of iterations goes to infinity.*

The proof of Corollary 1 can be obtained by applying the same method in Theorem 1 to the region-level GMRF. The region-level version of Lemmas 1–2, and Theorem 1, guarantee that each region is simulated infinitely often as $s \to \infty$, with probability 1. Within each region, the rule is the same as in the GMIA algorithm. Therefore, provided that each region is simulated infinitely often, all solutions in each region are simulated infinitely often with probability 1, which guarantees the global convergence of the multiresolution algorithm.

D Region-Level GMRF

In defining the precision matrix at the region level, we have to consider that the uncertainty about the response $z(\mathcal{P}_\ell)$ consists of the uncertainty due to having simulated only the subset of solutions $\mathbf{x} \in \Xi_2(\mathcal{P}_\ell)$, and also the intrinsic stochastic noise in the simulation of those solutions. As an approximation to the former source, we treat the set of solutions $\Xi_2(\mathcal{P}_\ell)$ as if they were selected randomly, uniformly and without replacement from $\mathcal{P}_\ell$, even though they would typically be selected systematically or due to their CEIs. Therefore, $\Xi_2(\mathcal{P}_\ell)$ is treated as a random subset. For the latter source of uncertainty, we define the GMRF $\mathbb{Z}_2^{\varepsilon} = \mathbb{Z}_2 + \boldsymbol{\varepsilon}$ where

$$\boldsymbol{\varepsilon} \sim \mathcal{N}(\vec{\mathbf{0}}_{|\Pi_2|\times 1}, \mathbf{T}_{\varepsilon}^{-1}),$$

and $\mathbf{T}_{\varepsilon}$ is the intrinsic precision matrix of the region-level GMRF, which is analogous to $\mathbf{Q}_{\epsilon}$ in the solution-level GMRF.

For design region $\mathcal{P}_\ell$, a realization of $\mathbb{Z}_2^{\varepsilon}(\mathcal{P}_\ell)$ is

$$\bar{\mathcal{Z}}(\mathcal{P}_\ell) = |\Xi_2(\mathcal{P}_\ell)|^{-1} \sum_{\mathbf{x}\in\Xi_2(\mathcal{P}_\ell)} \bar{Y}(\mathbf{x}), \tag{14}$$

where $\bar{Y}(\mathbf{x})$ is the sample mean from $r(\mathbf{x})$ replications at $\mathbf{x} \in \Xi_2(\mathcal{P}_\ell)$. Therefore, the diagonal element of $\mathbf{T}_{\varepsilon}$ that corresponds to region $\mathcal{P}_\ell$ is the reciprocal of

$$\mathrm{Var}(\bar{\mathcal{Z}}(\mathcal{P}_\ell)) = \mathrm{Var}\left(|\Xi_2(\mathcal{P}_\ell)|^{-1} \sum_{\mathbf{x}\in\Xi_2(\mathcal{P}_\ell)} y(\mathbf{x})\right) + \mathrm{E}\left(|\Xi_2(\mathcal{P}_\ell)|^{-2} \sum_{\mathbf{x}\in\Xi_2(\mathcal{P}_\ell)} \sigma^2(\mathbf{x})/r(\mathbf{x})\right). \tag{15}$$

Notice that the first term on the left-hand-side of (15) represents the spatial variability of $y(\mathbf{x})$ among $\mathbf{x} \in \mathcal{P}_\ell$ and the second term represents the stochastic variability in $\bar{Y}(\mathbf{x})$, assuming independent simulations, not CRN. The variance in the first term and the expectation in the second term are with respect to the spatial variability among $\mathbf{x}$ due to the random subset $\Xi_2(\mathcal{P}_\ell)$. Thus, the first term decreases as we simulate more solutions within $\mathcal{P}_\ell$ and the second term decreases as we simulate more replications from design points in $\mathcal{P}_\ell$.

Remark: Notice that $\bar{\mathcal{Z}}(\mathcal{P}_\ell)$ may not be an unbiased estimator of $z(\mathcal{P}_\ell)$ as we do not actually sample the solutions in $\mathcal{P}_\ell$ randomly. However, the purpose of region-level GMRF is not to make

precise predictions of the region-level average, but rather to provide a guide to more promising regions. As shown in Appendix C, MR-GMIA nevertheless convergences to a globally optimal solution asymptotically under mild conditions.

This completes the region-level GMRF framework; next we provide some specific choices to create the multiresolution algorithm MR-GMIA.

We assume a constant mean for the region-level GMRF, just as in the solution-level GMRF; i.e., $\boldsymbol{\eta} = (\eta_0, \eta_0, \ldots, \eta_0)^\top$. The region-level parameters η_0 and $\boldsymbol{\tau}$ are estimated using observations defined in (14) from the design regions via MLE, employing the following plug-in estimator for $\mathbf{T}_\varepsilon$: Since we treat the $y(\mathbf{x})$ as if they were independently randomly sampled from $\mathcal{P}_\ell$, the first term in (15) can be estimated by

$$(|\Xi_2(\mathcal{P}_\ell)|(|\Xi_2(\mathcal{P}_\ell)|-1))^{-1} \sum_{\mathbf{x}\in\Xi_2(\mathcal{P}_\ell)} (\bar{Y}(\mathbf{x}) - \bar{\mathcal{Z}}(\mathcal{P}_\ell))^2 (1 - |\Xi_2(\mathcal{P}_\ell)|/|\mathcal{P}_\ell|). \tag{16}$$

The multiplier $1 - |\Xi_2(\mathcal{P}_\ell)|/|\mathcal{P}_\ell|$ is a correction factor for the finite population of solutions in $\mathcal{P}_\ell$ [Cochran, 1977]. The second term in (15) can be estimated by

$$|\Xi_2(\mathcal{P}_\ell)|^{-2} \sum_{\mathbf{x}\in\Xi_2(\mathcal{P}_\ell)} S^2(\mathbf{x})/r(\mathbf{x}). \tag{17}$$

Thus, the estimator of $\mathrm{Var}(\bar{\mathcal{Z}}(\mathcal{P}_\ell))$, denoted $S^2(\mathcal{P}_\ell)$, is the sum of (16) and (17), and the diagonal element of $\widehat{\mathbf{T}}_\varepsilon$ that corresponds to design region $\mathcal{P}_\ell$ is $1/S^2(\mathcal{P}_\ell)$. If we simulate all solutions in $\mathcal{P}_\ell$ infinitely often, so that $|\Xi_2(\mathcal{P}_\ell)| \to |\mathcal{P}_\ell|$ and $r(\mathbf{x}) \to \infty$ for all $\mathbf{x} \in \mathcal{P}_\ell$, then $S^2(\mathcal{P}_\ell) \to 0$ with probability 1.

With $\widehat{\mathbf{T}}_\varepsilon$ and $\bar{\mathcal{Z}}_2 = \{\bar{\mathcal{Z}}(\mathcal{P}_\ell), \forall \mathcal{P}_\ell \in \Pi_2\}$, we can compute the MLEs $\widehat{\boldsymbol{\eta}}$ and $\widehat{\boldsymbol{\tau}}$, as well as the conditional distribution of $(\mathbb{Z}_1, \mathbb{Z}_2)^\top | \mathbb{Z}_2^\varepsilon = \bar{\mathcal{Z}}_2$. From this conditional distribution we can derive the region-level CEI for all regions in $\boldsymbol{\mathcal{P}}$, which we use as guidance to choose a region to simulate in our multiresolution algorithm; the solution-level CEI determines solutions to simulate within a selected region.