Submitted to *Operations Research*
manuscript (Please, provide the manuscript number!)

Bayesian Optimization Allowing for Common Random Numbers

Michael Pearce
Complexity Science, Warwick University, Coventry, UK m.a.l.pearce@warwick.ac.uk

Matthias Poloczek
Amazon, San Francisco, USA*, matpol@amazon.com

Juergen Branke
Warwick Business School, Coventry, UK, juergen.branke@wbs.ac.uk

Bayesian optimization is a powerful tool for expensive stochastic black-box optimization problems such as simulation-based optimization or machine learning hyperparameter tuning. Many stochastic objective functions implicitly require a random number seed as input. By explicitly reusing a seed, a user can exploit common random numbers, comparing two or more inputs under the same randomly generated scenario, such as a common customer stream in a job shop problem, or the same random partition of training data into training and validation set for a machine learning algorithm. With the aim of finding an input with the best average performance over infinitely many seeds, we propose a novel Gaussian process model that jointly models both the output for each seed and the average over seeds. We then introduce the Knowledge Gradient for Common Random Numbers that iteratively determines a combination of input and random seed to evaluate the objective and automatically trades off reusing old seeds and querying new seeds, thus overcoming the need to evaluate inputs in batches or measuring differences of pairs as suggested in previous methods. We investigate the Knowledge Gradient for Common Random Numbers both theoretically and empirically, finding it achieves significant performance improvements with only moderate added computational cost.

Key words: Kriging, Gaussian process regression, common random numbers, myopically optimal policies.

History: This paper was first submitted on August 5, 2019

---

* The work was done while Matthias was affiliated with the University of Arizona in Tucson, AZ.

1. Introduction

We consider the problem of expensive stochastic optimization with limited evaluations,

$$\arg\max_{x \in X} \mathbb{E}[\theta(x, s)], \tag{1}$$

where $\theta(x, s)$ is a real valued output, and $X \subset \mathbb{R}^d$ is the solution space, usually given by box constraints for continuous variables, or a set of discrete alternatives. The parameter s represents all of the stochasticity in the objective, i.e., $\theta(x, s)$ is deterministic. For example, s may be the seed of a pseudo random number generator that is called within a simulator and that uniquely defines a "scenario" passed to the objective function. Hence evaluating multiple x with the same s will reuse a set of common random numbers (CRN). The aim of optimization is to find an $x \in X$ that performs best when averaged over all possible randomly generated scenarios and we apply Bayesian optimization for the search. Example applications include

Control and Reinforcement Learning: x are parameters of a control policy, s defines a randomly generated environment (e.g. maze, race track, terrain) and $\theta(x, s)$ is final reward.

Machine Learning: x are hyperparameters of a machine learning algorithm or model, s defines a random split of training data into train and validation sets, and $\theta(x, s)$ is accuracy.

Simulation Optimization: In many optimization problems, a solution x can only be evaluated by a stochastic simulator $\theta(x, s)$ whose random number generator seed s we may choose.

In this work we empirically investigate the following two simulation optimization applications.

Inventory Management: x are target inventory levels below which more stock is ordered, s defines a random stream of customers and $\theta(x, s)$ is profit.

Base Location: x are spatial locations of ambulance bases, s defines times and locations of patients randomly appearing across the map, and $\theta(x, s)$ is average ambulance journey time.

CRN is well known in the simulation field as a powerful variance reduction technique which helps to discern the quality difference between alternatives. It is beneficial whenever the quality of two solutions for different seeds is positively correlated since

$$\begin{aligned}\mathrm{Var}(\theta(x,\cdot) - \theta(x',\cdot)) &= \mathrm{Var}(\theta(x,\cdot)) + \mathrm{Var}(\theta(x',\cdot)) - 2\mathrm{Cov}(\theta(x,\cdot), \theta(x',\cdot)) \\ &< \mathrm{Var}(\theta(x,\cdot)) + \mathrm{Var}(\theta(x',\cdot)).\end{aligned}$$

This is the case for example because some seeds create scenarios that are easier than others, such as in the above inventory problem, when a simulation with more customers is likely to lead to higher profit for most reorder levels.

From a surrogate modeling perspective, as a result of using CRN, the noise corrupting the objective output is correlated for the same seed. This is in contrast to the common assumption of independent noise for the objective outputs. There have been some previous works that use CRN in combination with Bayesian optimization, either evaluating pairs of candidates or multiple comparisons either "with CRN" or "without CRN".

In this work, we take a different perspective, generalizing past approaches and explicitly modeling the influence of the seed. We highlight that the domain of the objective is the cross-product of the *solution space* and positive integer seeds $X \times \{1, 2, ...\}$ and we refer to this domain as the *acquisition space*. Therefore, the surrogate model must be defined over $X \times \mathbb{N}^+$, the optimization algorithm must propose input pairs $(x, s) \in X \times \mathbb{N}^+$ and evaluate $\theta(x, s)$ with the goal of learning $\text{argmax}_x \bar{\theta}(x) = \text{argmax}_x \mathbb{E}[\theta(x, \cdot)]$. Given this perspective, we emphasize that the benefit of using CRN comes from the *emergent structure in the noise*, i.e., how the output for a single seed is uniquely different from the average over seeds,

$$\epsilon_s(x) = \theta(x, s) - \bar{\theta}(x). \tag{2}$$

In particular, if $\epsilon_1(x) = o_1$ is the constant function, this implies that $\text{argmax}_x \bar{\theta}(x) = \text{argmax}_x \theta(x, 1)$ and it is sufficient to optimize the single seed $s = 1$. Thus, first we propose a Gaussian process model for $\theta(x, s)$ that also yields a method for inferring $\bar{\theta}(x)$ and is a generalization of standard models. Second, we propose the Knowledge Gradient for Common Random Numbers (KG^{CRN}) that quantifies the value of a new point in $X \times \mathbb{N}^+$ for learning the optimizer of the average over infinitely many seeds, argmax $\bar{\theta}(x)$. Optimizing KG^{CRN} determines the most beneficial combination of solution x executed with seed s to learn $\text{argmax}_x \bar{\theta}(x)$. The KG^{CRN} algorithm is therefore able to automatically trade-off the benefits of evaluating x with a previously evaluated seed (i.e., utilizing CRN), and of evaluating x with a fresh new seed, by simply maximizing the expected benefit. This

removes both, the need to observe multiple x simultaneously in a batch with CRN or the need to consider differences in pairs of outputs evaluated with CRN.

In the following section we briefly summarize related work, then formally define the problem in Section 3. Section 4 describes and motivates the proposed surrogate model and Section 5 derives the new acquisition procedure and discuses practicalities. In Section 6 we draw parallels with a previous approach based on pairwise sampling. An empirical evaluation on both synthetic experiments and the two simulation optimization applications mentioned above is presented in Section 7. The paper concludes in Section 8.

2. Literature Review

Common random numbers (CRN) can be applied to any stochastic optimization problem where the user can control the randomness of the objective. A typical use case in stochastic computer simulation is Ranking and Selection, the problem of finding the best from a finite (small) set of uncorrelated solutions. In such a problem setting, a user is able to perform repeated evaluation of all solutions, see Kim (2013) and Frazier (2012) for a summary of frequentist and Bayesian techniques, respectively. Combining CRN with ranking and selection has been considered with two-stage methods (Nelson and Matejcik 1995, Chick and Inoue 2001) that initially sample all solutions multiple times to learn noise covariance structure and a second stage to exploit the learnt structure. Fu et al. (2004) further investigate the second stage of the two stage process. More recently, a sequential method has been proposed by Görder and Kolonko (2019) that keeps track of all sampled seeds and uses the same series of seeds for all candidates.

When the candidate solutions have associated features that can inform simulation output, then surrogate models can aid the optimization and enable search over much larger (possibly infinite) spaces X. Gaussian Random Fields allow to define a correlated prior over outputs that depends on similarity in inputs across the space. Gaussian processes (GP) (Rasmussen 2003), or Kriging (Ankenman et al. 2010), are often employed when the search space is numerical, i.e., continuous or integer. Jones et al. (1998) consider the optimization of a deterministic function using a Gaussian

process. Huang et al. (2006b) and Scott et al. (2011) among many others consider noisy functions assuming independent noise. For integer ordered spaces, or any lattice/network, one may employ Gaussian Markov Random Fields (Salemi et al. 2019) for faster computation. The consequence of GP modeling with correlated noise has been considered by Chen et al. (2012) when assuming constant noise correlation across the solution space X. Xie et al. (2016) propose a method to combine a GP with CRN for optimization. They sample either a single solution or a pair under a new seed in each iteration.

In this work we consider the seed s a (categorical) input to the objective $\theta(x, s)$ and the target of optimization $\bar{\theta}(x)$ is the objective with the s argument "integrated out". Hence this work is related to optimization of functions with (continuous) integrals (Toscano-Palmerin and Frazier 2018) or simulation optimization with an uncertain simulation input parameter (Pearce and Branke 2017). Both methods sequentially determine a solution and input parameter in order to optimize the objective integrated over input parameters. In such a problem setting, the surrogate model and data collection are defined over the multidimensional domain of decision variables and input parameters. However, in the CRN setting, the variable to be averaged out is categorical and there is no "similarity" over seeds. In this work we show how the structural assumptions of CRN lead to a specific model design and interactions with the acquisition procedure. This results in a *dynamically growing* acquisition space yet the algorithm still maintains minimal computational increase over an equivalent non-CRN algorithm.

3. Problem Definition

Let $\theta : X \times \mathbb{N}^+ \to \mathbb{R}$ be an expensive-to-evaluate, real valued function with arguments composed of a real valued solution $x \in X \subset \mathbb{R}^d$ and a nominal positive integer seed $s \in \mathbb{N}^+$ and the domain is the *acquisition space* $\tilde{X} = X \times \mathbb{N}^+$. We refer to $\theta(x, s)$ as the *objective function*. The random seed s controls all stochasticity in the function, i.e., $\theta(x, s)$ is deterministic. Note that we use the term 'seed' to represent all the stochasticity in the objective function. Usually a simulation requires many random numbers, often organised in synchronised streams, e.g., a queuing model has one

stream for arrival times and one stream for processing times. We use s to represent one complete set of streams of random numbers needed for one simulation, as could for example be generated by a pseudo random number generator called after setting a specific seed. It is not necessary to assume that each solution will use the same amount of random numbers from such streams.

The aim is to identify the solution x from the *solution space* X that maximizes the expectation of the objective over random number streams

$$\arg\max_x \bar{\theta}(x) = \arg\max_x \mathbb{E}[\theta(x, \cdot)]$$

and we refer to $\bar{\theta}(x)$ as the *target*. There is a limited budget of N objective function calls, and for each call, the user can choose a seed s and a solution x, then observe $y = \theta(x, s)$. Function evaluations may be collected sequentially so that after n measurements the user may determine the x and s for the $(n+1)^{th}$ function evaluation.

If every call to the function uses a new unique random seed, the problem reduces to standard stochastic optimization and the user only needs to determine x values for each evaluation of $\theta(x, s)$. The problem considered here is therefore a more general setting that allows the reuse of random number seeds by making the argument s explicit.

There are various extensions to this problem setting; one could allow the simulation run length to be varied, augmenting the domain with a run length variable. Similarly, the setting may be extended to account for variable cost of objective calls. Here we consider the fundamental setting where the simulator is a black-box that takes the pair (seed, solution) as input and returns the observed performance.

4. A Surrogate Model for Simulation with Common Random Numbers

Before we propose KG^{CRN} in Section 5, we describe our model for $\theta(x, s)$.

4.1. The Gaussian Process Generative Model

A generative model is a probability distribution over all observable and unobservable quantities and such a model can be sampled to generate realizations of all variables thereby synthesizing data.

Inference is the task of estimating the unobserved quantities that are consistent with both, the generative model and the observed quantities. In the case of optimization with CRN, we desire a generative model with two properties. First, sampling outputs from the generative model assuming each output comes from a different seed must recover a model used without CRN. Second, the seeds are labeled with arbitrary numbers, there is no exploitable "neighborhood" between seeds.

Following previous works without CRN, we first assume that the target, $\bar{\theta}(x)$, is a realization of a Gaussian process with (typically constant) prior mean $\bar{\mu}(x)$ and covariance function or kernel $k_{\bar{\theta}}(x, x')$ such as a $\frac{5}{2}$-Matérn or squared exponential. We use the following notation

$$\bar{\theta}(x) \sim \mathrm{GP}\big(\bar{\mu}(x),\ k_{\bar{\theta}}(x, x')\big). \tag{3}$$

Given n solutions $X^n = (x^1, ..., x^n)$ and corresponding seeds $S^n = (s^1, ..., s^n)$, when all seeds are unique, e.g. $s^i = i$, each output value is generated by evaluating the target and adding independent and identically distributed Gaussian noise $y^i \sim N(\bar{\theta}(x^i), \sigma^2_\epsilon(x^i))$ with variance $\sigma^2_\epsilon(x^i)$. The vector of outputs, $Y^n = (y^1, ..., y^n)$, is assumed to be a single multivariate Gaussian random vector with constant mean and a covariance matrix composed of a kernel matrix and diagonal noise matrix with entries $\sigma^2_\epsilon(X^n)$,

$$Y^n \sim N\big(\bar{\mu}(X^n),\ k_{\bar{\theta}}(X^n, X^n) + \mathrm{diag}(\sigma^2_\epsilon(X^n))\big). \tag{4}$$

For $\theta(x, s)$ in the CRN setting, we require a kernel over $\tilde{X} = X \times \mathbb{N}^+$ that when evaluated for unique seeds recovers the above covariance matrix. To reproduce the diagonal matrix, we require a Kronecker delta function over seeds $\delta_{s's}$. To model covariance in outputs for the same seed, we require a *difference kernel* $k_\epsilon(x, x')$ over $X \times X$ that must satisfy $k_\epsilon(x, x) = \sigma^2_\epsilon(x)$, we return to the design of $k_\epsilon(x, x')$ shortly. We propose the following model for the objective,

$$\theta(x, s) \sim \mathrm{GP}\big(\bar{\mu}(x, s),\ k_{\bar{\theta}}(x, x') + \delta_{s's} k_\epsilon(x, x')\big), \tag{5}$$

where $\bar{\mu}(x, s) = \bar{\mu}$ is the constant prior mean. Given X^n and S^n, the generative distribution of Y^n is thus

$$Y^n \sim N\big(\bar{\mu}(\tilde{X}^n),\ k_{\bar{\theta}}(X^n, X^n) + \mathbb{1}_{S^n} \circ k_\epsilon(X^n, X^n)\big), \tag{6}$$

where $\circ$ denotes matrix element-wise product and $\mathbb{1}_{S^n} \in \{0,1\}^{n\times n}$ is a binary masking matrix with elements equal to one at (i,j) when $s^i = s^j$. Hence for the noise matrix, $\mathbb{1}_{S^n} \circ k_\epsilon(X^n, X^n)$, the diagonal and also any off-diagonal elements where $s^i = s^j$ are non-zero with corresponding covariance $k_\epsilon(x^i, x^j)$. The model assumes the form of the objective consists of the target and *difference functions*, $\epsilon_s(x)$,

$$\theta(x,s) = \bar{\theta}(x) + \epsilon_s(x) \tag{7}$$

where each $\epsilon_s(x)$ is an independent GP realization from a common model with constant zero mean,

$$\epsilon_1(x), \epsilon_2(x), ... \sim \mathrm{GP}\big(0,\ k_\epsilon(x,x')\big). \tag{8}$$

This model structure has multiple desirable properties. Firstly, by design it mirrors the standard model for non-CRN use cases, $y = \bar{\theta}(x) + \epsilon$, where it is commonly assumed that all ϵ are independent Gaussian *variable* realizations. With CRN, the "noise" terms $\epsilon_s(x)$ are independent Gaussian *process* realizations. Secondly, $k_\epsilon(x,x')$ dictates the covariance in differences from the target (the covariance in noise) induced by CRN as a function of x and x', we discuss our choice below. Thirdly, $k_\epsilon(x,x')$ is typically a parametric function whose hyperparameters are learnt from multiple realizations, $\epsilon_1(x), \epsilon_2(x), ...$, of a single GP and each seed may be viewed as a task in a multi-task model. This differs slightly from other multi-task models commonly used for multi-fidelity optimization (Swersky et al. 2013, Poloczek et al. 2017), or for multi-objective optimization (Picheny 2015), where one task is not necessarily the same as others and a unique GP model for each task may be more suitable. However, because all $\epsilon_s(x)$ come from a single common GP, the kernel $k_\epsilon(x,x')$ must have the flexibility to model how the objective for any seed may differ from the target.

We assume a decomposition of the difference functions, $\epsilon_s(x)$, into three parts: a constant offset o_s, a bias function $b_s(\cdot)$, and white noise $w_s(\cdot)$:

$$\theta(x,s) \ = \ \bar{\theta}(x) + \epsilon_s(x) \ = \ \bar{\theta}(x) + o_s + b_s(x) + w_s(x). \tag{9}$$

Firstly, we aim to capture the notion that some seeds may result in scenarios that are "easy" and others "hard", for example if the demand for a product in a simulation scenario is higher, most

inventory policies or solutions x will be able to generate higher profit. Thus $\epsilon_s(x)$ may contain a global offset o_s modeled by the constant kernel $k_o(x, x') = \eta^2$,

$$o_s(x) \sim \mathrm{GP}(0, k_o(x, x')), \tag{10}$$

where the sample function is constant for all x and hence denoted by $o_s \sim N(0, \eta^2)$.

Secondly, each seed may create a scenario that favours some solutions while penalising others, leading to seed specific peaks and troughs thus the optimal solution differs across seeds. For example, given a stream with high demand, one solution x may perform above average while another solution x' is below average. A stream with low demand may reverse the solution ranking. Overall, we expect the response surface for a particular seed to be reasonably smooth, i.e., for very similar solutions x and x' we expect very similar performance. This is ensured by introducing correlation across nearby solutions in *bias* functions $b_s(x)$ modelled with the same kernel as the target, however rescaled $k_b(x, x') = \sigma_b k_{\bar{\theta}}(x, x')$,

$$b_s(x) \sim \mathrm{GP}\left(0, k_b(x, x')\right). \tag{11}$$

Thirdly, fixing the seed results in a deterministic simulator. Despite the fact that we assume a mostly smooth response surface, there may be discontinuities. For example, in discrete event simulation, there may exist x for which a small change may suddenly result in a different execution path of the simulator and a sudden change in outcome. In practice, such effects are impossible to model. Such effects not captured by o_s and $b_s(x)$, as well as other problems of model misfit, may simply be treated as noise. Thus, we follow Chen et al. (2012) and Xie et al. (2016) and include a realization of white noise $w_s(x)$ with kernel $k_w(x, x') = \delta_{x'x}\sigma_w^2$

$$w_s(x) \sim \mathrm{GP}\left(0, k_w(x, x')\right). \tag{12}$$

Therefore, this functional form of $\theta(x, s)$ is a realization of the Gaussian process

$$\theta(x, s) \sim \mathrm{GP}\big(\bar{\mu},\ k_{\bar{\theta}}(x, x') + \delta_{ss'}(\eta^2 + k_b(x, x') + \sigma_w^2\delta_{xx'})\big) \tag{13}$$

$$= \mathrm{GP}\big(\bar{\mu},\ k(x, s, x', s')\big). \tag{14}$$

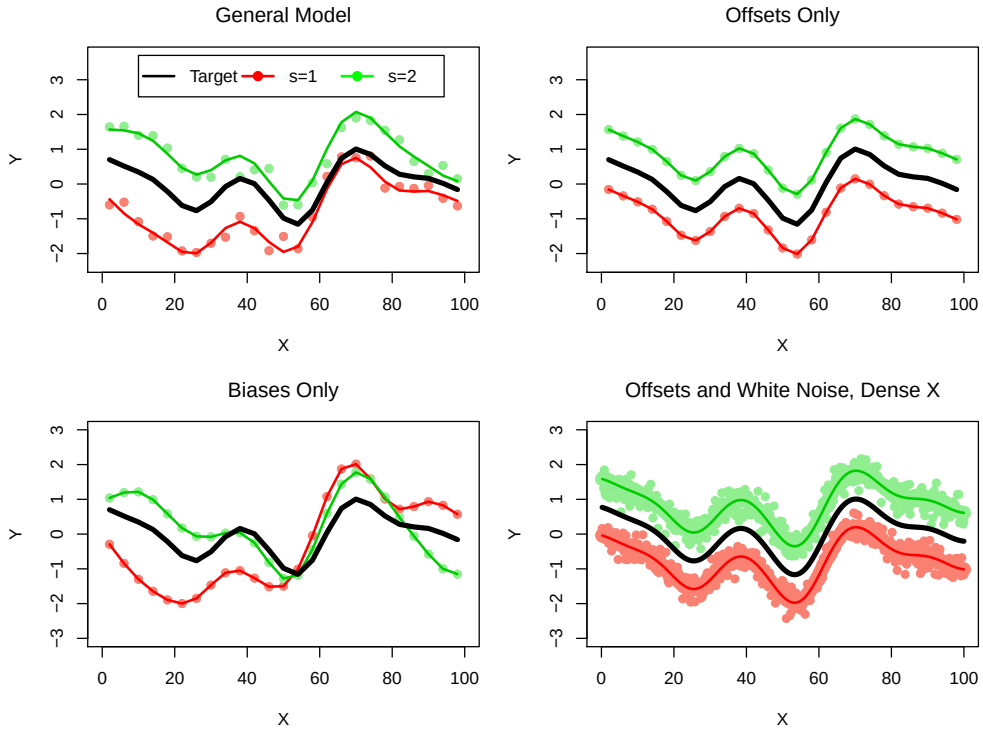

Figure 1 Samples from the generative model. In all plots, lines show $\bar{\theta}(x)$ and $\bar{\theta}(x) + o_s + b_s(x)$ (no white noise), points show $\theta(x, s)$ (including white noise). Left plots: an algorithm must evaluate multiple seeds to find optimum. Right plots: an algorithm can optimize one seed to find $\arg\max \bar{\theta}(x)$.

See Figure 1 for artificial example realizations, and Figure EC.8 in the Electronic Companion for some realizations on the two application examples we consider in this paper. These also show characteristics consistent with the above model.

Note that a standard homoscedastic Gaussian process model requires choosing a kernel $k_{\bar{\theta}}(x, x')$ and learning a single noise parameter w_s. The CRN model introduces just two further parameters o_s^2 and b_s^2. While more general models may be used, introducing significantly more parameters can easily lead to overfitting issues. We discuss this in more detail in Section 5.2.1. For the rest of this section, we assume that all kernels are known functions and the unknown $\theta(x, s)$ are to be inferred.

4.2. Inferring the Objective $\theta(x,s)$

We denote an observation at time n as (x^n, s^n, y^n), the sequence of observed solutions as $(x^1,...,x^n) = X^n$, the sequence of observed seed values as S^n and the sequence of input pairs, $\tilde{x}^i = (x^i, s^i)$, as $(\tilde{x}^1,...,\tilde{x}^n) = \tilde{X}^n$. The vector of observed outputs is denoted $(y^1,...,y^n) = Y^n$. And, abusing notation, we also treat these as sets, e.g., $\tilde{x} \in \tilde{X}^n$, and use both (x,s) and $\tilde{x}$ interchangeably to represent an input pair. The dataset of observed inputs and outputs we denote $D^n = ((\tilde{x}^1, y^1),...,(\tilde{x}^n, y^n))$. Inferring the underlying realization of $\theta(x,s)$ can be done analytically using the Bayesian update equations for multivariate Gaussian random variables,

$$\theta(x,s)|D^n \sim \mathrm{GP}\big(\mu^n(x,s),\ k^n(x,s,x's')\big)$$

$$\mu^n(x,s) = \mu^0(x,s) - k^0(x,s,\tilde{X}^n)K^{-1}(Y^n - \mu^0(\tilde{X}^n)) \tag{15}$$

$$k^n(x,s,x',s') = k^0(x,s,x',s') - k^0(x,s,\tilde{X}^n)K^{-1}k^0(\tilde{X}^n,x',s') \tag{16}$$

where $k^0(x,s,x's')$ is any positive semi-definite kernel over $X \times \mathbb{N}^+$. The matrix $K = k^0(\tilde{X}^n, \tilde{X}^n)$ is the generative covariance for Y^n. For the rest of this work, we use the shorthand $\mathbb{E}_n[\cdot] = \mathbb{E}[\cdot|D^n]$. Note that there is no added identity matrix as in Equation (4), thus the model assumes deterministic outputs for any given input pair (x,s). At first, this may appear at odds with the white-noise assumption. However, white noise may be viewed as a squared exponential kernel with infinitely short length scale and $w_s(x)$ is a realization of such a generative model. As a result, the posterior mean discontinuously interpolates the data as shown in Figure 2.

4.3. Inferring the Target $\bar{\theta}(x)$

The model of $\theta(x,s)$ and collected data is over the acquisition space $X \times \mathbb{N}^+$ while the aim of the optimization is to maximize $\bar{\theta}(x)$ over solution space X. The target is the objective averaged over infinite seeds and therefore the GP model of $\theta(x,s)$ averaged over infinite seeds induces another GP for the target $\bar{\theta}(x)$ as follows.

LEMMA 1. *For any given kernel over $X \times \mathbb{N}^+$ that is of the form $k_{\bar{\theta}}(x,x') + \delta_{ss'}k_\epsilon(x,x')$, and a dataset of n input-output triplets D^n, the posterior over the target is a Gaussian process given by*

$$\bar{\theta}(x)|D^n \sim GP(\mu^n_{\bar{\theta}}(x), k^n_{\bar{\theta}}(x,x')) \tag{17}$$

$$\mu^n_{\bar{\theta}}(x) = \mu^n(x,s') \tag{18}$$

$$k^n_{\bar{\theta}}(x,x') = k^n(x,s',x',s'') \tag{19}$$

where s', $s'' \in \mathbb{N}^+ \setminus S^n$ with $s' \neq s''$ are any two unobserved unequal seeds.

The intermediate steps and proof are given in the Electronic Companion EC.1.1. In practice, seed values are mapped to $\{1, 2, \dots\}$ (without gaps) and we construct a single GP model for $\theta(x,s)$ over $X \times \mathbb{N}^+$. We then use predictions from this model on the unobservable seed $s = 0$ as predictions of the target, $\mathbb{E}_n[\bar{\theta}(x)] = \mu^n(x,0)$.

5. Knowledge Gradient for Common Random Numbers Algorithm

The Knowledge Gradient for Common Random Numbers algorithm (KG^{CRN}) is summarized in Algorithm 1. Given a budget of N calls to $\theta(x,s)$, the proposed Bayesian optimization algorithm has two phases, an initialization phase where we evaluate a small number of candidates $n_{init} \ll N$, chosen as a space filling design in $X \times \{1,2,3,4,5\}$. That is, we instantiate five (arbitrarily chosen) seeds to collect data points that are then used to fit a GP model (we show in the Appendix that the algorithm is insensitive to the number of seeds used for initialisation). In the second phase an acquisition function (infill criterion) is used to sequentially allocate the remaining $N - n_{init}$ points of the budget, updating the model after each new point and determining the next point. Section 5 explains the details of the acquisition function, implementation details and algorithm properties are discussed in Sections 5.2 and 5.3, respectively.

5.1. Acquisition Function

Evaluations of $\theta : X \times \mathbb{N}^+ \to \mathbb{R}$ are collected in order to optimize $\bar{\theta} : X \to \mathbb{R}$. Given a joint model of both functions, the acquisition function quantifies the benefit of a new hypothetical observation

at $(x,s) \in \tilde{X}$. This function is then optimized to obtain the most informative $(x,s)^{n+1}$ and the objective is evaluated at $y^{n+1} = \theta(x^{n+1}, s^{n+1})$. The surrogate model is defined over the space of non-negative seeds $X \times \{0,1,...\}$, the model of the target is over $X \times \{0\}$ while the objective, and acquisition, is over $X \times \{1,2,..\}$. Therefore we require a 'correlation aware' acquisition function that computes the benefit of a sample at $(x,s)^{n+1}$ for $s^{n+1} > 0$ by measuring changes in the model at other locations $(x',0) \neq (x,s)^{n+1}$. This requirement excludes certain acquisition functions in their unmodified form such as Expected Improvement (Jones et al. 1998), Upper Confidence Bound (Srinivas et al. 2009) and Thompson sampling (Kandasamy et al. 2018). Two popular families of acquisition functions that naturally account for how the whole surrogate model changes include Entropy Search (Villemonteix et al. 2009), and Knowledge Gradient (Frazier et al. 2009). Knowledge Gradient quantifies the benefit of a new hypothetical point $(x,s,y)^{n+1}$ as the expected incremental increase in the predicted outcome for the user, peak posterior mean $\mathbb{E}[\max_x \mu^{n+1}(x) - \max_x \mu^n(x) | D^n, x^{n+1}]$. In this work we adopt the Knowledge Gradient for its principled value of information-based approach and provable performance guarantees.

In our setting the value of information is the expected increase in the predicted peak of the target, $\max \mu^{n+1}(x,0) - \max \mu^n(x,0)$, caused by a new sample y^{n+1} at $(x,s)^{n+1}$. The Knowledge Gradient for Common Random Numbers, $\mathrm{KG}_n^{\mathrm{CRN}} : \tilde{X} \to \mathbb{R}^+$, is given by

$$\mathrm{KG}_n^{\mathrm{CRN}}(x,s) = \mathbb{E}_n\left[\max_{x' \in X} \mu^{n+1}(x',0) - \max_{x'' \in X} \mu^n(x'',0) \middle| (x,s)^{n+1} = (x,s)\right] \tag{20}$$

$$= \mathbb{E}_n\left[\max_{x' \in X} \{\mu^n(x',0) + \tilde{\sigma}^n(x',0;x,s)Z\} - \max_{x'' \in X} \mu^n(x'',0)\right] \tag{21}$$

where, conditioned on D^n, the expectation is only over $Z \sim N(0,1)$ and

$$\tilde{\sigma}^n(x',0;(x,s)^{n+1}) = \frac{k^n(x',0,(x,s)^{n+1})}{\sqrt{k^n((x,s)^{n+1},(x,s)^{n+1})}}.$$

Note that $\tilde{\sigma}^n(x',0;(x,s)^{n+1}) \cdot Z = \mu^{n+1}(x',0) - \mu^n(x',0)$ gives the change in the posterior mean at x' of the target that results from sampling x with seed s. The numerator is the covariance between the sample location $(x,s)^{n+1}$ and location of the prediction $(x',0)$. The denominator

serves to normalize the posterior variance of the sample. A full derivation of $\tilde{\sigma}^n(\cdot)$ can be found in multiple previous works (Frazier et al. 2009, Pearce and Branke 2017). The next input to the objective, $(x,s)^{n+1}$, is determined by optimizing the above acquisition function $(x,s)^{n+1} = \arg\max_{x,s} \mathrm{KG}_n^{\mathrm{CRN}}(x,s)$. Evaluating of $\mathrm{KG}_n^{\mathrm{CRN}}(x,s)$ can be performed analytically when X is a finite set. For the more general case, approximations are required that we discuss in Section 5.2.2.

The acquisition space, $X \times \mathbb{N}^+$, contains an infinite number of seeds. However as a result of the assumed form of the GP, the posterior mean and correlation are identical for all *unobserved* new seeds $s \in \mathbb{N}^+ \setminus S^n$. Thus, the value under the acquisition function is identical for all new seeds, $\mathrm{KG}_n^{\mathrm{CRN}}(x,s) = \mathrm{KG}_n^{\mathrm{CRN}}(x,s')$ for all $s,s' \in \mathbb{N}^+ \setminus S^n$. Hence, it suffices to consider the acquisition function on all observed seeds $s \in S^n$ and only a single new seed $s = \max\{S^n\} + 1$. Over multiple iterations, new seeds may be evaluated and added to the set of observed seeds and the acquisition space grows accordingly by always including one new seed. Note that the acquisition function is maximized jointly over the old and the new seed. In particular, no heuristics or user input is used to make the exploration-exploitation trade-off over old vs. new seeds. Moreover, we show in Section 5.2.3 how to cheaply compute $\arg\max_{x,s} \mathrm{KG}_n^{\mathrm{CRN}}(x,s)$ in comparable computational time to standard KG.

A connection can be drawn between our algorithm and recent work on multi-information source optimization (Swersky et al. 2013, Poloczek et al. 2017). At a given iteration, each seed in the acquisition space may be viewed as an information source, and a user must choose a solution x and an information source s in order to optimize a target $s = 0$. However in the CRN case, the target itself cannot be observed, all sources have equal budget consumption, and the number of available sources is infinite.

5.2. Implementation Details

In Section 4 we assumed that $k_{\bar{\theta}}(x,x')$ and $k_\epsilon(x,x')$ are known while in practice they require hyperparameters estimated from data. Also, in Section 5 we assumed $\mathrm{KG}_n^{\mathrm{CRN}}(x,s)$ can be evaluated and maximized. These practical issues apply to non-CRN and CRN algorithms, however the CRN

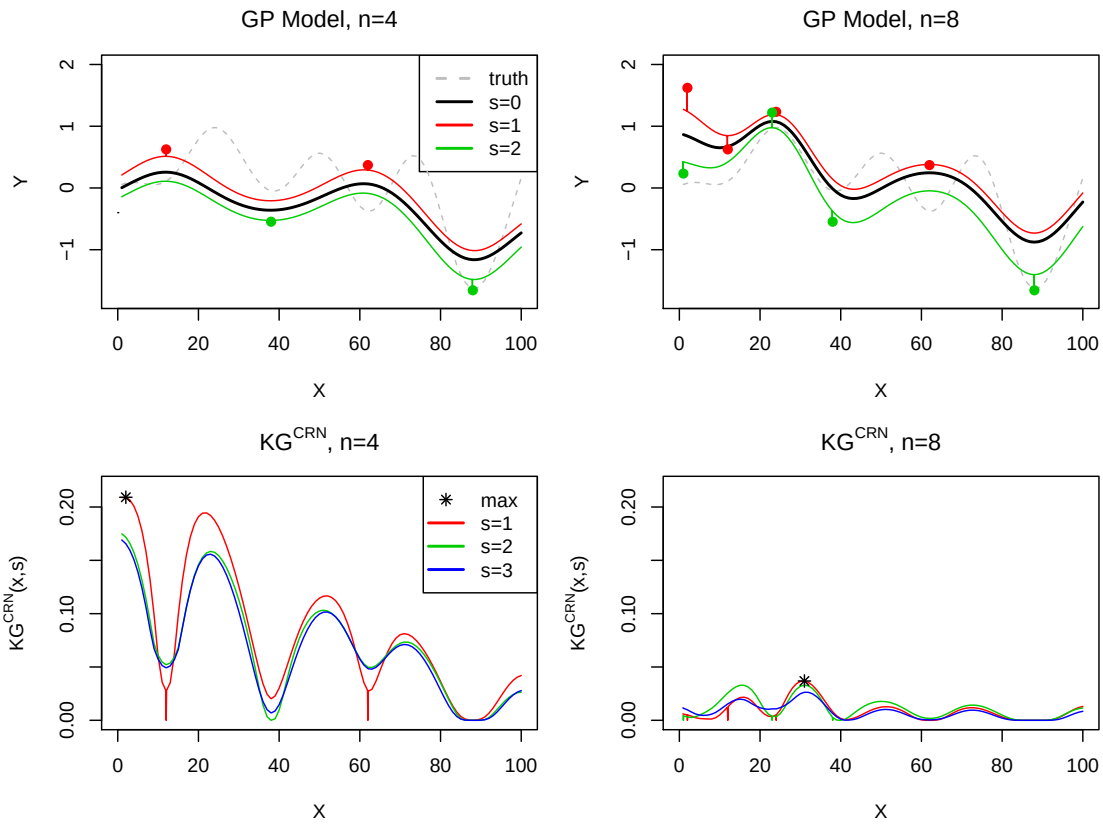

Figure 2 (top) The GP model with offsets, bias functions and white noise. (bottom) KG^{CRN} after 4 initial points on seeds $s = 1, 2$ (left) and an added 4 sequential points by KG^{CRN} (right). All new points were allocated to seeds $s = 1, 2$ and the next point will be allocated to $s = 1$.

model has both more hyperparameters and a larger acquisition space. Ideally, incorporating CRN should not require significantly more computational resources and we discuss such solutions below.

5.2.1. Gaussian Process Hyperparameters. In this work we assume that the target is modeled with the popular squared exponential (SE) kernel

$$k_{\bar{\theta}}(x, x') = \sigma_{\bar{\theta}}^2 \exp(-(x - x')^\intercal L(x - x')/2)$$

where $L = \text{diag}(1/l_1^2, ..., 1/l_d^2)$ is a diagonal matrix of inverse length scales. We also assume that the bias functions come from a squared exponential kernel $k_b(x, x') = \sigma_b^2 \exp(-(x - x')^\intercal L(x - x')/2)$ that shares the diagonal matrix L. The constant kernel and white noise kernel each have a single parameter η^2 and σ_w^2. The constant kernel, over X, models infinitely long range correlation in

differences while the white noise kernel models infinitely short range. Therefore the bias kernel only needs to model intermediate ranges. When determining an intermediate range, one option is to learn length scales for the bias kernel. However, modeling the length scales of the bias kernel adds additional model parameters, requiring more data for tuning. We chose to use the same length scales for the target and all bias functions. The empirical evaluation shows that this worked well in practice. Also, the resulting model complexity is comparable to models not taking CRN into account.

In total, the model has parameters $L, \sigma_{\bar{\theta}}, \eta^2, \sigma_b^2, \sigma_w^2$, two more than a non-CRN model. All parameters are learnt in three steps. First, we fit a non-CRN model with maximum marginal likelihood by simply clamping $\eta^2 = \sigma_b^2 = 0$. This is performed with multi-start conjugate gradient ascent. Second, and specifically for CRN, we perform fine tuning of the difference kernel parameters, η^2, σ_b^2, σ_w^2, with the constraint $\eta^2 + \sigma_b^2 + \sigma_w^2 = \sigma_{w,\text{non-CRN}}^2$. This forces the total variance of the difference functions to be the same as the independent model noise and simplifies model learning. With reparameterization, this is a simple hill-climb over the two dimensional unit square for which we use the Nelder-Mead algorithm. In the third step, we fine-tune all hyperparameters simultaneously without restriction, a further single conjugate gradient ascent. The difference between fitting a non-CRN model and a CRN model is in the two additional steps and can be appended to any Gaussian process code. For details, see the Electronic Companion EC.4.2.

5.2.2. Evaluation of $\mathbf{KG}_n^{\mathbf{CRN}}(x, s)$. The acquisition function, Equation (20), is a one-step look-ahead expected peak posterior mean, an expectation of maximizations over X. This may be evaluated analytically when X is a feasibly small finite set using Algorithm 1 from Frazier et al. (2009). Alternatively, when X is a continuous set, one may replace the expectation over the infinite Z with a Monte-Carlo average. For each Z sample, the inner maximization is performed over X numerically, yielding a stochastic unbiased estimate of $\text{KG}_n^{\text{CRN}}(x, s)$ (Wu et al. 2017).

In this work, we follow Poloczek et al. (2017) and Xie et al. (2016) that use a deterministic approximation. This allows us to reliably test a conjecture and allows direct comparison with prior

work both described in Section 6.2. The inner maximization over X may be replaced with a smaller random subset A that is frozen between iterations thus approximating $\mathrm{KG}^{\mathrm{CRN}}$ with

$$\mathrm{KG}_n^{\mathrm{CRN}}(x,s;A^n) = \mathbb{E}_n\left[\max_{x'\in A\cup\{x\}} \mu^n(x',0) + \tilde{\sigma}^n(x',0;x,s)Z - \max_{x''\in A\cup\{x\}} \mu(x'',0)\right]. \tag{22}$$

We desire a discretization, $A^n \subset X$, that is both dense around promising regions in X while still accounting for unexplored regions. Thus, we propose to construct A from a union of a latin hypercube over X with n points, A^n_{LHC}, and random perturbations of previously sampled points $A^n_P = \{x^i + \gamma | x^i \in X^n\}$ where $\gamma \sim N(\underline{0}, I)$ is Gaussian noise scaled for the application at hand. Finally, we let $A^n = A^n_{LHC} \cup A^n_P$.

5.2.3. Optimization over the Acquisition Space. Typically, acquisition functions are multi-modal functions over X and maximized by multi-start gradient ascent. For $\mathrm{KG}_n^{\mathrm{CRN}}(x,s)$, the acquisition space is larger $\tilde{X}^n_{acq} = X \times \{1,...,\max S^n + 1\}$, suggesting $\mathrm{KG}_n^{\mathrm{CRN}}(x,s)$ needs to be optimized over X for each s. However, recall the fundamental CRN modelling assumption that all seeds have the same latent $\bar{\theta}(x)$. As a result, $\mathrm{KG}_n^{\mathrm{CRN}}(x,s)$ for each seed often has peaks and troughs in similar locations, see Figure 2. Therefore, to maximize $\mathrm{KG}_n^{\mathrm{CRN}}(x,s)$, one may use the same multi-start gradient ascent method for a non-CRN method where instead each start is allocated to a random seed s_i and optimizes x over $X \times \{s_i\}$. Using the best point so far, (x_{ga}, s_{ga}), the same x_{ga} is evaluated for all seeds to find s_{final} and one run of gradient ascent over $X \times \{s_{final}\}$ starting from x_{ga} yields x_{final}. Thus, the only difference in computational cost of acquisition optimisation between a non-CRN method optimizing over X and a CRN method optimizing over $X \times \mathbb{N}^+$ is in the final phase from (x_{ga}, s_{ga}) to (x_{final}, s_{final}).

5.3. Algorithm Properties

The acquisition benefit obtained by sampling solution x with seed s is the expected gain in the quality of the best solution that can be selected given all the available information. In this regard, $\mathrm{KG}^{\mathrm{CRN}}$ is one-step Bayes optimal by construction. The following observation is trivial yet worth

Algorithm 1 The $\mathrm{KG}^{\mathrm{CRN}}$ Algorithm.

Require: $\theta(x,s)$, X, n_{init}, N, $k_{\bar{\theta}}(x,x')$, method to evaluate $\mathbb{E}[\{\max_{x'} a(x') + b(x',x)Z\}]$ and $\nabla_x \mathbb{E}[\{\max_{x'} a(x') + b(x',x)Z\}]$ (Sec.5.2.2), `Optimizer()` over $X \times \mathbb{N}^+$ (Sec. 5.2.3)

$\tilde{X}^{n_{init}} \leftarrow n_{init}$ sampled points by LHC over $X \times \{1,2,3,4,5\}$

$Y^{n_{init}} \leftarrow \theta(\tilde{X}^{n_{init}})$

for $n = n_{init}$ **to** $N-1$ **do**

  $\mu^n(x,s), k^n(x,s,x',s') \leftarrow \mathrm{GP}\big(\theta(x,s)\big|\tilde{X}^n, Y^n, L, \sigma^2_{\bar{\theta}}, \eta^2, \sigma^2_b, \sigma^2_w\big)$ with MLE hyperparameters

  $\mathrm{KG}^{\mathrm{CRN}}_n(x,s) \leftarrow \mathbb{E}[\{\max_{x'} \mu^n(x',0) + \tilde{\sigma}^n(x',0,x,s)Z\}] - \max_{x''} \mu^n(x'',0)$ and gradient w.r.t. x

  $(x,s)^{n+1} \leftarrow$ `Optimizer`$(\mathrm{KG}^{\mathrm{CRN}}_n(x,s))$

  $y^{n+1} \leftarrow \theta(x^{n+1}, s^{n+1})$

  $\tilde{X}^{n+1}, Y^{n+1} \leftarrow (\tilde{X}^n, (x,s)^{n+1}), (Y^n, y^{n+1})$

end for

$\mu^N(x,s) \leftarrow \mathrm{GP}\big(\theta(x,s)\big|\tilde{X}^N, Y^N, L, \sigma^2_{\bar{\theta}}, \eta^2, \sigma^2_b, \sigma^2_w\big)$ with MLE hyperparameters

return $x^N_r = \mathrm{argmax}_x \mu^N(x,0)$

highlighting: standard Knowledge Gradient (KG) is reproduced by constraining $\mathrm{KG}^{\mathrm{CRN}}$ to only acquire data for a new seed in each iteration. Thus, we have

$$\max_{x,s \in \mathbb{N}^+} \mathrm{KG}^{\mathrm{CRN}}_n(x,s) \geq \max_{x,s \in \mathbb{N}^+ \setminus S^n} \mathrm{KG}^{\mathrm{CRN}}_n(x,s) = \max_x \mathrm{KG}(x) \tag{23}$$

and sampling without CRN is a lower bound on the acquisition benefit achievable by $\mathrm{KG}^{\mathrm{CRN}}$.

Given an infinite budget, it is a desirable property for any algorithm to be able to discover the true optimum $x^{OPT} = \mathrm{argmax}_{x \in X} \bar{\theta}(x)$ (assuming there is only one optimizer). Here we give an additive bound on the loss when applying $\mathrm{KG}^{\mathrm{CRN}}$ to a finite subset, A, of continuous space X. Let $k_{\bar{\theta}}(x,x')$ be a Matérn class kernel, and $d = \max_{x' \in X} \min_{x \in A} \mathrm{dist}(x,x')$ the largest distance from any point in the continuous domain X to its nearest neighbor in A.

THEOREM 1. *Let $x^N_r \in A$ be the point that would be recommended after N iterations of the KG^{CRN} algorithm. For each $p \in [0,1)$, there is a constant K_p such that with probability p*

$$\lim_{N \to \infty} \bar{\theta}(x^N_r) > \bar{\theta}(x^{OPT}) - K_p d$$

holds.

The proof is given in the Electronic Companion EC.1.2. Note that this establishes consistency for the finite case where $A = X$ and $d = 0$. Clearly, this bound is conservative as A is randomized at each iteration to avoid "overfitting" and KG^{CRN} recommends the best predicted solution in X, not restricted to A.

6. Comparison with Previous Work

We first show how to recover the generative model considered by Xie et al. (2016) and Chen et al. (2012) as a special case of our proposed model, prove that for this model it is optimal to sample only a single seed, and show that the KG^{CRN} algorithm does exactly this. We then discuss the method of Xie et al. (2016) that also extended Knowledge Gradient to account for common random numbers.

6.1. Compound Sphericity

If the output of each seed differs from the target by a constant offset, then optimizing a single seed optimizes the target. In such an application, the model would learn that there are no bias functions, $\sigma_b = 0$, and the differences kernel reduces to $k_\epsilon(x, x') = \eta^2 + \sigma_w^2 \delta_{xx'}$. Thus, the differences matrix, $k_\epsilon(X^n, X^n)$, is $\eta^2 + \sigma_w^2$ on the diagonal and constant η^2 for all off-diagonal terms. This matrix composition is referred to as compound sphericity (Chen et al. 2012, Xie et al. 2016). Alternatively, the corresponding correlation matrix has a unit diagonal and off diagonal $\rho = \eta^2/(\eta^2 + \sigma_w^2)$ representing the correlation in noise. Let $\Delta^n = Y^n - \mu^0(\tilde{X}^n)$ and $\mathbb{1}_s = \mathbb{1}_{s \in S^n} \in \{0, 1\}^n$ be a binary masking vector. $\mathbb{1}_x$ is defined analogously. Then the posterior mean has the following simple form:

$$\begin{aligned}
\mu^n(x, s) &= \mu^0(x) - (k_{\bar{\theta}}(x, X^n) + \eta^2 \mathbb{1}_s + \sigma_w^2 \mathbb{1}_s \mathbb{1}_x) K^{-1} \Delta^n \\
&= \underbrace{k_{\bar{\theta}}(x, X^n) K^{-1} \Delta^n}_{\mu^n(x,0)} + \underbrace{\eta^2 \mathbb{1}_s K^{-1} \Delta^n}_{\text{independent of } x} + \underbrace{\sigma_w^2 \mathbb{1}_s \mathbb{1}_x K^{-1} \Delta^n}_{=0 \text{ except for } (x^i, s^i) \in \tilde{X}^n} \\
&= \mu^n(x, 0) + A_s + B_s \mathbb{1}_{(x,s) \in \tilde{X}^n}
\end{aligned} \tag{24}$$

and the posterior mean function for a given seed, $s > 0$, differs from the target, $s = 0$, by two additive terms. This leads to the following two lemmas. Both cases correspond to the second additive term

equating to zero. Firstly, if there is no white noise ($\sigma_w^2 = 0$) then for all seeds $\epsilon_s(x) = o_s$ is only a constant offset and a user may simply optimize a single seed to learn $\arg\max \bar{\theta}(x)$. This corresponds to compound sphericity with full correlation, $\rho = 1$, and may be viewed as a "best case" scenario for CRN.

LEMMA 2. *Let the function $\theta(x, s)$ be a realization of a Gaussian process with compound sphericity with full correlation, $\rho = 1$. Then for all $s \in \mathbb{N}^+$, the posterior mean functions have the same optimizer as the target estimate*

$$\arg\max_{x \in X} \mathbb{E}_n[\bar{\theta}(x)] = \arg\max_{x \in X} \mu^n(x, s') \qquad \forall s' \in \mathbb{N}^+.$$

Proof $\rho = 1$ implies $\sigma_w^2 = 0$ which implies $B_s = 0$ in Equation (24), and the posterior means for all seeds differ by only an additive constant A_s. Therefore the maximizer of any two seeds is the same and by Lemma 1 the same maximizer as the estimate of $\mathbb{E}_n[\bar{\theta}(x)]$. □

Secondly, if there is white noise and the set of solutions X is continuous, a user may simply optimize a single seed to learn $\arg\max \bar{\theta}(x)$ as above.

LEMMA 3. *Let the function $\theta(x, s)$ be a realization of a Gaussian process with compound sphericity over a continuous set of solutions X, then for all $s \in \mathbb{N}^+$, the posterior mean functions have the same optimizer excluding locations of past observations $X \setminus X^n$*

$$\arg\max_{x \in X \setminus X^n} \mathbb{E}_n[\bar{\theta}(x)] = \arg\max_{x \in X \setminus X^n} \mu^n(x, s') \qquad \forall s' \in \mathbb{N}^+.$$

Proof By excluding past evaluated solutions $x \in X^n$, the second additive term in Equation (24) vanishes ($B_s \mathbb{1}_{(x,s) \in \tilde{X}^n} = 0$). The posterior means for all seeds differ by only an additive constant, A_s, therefore the maximizer of any two seeds is the same and by Lemma 1 the same as $\mathbb{E}_n[\bar{\theta}(x)]$.

□

The right column of Figure 1 illustrates example functions for these cases and the top row of Figure 2 shows how the posterior mean is discontinuous at evaluated points. If there are no bias functions and these discontinuities are excluded, the posterior mean has the same shape for

all seeds. Consequently, using a compound spheric model in practice is equivalent to assuming that the response surfaces for all seeds have the same shape. This result agrees with those found by Chen et al. (2012): in the case $\rho = 1$ with data collected on seed $s = 1$, the intercept of the function $\bar{\theta}(x)$ is less accurately known while derivatives $\nabla_x \bar{\theta}(x)$ are more accurately known. This is because in the $\rho = 1$ case, the generative modeling assumption imposes the functional form as $\theta(x, s) = \bar{\theta}(x) + o_s$ implying $\nabla_x \theta(x, s) = \nabla_x \bar{\theta}(x)$. It is due to the presence of the *bias* functions, $b_s(x)$, that the optimizer of one seed, $\arg\max_x \theta(x, s)$, is not an accurate estimate of the optimizer of the target function, $\arg\max_x \bar{\theta}(x)$, and an optimization algorithm must evaluate multiple seeds.

Next, in Lemma 4 we show that if all solutions of a finite set X have been evaluated there is no more acquisition benefit according to $\mathrm{KG}^{\mathrm{CRN}}$, the optimizer is known even though its underlying value is unknown.

LEMMA 4. *Let $\theta(x, s)$ be a realization of a Gaussian process with the compound spheric kernel and $\rho = 1$. Let $X = \{x_1, ..., x_d\}$ and evaluated points $\tilde{X}^n = \{(x_1, 1), ..., (x_d, 1)\}$, then for all $(x, s) \in X \times \mathbb{N}^+$, there is no more value of any measurement*

$$KG_n^{CRN}(x, s) = 0 \tag{25}$$

and the maximizer $argmax_x \bar{\theta}(x)$ is known.

Proof is given in the Electronic Companion EC.1.3.

$\mathrm{KG}^{\mathrm{CRN}}$ may be evaluated according to the method proposed by Scott et al. (2011). The method discretizes the inner maximization over X with past evaluated points, X^n, and the new proposed point so that the integral over Z is analytically tractable. Then, in the full correlation case $\mathrm{KG}^{\mathrm{CRN}}$ is guaranteed to never choose a new seed and simplifies to Expected Improvement (Jones et al. 1998) applied to seed $s = 1$.

LEMMA 5. *Let $\theta(x, s)$ be a realization of a Gaussian process with the compound spheric kernel with $\rho = 1$. Let $X \subset \mathbb{R}^d$ be the set of possible solutions, $\tilde{X}^n = \{(x^1, 1), ..., (x^n, 1)\}$ be the set of sampled locations and $X^n = (x^1, ..., x^n)$. Define*

$$KG_n^{CRN}(x, s; A) = \mathbb{E}_n \left[\max_{x' \in A \cup \{x\}} \mu^{n+1}(x', 0) - \max_{x' \in A \cup \{x\}} \mu^n(x', 0) \middle| (x, s)^{n+1} = (x, s) \right]. \tag{26}$$

Then for all $x \in X$

$$KG_n^{CRN}(x,1;X^n) > KG_n^{CRN}(x,2;X^n)$$

and therefore $\max_x KG_n^{CRN}(x,1;X^n) > \max_x KG_n^{CRN}(x,2;X^n)$ *and seed* $s=2$ *will never be evaluated. Further*

$$KG_n^{CRN}(x,1;X^n) = \mathbb{E}_n\big[\max\{0, y^{n+1} - \max Y^n\}\big| x^{n+1}=x, s^{n+1}=1\big].$$

The proof is given in the Electronic Companion EC.1.3.

In the more general case, evaluating $\mathrm{KG}_n^{\mathrm{CRN}}(x,s)$ by any method, when using compound spheric with either full correlation or in a continuous domain X, we conjecture that the true myopically optimal behaviour is to never go to a new seed,

$$\max_{x\in X, s_{old}\in S^n} \mathrm{KG}_n^{\mathrm{CRN}}(x,s_{old}) > \max_{x\in X, s_{new}\notin S^n} \mathrm{KG}_n^{\mathrm{CRN}}(x,s_{new})$$

and a new seed $s \notin S^n$ will never be sampled. However, the above inequality cannot be proven because $\max_{x\in X} \mathrm{KG}^{\mathrm{CRN}}(x,s)$ has no analytic expression and must be found numerically via gradient ascent algorithms. (Note that $\mathrm{KG}_n^{\mathrm{CRN}}(x,s_{old}) > \mathrm{KG}_n^{\mathrm{CRN}}(x,s_{new})$ is not true in general, $x^i \in X^n$ are counterexamples.) Therefore we numerically demonstrate this conjecture in Section 7.

In practice, this conjectured behaviour comes with the risk that if compound sphericity is assumed as in Chen et al. (2012) and Xie et al. (2016), or artificially enforced in our model by clamping $\sigma_b^2 = 0$, the algorithm will try to optimize a single seed regardless of whether or not such behaviour is desirable for a given application. We observe this phenomenon in our experiments in Section 7 where compound sphericity on a continuous search space encourages greedy resampling of only observed seeds. While compound sphericity with full correlation is the best case scenario for CRN, rigidly enforcing this assumption can lead to poor performance. Including bias functions in the model allows it to *optionally* learn compound sphericity.

6.2. Comparison with Knowledge Gradient with Pairwise Sampling

The method proposed by Xie et al. (2016) is also an extension of Knowledge Gradient to use common random numbers. For the generative model, the method assumes that $\bar{\theta}(x)$ is a realization of a GP and considers compound spheric covariance for difference functions. For acquisition, the standard Knowledge Gradient acquisition function quantifies the value of a single observation without CRN (on a new seed) and this is extended with a second acquisition function that quantifies the value of a pair of observations with CRN (on the same new seed). The acquisition space is thus $\tilde{X}^{PW} = \{X, X \times X\}$. The method switches between the serial mode and the batch mode depending on which mode promises the larger value per sample. Since the value of a pair cannot be computed analytically, a lower bound is given by considering the *difference* between the pair of outcomes

$$\mathrm{KG}_n^{\mathrm{PW}}(x_i, x_j) = \frac{1}{2}\left(\mathbb{E}_n\left[\max_{x' \in X}\left\{\mu^n(x', 0) + \tilde{\tilde{\sigma}}^n(x', 0; x_i, x_j)Z\right\} - \max_{x'' \in X}\mu^n(x'', 0)\right]\right) \tag{27}$$

$$\tilde{\tilde{\sigma}}^n(x, 0; x_i, x_j) = \frac{k^n(x, 0, x_i, s^{n+1}) - k^n(x, 0, x_j, s^{n+1})}{\sqrt{k^n(x_i, s^{n+1}, x_i, s^{n+1}) + k^n(x_j, s^{n+1}, x_j, s^{n+1}) - 2k^n(x_i, s^{n+1}, x_j, s^{n+1})}} \tag{28}$$

where $s^{n+1} = n+1$ is a new seed and $\mathrm{KG}_n^{\mathrm{PW}}(x, x')$ is optimized over $X \times X$. Note we have adapted the notation from the original work where the seed is not an explicit argument to the formulation presented in this work. In the original work, numerical evaluation of $\mathrm{KG}^{\mathrm{PW}}$ is performed by discretizing the inner maximization, as discussed in Section 5.2.2. One call to $\mathrm{KG}^{\mathrm{PW}}$ requires evaluating both $k^n(x, 0, x_i, s^{n+1})$ and $k^n(x, 0, x_j, s^{n+1})$ for each x and is thus more expensive than one call to KG or $\mathrm{KG}^{\mathrm{CRN}}$.

In the large $|X|$ setting, it is efficient to use GP regression, with compound sphericity in the high ρ setting it is efficient to use CRN. Within both of these regimes, it is doubly beneficial to revisit old seeds as implied by both Lemmas 2 and 3. Therefore, the Knowledge Gradient with Pairwise Sampling combines an acquisition procedure that can only sample new seeds with a model of differences for which it is efficient to only sample old seeds. From a value of information perspective, both serial and batch modes of $\mathrm{KG}^{\mathrm{PW}}$ yield equal or lower value of information than sequential allocation by $\mathrm{KG}^{\mathrm{CRN}}$.

LEMMA 6. *Let D^n be a dataset of observation triplets. For a Gaussian process with a kernel of the form $k_{\bar{\theta}}(x,x') + \delta_{ss'}k_{\epsilon}(x,x')$, the expected increase in value after two steps allocated according to KG^{CRN} is at least as big as two steps allocated according to KG^{PW},*

$$\mathbb{E}_n\left[\max_{x'}\mu^{n+2}(x',0) - \max_{x''}\mu^{n}(x'',0)\middle|(x,s)^{n+1},(x,s)^{n+2}\sim KG^{CRN}\right]$$
$$\geq \mathbb{E}_n\left[\max_{x'}\mu^{n+2}(x',0) - \max_{x''}\mu^{n}(x'',0)\middle|(x,s)^{n+1},(x,s)^{n+2}\sim KG^{PW}\right]$$

The proof given in EC.1.4 relies on three sub-optimal aspects of KG^{PW}, (i) all samples are restricted to new seeds, (ii) the batch mode pre-allocates two samples that would be better allocated sequentially, and (iii) KG^{PW} uses a lower bound instead of the true value of information.

Instead, we make explicit the domain for the objective function as both a solution x and a seed s and build a surrogate model and acquisition procedure over the same space. This approach has many advantages. Firstly there is no need to consider batches/pairs, reducing the dimensionality of the search space for the acquisition, thus reducing the cost per call to the acquisition function, and increasing value of information. Secondly the structure in the noise, as modeled by the difference functions, can be more aggressively exploited allocating budget to either a few seeds or many new seeds as necessary. Thirdly, the GP model allows a user to replace KG with any multi-fidelity/multi-information source (Huang et al. 2006a, Poloczek et al. 2017) or 'correlation aware' serial acquisition procedure and a corresponding parallel batch acquisition function is not required.

On the other hand, when enabling resampling of old seeds, assuming compound sphericity incentivises sampling of old seeds. KG^{CRN} includes bias functions enabling accurate modeling and the appropriate trade-off between old and new seeds. KG^{PW} does not encounter such pitfalls as it does not sample old seeds.

7. Numerical Experiments

We perform three sets of experiments, first using synthetic GP sample functions and known hyper-parameters, allowing perfect comparison of just the acquisition procedures. The next two problems are taken from the SimOpt library (`http://simopt.org`), the Assemble-to-order problem (ATO) and the Ambulances in a Square problem (AIS). The code for all experiments is available at `https://bayesianblog.com/BO-CRN/`.

7.1. Compared Algorithms and Variants

We aim to investigate the empirical effects of including bias functions and the ability of the acquisition procedure to revisit old seeds whilst holding all other experimental factors constant. Therefore we consider the following five algorithms.

Knowledge Gradient (KG): A GP model with independent homoskedastic noise is fitted, $\eta^2 = \sigma_b^2 = 0$, $\sigma_w^2 > 0$. Acquisition is according to KG^{CRN} artificially constrained to a new seed.

KG with Pairwise Sampling ($\mathbf{KG^{PW}}$): Proposed by Xie et al. (2016). A GP with the compound spheric differences kernel is fitted $\sigma_b^2 = 0$, $\eta^2, \sigma_w^2 \geq 0$. For acquisition, the value of a single sample is given by KG^{CRN} and pairs by KG^{PW}, both are constrained to a new seed.

KG with Pairwise Sampling and Bias Functions ($\mathbf{KG^{PW}}$-bias): A GP with both offsets and bias functions is fitted, $\sigma_b^2, \eta^2, \sigma_w^2 \geq 0$. Acquisition is the same as above.

KG for Common Random Numbers with Compound Sphericity ($\mathbf{KG^{CRN}}$-CS): A GP with $\sigma_b^2 = 0$ and $\eta^2, \sigma_w^2 \geq 0$ is fitted. Acquisition can sample any seed according to KG^{CRN}.

KG for Common Random Numbers ($\mathbf{KG^{CRN}}$): A GP with both offsets and bias functions is fitted, $\sigma_b^2, \eta^2, \sigma_w^2 \geq 0$. Acquisition can sample any seed according to KG^{CRN}.

In Xie et al. (2016), it was shown that the Industrial Strength Compass (Xu et al. 2010) performed significantly worse than Pairwise KG in similar settings as in our paper, hence we do not consider it here.

7.2. Synthetic Data, no Bias Functions

We set $X = \{1, .., 100\}$ and generate synthetic data from a multivariate Gaussian $\bar{\theta}(X) \sim N(\underline{0}, k_{\bar{\theta}}(X, X))$ where $k_{\bar{\theta}}(x, x') = 100^2 \exp\left(-\frac{(x-x')^2}{2 \cdot 5^2}\right)$. The offsets are sampled $o_s \sim N(0, \rho 50^2)$ and the white noise $w_s(x) \sim N(0, (1-\rho)50^2)$. We vary $\rho \in \{0, 0.1, ..., 0.9, 1.0\}$ holding the total noise constant such that standard KG will always perform the same. We compare normal KG, KG^{PW} and KG^{CRN} all without bias functions. For each method, we evaluate the KG by Equation 22 and set $A = X$. We optimize the acquisition function by exhaustive search. In all cases we fit the GP regression model with known kernel hyperparameters except for KG where we force $\rho = 0$. This

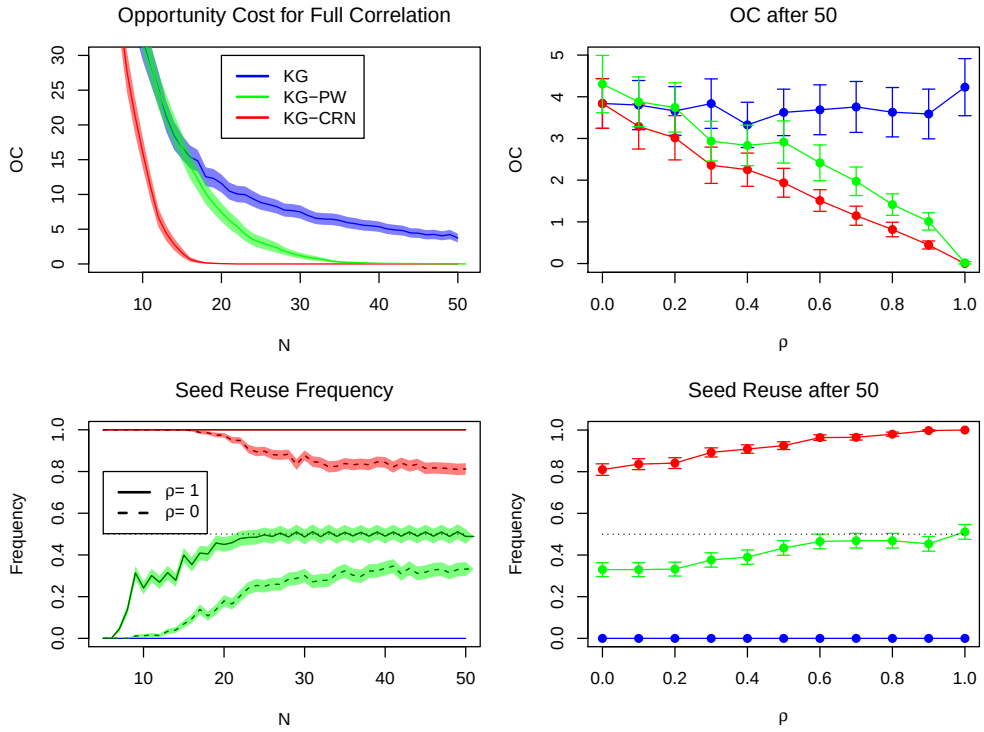

Figure 3 (top left) Opportunity Cost for the $\rho = 1$ case, the $\rho = 0$ case all algorithms equal KG. KG^{CRN} aggressively optimizes a single seed. (top right) final OC for a range of ρ values. For increasing ρ both CRN methods improve. (bottom left) the average seed reuse for the cases $\rho = 0, 1$. For large ρ, KG^{PW} is upper bounded by 0.5, KG^{CRN} never samples a new seed. (bottom right) final seed reuse over a range of ρ.

allows us to focus only on differences in the generative model and acquisition function. We measure opportunity cost at iteration n

$$OC^n = \max \bar{\theta}(x) - \bar{\theta}(x_r^n). \tag{29}$$

where $x_r^n = \arg\max_x \mu^n(x, 0)$. We report the frequency of seed reuse, how often at an iteration n the next sampled seed s^{n+1} was in the current history of observed seeds S^n. If KG^{PW} samples a pair for every iteration, the first sample of each pair would be new and the second would be old hence the average reuse frequency is upper bounded by 0.5.

From top row plots of Figure 3, for low ρ values, all algorithms have similar opportunity cost as there is no exploitable CRN structure. As ρ increases there is more CRN structure to exploit and KG^{PW} performance improves for larger budgets while KG^{CRN} performance improves for all budgets.

The bottom row plots of Figure 3 show seed reuse which we interpret as how much an algorithm uses CRN. For all ρ, KG^{CRN} starts by resampling old seeds, utilizing CRN, and later samples more new seeds only for low ρ, seed reuse dropping to 0.8, or querying new seeds 20% of the time. We see that this results in significantly faster convergence in the $\rho = 1$ case plotted.

KG^{PW} instead starts by sampling singles on new seeds, ignoring CRN and reproducing KG. For larger budgets KG^{PW} uses more pairs and improves upon KG for the range of ρ. However in the best case for CRN, $\rho = 1$, KG^{PW} quickly hits its seed reuse upper bound of 0.5, querying new seeds 50% of the time, and cannot fully utilize CRN.

In the Electronic Companion EC.2, we present the same experiment using only bias functions, and observe no improvement over standard KG, suggesting that local differences correlation is not as beneficial as global, i.e. constant, correlation.

7.3. Assemble to Order Benchmark

The Assemble to Order (ATO) simulator was introduced by Xu et al. (2010) and a slightly modified version was used in (Xie et al. 2016) to test the KG^{PW} algorithm and show that KG^{PW} outperforms other well-known simulation optimization methods such as COMPASS Xu et al. (2010). A shop sells five products assembled from eight items held in inventory. A random stream of customers arrives into the shop, each buying a product and consuming inventory. When an item in inventory drops below a user defined threshold, an order for more is placed. The shop aims to maximize profit, product sales minus storage cost, by optimizing the reorder thresholds for each item. A seed defines the stream of customers and the item delivery times. For this problem, the solution space is $X = \{1, .., 20\}^8$.

$\text{KG}_n^{\text{CRN}}(x, s)$ is evaluated and optimized as described in Section 5.2. The expectation of the maximizations within $\text{KG}^{\text{PW}}(x^{n+1}, x^{n+2})$ is evaluated exactly the same way and the function is optimized in two ways. First, x^{n+1} is found using KG^{CRN} on the new seed. $\text{KG}^{\text{PW}}(x^{n+1}, x^{n+2})$ is then optimized over X for x^{n+2} only with the same multi-start gradient ascent optimizer. Second, including the best pair so far as one start, we use multi-start gradient ascent over the full $X \times X$.

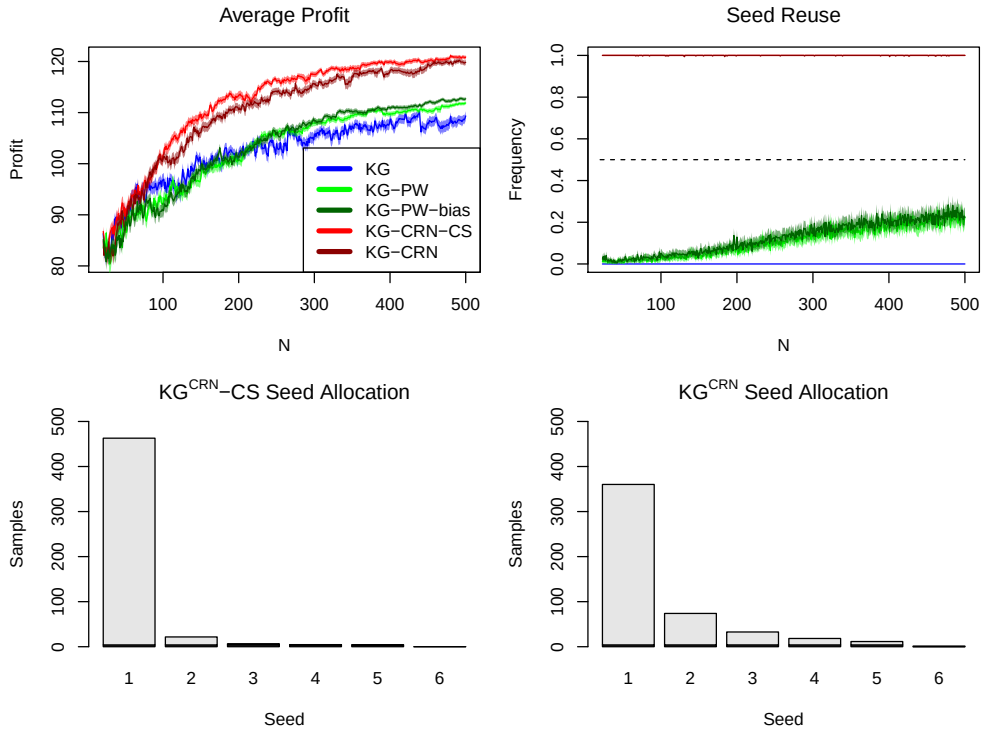

Figure 4 Top left: profit of x_r^N evaluated on a held-out set of 2,000 test seeds. Top right: average seed reuse over iterations. Bottom: seed allocation for KG^{CRN} without bias functions (left) and with bias functions (right) with the 5 initialization seeds sorted in order of decreasing sample size. Both KG^{CRN} variants mostly sample a single seed.

All methods start with $n_{init} = 20$. All hyperparameters are learnt by maximum likelihood and fine tuned after each new sample. We record the quality of the recommended $x_r^n = \text{argmax}_x \mu^n(x, 0)$ on a held-out test set of seeds. ATO results are reported in Figure 4.

Both algorithms with KG^{CRN} acquisition yield the largest profits and the KG^{PW} variants marginally improve upon KG. In this application, the KG^{CRN} variants *never* use new seeds after the initial five seeds, instead allocating almost all budget to a single seed suggesting that this ATO problem strongly benefits from reuse of seeds. This is also consistent with Figure EC.8 in the Appendix which shows that response surfaces for different seeds mostly differ by an offset, but have no significant influence on the optimizer.

From the previous experiment we observed that KG^{CRN} samples old seeds early and moves onto new seeds for large budgets. In this learnt hyperparameter case, as reported in the Electronic

Companion EC.2, the offset hyperparameter, η^2, grows over time as model fit improves and data collection focuses on the peak. Consequently, for larger budgets KG^{CRN} is even more likely to resample old seeds. With KG^{PW}, the early behavior samples singles (as opposed to pairs) on new seeds which cannot inform any CRN hyperparameters and the algorithm never learns a larger offset parameter. As a result it allocates very little of the budget to pairs failing to significantly exploit the CRN structure and hence producing marginally superior results to KG. In this application, the ability to revisit old seeds clusters observations on fewer seeds which allows for more robust learning of CRN hyperparameters.

7.4. Ambulances in a Square Problem

This simulator (AIS) was introduced by Pasupathy and Henderson (2006). Given a city over a 30km by 30km square, one must optimize the location of three ambulance bases to reduce the journey time to patients that appear across the city as a Poisson point process. The seed defines the times and locations of patients. The solution space is $X = [0, 30]^6$, the valid (x,y) locations for each of three ambulance bases. We run the simulator for 1800 simulated time units in which on average 30 patients appear. This problem is over a continuous search space and the optimal result for each realization of patients is to place the ambulance bases near the patients. Hence the peak x of one seed is not the same as the average of seeds and bias functions are required. Results are summarized in Figure 5

Both algorithms with the surrogate model that includes bias functions provide the best results in this benchmark, marginally improving upon KG. The $\text{KG}^{\text{CRN}} - \text{CS}$ algorithm that has the compound sphericity assumption in a continuous search space leads to excessive sampling of observed seeds agreeing with Lemma 3 and the conjectured behaviour of KG^{CRN} acquisition. Our proposed KG^{CRN} with bias functions on the other hand does not suffer and automatically queries many new seeds. Again, both KG^{PW} variants sample far more seeds which is less penalized in this benchmark.

We also performed experiments where the sum of ambulance journey times was optimized and where the number of patients was fixed. All results, including ATO, are summarized in Table 1. In

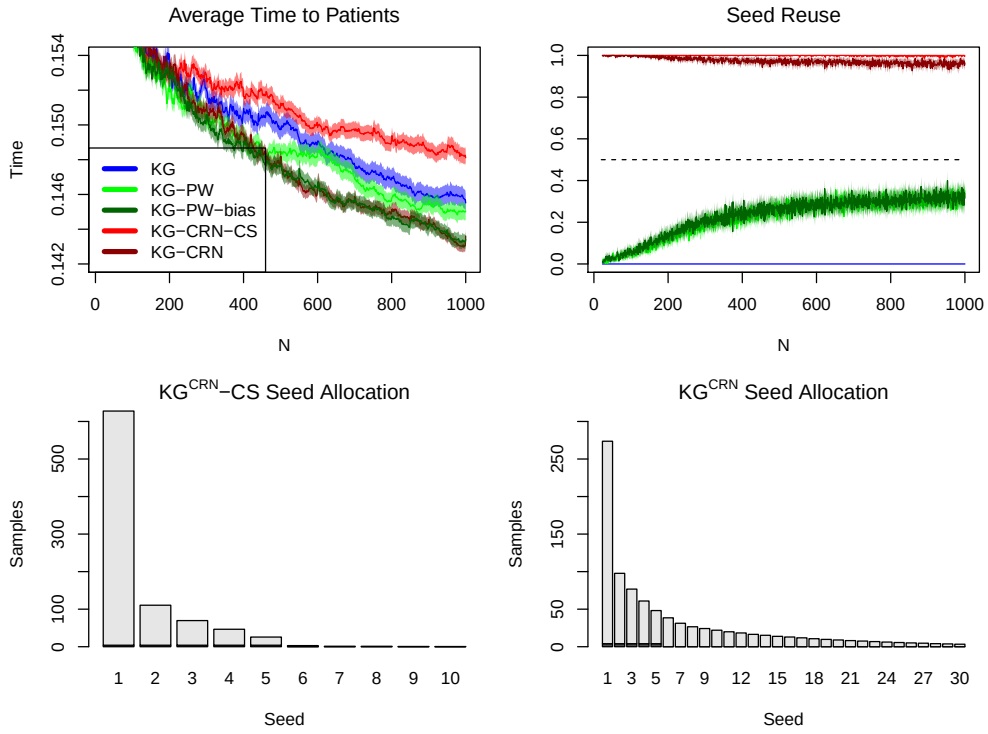

Figure 5 Top left: average journey time to patients. Top right: seed reuse over iterations. Bottom: seed allocation by KG^{CRN} without (left) and with (right) bias functions, with the 5 initialization seeds sorted in order of decreasing sample size. The algorithms with bias functions provide the best results. The compound spheric assumption, which is violated in this benchmark, leads to greedy sampling of observed seeds and sub optimal performance.

all experiments, the KG^{CRN}-CS without bias functions never sampled a new seed. In the Electronic Companion EC.2 we also report running time of all experiments and in all cases KG was quickest, followed by the KG^{CRN} variants and the KG^{PW} variants used the most computational time.

Therefore both, the ability to revisit old seeds and the modelling of bias functions, are necessary to make a robust algorithm that works across a variety of problems.

From the synthetic experiments, we observe that offsets alone yield a big benefit of using CRN while bias functions alone do not. Note that offsets are a natural phenomenon in many simulators. A single call to the ATO simulator generates a stream of customers, simulates stock levels for a fixed time period and then returns the profits summed over customers. Regardless of stock level, a stream with more customers yields larger profits, a global positive offset, than a stream with

Table 1 Mean ± 2 standard errors of average performance for all benchmarks, results that do not significantly differ from the best are in bold. The ability to revisit seeds improves the ATO results and including bias functions improves AIS results (or compound sphericity significantly harms AIS). $N = 500$ unless specified otherwise.

	KG	KG^{PW}	KG^{PW}-bias	KG^{CRN}-CS	KG^{CRN}
ATO	109.35 ± 1.88	111.86 ± 0.65	112.69 ± 0.67	**120.99 ± 0.71**	**119.84 ± 1.13**
AIS	$.1498 \pm .0011$	**.1483 ± 0.0010**	**.1477 ± .0010**	$.1512 \pm .0010$	**.1482± .0010**
AIS, N=1000	$.1455 \pm .0010$	$.1450 \pm 0.0010$	**.1435 ± .0009**	$.1481 \pm .0009$	**.1436 ± .0008**
AIS, sum time	4.66 ± 0.33	$4.611 \pm .045$	**4.449± .030**	$4.515 \pm .035$	**4.430 ± .034**
AIS, 30 patients	$.1498 \pm .0009$	**.1468 ± .0008**	**.1467 ± .0009**	.1482 ± .0008	**.1467 ± .0009**

fewer customers, a negative offset. The AIS simulator generates a stream of patients, and simulates ambulances driving for a fixed time period. It then returns the average per patient waiting time. Similarly, a stream with more patients will result in longer average waiting times for any ambulance locations, a positive offset, while a stream with fewer patients would yield lower times, a negative offset. In both ATO and AIS we see a benefit of CRN.

To explore this hypothesis, we consider two modifications of the AIS simulator. The first aims to increase CRN benefit: Instead of the mean journey time, the simulator returns the sum of journey times. Given two streams, one with many patients and one with few patients, the difference between mean journey times is purely due to crowding causing journey delays while the difference in the sum of journey times is due to crowding as well as the summations containing more patients. Consequently, we observe a much greater difference between the default AIS and the AIS-sum as shown in Table 1.

The second modification changes the stopping criterion of the AIS simulator such that each simulation runs until 30 patients have been visited. As can be seen in Table 1, in this case KG^{CRN} has no significant benefit over KG^{PW}.

8. Conclusion

We proposed a Bayesian approach to simulation optimization with common random numbers where the seed of the random number generator used within a stochastic objective function is an input to

be chosen by the optimization algorithm. We augment a standard Gaussian process model with two extra hyperparameters to model structured noise (seed/scenario influence), while maintaining the ability to predict the average output of the target function in closed form. Matching this augmented model, we propose KG^{CRN} that quantifies the benefit of evaluating the objective for a given solution and seed, providing a clean framework that allows Bayesian optimization to automatically exploit CRN where this is beneficial, and recovers standard KG where not. The proposed KG^{CRN} algorithm structure does not add significant computational burden over the equivalent non-CRN Knowledge Gradient due to the fundamental structure of CRN.

In this work we focus on global optimization, in future work we plan to augment other problem settings with common random numbers, such as multi-fidelity optimization, simulations with input uncertainty, and multi-objective optimization. The KG^{CRN} algorithm can also be extended to batch acquisition, e.g., using the technique of Wu and Frazier (2016), heteroscedastic noise settings, and to account for unequal simulation cost (e.g. by caching precomputed random number streams) using the method of Poloczek et al. (2017).

References

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

Chen X, Ankenman BE, Nelson BL (2012) The effects of common random numbers on stochastic kriging metamodels. *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 22(2):7.

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

Çınlar E (2011) *Probability and stochastics*, volume 261 (Springer Science & Business Media).

Frazier P (2012) Tutorial: Optimization via simulation with bayesian statistics and dynamic programming. *Proceedings of the 2012 Winter Simulation Conference (WSC)*, 1–16 (IEEE).

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

Fu MC, Hu JQ, Chen CH, Xiong X (2004) Optimal computing budget allocation under correlated sampling. *Proceedings of the 2004 Winter Simulation Conference*, volume 1 (IEEE).

Ghosal S, Roy A, et al. (2006) Posterior consistency of gaussian process prior for nonparametric binary regression. *The Annals of Statistics* 34(5):2413–2429.

Görder B, Kolonko M (2019) Ranking and selection: A new sequential bayesian procedure for use with common random numbers. *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 29(1):2.

Huang D, Allen TT, Notz WI, Miller RA (2006a) Sequential kriging optimization using multiple-fidelity evaluations. *Structural and Multidisciplinary Optimization* 32(5):369–382.

Huang D, Allen TT, Notz WI, Zeng N (2006b) Global optimization of stochastic black-box systems via sequential kriging meta-models. *Journal of global optimization* 34(3):441–466.

Jones DR, Schonlau M, Welch WJ (1998) Efficient global optimization of expensive black-box functions. *Journal of Global optimization* 13(4):455–492.

Kandasamy K, Krishnamurthy A, Schneider J, Póczos B (2018) Parallelised bayesian optimisation via thompson sampling. *International Conference on Artificial Intelligence and Statistics*, 133–142.

Kim SH (2013) Statistical ranking and selection. *Encyclopedia of Operations Research and Management Science* 1459–1469.

Nelson BL, Matejcik FJ (1995) Using common random numbers for indifference-zone selection and multiple comparisons in simulation. *Management Science* 41(12):1935–1945.

Pasupathy R, Henderson SG (2006) A testbed of simulation-optimization problems. *Proceedings of the 2006 winter simulation conference*, 255–263 (IEEE).

Pearce M, Branke J (2017) Bayesian simulation optimization with input uncertainty. *2017 Winter Simulation Conference (WSC)*, 2268–2278 (IEEE).

Picheny V (2015) Multiobjective optimization using gaussian process emulators via stepwise uncertainty reduction. *Statistics and Computing* 25(6):1265–1280.

Poloczek M, Wang J, Frazier P (2017) Multi-information source optimization. *Advances in Neural Information Processing Systems*, 4288–4298.

Rasmussen CE (2003) Gaussian processes in machine learning. *Summer School on Machine Learning*, 63–71 (Springer).

Salemi PL, Song E, Nelson BL, Staum J (2019) Gaussian markov random fields for discrete optimization via simulation: Framework and algorithms. *Operations Research* 67(1):250–266.

Scott W, Frazier P, Powell W (2011) The correlated knowledge gradient for simulation optimization of continuous parameters using gaussian process regression. *SIAM Journal on Optimization* 21(3):996–1026.

Srinivas N, Krause A, Kakade SM, Seeger M (2009) Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995* .

Swersky K, Snoek J, Adams RP (2013) Multi-task Bayesian optimization. *Advances in Neural Information Processing Systems*, 2004–2012.

Toscano-Palmerin S, Frazier PI (2018) Bayesian optimization with expensive integrands. *arXiv preprint arXiv:1803.08661* .

Villemonteix J, Vazquez E, Walter E (2009) An informational approach to the global optimization of expensive-to-evaluate functions. *Journal of Global Optimization* 44(4):509.

Wu J, Frazier P (2016) The parallel knowledge gradient method for batch bayesian optimization. *Advances in Neural Information Processing Systems*, 3126–3134.

Wu J, Poloczek M, Wilson AG, Frazier P (2017) Bayesian optimization with gradients. *Advances in Neural Information Processing Systems*, 5267–5278.

Xie J, Frazier PI, Chick SE (2016) Bayesian optimization via simulation with pairwise sampling and correlated prior beliefs. *Operations Research* 64(2):542–559.

Xu J, Nelson BL, Hong J (2010) Industrial strength compass: A comprehensive algorithm and software for optimization via simulation. *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 20(1):3.

E-Companion: Proofs and Additional Experiments

EC.1. Proofs of Statements

EC.1.1. Estimating the Target

The data collected and the surrogate model are over the domain $X \times \mathbb{N}^+$ whereas the target of optimization is a function over X. In what follows, we show how to derive an estimate for the target. This result is an immediate consequence of the symmetry of the model across unobserved seeds proven in Lemma EC.1. As a result of this symmetry, when taking the limit of the sum over infinite seeds, unobserved seeds dominate this sum yielding a simple form of the GP posterior for the target. This result is consistent with other CRN and non-CRN methods that do not make the seed explicit but do incorporate off-diagonal noise covariance matrix.

LEMMA 1 *For any given kernel over the domain $X \times \mathbb{N}^+$ that is of the form $k_{\bar{\theta}}(x,x') + \delta_{ss'}k_{\epsilon}(x,x')$, and a dataset of n input-output triplets D^n, the posterior over the target is a Gaussian process given by*

$$\bar{\theta}(x)|D^n \sim GP(\mu^n_{\bar{\theta}}(x), k^n_{\bar{\theta}}(x,x')), \tag{EC.1}$$

$$\mu^n_{\bar{\theta}}(x) = \mu^n(x,s'), \tag{EC.2}$$

$$k^n_{\bar{\theta}}(x,x') = k^n(x,s',x',s''), \tag{EC.3}$$

where $s', s'' \in \mathbb{N}^+ \setminus S^n$ with $s' \neq s''$ any two unobserved unequal seeds.

Recall that the Gaussian process is defined over the domain $X \times \mathbb{N}^+$, with infinite seeds. The following result states that the Gaussian process model makes identical predictions for all the unobserved seeds.

LEMMA EC.1. *Let $\theta(x,s)$ be a realization of a Gaussian Process with $\mu^0(x,s) = 0$ and any positive semi-definite kernel of the form $k(x,s,x',s') = k_{\bar{\theta}}(x,x') + \delta_{ss'}k_{\epsilon}(x,x')$. For all $x \in X$, $s_{obs} \in S^n$, and unobserved seeds $s, s', s'' \in \mathbb{N}^+ \setminus S^n$, the posterior mean and kernel satisfy*

$$\mu^n(x,s) = \mu^n(x,s'), \tag{EC.4}$$

$$k^n(x, s_{obs}, x', s) = k^n(x, s_{obs}, x', s'), \tag{EC.5}$$

$$k^n(x, s, x', s') = k^n(x, s, x', s'') = k^n(x, s', x', s''). \tag{EC.6}$$

Proof Writing out the posterior mean in full from Equation 15 in the main paper,

$$\begin{aligned}\mu^n(x,s) &= k^0(x,s,\tilde{X}^n)K^{-1}Y^n \\ &= \begin{cases} \left(k_{\bar{\theta}}(x,X^n) + (\mathbb{1}^\intercal_{s=S^n} \circ k_\epsilon(x,X^n))\right)K^{-1}Y^n & s \in S^n \\ k_{\bar{\theta}}(x,X^n)K^{-1}Y^n & s \in \mathbb{N}^+ \setminus S^n \end{cases}\end{aligned}$$

where $a \circ b$ is element-wise product and $\mathbb{1}_{s=S^n} \in \{0,1\}^n$ is a binary masking column vector that is zero for all $s \in \mathbb{N}^+ \setminus S^n$. The proofs for Equations EC.5 and EC.6 follow similarly from Equation 16 in the main paper. □

We next prove the main lemma. Keep in mind that the target for the optimization is the infinite average over seeds, and the Gaussian process model makes identical predictions for unobserved seeds. The infinite average is dominated by unobserved seeds with identical predictions. Hence we may simply use the prediction of any one unobserved seed as a model for the infinite average/target.

Proof of Lemma 1 The target of optimization, $\bar{\theta}(x)$, is given by the average output over infinitely many seeds which may be written as the limit

$$\bar{\theta}(x) \;=\; \lim_{N_s \to \infty} \frac{1}{N_s} \sum_{s=1}^{N_s} \theta(x,s). \tag{EC.7}$$

Adopting the shorthand $\mathbb{E}_n[...] = \mathbb{E}[...|D^n]$, we first consider the posterior expected performance,

$$\mathbb{E}_n[\bar{\theta}(x)] = \mathbb{E}_n\left[\lim_{N_s \to \infty} \frac{1}{N_s} \sum_{s=1}^{N_s} \theta(x,s)\right] \tag{EC.8}$$

$$= \lim_{N_s \to \infty} \frac{1}{N_s} \sum_{s=1}^{N_s} \mathbb{E}_n\left[\theta(x,s)\right] \tag{EC.9}$$

$$= \lim_{N_s \to \infty} \frac{1}{N_s} \sum_{s=1}^{N_s} \mu^n(x,s). \tag{EC.10}$$

Let $n_s = \max\{S^n\}$ be the largest observed seed. The sum of posterior means can be split into sampled seeds $s \in \{1, ..., n_s\}$ and unsampled seeds $s \in \{n_s + 1, ..., N_s\}$,

$$\mathbb{E}_n[\bar{\theta}(x)] = \lim_{N_s \to \infty} \frac{1}{N_s} \left(\sum_{s=1}^{n_s} \mu^n(x,s) + \sum_{s'=n_s+1}^{N_s} \mu^n(x,s') \right) \tag{EC.11}$$

$$= \lim_{N_s\to\infty} \frac{1}{N_s}\left(\sum_{s=1}^{n_s} \mu^n(x,s) + (N_s - n_s)\mu^n(x,n_s+1)\right) \tag{EC.12}$$

$$= \lim_{N_s\to\infty} \frac{1}{N_s}\left(\sum_{s=1}^{n_s} \mu^n(x,s) - n_s\mu^n(x,n_s+1)\right) + \mu^n(x,n_s+1) \tag{EC.13}$$

$$= \mu^n(x,n_s+1), \tag{EC.14}$$

where we have used Lemma EC.1 to simplify. Similarly for the covariance, writing each $\bar{\theta}(x)$ term as the limit of a sum over seeds,

$$\mathbb{E}_n\left[\left(\bar{\theta}(x) - \mathbb{E}_n[\bar{\theta}(x)]\right)\left(\bar{\theta}(x') - \mathbb{E}_n[\bar{\theta}(x')]\right)\right] \tag{EC.15}$$

$$= \mathbb{E}_n\left[\left(\lim_{N_s\to\infty} \frac{1}{N_s}\sum_{s=1}^{N_s} \theta(x,s) - \mu(x,s)\right)\left(\lim_{N_t\to\infty} \frac{1}{N_t}\sum_{s'=1}^{N_t} \theta(x',s') - \mu(x',s')\right)\right] \tag{EC.16}$$

$$= \lim_{N_s,N_t\to\infty} \frac{1}{N_sN_t}\sum_{s,s'=1}^{N_s,N_t} \mathbb{E}_n\left[\left(\theta(x,s) - \mu(x,s)\right)\left(\theta(x',s') - \mu(x',s')\right)\right] \tag{EC.17}$$

$$= \lim_{N_s,N_t\to\infty} \frac{1}{N_sN_t}\sum_{s,s'=1}^{N_s,N_t} k^n(x,s,x's'). \tag{EC.18}$$

The domain in the limit of the summation, $\mathbb{N}^+ \times \mathbb{N}^+$, is unaffected by setting $N_t = N_s$. The summation decomposes into four terms,

$$\begin{aligned}\sum_{s,s'=1}^{N_s} k^n(x,s,x',s') = &\underbrace{\sum_{s,s'=1}^{n_s} k^n(x,s,x',s')}_{\text{observed seeds full covariance}} + \underbrace{\sum_{s'=n_s+1}^{N_s}\sum_{s=1}^{n_s} k^n(x,s,x',s')}_{\text{observed-unobserved covariance}} \\ &+ \underbrace{\sum_{s=n_s+1}^{N_s} k^n(x,s,x',s)}_{\text{unobserved seeds variance}} + \underbrace{\sum_{n_s<s\neq s'\leq N_s} k^n(x,s,x',s')}_{\text{unobserved seeds covariance}} \\ = &\underbrace{\sum_{s,s'=1}^{n_s} k^n(x,s,x',s')}_{\text{constant with } N_s} + \underbrace{2(N_s-n_s)\sum_{s=1}^{n_s} k^n(x,s,x',s')}_{\text{linear with } N_s} \\ &+ \underbrace{(N_s-n_s)k^n(x,s',x',s')}_{\text{linear with } N_s} + \underbrace{(N_s-n_s)^2k^n(x,s',x',s'')}_{\text{quadratic with } N_s},\end{aligned}$$

where s' and s'' are two unequal unobserved seeds. Dividing the final equation by N_s^2 and taking the limit $N_s \to \infty$, only the final term remains. □

Given the assumed kernel with independent and identically distributed difference functions, the average of infinitely many seeds includes finite observed seeds and infinitely many identical

unobserved seeds. Unobserved seeds dominate the infinite average and the performance under any unobserved seed is an estimator for the objective function. Likewise the posterior covariance between infinite averages is the posterior covariance between any two unique unobserved seeds. Also note that the kernel of the Gaussian process prior for the objective evaluated at different seeds returns the prior kernel for the target $\bar{k}^0(x,x') = k^0(x,1,x',2) = k_{\bar{\theta}}(x,x')$ as desired.

EC.1.2. Proof of Theorem 1

We next show that, under certain assumptions on the target function, given an infinite sampling budget, $N \to \infty$, the $\mathrm{KG}^{\mathrm{CRN}}$ algorithm will discover the true optimum. We first restate the result.

THEOREM 1 *Let $x_r^N \in A$ be the point that KG^{CRN} recommends in iteration N. For each $p \in [0,1)$ there is a constant K_p such that with probability p*

$$\lim_{N\to\infty} \bar{\theta}(x_r^N) > \bar{\theta}(x^{OPT}) - K_p d.$$

We first prove properties of the $\mathrm{KG}_n^{\mathrm{CRN}}(x,s)$ function and then consider the error due to discretization.

Lemma EC.2 ensures the GP model exists in the limit of infinite data. We then show that $\mathrm{KG}_n^{\mathrm{CRN}}(x,s)$ is non-negative in Lemma EC.3 and that it is zero for sampled input pairs in Lemma EC.4. We then show that if a single x is sampled for infinitely many (not necessarily consecutive) seeds, again $\mathrm{KG}_n^{\mathrm{CRN}}(x,s)$ tends to zero also for all unevaluated seeds in Lemma EC.5. Then in Lemma EC.6 we show the opposite direction, if $\mathrm{KG}_n^{\mathrm{CRN}}(x,s)$ is zero, this implies that the peak of the target prediction will not change by sampling (x,s). This is extended in Lemma EC.7 that states that if for a new seed s, $\mathrm{KG}_n^{\mathrm{CRN}}(x,s) = 0$ for all X then no more samples will change the peak prediction of the target and the true peak is known when X is a discrete set.

The error due to discretization relies on the assumption of a differentiable GP kernel, such as Matérn, and using a Lipschitz continuity argument, the error may be bounded proving Theorem 1. The following result simply states that the GP model exists in the limit of infinite data. First we define $V^n(x,x') = \mathbb{E}_n[\bar{\theta}(x)\bar{\theta}(x')]$.

LEMMA EC.2. *Let $x, x' \in X$. Then the limits of the series $(\bar{\mu}^n(x))_n$ and $(V^n(x,x'))_n$ exist and are denoted by $\bar{\mu}^\infty(x)$ and $V^\infty(x,x')$, respectively. Then we have*

$$\lim_{n\to\infty} \bar{\mu}^n(x) = \bar{\mu}^\infty(x) \tag{EC.19}$$

$$\lim_{n\to\infty} V^n(x,x') = V^\infty(x,x') \tag{EC.20}$$

almost surely.

Proof $\bar{\theta}(x)$ and $\bar{\theta}(x)\bar{\theta}(x')$ are integrable random variables for all $x, x' \in X$ by choice of $\bar{\theta}$. Proposition 2.7 in Çınlar (2011) states that any sequence of conditional expectations of an integrable random variable under an increasing filtration is uniformly integrable martingale. Thus, both sequences converge almost surely to their respective limit. □

The next result states the $\mathrm{KG}^{\mathrm{CRN}}(x,s)$ is non-negative for all input pairs.

LEMMA EC.3. *$KG_n^{CRN}(x,s) \geq 0$ holds for all $(x,s) \in X \times \mathbb{N}^+$.*

Proof Adopting the shorthand $x_r^n = \mathrm{argmax}_{x\in X}\mu^n(x,0)$, we may write $\max_x \mu^n(x,0) = \mu^n(x_r^n,0)$ and

$$\begin{aligned}
\mathrm{KG}_n^{\mathrm{CRN}}(x,s) &= \mathbb{E}\left[\max_{x'\in X}\{\mu^n(x',0) + \tilde{\sigma}^n(x',0;x,s)Z\} - \mu^n(x_r^n,0)\right] \\
&= \mathbb{E}\left[\max_{x'\in X}\{\mu^n(x',0) + \tilde{\sigma}^n(x',0;x,s)Z\} - \mu^n(x_r^n,0)\right] - \underbrace{\mathbb{E}[cZ]}_{=0} \\
&= \mathbb{E}\left[\max_{x'\in X}\{\mu^n(x',0) + (\tilde{\sigma}^n(x',0;x,s) - c)Z\} - \mu^n(x_r^n,0)\right]
\end{aligned}$$

where the expectation is over $Z \sim N(0,1)$ and c is an arbitrary constant. In particular, by setting $c = \tilde{\sigma}^n(x_r^n,0;x,s)$, the inner expression, when evaluated at $x_r^n \in X$, satisfies

$$\mu^n(x_r^n,0) + (\tilde{\sigma}^n(x_r^n,0;x,s) - \tilde{\sigma}^n(x_r^n,0;x,s))Z - \mu^n(x_r^n,0) = 0$$

for all $Z \in \mathbb{R}$ and

$$\max_{x'\in X}\{\mu(x',0) + (\tilde{\sigma}^n(x',0;x,s) - c)Z - \mu_0\} \;\geq\; \mu(x_r^n,0) + (\tilde{\sigma}^n(x_r^n,0;x,s) - c)Z - \mu^n(x_r^n,0) \;=\; 0$$

for all Z and $\mathrm{KG}_n^{\mathrm{CRN}}(x,s)$ may be written as the expectation of a non-negative random variable.

□

The following result states that once an input pair (x,s) has been observed, $\theta(x,s)$ is known and $\mathrm{KG}_n^{\mathrm{CRN}}(x,s)$ is zero. Combined with the result that $\mathrm{KG}_n^{\mathrm{CRN}}(x,s)$ is non-negative, it follows that observed input pairs (x,s) are minima of the function $\mathrm{KG}_n^{\mathrm{CRN}}(x,s)$.

LEMMA EC.4. *Given deterministic simulation outputs, there is no improvement in re-sampling a sampled point.*

$$KG_n^{CRN}(x^i,s^i)=0$$

for all $(x^i,s^i)\in\tilde{X}^n$.

Proof The posterior covariance between the output at any point and the output at an observed point is zero, writing out the full matrix multiplication for the posterior kernel and simplifying yields

$$\begin{aligned}
k^n(x^i,s^i;x,s) &= k^0(x^i,s^i;x,s)-k^0(x^i,s^i;\tilde{X}^n)\left(k^0(\tilde{X}^n;\tilde{X}^n)\right)^{-1}k^0(\tilde{X}^n;x,s)\\
&= k^0(x^i,s^i;x,s)-\left[k^0(\tilde{X}^n;\tilde{X}^n)\right]_i\left(k^0(\tilde{X}^n;\tilde{X}^n)\right)^{-1}k^0(\tilde{X}^n;x,s)\\
&= k^0(x^i,s^i;x,s)-\mathbb{1}_i^{n\intercal}k^0(\tilde{X}^n;x,s)\\
&= k^0(x^i,s^i;x,s)-k^0(x^i,s^i;x,s)\\
&= 0
\end{aligned}$$

where $[\cdot]_i$ is the i^{th} row. The second line contains the i^{th} row of a matrix multiplied by its inverse returning the i^{th} row of the identity matrix denoted $\mathbb{1}_i^{n\intercal}$. Therefore $\tilde{\sigma}^n(x,s;x^i,s^i)=0$ for all (x,s) and $\mathrm{KG}_n^{\mathrm{CRN}}(x,s)=0$. $\square$

Let ω denote an arbitrary sample path, $\omega=((x,s)^1,(x,s)^2,.....)$, determining an input pair for each query to the objective as $n\to\infty$. Lemmas EC.3 and EC.4 imply sampled point inputs are minima of $\mathrm{KG}^{\mathrm{CRN}}$ and recall that according to the algorithm, new samples are allocated to maxima $(x,s)^{n+1}=\mathrm{argmax}\mathrm{KG}_n^{\mathrm{CRN}}(x,s)$. These facts together imply that no input (x,s) will be sampled more than once. We need only to consider sample paths ω where all sampled inputs pairs (x^i,s^i) are unique. Recall that we suppose a (finite) discretization of X, thus there must be an $x\in X$ that

is observed for an infinite number of seeds on ω as $n \to \infty$. We study the asymptotic behaviour $\mathrm{KG}_n^{\mathrm{CRN}}(x,s)$ for $n \to \infty$ as a function of $\mu^n(x,0)$, $\tilde{\sigma}^n(x',0,x,s)$.

If s is a new seed and x has been observed for infinitely many seeds, the next result states that $\mathrm{KG}_n^{\mathrm{CRN}}(x,s)$ tends to zero, there is less/no value in re-evaluating x for another new seed.

LEMMA EC.5. *If x is sampled for infinitely many (not necessarily consecutive) seeds, then $\tilde{\sigma}^\infty(x',0;x,s) = 0$ for all $x' \in X$ and all $s \in \mathbb{N}^+$ and $KG_\infty^{CRN}(x,s) = 0$ for all $s \in \mathbb{N}^+$ almost surely.*

Proof Setting $x^{n+1} = x$ and assuming (x^i, s^i) pairs are arranged such that s^{n+1} is always a new seed, the posterior variance reduces to zero

$$\begin{aligned}
\lim_{n\to\infty} |\tilde{\sigma}^n(x',0;x,s^{n+1})| &= \lim_{n\to\infty} \frac{|k^n(x',0,x,s^{n+1})|}{\sqrt{k^n(x,s^{n+1},x,s^{n+1})}} \\
&= \lim_{n\to\infty} \frac{\bar{k}^n(x',x)}{\sqrt{\bar{k}^n(x,x)+k_\epsilon(x,x)}} \\
&\leq \lim_{n\to\infty} \sqrt{\bar{k}^n(x',x')} \frac{\sqrt{\bar{k}^n(x,x)}}{\sqrt{\bar{k}^n(x,x)+k_\epsilon(x,x)}} \\
&= 0
\end{aligned}$$

where the final line is by noting that $\bar{k}^n(x,x) + k_\epsilon(x,x) > 0$ for all n and x. $\square$

The following result states that if there is no benefit of a new measurement for an input pair (x,s), then the change in the posterior mean, $\tilde{\sigma}^n(x',0;x,s)$ must be constant, i.e. the new sample at (x,s) will only have the effect of adding a constant to the prediction of the target, hence learning nothing about the peak of the target. The contrapositive is that for input points for which $\tilde{\sigma}^n(x',0;x,s)$ varies with x', $\mathrm{KG}^{\mathrm{CRN}}$ is strictly positive.

LEMMA EC.6. *Let (x,s) be an input pair for which $KG_n^{CRN}(x,s) = 0$. Then for all $x' \in X$*

$$\tilde{\sigma}^n(x',0;x,s) = c$$

where c is a constant.

Proof From Equation EC.21, $\mathrm{KG}_n^{\mathrm{CRN}}(x,s)$ can be written as the expectation of a non-negative random variable. Therefore the random variable itself must equate to zero almost surely implying

$$\begin{aligned}
\max_{x'\in X}\{\mu(x',0) + (\tilde{\sigma}(x',0;x,s) - c)Z - \mu^n(x_r^n,0)\} &= 0 \\
\max_{x'\in X}\{\mu(x',0) + (\tilde{\sigma}(x',0;x,s) - c)Z\} &= \max_{x''\in X}\{\mu(x'',0)\}
\end{aligned}$$

for all $Z \in \mathbb{R}$. This implies $\tilde{\sigma}(x', 0; x, s) = c$ for all $x' \in X$. $\square$

Note that in the case where $(x, s) \in \tilde{X}^n$ we have that $\tilde{\sigma}^n(x', 0; x, s) = 0$ for all $x' \in X$.

We next show that if there is no value in evaluating any input pair, then the optimizer of the target is known.

LEMMA EC.7. *Let $s \in \mathbb{N}^+ \setminus S^n$ be an unobserved seed, if $KG_n^{CRN}(x, s) = 0$ for all $x \in X$, then $argmax_x \mu^n(x, 0) = argmax_x \bar{\theta}(x)$*

Proof By Lemma EC.6, we have that $\bar{k}^n(x, x') = c$ for all $x, x' \in X$ and the covariance matrix $\bar{k}^n(X, X)$ is proportional to the all ones matrix. Hence $\bar{\theta}(x) - \mu^n(x, 0)$ is a normal random variable that is constant across all $x \in X$ and $\operatorname{argmax}_{x \in X} \mu(x, 0) = \operatorname{argmax}_{x \in X} \bar{\theta}(x)$ holds. $\square$

Lemmas EC.4, EC.5, consider evaluating $\mathrm{KG}^{\mathrm{CRN}}$ as the sampling budget increases in a specific way. More generally, recall that $\mathrm{KG}^{\mathrm{CRN}}$ picks $(x, s)^{n+1} \in \operatorname{argmax} \mathrm{KG}_n^{\mathrm{CRN}}(x, s)$ in each iteration n. Since $\theta(x, \cdot)$ is evaluated infinitely often (by choice of x), $\mathrm{KG}_n^{\mathrm{CRN}}(x, \cdot) \to 0$ for all $x \in A$ holds almost surely and by Lemma EC.7 the true optimizer is known.

Proof of Theorem 1 We give a bound on the loss due to discretization of a continuous search space. Suppose that $X \subset \mathbb{R}^d$ is compact and $A \subset X$ is a finite set of discretization points. Suppose that $\bar{\mu}^0(x) = 0$ for all x and that $k_{\bar{\theta}}(x, x')$ is a four times differentiable Matérn kernel, e.g., the popular squared exponential kernel. Moreover, suppose that $\bar{\theta}(x)$ is drawn from the prior, i.e., let $\bar{\theta}(x) \sim \mathrm{GP}(\bar{\mu}^0(x), k_{\bar{\theta}}(x, x'))$, then the sample $\bar{\theta}(x)$ from the set of functions is itself twice *continuously differentiable* in X with probability one (Ghosal et al. 2006). The extrema of $\frac{\delta}{\delta x_i} \bar{\theta}(x)$ over X are probabilistically bounded, since the derivative processes of $\bar{\theta}(x)$ are also Gaussian for our choice of $k_{\bar{\theta}}^0(x, x)$ (Ghosal et al. 2006). Let $x^{OPT} = \operatorname{argmax}_{x \in X} \bar{\theta}(x)$ and $d = \max_{x' \in X} \min_{x \in A} \operatorname{dist}(x, x')$ be the largest distance from any point in the continuous domain X to its nearest neighbor in A. We can compute for every $p \in [0, 1)$ a constant K_p such that $\bar{\theta}(x)$ is K_p Lipschitz continuous on X with probability at least p, thus there exists an $\bar{x} \in A$ with $\operatorname{dist}(\bar{x}, x^{OPT}) \leq d$ and

$$\bar{\theta}(\bar{x}) > \bar{\theta}(x^{OPT}) - K_p d$$

holds with probability p. Finally the point recommended by KG^{CRN} is the maximizer of $x_r^N \in \text{argmax}_{x\in A}\bar{\theta}(x)$ and therefore is not worse than $\bar{x}$

$$\begin{aligned}\lim_{N\to\infty} \bar{\theta}(x_r^N) &\geq \bar{\theta}(\bar{x}) \\ &\geq \bar{\theta}(x^{OPT}) - K_p d\end{aligned}$$

□

Thus, when applying the KG^{CRN} algorithm to a discretized search space, the true optimizer becomes known as the sampling budget increases without bound and if the underlying target function is continuous, the error is bounded simply due to Lipschitz continuity.

EC.1.3. Behavior in the Compound Spheric Case

We next provide proofs for Lemma 4 and Lemma 5 relating to the KG^{CRN} algorithm behaviour in the case of compound sphericity with full noise correlation. Recall this corresponds to the difference functions reducing to constant offsets and an algorithm may optimize one seed as a single seed is a deterministic function with the same optimizer as the target. This is essentially a best-case scenario for optimization with common random numbers. Lemma 2 of the main paper states that the difference $\mu^n(x,s) - \mu^n(x,s') = A_s - A_{s'}$ is constant for all x. Likewise the same relationship applies to $\tilde{\sigma}^n(x',s';x,s)$ that quantifies changes in the posterior mean and therefore must also maintain the symmetry over seeds s'. All results in this section assume $\theta(x,s)$ is a realization of a Gaussian process with the compound spheric kernel and full correlation $k_\epsilon(x,x') = \eta^2$.

The first result states that, when sampling a point (x,s), the update in the prediction for one seed differs from the update in prediction for another seed by an additive constant. Predictions for all seeds have the same shape/gradient and differ only by global constants.

LEMMA EC.8. *Let $x,x' \in X$, $s,s' \in \mathbb{N}^+$, then the difference in posterior mean updates satisfies*

$$\tilde{\sigma}^n(x',s';x,s) = \tilde{\sigma}^n(x',0;x,s) + h^n(s',x,s).$$

Proof

$$\begin{aligned}
\tilde{\sigma}^n(x',s';x,s) &= \frac{k^n(x',s';x,s)}{\sqrt{k^n(x,s,x,s)}} \\
&= \frac{1}{\sqrt{k^n(x,s,x,s)}}\left(k_{\bar{\theta}}(x',x)+\eta^2\delta_{ss'}-\left(k_{\bar{\theta}}(x',X^n)+\eta^2\mathbb{1}_{s'=S^n}^\intercal\right)K^{-1}\left(k_{\bar{\theta}}(X^n,x)+\eta^2\mathbb{1}_{s=S^n}\right)\right) \\
&= \tilde{\sigma}^n(x',0;x,s)+\underbrace{\frac{\eta^2\delta_{ss'}-\eta^2\mathbb{1}_{s'=S^n}^\intercal K^{-1}\left(k_{\bar{\theta}}(X^n,x)+\eta^2\mathbb{1}_{s=S^n}\right)}{\sqrt{k^n(x,s,x,s)}}}_{\text{independent of } x'} \\
&= \tilde{\sigma}^n(x',0;x,s)+h^n(s',x,s)
\end{aligned}$$

□

As a result of the symmetry over seeds it is possible to use any seed $s \in \mathbb{N}^+$ as the target of optimization formalized in the following Lemma.

LEMMA EC.9. *Let $x \in X$, $s, s' \in \mathbb{N}^+$, then*

$$KG_n^{CRN}(x,s) = \mathbb{E}[\max_{x'\in X}\mu^n(x',s')+\tilde{\sigma}^n(x',s';x,s)Z-\max_{x''\in X}\mu^n(x'',s')].$$

Proof

$$\begin{aligned}
\mathrm{KG}_n^{\mathrm{CRN}}(x,s) &= \mathbb{E}[\max_{x'\in X}\mu^n(x',0)+\tilde{\sigma}^n(x',0;x,s)Z-\max_{x''\in X}\mu^n(x'',0)] \\
&= \mathbb{E}[\max_{x'\in X}\mu^n(x',s')-A_{s'}+(\tilde{\sigma}^n(x',s';x,s)-h(s',x,s))Z-\max_{x''\in X}\mu^n(x'',s')-A_{s'}] \\
&= \mathbb{E}[\max_{x'\in X}\mu^n(x',s')+\tilde{\sigma}^n(x',s';x,s)Z-\max_{x''\in X}\mu^n(x'',s')]-h(s',x,s)\mathbb{E}[Z] \\
&= \mathbb{E}[\max_{x'\in X}\mu^n(x',s')+\tilde{\sigma}^n(x',s';x,s)Z-\max_{x''\in X}\mu^n(x'',s')]
\end{aligned}$$

We next prove Lemma 4 from the main paper: if there are finite solutions X and all have been evaluated on a common seed, then the value of sampling any solution on any seed is zero.

LEMMA 4 *Let $X=\{x_1,...,x_d\}$ and $\tilde{X}^n=\{(x_1,1),...,(x_d,1)\}$ then for all $(x,s)\in X\times\mathbb{N}^+$*

$$KG_n^{CRN}(x,s)=0$$

and the maximizer $\arg\max_x \bar{\theta}(x)$ *is known.*

Proof Lemma EC.9 shows that any seed can be used as the target of optimization. Therefore we may choose $s=1$ as the target. All x have been sampled for $s=1$ therefore $\tilde{\sigma}^n(x,1;x',s')=0$ for all $x\in X$ and $s'\in\mathbb{N}^+$. Hence

$$\begin{aligned}\mathrm{KG}^{\mathrm{CRN}}(x,s) &= \mathbb{E}[\max_{x'\in X}\mu^n(x',1)+0Z-\max_{x''\in X}\mu^n(x'',1)]\\ &= 0\end{aligned}$$

for all $x,s\in X\times\mathbb{N}^+$. By Lemma EC.7 the maximizer $\mathrm{argmax}_{x\in X}\bar{\theta}(x)$ is known (although it's underlying value, $\max\bar{\theta}(x)$, is not known). $\square$

We next prove Lemma 5 from the main paper that $\mathrm{KG}^{\mathrm{CRN}}$, when evaluated as in Scott et al. (2011), never samples a new seed and reduces to Expected Improvement (EI) of Jones et al. (1998).

LEMMA 5 *Let $X\subset\mathbb{R}^d$ be a set of possible solutions, $\tilde{X}^n=\{(x^1,1),...,(x^n,1)\}$ be the set of sampled input pairs and $X^n=(x^1,...,x^n)$. Define*

$$KG_n^{CRN}(x,s;A)=\mathbb{E}\left[\max_{x'\in A\cup\{x\}}\mu^{n+1}(x',0)-\max_{x'\in A\cup\{x\}}\mu^n(x',0)\middle|D^n,(x,s)^{n+1}=(x,s)\right].$$

Then for all $x\in X$

$$KG_n^{CRN}(x,1;X^n)>KG_n^{CRN}(x,2;X^n)$$

and therefore $\max_x KG_n^{CRN}(x,1;X^n)>\max_x KG_n^{CRN}(x,2;X^n)$ and seed $s=2$ will never be evaluated. Further

$$KG_n^{CRN}(x,1;X^n)=\mathbb{E}\big[\max\{0,y^{n+1}-\max Y^n\}\big|D^n,x^{n+1}=x,s^{n+1}=1\big].$$

Proof By Lemma EC.9, we may set $s=1$ as the target of optimization. For all sampled points $i=1,...,n$, we have that $\tilde{\sigma}^n(x^i,1;x,s)=0$ and $\mu^n(x^i,1)=y^i$ therefore $\max\mu^n(\tilde{X}^n)=\max Y^n$. Define $\bar{Y}^n=\max Y^n$. The expression for Knowledge Gradient becomes

$$\begin{aligned}\mathrm{KG}_n^{\mathrm{CRN}}(x,s;X^n) &= \mathbb{E}\big[\max\{\bar{Y}^n,\mu^n(x,1)+\tilde{\sigma}^n(x,1;x,s)Z\}\big]-\max\{\bar{Y}^n,\mu^n(x,1)\}\\ &= \mathbb{E}\big[\max\{0,\mu^n(x,1)+\tilde{\sigma}^n(x,1;x,s)Z-\bar{Y}^n\}\big]\\ &= \Delta(x)\Phi\left(\frac{\Delta(x)}{|\tilde{\sigma}^n(x,1;x,s)|}\right)-|\tilde{\sigma}^n(x,1;x,s)|\phi\left(\frac{\Delta(x)}{|\tilde{\sigma}^n(x,1;x,s)|}\right)\\ &= f\big(\Delta(x),\ |\tilde{\sigma}^n(x,1;x,s)|\big)\end{aligned}$$

where $\Phi(\cdot), \phi(\cdot)$ are cumulative and density functions of the Gaussian distribution, $\Delta(x) = \mu^n(x,1) - \bar{Y}^n$ and $f(a,b)$ is the well known expected improvement acquisition function derived from the expectation of a truncated Gaussian random variable. Note that the function $f(a,b)$ is monotonically increasing in b, $\frac{d}{db}f(a,b) = \phi(-a/b) > 0$. Hence, to prove the lemma, it is sufficient to show $|\tilde{\sigma}^n(x,1;x,1)| > |\tilde{\sigma}^n(x,1;x,2)|$ for all $x \in X$. Firstly we may simplify $\tilde{\sigma}^n(x,1;x,1)$ as follows

$$\tilde{\sigma}^n(x,1;x,1) = k^n(x,1,x,1)/\sqrt{k^n(x,1,x,1)} \tag{EC.21}$$

$$= \sqrt{k^n(x,1,x,1)}. \tag{EC.22}$$

Substituting this into the inequality yields

$$\begin{aligned}
|\tilde{\sigma}^n(x,1;x,1)| &> |\tilde{\sigma}^n(x,1;x,2)| \\
\sqrt{k^n(x,1,x,1)} &> \frac{|k^n(x,1,x,2)|}{\sqrt{k^n(x,2,x,2)}} \\
1 &> \frac{|k^n(x,1,x,2)|}{\sqrt{k^n(x,2,x,2)k^n(x,1,x,1)}} \\
-1 &< \operatorname{corr}(\theta(x,1),\theta(x,2)|D^n) \leq 1
\end{aligned}$$

where the last line is true by the positive semi-definiteness of the kernel, the correlation between two random variables cannot be greater than one. The above result demonstrates that allocating samples according to $\mathrm{KG}^{\mathrm{CRN}}$ will always sample seed $s = 1$. The target is stochastic however the objective is deterministic and the new output $y^{n+1} \sim N(\mu^n(x,1), k^n(x,1,x,1))$. The acquisition function simplifies to

$$\begin{aligned}
\mathrm{KG}_n^{\mathrm{CRN}}(x,1;X^n) &= \mathbb{E}\big[\max\{0, \mu^n(x,1) + \sqrt{k^n(x,1,x,1)}Z - \bar{Y}^n\}\big] \\
&= \mathbb{E}\big[\max\{0, y^{n+1} - \bar{Y}^n\}\big|D^n, x^{n+1} = x, s^{n+1} = 1\big]
\end{aligned}$$

where the last line is exactly the EI acquisition criterion of Jones et al. (1998). □

EC.1.4. Suboptimality of $\mathrm{KG}^{\mathrm{PW}}$

Finally we prove that the Value of Information achieved by $\mathrm{KG}^{\mathrm{PW}}$ is less than that of $\mathrm{KG}^{\mathrm{CRN}}$, given the same set of observations and Gaussian process models.

LEMMA 6 *Let D^n be a dataset of observation triplets. For a Gaussian process with a kernel of the form $k_{\bar{\theta}}(x,x') + \delta_{ss'}k_\epsilon(x,x')$, the expected increase in value after two steps allocated according to KG^{CRN} is at least as big as two steps allocated according to KG^{PW},*

$$\begin{aligned}
&\mathbb{E}_n\left[\max_{x'}\mu^{n+2}(x',0) - \max_{x''}\mu^n(x'',0)\middle|(x,s)^{n+1},(x,s)^{n+2} \sim KG^{CRN}\right] \\
\geq\ &\mathbb{E}_n\left[\max_{x'}\mu^{n+2}(x',0) - \max_{x''}\mu^n(x'',0)\middle|(x,s)^{n+1},(x,s)^{n+2} \sim KG^{PW}\right]
\end{aligned}$$

Proof The suboptimality of one or two steps of the serial mode of $\mathrm{KG}^{\mathrm{PW}}$ is clear by noting it is constrained to a new seed, a subset of the same acquisition space considered by $\mathrm{KG}^{\mathrm{CRN}}$ as mentioned in Equation (23). We focus on the suboptimality of one step of the batch mode

$$\begin{aligned}
&\mathbb{E}_n\left[\max_{x'}\mu^{n+2}(x',0) - \max_{x''}\mu^n(x'',0)\middle|(x,s)^{n+1},(x,s)^{n+2} \sim \mathrm{KG}^{\mathrm{CRN}}\right] \\
=\ &\max_{(x,s)^{n+1}}\mathbb{E}_n\left[\max_{(x,s)^{n+2}}\mathbb{E}_{n+1}\left[\max_{x'}\mu^{n+2}(x',0)\middle|(x,s)^{n+2}\right] - \max_{x''}\mu^n(x'',0)\middle|(x,s)^{n+1}\right] \\
\geq\ &\max_{x^{n+1}}\mathbb{E}_n\left[\max_{x^{n+2}}\mathbb{E}_{n+1}\left[\max_{x'}\mu^{n+2}(x',0)\middle|x^{n+2}\right] - \max_{x''}\mu^n(x'',0)\middle|x^{n+1}, s^{n+1}=s^{n+2}=n+1\right] \quad \text{(EC.23)} \\
\geq\ &\max_{x^{n+1},x^{n+2}}\mathbb{E}_n\left[\max_{x'}\mu^{n+2}(x',0) - \max_{x''}\mu^n(x'',0)\middle|x^{n+1},x^{n+2},s^{n+1}=s^{n+2}=n+1\right] \quad \text{(EC.24)} \\
\geq\ &\mathbb{E}_n\left[\max_{x'}\mu^{n+2}(x',0) - \max_{x''}\mu^n(x'',0)\middle|(x^{n+1},x^{n+2}) = \arg\max \mathrm{KG}_n^{\mathrm{PW}}(x,x'), s^{n+1},s^{n+2}=n+1\right] \\
=\ &\mathbb{E}_n\left[\max_{x'}\mu^{n+2}(x',0) - \max_{x''}\mu^n(x'',0)\middle|(x,s)^{n+1},(x,s)^{n+2} \sim \mathrm{KG}^{\mathrm{PW}}\right]
\end{aligned}$$

where the first inequality is due to constraining the acquisition space to a new seed, the second is by Jensen's inequality and the convexity of the max operator implying sub-optimality due to batch pre-allocation, and the third inequality is due to the approximation with differences used in $\mathrm{KG}^{\mathrm{PW}}$ as pairs are not allocated to maximize the true batch value. □

Sequentially allocating two singles to the same new seed is guaranteed to have higher value per sample than a corresponding batch mode pre-allocating a pair to a single seed as shown by Equations (EC.23) and (EC.24). However the serial and batch mode of $\mathrm{KG}^{\mathrm{PW}}$ compute the value over different subsets of the full acquisition space and therefore the batch mode can return higher value per sample.

EC.2. Further Experimental Results

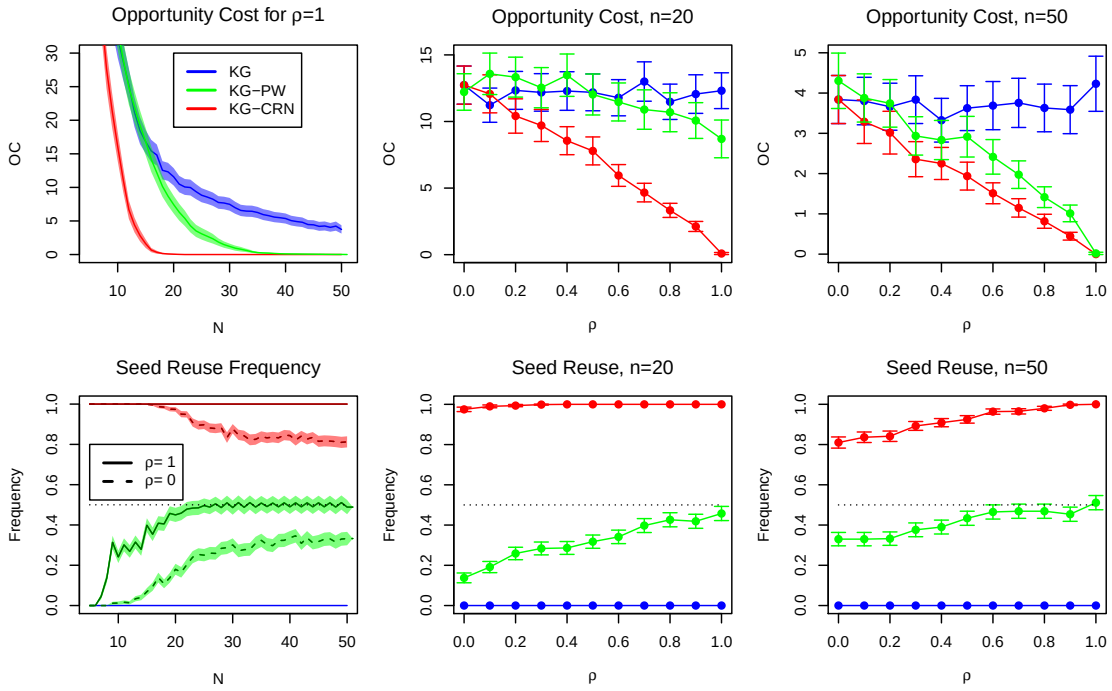

Figure EC.1 Results for synthetic data where the objective function was drawn from the algorithm's GP model with offsets and white noise ($\eta, \sigma_w \geq 0, \sigma_b = 0$) only. We let $\rho = \eta^2/(\eta^2+\sigma_w^2)$ and hold $\eta^2+\sigma_w^2 = 50^2$ constant. For low ρ, all algorithms perform similarly. As ρ increases, KG^{CRN} samples more old seeds and outperforms other methods. We observe that KG^{PW} samples singletons initially, hence its performance is comparable to KG in that phase. Later the performance of KG^{PW} improves over KG and when it samples correlated doubles later improving upon KG.

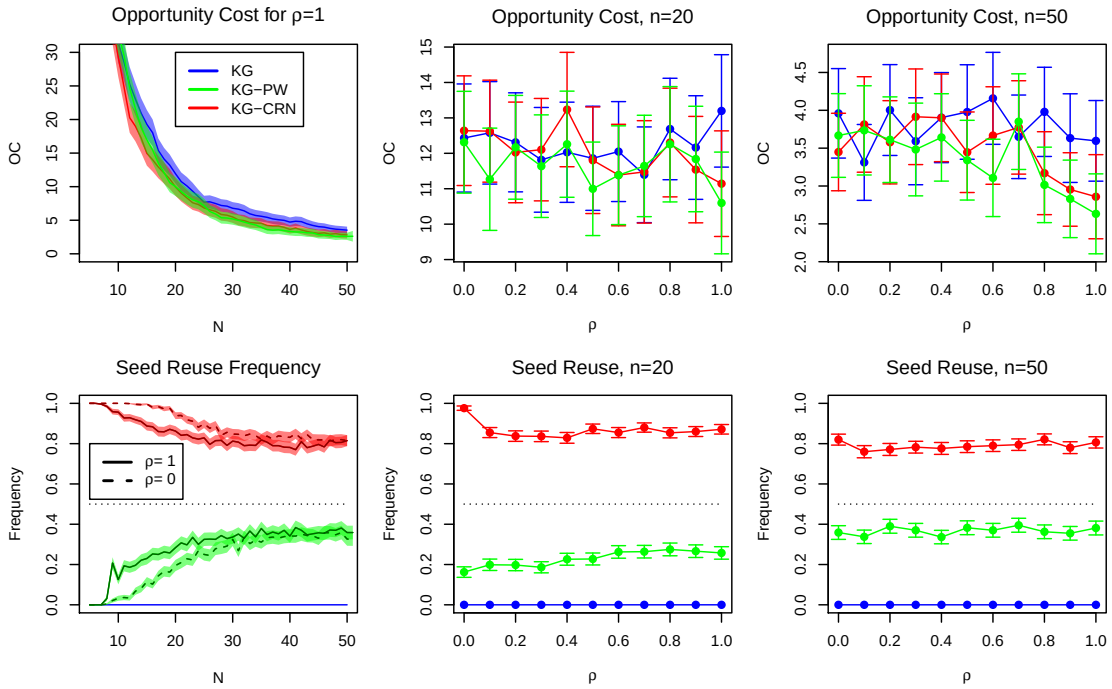

Figure EC.2 More results on GP synthetic data that was generated with $\eta^2 = 0$ and $\rho = \sigma_b^2/(\sigma_b^2 + \sigma_w^2)$, holding $\sigma_b^2 + \sigma_w^2 = 50^2$ constant. We do not observe a significant benefit from bias functions alone in the case of no offsets.

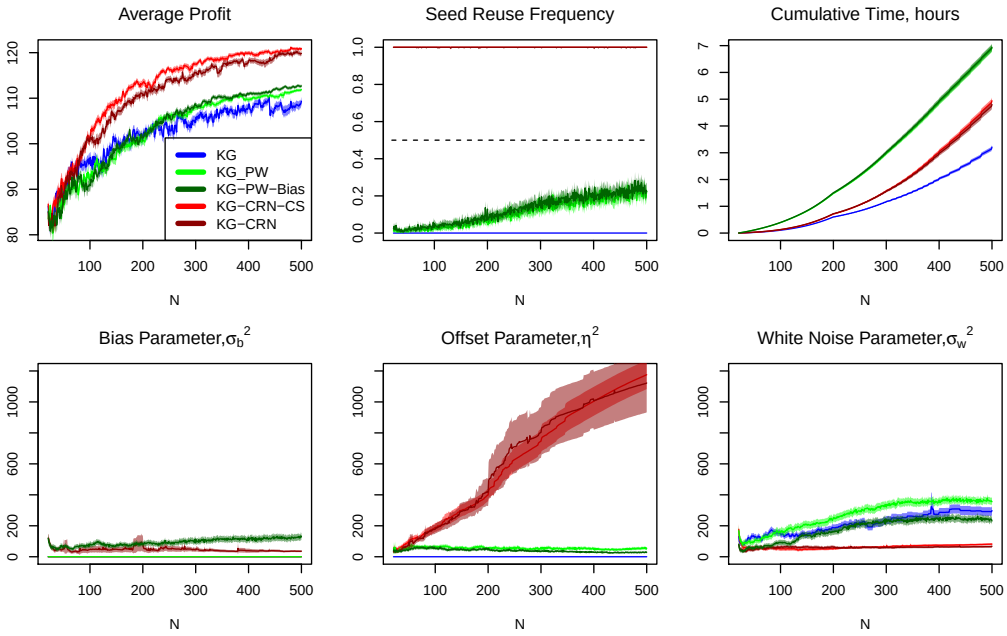

Figure EC.3 ATO results. The KG^{PW} algorithm samples singletons early on, and never learns a large offset parameter η^2. KG^{CRN} samples old seeds and eventually learns a large offset parameter and never samples any new seeds. KG has the smallest running time, followed by KG^{CRN} variants and the KG^{PW} variants.

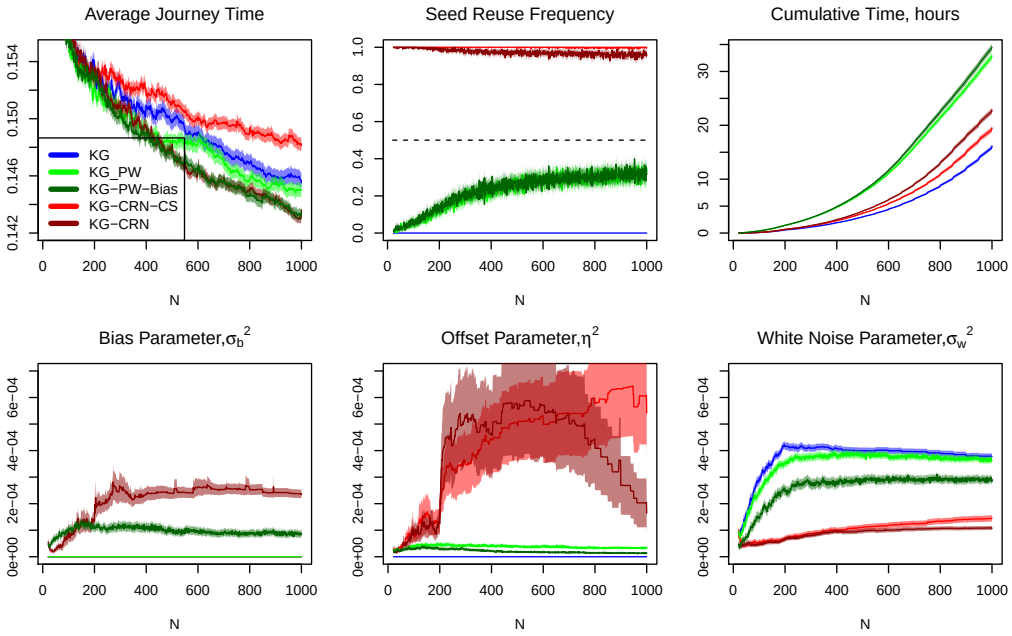

Figure EC.4 Ambulances in a square problem (AIS). The bias functions provide significant benefit to both KG^{CRN} and KG^{PW}. Excluding bias functions, $\text{KG}^{\text{CRN}} - CS$, leads to inefficient sampling of old seeds only. The KG^{CRN} variants learn larger offset parameters and require less computation time.

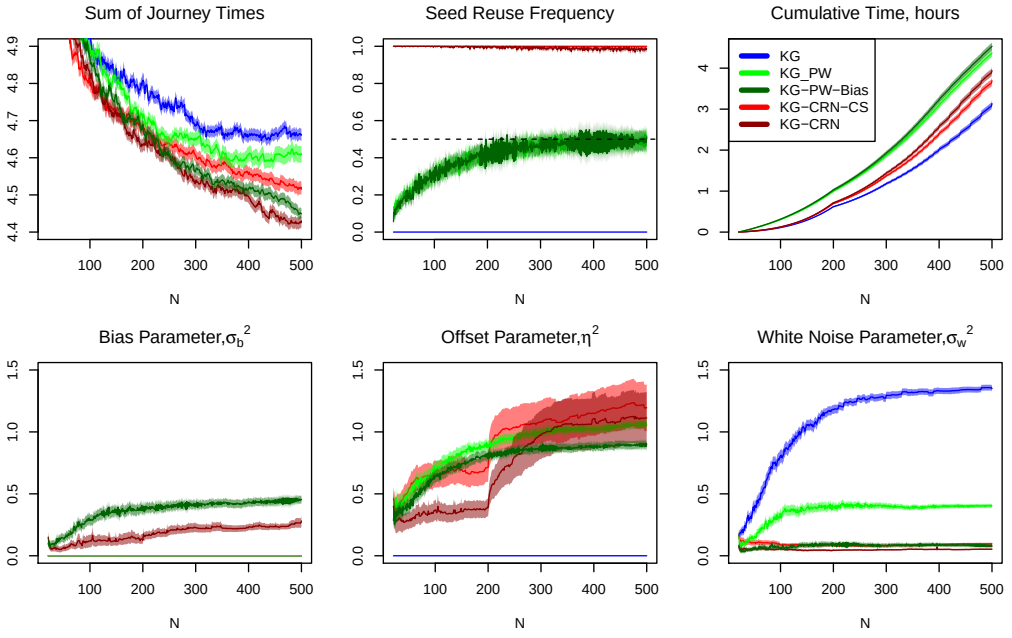

Figure EC.5 The AIS problem with the sum of journey times in a simulation as the objective. KG^{CRN} variants improve performance over KG, and bias functions improve performance over compound spheric variants. All methods reuse seeds to the extent that this is possible for them.

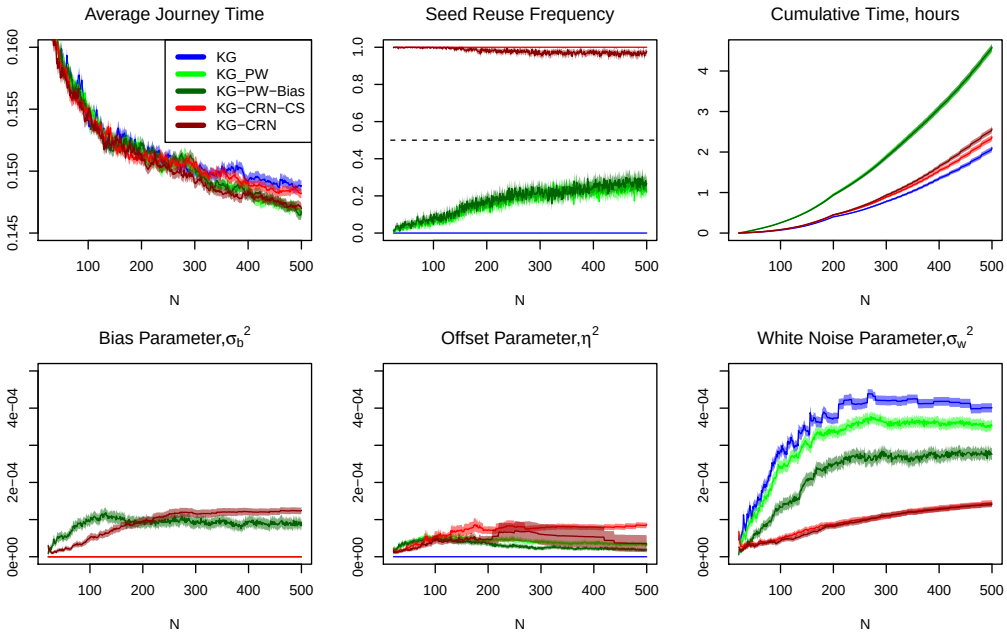

Figure EC.6 All algorithm variants perform similarly. The offset and bias parameters are considerably lower than the white noise parameter, suggesting there is little exploitable structure in the noise for this problem.

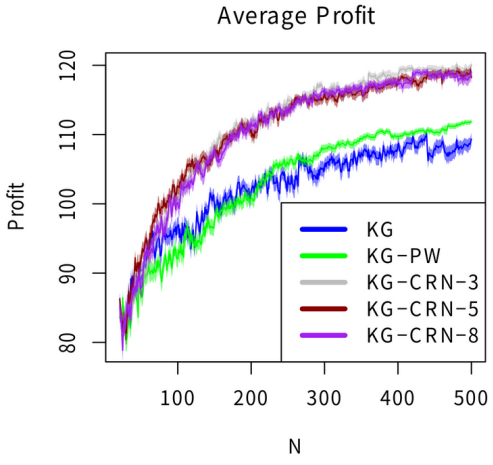

Figure EC.7 The ATO benchmark where the KG^{CRN} algorithm is initialized with 20 points spread over 3, 5, or 8 seeds. In all cases, the KG^{CRN} algorithm converges to exactly the same level of profit.

EC.3. Response Surfaces for ATO and Ambulances Problem

To understand the impact of the random seed on some real-world problems, Figure EC.8 depicts (empirically estimated) $\bar{\theta}(x)$ and $\theta(x,s)$ for each problem along a linear path within the higher dimensional solution space. For the ATO problem, the objective for each seed and the target have almost identical shape differing mostly by additive offsets, thus seed peaks align almost exactly with the target peak and CRN will be beneficial. For the Ambulance problem, each seed has an offset as well as more unique variation, the peaks of seeds are approximately similar to the target and CRN may be beneficial but less so than ATO.

To demonstrate the better fit of our proposed GP model, we also examined its predictive accuracy in a cross-validation analysis. For each of ATO, AIS, AIS-sum, AIS-1800s, using the sampled data from the KG^{CRN} optimization runs, X^n, S^n, Y^n, we fit both a standard GP to predict Y^n given X^n and our proposed CRN-GP to predict Y^n given X^n, S^n. Figure EC.9 depicts the average leave-one-out cross validation error for the 400 datasets, one from each experiment replication. We see that modeling the ATO and AIS-sum benefit greatly from the CRN-GP while AIS and AIS-1800s only show a marginal difference.

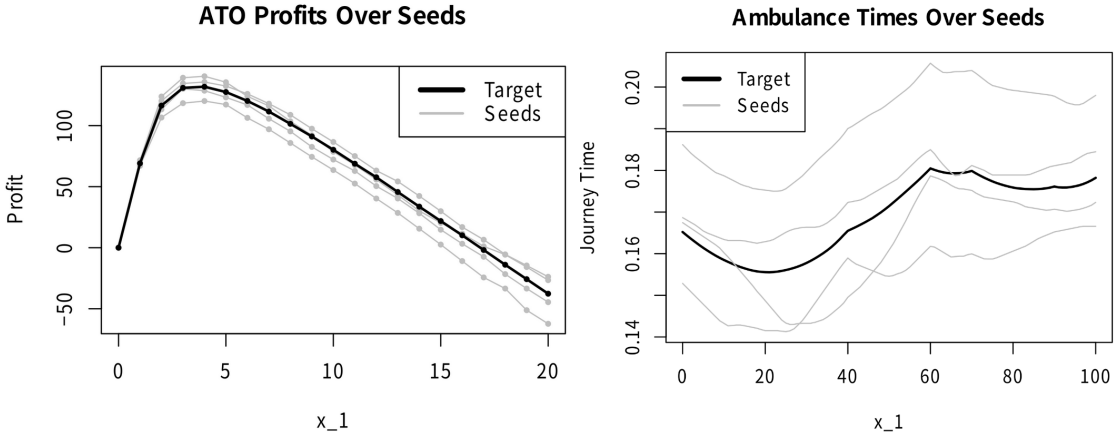

Figure EC.8 Left: ATO problem $x = (l, ..., l)$ for $l \in \{0, ..., 20\}$ and for 6 seeds of $\theta(x,s)$ (grey) and $\theta(x)$ (black). Each seed has shape and peak location almost identical to the target. Right: AIS where $x = (l, x_{2:6})$ with random fixed $x_{2:6}$. Each seed $\theta(x,s)$ has an offset and more seed specific variation from $\bar{\theta}(x)$.

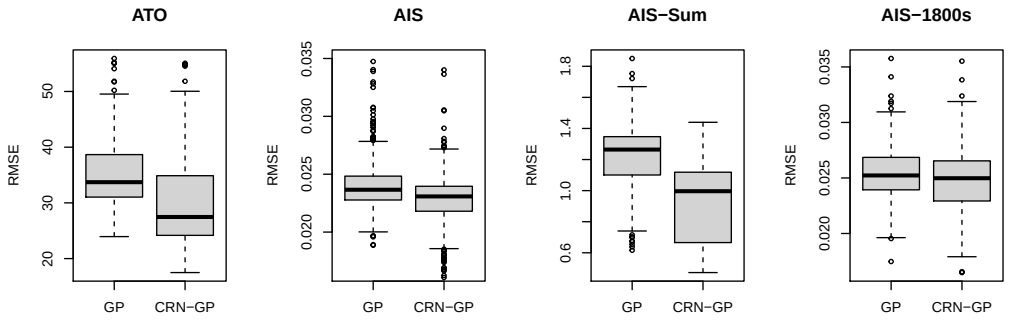

Figure EC.9 For ATO, and AIS-sum, the CRN-GP model performs significantly better and these problems also showed the most benefit from using CRN. For AIS and AIS-1800s, we observe no significant benefit from the CRN model.

EC.4. Algorithm Implementation Details

EC.4.1. Hyperparameter Learning

The hyperparameters of the GP prior are estimated by multi-start conjugate gradient ascent of the marginal likelihood (Rasmussen 2003). All parameters are lower bounded by zero. We set upper bounds for the length scale parameters to double the largest separation between points in each dimension. For all other parameters we set the upper bound to $1.5(\max Y^n - \min Y^n)^2$. We perform optimization in two ways, a full optimization and a finetuning optimization. For the full optimization, we evaluate the marginal likelihood at 1000 points that are randomly uniformly distributed. The best 20 points are used as starting points for 100 steps of conjugate gradient ascent each. This expensive search is used for each of the first 200 iterations of the algorithm, then at increasing intervals thereafter to save computation time. For all other iterations, we only fine tune by conducting 20 steps of gradient ascent using the current best hyperparameters as a starting point.

Recall that the likelihood has the following form:

$$\mathbb{P}[Y^n|\tilde{X}^n, L, \sigma_{\bar{\theta}}^2, \eta^2, \sigma_b^2, \sigma_w^2] = -\frac{1}{2}\left((Y^n - \bar{Y})^\intercal K^{-1}(Y^n - \bar{Y}) + \log(|K|) + n\log(2\pi)\right)$$

$$K_{ij} = \sigma_{\bar{\theta}}^2 \exp\left(-\frac{1}{2}(x^i - x^j)^\intercal L(x^i - x^j)\right)$$

$$+\mathbb{1}_{s^i=s^j}\left(\eta^2+\sigma_b^2\exp\left(-\frac{1}{2}(x^i-x^j)^\intercal L(x^i-x^j)\right)+\mathbb{1}_{x^i=x^j}\sigma_w^2\right).$$

Firstly, an independent noise model (IND) is fitted by clamping $\eta^2=\sigma_b^2=0$ to yield

$$L^{IND},{\sigma_{\tilde{\theta}}^2}^{IND},{\sigma_w^2}^{IND}=\text{argmax}\mathbb{P}[Y^n|\tilde{X}^n,L,\sigma_{\tilde{\theta}}^2,\eta^2=\sigma_b^2=0,\sigma_w^2]. \tag{EC.25}$$

Secondly, the noise parameters $\eta^2,\sigma_b^2,\sigma_w^2$ are optimized whilst keeping the total noise fixed $\eta^2+\sigma_b^2+\sigma_w^2={\sigma_w^2}^{IND}$ which is a two-dimensional optimization, we reparameterize as follows

$$\begin{aligned}
\eta^2(\alpha,\beta) &= \beta(1-\alpha){\sigma_w^2}^{IND}\\
\sigma_b^2(\alpha,\beta) &= (1-\beta)(1-\alpha){\sigma_w^2}^{IND}\\
\sigma_w^2(\beta) &= \alpha{\sigma_w^2}^{IND}\\
\alpha,\beta &= \text{argmax}_{[0,1]^2}\mathbb{P}[Y^n|\tilde{X}^n,L^{IND},{\sigma_{\tilde{\theta}}^2}^{IND},\eta^2(\alpha,\beta),\sigma_b^2(\alpha,\beta),\sigma_w^2(\beta)]
\end{aligned}$$

Thirdly, the final estimates of all hyperparameters are simultaneously fine-tuned by gradient ascent. This three-stage method guarantees that the found likelihood is greater than the equivalent non-CRN parameter estimates. Note that the second extra step of optimization is performed only over the unit square and is thus cheaper than learning all hyperparameters from scratch.

EC.4.2. Optimization of $\mathsf{KG}_n^{\mathsf{CRN}}(x,s)$

Derivatives of KG^{CRN} and KG^{PW}, when evaluated by discretization over X as we do, are easily (but tediously) derived and can be found in multiple previous works (Scott et al. 2011, Xie et al. 2016). Alternatively, any automatic differentiation package, (Autograd, TensorFlow, PyTorch) may be used as the mathematical operations are all common functions. We propose the following optimization procedure:

1. Evaluate $\text{KG}^{\text{CRN}}(x,s)$ across an initial Latin Hypercube design with 1000 points over the acquisition space $\tilde{X}_{acq}=X\times\{1,...,\max S^n+1\}$.

2. Use the top 20 initial points to initialize 100 steps of conjugate gradient ascent over X, holding the seed constant within each run.

3. For the largest (x, s) pair found, evaluate $\mathrm{KG}^{\mathrm{CRN}}(x, s)$ for the same x on all seeds $s \in \{1, ..., \max S^n + 1\}$

4. Perform 20 steps of gradient ascent to fine tune the x from the best seed.

When not using common random numbers, stages one and two use the same new seed and stages three and four are omitted.

Acknowledgments

The first author gratefully acknowledges funding through the UK Engineering and Physical Sciences Research Council (Grant no. EP/101358X/1).

Michael Pearce studied for his PhD in Mathematics at the University of Warwick, took internships at Google Deepmind and Uber AI Labs and finished his PhD in 2019. Michael is currently Senior Researcher at Zenith AI, a technology company based in Northern Ireland. His research interests include expensive black box optimization under uncertainty, particularly Bayesian optimization and Bayesian modelling in general such as deep variational inference and graphical models.

Matthias Poloczek is a Principal Scientist at Amazon and works at the intersection of machine learning (ML) and optimization, in particular on Bayesian optimization and bandits for industrial-scale applications. Previously, Matthias was a Senior Manager at Uber AI where he founded Uber?s Bayesian optimization team and led a cross-org effort to build a company-wide AutoML service to tune ML models at scale. Matthias received his PhD in CS from Goethe University in Frankfurt in 2013 and then worked as a postdoc at Cornell with David Williamson and Peter Frazier from 2014 until 2017. He was an Assistant Professor in the Department of Systems and Industrial Engineering at the University of Arizona from 2017 until 2019.

Juergen Branke Juergen Branke is Professor of Operational Research and Systems at Warwick Business School, University of Warwick (UK). His main research interests include metaheuristics and Bayesian optimisation applied to problems under uncertainty, such as simulation optimisation, dynamically changing problems, and multi-objective problems. Prof. Branke is Editor of ACM Transactions on Evolutionary Learning and Optimization, Area Editor of the Journal of Heuristics and the Journal on Multi-Criteria Decision Analysis, as well as Associate Editor of IEEE Transactions on Evolutionary Computation and the Evolutionary Computation Journal.