Proceedings of the 2019 Winter Simulation Conference
N. Mustafee, K.-H.G. Bae, S. Lazarova-Molnar, M. Rabe, C. Szabo, P. Haas, and Y.-J. Son, eds.

BAYESIAN SIMULATION OPTIMIZATION WITH COMMON RANDOM NUMBERS

Michael Pearce

Complexity Science
University of Warwick
Coventry, UK

Matthias Poloczek

Uber AI
1455, Market Street
San Francisco, USA

Juergen Branke

Warwick Business School
University of Warwick
Coventry, UK

ABSTRACT

We consider the problem of stochastic simulation optimization with common random numbers over a numerical search domain. We propose the Knowledge Gradient for Common Random Numbers (KG-CRN) sequential sampling algorithm, a simple elegant modification to the Knowledge Gradient that incorporates the use of correlated noise in simulation outputs with Gaussian Process meta-models. We compare this method against the standard Knowledge Gradient and a more recently proposed variation that allows for pairwise sampling. Our method significantly outperforms both baselines under identical laboratory conditions while greatly reducing computational cost compared to pairwise sampling.

1 INTRODUCTION

We consider the problem of finding the best input x from a finite set of real valued inputs X, where the quality of an input can only be observed by stochastic simulation. The simulation model, $\theta(x,s)$, is a function of both input x and random number stream s, and the best is defined as having highest expected output $\bar{\theta}(x)$ averaged over streams s,

$$\max_{x \in X} \mathbb{E}[\theta(x,s)] = \max_{x \in X} \bar{\theta}(x).$$

The argument s controls all the stochasticity in the simulation $\theta(x,s)$, a random number stream for example. We allow for reuse of the random number streams s such that two or more inputs may be evaluated with common random numbers introducing positive correlation in the observation noise

$$\mathbb{E}[(\theta(x,s) - \bar{\theta}(x))(\theta(x',s) - \bar{\theta}(x'))] > 0,$$

where expectation is over s. We aim to find the optimal x in as few simulation runs as possible. This problem has many applications, particularly in simulators where random number streams are used for the same purpose for any x. For example, queueing systems where two or more systems may be compared using the same sequence of random arrival times.

We consider a Bayesian approach using Gaussian Process Regression, or Kriging, to find the optimal x exploiting both similarity in expected outputs over x, $\bar{\theta}(x)$, and exploiting correlation in simulation

output noise when streams s are reused. We generalize the popular Knowledge Gradient algorithm by allowing the reuse of random number streams giving the optimization algorithm the ability to compare *two or more* different alternatives using the same random number stream. We provide a simple derivation of this algorithm and empirically compare it on synthetic problems to standard Knowledge Gradient without common random numbers (Frazier et al. 2009) and a more recently published variant of Knowledge Gradient that allows for pairwise sampling with common random numbers (Xie et al. 2016). Our proposed method performs comparably when there is little noise correlation, and significantly outperforms both baselines when there is high correlation and therefore more exploitable benefit from using common random numbers.

In Section 2 we give a brief review of previous approaches considering similar problems, in Section 3 we introduce notation and formalise the problem we consider and in Section 4 we describe our proposed method, the Knowledge Gradient with Common Random Numbers. We compare this method against standard Knowledge Gradient and Knowledge Gradient with Pairwise Sampling in Section 5 and finally conclude in Section 6.

2 BACKGROUND

Simulation optimization is typically treated as an expensive stochastic black box optimization problem. We focus on two classes of this problem, model based methods where the search domain of the optimization is typically numerical, i.e. finite discrete or continuous, and model free methods where the search domain is categorical.

When the input to a simulator can be mapped to a numerical domain, such as a target stock levels or features of a drug molecule, evaluating the simulator for a range of inputs X produces outputs Y that form a data set to build a surrogate model (Sacks et al. 1989). In a Bayesian setting the X values may be used to build a correlated prior over outputs which may then be conditioned on observations Y. Gaussian Processes or Krigging models (Rasmussen 2004) provide predictions as well as uncertainty estimates of the output at a new input and are therefore ideal for sequential information collection. The Efficient Global Optimisation (EGO) algorithm (Jones et al. 1998) combined a Gaussian Process model with the Expected Improvement (Mockus et al. 1978) of a new output over the current best output to sequentially query and optimize an unknown black box function. Other acquisition functions, or infill criteria, include Upper Confidence Bound (Srinivas et al. 2009), Probability of Improvement (Kushner 1964), Entropy Search (Hennig and Schuler 2012) and Knowledge Gradient (Frazier et al. 2009). Note, if the simulator input itself is not numerical, instead numerical properties or characteristics of the inputs may be used, for example using graph features to predict coloring algorithm performance (Smith-Miles et al. 2014), or molecule properties to predict vaccine binding rates (Krause and Ong 2011).

Contrasting with numerical inputs (or features of inputs) are black box functions with inputs X for which surrogate models cannot easily be used. We here refer to these as model-free methods, or optimization over an uninformative finite categorical domain. For example, the input to a job shop simulator is one of a given set of scheduling heuristics. Efficient ranking and selection of alternative solutions has been widely studied with frequentist methods (Kim and Nelson 2006; Branke et al. 2007) and in the Bayesian setting with *independent* priors over outputs that either aim to maximize the probability of correctly selecting the best alternative (Gupta and Miescke 1996; Chick and Inoue 2001), or maximize the expected output of the selected system (Frazier et al. 2008; Chen and Lee 2010). Racing algorithms evaluate all alternatives sequentially eliminating lesser alternatives until only one is left (Birattari et al. 2002; Birattari et al. 2010).

In either model-based or model-free settings, the use of common random numbers (CRN) in multiple calls to a simulator can induce positive correlation in the output noise for different inputs thereby reducing variance in the difference between outputs. Combining CRN with ranking and selection has been considered with 2-stage algorithms (Chick and Inoue 2001), the popular Optimal Computing Budget Allocation (Fu et al. 2004), and probability of correct selection with an indifference zone (Nelson and Matejcik 1995; Görder and Kolonko 2019). In the Gaussian process regression model based setting, the effects of correlated observations due to common random numbers can degrade inference (Chen et al. 2012) and augmenting

the Knowledge Gradient with Pairwise Sampling (Xie et al. 2016) to account for correlation in output noise was shown to significantly speed up optimization.

In this work we build upon the standard sequential Knowledge Gradient method that collects data to maximize the expected peak of a surrogate model thereby learning the true function optimizer. When the surrogate model is a Gaussian Process and there is a finite number of inputs X this may be computed exactly. We generalize both the regression model for inference as well as the one-step value of information procedure to allow use of common random numbers, all in closed form. This has been considered before in a special case which assumes that random number streams from the history of past evaluations cannot be reused. Instead, one may determine a pair of inputs and a new common random number stream and observe a pair of outputs that have CRN (Xie et al. 2016).

3 PROBLEM DEFINITION

We assume that we have an expensive to run simulator, $\theta : X \times \mathbb{S} \to \mathbb{R}$, that takes as input a decision variable from a finite set $x \in X$, such as integer vectors in a hyper-rectangle, and a random number stream $s \in \mathbb{S}$. For a given stream, the simulator is a deterministic function of x. The aim of the user is to optimize the expectation of the simulator over random number streams

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

We assume we have a limited budget of N simulation runs and for each run one must choose a stream s and a decision variable x then observe $\theta(x,s)$. If we add the constraint that every call to the simulator uses a unique stream, the problem reduces to a standard stochastic simulation optimization and the user only needs to determine x values for each evaluation of $\theta(x,s)$. This framework is therefore a more general setting that allows the reuse of random number streams and therefore we make the argument s explicit.

The benefit from the use of common random numbers is likely to depend on the particular problem. For example, those in which the stream models simulated environment randomness unaffected by the decision variables x are likely to benefit. Examples include

- a periodic time controlled traffic light at a 4-way road intersection. Vehicles arrive at random times given by a stream s and the timing of red and green phases for the different traffic streams are decision variables x. A designer aims to minimize delays to driver journeys in a simulated day, $\theta(x,s)$, averaged over all possible realisations of vehicle arrivals s.
- a shop where inventory storage is expensive and restocking is delayed. Customers arrive randomly according to stream s, a shop manager aims to find the optimal target inventory level x that maximizes sales of available inventory minus storage cost in a simulated week, $\theta(x,s)$, averaged over all possible customer streams s.

4 METHOD

4.1 The Probabilistic Model

We propose to use Gaussian Process Regression as a surrogate model for the simulator $\theta(x,s)$. Without loss of generality, we may define a random number stream by the positive integer seed used in the random number generator within the simulator and therefore $s \in \mathbb{N}^+ = \{1,2,3,\cdots\}$. To define a Gaussian Process we require a prior mean function $\mu^0 : X \times \mathbb{N}^+ \to \mathbb{R}$ which is typically set to 0 and a positive semi-definite kernel $k^0 : X \times \mathbb{N}^+ \times X \times \mathbb{N}^+ \to \mathbb{R}$ that defines abstract properties of the surrogate model such as smoothness or periodicity. This is chosen by the user to incorporate prior knowledge of the true function $\theta(x,s)$, we discuss further details below. Intuitively, a Gaussian Process model assumes that a finite set of n observations of the function $\theta(x,s)$ for n different x^i, s^i inputs are a single sample from an n dimensional multivariate normal distribution whose mean vector and covariance matrix are given by evaluating the prior mean and kernel functions at the inputs. If we augment this set of n inputs with the points $\theta(x^j, s^j)$ and $\theta(x^k, s^k)$,

and consider the $n+2$ dimensional multivariate normal where the first n dimensions are fixed to their observed values, this yields a bivariate posterior distribution for the function at unobserved inputs $\theta(x^j,s^j)$ and $\theta(x^k,s^k)$. Mathematically, we define the sequence of input pairs $\tilde{X}^n = \{(x^i,s^i)\}_{i=1}^n$ and $Y^n \in \mathbb{R}^n$ as the vector of observed simulation outputs $[Y^n]_i = y^i = \theta(x^i,s^i)$. Given a dataset of n input-output triplets $D^n = \{(x^1,s^1,y^1),...,(x^n,s^n,y^n)\}$ the posterior distribution over the function values for new input pairs $\theta(x,s)$ and $\theta(x',s')$, the surrogate model, is given by

$$\mathbb{E}[\theta(x,s)|D^n] = \quad \mu^n(x,s) \quad = \mu^0(x,s) + k^0(x,s,\tilde{X}^n)K^{-1}(Y^n - \mu^0(\tilde{X}^n)) \tag{1}$$

$$\mathrm{Cov}[\theta(x,s),\theta(x',s')|D^n] = \quad k^n(x,s,x',s') \quad = k^0(x,s,x's') - k^0(x,s,\tilde{X}^n)K^{-1}k^0(\tilde{X}^n,x's') \tag{2}$$

where $k^0(x,s,\tilde{X}^n) \in \mathbb{R}^{1\times n}$ is the matrix of the kernel evaluated at the input (x,s) and the set of n observed inputs. Similarly for $k^0(\tilde{X}^n,x's')$, $K^{-1} = \left(k^0(\tilde{X}^n,\tilde{X}^n)\right)^{-1} \in \mathbb{R}^{n\times n}$ is the inverse of the prior covariance matrix that was assumed to have generated the observed output vector Y^n with prior mean $\mu^0(\tilde{X}^n)$. Further information can be found in (Rasmussen 2004).

We assume a prior mean $\mu^0(x,s) = 0$ and we next discuss choice of kernel. In this case, we note that the seed with index s is a categorical variable, the magnitude of the integer value of s is not informative about output therefore must not be used in the kernel. Following previous work (Xie et al. 2016), we use the following kernel

$$k^0(x,s,x',s') = k_{\bar{\theta}}(x,x') + \delta_{ss'}(\eta^2 + \sigma^2\delta_{xx'}).$$

where $k_{\bar{\theta}}(x,x')$ models the underlying latent function $\bar{\theta}(x)$, δ_{ij} is the Kronecker delta function or the white noise kernel, η^2 and σ^2 are two parameters that dictate the noise model. Note that any kernel over $X\times X$ may be used within the parenthesis of the second term, in this case it is a sum of the constant kernel η^2 and white noise kernel $\sigma^2\delta_{xx'}$. In general, the kernel defines a generative model for $\theta(x,s)$, firstly $\bar{\theta}(x)$ is a realisation of a Gaussian Process with kernel $k_{\bar{\theta}}(x,x')$, typically the squared exponential or Matern 5/2 kernel (Rasmussen 2004). Secondly, $\delta_{ss'}\eta^2$ implies the effect of each seed s is to modify the underlying function $\bar{\theta}(x)$ by adding a constant offset $c(s)$ and thirdly, $\delta_{ss'}\delta_{xx'}\sigma^2$ assumes the simulator output for each each input pair x,s is modelled as a further unique offset $g(x,s)$. $c(s)$ and $g(x,s)$ are therefore realisations of white noise processes with variances η^2 and σ^2 respectively. In summary, we are assuming the simulator is a function of the form

$$\theta(x,s) = \bar{\theta}(x) + c(s) + g(x,s),$$

where $c(s) \sim N(0,\eta^2)$ are independent and identically distributed offsets constant for each seed and $g(x,s) \sim N(0,\sigma^2)$ are further independent and identically distributed offsets unique for each input pair (x,s). The total noise in observations is given by $\eta^2+\sigma^2$ and the *correlation coefficient* in noise for a fixed s is given by the ratio

$$\rho = \frac{\eta^2}{\eta^2+\sigma^2}.$$

This sum of white noise process realisations is known as the *compound spheric* noise assumption (Chen et al. 2012). In Figure 1, we provide example realisations from the assumed generative model with high and low noise correlation. This kernel has the advantage that, when compared with a standard noisy Gaussian Process, the differences between observations and ground truth $Y^i - \bar{\theta}(x^i)$ are modelled as structured noise specified by only two parameters η^2 and σ^2. If s is not informative at all about noise correlation then setting (or learning) $\eta^2 = 0$ or $\rho = 0$ recovers a standard noise model with variance σ^2. However the disadvantage is that the numerical value of the inputs X are not used to inform noise correlation, the correlation predictor is effectively "model-free". Likewise it cannot naively be applied to a continuous input domain, X. $\delta_{xx'}$ is not continuous at $x = x'$, such a white noise generative model assumes that infinitesimally close x and x' may still have arbitrarily different $\theta(x,s)$ and $\theta(x',s)$ which may be rather unrealistic in practice and somewhat contrary to the deterministic output assumption.

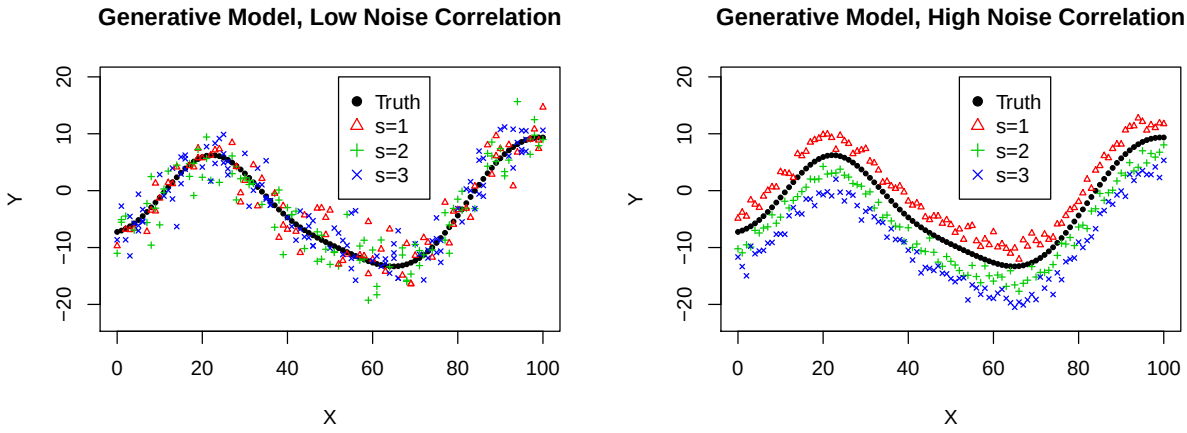

Figure 1: Realization from the assumed generative model with $\rho = 0.1$ (Left) and $\rho = 0.9$ (Right) whilst holding constant $\eta^2 + \sigma^2 = 10^2$. With higher correlation, more of the noise is in the constant offset $c(s)$ and less of the noise is $g(x,s)$.

Note that for seeds that have not been evaluated, it is easily shown that the model makes the same predictions, if $s, s', s'' \notin \{s^1, ..., s^n\}$ then the posterior mean and kernel satisfy

$$\begin{aligned} \mu^n(x,s) = \quad & \mu^n(x,s') \quad && = \mu^n(x,s''), \\ k^n(x,s,x's') = \quad & k^n(x,s',x',s'') \quad && = k^n(x,s,x,s'') \end{aligned}$$

reflecting the prior belief that all seeds are treated the same, i.e. assumptions like "odd seeds are always better" or "seed 20 is always lower" are not accidentally incorporated. We next describe some special cases of the above model. If all evaluations are assumed to be on unique seeds, then each observation will be the result of the ground truth and a unique realisation of the offsets, therefore $y^i \sim \mathrm{N}(\bar{\theta}(x^i), \eta^2 + \sigma^2)$. The covariance matrix of past evaluations is of the form

$$K(\tilde{X}^n, \tilde{X}^n) = k_\theta(X^n, X^n) + I(\eta^2 + \sigma^2)$$

where X^n is the set of observed input x values only and I is the $n \times n$ identity matrix. This is the standard covariance matrix that assumes observation vector Y^n is from an underlying function $\bar{\theta}(X^n)$ with independent and identically distributed noise added on top given by the second diagonal noise matrix.

If all observations are on a single seed $s = 1$, the covariance matrix of the observation vector Y^n is

$$K(\tilde{X}^n, \tilde{X}^n) = k_\theta(X^n, X^n) + \eta^2 \mathbb{1}^{n \times n} + I\sigma^2$$

where $\mathbb{1}^{n \times n}$ is a matrix of ones and the total noise matrix $\eta^2 \mathbb{1}^{n \times n} + I\sigma^2$ now contains off-diagonal elements. A multivariate sample from such a matrix may be viewed as an underlying function $\bar{\theta}(X^n)$ plus a constant $c(1)$ and smaller, σ^2, independent and identically distributed noise given by the final diagonal matrix.

4.2 Sampling Method

We next derive the Knowledge Gradient for Common Random Numbers, $\mathrm{KG}^{\mathrm{CRN}}(x,s)$, that assigns a value to executing the simulator with input pair x, s and observing its output. This value, or acquisition function, can then be cheaply optimized to find the most informative input for the simulator. It is easy to see that by using the kernel specified above, the model provides an estimate for the ground truth $\bar{\theta}(x)$ if we

simply evaluate the posterior mean at an unobserved seed $s \notin \{s^1, .., s^n\}$. Therefore, we may dictate that the unobservable seed $s = 0$ is the ground truth estimate

$$\mathbb{E}[\bar{\theta}(x)|D^n] = \mu^n(x,0).$$

At time n, if we were to stop collecting data, the optimal risk neutral decision would be to recommend the peak of the ground truth estimate $x_r^n = \text{argmax}_x \mu^n(x,0)$ and the expected performance would be

$$\max_x \mu^n(x,0).$$

If we were to choose the next input $(x,s)^{n+1}$ to simulate and observe $y^{n+1} = \theta(x^{n+1}, s^{n+1})$, the performance would be updated $\max_x \mu^{n+1}(x,0)$. At time n, before observing y^{n+1}, the incremental improvement in expected predicted performance is given by

$$\mathbb{E}_{y^{n+1}}[\max_{x'} \mu^{n+1}(x',0)|D^n, (x,s)^{n+1}] - \max_{x'} \mu^n(x,0).$$

The predictive distribution of y^{n+1} given $(x,s)^{n+1}$ and the Gaussian process model with the data so far D^n is given by

$$\mathbb{P}[y^{n+1}|D^n, (x,s)^{n+1}] = \text{N}(y^{n+1}|\mu^n(x^{n+1}, s^{n+1}), k^n(x^{n+1}, s^{n+1}, x^{n+1}, s^{n+1})).$$

The mean function after one time step is easily derived from Equation (1) with a change of indices from 0 and n to n and $n+1$, an instance of Bayesian updating,

$$\mu^{n+1}(x,0) \quad = \quad \mu^n(x,0) + \frac{k^n(x,0,x^{n+1},s^{n+1})(y^{n+1} - \mu^n(x^{n+1},s^{n+1}))}{k^n(x^{n+1},s^{n+1},x^{n+1},s^{n+1})}.$$

The above expression may be factorised into deterministic and stochastic factors,

$$\begin{aligned} &= \mu^n(x,0) + \frac{k^n(x,0,x^{n+1},s^{n+1})}{\sqrt{k^n(x^{n+1},s^{n+1},x^{n+1},s^{n+1})}} \frac{(y^{n+1} - \mu^n(x^{n+1},s^{n+1}))}{\sqrt{k^n(x^{n+1},s^{n+1},x^{n+1},s^{n+1})}} \\ &= \mu^n(x,0) + \tilde{\sigma}^n(x,0;(x,s)^{n+1})Z \end{aligned}$$

where Z is the z-score of y^{n+1} on its predictive distribution which is a standard univariate normally distributed random variable. $\tilde{\sigma}^n(x,0;(x,s)^{n+1})$ is a deterministic additive update to the posterior mean of the ground truth estimate, it is a function of x parameterized by $(x,s)^{n+1}$ and the scale of the added update is given by the random Z. Finally, we define the expected improvement in predicted performance as the Knowledge Gradient for Common Random Numbers,

$$\text{KG}^{\text{CRN}}(x,s) = \mathbb{E}\left[\max_{x' \in X} \mu^n(x',0) + \tilde{\sigma}^n(x',0;x,s)Z\right] - \max_{x'} \mu(x',0) \tag{3}$$

where the expectation is over $Z \sim N(0,1)$ and $\mu^n(x,0)$ and $\tilde{\sigma}^n(x,0;(x,s)^{n+1})$ only depend on the known data collected up to time n. The input to the next simulation $(x,s)^{n+1}$ is determined by optimizing the above acquisition function $(x,s)^{n+1} = \text{argmax}_{x,s} \text{KG}^{\text{CRN}}(x,s)$ where x is optimized over X for each seed in the set of past evaluated seeds and one new seed $s \in \{1, .., \max s^i, \max s^i + 1\}$. This way the sampling procedure is free to "recycle" old seeds/random number streams as well as query new random number streams. As the number of seeds grows, so too does the *acquisition* search space, which always contains one new seed. This allows the algorithm to *dynamically* decide how many seeds to query or whether to stay on old seeds. In Section 5 we show that old seeds are sampled more for small budgets and new seeds are sampled more for larger budgets.

The above derivation is exactly that of the one step Bayes optimal Knowledge Gradient family of algorithms and the evaluation of the required expectation can only be computed analytically if the set X is finite using the Knowledge Gradient for correlated priors (Frazier et al. 2009). If the set X is continuous (or very large finite) there are two alternative approaches. Either the $\max_{x' \in X}$ may be approximated with a maximization over smaller finite subset $\max_{x' \in \bar{X}}$. Or the continuous integral over Z may be approximated with a sum over a smaller finite subset of Monte-Carlo samples $Z_i \sim \mathrm{N}(0,1)$. For each sample, $\max_{x' \in X} \mu^n(x') + \tilde{\sigma}^n(x', 0'x, s)Z_i$ is continuously optimized over the full X with an off-the-shelf non-linear optimizer and the average of optimizer outputs is used as an estimate of $\mathrm{KG}^{\mathrm{CRN}}(x,s)$ (Wu et al. 2017; Wu and Frazier 2016).

Contrasting with many previous algorithms for common random numbers, this algorithm does not exploit the correlation in noise by simulating pairs of inputs and only using the difference. Instead, note that the compound spheric noise assumption is equivalent to assuming there exists a global offset, $c(s)$, added to all outputs for a single seed and each output has a further unique random deviation $g(x,s)$. In such a scenario, if ρ is large, then allocating simulation budget to learn the optimizer of a single seed may be more informative than allocating the same budget to all unique seeds. The surrogate model may learn the offset $c(s)$ reducing the noise to only $g(x,s)$ and exploiting that $\mathrm{argmax}_x \theta(x,s) \approx \mathrm{argmax}_x \bar{\theta}(x)$. However due to the presence of the uninformative noise, $g(x,s)$, deterministic outputs, and the limited number of alternatives X, evaluating a single seed cannot provide full information about the true optimizer of $\bar{\theta}(x)$. The $\mathrm{KG}^{\mathrm{CRN}}(x,s)$ function measures the value of an observation from an old seed and a new seed in a single function. Over the course of sampling, the procedure can automatically trade off between evaluating multiple x on individual seeds and evaluating x on new seeds, always with the goal of learning the most about the true optimum $\max_x \bar{\theta}(x)$.

4.3 Comparison with Pairwise Sampling

Extending Knowledge Gradient to be able to exploit correlated noise due to common random numbers was previously considered in the Knowledge Gradient with Pairwise Sampling algorithm (Xie et al. 2016). The proposed Gaussian process (and compound spheric noise assumption) is the model we consider here. However, the proposed data acquisition method additionally assumes that each new call to the simulator cannot recall past random number streams $s^{n+1} \notin \{s^1, ..., s^n\}$. Instead, the proposed method exploits noise correlation by allowing the algorithm to either collect a single observation on a new unique stream, $s^{n+1} = n+1$, or, at twice the cost, the algorithm may simultaneously collect two observations on a new unique steam, $s^{n+1} = s^{n+2} = n+1$. However the expected value of information for two *correlated* observations, y^{n+1}, y^{n+2}, cannot be computed in closed form even for finite X. It must either be computed by Monte-Carlo (Wu and Frazier 2016) (Ginsbourger et al. 2010) or, alternatively, an analytically tractable lower bound of the two step ahead improvement may be derived by considering only the *univariate difference* between outputs $y^{n+1} - y^{n+2}$. This latter method is used in the Knowledge Gradient with Pairwise Sampling, adding to the standard Knowledge Gradient for a single sample with a second acquisition function to be separately optimized over pairs of inputs,

$$\begin{aligned} \mathrm{KG}^{\mathrm{PW}}(x_i, x_j) &= \frac{1}{2}\left(\mathbb{E}\left[\max_{x' \in X} \mu^n(x', 0) + \tilde{\tilde{\sigma}}^n(x', 0; x_i, x_j)Z\right] - \max_{x'} \mu^n(x', 0)\right) \qquad (4) \\ \tilde{\tilde{\sigma}}^n(x, 0; x_i, x_j) &= \frac{k^n(x, 0, x_i, s^{n+1}) - k^n(x, 0, x_j, s^{n+1})}{\sqrt{k^n(x_i, s^{n+1}, x_i, s^{n+1}) + k^n(x_j, s^{n+1}, x_j, s^{n+1}) - 2k^n(x_i, s^{n+1}, x_j, s^{n+1})}.} \end{aligned}$$

where $s^{n+1} = n+1$ and the factor of $1/2$ in Equation (4) is required because the improvement consumes twice the units of simulation budget. $\mathrm{KG}^{\mathrm{PW}}(x_i, x_j)$ is optimized over $(x_i, x_j) \in X \times X$ which is expensive if naively optimized by exhaustive evaluation as is required for categorical domains. However in (finite or infinite) numerical domains with a surrogate model, this may be continuously optimized and the candidate in $X \times X$ nearest the optimizer output may be used. The proposed Gaussian Process model is the same as

we use here and therefore the model does infer past offsets $c(s^1),\ldots,c(s^n)$. However, specifically due to the constraint that old random number streams cannot be reused, such information is automatically excluded from being used in the acquisition of future observations.

5 NUMERICAL EXPERIMENTS

We compare our proposed method with standard Knowledge Gradient and Knowledge Gradient with Pairwise sampling in laboratory conditions on synthetic functions. We generate random ground truth functions $\bar{\theta}(X)$ and offsets $c(s)$ and $g(x,s)$ from distributions described below. The parameters used to generate data are known to the Gaussian processes for inference and therefore the only difference between methods is the acquisition function. 800 synthetic ground truth functions and sets of offsets were generated and the three methods described below were applied. One example output of both $\mathrm{KG}^{\mathrm{CRN}}$ and $\mathrm{KG}^{\mathrm{PW}}$ are shown in Figure 2.

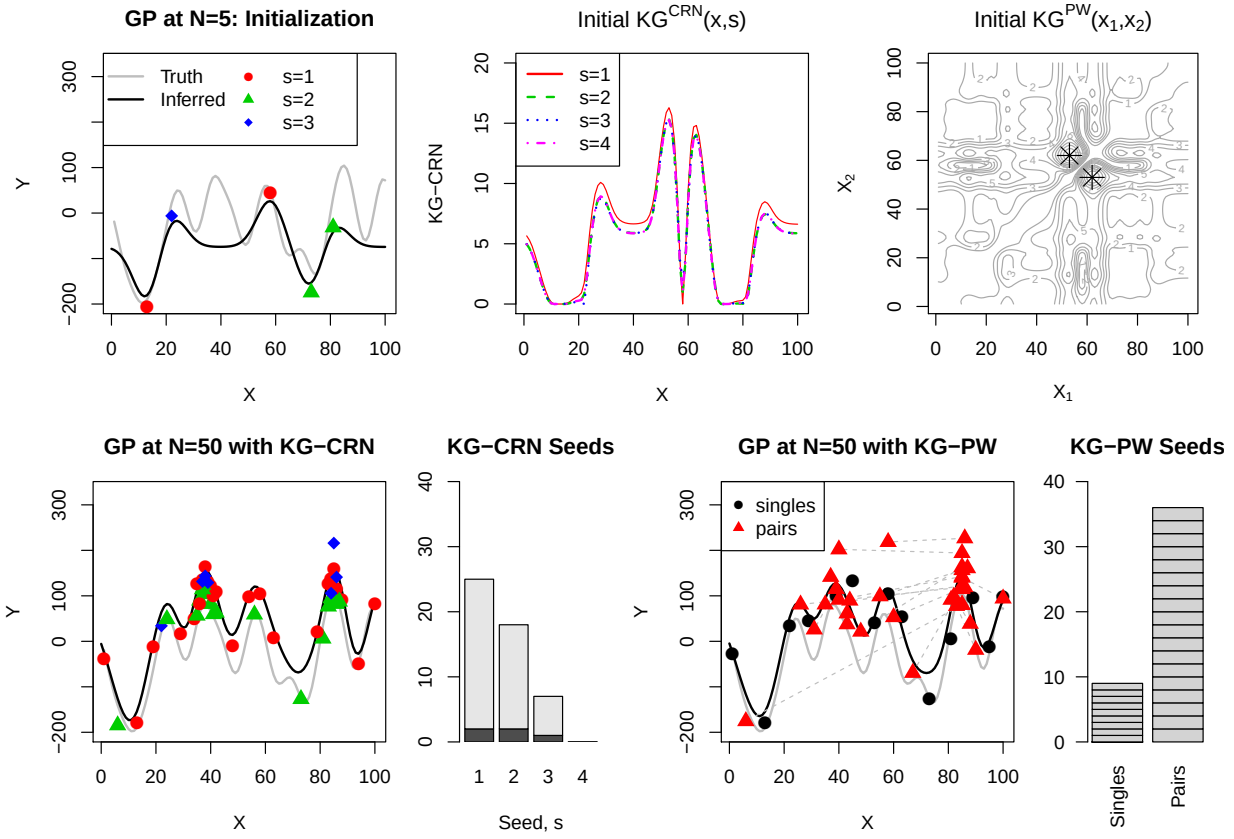

Figure 2: An experiment with $\rho = 0.8$. Top row: Initial state. (L) data and inferred $\mu^5(x,0)$ in black, true function $\bar{\theta}(x)$ in grey. (C) $\mathrm{KG}^{\mathrm{CRN}}(x,s)$ for observed seeds $s \in \{1,2,3\}$ and one new seed $s = 4$ which is also used for the single sample in the pairwise algorithm. (R) the $\mathrm{KG}^{\mathrm{PW}}(x_1,x_2)$, is symmetric in its arguments and the peak marked by the star. Bottom row: final state after 50 samples. (L) Knowledge Gradient for Common Random Numbers after 50 samples and seed allocation, dark bars are initialization. All data has been collected on the initial 3 seeds around peaks at $x \in \{40,85\}$. (R) Pairwise KG, pairs are shown by triangles linked by dashed lines, again single vs pairs seed allocation is given, without initialization. Pairs are frequently collected at the two possible peaks. However due to using new seeds there is much more stochasticity.

5.1 Test Functions

We set $X = \{1, ..., 100\}$, we generate ground truth values from a multivariate normal

$$\bar{\theta}(x) \sim \mathrm{N}(\underline{0}, k_{\bar{\theta}}(X, X))$$

where $k_{\bar{\theta}}(i, j) = 100^2 \exp(-(i-j)^2/2 \cdot 5^2)$. For a given noise correlation, $\rho \in \{0.2, 0.8\}$, each seed offset is sampled from $c(s) \sim N(0, \rho 50^2)$ and the offsets for each (x, s) are given by $g(x, s) \sim \mathrm{N}(0, (1-\rho)50^2)$ and all generated values are held constant throughout each experiment. We run experiments with low noise correlation $\rho = 0.2$ and high noise correlation $\rho = 0.8$ holding the total noise constant $\eta^2 + \sigma^2 = 50^2$. Because the total noise is held constant, the standard Knowledge Gradient will perform exactly the same in both cases.

5.2 Algorithms and Performance

All algorithms are initialized with 5 observations distributed by Latin hypercube sampling, i.e., five equally spaced intervals over $\{1, .., 100\}$ each containing a randomly selected x. The initial seed allocation is $\{1, 1, 2, 2, 3\}$ which are randomly shuffled and paired with the initial $X^5 = \{x^1, ..., x^5\}$ from the Latin hypercube. For inference, we make the true parameters known to the fitting Gaussian process regression models. Given that the input set X is finite, the $\mathrm{KG}^{\mathrm{CRN}}(x, s)$ and $\mathrm{KG}^{\mathrm{PW}}(x_i, x_j)$ functions can be computed exactly and note that $\mathrm{KG}^{\mathrm{PW}}(x_i, x_j)$ requires more computation. The following three methods are applied to find the optimal $\bar{\theta}(x)$ from noisy observations:

- **Knowledge Gradient for Common Random Numbers.** At each time step, the $\mathrm{KG}^{\mathrm{CRN}}(x, s)$ function is exhaustively evaluated for all $x \in X$ and for all seeds in the observation history and one new seed $s \in \{1, .., \max_i s^i, \max_i s^i + 1\}$. The seed history grows when the algorithm decides to sample a new seed and accordingly the search space for optimizing $\mathrm{KG}^{\mathrm{CRN}}(x, s)$ always includes one new extra unobserved seed. Each seed requires $|X| = 100$ calls to $\mathrm{KG}^{\mathrm{CRN}}(x, s)$ and in these experiments there are typically up to 5 seeds hence 500 calls.
- **Knowledge Gradient.** (Frazier, Powell, and Dayanik 2009) We make two modifications to the Knowledge Gradient for common random numbers, firstly the associated seed values of the observed inputs from initialization are overwritten to unique integers $\{s^1, ..., s^5\} = \{1, .., 5\}$. Secondly, by exhaustive evaluation, we optimise the $\mathrm{KG}^{\mathrm{CRN}}(x, s)$ function for a new seed only, $s = n + 1$, thereby removing any benefit due to common random numbers in inference and in acquisition recovering standard Knowledge Gradient. This requires $|X| = 100$ calls to $\mathrm{KG}^{\mathrm{CRN}}(x, s)$.
- **Knowledge Gradient with Pairwise Sampling.** (Xie, Frazier, and Chick 2016) We make two modifications to the Knowledge Gradient for common random numbers, firstly we optimise the $\mathrm{KG}^{\mathrm{CRN}}(x, s)$ function for $s = n + 1$ only, reproducing the single sample acquisition function. Secondly, we also optimise the $\mathrm{KG}^{\mathrm{PW}}(x, x')$ acquisition function over all $X \times X$ by exhaustive evaluation of all possible $|X||X|/2 = 5{,}000$ unique input pairs . This is much more expensive however removes any possible deficiencies due to optimizer implementation and is therefore a best case scenario. In practice this takes only 3 to 4 seconds per iteration and may determine two new samples.

Observations are sequentially collected starting from 5 up to 50 and at each time step the recommended x value is found by exhaustive evaluation over X and recorded,

$$x_r^n = \mathrm{argmax} \mu^n(x, 0).$$

For evaluation, each experiment is repeated $K = 800$ times with different $\bar{\theta}^k(x)$, $c^k(s)$ and $g^k(x, s)$ for $k \in \{1, ..., 800\}$. We measure the average opportunity cost, the difference between the recommended ground truth value $\bar{\theta}^k(x_r^{nk})$ and best possible $\bar{\theta}^k(x)$ (note that these measurements are not known to the

algorithms),

$$\text{Opportunity Cost at time } n = \frac{1}{K}\sum_{k=1}^{K}\left[\max\bar{\theta}(x) - \bar{\theta}(x_r^n)\right]_k.$$

where $[..]_k$ denotes experiment k to avoid cluttering notation. For each of the CRN algorithms, we also report the relative frequency of reusing an old seed at each time step n,

$$\text{Frequency of reusing an old seed at time } n = \frac{1}{K}\sum_{k=1}^{K}\left[\mathbb{1}_{\{s^n\in\{s^1,\ldots,s^{n-1}\}\}}\right]_k.$$

Note that for Pairwise Sampling, even if a pair is sampled at every time step, the first seed of each pair will not be in the history therefore this frequency is upper bounded by 0.5. The initial and final states of one experiment with high correlation are shown in Figure 2 and Opportunity cost and seed reuse are reported in Figure 3.

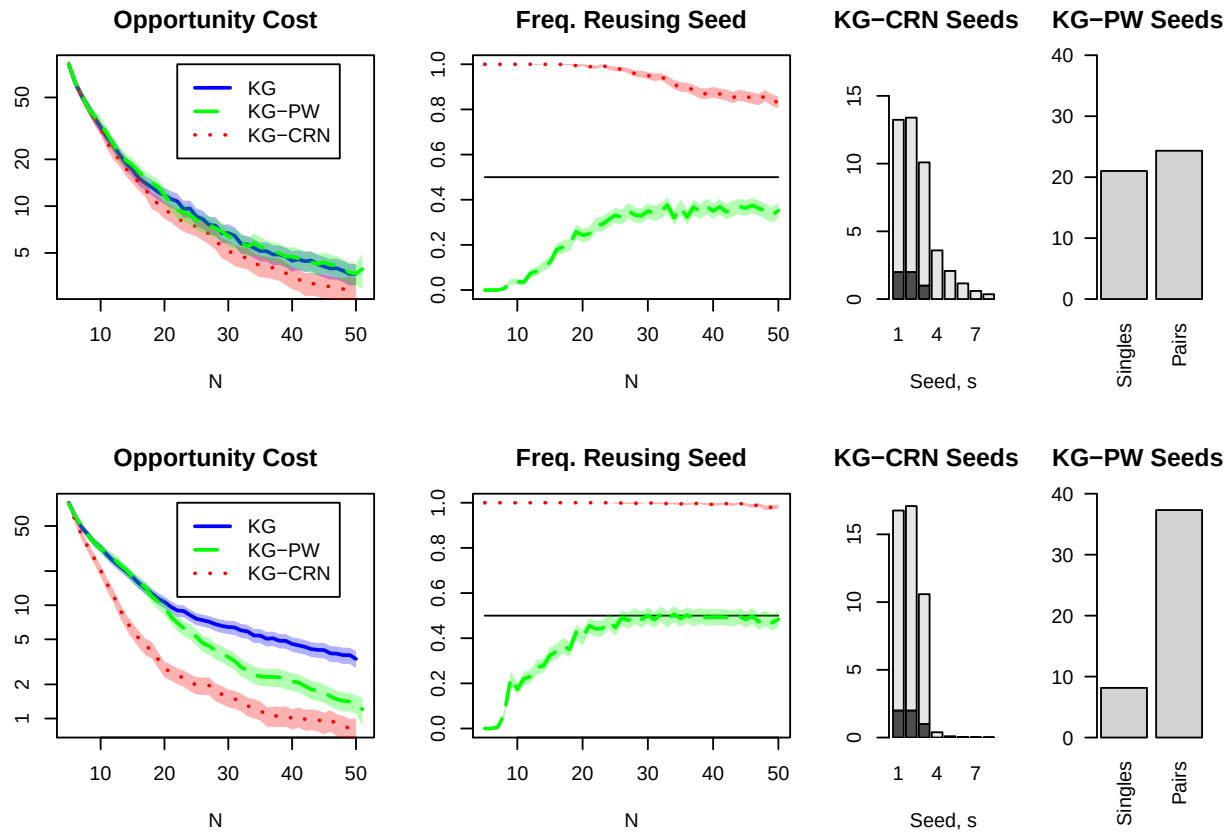

Figure 3: Top row: $\rho = 0.2$, bottom: $\rho = 0.8$. (L) Average opportunity cost for increasing sampling budget with 95% confidence intervals. (C-L) The average frequency of reusing a seed during sampling, dashed line at 0.5, the upper bound for KG-PW. (C-R) The average final number of samples allocated to each seed by the KG-CRN, the dark bars are the initialisation samples. (R) The average final number of seeds queried once (singles) and twice (pairs) by the KG-PW excluding the initialization seeds. For small ρ, there is little benefit from correlated noise. For large ρ, opportunity cost reduces faster for CRN methods, more pairs are sampled by the Pairwise algorithm, likewise more budget is allocated to old seeds by the Common Random Number algorithm which converges significantly faster.

5.3 Results

The top row of Figure 3 shows results for the low correlation benchmarks. All algorithms have similar opportunity cost, both CRN methods still utilise many old seeds yet this does not significantly affect performance. For KG-CRN, the frequency of reusing an old seed decreases for larger budgets, the early samples are allocated to old seeds for exploration, and new seeds are sampled later on in optimization for exploitation to learn about the peak. However the KG-PW method does the opposite, initially sampling more new seeds, filling space with singletons to explore and sampling pairs later in the optimisation to exploit and compare peaks. For high correlation, see Figure 3 bottom row, the KG-CRN method significantly outperforms both baselines. The KG-PW initially samples singletons and performs similarly to KG, and soon hits the upper bound of seed reuse, 0.5. Then it begins to outperform KG. Whereas the KG-CRN method is free to reuse old seeds and very rarely goes to a new seed even for larger budgets. In the high correlation case, with this model free compound spheric noise assumption, optimizing a single seed is much better than going to new seeds.

6 CONCLUSION

We proposed the Knowledge Gradient for Common Random Numbers, a simple generalization of the standard Knowledge Gradient that allows the reuse of old random number streams dramatically improving sample efficiency. This also avoids the need to consider pairs of inputs and optimization over an exponentially larger search domain while simultaneously finding vastly better optima given the same sampling budget. In further work, we intend to investigate noise models that are informed by the inputs X and the interaction of learning extra hyper parameters of the Gaussian Process model. We also aim to apply the new method to more realistic problems and derive theoretical properties.

ACKNOWLEDGMENTS

The first author's PhD is funded by Engineering and Physical Sciences Research Council, UK.

REFERENCES

Birattari, M., T. Stützle, L. Paquete, and K. Varrentrapp. 2002. "A Racing Algorithm for Configuring Metaheuristics". In *Proceedings of the 4th Annual Conference on Genetic and Evolutionary Computation*, 11–18. San Francisco, California, USA: Morgan Kaufmann Publishers Inc.

Birattari, M., Z. Yuan, P. Balaprakash, and T. Stützle. 2010. "F-Race and Iterated F-Race: An Overview". In *Experimental Methods for the Analysis of Optimization Algorithms*, 311–336. Berlin, Heidelberg: Springer.

Branke, J., S. E. Chick, and C. Schmidt. 2007. "Selecting a Selection Procedure". *Management Science* 53(12):1916–1932.

Chen, C.-H., and L. H. Lee. 2010. *Stochastic Simulation Optimization: An Optimal Computing Budget Allocation*. Singapore: World Scientific.

Chen, X., B. E. Ankenman, and B. L. Nelson. 2012, March. "The Effects of Common Random Numbers on Stochastic Kriging Metamodels". *ACM Transactions on Modeling and Computer Simulation* 22(2):7:1–7:20.

Chick, S. E., and K. Inoue. 2001. "New Two-Stage and Sequential Procedures for Selecting the Best Simulated System". *Operations Research* 49(5):732–743.

Frazier, P., W. Powell, and S. Dayanik. 2008. "A Knowledge-Gradient Policy for Sequential Information Collection". *SIAM Journal on Control and Optimization* 47(5):2410–2439.

Frazier, P., W. Powell, and S. Dayanik. 2009. "The Knowledge-Gradient Policy for Correlated Normal Beliefs". *INFORMS Journal on Computing* 21(4):599–613.

Fu, M. C., J. . Hu, C. . Chen, and X. Xiong. 2004. "Optimal Computing Budget Allocation Under Correlated Sampling". In *Proceedings of the 2004 Winter Simulation Conference, 2004.*, edited by R. G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, 603–612. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Ginsbourger, D., R. Le Riche, and L. Carraro. 2010. "Kriging Is Well-Suited to Parallelize Optimization". In *Computational Intelligence in Expensive Optimization Problems*, edited by Y. Tenne and C.-K. Goh, 131–162. Berlin, Heidelberg: Springer.

Görder, B., and M. Kolonko. 2019, January. "Ranking and Selection: A New Sequential Bayesian Procedure for Use with Common Random Numbers". *ACM Transactions on Modeling and Computer Simulation* 29(1):2:1–2:24.

Gupta, S. S., and K. J. Miescke. 1996. "Bayesian Look Ahead One-Stage Sampling Allocations for Selection of the Best Population". *Journal of Statistical Planning and Inference* 54(2):229–244.

Hennig, P., and C. J. Schuler. 2012. "Entropy Search for Information-Efficient Global Optimization". *Journal of Machine Learning Research* 13:1809–1837.

Jones, D. R., M. Schonlau, and W. J. Welch. 1998. "Efficient Global Optimization of Expensive Black-Box Functions". *Journal of Global Optimization* 13(4):455–492.

Kim, S.-H., and B. L. Nelson. 2006. "Selecting the Best System". *Handbooks in Operations Research and Management Science* 13:501–534.

Krause, A., and C. S. Ong. 2011. "Contextual Gaussian Process Bandit Optimization". In *Advances in Neural Information Processing Systems*, 2447–2455. Cambridge, Massachusetts: The MIT Press.

Kushner, H. J. 1964. "A New Method of Locating the Maximum Point of an Arbitrary Multipeak Curve in the Presence of Noise". *Journal of Basic Engineering* 86(1):97–106.

Mockus, J., V. Tiesis, and A. Zilinskas. 1978. "Toward global optimization". *The Application of Bayesian Methods for Seeking the Extremum* 2:117–128.

Nelson, B. L., and F. J. Matejcik. 1995. "Using Common Random Numbers for Indifference-Zone Selection and Multiple Comparisons in Simulation". *Management Science* 41(12):1935–1945.

Rasmussen, C. E. 2004. *Gaussian Processes in Machine Learning*, 63–71. Berlin, Heidelberg: Springer.

Sacks, J., W. J. Welch, T. J. Mitchell, and H. P. Wynn. 1989. "Design and Analysis of Computer Experiments". *Statistical Science* 4(4):409–423.

Smith-Miles, K., D. Baatar, B. Wreford, and R. Lewis. 2014. "Towards Objective Measures of Algorithm Performance Across Instance Space". *Computers & Operations Research* 45:12–24.

Srinivas, N., A. Krause, S. M. Kakade, and M. Seeger. 2009. "Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design". *arXiv preprint arXiv:0912.3995*.

Wu, J., and P. Frazier. 2016. "The Parallel Knowledge Gradient Method for Batch Bayesian Optimization". In *Advances in Neural Information Processing Systems*, 3126–3134. Cambridge, Massachusetts: The MIT Press.

Wu, J., M. Poloczek, A. G. Wilson, and P. Frazier. 2017. "Bayesian Optimization with Gradients". In *Advances in Neural Information Processing Systems*, 5267–5278. Cambridge, Massachusetts: The MIT Press.

Xie, J., P. I. Frazier, and S. E. Chick. 2016. "Bayesian Optimization via Simulation with Pairwise Sampling and Correlated Prior Beliefs". *Operations Research* 64(2):542–559.

AUTHOR BIOGRAPHIES

MICHAEL PEARCE is a PhD student at the University of Warwick's Complexity Science Centre. He graduated from the University of Bristol in 2009 with MSci. in Mathematics and in 2015 with an MSc in Complexity Science from the University of Warwick. He has interned a Google Deepmind and is currently interning at Uber AI Labs. His interests are in various applications of model based stochastic optimization. His e-mail address is m.a.l.pearce@warwick.ac.uk.

MATTHIAS POLOCZEK leads the Bayesian optimization efforts at Uber AI Labs and is an assistant professor at the University of Arizona. The work was done while the second author was affiliated with the University of Arizona in Tucson, AZ. His research interests lie at the intersection of machine learning and optimization. Recently, he has focused on enabling Bayesian optimization for exotic black-box problems that have cheap approximations, provide derivative information, or are formulated over a combinatorial domain. His email address is poloczek@uber.com.

JUERGEN BRANKE is Professor of Operational Research and Systems of Warwick Business School, University of Warwick, UK. He is Area Editor for the Journal of Heuristics and the Journal on Multi Criteria Decision Analysis, and Associate Editor for IEEE Transaction on Evolutionary Computation, and the Evolutionary Computation Journal. His research interests include metaheuristics and Bayesian optimization, multiobjective optimization and decision making, optimization in the presence of uncertainty, and simulation-based optimization. He has published over 180 peer-reviewed papers in international journals and conferences. His e-mail address is Juergen.Branke@wbs.ac.uk.