Proceedings of the 2017 Winter Simulation Conference
W. K. V. Chan, A. D'Ambrogio, G. Zacharewicz, N. Mustafee, G. Wainer, and E. Page, eds.

BAYESIAN SIMULATION OPTIMIZATION WITH INPUT UNCERTAINTY

Michael Pearce
Juergen Branke

University of Warwick
Gibbet Hill Road
Coventry, CV4 7AL, UK

ABSTRACT

We consider simulation optimization in the presence of input uncertainty. In particular, we assume that the input distribution can be described by some continuous parameters, and that we have some prior knowledge defining the probability distribution for these parameters. We then seek the simulation design that has the best *expected* performance over the possible parameters of the input distributions. Assuming correlation of performance between solutions and also between input distributions, we propose modifications of two well-known simulation optimization algorithms, Efficient Global Optimization and Knowledge Gradient with Continuous Parameters, so that they work efficiently under input uncertainty.

1 INTRODUCTION

Simulation optimization, i.e., the search for a design or solution that optimizes some output value of the simulation model, allows to automate the design of complex systems and has many real-world applications. Consequently, various simulation optimization methods have been developed, including metaheuristics such as simulated annealing (Branke, Meisel, and Schmidt 2008) or response surface methods such as Efficient Global Optimization (EGO) (Jones, Schonlau, and Welch 1998) or Knowledge Gradient for Continuous Parameters (KG) (Scott, Frazier, and Powell 2011). These methods can deal with the stochastic output usually produced by stochastic simulation models. However, a simulation is never a perfect model of reality. When constructing the simulation model, the decision maker often faces the challenge of defining proper *input* distributions (eg. the mean of an arrival time distribution) in particular if multiple candidate distributions can fit the input data reasonably well, or the distributions are based on predictions and expert judgement. This issue, generally known as "input uncertainty", has obtained increasing interest of the simulation community in recent years.

This paper adapts two successful and well-known simulation optimization methods, namely EGO and KG, to allow for input uncertainty. We assume that the design as well as the possible input distributions can be described by some continuous parameters each and that one has a probability distribution over the parameter determining the input distribution. This is, for example, the case if the input distribution parameters are estimated by a group of experts, and the different opinions are aggregated into one probability distribution over the input distribution parameter. We furthermore assume that the output of the simulation model is correlated across designs as well as input distribution parameters. Then, given a budget of simulation runs (i.e., runs of the simulation with a given design and input distribution parameter), the goal is to identify the design with the best expected performance over the probability distribution of input distribution parameters.

The paper is structured as follows. We start with an overview of related work in Section 2, followed by a formal definition of the problem in Section 3. Section 4 explains the newly proposed methods for simulation optimization in the presence of input uncertainty, which are then empirically tested and compared

to some benchmarks in Section 5. The paper concludes with a summary and some suggestions for future work.

2 LITERATURE REVIEW

When trying to find the best design based on a small number of simulation runs, Bayesian Optimization based on Gaussian Processes, or Kriging models, have gained much attention. These methods build a surrogate model of the response surface based on a few initial samples and then use an acquisition function (sometimes called infill criterion) to iteratively decide where to sample next to improve the model and find better solutions. The most popular Bayesian Optimization algorithm is the Efficient Global Optimization (EGO) algorithm of Jones, Schonlau, and Welch (1998) that combines a Gaussian Process to interpolate an expensive function and an expected improvement criterion for deciding where to sample next. The SKO algorithm (Huang et al. 2006) extends the EGO algorithm by proposing an acquisition function that also accounts for noise in function evaluations. The Knowledge Gradient for Correlated Normal Beliefs (Frazier, Powell, and Dayanik 2009) is a myopic sampling policy that aims to maximize the new predicted performance after one sample. It can be applied when using a Gaussian Process with a discretized input, has a theoretical basis in dynamic programming and provides myopic and asymptotic guarantees. The Knowledge Gradient policy for Continuous Parameters (Scott, Frazier, and Powell 2011) generalizes the EGO algorithm to noisy functions. Perhaps the most interesting difference between the Knowledge Gradient policy and previous policies is that the Knowledge Gradient accounts for covariance when judging the value of a sample, i.e., the expected improvement takes into account the possibility that as a result of the new sample, the predicted performance at other previously sampled points may change.

The topic of input uncertainty has recently gained significant attention in the simulation community, for a general introduction see, e.g., Lam (2016). However, so far simulation optimization usually assumes the input distributions are known. Zhou and Xie (2015) argue that input uncertainty should be taken into account in simulation optimization and discuss various risk formulations that could be used to capture input uncertainty in the optimization criterion including worst-case optimization (minimizing risk) and expected performance (risk neutral) and demonstrate that this may be beneficial.

One area of simulation optimization that focuses on the case of few alternatives is ranking and selection, see, e.g., Branke, Chick, and Schmidt (2007); Chen et al. (2015), and a few recent papers in this area have considered input uncertainty, although mostly in the sense of worst-case performance.

Song, Nelson, and Hong (2015) examine the impact of input uncertainty on indifference zone ranking and selection procedures. They conclude that a straightforward application of IZ selection can invalidate the probability of correct selection guarantee. Fan, Hong, and Zhang (2013) and Gao et al. (2016) consider a discrete set of alternatives and scenarios, and no correlation between either alternatives or scenarios. They aim to identify the best alternative according to the worst case input distribution. Fan, Hong, and Zhang (2013) propose an indifference-zone based ranking and selection procedure that works in two-stages: first it identifies for each design the worst case, then it identifies the best design based on the identified worst case scenario. On the other hand, the algorithm by Gao et al. (2016) is based on the optimal computing budget allocation (OCBA) framework. Zhang and Ding (2016) while still assuming independence among alternatives, take into account correlations among the performance measures of an alternative under different input distributions. They still aim for the best alternative in the worst case, and propose an algorithm based on the Knowledge Gradient framework.

Different from previous work, in this paper we focus on a continuous set of alternatives (continuous parameter to be optimized), a set of input distributions described by a continuous parameter, and the correlation between alternatives and across input distributions. Also, our aim is not to find the alternative which is best in the worst case, but the solution that has the best expected performance based on a prior distribution on the input distribution parameter. We propose myopic sampling strategies based on EGO and KG.

3 PROBLEM FORMULATION

Given a simulation model with a tunable parameter that specifies the possible designs $a \in A$ and an input distribution with parameter $x \in X$ (henceforth simply called "input") following an assumed distribution $\mathbb{P}[x]$ independent of a. When running a simulation with design a and input x, we observe an output following an unknown noisy function $f(x,a) = \theta(x,a) + \varepsilon$ where $\theta(x,a) = \mathbb{E}[f(x,a)]$ is a deterministic latent function defining the expected outcome and $\varepsilon \sim N(0, \sigma_\varepsilon^2)$ is independent white noise with constant variance. The objective of the user is to identify the design a that maximizes the expected performance simultaneously across the input parameter distribution

$$F(a) = \int_X \theta(x,a)\mathbb{P}[x]dx. \tag{1}$$

We assume we have a fixed budget of N samples (simulation runs with a specific design and input), and that we can sample iteratively, i.e., we can select the design and the input from which to sample performance $f(x,a)$ based on the information collected so far.

Although for reasons of simplicity we use scalar notation, the approach applies to multi-dimensional inputs and designs.

4 SAMPLING METHODS

In this section, we show how to modify two well known global optimization methods, Efficient Global Optimization (EGO) and Knowledge Gradient with Continuous Parameters (KG) to the case of input uncertainty. We assume that we can use a Gaussian Process to model the underlying latent performance function $\theta(x,a)$ given the data observed. For simplicity, let us denote the location and value of the n-th observation by x^n, a^n and y^n. Given all the data collected $(x^1,a^1,y^1),...,(x^n,a^n,y^n)$, we define $\tilde{X}^n = \{(x^1,a^1),...,(x^n,a^n)\}$ and $Y^n = (y^1,...,y^n)$, and the sigma algebra $\mathscr{F}^n$ generated by all the data $\{(x^1,a^1,y^1),\ldots,x^n,a^n,y^n)\}$. A Gaussian Process regression model treats the underlying latent function, $\theta(x,a)$, at any given finite set of points (e.g. (x,a) and (x',a')) as a multivariate Gaussian random variable whose mean and covariance are given by

$$\begin{aligned}\mathbb{E}[f(x,a)|\mathscr{F}^n] &= \mu^n(x,a) \\ &= \mu^0(x,a) - k^0((x,a),\tilde{X}^n)(k^0(\tilde{X}^n,\tilde{X}^n) + I\sigma)^{-1}(Y^n - \mu^0(\tilde{X}^n)) \end{aligned} \tag{2}$$

$$\begin{aligned}Cov[f(x,a),f(x',a')|\mathscr{F}^n] &= k^n((x,a),(x',a')) \\ &= k^0((x,a),(x',a')) - k^0((x,a),\tilde{X}^n)(k^0(\tilde{X}^n,\tilde{X}^n) + I\sigma)k^0(\tilde{X}^n,(x',a')) \end{aligned} \tag{3}$$

where $\mu^0(x,a)$ is the prior mean which is typically set to $\mu^0(x,a) = 0$ and $k^0((x,a),(x',a'))$ is the kernel of the Gaussian Process that allows the user to encode known properties of the latent function $\theta(x,a)$ such as smoothness and periodicity. In Section 5, we use the popular squared exponential kernel, though also the Matern class of kernels has been widely used for simulation optimization.

4.1 Efficient Global Optimization for Input Uncertainty

The well-known EGO algorithm sequentially collects objective function evaluations and was originally designed for deterministic global optimization problems. It starts by evaluating the function at a randomly distributed set of inputs to the function. Then, the location of the next sample is determined by maximizing the expected improvement of a new function evaluation over the current best evaluation. The predictive distribution of the new noiseless sample is given by $y^{n+1} \sim N(\mu^n(a), k^n(a,a))$ and so the expected

improvement over the current best of a hypothetical sample at parameter a is given by

$$\begin{aligned} EGO(a) &= \mathbb{E}[\max\{y^1,...,y^{n+1}\}] - \max_{i=1,...,n} y^i & (4)\\ &= \mathbb{E}[\max\{0, y^{n+1} - \max_{i=1,...,n} y^i\}] & (5)\\ &= \Delta\Phi(\Delta/\sigma) - \sigma\phi(\Delta/\sigma) & (6)\end{aligned}$$

where $\phi(x)$ and $\Phi(x)$ are the standard normal density and cumulative functions, $\Delta = \mu^n(a) - \max_{i=1,...,n} y^i$ and $\sigma = \sqrt{k^n(a,a)}$. Note that the expectation in Equation 5 is over the maxima of two linear functions, one of which is a constant, with a normally distributed input. When function evaluations are deterministic, the previous best sample value does not change with the new sample. The EGO acquisition function given in Equation 6 is easily and cheaply optimised to find the location of the most promising new function evaluation, (x^{n+1}, a^{n+1}), and then the objective function is evaluated and a stochastic $y^{n+1} = f(x^{n+1}, a^{n+1})$ is observed. The Gaussian Process model is updated and the search for the location of the next sample is performed again, a new function value is observed and the algorithm repeats until the budget of function evaluations is exhausted.

Adapting this method to allow for noise and input uncertainty we need to answer two questions, namely what is the value of the current best upon which we aim to improve, and what is the predictive distribution of the value after a new hypothetical sample is generated. We calculate a prediction of the current best upon which we aim to improve by adapting Equation 1, replacing the unknown $\theta(x,a)$ with the model prediction, $\mu^n(x,a)$, and the continuous integral over X can be replaced by a Monte-Carlo integral, a summation over N_X random inputs $X_{MC} = \{x_1,...,x_{N_X}\} \sim \mathbb{P}[x]$

$$\hat{F}^n(a) = \frac{1}{N_X} \sum_{x_i \in X_{MC}} \mu^n(x_i, a). \tag{7}$$

Then, the current best upon which we aim to improve is found by maximizing $\hat{F}^n(a)$ over a. This maximization can be done cheaply using any off-the-shelf optimization algorithm as the function is based only on posterior means. This is further explained in Section 5. N_X is a parameter that may be chosen by the user to determine accuracy and in Section 5 we use $N_X = n$, so that accuracy increases over the run.

In order to answer the second question, we require an updating formula for the posterior mean to derive the predictive distribution of $\hat{F}^{n+1}(a)$ given only the data available at time n. By setting the posterior mean and covariance after n samples, $\mu^n(x,a), k^n(x,a,x',a')$, as the prior mean and covariance in Equation 2, we can write the formula for the mean for the $(n+1)^{th}$ sample as

$$\mu^{n+1}(x,a) = \mu^n(x,a) + \frac{k^n((x,a),(x,a)^{n+1})}{k^n((x,a)^{n+1},(x,a)^{n+1}) + \sigma_\varepsilon^2}(y^{n+1} - \mu^n(x,a)), \tag{8}$$

where $(x,a)^{n+1}$ is determined by the algorithm and y^{n+1} is unknown. However the Gaussian Process model provides a prior predictive distribution for the new function value

$$y^{n+1} \sim N(\mu^n((x,a)^{n+1}), k^n((x,a)^{n+1},(x,a)^{n+1}) + \sigma_\varepsilon^2)$$

and therefore it is possible to factorize Equation 8 as follows

$$\mu^{n+1}(x,a) = \mu^n(x,a) + \tilde{\sigma}^n(x,a,(x,a)^{n+1})Z \tag{9}$$

where $Z \sim N(0,1)$ is the z-score of y^{n+1} on it's prior predictive distribution, and $\tilde{\sigma}^n((x,a);(x,a)^{n+1})$ is a deterministic function of x, a parametrized by $(x,a)^{n+1}$ that is the additive update to the posterior mean scaled by Z

$$\tilde{\sigma}^n((x,a);(x,a)^{n+1}) = \frac{k^n((x,a),(x,a)^{n+1})}{\sqrt{k^n((x,a)^{n+1},(x,a)^{n+1}) + \sigma_\varepsilon^2}}.$$

Therefore the predictive distribution of the new posterior mean is then given by

$$\mu^{n+1}(x,a) \sim N(\mu^n(x,a),\ \tilde{\sigma}((x,a);(x,a)^{n+1})^2)$$

and the predicted performance after a new sample can then be written as

$$\begin{aligned}
\hat{F}^{n+1}(a;(x,a)^{n+1}) &= \frac{1}{N_X}\sum_{x_i\in X_{MC}} \mu^{n+1}(x_i,a) && (10)\\
&= \frac{1}{N_X}\sum_{x_i\in X_{MC}} \mu^{n}(x_i,a) + Z\frac{1}{N_X}\sum_{x_i\in X_{MC}} \tilde{\sigma}^n(x_i,a;(x,a)^{n+1}) && (11)\\
&= \hat{F}^n(a) + Z\,\tilde{\Sigma}^n(a;(x,a)^{n+1}). && (12)
\end{aligned}$$

where $\tilde{\Sigma}^n(a,(x,a)^{n+1})$ is given by the final term in Equation 11. The predictive distribution of the new design a after a random sample at $(x,a)^{n+1}$ is then given by

$$\hat{F}^{n+1}(a;(x,a)^{n+1}) \sim N\left(\hat{F}^n(a), \tilde{\Sigma}^n(a;(x,a)^{n+1})^2\right).$$

The above summation does not include the sampled input x^{n+1} however this can be included with a unique weighting and is discussed in Section 4.3. The expression gives the updated value for arbitrary a caused by a new sample at $(x,a)^{n+1}$. If we only consider the updated value at the sampled design $a^{n+1}=a$ then the expected improvement over the previous optimal predicted design is given by

$$\begin{aligned}
EGO(x,a) &= \mathbb{E}[\max\{\max_{a'}\hat{F}^n(a'), \hat{F}^{n+1}(a;(x,a))\}] - \max_{a'}\hat{F}^n(a') && (13)\\
&= \mathbb{E}[\max\{0, \hat{F}^n(a) - \max_{a'}\hat{F}(a') + Z\tilde{\Sigma}^n(a;(x,a))\}] && (14)\\
&= \Delta\Phi(\Delta/\tilde{\Sigma}) - \tilde{\Sigma}\phi(\Delta/\tilde{\Sigma}) && (15)
\end{aligned}$$

where $\Delta = \hat{F}^n(a) - \max_{a'}\hat{F}^n(a')$ and $\tilde{\Sigma} = \tilde{\Sigma}^n(a;(x,a))$. Samples are sequentially allocated to (x,a) pairs that maximise $EGO(x,a)$. After each sample, the Monte-Carlo points, X_{MC}, may be regenerated to avoid overfitting to one particular discretization of the distribution $\mathbb{P}[x]$.

4.2 Knowledge Gradient for Input Uncertainty

The new sample at $(x,a)^{n+1}$ causes the posterior mean to change at other designs and inputs according to the additive update $Z\tilde{\sigma}^n((x,a);(x,a)^{n+1})$. Therefore the design a with the highest value after the new sample varies and may not be the previous best nor the sampled design a^{n+1}. The EGO algorithm compares the old highest value (which is assumed to be constant) with the value of the new sampled design. The Knowledge Gradient compares the old highest value with the new highest value which may or may not correspond to the previous best design or the design sampled. We define the set of previously evaluated designs as $A^n = \{a^1,...,a^n\}$ and A^{n+1} includes the next sampled design a assuming $a^{n+1}=a$. Traditional Knowledge Gradient for Continuous Parameter using Gaussian Processes (Scott, Frazier, and Powell 2011) allocates samples to maximize the following expected improvement

$$\begin{aligned}
KG(a) &= \mathbb{E}[\max_{a''\in A^{n+1}}\{\mu^{n+1}(a'')\}] - \max_{a'\in A^{n+1}}\{\mu^n(a')\} && (16)\\
&= \mathbb{E}[\max\{\mu^{n+1}(a^1),...,\mu^{n+1}(a^n),\mu^{n+1}(a)\}] - \max_{a'\in A^{n+1}}\{\mu^n(a')\} && (17)\\
&= \mathbb{E}[\max\{\mu^n(a^1)+Z\tilde{\sigma}(a^1;a),...,\mu^n(a)+Z\tilde{\sigma}(a;a)\}] - \max_{a'\in A^{n+1}}\{\mu^n(a')\} && (18)\\
&= \mathbb{E}[\max\{c_1+Zm_1,...,c_{n+1}+Zm_{n+1}\}] && (19)
\end{aligned}$$

where $c_i = \mu^n(a^i) - max_{a' \in A^{n+1}} \mu^n(a')$ and $m_i = \tilde{\sigma}^n(a^i, a)$. The final expectation is the maximum of $(n+1)$ linear functions with a normally distributed argument and may be computed using Algorithm 1 in (Scott, Frazier, and Powell 2011). In contrast, $EGO(a)$ is the maximum over only two linear functions with a normally distributed argument as a result of not accounting for changes in the posterior mean at unsampled designs $a \neq a^{n+1}$. We adapt $KG(a)$ to the input uncertain case $KG(x,a)$ by replacing $\mu^n(a)$ and $\tilde{\sigma}(a, a^{n+1})$ in the above equations with their Monte-Carlo counterparts $\hat{F}^n(a)$ and $\tilde{\Sigma}(a;(x,a)^{n+1})$.

$$
\begin{aligned}
KG(a,x) &= \mathbb{E}[\max_{a \in A^{n+1}} \{\hat{F}^{n+1}(a)\}] - \max_{a' \in A^{n+1}} \{\hat{F}^n(a')\} && (20)\\
&= \mathbb{E}[\max\{\hat{F}^n(a^1) + Z\tilde{\Sigma}(a^1;(x,a))), ..., \hat{F}^n(a) + Z\tilde{\Sigma}(a;(x,a))\}] - \max_{a' \in A^n} \{\hat{F}^{n+1}(a')\} && (21)\\
&= \mathbb{E}[\max\{c_1 + Zm_1, ..., c_{n+1} + Zm_{n+1}\}] && (22)
\end{aligned}
$$

where $c_i = \hat{F}^n(a^i) - \max_{a' \in A^{n+1}} \hat{F}^n(a')$ and $m_i = \tilde{\Sigma}^n(a^i, (x,a))$. The final expectation may be evaluated using Algorithm 1 below that takes a vector of gradients $\underline{m}$ and intercepts $\underline{c}$ and returns the expectation over the normally distributed Z. As with the adapted EGO algorithm, this KG for input uncertainty algorithm also has a single parameter N_X that determines the granularity of the MonteCarlo integration and can be chosen by the user. In our benchmarks we set $N_X = n$ so that the accuracy increases with the sample size over the run.

4.3 Including the Sampled Input in the Monte-Carlo Integral

The proposed Monte-Carlo integral may be improved by importance sampling by setting $X_{MC} \sim G[x]$ where $G[x]$ is a proposal distribution such that $G[x] \propto \mathbb{P}[x]\tilde{\sigma}((x,a),(a,x)^{n+1})$. Meaning that in order to minimise error the Monte-Carlo integral should focus samples where the density of inputs is high and also where the new function evaluation has great affect on the prediction of other inputs. The above proposed methods set $X_{MC} \sim \mathbb{P}[x]$, focusing the integration only where $\mathbb{P}[x]$ is high. When using a stationary kernel, one may set X_{MC} to be a cluster around x^{n+1} so that samples are allocated to where $G[x] \propto \tilde{\sigma}((a,x),(a,x)^{n+1})$, focussing where changes in the guassian process are high. However in practice this leads to expensive computation and the EGO and KG functions become less smooth and harder to optimise. In order to appropriatly focus the Monte-Carlo integration whilst still being generelizable to any input uncertainty distibrution and kernel we therefore propose a basic mix of these two possible approaches. Using a standard Monte-Carlo integral as well as the sampled point x^{n+1} which may be seen as a cluster of size 1 focussed where $\tilde{\sigma}((x,a),(a^{n+1},x^{n+1}))$ is likely to be greatest. The sampled input is not a sample from $\mathbb{P}[x]$ therefore simply including the input in the Monte-Carlo sum would lead to biases, for example if $\mathbb{P}[x^{n+1}] = 0$, the sampled input should not be included at all. Therefore we include the sampled input with a unique weight that assumes it is from a single point from a uniform distribution $G[x] = 1/V_X$ where $V_X = \int_X dx$ is the volume of the input parameter domain. Therefore the importance weight is simply $\mathbb{P}[x^{n+1}]/G[x^{n+1}] = \mathbb{P}[x^{n+1}]V_X$. Therefore we may adjust the Monte-Carlo integrals as follows

$$
\begin{aligned}
\hat{F}^n(a) &= \frac{1}{N_X+1}\left(\sum_{x_i \in X_{MC}} \mu^n(x_i,a) + \mathbb{P}[x^{n+1}]V_x\mu^n(x^{n+1},a)\right) && (23)\\
\tilde{\Sigma}^n(a;(x,a)^{n+1}) &= \frac{1}{N_X+1}\left(\sum_{x_i \in X_{MC}} \tilde{\sigma}^n((x_i,a);(x,a)^{n+1}) + \mathbb{P}[x^{n+1}]V_x\tilde{\sigma}^n((x^{n+1},a);(x,a)^{n+1})\right). && (24)
\end{aligned}
$$

Secondly, we combine the original Monte-Carlo integral and the single sample Monte-Carlo integral according to their sample size N_X and 1. The modified Monte-Carlo integrals may be directly used in the $EGO(x,a)$ and $KG(x,a)$ and are used for the numerical experiments in Section 5.

Algorithm 1 The KG Function The following algorithm takes a vector of intercepts and gradients and finds the epigraph, or "ceiling", of all the linear functions and calculates the expectation over a normally distributed input. $\tilde{Z}$ is the vector of Z values at the vertices of the epigraph, I is the vector of indices of the corresponding linear functions that are part of the epigraph. The algorithm starts with an epigraph of two functions with the lowest gradients, and at each step adds a steeper function and updates the epigraph. All vector indices start from 1.

Require: $\underline{c}$, $\underline{m} \in \mathbb{R}^{n_A}$
  Sort the elements of $\underline{c}$ and $\underline{m}$ in increasing order of $\underline{m}$
  $\underline{c} \leftarrow \underline{c} - \max\{\underline{c}\}$
  $I \leftarrow [1,2]$
  $\tilde{Z} \leftarrow [-\infty, \frac{c_1 - c_2}{m_2 - m_1}]$
  for $i = 3$ **to** n_A **do**
    $endloop \leftarrow False$
    repeat
      $j \leftarrow last(I)$
      $z \leftarrow \frac{c_i - c_j}{m_j - m_i}$
      if $z < last(\tilde{Z})$ **then**
        Delete last element of I
        Delete last element of $\tilde{Z}$
      else
        Add i to end of I
        Add z to end of $\tilde{Z}$
        $endloop \leftarrow True$
      end if
    until *endloop*
  end for
  $\tilde{Z} \leftarrow [\tilde{Z}, \infty]$
  return $\sum_{i=1}^{length(I)} c_{I_i}(\Phi(\tilde{Z}_{i+1}) - \Phi(\tilde{Z}_i)) + m_{I_i}(\phi(\tilde{Z}_i) - \phi(\tilde{Z}_{i+1}))$

5 NUMERICAL BENCHMARK

We apply our algorithms to two benchmarks based on the same test function but with different assumed distributions of the input. The set of inputs is $X = [0,100]$, the set of designs is $A = [0,100]$ and the test function $\theta : X \times A \to \mathbb{R}$. We generate a synthetic test function by sampling from a Gaussian Process with a squared exponential kernel with hyper-parameters $l_X = 10$, $l_A = 10$, $\sigma_0^2 = 1$, $\sigma_\varepsilon^2 = (0.1)^2$, the test function $\theta(x,a)$ is shown in Figure 1 (a) top. The first input parameter distribution is uniform $\mathbb{P}[x] = 1/100$, thus the sampling procedure must sample across all inputs to learn about the best alternative. The second distribution is a triangular distribution $\mathbb{P}[x] = x\frac{2}{10{,}000}$ such that the mode input is $x = 100$ and the mean input is $x = 66.67$ and the sampling procedure must prioritize high $\mathbb{P}[x]$. Given these input parameter distributions, the true $F(a)$ is calculated via numerical integration and shown in Figure 1 (a) bottom.

At the start of sampling, 10 samples are allocated by the random sampling methods described below, the Gaussian process prediction of $\theta(x,a)$ and $\hat{F}^{10}(a)$ after the initial allocation are shown in Figures 1 (b) and (e) for the uniform and triangular distribution cases respectively. Then the sequential methods are used to allocate an additional 90 samples reaching a full budget of 100, Figures 1 (d) and (g) show $\theta(x,a)$ and $\hat{F}^{100}$ after 100 samples have been allocated according to EGO. Then, based on the learned Gaussian Process model, the design a^* with the largest predicted performance over a sample of 1000 inputs, X_R, is

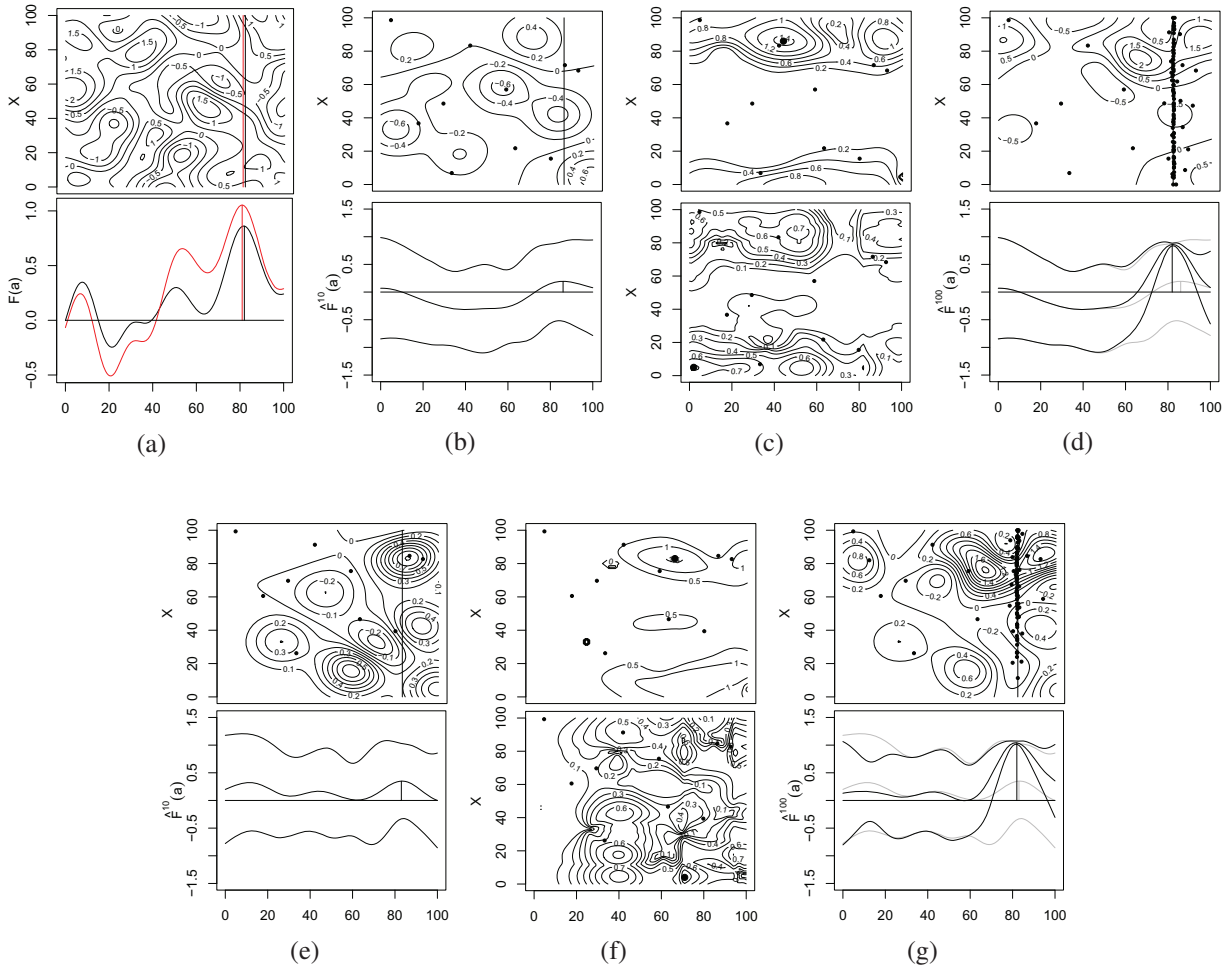


Figure 1: In all plots, A is on the horizontal axis, small points represent function evaluations. (a) $\theta(x,a)$ and $F(a)$ measured using the uniform test inputs (black) and triangular test inputs (red). $\mu^{10}(x,a)$ and $\hat{F}^{10}(a)$ with upper and lower confidence bounds after 10 samples are shown for uniform inputs (b) and triangular inputs (e). (c) and (f) top $EGO(x,a)$, bottom $KG(x,a)$ after the initial 10 samples with uniform inputs (c) and triangular inputs (f), large points show the peaks. (d) and (g) $\mu^{100}(x,a)$ and $\hat{F}^{100}(a)$ (with $\hat{F}^{10}(a)$ in grey) after 100 samples allocated by EGO, (d) uniform and (g) triangular.

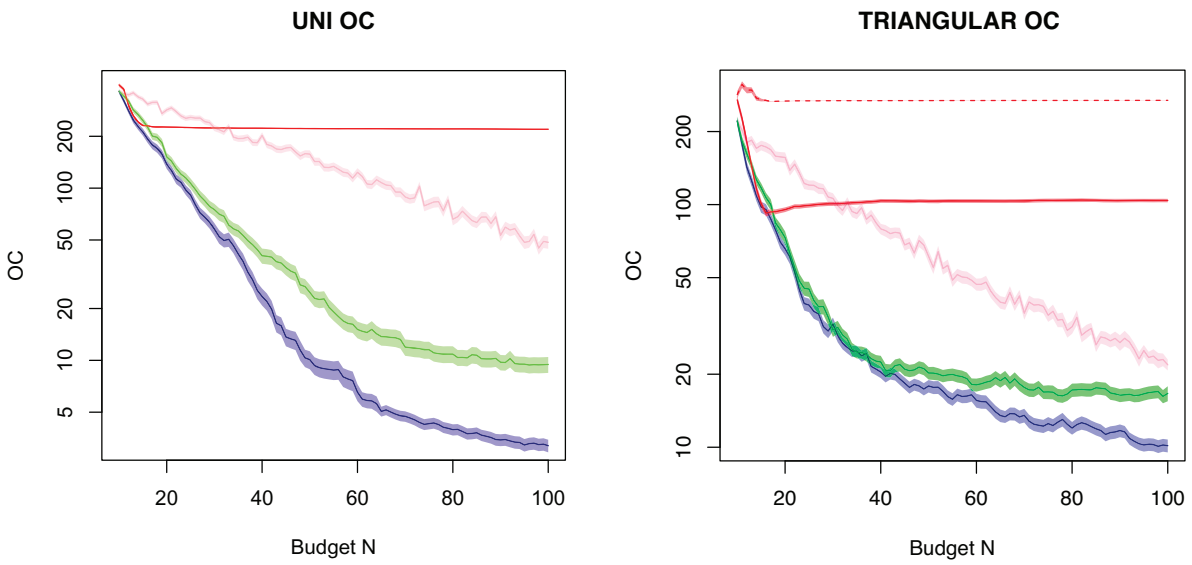
recommended to the user

$$a^* = \underset{a'}{\operatorname{argmax}}\, \hat{F}^N(a') = \underset{a'}{\operatorname{argmax}}\, \frac{1}{1000} \sum_{x_i \in X_R} \mu^N(x_i, a')$$

where $\hat{F}^N(a)$ is optimized by evaluating for all integer values $a \in 0, 1, ..., 100$ and the highest value is used as a seed for sequential parabolic interpolation to find the optimal a^* to high accuracy.

The quality of the sampling procedure is determined by the opportunity cost, the difference in true performance between the design with the highest predicted value and the true best design, which is measured over a separate random sample of 1000 inputs X_{test} that have not been used in the algorithm. The true value of a given alternative is calculated by

$$F(a) = \frac{1}{1000} \sum_{x_i \in X_{test}} \theta(x_i, a).$$



(a) (b)

Figure 2: (a) the opportunity cost when the input distribution is uniform, (b) when the input distribution is triangular. EGO on the mean input (red, solid) and mode input (red, dashed) perform worst, followed by Random Sampling (pink) which is significantly worse than EGO (green) and KG (blue).

Therefore the opportunity cost is given by

$$\text{Opportunity Cost} \quad = \quad \max_{a'} F(a') - F(a^*). \tag{25}$$

Code is available online for all experiments and benchmarks at http://www2.warwick.ac.uk/fac/cross_fac/complexity/people/students/dtc/students2013/pearce/.

5.1 Benchmark Methods

- *Random Sampling* Given a budget N, samples are randomly distributed over the joint input-design space by Latin Hyper Cube (in the uniform input case) or by sampling from the input distribution and latin hypercube in the A space. We consider this the simplest uninformed brute-force approach.
- *EGO on the mean and mode input* We apply a single parameter EGO to the mean input and to the mode input. This represents a typical approach used in practice, where the input uncertainty is simply reduced to using the most likely or most typical input parameter value. Technically this is equivalent to using the EGO algorithm described above with only one constant sample in the Monte-Carlo integral. In the uniform case we only use the mean input $x = 50$, and in the triangular case we use the mean $x = 66.67$ and the mode $x = 100$. At the end of sampling the best a on the single input alone is recommended while opportunity cost is measured over all test inputs.

5.2 Results

$\hat{F}^n(a)$ provides a point estimate of the true performance, $\sum \theta(x_i, a)$, averaged over a given set of inputs, X_{MC}, the posterior variance of the estimate is given by

$$\frac{1}{N_X^2} \sum_{x_i \in X_{MC}} \sum_{x_j \in X_{MC}} k^n((x_i, a), (x_j, a))$$

and is plotted in Figure 1 as confidence bounds. In Figures 1 (d) and (g) it can be seen that samples are focussed around the true optimal a and the error in $\hat{F}^{100}(a)$ is much lower around the optimal a.

In Figure 2, in both cases applying EGO to the mean input and the mode input results in the opportunity cost not decreasing to zero and the algorithm converges to the wrong design, so reducing input uncertainty to just a typical input parameter value can clearly lead to inferior solutions. In this example, the EGO focusing on the mode input even converges to a solution that is worse than the solution obtained after the initial 10 samples of the methods that take input uncertainty into account. Of the methods that account for input uncertainty, KG works best, followed by EGO and then the simple random sampling.

6 CONCLUSION

When building a simulation model, the user usually faces the challenge to define proper input distributions. Input uncertainty arises when one is not completely certain what distributions and/or parameters to use. In this paper, we proposed two simulation optimization methods based on EGO and KG ideas that are capable to take into account input uncertainty, and identify the design that has the best expected performance over the assumed known distribution of input distribution parameters.

Numerical experiments demonstrated that the new algorithms indeed sample the search space very efficiently, and are much more efficient than random sampling. The approach to simply use EGO on a typical input distribution parameter such as the mode or the mean of the assumed distribution clearly performed worse, which demonstrates the importance of properly accounting for input uncertainty.

There are various avenues for future work. While we assumed in this paper that the design space and the input distribution parameter space can each be described by a single continuous parameter, the proposed methods should also be tested in higher dimensions and with discrete parameters. It would be interesting to examine the impact of parameter N_x. Finally, one could consider worst case performance rather than expected performance.

REFERENCES

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

Branke, J., S. Meisel, and C. Schmidt. 2008. "Simulated Annealing in the Presence of Noise". *Journal of Heuristics* 14 (6): 627–654.

Chen, C.-H., S. E. Chick, L. H. Lee, and N. A. Pujowidianto. 2015. "Ranking and Selection: Efficient Simulation Budget Allocation". In *Handbook of Simulation Optimization*, 45–80. Springer.

Fan, W. L., J. Hong, and X. Zhang. 2013. "Robust Selection of the Best". In *Proceedings of the 2013 Winter Simulation Conference*, edited by R. Pasupathy, S.-H. Kim, A. Tolk, R. Hill, and M. E. Kuhl, 868–876. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

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

Gao, S., H. Xiao, E. Zhou, and W. Chen. 2016. "Optimal Computing Budget Allocation with Input Uncertainty". In *Proceedings of the 2016 Winter Simulation Conference*, edited by T. M. K. Roeder, P. I. Frazier, R. Szechtman, E. Zhou, T. Huschka, and S. E. Chick, 839–846. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Huang, D., T. Allen, W. Notz, and R. Miller. 2006. "Sequential Kriging Optimization using Multiple-Fidelity Evaluations". *Structural and Multidisciplinary Optimization* 32 (5): 369–382.

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.

Lam, H. 2016. "Advanced Tutorial: Input Uncertainty and Robust Analysis in Stochastic Simulation". In *Proceedings of the 2016 Winter Simulation Conference*, edited by T. M. K. Roeder, P. I. Frazier,

R. Szechtman, E. Zhou, T. Huschka, and S. E. Chick, 178–192. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Scott, W., P. Frazier, and W. Powell. 2011. "The Correlated Knowledge Gradient for Simulation Optimization of Continuous Parameters using Gaussian Process Regression". *SIAM Journal on Optimization* 21 (3): 996–1026.

Song, E., B. L. Nelson, and L. J. Hong. 2015. "Input Uncertainty and Indifference-Zone Ranking & Selection". In *Proceedings of the 2015 Winter Simulation Conference*, edited by L. Yilmaz, W. K. V. Chan, I. Moon, T. M. K. Roeder, C. Macal, and M. D. Rossetti, 414–424. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Zhang, X., and L. Ding. 2016. "Sequential Sampling for Bayesian Robust Ranking and Selection". In *Proceedings of the 2016 Winter Simulation Conference*, edited by T. M. K. Roeder, P. I. Frazier, R. Szechtman, E. Zhou, T. Huschka, and S. E. Chick. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Zhou, E., and W. Xie. 2015. "Simulation Optimization when Facing Input Uncertainty". In *Proceedings of the 2015 Winter Simulation Conference*, edited by L. Yilmaz, W. K. V. Chan, I. Moon, T. M. K. Roeder, C. Macal, and M. D. Rossetti, 3714–3724. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

AUTHOR BIOGRAPHIES

MICHAEL PEARCE is currently 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. His e-mail address is m.a.l.pearce@warwick.ac.uk.

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 Associate Editor for IEEE Transaction on Evolutionary Computation, for the Evolutionary Computation Journal, and for the Journal on Multi Criteria Decision Analysis. His research interests include metaheuristics, multiobjective optimisation and decision making, optimisation in the presence of uncertainty, and simulation-based optimisation, and he has published over 170 peer-reviewed papers in international journals and conferences. His e-mail address is Juergen.Branke@wbs.ac.uk.